

Oracle WebLogic Server 11g: Administration Essentials

Volume II • Student Guide

D58682GC10

Edition 1.0

July 2009

D61315

ORACLE®

Authors

Shankar Raman

Steve Friedberg

**Technical Contributors
and Reviewers**

Werner Bauer

Mike Blevins

Steve Button

David Cabelus

Shailesh Dwivedi

Will Hopkins

Bala Kothandaraman

Mike Lehmann

Serge Moiseev

Nagavalli.Pataballa

TJ Palazzolo

Holger Dindler Rasmussen

Anand Rudrabatla

Matthew Slingsby

Graphic Designer

Priya Saxena

Editors

Aju Kumar

Nita Pavitran

Raj Kumar

Publishers

Jobi Varghese

Pavithran Adka

Copyright © 2009, Oracle. All rights reserved.

Disclaimer

This document contains proprietary information and is protected by copyright and other intellectual property laws. You may copy and print this document solely for your own use in an Oracle training course. The document may not be modified or altered in any way. Except where your use constitutes "fair use" under copyright law, you may not use, share, download, upload, copy, print, display, perform, reproduce, publish, license, post, transmit, or distribute this document in whole or in part without the express authorization of Oracle.

The information contained in this document is subject to change without notice. If you find any problems in the document, please report them in writing to: Oracle University, 500 Oracle Parkway, Redwood Shores, California 94065 USA. This document is not warranted to be error-free.

Restricted Rights Notice

If this documentation is delivered to the United States Government or anyone using the documentation on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS

The U.S. Government's rights to use, modify, reproduce, release, perform, display, or disclose these training materials are restricted by the terms of the applicable Oracle license agreement and/or the applicable U.S. Government contract.

Trademark Notice

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Contents

Preface

I Introduction

- Objectives I-2
- Course Prerequisites I-3
- Course Objectives I-4
- Course Schedule I-6
- Facilities in Your Location I-8
- Summary I-9

1 Introducing Oracle Fusion Middleware Platform

- Objectives 1-2
- Oracle Fusion Middleware 1-3
- Oracle SOA and Oracle Web Center Suites 1-5
- Oracle Identity and Access Management 1-6
- Oracle Business Intelligence 1-7
- Oracle Portal Forms Reports 1-8
- Oracle Fusion Middleware Management Infrastructure 1-9
- Web Tier Components 1-10
- Relationship of Fusion Middleware Products to WebLogic Server 1-11
- Typical Oracle Fusion Middleware Environment 1-12
- Summary 1-13
- Practice 1 Overview: Logging In to the Lab Environment 1-14

2 Defining Java Enterprise Edition Terminology and Architecture

- Objectives 2-2
- Distributed Systems 2-3
- How Standards Help 2-5
- Java EE Standard 2-6
- Java EE Architecture 2-7
- Java Servlets 2-8
- SimplestServlet.java 2-9
- JavaServer Pages (JSPs) 2-10
- realsimple.jsp 2-11
- Enterprise JavaBeans (EJBs) 2-12

| | |
|--|------|
| Java Database Connectivity (JDBC) | 2-13 |
| Java Naming and Directory Interface (JNDI) | 2-14 |
| JNDI Tree | 2-15 |
| JNDI Contexts and Subcontexts | 2-17 |
| Java Transaction API (JTA) | 2-18 |
| Java Message Service (JMS) | 2-19 |
| Java Authentication and Authorization (JAAS) | 2-20 |
| Java Management Extensions (JMX) | 2-21 |
| Java EE Connector Architecture (JCA) | 2-22 |
| Client Application | 2-23 |
| Web Client | 2-24 |
| Proxy Server | 2-25 |
| Web Server | 2-26 |
| Firewalls | 2-27 |
| Application Servers | 2-28 |
| Web Application Server Configuration | 2-29 |
| Application Server Configuration | 2-30 |
| Quiz | 2-31 |
| Summary | 2-32 |
| Practice 2 Overview: Defining Terminology and Architecture | 2-33 |

3 Installing Oracle WebLogic Server 11g

| | |
|--|------|
| Objectives | 3-2 |
| Road Map | 3-3 |
| Oracle WebLogic Server Installation | 3-4 |
| System Requirements | 3-5 |
| GUI Mode Installation | 3-6 |
| Choosing or Creating a Home Directory | 3-7 |
| Registering for Support | 3-8 |
| Choosing an Installation Type and Products | 3-9 |
| Choosing the JDK and Product Directory | 3-10 |
| Installation and Summary | 3-11 |
| QuickStart | 3-12 |
| Road Map | 3-13 |
| Console and Silent Mode Installations | 3-14 |
| Postinstallation: Oracle Home | 3-15 |
| Oracle WebLogic Server Directory Structure | 3-16 |
| Setting Environment Variables | 3-18 |
| Defining Environment Variables | 3-19 |
| List of Environment Variables and Their Meanings | 3-21 |
| Documentation | 3-23 |

| | |
|--|------|
| Downloading Software from OTN | 3-24 |
| Quiz | 3-25 |
| Summary | 3-27 |
| Practice 3 Overview: Installing Oracle WebLogic Server 11g | 3-28 |

4 Configuring a Simple Domain

| | |
|---|------|
| Objectives | 4-2 |
| Road Map | 4-4 |
| Domain: Overview | 4-5 |
| Domain Diagram | 4-7 |
| Configuring a Domain | 4-8 |
| Starting the Domain Configuration Wizard | 4-10 |
| Creating a Domain Using the Domain Configuration Wizard | 4-11 |
| Creating a New WebLogic Domain and Selecting the Domain Source | 4-12 |
| Configuring Administrator Settings, Start Mode, and JDK | 4-13 |
| Customizing Advanced Configuration | 4-14 |
| Configuring the Administration and Managed Servers | 4-15 |
| Configuring Clusters and Assigning Servers to Clusters | 4-16 |
| Creating an HTTP Proxy Application and Configuring Machines | 4-18 |
| Assigning Servers to Machines | 4-20 |
| Configuring JDBC Data Sources | 4-21 |
| Testing Data Source Connections | 4-24 |
| Running Database Scripts | 4-25 |
| Configuring the JMS File Store | 4-26 |
| Customizing Application and Service Targeting Configuration | 4-28 |
| Configuring RDBMS Security Store Database | 4-29 |
| Reviewing the WebLogic Domain | 4-31 |
| Creating the WebLogic Domain | 4-32 |
| Domain Directory Structure | 4-33 |
| Road Map | 4-35 |
| JVM Run-Time Arguments | 4-36 |
| Oracle WebLogic Server Dependencies | 4-37 |
| Configuring CLASSPATH | 4-38 |
| Starting Oracle WebLogic Administration Server | 4-40 |
| Starting Administration Server Using <code>startWebLogic.sh</code> | 4-42 |
| Starting the Administration Server by Using the <code>java weblogic.Server</code> Command | 4-44 |
| Stopping the Administration Server | 4-45 |
| Quiz | 4-46 |
| Summary | 4-52 |
| Practice 4 Overview: Configuring a Simple Domain | 4-53 |

5 Configuring a Domain Using Templates

Objectives 5-2
Road Map 5-3
Custom Domain Templates 5-4
Domain Template Builder 5-6
Creating a Domain Template 5-7
Comparing Domain Templates 5-8
`wls.jar` Template 5-9
`MedRecTemplate.jar` Template 5-10
Road Map 5-11
Templates for SOA, JDeveloper, and Others 5-12
`oracle.soa_template_11.1.1.jar` Template 5-13
WLS Configuration in the Context of Other Products in the Fusion Middleware Suite 5-15
Repository Creation Utility (RCU) 5-16
SOA Installation 5-17
Quiz 5-18
Summary 5-21
Practice 5 Overview: Using a Domain Template 5-22

6 Using Administration Console and WLST

Objectives 6-2
Road Map 6-4
Benefits of Using the Administration Console 6-5
Accessing the Administration Console 6-6
Administration Console Login 6-7
Basic Navigation 6-8
Using the Help System 6-9
General Administration Console User Preferences 6-10
Setting Basic Properties 6-12
Configuration Change Management 6-14
Configuration Change Management Using the Administration Console Change Center 6-15
Domain Configuration Repository 6-16
Configuration Management Architecture 6-18
XML Schema for `config.xml` 6-20
Road Map 6-22
WebLogic Scripting Tool (WLST) 6-23
Using Jython 6-25
WLST Example 6-27

| | |
|--|------|
| WLST Command Requirements | 6-28 |
| Running WLST Scripts | 6-29 |
| Importing WLST as a Jython Module | 6-30 |
| General WLST Commands | 6-31 |
| Offline WLST Commands | 6-32 |
| Creating a Domain: Example | 6-33 |
| Online WLST Commands | 6-34 |
| Navigating JMX MBeans | 6-36 |
| Generating a WLST Script | 6-37 |
| Quiz | 6-38 |
| Summary | 6-44 |
| Practice 6 Overview: Using the Administrative Console and WLST | 6-45 |

7 Configuring Managed Servers

| | |
|--|------|
| Objectives | 7-2 |
| Road Map | 7-3 |
| Configuring Managed Servers | 7-4 |
| Creating a Managed Server with WLST | 7-5 |
| Starting Oracle WebLogic Managed Servers | 7-7 |
| Starting a Managed Server Using <code>startManagedWebLogic.sh</code> | 7-8 |
| Command-Line Requirements for Starting the Managed Server Using <code>java weblogic.Server</code> | 7-10 |
| Starting a Managed Server Using the Administration Console | 7-12 |
| Shutting Down a Server | 7-13 |
| Shutting Down a Domain | 7-14 |
| Creating a Boot Identity File | 7-16 |
| Monitoring All Servers | 7-18 |
| Customizing the View for All Servers | 7-20 |
| Monitoring Individual Servers | 7-21 |
| Demonstration | 7-22 |
| Road Map | 7-23 |
| Creating a Managed Server on a Remote Computer | 7-24 |
| <code>pack</code> and <code>unpack</code> : Examples | 7-25 |
| Road Map | 7-26 |
| Managed Server Independence (MSI) | 7-27 |
| MSI Search Order | 7-28 |
| When the Administration Server Is Down | 7-30 |
| Running Multiple WLS Instances | 7-31 |
| Quiz | 7-32 |
| Summary | 7-36 |
| Practice 7 Overview: Configuring a Managed Server | 7-37 |

8 Configuring Node Managers

| | |
|--|------|
| Objectives | 8-2 |
| Road Map | 8-3 |
| What Node Managers Can Do | 8-4 |
| Road Map | 8-6 |
| What Is a Machine? | 8-7 |
| Relationship of Machines to Other Components | 8-8 |
| Creating a Machine | 8-9 |
| Defining Names and OS of Machines | 8-10 |
| Assigning Servers to a Machine | 8-11 |
| Monitoring Machines and Servers | 8-12 |
| Configuring a Machine to Use a Node Manager | 8-13 |
| Node Manager Architecture | 8-14 |
| How a Node Manager Starts an Administration Server | 8-15 |
| How a Node Manager Starts a Managed Server | 8-16 |
| How a Node Manager Restarts an Administration Server | 8-17 |
| How a Node Manager Restarts a Managed Server | 8-18 |
| How a Node Manager Shuts Down a Server Instance | 8-19 |
| Versions of Node Managers | 8-20 |
| Road Map | 8-22 |
| Node Manager Default Behaviors | 8-23 |
| Configuring a Java-Based Node Manager | 8-24 |
| Reconfiguring the Startup Service for a Windows Installation | 8-26 |
| Node Manager as a UNIX Daemon | 8-27 |
| Reviewing <code>nodemanager.properties</code> | 8-28 |
| Configuring a Script-Based Node Manager | 8-30 |
| Creating Management OS Users | 8-31 |
| Additional Configuration Information | 8-32 |
| Configuring the <code>nodemanager.domains</code> File | 8-33 |
| Defining the Administration Server Address | 8-34 |
| Setting Node Manager Environment Variables | 8-35 |
| Node Manager Configuration and Log Files | 8-36 |
| Quiz | 8-40 |
| Summary | 8-43 |
| Practice 8 Overview: Configuring Machines and Node Managers | 8-44 |

| |
|--|
| 9 Viewing and Managing Logs in Oracle WLS Environment |
| Objectives 9-2 |
| Road Map 9-3 |
| Oracle WebLogic Server Logs 9-4 |
| Configuring Server Logging 9-7 |
| Configuring Server Logging: Advanced 9-8 |
| HTTP Access Logs 9-10 |
| Apache Commons Logging API 9-11 |
| Using the Console to View Logs 9-12 |
| Using WLST to View Logs 9-13 |
| Message Attributes 9-14 |
| Message Severity 9-15 |
| Message Catalog Using the Web 9-16 |
| Message Catalog Cross-Reference 9-17 |
| Road Map 9-18 |
| Creating a Log Filter 9-20 |
| Applying a Log Filter 9-21 |
| Using the Console to Monitor 9-22 |
| Monitoring Running Servers 9-23 |
| Customizing Views 9-24 |
| Monitoring Individual Servers 9-25 |
| Network-Addressing Features 9-26 |
| Quiz 9-28 |
| Summary 9-29 |
| Practice 9 Overview: Viewing and Managing WLS Logs 9-30 |

10 Deployment Concepts

| |
|--|
| Objectives 10-2 |
| Road Map 10-3 |
| Overview of Deployment 10-4 |
| What Is Deployed? 10-5 |
| Deployment Process 10-7 |
| Deployment Methods 10-8 |
| Deployment Tools 10-9 |
| Console Deployment Method 10-10 |
| Console Deployment Production Mode 10-11 |
| Preparing a New Application 10-12 |
| Preparing a New Application: Targeting 10-13 |
| Preparing a New Application: Settings 10-14 |
| Deploying or Undeploying Applications 10-15 |
| Redeploying an Application 10-16 |

| | |
|---|-------|
| Starting and Stopping an Application | 10-17 |
| Editing Deployment Descriptors | 10-18 |
| Monitoring an Application | 10-19 |
| Application Testing | 10-20 |
| Deleting Applications | 10-21 |
| Command-Line Deployment | 10-22 |
| Deployment with <code>weblogic.Deployer</code> | 10-23 |
| More <code>weblogic.Deployer</code> Examples | 10-24 |
| Deploying Applications with WLST | 10-25 |
| Deploying an Application with WLST | 10-26 |
| Deployment with WLST | 10-27 |
| Road Map | 10-28 |
| Autodeployment | 10-29 |
| Autodeploying Using an Expanded Directory | 10-30 |
| FastSwap and On-Demand Deployment | 10-31 |
| Production Mode Flag | 10-33 |
| Road Map | 10-34 |
| Role of Web Servers | 10-35 |
| A Typical Web Interaction | 10-36 |
| MIME Types | 10-38 |
| HTTP Status Codes | 10-39 |
| Static Content | 10-40 |
| Dynamic Content | 10-41 |
| Configuring Oracle HTTP Server to Serve Multiple WebLogic Servers | 10-42 |
| <code>mod_wl_ohs.conf</code> | 10-43 |
| Verifying Ports Used by OHS | 10-44 |
| Quiz | 10-45 |
| Summary | 10-48 |

11 Deploying Java EE Applications

| | |
|---|-------|
| Objectives | 11-2 |
| Road Map | 11-3 |
| Java EE Web Applications | 11-4 |
| Packaging Web Applications | 11-6 |
| Web Application Structure | 11-7 |
| Web Application Archive | 11-8 |
| Optional Configuration of Web Applications | 11-9 |
| <code>web.xml</code> | 11-10 |
| <code>weblogic.xml</code> | 11-11 |
| <code>weblogic.xml</code> Deployment Descriptor | 11-12 |

| | |
|---|-------|
| URLs and Web Applications | 11-13 |
| Virtual Directory Mappings | 11-15 |
| Virtual Directory Mapping: Example | 11-16 |
| Road Map | 11-17 |
| Types of EJBs | 11-19 |
| EJB Application Structure | 11-21 |
| <code>weblogic-ejb-jar.xml</code> | 11-22 |
| Administrator Tasks with EJBs | 11-23 |
| Road Map | 11-24 |
| What Is an Enterprise Application? | 11-25 |
| A Typical Java EE System | 11-26 |
| Java EE Enterprise Application | 11-27 |
| Why Enterprise Applications? | 11-29 |
| Enterprise Application Structure | 11-30 |
| <code>weblogic-application.xml</code> | 11-31 |
| Application Scoping | 11-32 |
| EAR Class Libraries | 11-33 |
| Java EE Library Support | 11-34 |
| WebLogic Java EE Shared Libraries | 11-35 |
| Quiz | 11-37 |
| Summary | 11-40 |
| Practice 11 Overview: Web Application Deployment Concepts | 11-41 |

12 Advanced Deployment

| | |
|---|-------|
| Objectives | 12-2 |
| Road Map | 12-3 |
| What Is a Deployment Plan? | 12-4 |
| Configuring an Application for Multiple Deployment Environments | 12-5 |
| Sample Deployment Plan | 12-7 |
| Creating a Deployment Plan | 12-8 |
| Creating a New Deployment Plan | 12-10 |
| <code>weblogic.PlanGenerator</code> | 12-11 |
| Using the Administration Console to Generate a Deployment Plan | 12-12 |
| Modifying and Saving Data to Create a New Plan | 12-13 |
| New Deployment Plan Shows Changed Values | 12-14 |
| Using an Existing Deployment Plan to Configure an Application | 12-15 |
| Using an Existing Deployment Plan | 12-17 |
| Generic File-Loading Overrides | 12-18 |
| Directory Structure for Easier Production Deployment | 12-19 |
| Performing a Sanity Check in Production Without Disruption to the Clients | 12-20 |

| | |
|---|-------|
| Road Map | 12-21 |
| Staged Deployment | 12-22 |
| Road Map | 12-23 |
| Application Availability | 12-24 |
| Production Redeployment and Application Versioning | 12-25 |
| WebLogic Production Redeployment | 12-27 |
| Production Redeployment | 12-28 |
| Advantages of Production Redeployment | 12-29 |
| Requirements and Restrictions for Production Redeployment | 12-30 |
| Redeploying a New Application Version | 12-31 |
| Redeploying Versus Distributing | 12-32 |
| Distributing a New Version of the Production Application | 12-33 |
| Distributing a New Application Version | 12-35 |
| Production Redeployment | 12-36 |
| Quiz | 12-37 |
| Summary | 12-40 |
| Practice 12 Overview: Deploying Production Applications | 12-41 |

13 Understanding JDBC and Configuring Data Sources

| | |
|---|-------|
| Objectives | 13-2 |
| Road Map | 13-3 |
| JDBC Review | 13-4 |
| JDBC Data Sources | 13-5 |
| Data Source Scope | 13-6 |
| Multi-Tier Architecture | 13-7 |
| Type 4 Drivers | 13-8 |
| WebLogic JDBC Drivers | 13-9 |
| Road Map | 13-10 |
| What Is a Connection Pool? | 13-11 |
| JDBC Connection Pooling | 13-12 |
| Benefits of Connection Pools | 13-13 |
| Modular Configuration and Deployment of JDBC Resources | 13-14 |
| How Data Source Connection Pools Are Used | 13-15 |
| Creating a Data Source Using the Administration Console | 13-16 |
| Non-XA Configuration | 13-17 |
| Data Source Connection Properties | 13-18 |
| Test Configuration | 13-19 |
| Connection Pool Configuration | 13-20 |
| Connection Pool Advanced | 13-21 |
| Targeting a Data Source | 13-22 |
| Viewing the Server JNDI Tree via the Administration Console | 13-23 |

| | |
|---|-------|
| Listing the JNDI Contents via WLST | 13-24 |
| Demonstration | 13-25 |
| JDBC URLs | 13-26 |
| Connection Properties | 13-27 |
| Specifying Connection Properties | 13-28 |
| Road Map | 13-29 |
| Monitoring and Testing a Data Source | 13-30 |
| Connection Pool Life Cycle | 13-31 |
| Quiz | 13-32 |
| Summary | 13-35 |
| Practice 13 Overview: Configuring JDBC Data Sources | 13-36 |

14 Setting Up Java Message Service (JMS) Resources

| | |
|---|-------|
| Objectives | 14-2 |
| Road Map | 14-3 |
| Message-Oriented Middleware | 14-4 |
| Point-To-Point Queue | 14-5 |
| Publish/Subscribe Topics | 14-6 |
| Oracle WebLogic Server JMS Features | 14-7 |
| Oracle WLS JMS Architecture | 14-9 |
| Typical JMS Messaging Process | 14-10 |
| Transacted Messaging | 14-11 |
| JMS Administrative Tasks | 14-12 |
| Oracle WLS JMS Implementation | 14-13 |
| Road Map | 14-14 |
| Oracle WLS JMS Server | 14-15 |
| Creating a JMS Server | 14-16 |
| Configuring a JMS Server | 14-17 |
| Targeting a JMS Server to a Managed Server | 14-18 |
| JMS Modules | 14-19 |
| Modular JMS Resource Configuration and Deployment | 14-21 |
| Connection Factories | 14-22 |
| Creating a Connection Factory | 14-24 |
| Configuring a Connection Factory | 14-25 |
| Destination | 14-26 |
| Queue Destinations | 14-27 |
| Topic Destinations | 14-28 |
| Creating a Destination (Topic) | 14-29 |
| Threshold, Quota, and Paging | 14-31 |
| Configuring Thresholds and Quotas | 14-32 |
| Road Map | 14-33 |

| | |
|--|-------|
| Durable Subscribers and Subscriptions | 14-34 |
| How a Durable Subscription Works | 14-35 |
| Configuring a Durable Subscription | 14-36 |
| Persistent Messaging | 14-37 |
| Creating a JMS Store | 14-38 |
| Creating a JDBC Store for JMS | 14-39 |
| Creating a JMS JDBC Store | 14-40 |
| Assigning a Store to a JMS Server | 14-41 |
| Persistent Connection Factory | 14-42 |
| Configuring Destination Overrides | 14-43 |
| Road Map | 14-44 |
| Monitoring JMS Servers | 14-45 |
| Monitoring and Managing Destinations | 14-46 |
| Monitoring Queues | 14-47 |
| Viewing Active Queues and Topics | 14-48 |
| Managing Messages in a Queue | 14-49 |
| Quiz | 14-50 |
| Summary | 14-52 |
| Practice Overview: Configuring JMS Resources | 14-53 |

15 Introduction to Clustering

| | |
|--|-------|
| Objectives | 15-2 |
| Road Map | 15-3 |
| What Is a Cluster? | 15-4 |
| Benefits of Clustering | 15-5 |
| What Can Be Clustered | 15-6 |
| Proxy Servers for HTTP Clusters | 15-7 |
| High Availability for EJBs | 15-8 |
| Clustering EJB Objects: Replica-Aware Stub | 15-9 |
| EJB: Server Failure Situations | 15-10 |
| Load-Balancing Clustered EJB Objects | 15-11 |
| Stateless Session Bean Failover | 15-12 |
| Road Map | 15-13 |
| Selecting a Cluster Architecture | 15-14 |
| Cluster Architecture | 15-15 |
| Basic Cluster Architecture | 15-16 |
| Basic Cluster Architecture: Advantages and Disadvantages | 15-17 |
| Multitier Cluster Architecture | 15-18 |
| Multitier: Advantages and Disadvantages | 15-19 |
| Basic Cluster Proxy Architecture | 15-21 |
| Multitier Cluster Proxy Architecture | 15-22 |

| | |
|---|-------|
| Proxy Web Server Plug-In Versus Load Balancer | 15-23 |
| Proxy Plug-Ins | 15-24 |
| OHS as Proxy Web Server | 15-25 |
| Request Flow When Using OHS | 15-26 |
| WLS <code>HttpClusterServlet</code> | 15-27 |
| Road Map | 15-28 |
| Server Communication in a Cluster | 15-29 |
| One-to-Many Communications | 15-31 |
| Considerations When Using Unicast | 15-33 |
| Peer-to-Peer Communications | 15-34 |
| Clusterwide JNDI Naming Service | 15-35 |
| Name Conflicts and Resolution | 15-36 |
| Quiz | 15-37 |
| Summary | 15-39 |

16 Configuring a Cluster

| | |
|--|-------|
| Objectives | 16-2 |
| Road Map | 16-3 |
| Preparing Your Environment | 16-4 |
| Hardware | 16-5 |
| IP Addresses and Host Names | 16-6 |
| Cluster Address | 16-7 |
| Road Map | 16-8 |
| Methods of Configuring Clusters | 16-9 |
| Creating a Cluster by Using the Administration Console | 16-10 |
| Setting Cluster Attributes | 16-12 |
| Configuring Cluster Communication | 16-13 |
| Adding Cluster Members: Option 1 | 16-14 |
| Adding Cluster Members: Option 2 | 16-15 |
| Creating a Cluster with the Configuration Wizard | 16-16 |
| Clusters and the Configuration Wizard | 16-17 |
| Clusters and WLST | 16-18 |
| Creating a Cluster Using the Cluster MBean | 16-19 |
| Synchronization When Starting Servers in a Cluster | 16-20 |
| Configuring OHS as Proxy Server | 16-22 |
| Starting and Stopping OHS Manually | 16-23 |
| Verifying Access Through OHS | 16-24 |
| Quiz | 16-25 |
| Summary | 16-26 |
| Practice 16 Overview: Configuring Clusters | 16-27 |

17 Managing Clusters

- Objectives 17-2
- Road Map 17-3
- Deploying Applications to a Cluster 17-4
- Two-Phase Deployment 17-5
- Considerations for Deploying to Cluster 17-6
- Production Redeployment in a Cluster 17-7
- Road Map 17-8
- HTTP Session State Replication 17-10
- HTTP Session: In-Memory Replication 17-11
- In-Memory Replication 17-14
- Requirements for In-Memory Replication 17-15
- Configuring In-Memory Replication 17-16
- HTTP Session: Replication Using JDBC 17-18
- HTTP Session Replication Using JDBC 17-19
- Configuring JDBC Replication 17-20
- JDBC Persistent Table Configuration 17-21
- HTTP Session Replication Using File 17-23
- Configuring File Replication 17-24
- Replication Groups 17-26
- Configuring Replication Groups 17-28
- Failover with Replication Groups 17-29
- HTTP State Management Best Practices 17-30
- Road Map 17-31
- Configuring EJB Clustering in Deployment Descriptors 17-32
- Configuring EJB Clustering Using the Administration Console 17-33
- Configuring Clusterable Stateless Session EJBs 17-34
- Clusterable EJBs: Idempotent Methods 17-35
- Stateful Session Beans 17-36
- Configuring Clusterable Stateful Session EJBs 17-37
- Read/Write Versus Read-Only 17-38
- Entity Bean Cluster-Aware Home Stubs 17-39
- EJB Best Practices 17-40
- Quiz 17-41
- Summary 17-45
- Practice 17: Overview Managing Clusters 17-46

18 Security Concepts and Configuration

- Objectives 18-2
- Road Map 18-3
- Introduction to Oracle WebLogic Security Service 18-4

| | |
|--|-------|
| Oracle Platform Security Services | 18-5 |
| Oracle WLS Security Architecture | 18-6 |
| Security Services | 18-7 |
| Overview of Security Concepts | 18-8 |
| Confidentiality | 18-9 |
| Credential Mapping | 18-11 |
| Road Map | 18-12 |
| Security Realms | 18-13 |
| Security Model Options for Applications | 18-14 |
| How WLS Resources Are Protected | 18-16 |
| Users and Groups | 18-17 |
| Configuring New Users | 18-18 |
| Groups | 18-19 |
| Configuring New Groups | 18-20 |
| Configuring Group Memberships | 18-21 |
| Road Map | 18-22 |
| Security Roles | 18-23 |
| Configuring the Global Security Role | 18-25 |
| Security Policies | 18-26 |
| Policy Conditions | 18-27 |
| Protecting Web Applications | 18-28 |
| Specifying Protected Web Resources | 18-29 |
| Defining Policies and Roles for Other Resources | 18-30 |
| Embedded LDAP Server | 18-31 |
| Configuring an Embedded LDAP | 18-32 |
| Configuring Authentication | 18-34 |
| Authentication Examples | 18-35 |
| Migrating Security Data | 18-36 |
| Exporting the WLS Default Authenticator Provider | 18-38 |
| Importing into a Different Domain | 18-39 |
| Summary | 18-40 |
| Practice 18: Overview Configuring Security for WLS Resources | 18-41 |

19 Protecting Against Attacks

| | |
|---|-------|
| Objectives | 19-2 |
| Road Map | 19-3 |
| What Is SSL? | 19-4 |
| Trust and Identity | 19-5 |
| Using an SSL Connection | 19-6 |
| Enabling Secure Communication | 19-8 |
| Oracle WebLogic Server SSL Requirements | 19-10 |

keytool Utility 19-11
Obtaining a Digital Certificate: keytool Examples 19-12
Configuring Keystores 19-14
Configuring SSL for an Oracle WebLogic Server 19-15
Road Map 19-16
Protecting Against Attacks 19-17
Man-in-the-Middle Attacks 19-18
Man-in-the-Middle: Countermeasures 19-19
Configuring a Hostname Verifier 19-21
Denial of Service Attacks 19-22
Denial of Service Attacks: Countermeasures 19-23
Filtering Network Connections 19-24
Connection Filter 19-25
Excessive Resource Consumption 19-26
Large Buffer Attacks 19-27
Setting the Post Size 19-28
Connection Starvation 19-29
User Lockout 19-31
Configuring User Lockout 19-32
Unlocking Users 19-33
Protecting the Administration Console 19-34
Quiz 19-35
Summary 19-37
Practice 19: Overview 19-38

20 Backup and Recovery Operations

Objectives 20-2
Road Map 20-3
Review of Terms and Components 20-4
Homes: Oracle, Middleware, WebLogic 20-6
Understanding Backup and Recovery 20-7
Types of Backups 20-9
Backup Recommendations 20-11
Limitations and Restrictions for Backing Up Data 20-12
Performing a Full Offline Backup 20-13
Backing Up a Domain Configuration 20-15
Backing Up an Instance Home 20-16
Creating a Record of Installations 20-17
Road Map 20-18
Directories to Restore 20-19
Recovery After Disaster 20-20

- Recovery of Homes 20-21
- Recovery of a Managed Server 20-22
- Recovery of the Administration Server Configuration 20-23
- Restarting an Administration Server on a New Computer 20-24
- Recovery of a Cluster 20-25
- Restoring OPMN-Managed Components to a New Computer 20-26
- Quiz 20-27
- Summary 20-32
- Practice 20 Overview: Backing Up and Restoring Configuration and Data 20-33

A Practices and Solutions

Glossary

Index

Preface

Profile

Before You Begin This Course

Before you begin this course, you should be able to

- Issue basic UNIX user-level commands
- Perform UNIX desktop navigation tasks
- Describe basic XML concepts
- Describe basic TCP/IP networking client/server concepts

How This Course Is Organized

Oracle WebLogic Server 11g: Administration Essentials is an instructor-led course featuring lectures and hands-on exercises. Online demonstrations and written practice sessions reinforce the concepts and skills that are introduced.

Related Publications

Oracle Publications

| Title | Part Number |
|---|-------------|
| <i>Oracle Fusion Middleware Online Documentation Library 11g Release 1 (11.1.1)</i> | E12839-01 |
| <i>Oracle Fusion Middleware Administrator's Guide 11g Release 1 (11.1.1)</i> | E10105-01 |
| <i>Oracle Fusion Middleware Upgrade Planning Guide 11g Release 1 (11.1.1)</i> | E10125-01 |
| <i>Oracle Fusion Middleware Administrator's Guide for Oracle Web Cache 11g Release 1 (11.1.1)</i> | E10143-01 |
| <i>Oracle Fusion Middleware Administrator's Guide for Oracle HTTP Server 11g Release 1 (11.1.1)</i> | E10144-01 |

Additional Publications

- System release bulletins
- Installation and user's guides
- *read.me* files
- International Oracle User's Group (IOUG) articles
- *Oracle Magazine*

Typographic Conventions

The following two lists explain Oracle University typographical conventions for words that appear within regular text or within code samples.

1. Typographic Conventions for Words Within Regular Text

| Convention | Object or Term | Example |
|-----------------|---|---|
| Courier New | User input; commands; column, table, and schema names; functions; PL/SQL objects; paths | Use the SELECT command to view information stored in the LAST_NAME column of the EMPLOYEES table. Enter 300. Log in as scott |
| Initial cap | Triggers; user interface object names, such as button names | Assign a When-Validate-Item trigger to the ORD block. Click the Cancel button. |
| Italic | Titles of courses and manuals; emphasized words or phrases; placeholders or variables | For more information on the subject see <i>Oracle SQL Reference Manual</i> Do <i>not</i> save changes to the database. Enter <i>hostname</i> , where <i>hostname</i> is the host on which the password is to be changed. |
| Quotation marks | Lesson or module titles referenced within a course | This subject is covered in Lesson 3, “Working with Objects.” |

Typographic Conventions (continued)

2. Typographic Conventions for Words Within Code Samples

| Convention | Object or Term | Example |
|----------------------|---|---|
| Uppercase | Commands, functions | <code>SELECT employee_id FROM employees;</code> |
| Lowercase, italic | Syntax variables | <code>CREATE ROLE <i>role</i>;</code> |
| Initial cap | Forms triggers | <code>Form module: ORD Trigger level: S_ITEM.QUANTITY item Trigger name: When-Validate-Item . . .</code> |
| Lowercase | Column names, table names, filenames, PL/SQL objects | <code>. . . OG_ACTIVATE_LAYER (OG_GET_LAYER ('prod_pie_layer')) . . . SELECT last_name FROM employees;</code> |
| Bold | Text that must be entered by a user | <code>CREATE USER scott IDENTIFIED BY tiger;</code> |

13

Understanding JDBC and Configuring Data Sources

ORACLE®

Copyright © 2009, Oracle. All rights reserved.

Objectives

After completing this lesson, you should be able to:

- Configure JDBC and JDBC data sources
- Configure data source scope
- Contrast two-tier and multi-tier JDBC architecture
- Configure a connection pool
- List the benefits of connection pools
- Describe how data sources are used
- Deploy JDBC resources to a target
- View the server JNDI tree
- Complete a connection pool checklist
- Explain the components of JDBC URLs
- Monitor and test a data source



Copyright © 2009, Oracle. All rights reserved.

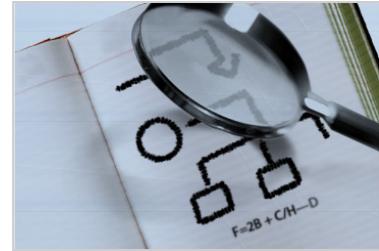
Objectives

Scenario

The Medical Records application needs to store data in a relational database. The application programmers do not have experience with the particular database vendor that you have chosen, but are familiar with SQL from another vendor. They want to isolate the vendor- and platform-specific commands and write generic SQL that would work against any kind of relational database. Eventually, they plan to migrate to Oracle Database, and would like to preserve all of their work now as being vendor-agnostic.

Road Map

- Overview of JDBC
 - High-level architecture of JDBC and the driver model
 - Design of a multi-tier architecture
 - Drivers provided by Oracle WebLogic Server
- Data sources
- Monitoring and testing data sources

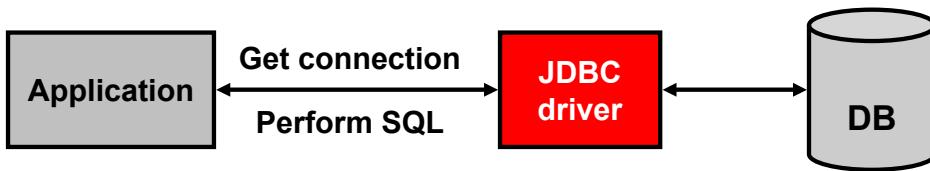


ORACLE®

Copyright © 2009, Oracle. All rights reserved.

JDBC Review

- The Java Database Connectivity (JDBC) specification:
 - Is a platform- and vendor-independent mechanism for accessing and updating a database
 - Provides transparency from proprietary vendor issues
 - Requires the use of a *driver*
- JDBC drivers are supplied by WebLogic Server or by your database vendor.



ORACLE

Copyright © 2009, Oracle. All rights reserved.

JDBC Review

The JDBC API is a natural Java interface for working with SQL. It builds on Open Database Connectivity (ODBC) rather than starting from the beginning, so programmers familiar with ODBC find it very easy to learn.

The value of JDBC lies in the fact that an application can access virtually any data source and run on any platform with a Java Virtual Machine (JVM). That is, with JDBC, it is not necessary to write one program to access a Sybase database, another to access an Oracle database, another to access an IBM DB2 database, and so on. You can write a single program using the JDBC API. Because the application is written in Java, you need not write different applications to run on different platforms, such as Windows and Linux.

JDBC accomplishes database connections by using a driver mechanism that translates the JDBC calls to native database calls. Although most available drivers are fully written in Java (Type 4) and are thus platform-independent, some drivers (Type 2) use native libraries and are targeted to specific platforms.

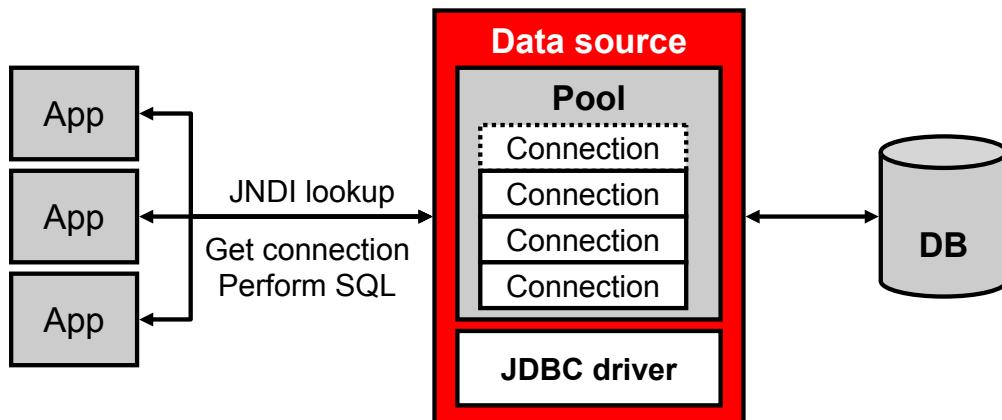
Oracle WebLogic Server includes several Type 4 JDBC drivers, which are compliant with the JDBC 3.0 specification. In addition, the Type 4 drivers support the following JDBC 4.0 specification features:

- Connection validation
- Client information storage and retrieval
- Auto-load driver classes (when using Java Platform, Standard Edition 6 (Java SE 6))

JDBC Data Sources

Data sources:

- Enable database connectivity to be managed by the application server
- Are obtained by applications from the server's JNDI tree
- Use a dynamic pool of reusable database connections



ORACLE

Copyright © 2009, Oracle. All rights reserved.

JDBC Data Sources

Oracle WebLogic Server can manage your database connectivity through JDBC data sources and multidata sources. Each data source that you configure contains a pool of database connections that are created when the data source instance is created—when it is deployed or targeted, or at server startup. The connection pool can grow or shrink dynamically to accommodate the demand, as indicated by the dotted connection at the top of the pool.

Applications look up a data source on the Java Naming and Directory Interface (JNDI) tree or in the local application context (`java:comp/env`), depending on how you configure and deploy the object, and then request a database connection. When finished with the connection, the application uses the close operation on the connection, which simply returns the connection to the connection pool in the data source.

Oracle WebLogic Server data sources allow connection information such as the JDBC driver, the database location (URL), and the username and password to be managed and maintained in a single location, without requiring the application to worry about these details. In addition, limiting the number of connections is important if you have a licensing limitation on your database or it can support only a specific capacity.

Data Source Scope

- Each data source configuration or “module” is persisted as a separate XML document.
- The system modules that are created with the console or WLST are:
 - Stored in the domain’s config/jdbc directory
 - Available to all applications in the domain
- Application-specific modules are:
 - Deployed as part of Java Platform, Enterprise Edition (Java EE) enterprise applications
 - Accessible only by the containing application



Copyright © 2009, Oracle. All rights reserved.

Data Source Scope

Both Oracle WebLogic Server administrators and developers can define the JDBC data sources. Regardless of which approach you take, each JDBC data source is represented by an XML file that is called a module. The concept of scope is useful when there is a potential namespace clash. For example, if developer 1 makes application 1 that uses data source X, and developer 2 makes application 2 that also uses a *different* data source X, then the scope is set at an application level. (You may wonder why not name it 1X and 2X, but that is beside the point.) Alternatively, if both application 1 and application 2 wanted to use the *same* data source X, it would be scoped at the server level by the administrator.

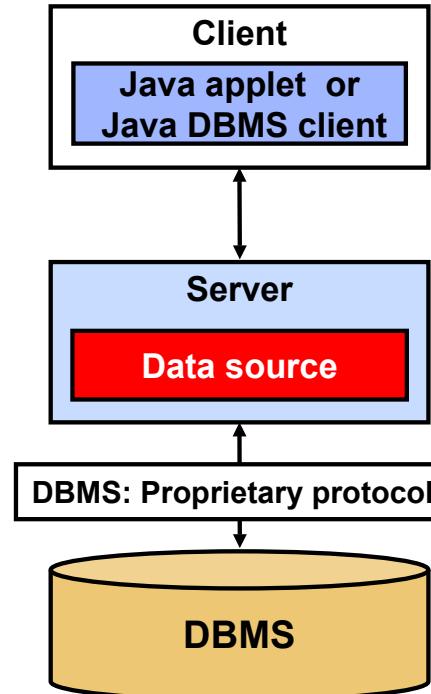
WebLogic administrators typically use the Administration Console or the WebLogic Scripting Tool (WLST) to create and deploy (target) JDBC modules. These JDBC modules are considered system modules, are stored in the domain’s configuration repository as separate XML files, and are referred to by the domain’s config.xml file.

Alternatively, developers define data sources in XML descriptor files, and then package the JDBC modules within a Java EE enterprise application for administrators to deploy. These JDBC modules are considered application modules. Because the modules are deployment descriptors, they can also be modified for different environments using Java EE deployment plans.

All WebLogic JDBC module files must end with the -jdbc.xml suffix, such as examples-demo-jdbc.xml. Oracle WebLogic Server checks the file name when you deploy the module.

Multi-Tier Architecture

- In the multi-tier model, commands are sent to a “middle tier” of services, which then sends the commands to the DBMS.
- The DBMS processes the commands and sends the results back to the middle tier, which then sends them to the client.



ORACLE

Copyright © 2009, Oracle. All rights reserved.

Multi-Tier Architecture

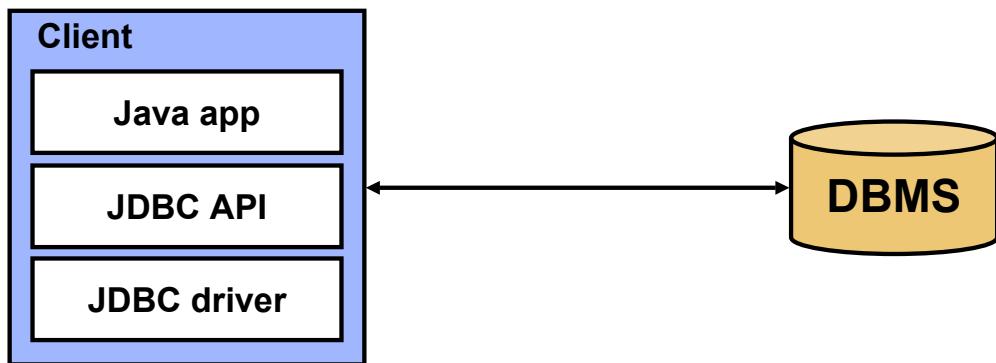
The middle tier makes it possible to maintain control over access and the kinds of updates that can be made to corporate data. Another advantage is that it simplifies the deployment of applications. Finally, in many cases, the multi-tier architecture can provide performance advantages.

Until recently, the middle tier has typically been written in languages such as C or C++, which offer fast performance. However, with the introduction of optimizing compilers that translate Java bytecode into efficient machine-specific code and technologies, such as Enterprise JavaBeans, the Java platform is fast becoming the standard platform for middle-tier development. This is a big plus, making it possible to take advantage of Java’s multithreading and security features.

With enterprises increasingly using the Java programming language for writing server code, the JDBC API is being used more and more in the middle tier of a three-tier architecture. Some of the features that make JDBC a server technology are its support for connection pooling, distributed transactions, and disconnected rowsets. The JDBC API is what allows access to a data source from a Java middle tier.

Type 4 Drivers

Type 4 drivers are “all-Java” driver implementations that do not require client-side configuration.



ORACLE

Copyright © 2009, Oracle. All rights reserved.

Type 4 Drivers

A Type 4 driver is a database driver that is written in 100% pure Java. Drivers that are written in Java have all the performance benefits because they do not have the extra layers between the program and the database. They can operate on any platform and can be downloaded from a server (when using an applet, for example). Because the driver can be downloaded from a server, the client machine does not require preconfiguration of a native driver. This preconfiguration is why the Type 1, 2, and 3 drivers are now deprecated. All that remains is Type 4 drivers.

WebLogic JDBC Drivers

- Oracle and third-party drivers are included in the WLS installation for many popular database products:
 - Oracle 9i, 10g, and 11g
 - Sybase Adaptive Server
 - Microsoft SQL Server
 - IBM DB2
 - Informix
 - MySQL
 - PointBase
- By default, these drivers are added to the server's classpath.

The red bar spans most of the width of the slide, centered horizontally.

ORACLE

Copyright © 2009, Oracle. All rights reserved.

WebLogic JDBC Drivers

The WebLogic Type 4 JDBC drivers are installed with Oracle WebLogic Server in the `<WL_HOME>/server/lib` folder, where `<WL_HOME>` is the directory in which you installed Oracle WebLogic Server. Driver class files are included in the manifest classpath in `weblogic.jar`, so the drivers are automatically added to your classpath on the server.

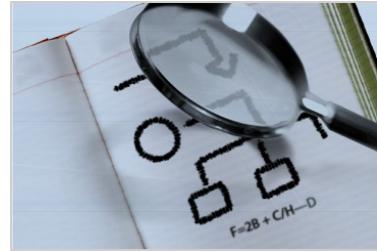
The WebLogic Type 4 JDBC drivers are installed by default when you perform a complete installation of Oracle WebLogic Server. If you choose a custom installation, ensure that the WebLogic JDBC Drivers check box is selected. If this option is deselected, the drivers are not installed.

The WebLogic Type 4 JDBC drivers included with Oracle WebLogic Server are provided from DataDirect.

This release includes support for Oracle 11g and 11g Real Application Clusters (RAC). Support for 11g RAC continues to rely on the well-proven integration architecture using multidata sources for X/Open Distributed Transaction Processing (XA) with load balancing.

Road Map

- Overview of JDBC
- Data sources
 - Describing a data source and how it works
 - Using the Administration Console to create a data source
- Monitoring and testing data sources



ORACLE®

Copyright © 2009, Oracle. All rights reserved.

What Is a Connection Pool?

- A connection pool is a group of ready-to-use database connections associated with a data source.
- Connection pools:
 - Are created at Oracle WebLogic Server startup
 - Can be administered using the Administration Console
 - Can be dynamically resized to accommodate increasing or decreasing load



ORACLE®

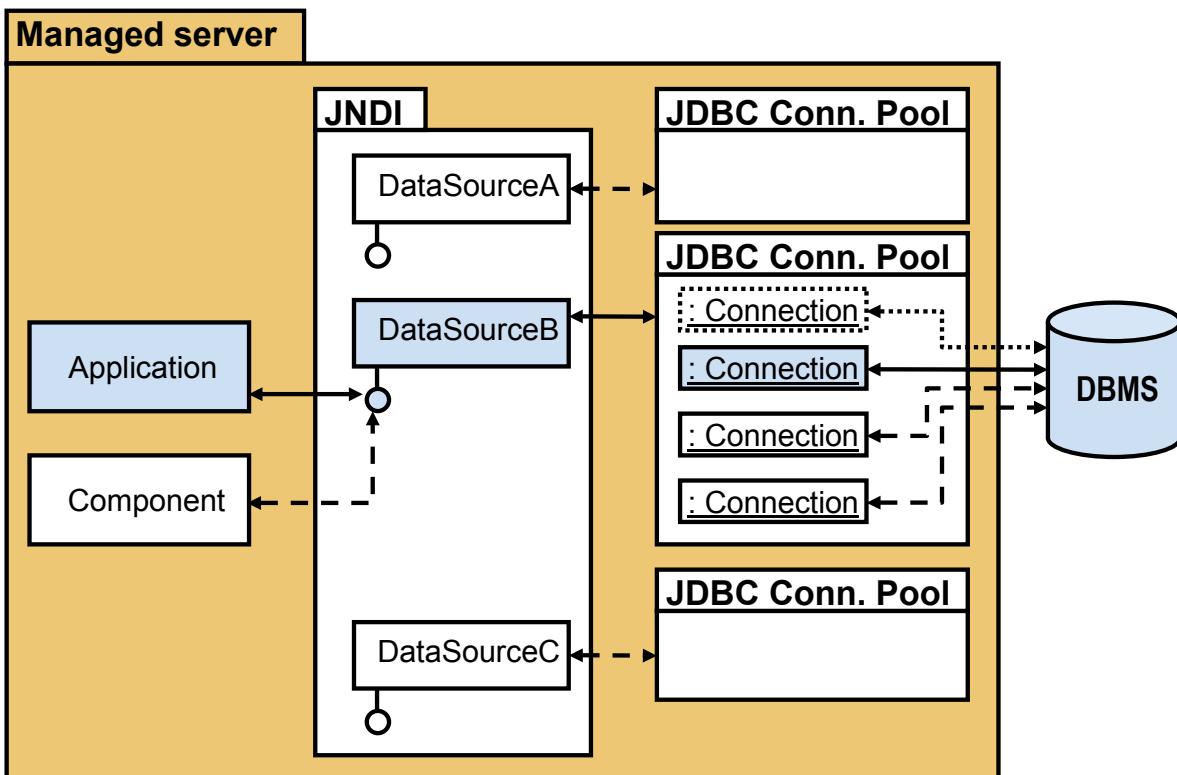
Copyright © 2009, Oracle. All rights reserved.

What Is a Connection Pool?

Oracle WebLogic Server opens JDBC connections to the database during the WebLogic startup process and adds the connections to the pool. This is faster than creating a new connection on demand. The size of the pool is dynamic and can be fine-tuned.

The connection pool within a JDBC data source contains a group of JDBC connections that applications reserve, use, and then return to the pool. The connection pool and the connections within it are created when the connection pool is registered, usually when starting up Oracle WebLogic Server or when deploying the data source to a new target.

JDBC Connection Pooling



ORACLE

Copyright © 2009, Oracle. All rights reserved.

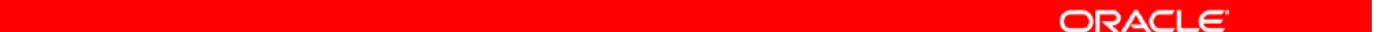
JDBC Connection Pooling

The diagram in the slide shows the flow from the applications through the JNDI tree, through the JDBC connection pools, and finally to the database. In Oracle WebLogic Server, you can configure database connectivity by configuring the JDBC data sources and the multi-data sources, and then targeting or deploying the JDBC resources to the servers or clusters in your WebLogic domain.

Each data source that you configure contains a pool of database connections that are created when the data source instance is created—when it is deployed or targeted, or at server startup. Applications look up a data source on the JNDI tree or in the local application context (`java:comp/env`), depending on how you configure and deploy the object, and then request a database connection. When finished with the connection, the application calls `connection.close()`, which returns the connection to the connection pool in the data source.

Benefits of Connection Pools

- The following are some advantages of connection pooling:
 - Connection time and overhead are saved by using an existing database connection.
 - It facilitates easier management because connection information is managed in one location.
 - The number of connections to a database can be controlled.
 - The DBMS can be changed without the application developer having to modify the underlying code.
- A connection pool allows an application to “borrow” a DBMS connection.



ORACLE

Copyright © 2009, Oracle. All rights reserved.

Benefits of Connection Pools

Making a DBMS connection is a very slow process when compared to assigning an existing connection. When Oracle WebLogic Server starts, connections from the connection pools are opened and are available to all clients. When a client closes a connection from a connection pool, the connection is returned to the pool and is available for other clients; the connection itself is not closed. There is little cost in opening and closing pool connections. The alternative is for application code to make its own JDBC connections as needed. A DBMS runs faster with dedicated connections than if it has to handle incoming connection attempts at run time.

Connection information, such as the JDBC driver class name, the database location (URL), and the username and password can be managed in one location using the Administration Console. Application developers can obtain a connection without having to worry about these details.

Limiting the number of DBMS connections is important if you have a licensing limitation for DBMS connections or a resource concern.

Clients use a connection pool by “borrowing” a connection, using it, and then returning it to the pool by closing it. The connection pool can grow or shrink dynamically to accommodate demand. The Administration Console is used to set a connection pool’s *initial capacity*, *maximum capacity*, and *capacity increment*.

Modular Configuration and Deployment of JDBC Resources

- The JDBC configurations in WebLogic Server are stored in XML documents:
 - All JDBC configurations must conform to the new `weblogic-jdbc.xsd` schema.
 - IDEs and other tools can validate the JDBC modules based on the schema.
- You create and manage JDBC resources either as system modules or as application modules.
- The JDBC application modules are a WLS-specific extension of Java EE modules and can be deployed either within a Java EE application or as stand-alone modules.



Copyright © 2009, Oracle. All rights reserved.

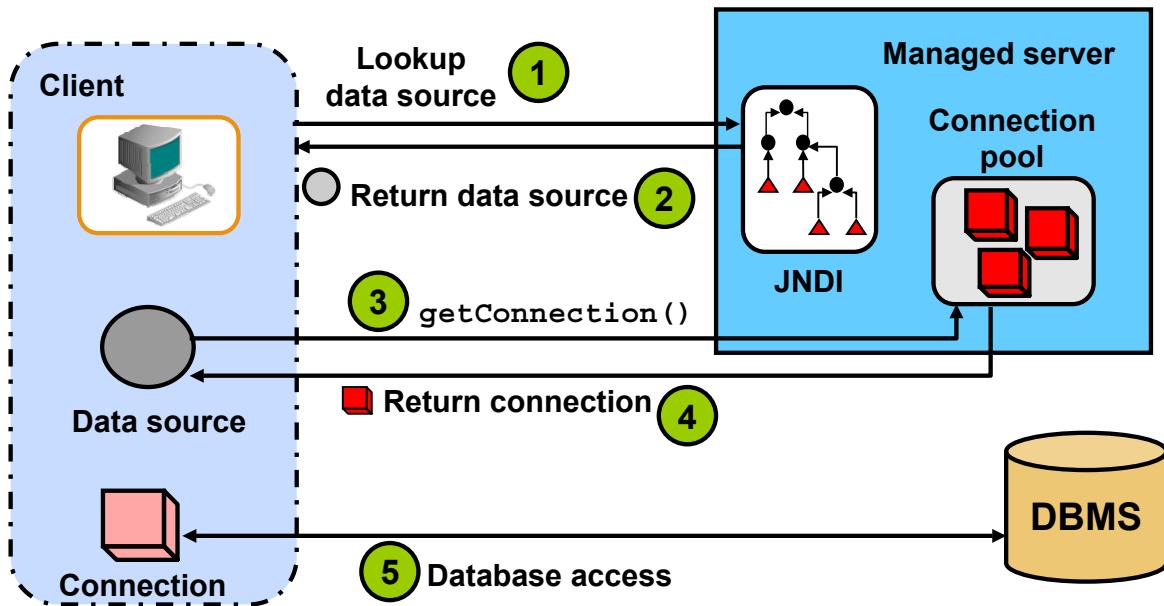
Modular Configuration and Deployment of JDBC Resources

Example of a JDBC configuration:

```
<?xml version="1.0" encoding="UTF-8"?>
<jdbc-data-source xsi:schemaLocation="http://xmlns.oracle.com/weblogic/jdbc-data-
source http://xmlns.oracle.com/weblogic/jdbc-data-source/1.0/jdbc-data-
source.xsd" xmlns="http://xmlns.oracle.com/weblogic/jdbc-data-source"
xmlns:sec="http://xmlns.oracle.com/weblogic/security"
xmlns:wls="http://xmlns.oracle.com/weblogic/security/wls"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <name>MedRecGlobalDataSourceXA</name>
  <jdbc-driver-params>
    <url>jdbc:oracle:thin:@localhost:1521:orcl</url>
    <driver-name>oracle.jdbc.xa.client.OracleXADatasource</driver-name>
    <properties>
      <property>
        <name>user</name>
        <value>medrec</value>
      </property>
    </properties>
    <password->
      <encrypted>{AES}fy0q41+FkMM+ZhcliHQTX21fDGIyK0vdNwHi1B8P528=</password-encrypted>
    </password->
  </jdbc-driver-params>
  <jdbc-connection-pool-params>
    <initial-capacity>5</initial-capacity>
    <max-capacity>10</max-capacity>
    <capacity-increment>1</capacity-increment>
  </jdbc-connection-pool-params>
  <jdbc-data-source-params>
    <jndi-name>jdbc/MedRecGlobalDataSourceXA</jndi-name>
    <global-transactions-protocol>TwoPhaseCommit</global-transactions-protocol>
  </jdbc-data-source-params>
</jdbc-data-source>
```

How Data Source Connection Pools Are Used

A client retrieves a data source through a JNDI lookup and uses it to obtain a database connection.



ORACLE

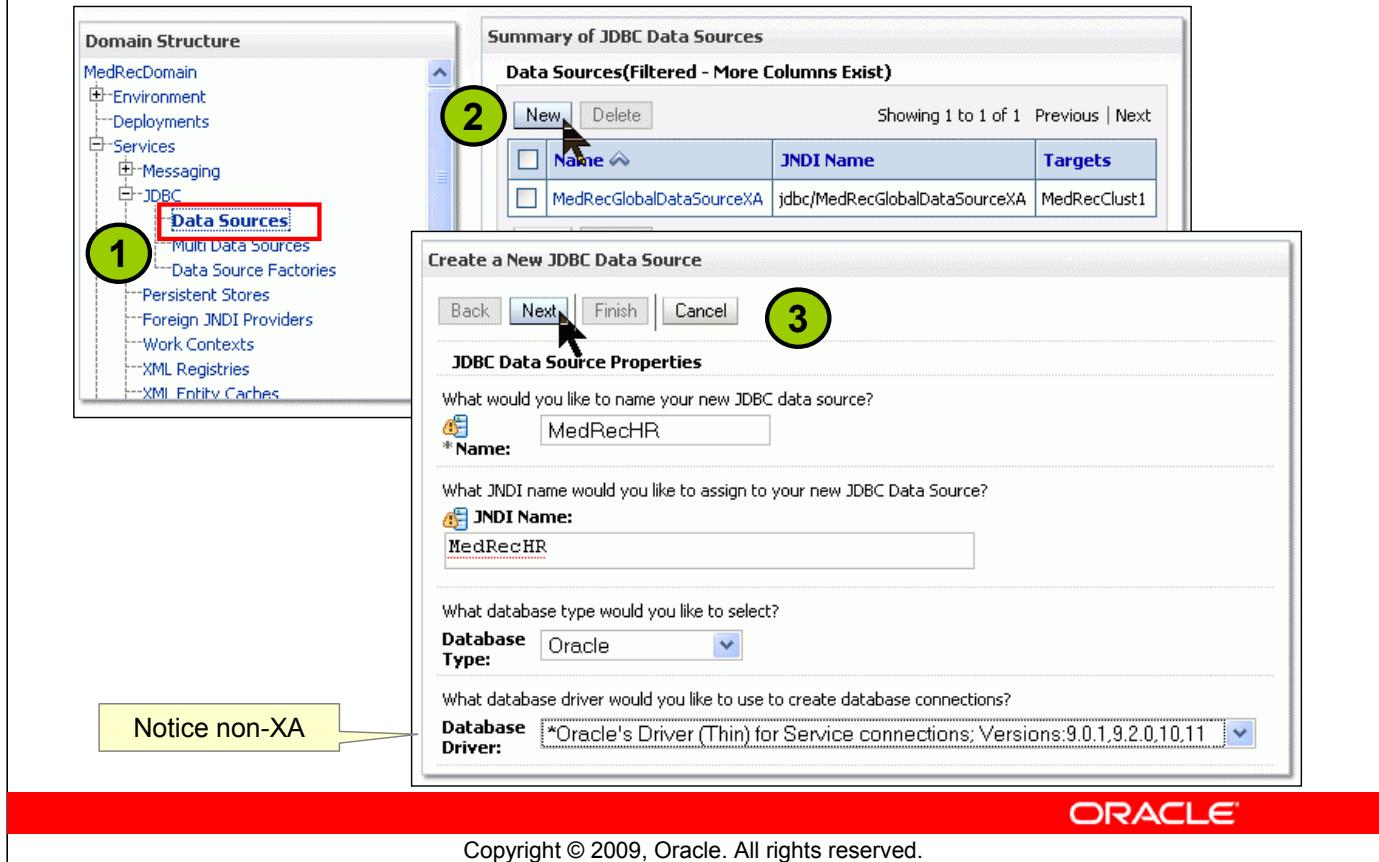
Copyright © 2009, Oracle. All rights reserved.

How Data Source Connection Pools Are Used

This is an example of how a data source is used by the client. The sequence of events is as follows:

- A client retrieves a data source object by performing a lookup in the Oracle WebLogic Server JNDI tree (1 and 2). A data source object contains a reference to the connection pool.
- After a data source object is obtained, the client can obtain a database connection (3 and 4).
- The connection then directly accesses the database (5).

Creating a Data Source Using the Administration Console



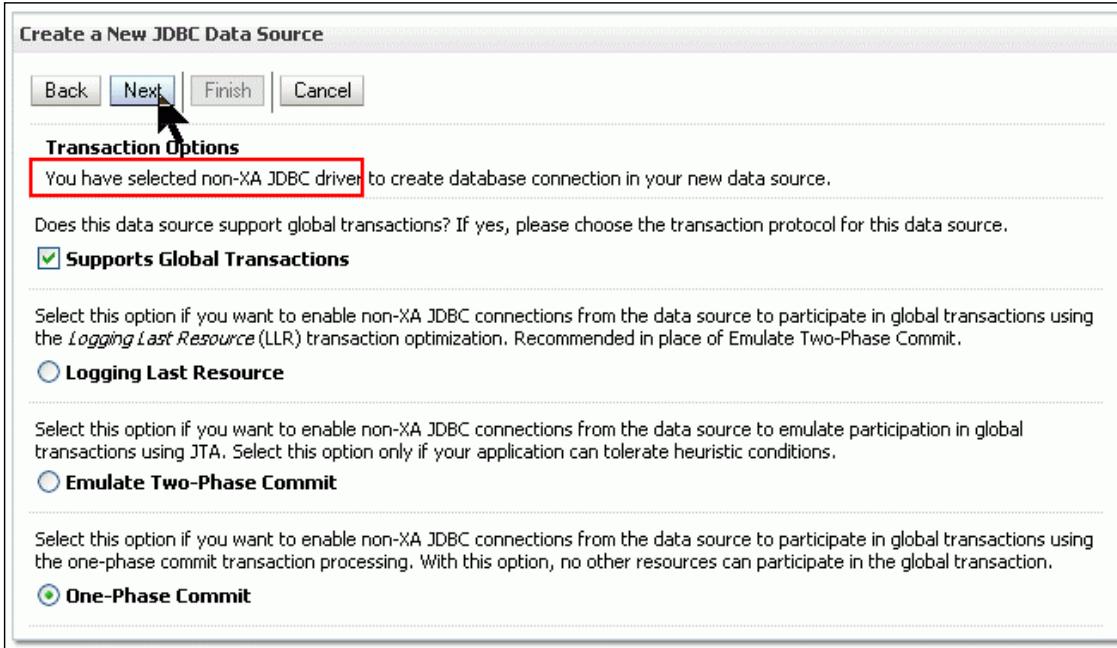
Creating a Data Source Using the Administration Console

You can create data sources via the Administration Console (as shown here) or WLST. Make sure that the JDBC drivers that you want to use to create database connections are installed on all the servers on which you want to configure database connectivity. Some JDBC drivers are installed with Oracle WebLogic Server, including the WebLogic Type 4 JDBC drivers for DB2, Informix, MS SQL Server, Oracle, and Sybase.

1. In the Domain Structure tree, expand Services > JDBC and then select Data Sources.
2. On the Summary of Data Sources page, click New.
3. On the JDBC Data Source Properties page, enter or select the following information and click Next:
 - **Name:** Enter a configuration name for this JDBC data source.
 - **JNDI Name:** Enter the JNDI path to which this JDBC data source will be bound. Applications look up the data source on the JNDI tree by this name when reserving a connection.
 - **Database Type:** Select the database that you want to connect to. If your DBMS is not listed, select Other.
 - **Database Driver:** Select the JDBC driver that you want to use to connect to the database. The list includes common JDBC drivers for the selected DBMS. For example, the non-XA driver was selected, but you could have selected the XA driver. The non-XA will show an extra page for configuration.

Non-XA Configuration

This appears only if a non-XA driver was selected previously.

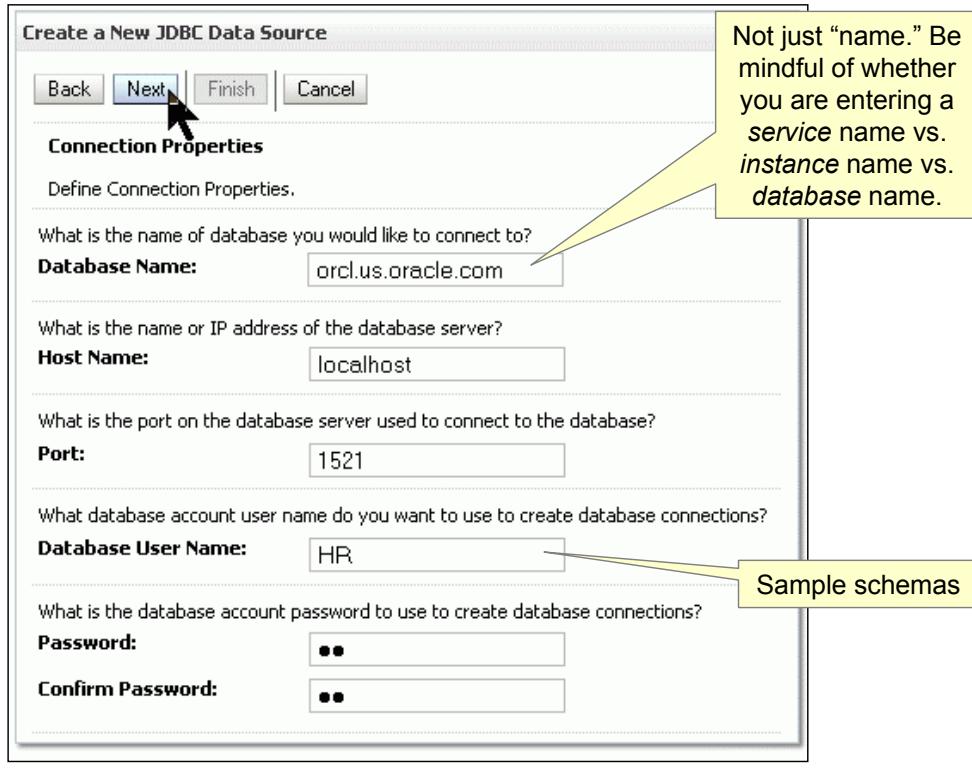


Copyright © 2009, Oracle. All rights reserved.

Non-XA Configuration

If you selected a non-XA JDBC driver, you are presented with two transaction options: Supports Global Transactions and Supports Local Transactions. If you select the non-XA option, WebLogic can use several alternative strategies to emulate XA on your non-XA driver.

Data Source Connection Properties



ORACLE

Copyright © 2009, Oracle. All rights reserved.

Data Source Connection Properties

On the Connection Properties page, enter values for the following properties and click Next:

- **Database Name:** This field is *overloaded*, which means there are multiple kinds of information that could go in this field depending on the context. It is not always the name of the database that you want to connect to. Exact database name requirements vary by JDBC driver and by DBMS. If you used Oracle's Driver for Service Connections, the *service* name would be the full name `orcl.example.com`; if you used Oracle's Driver for Instance Connections, the *instance* name would be just `orcl`. RAC naming is different as well. In any case for Oracle, it is not the *database* name.
- **Host Name:** Enter the DNS name or IP address of the server that hosts the database.
- **Port:** Enter the port on which the database server listens for connections requests. For Oracle databases, you can verify this by entering `lsnrctl status`.
- **Database User Name:** Enter the database user account name that you want to use for each connection in the data source.
- **Password/Confirm Password:** Enter the password for the database user account.

Test Configuration

The screenshot shows two panels of a wizard for creating a new JDBC data source.

Left Panel: Test Configuration

- Test Database Connection:** Test the database availability and connection properties you provided.
- Driver Class Name:** oracle.jdbc.OracleDriver
- URL:** jdbc:oracle:thin:@localhost
- Database User Name:** HR
- Password:** [REDACTED]
- Confirm Password:** [REDACTED]
- Properties:** user=HR
- Test Table Name:** SQL SELECT 1 FROM DUAL

Right Panel: Select Targets

- Servers:** MedRecAdmSvr (unchecked), MedRecSvr3 (checked)
- Clusters:** MedRecClust1
 - All servers in the cluster (radio button)
 - Part of the cluster (radio button)
 - MedRecSvr2 (unchecked)
 - MedRecSvr1 (unchecked)

ORACLE

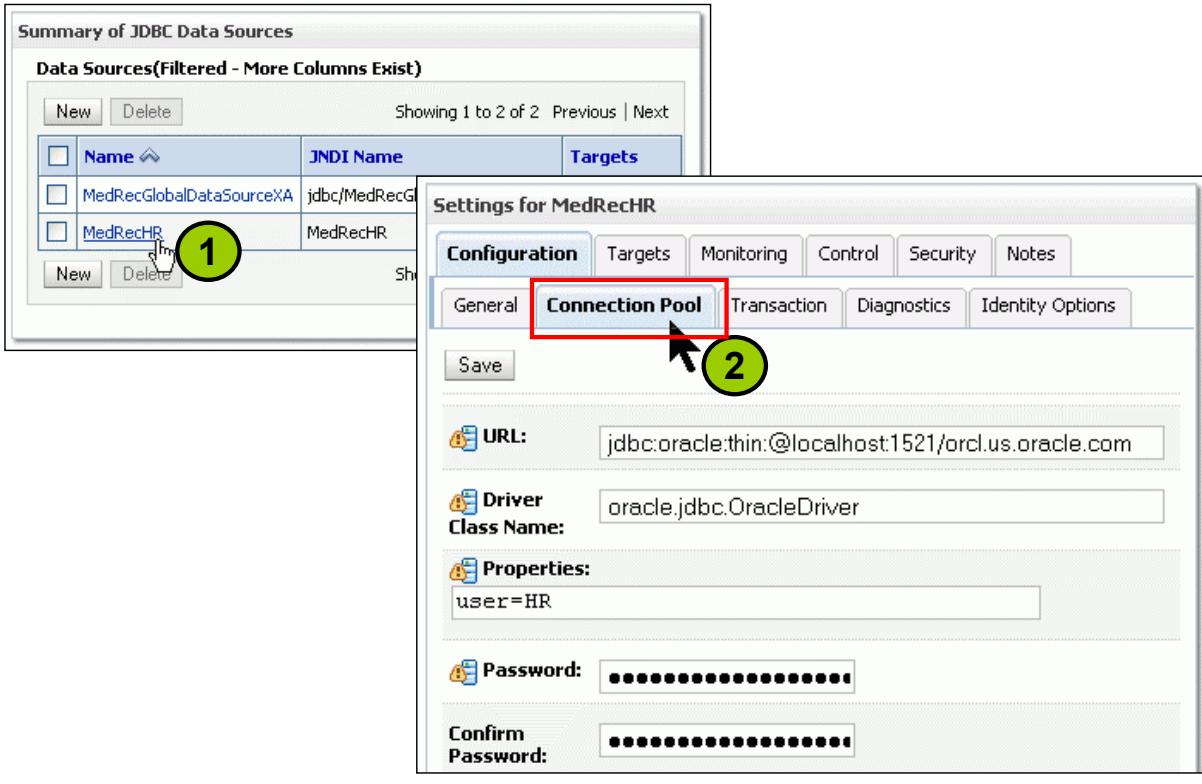
Copyright © 2009, Oracle. All rights reserved.

Test Configuration

On the Test Database Connection page, review the connection parameters and click Test Configuration. WebLogic attempts to create a connection from the Administration Server to the database. Results from the connection test are displayed at the top of the page. If the test is unsuccessful, you should correct any configuration errors and retry the test.

Selecting a target is optional. You can click Finish after testing without assigning a target. The JDBC source will be configured, but not deployed. If you skip selecting the target, there is a chance to deploy the JDBC source later. Select a server target (or not), and then click Finish.

Connection Pool Configuration



ORACLE

Copyright © 2009, Oracle. All rights reserved.

Connection Pool Configuration

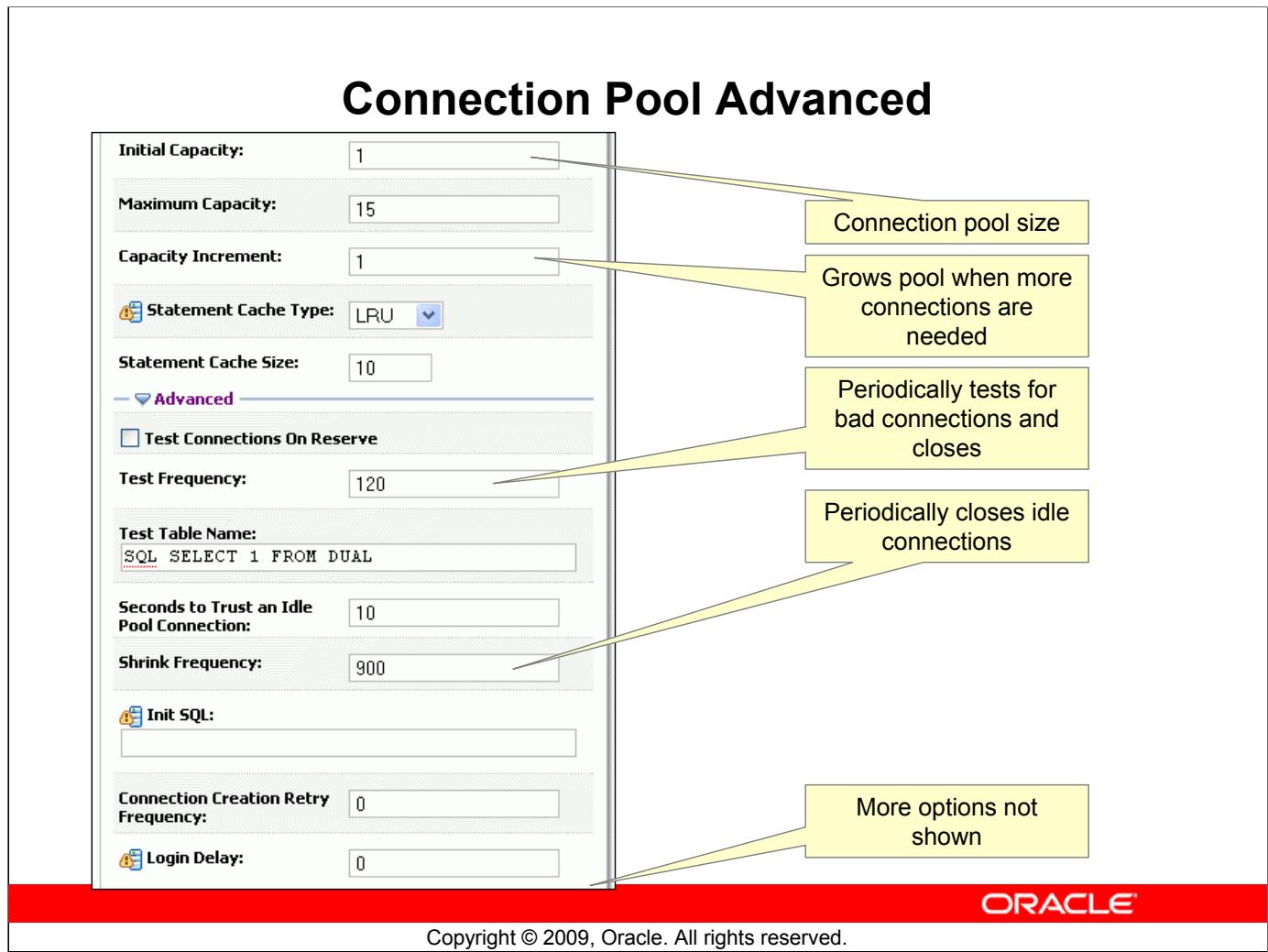
The screenshot in the slide shows how you can modify a connection pool after the data source is created.

Before modifying a connection pool, you should know:

- The JDBC URL of the database
- The connection properties used to authenticate a user or optionally configure the driver
- The maximum number of connections that your application will be allowed by the DBA

After creating your initial data source configuration in the console, you can tune its connection pool settings:

1. In the Domain Structure tree, expand Services > JDBC and then select Data Sources. After selecting your data source, click the Configuration > Connection Pool tab.
 2. Enter values for the available data source attributes.
- Note:** The exclamation mark in a yellow triangle means that changing these values requires restarting some components.



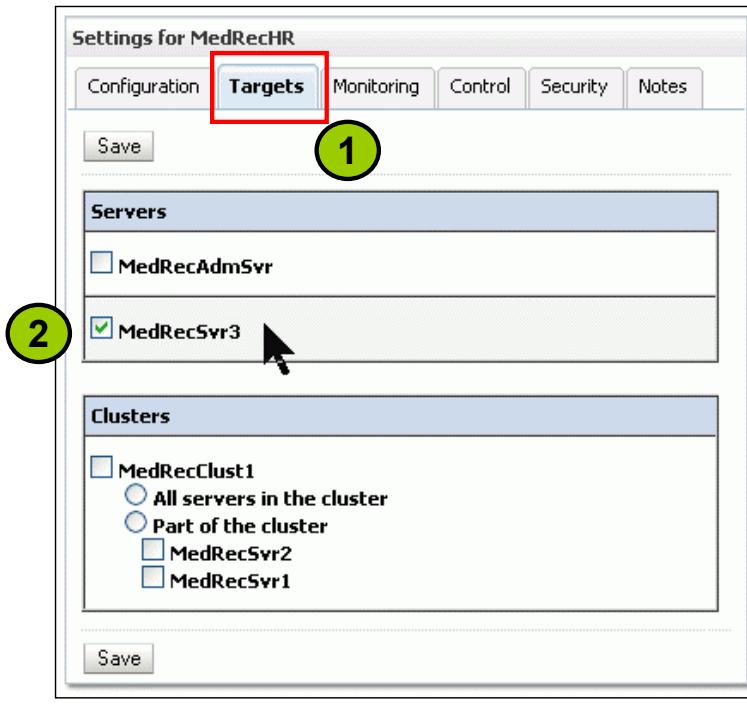
Connection Pool Advanced

Some of the key options are found under the Advanced section, including:

- **Initial Capacity:** This is the number of physical connections to create when deploying the connection pool. This is also the minimum number of physical connections that the connection pool will keep available.
- **Maximum Capacity:** This is the maximum number of physical connections that this connection pool can contain. For optimal performance, set the value of Initial Capacity equal to the value for Maximum Capacity, although that disables the dynamic resizing.
- **Capacity Increment:** When there are no more available physical connections to satisfy connection requests, Oracle WebLogic Server creates this number of additional physical connections and adds them to the connection pool up to the maximum capacity.
- **Test Frequency:** This is the number of seconds between when Oracle WebLogic Server tests unused connections. This requires that you specify a Test Table Name. DUAL is included in all Oracle database installations for such a purpose as this. Connections that fail the test are closed and reopened to reestablish a valid physical connection. If the test fails again, the connection is closed.

Targeting a Data Source

Deploy data sources to one or more servers in your domain.



ORACLE

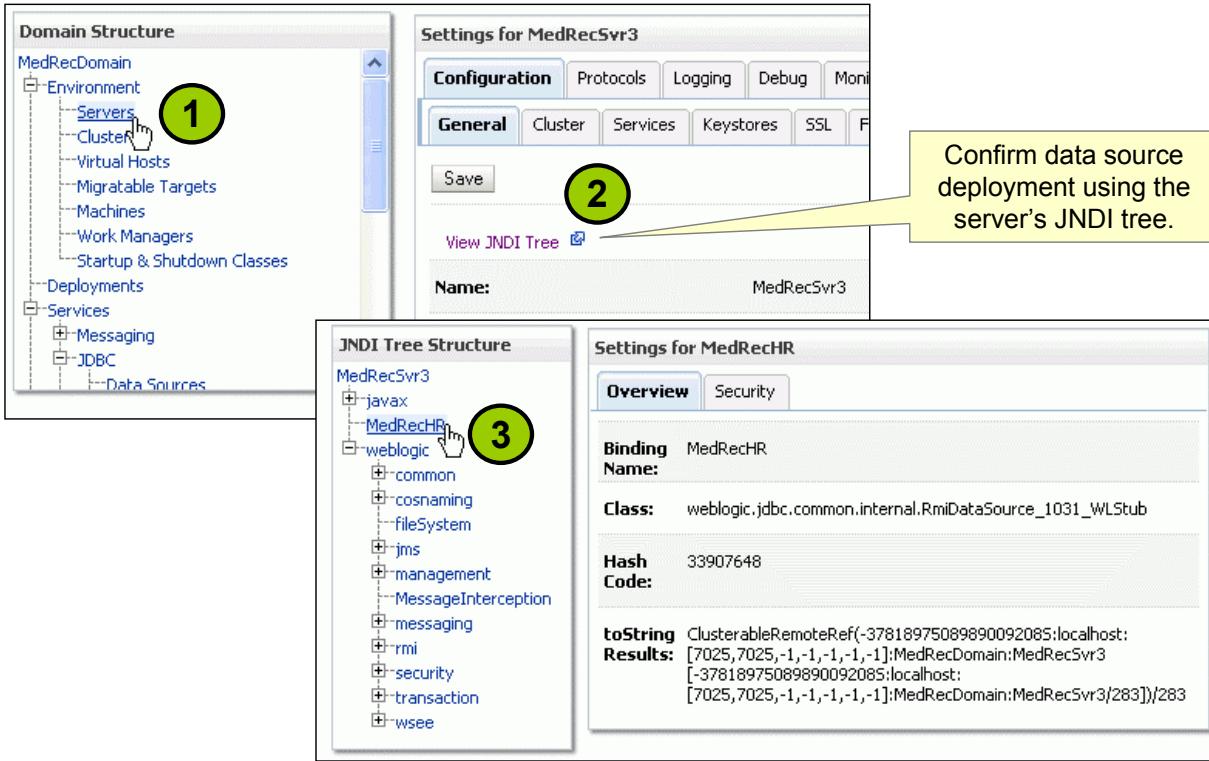
Copyright © 2009, Oracle. All rights reserved.

Targeting a Data Source

This is the second opportunity to deploy a JDBC to a target. Any previous targets are prechecked when this page is displayed. When you target a JDBC data source, a new instance of the data source is created on the target. When you select a server as a target, an instance of the data source is created on the server. When you select a cluster as a target, an instance of the data source is created on all member servers in the cluster.

1. Navigate to the data source that you want to modify and click the Targets tab.
2. Select each server or cluster on which you want to deploy the data source and click Save.

Viewing the Server JNDI Tree via the Administration Console



ORACLE

Copyright © 2009, Oracle. All rights reserved.

Viewing the Server JNDI Tree via the Administration Console

The screenshot in the slide shows viewing the Java Naming and Directory Interface (JNDI) tree. If the data source is deployed successfully, a new entry should be added to the local JNDI tree of the target servers. The name of the entry should match the JNDI name that is used to configure the data source. If you use a fully qualified JNDI name containing path separators (for example, “hr.datasource.HRDataSource” instead of just MedRecHR), the entry will not be found at the root of the tree. Instead, directories are created to match the fully qualified name, if they do not already exist, with plus and minus icons to expand and collapse them.

1. In the left pane, expand Environment > Servers. Then select a specific server.
2. On the default Configuration > General tab of the server, click View JNDI Tree. The JNDI tree is displayed in a new browser window or browser tab.
3. Use the left panel to navigate the directories of the JNDI tree.

Note: When you create contexts and bind objects programmatically, the subcontext will not be auto-created (therefore, subcontexts must be programmatically created before objects are placed into them); but when a JNDI entry is configured using the Administration Console as shown here, then the subcontext will be automatically created for you.

Listing the JNDI Contents via WLST

- WLST provides a command-line utility for viewing the JNDI bindings.
- `jndi()` changes to the JNDI tree and `ls()` lists the bindings.

```
wls:/offline> connect("weblogic","welcome1","t3://localhost:7020")
wls:/base_domain/serverConfig> jndi()
wls:/base_domain/jndi> cd('AdminServer')

wls:/base_domain/jndi/AdminServer> ls()
dr-- ejb
dr-- javax
dr-- weblogic
-r-- cgDataSource
-r-- cgDataSource-nonXA
-r-- mejbmejb_jarMejb_EO
-r-- samplesDataSource
                                         weblogic.rmi.cluster.ClusterableRemoteObject
                                         weblogic.rmi.cluster.ClusterableRemoteObject
                                         weblogic.rmi.cluster.ClusterableRemoteObject
                                         weblogic.rmi.cluster.ClusterableRemoteObject
```

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Listing the JNDI Contents via WLST

`jndi()` navigates to the JNDI tree for the server to which the Oracle WebLogic Scripting Tool (WLST) is currently connected. This read-only tree holds all the elements that are currently bound in JNDI.

In the event of an error, the command returns a `WLSTException`.

The following example navigates from the run-time MBean hierarchy to the domain JNDI tree in an administration server instance.

```
wls:/myserver/runtime> jndi()
Location changed to jndi tree. This is a read-only tree with No
root. For more help, use help('jndi')
wls:/myserver/jndi> ls()
dr-- ejb
dr-- javax
dr-- jms
dr-- weblogic
```

Demonstration

- Configure data sources for Oracle Database.
- Go to OTN > Tutorials > Fusion Middleware > Oracle WebLogic Server 10.3 > Deploy J2EE Applications > [Configure Data Sources.](#)

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Demonstration

See the demonstration at the following URL:

http://www.oracle.com/technology/obe/fusion_middleware/wls103/appdeploy/configure/datasource/Conf_DS_WLS.htm

JDBC URLs

Database locations are specified using a JDBC Uniform Resource Locator (URL).

- Example 1:
 - This URL specifies that the `oracle:thin` subprotocol should be used to connect to an Oracle Database:

```
jdbc:oracle:thin:@dbhost:1521:SALESINFO
```

- Example 2:
 - This URL can be used to access a PointBase database:

```
jdbc:pointbase:server://dbhost:9092/HRDATABASE
```

Copyright © 2009, Oracle. All rights reserved.

JDBC URLs

If you use a JDBC driver developed by a third party, the documentation tells you what subprotocol to use—that is, what to put after “`jdbc :`” in the JDBC URL. The syntax for a JDBC URL is `jdbc : subprotocol : subname`.

- `subprotocol` identifies the database connectivity mechanism.
- `subname` identifies the data source. The subname can vary depending on the subprotocol.

The contents and syntax of `subname` depends on `subprotocol`. `subname` can also specify a network address for the database—for example, `subname` can be specified using “`//hostname:port/dbname`.”

For Example 1

- `dbhost` : The host name or IP address
- `1521` : The default listener port
- `SALESINFO` : The system identifier (SID), the name of the database

For Example 2

- `subprotocol` is `pointbase:server`.
- `subname` is a location of the PointBase database named `HRDATABASE`.

Connection Properties

- Are key/value pairs
- Are used to configure JDBC connections
- Are passed to the driver during connection setup



Copyright © 2009, Oracle. All rights reserved.

Connection Properties

Connection properties are a set of key/value pairs that are passed to the driver when database connections are created. Connection properties are specific to the driver. For a complete list, see your driver documentation.

Specifying Connection Properties

A partial list of connection properties for the supplied drivers:

| Driver | Some Connection Properties |
|-----------|--|
| Oracle | User, Password, ServerName, ServiceName, PortNumber |
| Sybase | User, Password, ServerName, DatabaseName, PortNumber |
| MSSQL | User, Password, ServerName, DatabaseName, PortNumber |
| Informix | User, Password, ServerName, DatabaseName, PortNumber |
| PointBase | cache.size, crypto.communication, database.home, database.pagesize |

Copyright © 2009, Oracle. All rights reserved.

Specifying Connection Properties

PointBase connection properties can be set in the `pointbase.ini` file. You can select the `pointbase.ini` parameters to configure the database properties. By configuring the database properties, you can increase the performance of your system. However, PointBase should not be used in a production environment, so performance for that DBMS is less critical.

Road Map

- Overview of JDBC
- Data sources
- Monitoring and testing data sources
 - Monitoring
 - Testing
 - Suspend/resume



ORACLE®

Copyright © 2009, Oracle. All rights reserved.

Monitoring and Testing a Data Source

The screenshot shows two panels of the Oracle WebLogic Administration Console. The left panel displays a 'Messages' section with a green checkmark indicating a successful test of the MedRecHR data source on server MedRecSrv3. Below this is the 'Settings for MedRecHR' page with tabs for Configuration, Targets, Monitoring, Control, Security, and Notes. The Monitoring tab is highlighted with a red box. Under the Monitoring tab, the 'Testing' sub-tab is selected and highlighted with a red box. A button labeled 'Test Data Source' is visible. The right panel shows the 'Deployed Instances of this Data Source' table. The table has columns: Server, State, Connections Total Count, Current Capacity, Waiting For Connection High Count, Highest Num Available, and Active Connections Average Count. One instance is listed: MedRecSrv3, Running, 1, 1, 0, 1, 0. A callout box points to the 'Monitoring' tab with the text 'Data source retested successfully.' Another callout box points to the 'Statistics' tab with the text 'Monitor data source statistics.' The Oracle logo is at the bottom right.

| Server | State | Connections Total Count | Current Capacity | Waiting For Connection High Count | Highest Num Available | Active Connections Average Count |
|------------|---------|-------------------------|------------------|-----------------------------------|-----------------------|----------------------------------|
| MedRecSrv3 | Running | 1 | 1 | 0 | 1 | 0 |

Monitoring and Testing a Data Source

After you create a JDBC data source and target it to one or more servers, you can monitor it in the Administration Console. Locate and select your new data source and click the Monitoring > Statistics tab. Statistics are displayed for each deployed instance of the data source. Optionally, click “Customize this table” to change the columns displayed in the Statistics table. For example, some of the available columns (not displayed by default) include:

- **Active Connections Current Count:** The number of connections currently in use by applications
- **Active Connections Average Count:** The average number of active connections from the time that the data source was deployed
- **Connections Total Count:** The cumulative total number of database connections created in this data source from the time that the data source was deployed
- **Current Capacity:** The current count of JDBC connections in the connection pool in the data source
- **Highest Num Available:** The highest number of database connections that were available at any time in this instance of the data source from the time that the data source was deployed
- **Waiting for Connection High Count:** The highest number of application requests concurrently waiting for a connection from this instance of the data source

Connection Pool Life Cycle

For this data source...

...on a given server...

...take this action.

Settings for testSample

Control

Use this page to manually control each instance of this JDBC data source.

Customize this table

Deployed Instances of this Data Source

| | Server Name | State | Status of Last Action |
|-------------------------------------|-------------|---------|-----------------------|
| <input checked="" type="checkbox"/> | MedRecSvr1 | Running | None |

Suspend Resume Shutdown Start

Suspend Force Suspend

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Connection Pool Life Cycle

By default, a connection pool is automatically started when it is deployed. You can manually stop and restart the connection pool. This might be necessary if you change the username/password or some other characteristic of the connection. If you wanted to gracefully shut down an application, you might start by shutting down the connection pool.

Quiz

Which of the following is NOT an available configuration attribute for a JDBC data source?

1. Host name
2. Queue size
3. Test frequency
4. Initial capacity
5. Capacity increment



Copyright © 2009, Oracle. All rights reserved.

Answer: 2

All these are valid settings for a data source except queue size. Data sources use connection pooling, but not a queue.

Quiz

Which are the two levels of data sources available in Oracle WebLogic Server?

1. Connection
2. Web
3. Application
4. Process
5. System



Copyright © 2009, Oracle. All rights reserved.

Answers: 3, 5

Remember that system data sources are scoped to the domain, whereas application data sources are deployed as part of an application.

Quiz

Client applications look up data sources from the local server's
_____ tree:

1. Application
2. Web
3. LDAP directory
4. JNDI
5. System



Copyright © 2009, Oracle. All rights reserved.

Answer: 4

Summary

In this lesson, you should have learned how to:

- Define JDBC high-level architecture
- Configure Oracle WebLogic Server–provided JDBC driver types
- Create data source definitions
- Create connection pool definitions
- Manage JDBC resources using the Administration Console



Copyright © 2009, Oracle. All rights reserved.

Practice 13 Overview: Configuring JDBC Data Sources

This practice covers the following topics:

- Creating JDBC modules (via GUI and WLST)
- Deploying JDBC modules
- Testing JDBC modules



Copyright © 2009, Oracle. All rights reserved.

Practice 13 Overview: Configuring JDBC Data Sources

See Appendix A for the complete steps to do the practice.

Setting Up Java Message Service (JMS) Resources

14

ORACLE®

Copyright © 2009, Oracle. All rights reserved.

Objectives

After completing this lesson, you should be able to do the following:

- Describe Java Message Service
- Describe how Oracle WebLogic Server JMS is implemented
- Configure JMS server
- Configure connection factories
- Configure queues and topics
- Configure persistent messages
- Deploy an application that uses JMS
- Monitor JMS resources and messages



Copyright © 2009, Oracle. All rights reserved.

Objectives

Scenario

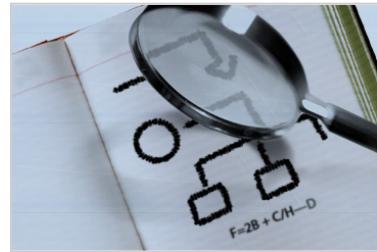
Consider an online order entry application that integrates with a shipping application. In this case, you may not want the online customer to keep waiting for the shipping application to finalize the shipping process.

Generally, in such cases, the following steps are performed:

1. The customer places an order using the order entry application.
2. When the order is completed and confirmed (may involve a credit check and so on), the order details are placed in a message queue.
3. The shipping application regularly checks the order message queue, picks up the orders from the message queue, assigns the appropriate shipping agency (for example, UPS, FedEx, or USPS), and appropriately generates shipping labels.
4. In addition, the shipping may append the shipping details to the order message.

Road Map

- Oracle WebLogic Server JMS administration
 - JMS overview
 - JMS server and modules
 - Types of JMS destinations
- Configuring JMS objects
- Durable subscribers and persistent messaging
- Monitoring JMS

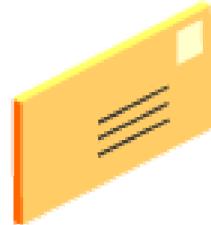


ORACLE®

Copyright © 2009, Oracle. All rights reserved.

Message-Oriented Middleware

- The message-oriented architecture enables asynchronous and cross-platform integration of applications.
- Message-oriented middleware refers to an infrastructure that supports messaging.
- Typical message-oriented middleware architectures define the following elements:
 - Message structure
 - The way to send and receive messages
 - Scaling guidelines



ORACLE

Copyright © 2009, Oracle. All rights reserved.

Message-Oriented Middleware

The message-oriented middleware became widely used when providers created architectures that could operate in a standard way on a variety of platforms and enabled asynchronous communication between applications. These providers gained popularity in enabling integration of mainframes and personal computers.

Even though there is much competition and variety in message-oriented middleware products, they tend to fall into one of the following categories:

- Point-to-point
- Publish/Subscribe
- Request-reply

JMS Messaging Models

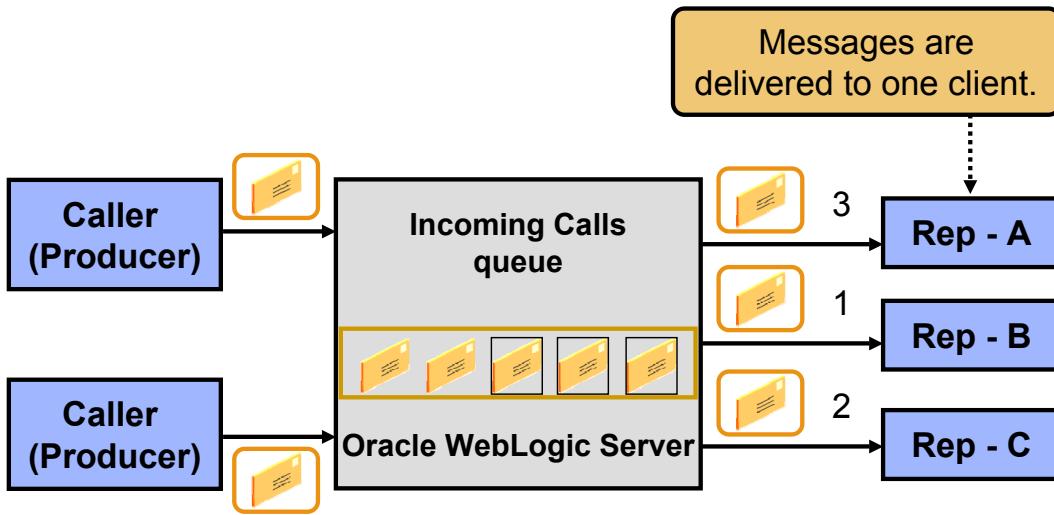
JMS supports the point-to-point (PTP) and Publish/Subscribe messaging models. The models are very similar, except the following:

- The PTP messaging model enables delivery of a message to exactly one recipient.
- The Publish/Subscribe messaging model enables delivery of a message to multiple recipients.

Request-reply messaging model is more suited in a synchronous messaging environment where the requester and replier are in conversational mode—the requester waits for a response from the replier before continuing work. It is not explicitly supported in JMS.

Point-To-Point Queue

Many message producers can serialize messages to multiple receivers in a queue.



ORACLE

Copyright © 2009, Oracle. All rights reserved.

Point-To-Point Queue

When using a PTP queue, multiple message producers can put messages onto a single queue. The queue serializes the messages in a linear order. Multiple receivers can take messages off the queue; the messages typically come off in a first-in, first-out (FIFO) order; the oldest message on the queue is the first one to be taken off.

A message can be delivered only to one receiver. Receivers are also referred to as consumers.

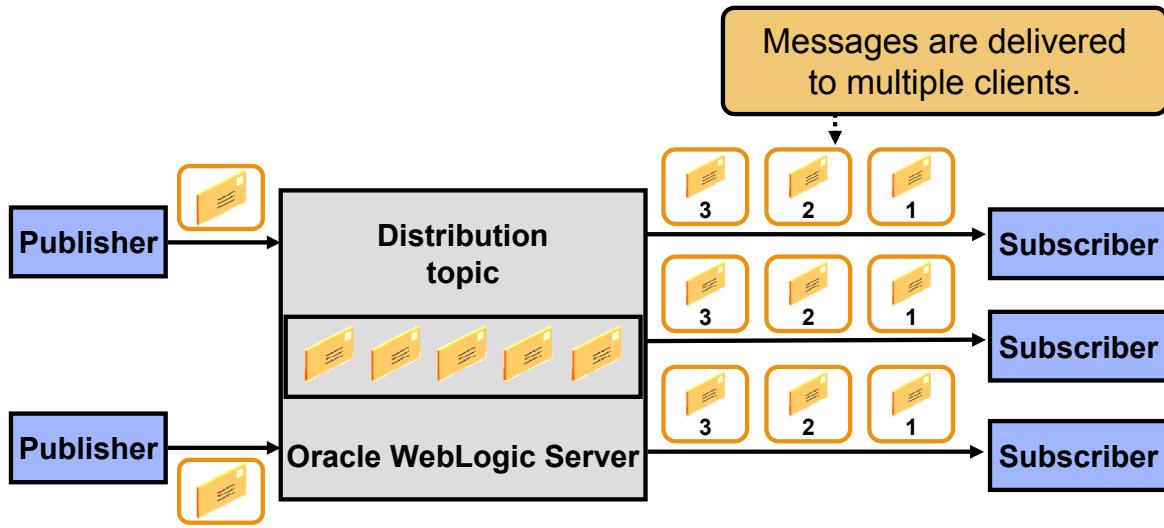
An example of when to use a PTP queue would be at a call center.

- Calls are routed into the network through a PBX. The PBX system places incoming calls onto an Incoming Calls queue. When a service representative is available, the representative requests for the next caller in the system.
- The system pulls off the queue the caller who has been waiting the longest (FIFO method) and routes the caller to the service representative.
- After the conversation is established between an in-queue customer and a representative, it becomes a synchronous communication. (This is similar to request-reply mode).

This is only an example and, in many cases, the responses are not just pure FIFO but weightings assigned by the organizations.

Publish/Subscribe Topics

Publishing and subscribing to a topic decouples producers from consumers.



Copyright © 2009, Oracle. All rights reserved.

Publish/Subscribe Topics

Having the publishers publish to a topic rather than directly to a list of subscribers decouples the publishers and subscribers.

By doing this, a publisher is not required to manage the number of subscribers (if any) that must receive the message. By delegating the message delivery work to the message-oriented middleware server (which manages the topic), the publisher does not have to manage the delivery of guaranteed messages, fault tolerance of its production, load balancing, or other issues. By decoupling a subscriber from the publisher, the subscriber does not have to determine whether its publisher is active. If the message-oriented middleware server is executing, the needs of both the publishers and the subscribers are met.

An example of using a Publish/Subscribe topic is a stock ticker application.

- A typical system would set up a topic for each stock that is traded on the exchanges.
- When a trade is made on a stock, the respective exchange publishes a message to the topic that is associated with the stock traded.
- Clients who are interested in receiving updates about the status of their stocks use a program to subscribe to the topics of each stock they are interested in.
- When the topic update is recognized, the message server broadcasts the message to all the interested (clients) stock ticker programs.

Oracle WebLogic Server JMS Features

Oracle WebLogic Server JMS supports:

- Both the point-to-point and Publish/Subscribe JMS models
- Acknowledgement-based guaranteed delivery
- Transactional message delivery
- Durable subscribers
- Distributed destinations
- Recovery from failed servers



ORACLE

Copyright © 2009, Oracle. All rights reserved.

Oracle WebLogic Server JMS Features

An enterprise messaging system enables applications to asynchronously communicate with one another through the exchange of messages. A message is a request, report, and/or event that contains information needed to coordinate communication between different applications. A message provides a level of abstraction, which allows you to separate the details about the destination system from the application code.

The Oracle WebLogic Server implementation of JMS fully supports the point-to-point and Publish/Subscribe models of the messaging middleware.

Oracle WebLogic Server also provides acknowledgement-based (ACK) guaranteed message delivery (GMD) by enabling persistent storage of messages until the receiver of the message issues an acknowledgement of receipt.

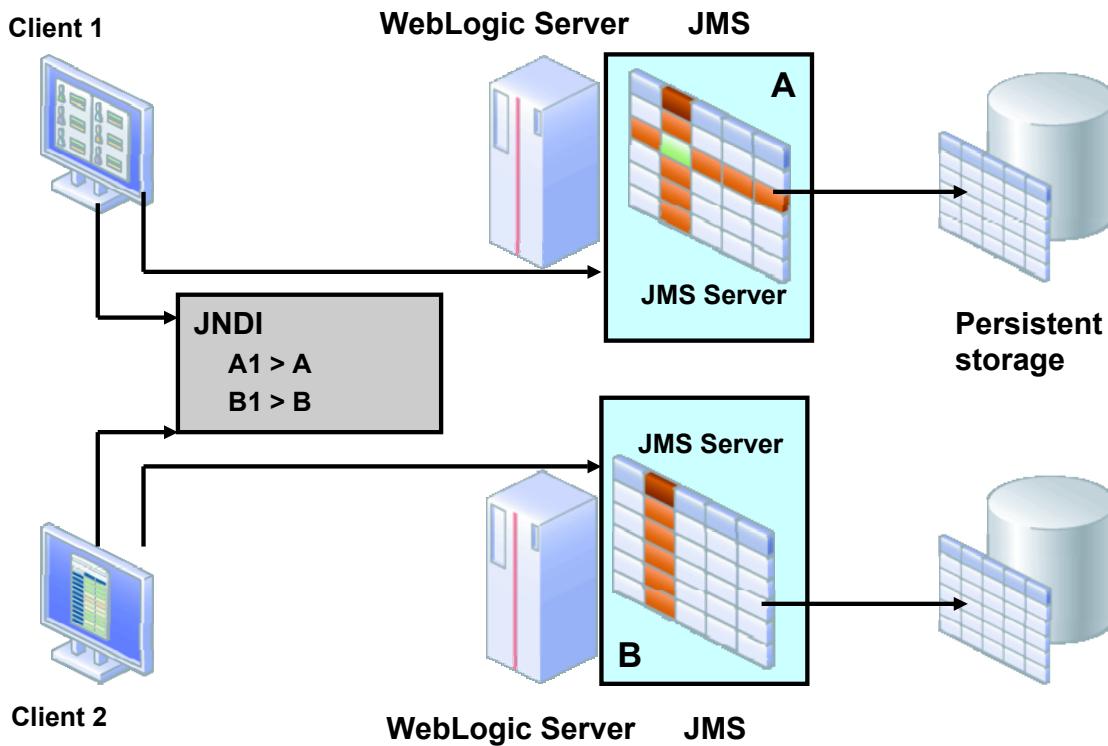
Oracle WebLogic Server JMS uses its built-in support for JDBC and JDBC connection pools to persist JMS messages in a database.

Oracle WebLogic Server supports transactional message delivery. Transactional message delivery gives the developer the ability to put a JMS session into a transaction context. In Oracle WebLogic Server JMS, the message is not visible or available for consumption until the transaction is committed. A session can optionally roll back the transaction, which has the transaction “drop” the messages it had previously buffered.

Oracle WebLogic Server JMS Features (continued)

Oracle WebLogic Server allows clients to register themselves as durable subscribers. A durable subscriber is a client that expects to receive all persistent messages that are sent to a particular destination, whether the client is currently executing or not. If the durable subscriber is not currently executing, Oracle WebLogic Server stores the messages in a persistent store until the durable subscriber reactivates and retrieves the stored messages.

Oracle WLS JMS Architecture



ORACLE

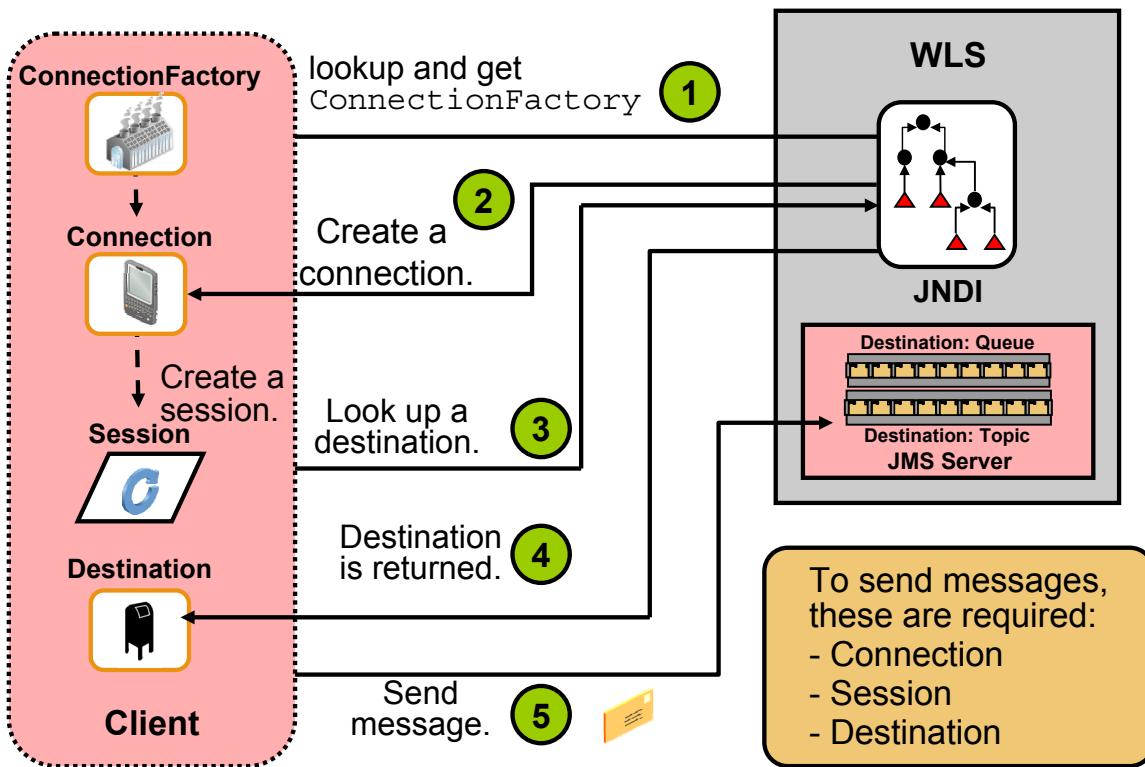
Copyright © 2009, Oracle. All rights reserved.

Oracle WLS JMS Architecture

The major components of the WebLogic JMS Server architecture include:

- JMS servers that can host a defined set of modules and any associated persistent storage that resides on a WebLogic Server instance. JMS server configuration is stored in the domain config.xml file.
- JMS modules that contain configuration resources (such as queues, topics, and connections factories) and are defined by XML documents that conform to the weblogic-jms.xsd schema
- Client JMS applications that either produce messages to destinations or consume messages from destinations
- Java Naming and Directory Interface (JNDI), which provides a resource lookup facility. JMS resources such as connection factories and destinations are configured with a JNDI name. The run-time implementations of these resources are then bound to JNDI using the given names.
- WebLogic persistent storage (file store or JDBC-accessible) for storing persistent message data

Typical JMS Messaging Process



ORACLE

Copyright © 2009, Oracle. All rights reserved.

Typical JMS Messaging Process

In JMS implementations, developers use a connection factory to enable their applications to connect to a queue or topic.

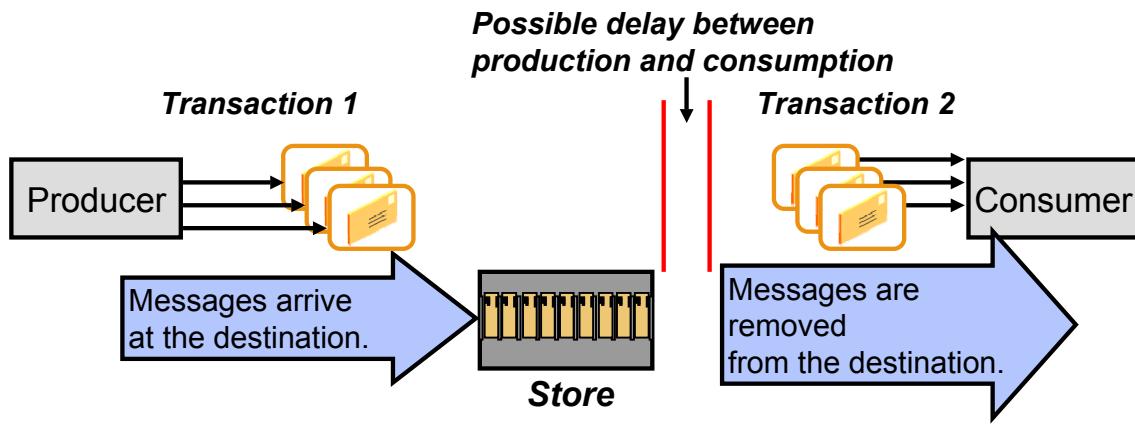
A connection factory is a lightweight object stored on a Java Naming and Directory Interface (JNDI) tree that is used to create connections to destinations. A connection is a communication link to the JMS server that is used to create sessions.

A session is used to create senders, receivers, and empty messages. A session is also used to demarcate transactions.

Destination, a lightweight object stored on JNDI, is the target for the messages.

Transacted Messaging

- A JMS client can use Java Transaction API (JTA) to participate in a distributed transaction.
- Alternatively, a JMS client can demarcate transactions that are local to the JMS session through a transacted session.
- Participation in a transaction is optional.



ORACLE

Copyright © 2009, Oracle. All rights reserved.

Transacted Messaging

JMS clients can participate in a distributed or local transaction. There are two scenarios:

- On the Producer side, a transaction begins and some operations, such as sending messages, are performed. If the transaction commits, all the messages are sent to the destination. If the transaction rolls back, none of the messages arrive at the destination.
- On the Consumer side, a transaction begins and some operations, such as processing messages, are performed. If the transaction commits, the processed messages are removed from the destination. If the transaction rolls back, the messages stay in the destination.

JMS Administrative Tasks

- Creating and monitoring JMS servers
- Creating connection factories
- Creating and monitoring destinations
- Creating JMS stores
- Configuring paging thresholds and quotas
- Configuring durable subscriptions
- Managing JMS service failover



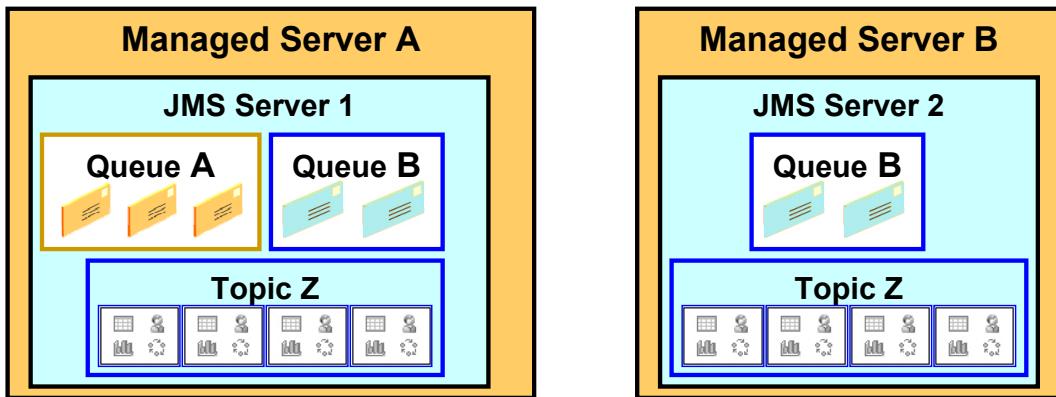
ORACLE

Copyright © 2009, Oracle. All rights reserved.

JMS Administrative Tasks

As an administrator, you are responsible for configuring and monitoring most aspects of JMS. The architecture of your system determines the type of JMS destinations to configure. It is your responsibility to monitor the Oracle WebLogic Server JMS and gather statistics. All these administrative tasks are discussed throughout this lesson.

Oracle WLS JMS Implementation



Resource definitions: In JMS modules

JMS Module A

SubDeployment 1:

Queue A: Target (JMS Server1)

SubDeployment 2:

Queue B: Target (JMS Server 1 and JMS Server 2)

SubDeployment 3:

Topic Z: Target (JMS Server 1 and JMS Server 2)

ORACLE

Copyright © 2009, Oracle. All rights reserved.

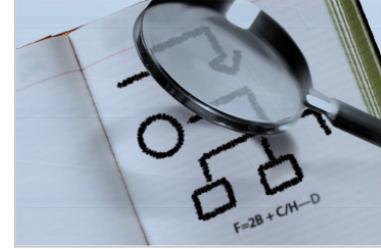
Oracle WLS JMS Implementation

When you implement JMS in WLS, you configure the following JMS resources:

- Configure the necessary JMS servers and target them to the appropriate managed servers.
- Configure JMS modules.
- Within the JMS modules, you define the queue or topic resources.
- Then using the subdeployment definitions, target the queues to the appropriate JMS servers.

Road Map

- Oracle WebLogic Server JMS administration
- Configuring JMS objects
 - Configuring JMS servers
 - Configuring JMS modules and subdeployments
 - Configuring connection factories
 - Configuring destinations
- Durable subscribers and persistent messaging
- Monitoring JMS

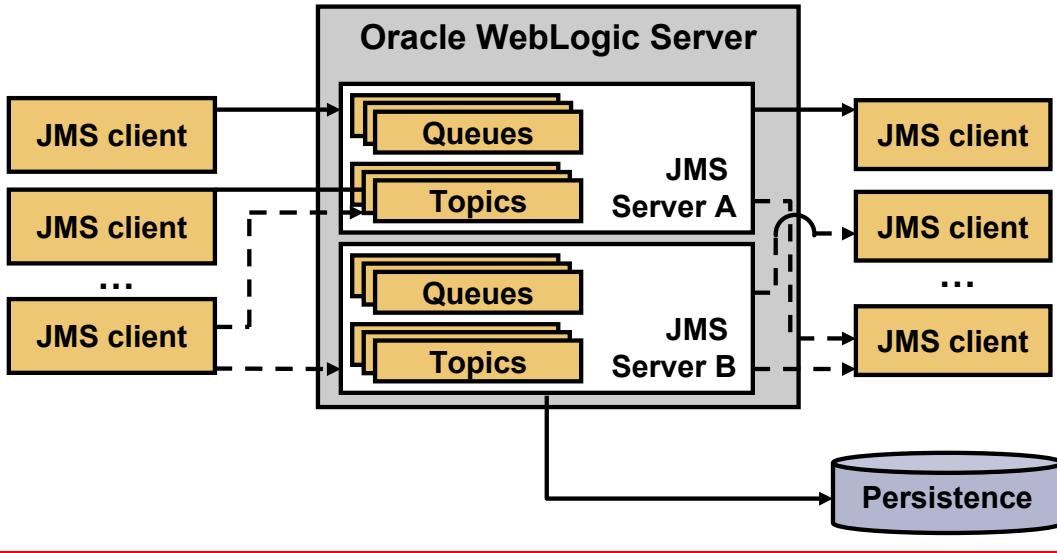


ORACLE®

Copyright © 2009, Oracle. All rights reserved.

Oracle WLS JMS Server

- In Oracle WLS, the messaging service is implemented through a JMS server.
- A JMS server receives and distributes messages.



Copyright © 2009, Oracle. All rights reserved.

ORACLE

Oracle WLS JMS Server

JMS servers are configuration entities that act as management containers for the JMS queues and topics in JMS modules that are targeted to specific JMS servers. A JMS server is configured and targeted to a server. There can be multiple JMS servers targeted to the same Oracle WebLogic Server instance.

The configuration of JMS servers is in the domain's `config.xml` file. Multiple JMS servers can be configured as long as they are uniquely named. Each JMS server handles requests for all targeted modules' destinations. Requests for destinations that are not handled by a JMS server are forwarded to the appropriate server instance.

A JMS server's primary responsibility for its destinations is to maintain information about the persistent store that is used for any persistent messages that arrive on the destinations and to maintain the states of the durable subscribers created on the destinations. JMS servers also manage message paging on destinations. Optionally, they can manage message and/or byte thresholds, as well as server-level quota for their targeted destinations. Because it is a container for targeted destinations, any configuration or run-time changes to a JMS server can affect all the destinations that it hosts.

Creating a JMS Server

The screenshot shows the Oracle WebLogic Administration Console interface. On the left, the 'Domain Structure' tree view is expanded to show 'MedRecDomain' with 'Services' selected. Under 'Services', 'Messaging' is expanded, and 'JMS Servers' is highlighted with a green circle labeled '1'. In the center-right pane, a table titled 'JMS Servers(Find)' shows several existing servers like 'HRJMSServer' and 'MedRecJM'. A green circle labeled '2' is over the 'New' button in the table header. On the right, a modal dialog box titled 'Create a New JMS Server' is open. It has buttons for 'Back', 'Next', 'Finish', and 'Cancel'. The 'JMS Server Properties' section contains a note about identifying the new server and a required field indicator (*). Below it, a 'Name:' field is highlighted with a green circle labeled '3', containing the value 'HR-JMS-Server'. Other fields include 'Persistent Store:' with '(none)' selected and a 'Create a New Store' button.

Creating a JMS Server

You can create and configure a JMS server by using the Administration Console. To create a JMS server, perform the following steps:

1. Expand the Services node in Domain Structure in the left panel, and then expand the Messaging node. Click JMS Servers. The summary of JMS servers appears in the right pane.
2. Click Lock & Edit to enable editing configuration. Then click New at the JMS Servers table. The “Create a New JMS Server” dialog box appears.
3. Enter values for the following configuration parameters:
 - Name: The name of the JMS server
 - Persistent Store: The backing store used by destinations. A value of none means that the JMS server will use the default persistent store that is configured on each targeted WLS instance.
4. Click Next to target a JMS server.
5. When you specify that you want to create a new store in step 3, the “Select store type” page appears. You can select File Store or JDBC Store.
 - If you specify File Store, the “File store properties” page appears. When creating a file store for the JMS Persistent store, the path name to the directory must exist on your system, so be sure to create it before completing this page.
 - If you selected JDBC Store, in the “Create new JDBC Store” page, select a configured JDBC data source or configure a new JDBC data source for the store. You cannot configure a JDBC data source that is configured to support global transactions.

Configuring a JMS Server

JMS servers act as management containers for the queues and topics in JMS modules that are targeted.

This page summarizes the JMS servers that have been created.

Customize this table

JMS Servers(Filtered - More Columns Exist)

| | Name | P |
|--------------------------|------------------------|---|
| <input type="checkbox"/> | HRJMSServer | 1 |
| <input type="checkbox"/> | MedRecJMSServer_auto_1 | |
| <input type="checkbox"/> | MedRecJMSServer_auto_2 | |

Save

JMS servers act as management containers for the queues and topics in JMS modules that are targeted. Its destinations is to maintain information on what persistent store is used for any maintain the states of durable subscribers created on the destinations.

Use this page to define the general configuration parameters for this JMS server.

Name: 2 HRJMSServer

Persistent Store: (none)

Paging Directory: (empty field)

Message Buffer Size: -1

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Configuring a JMS Server

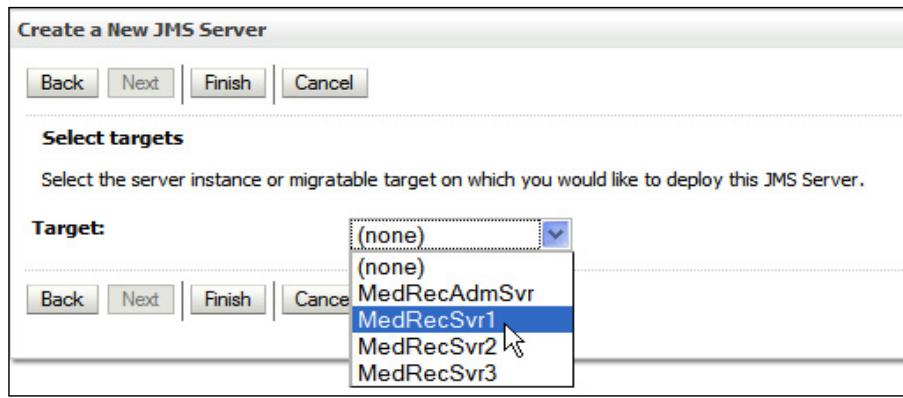
You can change the configuration of already created JMS servers or add configurations on a JMS server by performing the following steps:

1. Select **Services > Messaging > JMS Servers** from the Domain Structure pane. Locate and click the link to the JMS server that you want to configure.
2. Enter the values appropriately in the “Settings for the JMS server” page.

You can set persistent stores at the time of creating a JMS server. If you have already configured persistent stores, you can assign one of them when configuring the JMS server.

Targeting a JMS Server to a Managed Server

- By appropriately selecting managed servers, you can target where the JMS queue or topic will be managed.
- The JMS server is associated with only one WebLogic Server instance.
- If you want a JMS server on each server in a cluster, you must configure a JMS server for each server.



Copyright © 2009, Oracle. All rights reserved.

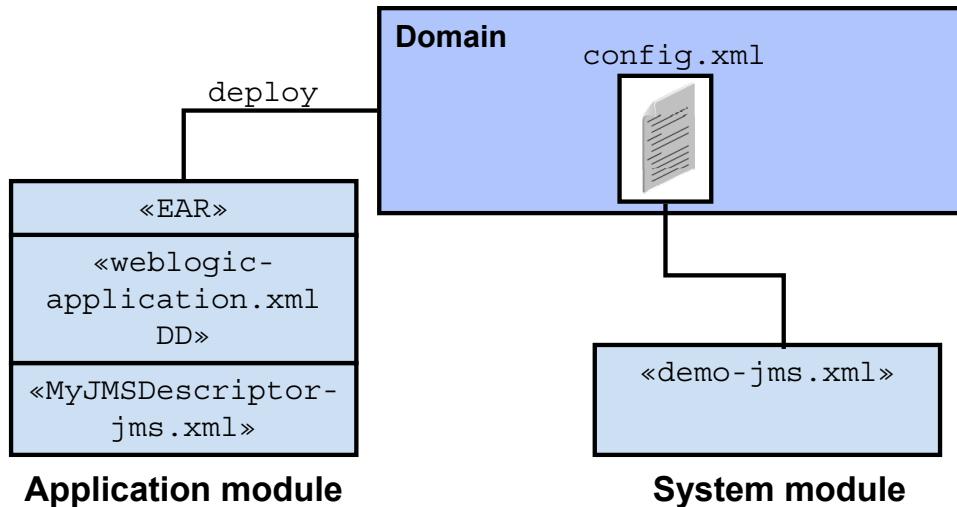
Targeting a JMS Server to a Managed Server

When the JMS server that you created is selected in the left pane, a dialog box appears in the right pane showing the tabs that are associated with this instance.

1. Select a target from the Target drop-down list.
(Note: A JMS server can be targeted to only one WebLogic Server.)
2. Click **Finish**.

JMS Modules

- JMS resources can be configured either as system modules or as application modules.
- As an administrator, you normally configure system modules.



Copyright © 2009, Oracle. All rights reserved.

ORACLE

JMS Modules

JMS modules are application-related definitions that are independent of the domain environment. You create and manage JMS resources either as system modules or application modules.

- JMS system modules are typically configured using the Administration Console or the WebLogic Scripting Tool (WLST), which adds a reference to the module in the domain's config.xml file. System modules are owned and modified by the WebLogic administrator and are available to all applications.
- JMS application modules are a WebLogic-specific extension of Java EE modules and can be deployed either with a Java EE application (as a packaged resource) or as stand-alone modules that can be made globally available. Application modules are owned and modified by WebLogic developers, who package JMS resource modules with the application's EAR file.

After the initial deployment is completed, an administrator has only limited control over the deployed applications. For example, administrators are allowed only to ensure the proper life cycle of these applications (deploy, undeploy, redeploy, remove, and so on) and tune parameters, such as increasing or decreasing the number of instances of any given application to satisfy the client needs. Other than life cycle and tuning, any modification to these applications must be completed by the application development team.

JMS Modules

- Configuration of JMS resources such as queues, topics, and connection factories are within JMS modules.
- Similar to other Java EE modules such as data sources, the configurations are in XML files that conform to the `weblogic-jmsmd.xsd` schema.
- An administrator can create and manage JMS modules as:
 - Global system resources
 - Global stand-alone modules
 - Modules packaged with an enterprise application



ORACLE

Copyright © 2009, Oracle. All rights reserved.

JMS Modules (continued)

During the process of deploying a JMS application, you link the application components to the environment-specific JMS resource definitions, such as the server instances (deployment target) that should host a given application component, and the location to use for persisting JMS messages.

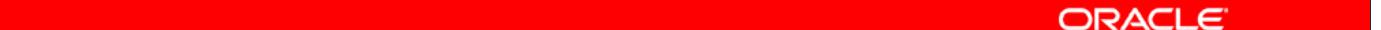
With modular deployment of JMS resources, you can migrate your application and the required JMS configuration from environment to environment, such as from a testing environment to a production environment, without opening an enterprise application file (such as an EAR file) or a stand-alone JMS module, and without extensive manual JMS reconfiguration.

JMS configuration resources, such as destinations and connection factories, are stored outside of the WebLogic domain configuration file as module descriptor files, which conform to the `weblogic-jms.xsd` schema. JMS modules do not include the JMS server definitions.

The JMS system modules must be targeted to one or more Oracle WebLogic Server instances or to a cluster. The targetable resources that are defined in a system module must also be targeted to a JMS server or the Oracle WebLogic Server instances within the scope of a parent module's targets. Additionally, the targetable JMS resources within a system module can be further grouped into *subdeployments* during the configuration or targeting process to provide further loose-coupling of the JMS resources in a WebLogic domain.

Modular JMS Resource Configuration and Deployment

- Modular deployment simplifies the task of migrating JMS resources between environments, such as:
 - From development to integration
 - From system test to production
- You can migrate your application and the required JMS configuration:
 - Without opening an EAR file
 - Without extensive manual JMS reconfiguration



ORACLE

Copyright © 2009, Oracle. All rights reserved.

Modular JMS Resource Configuration and Deployment

A subdeployment for JMS destinations is a mechanism by which queues and topics, and possibly connection factories, are grouped and targeted to a single JMS server. Queues and topics depend on the JMS servers they are targeted to for the management of persistent messages, durable subscribers, and message paging. To reconfigure a subdeployment's targets, use the parent system module's subdeployment management page.

For example, if you want to co-locate a group of queues with a connection factory that is targeted to a specific JMS server, you can associate the queues with the subdeployment that the connection factory belongs to, provided that the connection factory is not already targeted to multiple JMS servers (for example, targeted to a server instance hosting multiple JMS servers).

Creating Packaged JMS Modules: You create packaged JMS modules using an enterprise-level integrated development environment (IDE) or a development tool that supports the editing of XML descriptor files. You then deploy and manage stand-alone modules using the JSR 88-based tools, such as the `weblogic.Deployer` utility or the WebLogic Administration Console.

Deploying a Packaged JMS Module: The deployment of packaged JMS modules follows the same model as all the other components of an application: individual modules can be deployed to a single server, a cluster, or individual members of a cluster.

Connection Factories

- JMS connection factories are used to set default client connection parameters, including:
 - Message priority
 - Message time-to-live (TTL)
 - Message persistence
 - Transactional behavior
 - Acknowledgement policy
 - Flow control
- WLS provides a default client connection factory that:
 - Uses WebLogic's default connection settings
 - Is located on the server JNDI tree at `weblogic.jms.ConnectionFactory`



ORACLE

Copyright © 2009, Oracle. All rights reserved.

Connection Factories

Connection factories are resources that enable JMS clients to create JMS connections. A connection factory supports concurrent use, enabling multiple threads to access the object simultaneously. WebLogic JMS provides preconfigured default connection factories that can be enabled or disabled on a per-server basis. You can also configure one or more connection factories to create connections with predefined options that better suit your application.

Some connection factory options are dynamically configurable. You can modify the following parameters for connection factories:

- General configuration parameters, including modifying the default client parameters, default message delivery parameters, load-balancing parameters, unit-of-order parameters, and security parameters
- Transaction parameters, which enable you to define a value for the transaction timeout option and to indicate whether an XA queue or XA topic connection factory is returned, and whether the connection factory creates sessions that are XA-aware
- Flow control parameters, which enable you to tell a JMS server or a destination to slow down message producers when it determines that it is becoming overloaded

When connection factory options are modified at run time, only the incoming messages are affected; stored messages are not affected.

Connection Factories (continued)

Within each JMS module, the connection factory resource names must be unique. All connection factory JNDI names in any JMS module must be unique across an entire WebLogic domain. Oracle WebLogic Server adds the connection factory names to the JNDI space during startup, and the application then retrieves a connection factory using the WebLogic JNDI APIs.

You can establish clusterwide, transparent access to JMS destinations from any server in the cluster, either by using the default connection factories for each server instance or by configuring one or more connection factories and targeting them to one or more server instances in the cluster. This way, each connection factory can be deployed on multiple Oracle WebLogic Server instances.

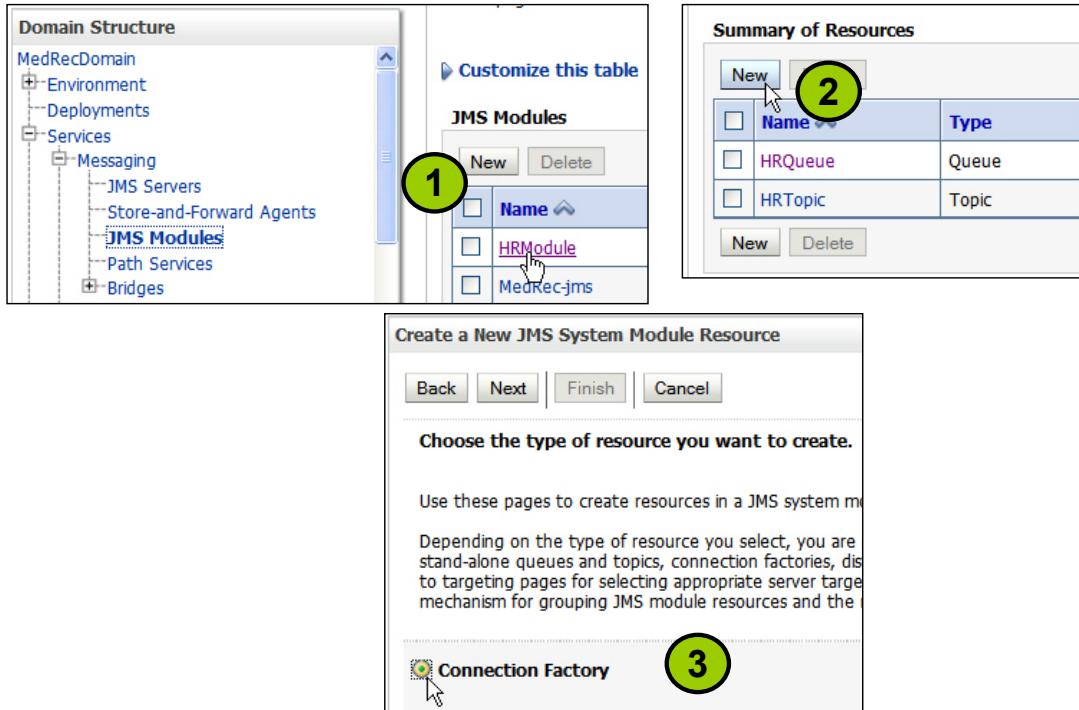
Using a Default Connection Factory

Oracle WebLogic Server defines two default connection factories, which can be looked up using the following JNDI names:

- `weblogic.jms.ConnectionFactory`
- `weblogic.jms.XAConnectionFactory`

You need to configure a new connection factory only if the preconfigured settings of the default factories are not suitable for your application. The main difference between the preconfigured settings for the default connection factories and a user-defined connection factory is the default value for the XA Connection Factory Enabled option to enable JTA transactions. Also, using default connection factories means that you have no control over targeting the Oracle WebLogic Server instances where the connection factory may be deployed. However, you can enable or disable the default connection factories on a per-Oracle WebLogic Server basis.

Creating a Connection Factory



ORACLE

Copyright © 2009, Oracle. All rights reserved.

Creating a Connection Factory

Within each JMS module, the connection factory resource names must be unique. All the connection factory JNDI names in any JMS module must be unique across an entire WebLogic domain.

1. In the Administration Console, expand Services > Messaging, and click **JMS Modules**. Select an existing JMS module.
2. In the Summary of Resources table click **New**.
3. Select the Connection Factory resource type and click **Next**.
4. (*Not shown*) Enter Name and JNDI Name for the new connection factory, and click **Next**.
5. (*Not shown*) For basic default targeting, accept the default targets that are presented in the Targets box, and then click **Finish**. For advanced targeting, click **Advanced Targeting**, which allows you to select an existing subdeployment or to create a new one. When a valid subdeployment is selected, its targeted JMS servers, servers, or cluster appear as selected in the Targets box.

Configuring a Connection Factory

Settings for HR_Conn_Fact

Configuration Subdeployment Notes

General Default Delivery Client Transactions Flow Control Load Balance

Save

Use this page to define the default delivery configuration parameters for this JMS connection factory, such as priority, time-to-live, etc.

| | |
|--------------------------------|------------|
| Default Priority: | 4 |
| Default Time-to-Live: | 0 |
| Default Time-to-Deliver: | 0 |
| Default Delivery Mode: | Persistent |
| Default Redelivery Delay: | 0 |
| Default Compression Threshold: | 2147483647 |

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Configuring a Connection Factory

In the Administration Console, navigate to the connection factory resource that you want to configure. Click the Configuration tab as shown in the slide. You can configure the properties using the Default Delivery subtab.

- **Default Priority:** The default priority that is used for messages when a priority is not explicitly defined. Values are between 0 and 9.
- **Default Time-to-Live:** The maximum length of time, in milliseconds, that a message will exist. This value is used for messages when a priority is not explicitly defined. A value of 0 indicates that the message has an infinite amount of time to live.
- **Default Time-to-Deliver:** The delay time, in milliseconds, between when a message is produced and when it is made visible on its destination
- **Default Delivery Mode:** Whether or not messages should use a persistent store, if one is associated with the JMS server
- **Default Redelivery Delay:** The delay time, in milliseconds, before rolled back or recovered messages are redelivered
- **Send Timeout:** The maximum length of time, in milliseconds, that a sender will wait when there is not enough space available (no quota) on a destination to accommodate the message being sent. The default time is 10 milliseconds.

Destination

- A destination is a lightweight object that is stored in JNDI.
- It is the target on a JMS server for sending messages and the location from where messages will be consumed.
- The JMS destination types are:
 - Queue (for the point-to-point model)
 - Topic ((for the Publish/Subscribe model)



Copyright © 2009, Oracle. All rights reserved.

Destination

A JMS destination identifies a queue (point-to-point) or topic (Publish/Subscribe) for a JMS server.

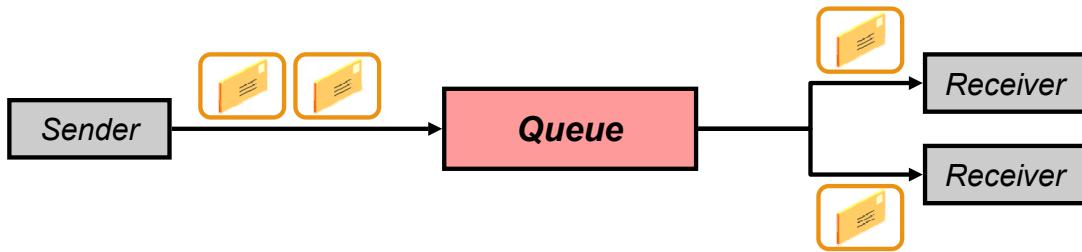
After configuring a JMS server, configure one or more queue or topic destinations for each JMS server. You configure destinations explicitly or by configuring a destination template that can be used to define multiple destinations with similar attribute settings.

A JMS destination identifies a queue (point-to-point) or topic (Publish/Subscribe) resource within a JMS module. Each queue and topic resource is targeted to a specific JMS server. A JMS server's primary responsibility for its targeted destinations is to maintain information about the persistent store that is used for any persistent messages that arrive on the destinations and to maintain the states of the durable subscribers created on the destinations.

Queue Destinations

In JMS point-to-point messaging, note the following:

- Clients communicate with a queue destination.
- Messages are distributed to consumers in a serial fashion (first in, first out).
- Each message is delivered only to a single consumer.



ORACLE

Copyright © 2009, Oracle. All rights reserved.

Queue Destinations

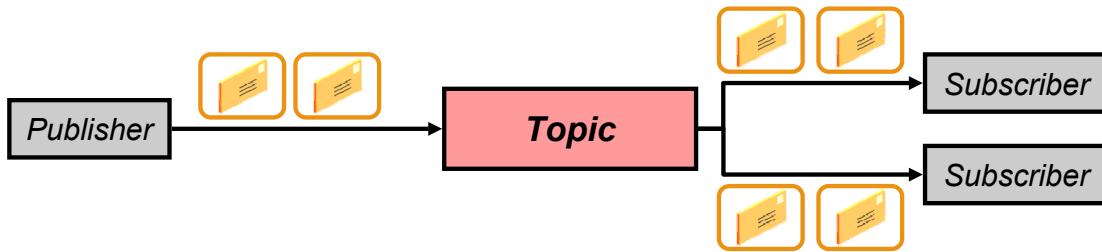
The point-to-point (PTP) messaging model enables one application to send a message to another. PTP messaging applications send and receive messages using named queues. A queue sender (producer) sends a message to a specific queue. A queue receiver (consumer) receives messages from a specific queue. Multiple queue senders and queue receivers can be associated with a single queue, but an individual message can be delivered to only one queue receiver.

If multiple queue receivers are listening for messages on a queue, WebLogic JMS determines which one will receive the next message on a first come, first serve basis. If no queue receivers are listening on the queue, messages remain in the queue until a queue receiver attaches to the queue.

Topic Destinations

In JMS Publish/Subscribe messaging, the following is true:

- Clients communicate with a topic destination.
- Messages are broadcast to all subscribers.
- A message can be saved until at least one subscriber has consumed it (“durable”).



ORACLE

Copyright © 2009, Oracle. All rights reserved.

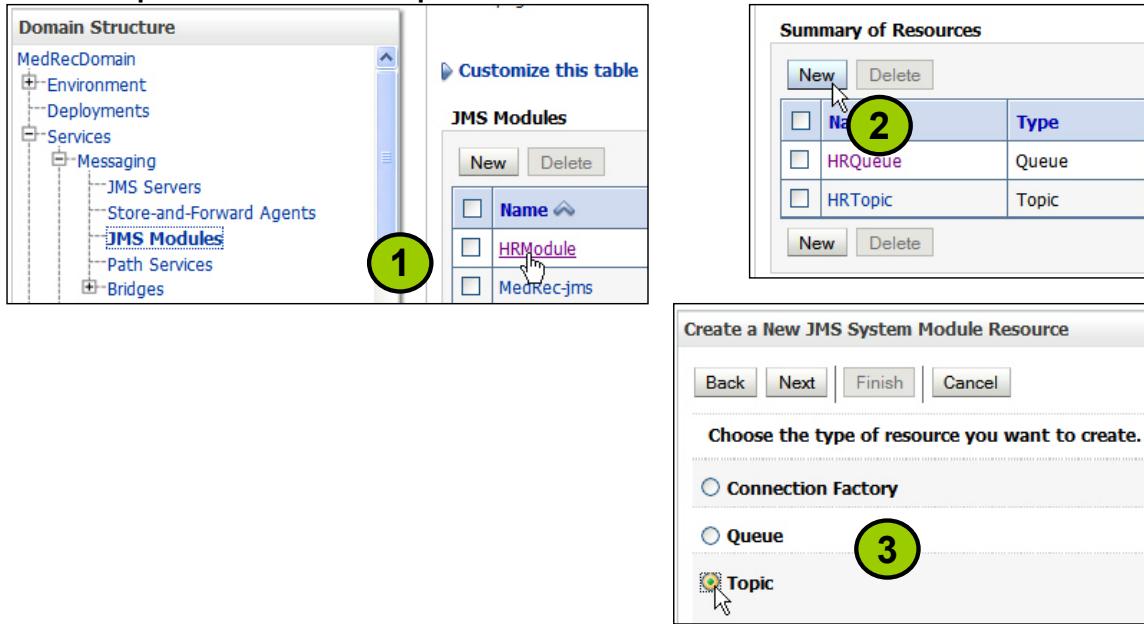
Topic Destinations

The Publish/Subscribe (pub/sub) messaging model enables an application to send a message to multiple applications. Pub/sub messaging applications send and receive messages by subscribing to a topic. A topic publisher (producer) sends messages to a specific topic. A topic subscriber (consumer) retrieves messages from a specific topic. Unlike the PTP messaging model, the pub/sub messaging model allows multiple topic subscribers to receive the same message. JMS retains the message until all topic subscribers have received it.

The pub/sub messaging model supports durable subscribers. For durable subscriptions, WebLogic JMS stores a message in a persistent file or database until the message is delivered to the subscribers or has expired, even if those subscribers are not active at the time the message is delivered. To support durable subscriptions, a client identifier (client ID) must be defined for the connection by the JMS client application. Support for durable subscriptions is a feature that is unique to the pub/sub messaging model, so client IDs are used only with topic connections; queue connections also contain client IDs, but JMS does not use them.

Creating a Destination (Topic)

- The steps to create a topic are shown here.
- Steps to create a queue are also similar.



ORACLE

Copyright © 2009, Oracle. All rights reserved.

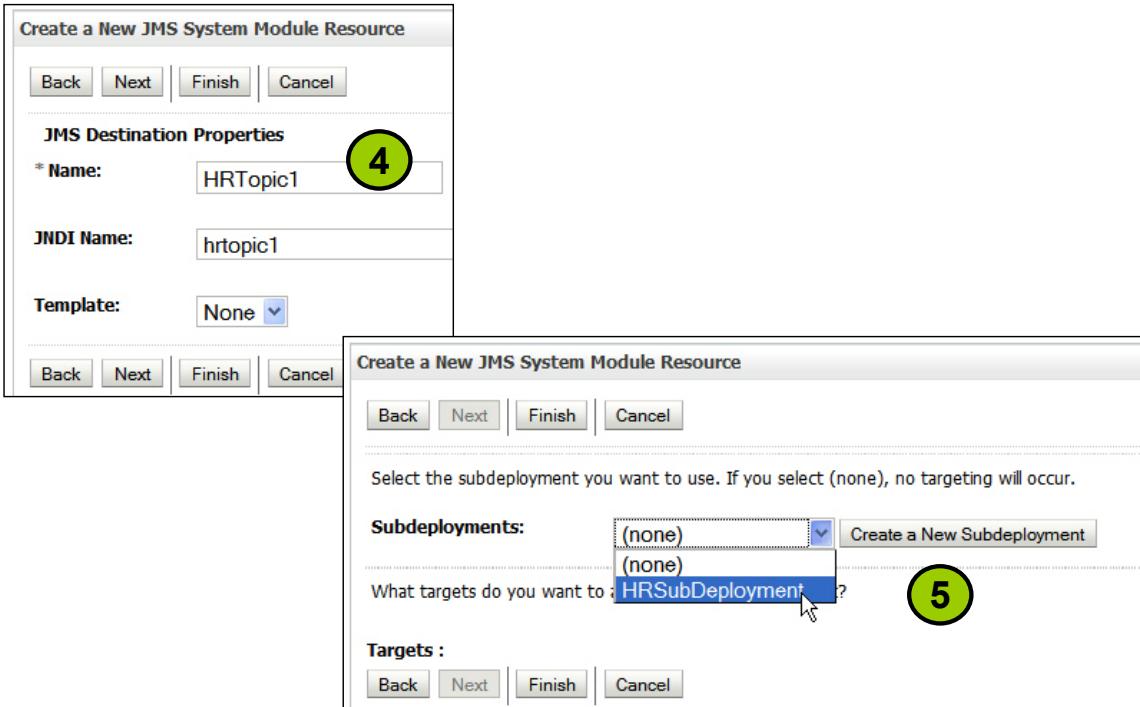
Creating a Destination (Topic)

After you create a JMS system module, you can configure resources for the module, including stand-alone queues and topics, distributed queues and topics, connection factories, JMS templates, destination sort keys, destination quota, foreign servers, and JMS store-and-forward (SAF) parameters.

For each destination, you can optionally select a subdeployment or create a new subdeployment for the resource. A subdeployment is a mechanism by which targetable JMS module resources (such as queues, topics, and connection factories) are grouped and targeted to a server resource (such as JMS servers, server instances, or a cluster).

- In the Administration Console, expand Services > Messaging and click **JMS Modules**. Select an existing JMS module.
- On the Configuration page, click **New** above the Summary of Resources table.
- Select the type of destination to create: Queue or Topic. Click **Next**.

Creating a Destination (Topic)

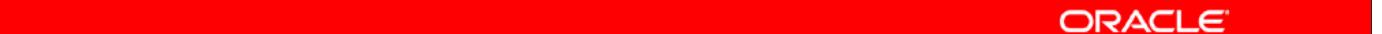


Creating a Destination (Topic) (continued)

4. Enter Name and JNDI Name for the destination. Click Next.
5. Select an existing Subdeployment from this JMS module. Your new JMS destination will be targeted to the JMS servers indicated by the subdeployment.

Threshold, Quota, and Paging

- Thresholds and Quotas enable you to control the size and number of message flow through JMS Servers.
- A threshold is a limit that triggers flow control, and logged warnings.
- A quota is a limit defined for the JMS-administered objects; it includes the following values:
 - The maximum number of bytes that can be stored
 - The maximum number of messages that can be stored
- The Message Paging feature enables automatic clearing of virtual memory especially for non-persistent messages.
- You can specify an appropriate folder structure for writing paged-out messages.

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Threshold and Quota

With the Flow Control feature, you can direct a JMS server or destination to slow down message producers when it determines that it is becoming overloaded. Flow control thresholds attributes used for configuring size and number of message thresholds for the JMS server and its destinations. When the upper or lower threshold is reached, triggered events are launched. .

Quotas limit the number of messages or the size of all the messages that can be stored. Messages sent that would put the intended target over its quota are rejected, and the sender receives an exception. Quotas enable administrators to control the size of the backlog.

Paging

With the message paging feature, JMS servers automatically attempt to free up virtual memory during peak message load periods. Message paging is always enabled on JMS servers, and so a message paging directory is automatically created without having to configure one. You can specify a directory and paged-out messages are written to files in this directory.

If a JMS server is associated with a file store (either user-defined or the server's default store), paged persistent messages are generally written to that file store, while non-persistent messages are always written to the JMS server's paging directory.

Configuring Thresholds and Quotas

| Settings for HRJMSServer | |
|--|------------------------------|
| Configuration Logging Targets Monitoring Control Notes | |
| General | Thresholds and Quotas |
| Session Pools | |
| Thresholds | |
| Bytes Threshold High: | -1 |
| Bytes Threshold Low: | -1 |
| Messages Threshold High: | -1 |
| Messages Threshold Low: | -1 |
| Quotas | |
| Bytes Maximum: | -1 |
| Messages Maximum: | -1 |
| Blocking Send Policy: | FIFO |
| Maximum Message Size: | 2147483647 |

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Configuring Thresholds and Quotas

After you have created either servers or destinations, you may configure their thresholds and quotas. This slide shows a sample panel for a queue. A value of –1 means that the threshold is disabled or there is no quota limit.

- **Bytes Threshold High:** The upper byte threshold beyond which specified JMS events are triggered
- **Bytes Threshold Low:** The lower byte threshold below which specified JMS events are triggered
- **Messages Threshold High:** The upper threshold to trigger events based on the number of messages stored in the destination
- **Messages Threshold Low:** The lower threshold that triggers events based on the number of messages stored in the destination
- **Bytes Maximum:** The maximum number of bytes that may be stored in this destination
- **Messages Maximum:** The maximum number of messages that may be stored in the destination
- **Blocking Send Policy:** Determines whether smaller messages are delivered before larger ones when a destination has exceeded its maximum number of messages
 - FIFO prevents the JMS server from delivering smaller messages when larger ones are already waiting for space.
 - Preemptive allows smaller send requests to preempt previous larger ones when there is sufficient space for smaller messages on the destination.

Road Map

- Oracle WebLogic Server JMS administration
- Configuring JMS objects
- Durable subscribers and persistent messaging
 - Durable subscribers
 - Configuring durable subscribers
 - Persistent and nonpersistent messages
 - Persistent backing stores using the Console
- Monitoring JMS

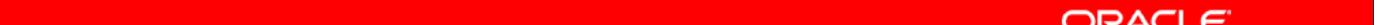


ORACLE®

Copyright © 2009, Oracle. All rights reserved.

Durable Subscribers and Subscriptions

- Durable subscribers register durable subscriptions for guaranteed message delivery even if the subscribers are inactive.
- A subscriber is considered active if the Java object that represents it exists.
- By default, subscribers are nondurable.
- Administrators:
 - Specify where the messages are persisted
 - Configure persistent connection factories and destinations



ORACLE

Copyright © 2009, Oracle. All rights reserved.

Durable Subscribers and Subscriptions

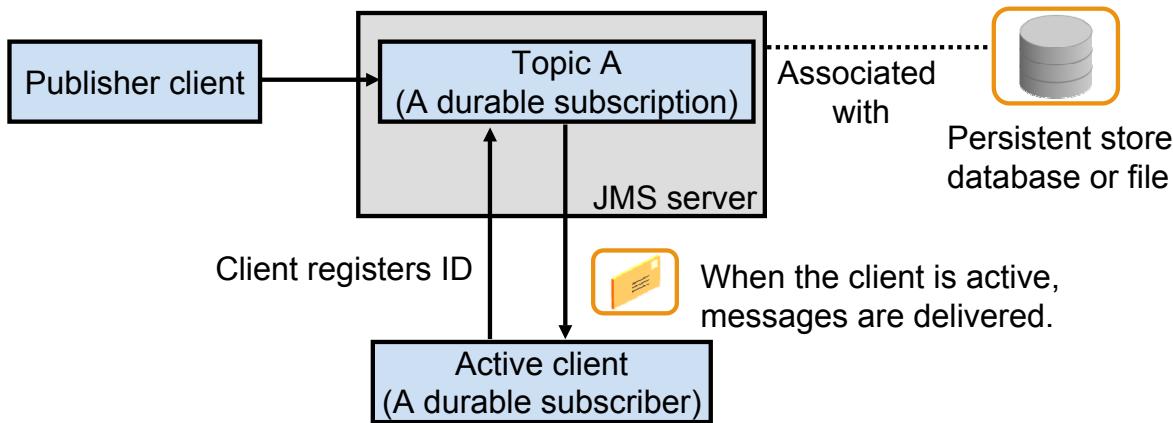
Nondurable subscriptions last for the lifetime of their subscriber object. That is, a client will see the messages published on a topic only while its subscriber is active. An inactive subscriber does not see messages that are published on its topic.

Support for durable subscriptions is a feature that is unique to the Publish/Subscribe messaging model. Client IDs are used only with topic connections.

An inactive durable subscription is one that exists but does not currently have a message consumer subscribed to it.

How a Durable Subscription Works

- Durable subscription is effective only when the client is inactive during the time that the message is published.
- When the client becomes active again, its ID is used to retrieve and redeliver messages.



ORACLE

Copyright © 2009, Oracle. All rights reserved.

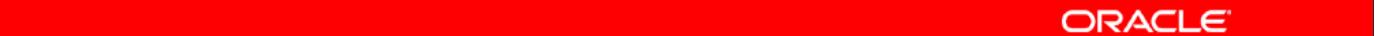
How a Durable Subscription Works

A durable subscriber registers a durable subscription with a unique identity that is retained by JMS. Subsequent subscriber objects with the same identity resume the subscription in the state it was left in by the previous subscriber. If there is no active subscriber for a durable subscription, JMS retains the subscriber's messages until they are received by the subscriber or until they expire.

Sessions with durable subscribers must always provide the same client identifier. Each client must specify a name that uniquely identifies (within the client identifier) each durable subscription that it creates. Only one session at a time can have a TopicSubscriber for a particular durable subscription.

Configuring a Durable Subscription

- To configure durable subscriptions, an administrator must:
 - Create and configure a JMS store
 - Configure connection factories or destinations as persistent
 - Associate the JMS store with the JMS server
- The JMS store can be configured to use either of the following:
 - A file store
 - A JDBC Store (a connection pool)

The red bar spans most of the width of the slide, centered horizontally.

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Configuring a Durable Subscription

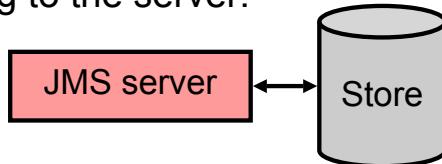
No two JMS servers can use the same backing store.

File persistence is much faster than JDBC because JDBC persistence relates to reads from and writes to a database that could potentially be a bottleneck for your system. Synchronization occurs one by one. To enhance the speed and efficiency of persisting to a database, you may like to consider the use of Oracle Coherence.

JMS backing stores can increase the amount of memory required during the initialization of an Oracle WebLogic Server as the number of stored messages increases. If initialization fails due to insufficient memory, when you are rebooting an Oracle WebLogic Server, increase the heap size of the Java Virtual Machine (JVM) proportional to the number of messages that are stored in the JMS backing store. Try to reboot again.

Persistent Messaging

- A persistent store is a physical repository for storing persistent JMS messages.
- WebLogic JMS writes persistent messages to a disk-based file or JDBC-accessible database.
- WebLogic supports guaranteed messaging using persistent stores:
 - In-progress messages can be delivered despite server restart.
 - Topic subscribers can consume missed messages despite reconnecting to the server.



ORACLE

Copyright © 2009, Oracle. All rights reserved.

Persistent Messaging

A persistent message is guaranteed to be delivered only once. It is not considered sent until it has been safely written to a WebLogic persistent store that is assigned to each JMS server during configuration.

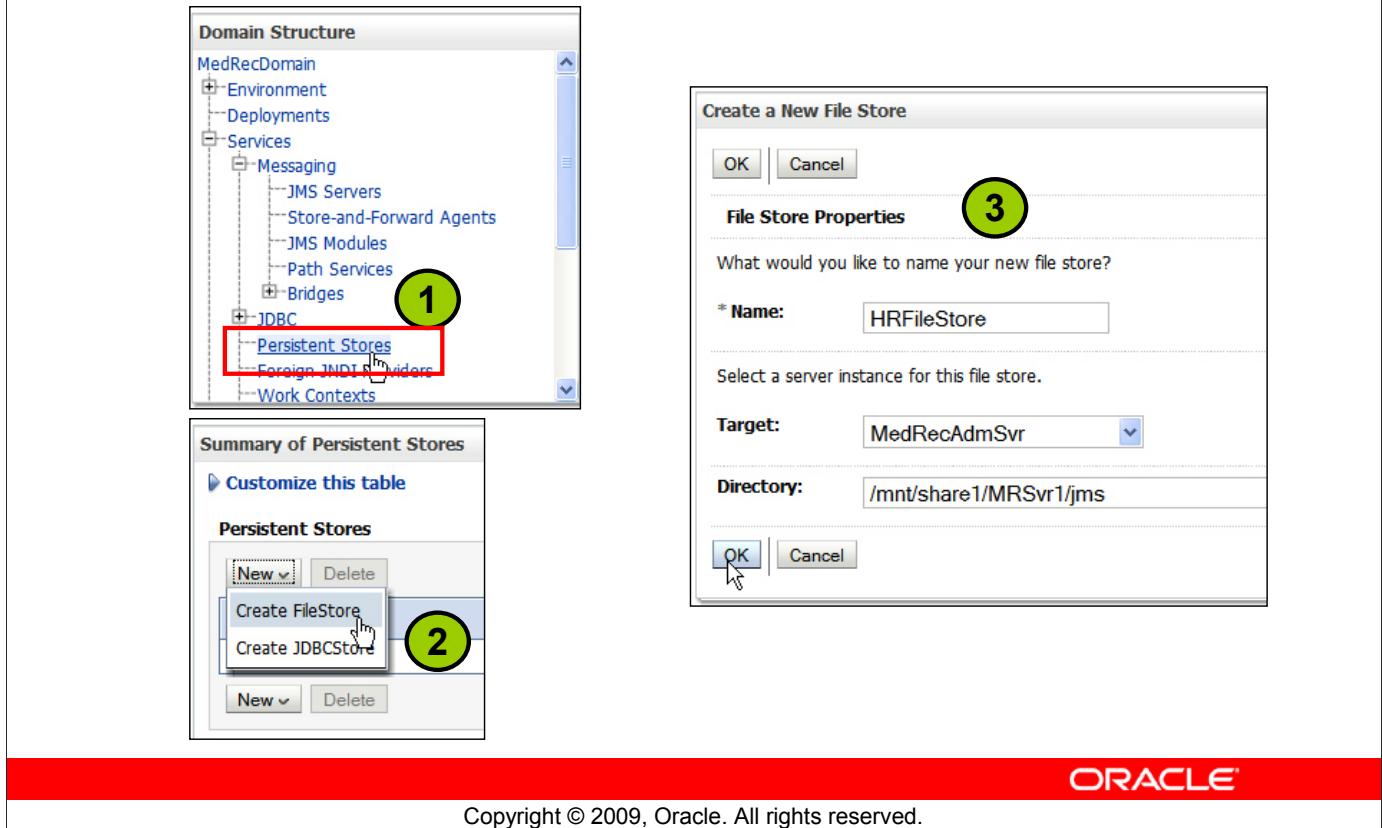
Nonpersistent messages are not stored. If a connection is closed or recovered, all nonpersistent messages that have not yet been acknowledged will be redelivered. After a nonpersistent message is acknowledged, it will not be redelivered.

WebLogic persistent stores provide built-in, high-performance storage solutions for the Oracle WebLogic Server subsystems and services that require persistence. For example, they can store persistent JMS messages or temporarily store messages that are sent using the JMS store-and-forward feature. The persistent store supports persistence to a file-based store or to a JDBC-enabled database. Each server instance, including the administration server, has a default persistent store that requires no configuration. The default store is a file-based store that maintains its data in a group of files in the `data\store\default` directory of a server instance.

Configure persistent messaging if:

- Development requires durable subscriptions (use durable subscribers in the application)
- You require that in-progress messages persist across server restarts

Creating a JMS Store



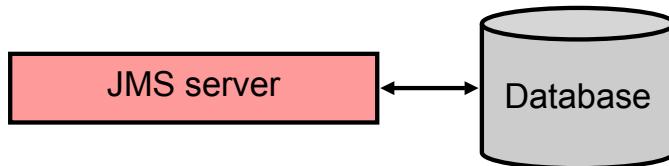
Creating a JMS Store

There is a default store for every WebLogic Server instance. The default store can be configured, but cannot point to a database. You may need to create a custom store to point to a database. Similarly, you may want to create a custom file store on your choice of storage device that can enable you to migrate the store to another server member in a cluster. When configuring a file store directory, the directory must be accessible to the server instance on which the file store is located.

1. In the left pane of the console, expand Services and select **Persistent Stores**.
2. On the Summary of Persistent Stores page, select the store type from the New list.
3. If File Store is selected, update the following on the Create a new File Store page:
 - **Name:** Name of the store
 - **Target:** Server instance on which to deploy the store
 - **Directory:** Path name to the directory on the file system where the file store is placed. This directory must exist on your system, so be sure to create it before completing this tab.

Creating a JDBC Store for JMS

- You can create a persistent store to a database using JDBC Store.
- To configure JMS JDBC persistence, perform the following:
 - Create a JDBC DataSource.
 - Create a JDBC Store and refer to the JDBC DataSource.
 - Refer to the JMS store from the JMS server configuration.
- The required infrastructure (tables and so on) is created automatically using Data Definition Language (DDL).



ORACLE

Copyright © 2009, Oracle. All rights reserved.

Creating a JDBC Store for JMS

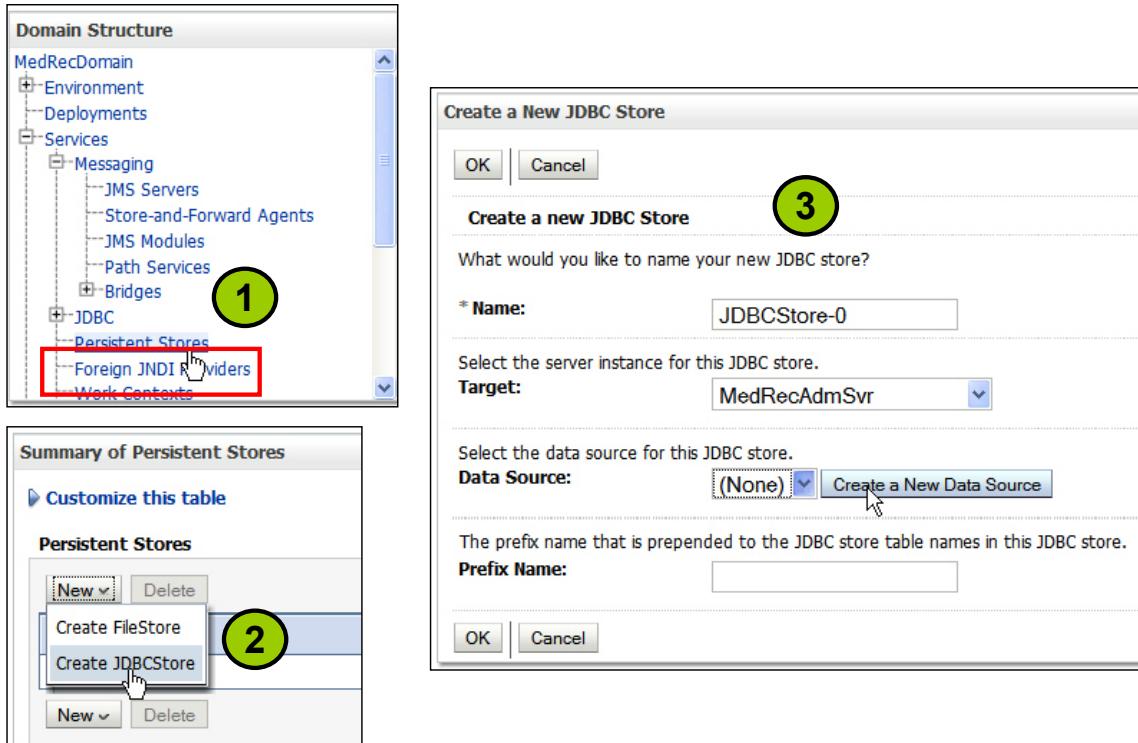
When using JMS JDBC Store, use a separate schema.

The JDBC Store Configuration page provides an optional “Create Table from DDL File” option with which you can access a preconfigured DDL file that is used to create the JDBC Store’s backing table (WLStore). This option is ignored when the JDBC Store’s backing table exists. It is mainly used to specify a custom DDL file created for an unsupported database or when upgrading the JMS JDBC Store tables from a prior release to a current JDBC Store table.

If a DDL file name is *not* specified in the “Create Table from DDL File” field, and the JDBC Store detects that its backing table does not exist, the JDBC Store automatically creates the table by executing a preconfigured DDL file that is specific to the database vendor.

If a DDL file name is specified in the “Create Table from DDL File” field and the JDBC Store detects that its backing table does not already exist, the JDBC Store searches for the specified DDL file in the file path first, and then, if not found, searches for the DDL file in the CLASSPATH. After it is found, the SQL within the DDL file is executed to create the JDBC Store’s backing table. If the configured file is not found and the table does not already exist, the JDBC Store fails to boot.

Creating a JMS JDBC Store



ORACLE

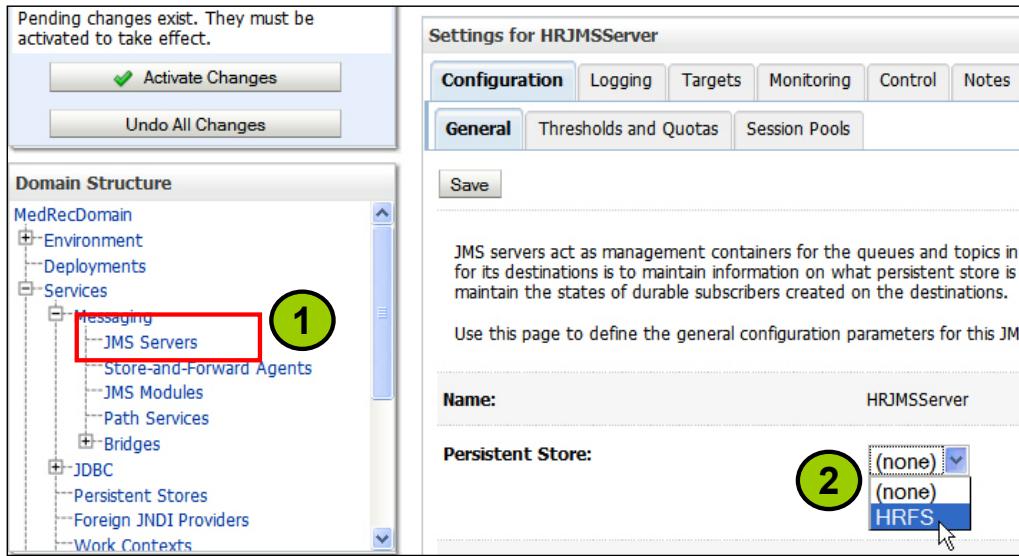
Copyright © 2009, Oracle. All rights reserved.

Creating a JMS JDBC Store

Prefix Name: The prefix for table names, if:

- The database management system requires fully qualified names, such as schema
- You must differentiate between the JMS tables for two Oracle WebLogic Servers to store multiple tables on one DBMS

Assigning a Store to a JMS Server



ORACLE

Copyright © 2009, Oracle. All rights reserved.

Assigning a Store to a JMS Server

Associate your new custom persistent store with a JMS server by using the Persistent Store field of the Configuration > General tab. If this field is set to “(none),” the JMS server uses the default file store that is automatically configured on each targeted server instance.

Persistent Connection Factory

Settings for HRCF

Configuration Subdeployment Notes

General Default Delivery Client Transactions Flow Control Load Balance Security

Save

Use this page to define the default delivery configuration parameters for this JMS connection factory. This includes delivery mode, time to live, etc.

Default Priority: 4

Default Time-to-Live: 0

Default Time-to-Deliver: 0

Default Delivery Mode: Persistent

Default Redelivery Delay: 0

Default Compression Threshold: 2147483647

Send Timeout: 10

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Persistent Connection Factory

Default Delivery Mode: Used for messages for which a delivery mode is not explicitly defined. It can be persistent or nonpersistent.

The preferred method, according to the JMS specification, is to configure the connection factory with the client ID. For Oracle WebLogic Server JMS, this means adding a separate connection factory definition during configuration for each client ID. Applications look up their own topic connection factories in JNDI and use them to create connections containing their own client IDs.

Alternatively, an application can set its client ID programmatically.

Configuring Destination Overrides

| | |
|---------------------------|-------------|
| Priority Override: | -1 |
| Time-to-Live Override: | -1 |
| Time-to-Deliver Override: | -1 |
| Delivery Mode Override: | No-Delivery |

ORACLE®

Copyright © 2009, Oracle. All rights reserved.

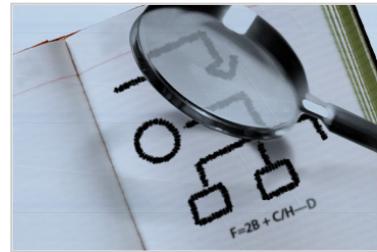
Configuring Destination Overrides

The delivery mode assigned to all messages that arrive at the destination can be set to override the delivery mode specified by the message producer. A value of No-Delivery specifies that the producer's delivery mode will not be overridden.

This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.

Road Map

- Oracle WebLogic Server JMS administration
- Configuring JMS objects
- Durable subscribers and persistent messaging
- Monitoring JMS
 - Monitoring JMS servers
 - Monitoring JMS modules



ORACLE®

Copyright © 2009, Oracle. All rights reserved.

Monitoring JMS Servers

Statistics are provided for the following JMS objects:

- JMS servers
- Connections
- Destinations

The screenshot shows a table titled "Statistics(Filtered - More Columns Exist)". The table has columns: Name, Messages Current, Messages High, Bytes Current, and Session Pools Current. There is one row for "PayrollJMSServer" with values 3, 3, 36, and 0 respectively. A "Customize this table" link is at the top left.

| Statistics(Filtered - More Columns Exist) | | | | | |
|---|------------------|------------------|---------------|---------------|-----------------------|
| Showing 1 to 1 of 1 Previous Next | | | | | |
| | Name | Messages Current | Messages High | Bytes Current | Session Pools Current |
| <input type="checkbox"/> | PayrollJMSServer | 3 | 3 | 36 | 0 |

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Monitoring JMS Servers

You can monitor the run-time statistics for an active JMS server. From the Monitoring tab, you can also access run-time information for a JMS server's destinations, transactions, connections, and server session pools.

1. Expand Services > Messaging and click JMS Servers. Select a JMS server.
2. Click the Monitoring tab. By default, a Monitoring subtab is displayed, which provides general statistics for all destinations on every JMS server in the domain. These statistics include the number and size of messages processed by the JMS server.

The Active Destinations tab displays the statistics for each active JMS destination for the domain.

The Active Transactions tab displays all active JMS transactions for the domain. For troubleshooting, you can force commits or rollbacks on selected transactions. Simply select a transaction, and then click either the Force Commit or Force Rollback button.

The Active Connections tab displays all active client JMS connections for the domain. For troubleshooting, you can destroy selected connections. Simply select a connection, and then click the Destroy button above the table.

Monitoring and Managing Destinations

This page allows you to view active destinations targeted to this JMS server.

You can suspend or resume message production and consumption.

| <input type="checkbox"/> | Name | Messages Current | Messages Pending | Messages High | Messages Received | Messages Threshold | DestinationType | State | Production Paused | Insertion Paused | Consumption Paused |
|--------------------------|--|------------------|------------------|---------------|-------------------|--------------------|----------------------------|-------|-------------------|------------------|--------------------|
| <input type="checkbox"/> | examples-jms!exampleQueue | 0 | 0 | 0 | 0 | 0 | advertised_in_local_jndi | false | false | false | |
| <input type="checkbox"/> | examples-jms!exampleTopic | 0 | 0 | 0 | 0 | 0 | advertised_in_local_jndi | false | false | false | |
| <input type="checkbox"/> | examples-jms!jms/MULTIDATASOURCE_MDB_QUEUE | 0 | 0 | 0 | 0 | 0 | advertised_in_local_jndi | false | false | false | |
| <input type="checkbox"/> | examples-jms!quotes | 0 | 0 | 0 | 0 | 0 | advertised_in_cluster_jndi | false | false | false | |

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Monitoring and Managing Destinations

For troubleshooting, you can temporarily pause all run-time message production, insertion (in-flight messages), and consumption operations on any or all destinations targeted to the selected JMS server. These “message pausing” options enable you to assert administrative control over the JMS subsystem behavior in the event of an external resource failure.

The available columns include:

- **Messages Current:** The current number of messages in the destination. This does not include the pending messages.
- **Messages Pending:** The number of pending messages in the destination. A pending message is one that has either been sent in a transaction and not been committed or that has been received and not been committed or acknowledged.
- **Messages High:** The peak number of messages in the destination since the last reset
- **Messages Received:** The number of messages received in this destination since the last reset
- **Messages Threshold:** The amount of time in the threshold condition since the last reset
- **Consumers Current:** The current number of consumers accessing this destination

Monitoring Queues

- In the Administration console, navigate to Services > Messaging > JMS Modules.
- In the JMS Modules table, click the JMS module you have created.
- In the Summary of Resources table, click the link to your queue, and then click the Monitoring tab.
- The Messages High and Messages Total columns show nonzero values indicating that messages have been received.

| Destinations (Filtered - More Columns Exist) | | | | | | |
|--|----------------------------------|-------------------------------------|----------------|-----------------|---------------|----------------|
| | | Showing 1 to 1 of 1 Previous Next | | | | |
| | Name | Consumers Current | Consumers High | Consumers Total | Messages High | Messages Total |
| <input type="checkbox"/> | dizzyworldModuleIdizzyworldQueue | 0 | 0 | 0 | 1 | 1 |
| Show Messages | | Showing 1 to 1 of 1 Previous Next | | | | |

ORACLE®

Copyright © 2009, Oracle. All rights reserved.

Monitoring Queues

Use this page to view run-time statistics about the current queue resource. Run-time statistics include counts, pending, and threshold data for consumers, bytes, and messages for the queue.

To access the queue's message management page, select the check box next to its name, and then click the Show Messages button.

Viewing Active Queues and Topics

In the Administration Console, navigate to the JMS Modules and click the Active Destinations tab.

| Name | Messages Current | Messages Pending | Messages High | Messages Received | Messages Threshold | DestinationType | State | Production Paused | Insertion Paused | Consumption Paused |
|--|------------------|------------------|---------------|-------------------|--------------------|----------------------------|-------|-------------------|------------------|--------------------|
| examples-jms:exampleQueue | 0 | 0 | 0 | 0 | 0 | advertised_in_local_jndi | false | false | false | false |
| examples-jms:exampleTopic | 0 | 0 | 0 | 0 | 0 | advertised_in_local_jndi | false | false | false | false |
| examples-jms:jms/MULTIDATASOURCE_MDB_QUEUE | 0 | 0 | 0 | 0 | 0 | advertised_in_local_jndi | false | false | false | false |
| examples-jms:quotes | 0 | 0 | 0 | 0 | 0 | advertised_in_cluster_jndi | false | false | false | false |

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Viewing Active Queues and Topics

You can use this page to monitor information about a JMS consumer, which receives messages from a JMS queue (QueueReceiver) or topic (TopicSubscriber).

You can show fewer or additional data points on this page by expanding “Customize this table” and modifying the Column Display list. Each data point displays in its own table column.

Managing Messages in a Queue

- You can enable messages to be viewed in the Administration Console.
- After they are enabled, you can view and manage the messages in a queue using the Administration Console.

The screenshot shows the 'Settings for HRQueue' page in the Administration Console. The 'Monitoring' tab is selected. On the left, there's a 'Customize this table' section and a 'Destinations' table. In the 'Destinations' table, the 'HRModule!HRQueue' row has its 'Name' column checked, and a 'Show Messages' button is visible. On the right, a larger window titled 'Customize this table' displays the 'JMS Messages(Filtered - More Columns Exist)' table. It has columns for 'ID', 'CorrId', and 'Time Stamp'. Two rows are listed, both with ID values starting with 'ID:<28135.1238368149351.0>' and a timestamp of 'Sun Mar 29 19:09:09 EDT 2009'. Below the table are 'New', 'Delete', 'Move', 'Import', and 'Export' buttons.

| ID | CorrId | Time Stamp |
|----------------------------|--------|------------------------------|
| ID:<28135.1238368149351.0> | | Sun Mar 29 19:09:09 EDT 2009 |
| ID:<28135.1238368149351.0> | | Sun Mar 29 19:09:09 EDT 2009 |

Managing Messages in a Queue

You can enable viewing of messages in the Administration Console using these steps:

1. In the Administration Console, navigate to the queue resource that you want to configure:
 - Navigate to JMS Resources in System Modules, and then to JMS resources in an application module
2. Click the Monitoring tab.
3. Select the check box next to the queue, and then click Show Messages to access the queue's JMS Messages table.
4. You can then perform the following administrative procedures on a specific message or selected messages:
 - Click a message in the queue to open the View Contents page, where you can view the contents of a JMS message.
 - Click New to create a new JMS message. You can specify header and body content when creating the message, which will then be produced on the current queue.
 - Select messages for deletion and click Delete to delete them from the current queue.
 - Click Move to transfer selected JMS messages from the current queue to another destination, including a destination on a different JMS server.

Quiz

Which are the correct messaging model and JMS destination type associations?

1. Queue: Publish/Subscribe
2. Queue: Point-to-Point
3. Topic: Publish/Subscribe
4. Topic: Point-to-Point



Copyright © 2009, Oracle. All rights reserved.

Answers: 2, 3

Remember that a JMS queue is for simple point-to-point messaging, whereas a topic is for Publish/Subscribe messaging in which messages are broadcast to all listening consumers.

Quiz

Which are the available resource types within an Oracle WebLogic Server JMS module?

1. Connection factory
2. Queue
3. Topic
4. Server
5. Store



Copyright © 2009, Oracle. All rights reserved.

Answers: 1, 2, 3

Remember that JMS destinations (queues and topics) and connection factories are commonly deployed as part of a JMS module.

Summary

In this lesson, you should have learned how to:

- Describe how Oracle WebLogic Server JMS is implemented
- Configure JMS server
- Configure connection factories
- Configure queues and topics
- Configure persistent messages
- Monitor JMS resources and messages



Copyright © 2009, Oracle. All rights reserved.

Practice Overview: Configuring JMS Resources

This practice covers the following topics:

- Configuring JMS resources such as:
 - JMS server, JMS module, queue, and topic
- Posting messages to the queue and topic
- Monitoring a queue in the Administration Console



Copyright © 2009, Oracle. All rights reserved.

15

Introduction to Clustering

ORACLE®

Copyright © 2009, Oracle. All rights reserved.

Objectives

After completing this lesson, you should be able to do describe the following:

- Benefits of Oracle WebLogic cluster
- Basic cluster architecture
- Multitier cluster architecture
- Communication among clustered server instances
- Key criteria for selecting suitable cluster architecture



Copyright © 2009, Oracle. All rights reserved.

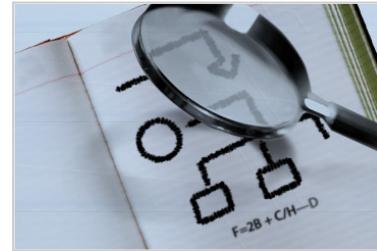
Objectives

Scenario

Clustering provides availability and scalability benefits. As the administrator at MedRec, you want to understand the benefits of clustering and the architectural considerations to help you decide on the appropriate structure for your environment.

Road Map

- Oracle WebLogic cluster introduction
 - What is a cluster?
 - Benefits of clustering
 - HTTP clustering and proxy plug-in
 - Introduce EJB clustering
- Cluster architecture
- Cluster communication



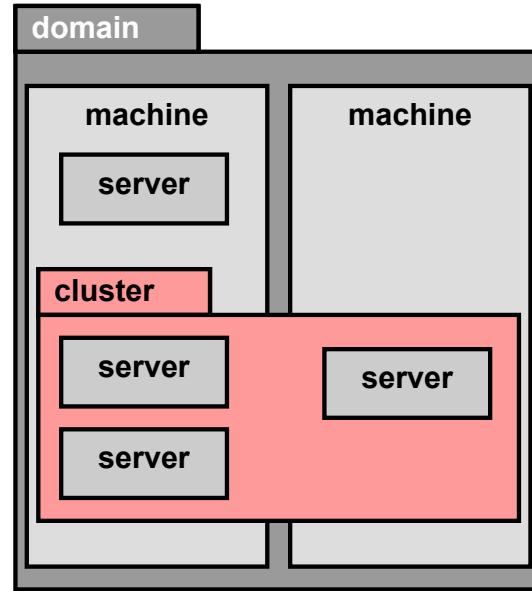
ORACLE®

Copyright © 2009, Oracle. All rights reserved.

What Is a Cluster?

A cluster:

- Is a logical group of managed servers within a domain
- Supports features to provide high availability for:
 - Whole servers
 - Web applications and services
 - EJB applications
 - JDBC resources
 - JMS
- Is transparent to clients



ORACLE

Copyright © 2009, Oracle. All rights reserved.

What Is a Cluster?

An Oracle WebLogic Server cluster consists of one or more Oracle WebLogic Server instances running simultaneously and working together to provide increased scalability and reliability. A cluster appears to clients as one Oracle WebLogic Server instance. The server instances that constitute a cluster can run on one machine or on different machines.

By replicating the services provided by one instance, an enterprise system achieves a fail-safe and scalable environment. It is good practice to set all the servers in a cluster to provide the same services.

You can increase a cluster's capacity by adding server instances to the cluster on an existing machine, or by adding machines to the cluster to host the incremental server instances.

The clustering support for different types of applications is as follows:

- For Web applications, the cluster architecture enables replicating the HTTP session state of clients. You can balance the Web application load across a cluster by using an Oracle WebLogic Server proxy plug-in or an external load-balancer.
- For Enterprise JavaBeans (EJBs) and Remote Method Invocation (RMI) objects, clustering uses the object's replica-aware stub. When a client makes a call through a replica-aware stub to a service that fails, the stub detects the failure and retries the call on another replica.
- For JMS applications, clustering supports clusterwide transparent access to destinations from any member of the cluster.

Benefits of Clustering

| Concept | Description |
|----------------------|---|
| Scalability | It provides more capacity for an application by adding servers, without having to make major architectural changes. |
| Load balancing | It distributes work (client requests and so on) across the members of a cluster. |
| Application failover | When an object in an application that is performing a task becomes unavailable, the object from the application in another server takes over to finish the job. |
| Availability | After a system failure on one server, it automatically continues ongoing work on another server. |
| Migration | After a system failure on one server, it continues ongoing work by moving the component to another server. |

Copyright © 2009, Oracle. All rights reserved.

Benefits of Clustering

An Oracle WebLogic Server cluster provides the following benefits:

- **Scalability:** The capacity of a cluster is not limited to one server or one machine. Servers can be added to the cluster dynamically to increase capacity. If more hardware is needed, a new server on a new machine can be added.
- **Load Balancing:** The distribution of jobs and associated communications across the computing and networking resources in your environment can be even or weighted, depending on your environment. Even distributions include round-robin and random algorithms.
- **Application Failover:** Distribution of applications and their objects on multiple servers enables easier failover of the session-enabled applications.
- **Availability:** A cluster uses the redundancy of multiple servers to insulate clients from failures. The same service can be provided on multiple servers in a cluster. If one server fails, another can take over. The capability to execute failover from a failed server to a functioning server increases the availability of the application to clients.
- **Migration:** This ensures uninterrupted availability of pinned services or components—those that must run only on a single server instance at any given time, such as the Java Transaction API (JTA) transaction recovery system, when the hosting server instance fails.

Understanding the technical infrastructure that enables clustering helps programmers and administrators to maximize the scalability and availability of their applications.

What Can Be Clustered

The following types of objects can be clustered:

- Servlets
- JSP
- EJB
- Remote Method Invocation (RMI) objects
- Java Messaging Service (JMS) destinations
- Java Database Connectivity (JDBC) connections



Copyright © 2009, Oracle. All rights reserved.

What Can Be Clustered

WebLogic Server provides clustering support for servlets and JSPs by replicating the HTTP session state of clients that access clustered servlets and JSPs. WebLogic Server can maintain HTTP session states in memory, a file system, or a database.

Load balancing and failover for EJBs and RMI objects are handled using replica-aware stubs, which can locate instances of the object throughout the cluster. Replica-aware stubs are created for EJBs and RMI objects as a result of the object compilation process. EJBs and RMI objects are deployed homogeneously to all the server instances in the cluster.

WebLogic Java Messaging Service (JMS) architecture implements clustering of multiple JMS servers by supporting clusterwide, transparent access to destinations from any WebLogic Server instance in the cluster.

WebLogic Server enables you to cluster JDBC objects, including data sources and multidata sources, to improve the availability of cluster-hosted applications. Each JDBC object that you configure for your cluster must exist on each managed server in the cluster—when you configure the JDBC objects, target them to the cluster.

Proxy Servers for HTTP Clusters

- Proxy servers are used to provide load balancing and failover for a cluster. They:
 - Are the client's first level of interaction with the cluster
 - Give the cluster its single server appearance
- A proxy server can be either software based or hardware based.
- A software-based proxy server may be a WebLogic servlet, Web server plug-in, or a third-party application.
- A hardware-based proxy server is typically a physical load balancer.

The red bar spans most of the width of the slide, centered horizontally.

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Proxy Servers for HTTP Clusters

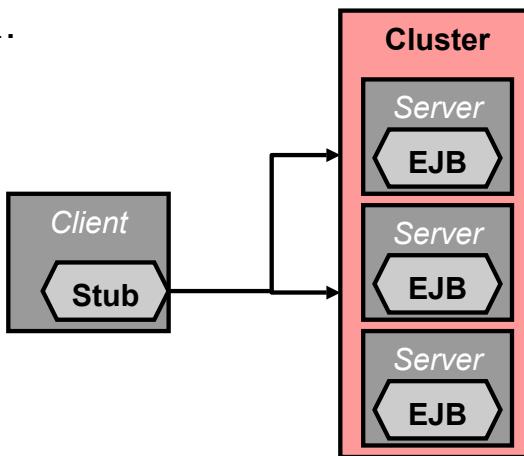
Proxies are how clients interact with the cluster, whether they are hardware or software based. You have three basic choices of proxy depending on your cluster architecture: `HTTPClusterServlet`, a Web server plug-in, or a physical load balancer (such as Local Director or F5 Networks Big IP). These proxy choices are generally available regardless of the architecture type, but some architectures might dictate the type of proxy that will be needed.

You can configure Oracle WebLogic Server clusters to operate alongside existing Web servers. In such an architecture, a bank of Web servers provides static HTTP content for the Web application, using a WebLogic proxy plug-in or `HttpClusterServlet` to direct servlet and JSP requests to a cluster.

There are two alternative proxy architectures: two-tier and multitier.

High Availability for EJBs

- WebLogic provides the EJB client applications with cluster-aware stubs that transparently perform load balancing and failover.
- You can enable and configure clustering for each EJB using the application deployment descriptor `weblogic-ejb-jar.xml`.



ORACLE

Copyright © 2009, Oracle. All rights reserved.

High Availability for EJBs

Failover for clustered EJBs is accomplished using the object's replica-aware stub. When a client makes a call through a replica-aware stub to a service that fails, the stub detects the failure and retries the call on another replica.

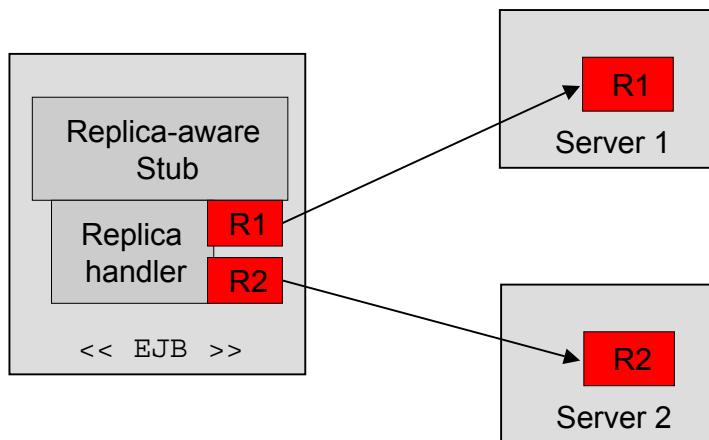
With clustered objects, automatic failover generally occurs only in cases where the object is idempotent. An object is idempotent if any method can be called multiple times with no different effect than calling the method once.

Method-level failover for a stateful service requires state replication. Oracle WebLogic Server satisfies this requirement by replicating the state of the primary bean instance to a secondary server instance, using a replication scheme similar to that used for HTTP session state.

Oracle WebLogic Server uses the round-robin algorithm as the default load-balancing strategy for clustered object stubs when no algorithm is specified. Weight-based load balancing improves on the round-robin algorithm by taking into account a preassigned weight for each server. Oracle WebLogic Server provides server affinity that can be used to turn off load balancing for external client connections. If an object is configured for server affinity, the client-side stub attempts to choose a server instance to which it is already connected and continues to use the same server instance for method calls.

Clustering EJB Objects: Replica-Aware Stub

- Failover and load-balancing of EJBs is done with replica-aware stubs.
- Replica-aware stubs are generated at compile time for clusterable EJBs.



ORACLE

Copyright © 2009, Oracle. All rights reserved.

Clustering EJB Objects: Replica-Aware Stub

Load balancing for clustered EJBs and RMIs is accomplished using the object's replica-aware stub. When a clusterable EJB is compiled, replica-aware stubs are generated for the bean. The replica-aware stub represents a collection of replicas and contains the logic required to locate an EJB or RMI class on any WebLogic Server instance on which the object is deployed.

When a client accesses a clustered object, the replica-aware stub is sent to the client. The stub contains the load-balancing algorithm to balance method calls to the object. On each call, the stub can employ its load algorithm to select a replica. This provides load balancing across the cluster in a way that is transparent to the caller. If a failure occurs during the call, the stub intercepts the exception and retries the call on another replica. This provides failover that is transparent to the caller.

With clustered objects, automatic failover typically occurs only if the object is idempotent. An object is idempotent if any method can be called multiple times with no other effect than calling the method once. This is always true for methods that have no permanent side effects. Methods that have side effects must be written with idempotence in mind.

EJB: Server Failure Situations

A replica-aware stub has to detect an invocation failure from the exceptions it receives:

- Application exception
- System exception
- Network or communication exception

These are not indicative of a critical failure, as your application handles them.

A network exception would occur if a server, container, or skeleton crashed.

Note: If a communication exception occurs, the stub does not know if the method started, was currently executing, or finished but was unable to return a response.

ORACLE

Copyright © 2009, Oracle. All rights reserved.

EJB: Server Failure Situations

Because a stub is Java code, it will be able to receive exceptions that are generated by the skeleton, EJB, or the RMI handler (at the network level). Because system and application exceptions are expected, they are not considered failure situations. Application and system exceptions are notifications of abnormalities on the server, but the server is still in a state where it can be used.

If a network or communication exception occurs, this means that the network TCP/IP socket communication with the server has failed. Even though this exception could be the result of a faulty network, it is unlikely. The stub assumes that the server, container, or skeleton has crashed and is temporarily unavailable. Unfortunately, when this scenario occurs, the stub cannot determine the status of the method invocation. The failure may have occurred before, during, or after the method invocation.

Load-Balancing Clustered EJB Objects

- WebLogic Server supports the following load-balancing algorithms for clustered EJB objects:
 - Round-robin
 - Weight-based
 - Random
 - Parameter-based routing (programmatic)
- Server affinity configuration enables calls to objects to remain with the same server and minimizes client-side load balancing.



Copyright © 2009, Oracle. All rights reserved.

Load-Balancing Clustered EJB Objects

The following algorithms are supported for clustered EJB objects:

- **Round-Robin (default):** The round-robin algorithm is the default load-balancing strategy for clustered object stubs when no algorithm is specified.
- **Weight-Based:** The weight-based algorithm takes into account a preassigned weight for each server. Each server in the cluster is assigned a weight in the range (1–100). For example, suppose that A is 4, B is 2, and C is 1, the usage will be ABCABAA....
- **Random:** This algorithm chooses the next replica at random. This tends to distribute calls evenly among the replicas. It is recommended only in a cluster where each server has the same power and hosts the same services. The advantages are that it is simple and relatively cheap. The primary disadvantage is that there is a small cost to generating a random number on every request, and there is a slight probability that the load will not be evenly balanced over a small number of runs.
- **Parameter-Based Routing:** It is also possible to have a finer grain of control over load-balancing and implemented in the application by the programmer.

Server affinity is accomplished by causing method calls on objects to “stick” to an existing connection, instead of being load-balanced among the available server instances. With server affinity algorithms, the load balancing is disabled only for external client connections.

Stateless Session Bean Failover

A replica-aware stub uses a selection process to implement fault tolerance.

1. Client calls a method on the stub.
2. The stub calls replica-handler to choose server-replica. Load balancing can occur here.
3. The stub calls a method on the replica, (which sends the method to the server).
 - If no exception occurs, the stub returns successfully.
 - If an application or system exception occurs, the stub propagates the exception to the client.
 - If a network or communication exception occurs, the stub calls the replica-handler to choose another replica *if* the method is marked as being idempotent.
 - If a network or communication exception occurs, the stub propagates the exception *if* the method is not marked as being idempotent.



Copyright © 2009, Oracle. All rights reserved.

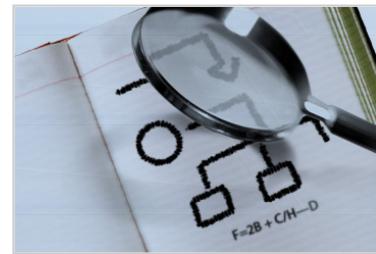
Stateless Session Bean Failover

The algorithm employed in the slide is used by a replica-aware stub on a stateless session bean to choose a replica. Failover should occur when a method that is invoked fails while it is being executed. Because a method can fail in various ways, the different failure situations that can occur need to be analyzed. Depending on the type of exception or failure that is encountered, a stub can react in different ways.

If no exception occurs, the stub will normally return. If a system or application exception is propagated over the wire, the stub does not react to that exception and propagates the exception back to the client. If a network or communication exception occurs, the stub will perform a high availability switch over to another replica if the method is marked as being idempotent. If the method is not marked as being idempotent, the stub just propagates the exception.

Road Map

- Oracle WebLogic cluster introduction
- Cluster architecture
 - Considerations for selecting an appropriate cluster architecture
 - Basic cluster architecture
 - Multitier cluster architecture
 - Proxy servers
- Cluster communication

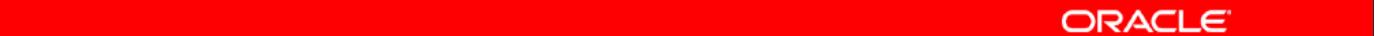


ORACLE®

Copyright © 2009, Oracle. All rights reserved.

Selecting a Cluster Architecture

- Consider the following factors when selecting a suitable architecture:
 - Performance
 - Efficient state persistence
 - Optimal load balancing
 - Effective failover
 - Reliable communication
- There are two primary cluster architectures to choose from:
 - Basic cluster architecture
 - Multitier architecture

The red bar spans most of the width of the slide, positioned above the copyright notice.

ORACLE®

Copyright © 2009, Oracle. All rights reserved.

Selecting a Cluster Architecture

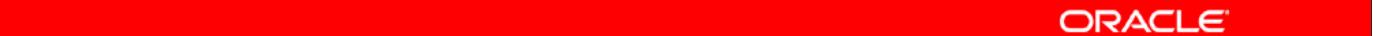
Although architecture is considered subjective and good architecture is usually a point of debate, there are some general considerations that should be addressed when selecting a cluster architecture:

- Performance
- Efficient state persistence (through replication or other means)
- Optimal load balancing
- Effective failover
- Reliable communication

The preceding factors ultimately decide the success or failure of your clustered services.

Cluster Architecture

- Applications are generally deployed in multiple tiers, each tier representing a distinct functionality:
 - Web tier
 - Presentation tier
 - Business or object tier
- WebLogic provides clustering support for all three tiers.
- Other services, such as JMS and JDBC, can take advantage of clusters. The load balancing and failover operations for these services are handled differently.

The red bar spans most of the width of the slide, centered horizontally.

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Cluster Architecture

Applications are usually broken into three functional tiers: Web tier, presentation tier, and object tier. In programming circles, these are also known as the model, view, and control. You tend to abstract them a little more when talking about clustering, but they are effectively the same.

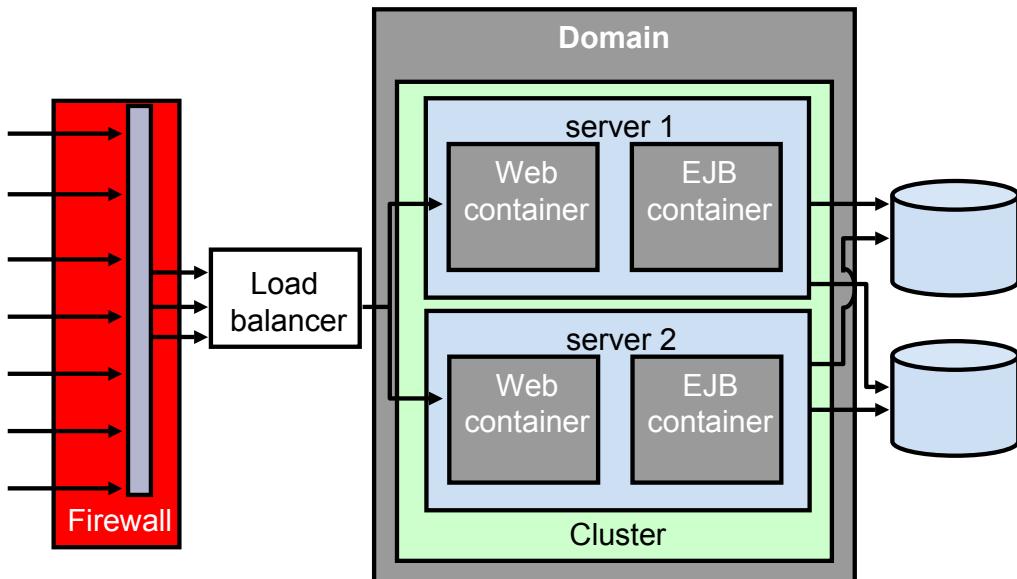
The Web tier provides the static, idempotent presentation of a Web application and is generally the first piece that clients come in contact with. Often, the Web tier is handled by a Web server, such as Oracle HTTP Server, Apache, Internet Information Server (IIS), or Netscape Enterprise Server (NES).

The presentation tier provides the dynamic content, such as servlets, JSP, and so forth. This tier also acts as a consumer to the business logic represented in the business tier. The presentation tier typically contains implemented design patterns or run-time frameworks that allow the client to interact with the business tier and generate a dynamic view of that tier per request or session. The presentation tier is handled by WebLogic and is accessed via direct or indirect client requests to the presentation tier elements.

The business tier provides access to business logic, middleware, and integrated systems. Typically, these are handled by various types of EJBs or server services, such as JMS and JDBC. WebLogic also handles this tier, but there are other applications, services, and servers that participate at this level.

Basic Cluster Architecture

A basic cluster architecture combines static HTTP, presentation logic, business logic, and objects into one cluster.



ORACLE

Copyright © 2009, Oracle. All rights reserved.

Basic Cluster Architecture

The basic recommended cluster architecture combines all Web application tiers and puts the related services (static HTTP, presentation logic, and objects) into one cluster.

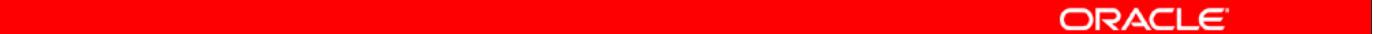
The basic architecture has the following advantages:

- **Easy administration:** Because one cluster hosts static HTTP pages, servlets, and EJBs, you can configure the entire Web application and deploy or undeploy objects using one Administration Console. You do not need to maintain a separate bank of Web servers (and configure Oracle WebLogic Server proxy plug-ins) to benefit from clustered servlets.
- **Flexible load balancing:** Using load-balancing hardware directly, in front of the Oracle WebLogic Server cluster, enables you to use advanced load-balancing policies for access to both HTML and servlet content.
- **Robust security:** Putting a firewall in front of your load-balancing hardware enables you to set up a demilitarized zone (DMZ) for your Web application using minimal firewall policies.
- **Optimal performance:** The combined-tier architecture offers the best performance for applications in which most or all the servlets or JSPs in the presentation tier typically access objects in the object tier, such as EJBs or JDBC objects.

A DMZ is a logical collection of hardware and services that is made available to outside, untrusted sources.

Basic Cluster Architecture: Advantages and Disadvantages

- Advantages:
 - Easy administration
 - Flexible load balancing
 - Robust security
- Disadvantages:
 - It cannot load-balance EJB method calls.
 - Load-balancing across the tiers may become unbalanced.

The red horizontal bar spans most of the width of the slide, positioned just above the footer area.

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Basic Cluster Architecture: Advantages and Disadvantages

Load balancing and failover can be introduced only at the interfaces between Web application tiers. So, when tiers are deployed to a single cluster, you can load-balance only between clients and the cluster. Because most load balancing and failover occur between clients and the cluster itself, a combined-tier architecture meets the needs of most Web applications.

However, such basic clusters provide no opportunity for load-balancing method calls to clustered EJBs. Because clustered objects are deployed on all Oracle WebLogic Server instances in the cluster, each object instance is available locally. Oracle WebLogic Server optimizes method calls to clustered EJBs by always selecting the local object instance, rather than distributing requests to remote objects.

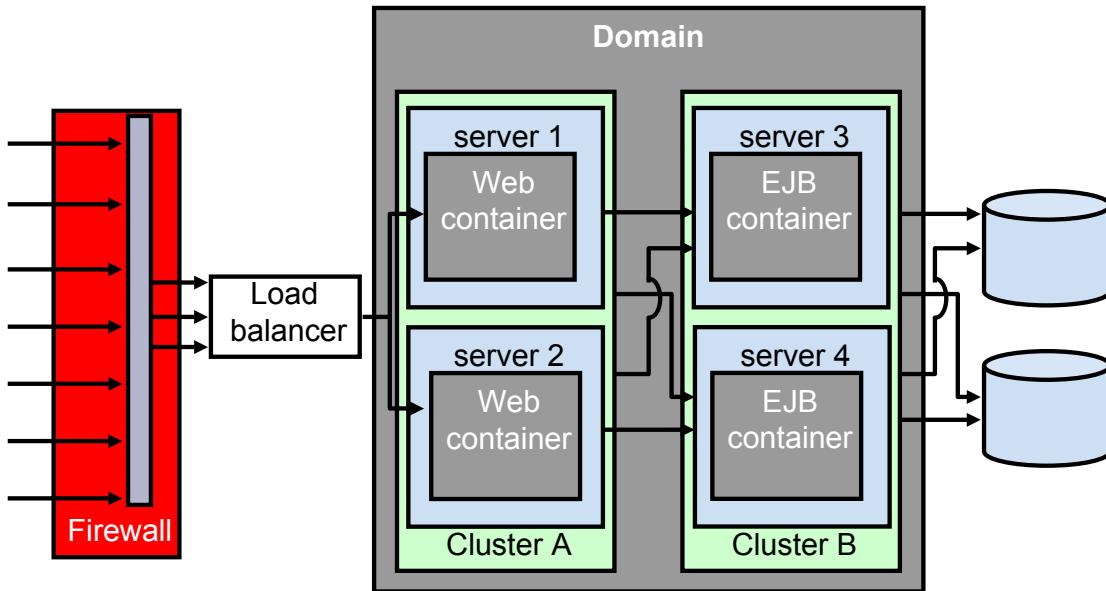
If the processing load on individual servers becomes unbalanced, it may eventually become more efficient to submit method calls to remote objects rather than process methods locally.

To use load balancing for method calls to clustered EJBs, you must split the presentation and object tiers of the Web application onto separate physical clusters, thereby ensuring that all the object calls are remote calls and the load is balanced.

Consider the frequency of invocations of the object tier by the presentation tier when you decide between a combined-tier and a multtier architecture. If presentation objects usually invoke the object tier, a combined-tier architecture may offer better performance than a multtier architecture.

Multitier Cluster Architecture

The Web tier and the business logic with services can be separated into two clusters.



ORACLE

Copyright © 2009, Oracle. All rights reserved.

Multitier Cluster Architecture

In the architecture illustrated in the slide, two separate Oracle WebLogic Server clusters are configured:

- Cluster A to serve static HTTP content and clustered servlets
- Cluster B to serve clustered EJBs

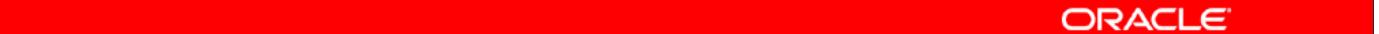
Multitier architecture is recommended for applications that require:

- Load balancing for method calls to clustered EJBs
- Flexibility for load balancing between servers that provide HTTP content and servers that provide clustered objects
- Higher availability (fewer single points of failure)
- More flexible security

Note: Consider the frequency of invocations from the presentation tier to the object tier when considering a multitier architecture. If presentation objects usually invoke the object tier, a combined-tier architecture may offer better performance than a multitier architecture.

Multitier: Advantages and Disadvantages

- Advantages:
 - Improved load balancing
 - Load balancing of EJB methods
 - Higher availability
 - Improved security options
- Disadvantages:
 - Can create a bottleneck when the presentation tier makes frequent calls to the business logic
 - Can lead to increased licensing cost
 - Can lead to added firewall configuration complexity

The red horizontal bar spans most of the width of the slide, positioned just above the copyright notice.

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Multitier: Advantages and Disadvantages

The multitier architecture provides the following advantages:

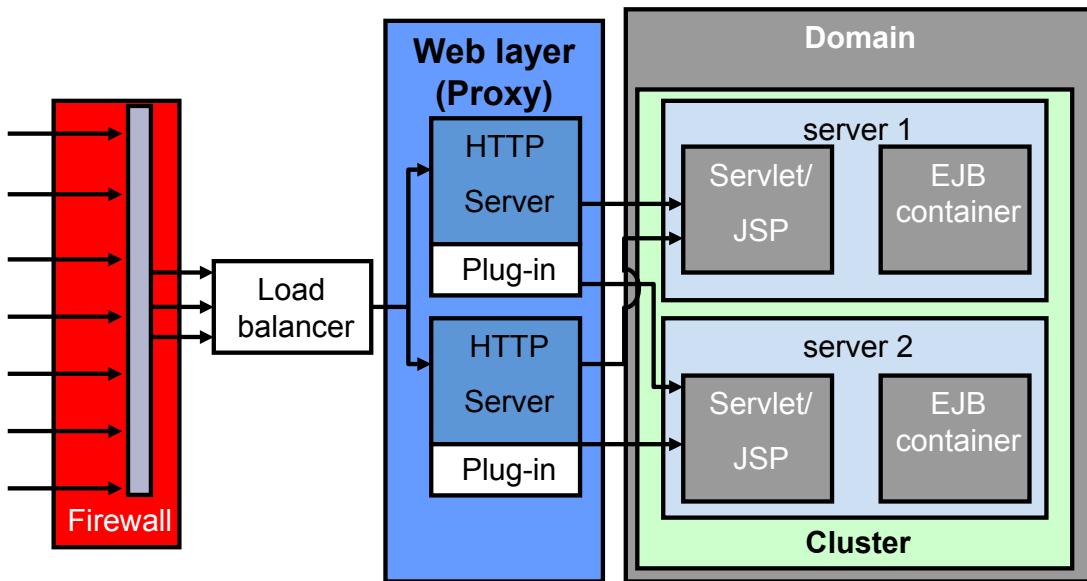
- **Load-balancing EJB methods:** By hosting servlets and EJBs on separate clusters, the servlet-method calls to the EJBs can be load-balanced across multiple servers.
- **Improved server load balancing:** Separating the presentation and object tiers onto separate clusters provides you with more options for distributing the load of the Web application. For example, if the application accesses HTTP and servlet content more often than EJB content, you can use a large number of Oracle WebLogic Server instances in the presentation tier cluster to concentrate access to a smaller number of servers that host the EJBs. For example, if your Web clients make heavy use of servlets and JSPs but access a relatively small set of clustered objects, the multitier architecture enables you to concentrate the load of servlets and EJB objects appropriately. You may configure a servlet cluster of 10 Oracle WebLogic Server instances and an object cluster of three managed servers, while still fully using each server's processing power.
- **Higher availability:** By using additional Oracle WebLogic Server instances, the multitier architecture has fewer points of failure than the basic cluster architecture. For example, if an Oracle WebLogic Server that hosts the EJBs fails, the HTTP- and servlet-hosting capacity of the Web application is not affected.

Multitier: Advantages and Disadvantages (continued)

- **Improved security options:** By separating the presentation and object tiers onto separate clusters, you can use a firewall policy that places only the servlet/JSP cluster in the DMZ. Servers that host the clustered objects can be further protected by denying direct access from untrusted clients.
- **Limitations of Multitier Architectures**
 - **No Collocation Optimization:** The multitier architecture cannot optimize object calls by using the collocation strategy. So, the Web application may incur network overhead for all method calls to the clustered objects.
 - **Firewall Restrictions:** If you place a firewall between the servlet cluster and an object cluster in a multitier architecture, you must bind all the servers in the object cluster to public DNS names, rather than IP addresses. Binding those servers with IP addresses can cause address translation problems and prevent the servlet cluster from accessing individual server instances.

Basic Cluster Proxy Architecture

This is similar to the basic cluster architecture, except that static content is hosted on HTTP servers.



ORACLE

Copyright © 2009, Oracle. All rights reserved.

Basic Cluster Proxy Architecture

The two-tier proxy architecture contains two physical layers of hardware and software.

Web Layer

The proxy architecture uses a layer of hardware and software that is dedicated to the task of providing the application's Web tier. This physical Web layer can consist of one or more identically configured machines that host one of the following application combinations:

- Oracle WebLogic Server with `HttpClusterServlet`
- Apache with the Oracle WebLogic Server Apache proxy plug-in
- Netscape Enterprise Server with the Oracle WebLogic Server NSAPI proxy plug-in
- Microsoft Internet Information Server with the Oracle WebLogic Server Microsoft-IIS proxy plug-in

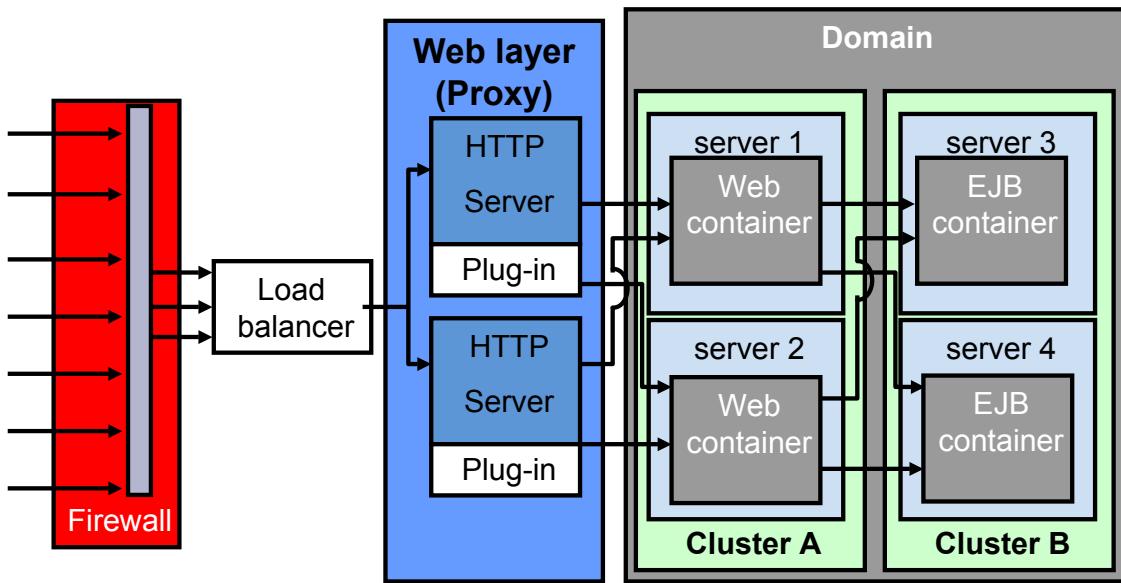
Regardless of which Web server software you select, remember that the physical tier of the Web servers should provide only static Web pages. Dynamic content—servlets and JSPs—are proxied via the proxy plug-in or `HttpClusterServlet` to an Oracle WebLogic Server cluster that hosts servlets and JSPs for the presentation tier.

Servlet/Object Layer

The recommended two-tier proxy architecture hosts the presentation and object tiers on a cluster of Oracle WebLogic Server instances. This cluster can be deployed either on a single machine or on multiple separate machines. The Servlet/Object layer differs from the combined-tier cluster in that it does not provide static HTTP content to application clients.

Multitier Cluster Proxy Architecture

This is similar to the multitier cluster architecture, except that static content is hosted on HTTP servers.



ORACLE

Copyright © 2009, Oracle. All rights reserved.

Multitier Cluster Proxy Architecture

You can also use a bank of Web servers as the front end to a pair of Oracle WebLogic Server clusters that host the presentation and object tiers.

Using stand-alone Web servers and proxy plug-ins provides the following advantages:

- You can use existing hardware.
- If you already have a Web application architecture that provides static HTTP content to clients, you can easily integrate the existing Web servers with one or more Oracle WebLogic Server clusters to provide dynamic HTTP and clustered objects.
- You can use familiar firewall policies.

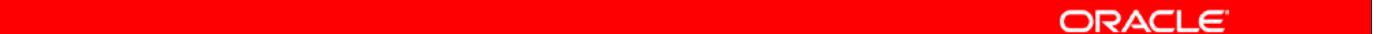
Using a Web server proxy at the front end of your Web application enables you to use familiar firewall policies to define your DMZ. In general, you can continue placing the Web servers in your DMZ while disallowing direct connections to the remaining Oracle WebLogic Server clusters in the architecture. The diagram in the slide depicts this DMZ policy.

However, there are some disadvantages:

- Additional administration
- Limited load balancing options

Proxy Web Server Plug-In Versus Load Balancer

- There are many advantages to using a physical load balancer instead of the proxy plug-in:
 - There is no need to configure the client plug-ins.
 - It eliminates the proxy layer, thereby reducing the number of connections.
 - There are more sophisticated load-balancing algorithms.
- There are a number of disadvantages as well:
 - Additional administration
 - Explicit configuration of “sticky” sessions for stateful Web applications

The red horizontal bar spans most of the width of the slide, positioned just above the copyright notice.

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Proxy Web Server Plug-In Versus Load Balancer

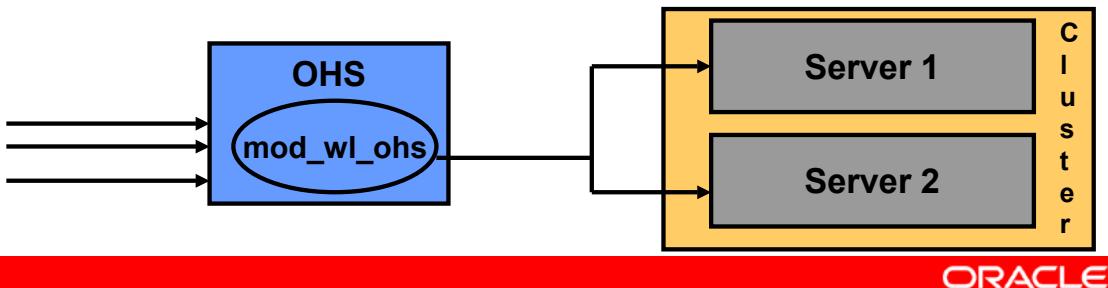
Using a load balancer directly with an Oracle WebLogic Server cluster provides several benefits over proxying servlet requests. First, using Oracle WebLogic Server with a load balancer requires no additional administration for client setup—you do not need to set up and maintain a separate layer of HTTP servers and you do not need to install and configure one or more proxy plug-ins. Removing the Web proxy layer also reduces the number of network connections that are required to access the cluster.

Using a load-balancing hardware provides more flexibility for defining the load-balancing algorithms that suit the capabilities of your system. You can use any load-balancing strategy (for example, load-based policies) that your load-balancing hardware supports. With proxy plug-ins, you are limited to a simple round-robin algorithm for clustered servlet requests.

Note, however, that using a third-party load balancer may require additional configuration if you use in-memory session state replication. In this case, you must ensure that the load balancer maintains a “sticky” connection between the client and its point-of-contact server, so that the client accesses the primary session state information. When using proxy plug-ins, no special configuration is necessary because the proxy automatically maintains a sticky connection.

Proxy Plug-Ins

- Proxy plug-ins:
 - Delegate dynamic content requests to WLS servers and balance load across a cluster in a round-robin fashion
 - Route HTTP requests to back-end WLS instances based on session cookie or URL rewriting
 - Avoid routing to failed servers in the cluster
- Oracle HTTP Server contains `mod_wl_ohs`, which is a plug-in for WLS by default.
- WLS provides plug-ins to other major Web servers as well.



Copyright © 2009, Oracle. All rights reserved.

Proxy Plug-Ins

A proxy plug-in may be essential in an environment where Oracle HTTP Server or other Web servers serve static pages, and an Oracle WebLogic Server (possibly on a different host) is delegated to serve dynamic pages (such as JSPs or pages generated by HTTP servlets). To the end user (the browser), the HTTP responses still appear to come from the same source—the Web server running the plug-in. Oracle WebLogic Server on the back end is invisible. The HTTP-tunneling facility of the WebLogic client/server protocol can operate through the plug-in, providing access to all Oracle WebLogic Server services (not just dynamic pages).

Oracle WebLogic Server plug-ins provide efficient performance by reusing connections from the plug-in to Oracle WebLogic Server. The plug-in maintains “keep-alive” connections between the plug-in and Oracle WebLogic Server.

For documentation on plug-ins, see *Oracle Fusion Middleware Using Web Server Plug-Ins with Oracle WebLogic Server 11g Release 1 (10.3.1)*.

OHS as Proxy Web Server

Oracle HTTP Server (OHS) is a Web server that:

- Is based on Apache
- Serves static and dynamic content
- Supports content generation in many languages, such as Java, C, C++, PHP, PERL, or PL/SQL
- Contains a WebLogic Server plug-in (`mod_wl_ohs`) by default
- Can be easily integrated with other Oracle Fusion Middleware components
- Can be managed using the Fusion Middleware Control along with other components



Copyright © 2009, Oracle. All rights reserved.

OHS as Proxy Web Server

Oracle HTTP Server is based on the Apache Web server. It serves both static and dynamic content and supports applications developed in Java, PL/SQL, C, C++, PHP, or PERL. OHS supports single sign-on, clustered deployment and high availability, and Web Cache. In addition, plug-ins that are available as separate components enable integration with non-Oracle HTTP Servers.

A `mod_wl_ohs` module is available in OHS and enables you to integrate your WebLogic Server environment with OHS immediately after the configuration of the OHS instance and the domains.

Request Flow When Using OHS

- The client sends an HTTP request to OHS for access to a Java EE application.
- The `mod_wl_ohs` plug-in at OHS receives the request and determines from the cookie (in request) which WLS server should serve the request.
- If no cookie exists, the request is assigned to the next available WLS server in the cluster (round-robin algorithm).
- The WLS server that responds places the appropriate cookie in the response.
- OHS routes the response to the client (with the cookie).



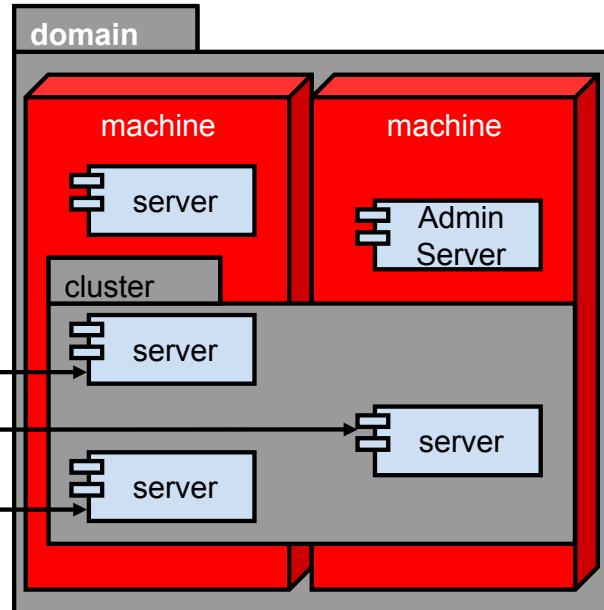
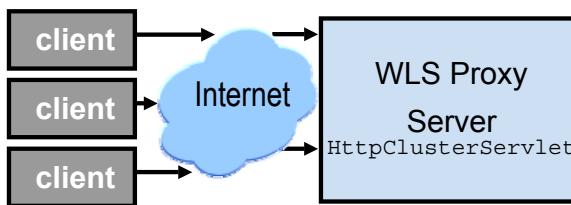
ORACLE

Copyright © 2009, Oracle. All rights reserved.

WLS HttpServlet

HttpClusterServlet:

- Is deployed in the default Web application of the proxy server
- Delivers client requests in round-robin style to servers in the cluster



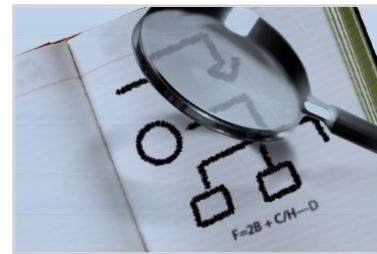
Copyright © 2009, Oracle. All rights reserved.

WLS HttpServlet

HttpClusterServlet proxies the requests from an Oracle WebLogic Server to other Oracle WebLogic Server instances within a cluster. HttpClusterServlet provides load balancing and failover for the proxied HTTP requests.

Road Map

- Oracle WebLogic cluster introduction
- Cluster architecture
- Cluster communication
 - Server communication in a cluster
 - Detecting a failure
 - Multitier communication

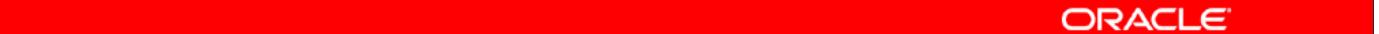


ORACLE®

Copyright © 2009, Oracle. All rights reserved.

Server Communication in a Cluster

- WebLogic Server instances in a cluster communicate with one another using:
 - IP sockets, which are the conduits for peer-to-peer communication between clustered server instances
 - IP unicast or multicast, which server instances use to broadcast availability of services and heartbeats that indicate continued availability
- Multicast broadcasts one-to-many communications among clustered instances.
- Unicast is an alternative to multicast to handle cluster messaging and communications. The unicast configuration is much easier because it does not require cross-network configuration that multicast requires.

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Server Communication in a Cluster

Peer-to-peer communications between server instances in a cluster use IP sockets. IP sockets provide a simple, high-performance mechanism for transferring messages and data between two applications.

WebLogic Server uses IP multicast for all one-to-many communications among server instances in a cluster. This communication includes:

- **Clusterwide JNDI updates:** Each WebLogic Server instance in a cluster uses multicast to announce the availability of clustered objects that are deployed or removed locally. Each server instance in the cluster monitors these announcements and updates its local JNDI tree to reflect current deployments of clustered objects. For more details, see the section titled “Clusterwide JNDI Naming Service” later in this lesson.
- **Cluster heartbeats:** Each WebLogic Server instance in a cluster uses multicast to broadcast regular “heartbeat” messages that advertise its availability. By monitoring heartbeat messages, server instances in a cluster determine when a server instance has failed. (Clustered server instances also monitor IP sockets as a more immediate method of determining when a server instance has failed.)

IP multicast is a broadcast technology that enables multiple applications to subscribe to an IP address and port number and listen for messages. A multicast address is an IP address in the range 224.0.0.0–239.255.255.255.

Server Communication in a Cluster (continued)

WebLogic Server provides an alternative to using multicast to handle cluster messaging and communications. Unicast configuration is much easier because it does not require cross-network configuration that multicast requires. Additionally, it reduces potential network errors that can occur from multicast address conflicts.

When creating a new cluster, it is recommended that you use unicast for messaging within a cluster.

One-to-Many Communications

- Oracle WebLogic Server uses one-to-many communication for:
 - Clusterwide JNDI updates
 - Cluster “heartbeats”
- Because all one-to-many communications occur over IP multicast, when you design a cluster, consider the following factors:
 - If your cluster spans multiple subnets, your network must be configured to reliably transmit messages.
 - A firewall can break IP multicast transmissions.
 - The multicast address should not be shared with other applications.
 - Multicast storms may occur.



Copyright © 2009, Oracle. All rights reserved.

One-to-Many Communications

Oracle WebLogic Server uses multicast to broadcast regular “heartbeat” messages that advertise the availability of individual server instances in a cluster. The servers in a cluster listen to heartbeat messages to determine when a server has failed. (Clustered servers also monitor IP sockets as a more immediate method of determining when a server has failed.)

All servers use multicast to announce the availability of clustered objects that are deployed or removed locally. Servers monitor the announcements so that they can update their local JNDI tree to indicate the current deployments of clustered objects.

Because multicast controls the critical functions related to detecting failures and maintaining the clusterwide JNDI tree, it is important that neither the cluster architecture nor the network topology interfere with multicast communications.

If server instances in a cluster do not process incoming messages on a timely basis, increased network traffic and heartbeat retransmissions can result. The repeated transmission of multicast packets on a network is referred to as a multicast storm, and can stress the network and attached stations, potentially causing end-stations to hang or fail. Increasing the size of the multicast buffers can improve the rate at which announcements are transmitted and received, and prevent multicast storms.

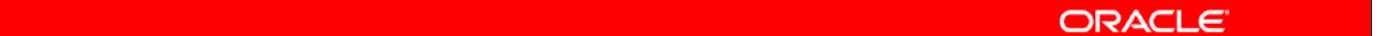
One-to-Many Communications (continued)

Therefore, you should keep the following in mind.

- You may want to distribute a WebLogic Server cluster across multiple subnets in a Wide Area Network (WAN) to increase redundancy, or to distribute clustered server instances over a larger geographical area.
- Firewalls can break multicast transmissions. Although it might be possible to tunnel multicast transmissions through a firewall, this practice is not recommended for Oracle WebLogic Server clusters. Each Oracle WebLogic Server cluster should be treated as a logical unit that provides one or more distinct services to the client. Such a logical unit should not be split between security zones.
- Using the same multicast address in other applications will cause the server instances to process unnecessary messages.
- If the server instances do not process incoming messages in a timely manner, repeated transmissions on a network can cause a multicast storm.

Considerations When Using Unicast

- Unicast messaging type:
 - Is much easier to configure because it does not require cross-network configuration that multicast requires
 - Reduces potential network errors that can occur from multicast address conflicts
- You cannot mix and match cluster messaging types within a cluster.

The red bar spans most of the width of the slide, centered horizontally.

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Considerations When Using Unicast

The following considerations apply when using unicast to handle cluster communications:

- All members of a cluster must use the same message type. Mixing between multicast and unicast messaging is not allowed.
- You must use multicast if you need to support a previous version of WebLogic Server within your cluster.
- Individual cluster members cannot override the cluster messaging type.
- The entire cluster must be shut down and restarted to change the messaging type.
- JMS topics configured for multicasting can access WebLogic clusters configured for unicast because a JMS topic publishes messages on its own multicast address that is independent of the cluster address. However, the following considerations apply:
 - The router hardware configurations that allow unicast clusters may not allow JMS multicast subscribers to work.
 - JMS multicast subscribers need to be in a network hardware configuration that allows multicast accessibility.
 - For more details, see “Create and Configure Clusters” in *Programming WebLogic JMS*.

Peer-to-Peer Communications

Oracle WebLogic Server uses peer-to-peer communications for:

- Accessing nonclustered or pinned objects that reside on a remote server instance in the cluster
- Replicating HTTP session states and stateful session EJB states between a primary and a secondary server
- Accessing the clustered objects that reside on a remote server instance (typically, in a multitier cluster architecture)



Copyright © 2009, Oracle. All rights reserved.

Peer-to-Peer Communications

Proper socket configuration is crucial to the performance of an Oracle WebLogic Server cluster. Two factors determine the efficiency of socket communications in Oracle WebLogic Server:

- Whether the server's host system uses a native or a pure-Java socket reader implementation
- For systems that use pure-Java socket readers, whether or not the server is configured to use enough socket reader threads

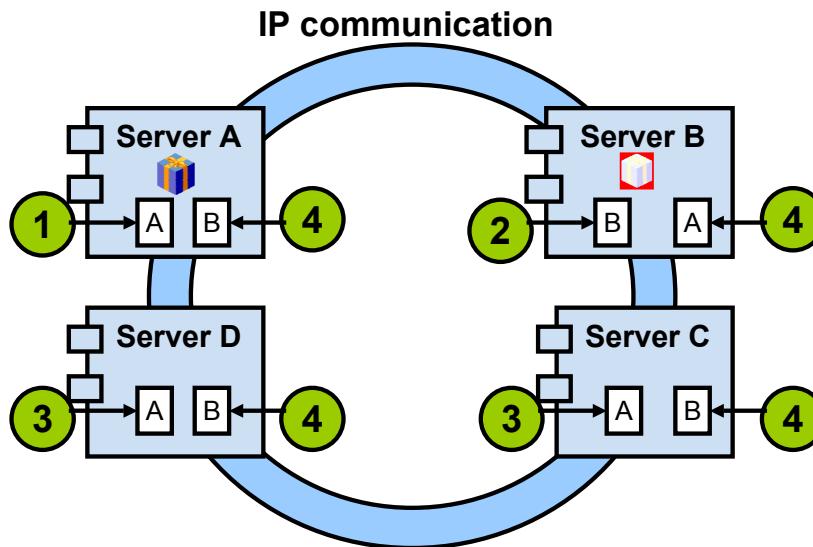
IP sockets provide a simple, high-performance mechanism for transferring messages and data between two applications. Clustered Oracle WebLogic Server instances use IP sockets for the following:

- Accessing nonclustered objects that are deployed to another clustered server instance on a different machine
- Replicating HTTP session states and stateful session EJB states between a primary and secondary server instance
- Accessing clustered objects that reside on a remote server instance (This generally occurs only in a multitier cluster architecture.)

Note: The use of IP sockets in Oracle WebLogic Server actually extends beyond the cluster scenario—all RMI communication takes place using sockets (for example, when a remote Java application accesses a remote object).

Clusterwide JNDI Naming Service

Each WebLogic Server in a cluster builds and maintains its own local copy of the clusterwide JNDI tree, which lists the services offered by all members of the cluster.



ORACLE

Copyright © 2009, Oracle. All rights reserved.

Clusterwide JNDI Naming Service

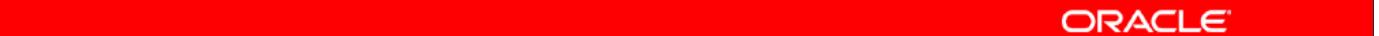
Clients access objects and services by using a JNDI-compliant naming service. Server instances in a cluster use a clusterwide JNDI tree. A clusterwide JNDI tree contains a list of locally available services and the services offered by clustered objects from other servers in the cluster.

Each WebLogic Server in a cluster builds and maintains its own local copy of the clusterwide JNDI tree. As a server instance boots or as new services are dynamically deployed to a running server instance, the server instance first binds the implementations of those services to the local JNDI tree. The slide shows the following steps of clusterwide JNDI tree formation:

1. Server A has successfully bound an implementation of a clustered object into its local JNDI tree. Because the object is clustered, it offers this service to all other members of the cluster.
2. Server B initiates binding an implementation of the object into its local JNDI. If the server instance already has a binding for the cluster-aware service, it updates its local JNDI tree to indicate that a replica of the service is also available on Server A.
3. Other server instances in the cluster listening to the multicast or unicast address note that Server A offers a new service for the clustered object. These server instances update their local JNDI trees to include the new service.
4. After Server B completes binding, it notifies other servers. Also, it updates its own JNDI to note that Server A offers the service for the object. Subsequently, other servers (C and D) in the cluster also get the object and create the binding.

Name Conflicts and Resolution

- Cluster-level JNDI conflicts may occur when new services are added to the cluster.
- In case of name conflicts, local binding may succeed, but the binding of other object names from other servers will fail.
- To avoid cluster-level JNDI conflicts, you must deploy all replica-aware objects to all WebLogic Server instances in a cluster.

The red bar spans most of the width of the slide, centered horizontally.

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Name Conflicts and Resolution

Cluster-level JNDI conflicts may occur when new services are advertised over multicast or unicast. For example, if you deploy a pinned RMI object on one server instance in the cluster, you cannot deploy a replica-aware version of the same object on another server instance.

If two server instances in a cluster attempt to bind objects using the same name, local binding may succeed. However, the server instances with conflicting names will refuse to bind the server instances' replica-aware stub in to the JNDI tree. A conflict of this type would remain until one of the two server instances was shut down or until the clustered object is undeployed from all servers.

To avoid name conflicts, deploy all cluster-level objects to all members of the cluster. Also, avoid deploying clustered and non-clustered objects in a server.

Quiz

Which of the following is a benefit of multitier cluster architecture?

1. Requires fewer servers compared to the basic architecture
2. Possibility to load-balance method calls to clustered EJBs
3. Easier security implementation
4. None



Copyright © 2009, Oracle. All rights reserved.

Answer: 2

With multitier architecture, you can balance load on EJBs clustered across multiple servers.

Quiz

In a multitier cluster architecture where you want to load-balance EJB objects, you configure them:

1. Within one cluster
2. In different clusters
3. Along with the Web-tier clients in the same server
4. In different domains



Copyright © 2009, Oracle. All rights reserved.

Answer: 1

Load balancing in Oracle WebLogic Server works within a cluster. You cannot load balance across multiple clusters or domains. Because you intend to use multitier cluster, the Web server and EJB objects need to be separated. So options 2, 3, and 4 are not applicable in this case.

Summary

In this lesson, you should have learned about:

- Benefits of the Oracle WebLogic cluster
- Basic cluster architecture
- Multitier cluster architecture
- Communication among clustered server instances
- Key criteria for selecting a suitable cluster architecture



Copyright © 2009, Oracle. All rights reserved.

16

Configuring a Cluster

ORACLE®

Copyright © 2009, Oracle. All rights reserved.

Objectives

After completing this lesson, you should be able to do the following:

- Prepare your environment for a cluster
- Create and configure a cluster
- Add servers to a cluster
- Start up and shut down clustered servers



Copyright © 2009, Oracle. All rights reserved.

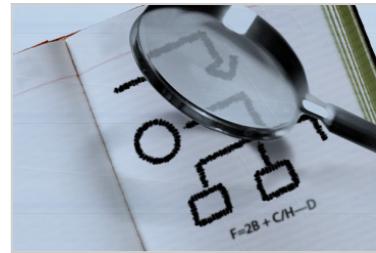
Objectives

Scenario

The Medical Records department has decided to implement and evaluate clustering on a test application to better understand the clustering functionality. Before implementing a cluster, you need to configure the Oracle HTTP Server as the Web tier front end for your applications. You create a basic cluster using MedRecSvr2 and MedRecSvr3 managed servers. Later, you deploy and configure the test application so that HTTP session replication is enabled.

Road Map

- Preparing for a cluster
 - Cluster architecture
 - Network and security topology
 - Machines
 - Names and addresses
- Configuring a cluster



ORACLE®

Copyright © 2009, Oracle. All rights reserved.

Preparing Your Environment

Before you configure a cluster, you need to prepare your environment.

- Determine your cluster architecture.
- Understand your network and security topologies.
- Choose the machines for the cluster installation.
- Identify IP addresses or DNS names, and port numbers for the server instances in the cluster.
- For proxy architectures, you could have:
 - A single firewall between untrusted clients and the Web server layer
 - A firewall between the proxy layer and the cluster
- Configure the Node Manager



Copyright © 2009, Oracle. All rights reserved.

Preparing Your Environment

The architecture that you choose affects how you set up your cluster. The cluster architecture may also require that you install or configure other resources, such as load balancers, HTTP servers, and proxy plug-ins.

Depending on the network topology that you choose, your security requirements will change.

- Some network topologies can interfere with multicast communications.
- Avoid deploying server instances in a cluster across a firewall.

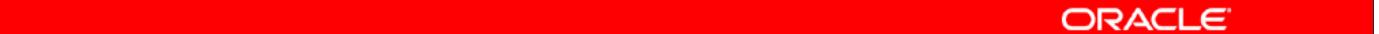
A single firewall between untrusted clients and the Web server layer can be used with both the basic cluster architecture and the multitier cluster architecture. This creates a demilitarized zone around the Web servers.

A firewall between the proxy layer and the cluster means that you need to bind the clustered server instances to publicly listed DNS names. If the internal and external DNS names are not identical, you need to configure the `ExternalDNSName` property for each server instance.

The Node Manager is useful for starting a managed server that resides on a different machine than its administration server. The Node Manager also provides features that help increase the availability of managed servers in your cluster.

Hardware

- You can set up a cluster on a single computer for demonstration or development.
 - This is not practical for production environments.
- Each computer involved in a cluster should have a static IP address.
- There is no built-in limit for the number of server instances in a cluster.
 - Large multiprocessor servers can host clusters with numerous servers.
 - The recommendation is one server instance for every two CPUs.

The red bar spans most of the width of the slide, centered horizontally.

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Hardware

The main benefits of a cluster are load balancing and failover. If multiple servers in a cluster are on the same computer, these benefits are minimized. If the computer fails, all the servers on it fail and, although you may be load balancing, it is still only the computer that handles the processing.

Load balancers and proxy servers need to know which servers are in a cluster. So, in general, you need to configure the IP address of each server in a cluster in the load balancer or proxy server. If the servers are assigned to a machine with a dynamically assigned IP address, the IP address can change, and the load balancer or proxy server would not be able to find it. So ensure that you configure the cluster on machines that have static IP addresses.

IP Addresses and Host Names

- The IP address and host name information is needed for configuring and managing:
 - The administration server
 - Managed servers
 - Multicast communication
- For a production environment, use the host name resolved at DNS rather than IP addresses.
 - Firewalls can cause IP address translation errors.
- Each server should have a unique name.
- The multicast address should not be used for anything other than cluster communications.

The red bar spans most of the width of the slide, centered horizontally.

ORACLE

Copyright © 2009, Oracle. All rights reserved.

IP Addresses and Host Names

If the internal and external DNS names of an Oracle WebLogic Server instance are not identical, use the `ExternalDNSName` attribute for the server instance to define the server's external DNS name. Outside the firewall, `ExternalDNSName` should translate to the external IP address of the server.

If clients access Oracle WebLogic Server over the default channel and T3, do not set the `ExternalDNSName` attribute, even if the internal and external DNS names of an Oracle WebLogic Server instance are not identical to avoid unnecessary DNS lookups.

Cluster Address

- The cluster address is used to communicate with entity and session beans by constructing the host name portion of the request URLs.
- You can explicitly define the address of a cluster.
 - The cluster address should be a DNS name that maps to the IP addresses or DNS names of each Oracle WebLogic Server instance in the cluster.
- You can also have Oracle WebLogic Server dynamically generate an address for each new request.
 - Minimizes configuration
 - Ensures an accurate cluster address
- The dynamic cluster address is created in the form of:

```
listenaddress1:listenport1,listenaddress2:liste  
nport2,listenaddress3:listenport3
```

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Cluster Address

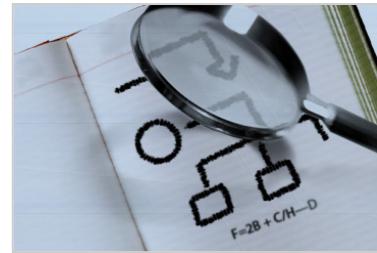
Each ListenAddress:ListenPort combination in the cluster address corresponds to the managed server and network channel that received the request. The order in which the ListenAddress:ListenPort combinations appear in the cluster address is random; the order varies from request to request.

The cluster address forms a portion of the URL that a client uses to connect to the cluster. The cluster address is used for generating EJB handles and entity EJB failover addresses. (This address may be either a DNS host name that maps to multiple IP addresses or a comma-separated list of single address host names or IP addresses.)

If network channels are configured, it is possible to set the cluster address on a per-channel basis.

Road Map

- Preparing for a cluster
- Configuring a cluster
 - Administration Console
 - Configuration Wizard
 - WLST
 - Ant



ORACLE®

Copyright © 2009, Oracle. All rights reserved.

Methods of Configuring Clusters

There are multiple ways to create and configure an Oracle WebLogic Server cluster:

- Configuration Wizard
- Administration Console
- WebLogic Scripting Tool (WLST)
- Java Management Extensions (JMX)
- WebLogic Server API



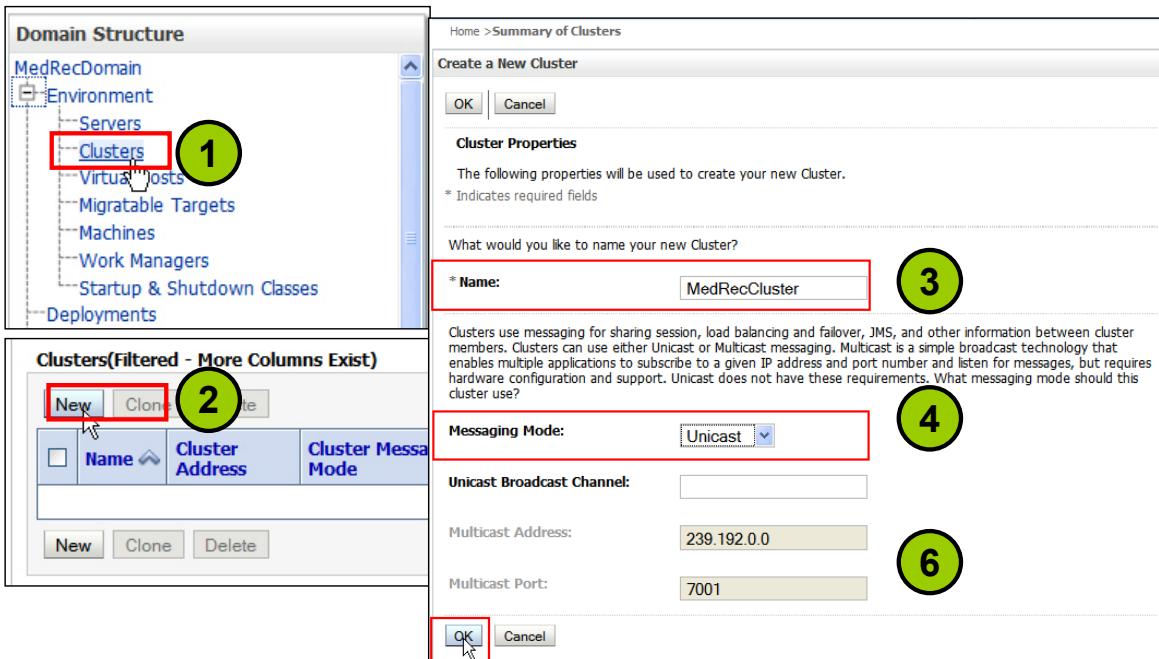
Copyright © 2009, Oracle. All rights reserved.

Methods of Configuring Clusters

You can use different methods to configure a cluster.

- **Configuration Wizard:** The Configuration Wizard is the recommended tool for creating a new domain with the cluster.
- **WebLogic Server Administration Console:** If you have an operational domain within which you want to configure a cluster, you can use the Administration Console.
- **WebLogic Scripting Tool (WLST):** You can use the WebLogic Scripting Tool (WLST) in a command-line scripting interface to monitor and manage clusters.
- **Java Management Extensions (JMX):** WebLogic Server provides a set of MBeans that you can use to configure, monitor, and manage WebLogic Server resources through JMX.
- **WebLogic Server Application Programming Interface (API):** You can write a program to modify the configuration attributes, based on the configuration application programming interface (API) provided with WebLogic Server. This method is not recommended for initial cluster implementation. For further information, refer to the documentation:
Oracle® Fusion Middleware Developing Custom Management Utilities With JMX for Oracle WebLogic Server 11g Release 1 (10.3.1).

Creating a Cluster by Using the Administration Console



ORACLE

Copyright © 2009, Oracle. All rights reserved.

Creating a Cluster by Using the Administration Console

To configure a cluster using the Administration Console, perform the following steps:

1. In the Administration Console, expand **Environment** and click **Clusters**.
2. Click **New**.
3. Enter the name of the new cluster.
4. Select the **Messaging Mode** that you want to use for this cluster:
 - In Oracle WebLogic Server 10.3.1 environments, unicast is the default messaging mode. Unicast requires less network configuration than multicast.
 - Multicast messaging mode is also available and may be appropriate in the environments that use previous versions of Oracle WebLogic Server. However, unicast is the preferred mode considering the simplicity of configuration and flexibility.
5. If you are using Unicast message mode, enter the Unicast Broadcast Channel. This channel is used to transmit messages within the cluster. If you anticipate high volume of traffic and your applications use session replication, you may prefer to define a separate channel for cluster messaging mode. If you do not specify a channel, the default channel is used.

Creating a Cluster by Using the Administration Console (continued)

6. If you are using Multicast message mode:
 1. Enter the multicast address of the new cluster. A multicast address is an IP address in the range from 224.0.0.0 through 239.255.255.255. The default value used by Oracle WebLogic Server is 239.192.0.0. You should avoid using x.0.0.1 multicast addresses in the range of the permitted multicast address. The multicast address you configure must be unique to a cluster and should not be shared by other clusters.
 2. Enter the Multicast Port for the new cluster. The multicast port is used by cluster members to communicate with each other. Valid values are between 1 and 65535.
7. Click **OK**.

Setting Cluster Attributes

The screenshot shows the Oracle WebLogic Server Administration Console interface. At the top, there is a header bar with buttons for 'New', 'Clone', and 'Delete'. Below this is a table titled 'Clusters(Filtered - More Columns Exist)' showing one cluster entry:

| <input type="checkbox"/> | Name | Cluster Address | Cluster Messaging Mode | Migration Basis | Default Load Algorithm | Replication Type | Cluster Broadcast Channel | Servers |
|--------------------------|---------------|-----------------|------------------------|-----------------|------------------------|------------------|---------------------------|---------|
| <input type="checkbox"/> | MedRecCluster | | Unicast | Database | Round Robin | (None) | | |

Below the table, a message says 'Showing 1 to 1 of 1 Previous | Next'. The main content area is titled 'Settings for MedRecCluster' and has tabs for Configuration, Monitoring, Control, Deployments, Services, and Notes. Under Configuration, there are sub-tabs for General, Messaging, Servers, Replication, Migration, Singleton Services, Scheduling, Overload, Health Monitoring, and HTTP.

The 'General' tab settings are displayed:

- Name:** MedRecCluster
- Default Load Algorithm:** round-robin (selected from a dropdown menu)
- Cluster Address:** (empty input field)
- Number Of Servers In Cluster Address:** 3

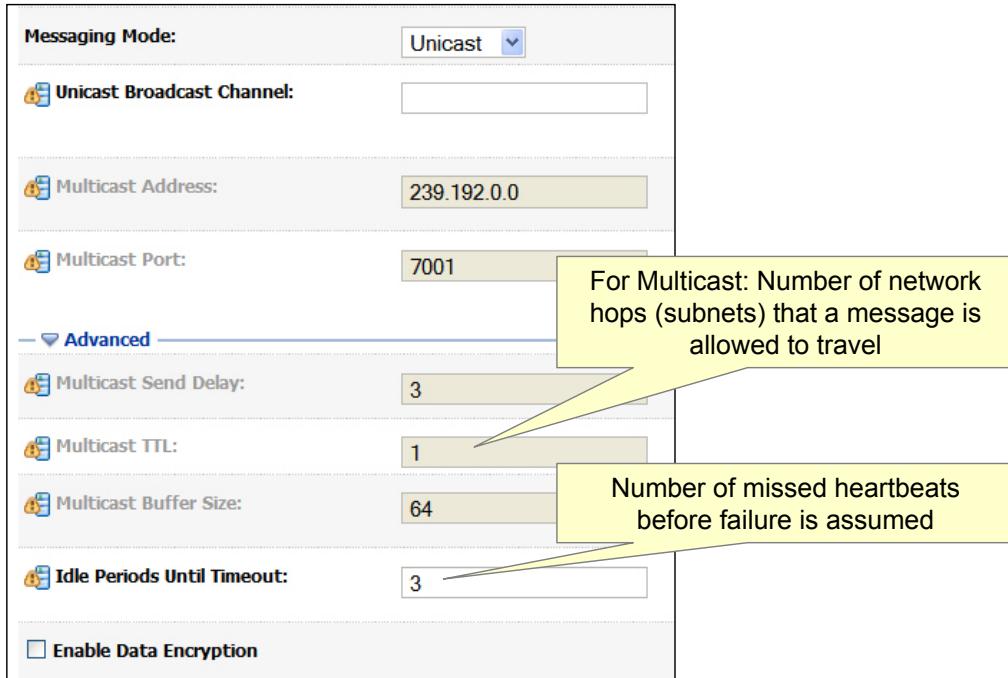
At the bottom right of the interface is the ORACLE logo, and at the very bottom is the copyright notice: Copyright © 2009, Oracle. All rights reserved.

Setting Cluster Attributes

Some of the important cluster attributes are:

- **Default Load Algorithm:** The algorithm to be used for load balancing between replicated services if none is specified for a particular service. The *round-robin* algorithm cycles through a list of Oracle WebLogic Server instances in order. *Weight-based* load balancing improves on the round-robin algorithm by taking into account a preassigned weight for each server. In *random* load balancing, requests are routed to servers at random.
- **Cluster Address:** The address that is to be used by clients to connect to this cluster. This address may be either a DNS host name that maps to multiple IP addresses or a comma-separated list of single address host names or IP addresses.
- **Number Of Servers In Cluster Address:** The number of servers to be listed from this cluster when generating a cluster address automatically. This setting has no effect if Cluster Address is explicitly set.

Configuring Cluster Communication



ORACLE

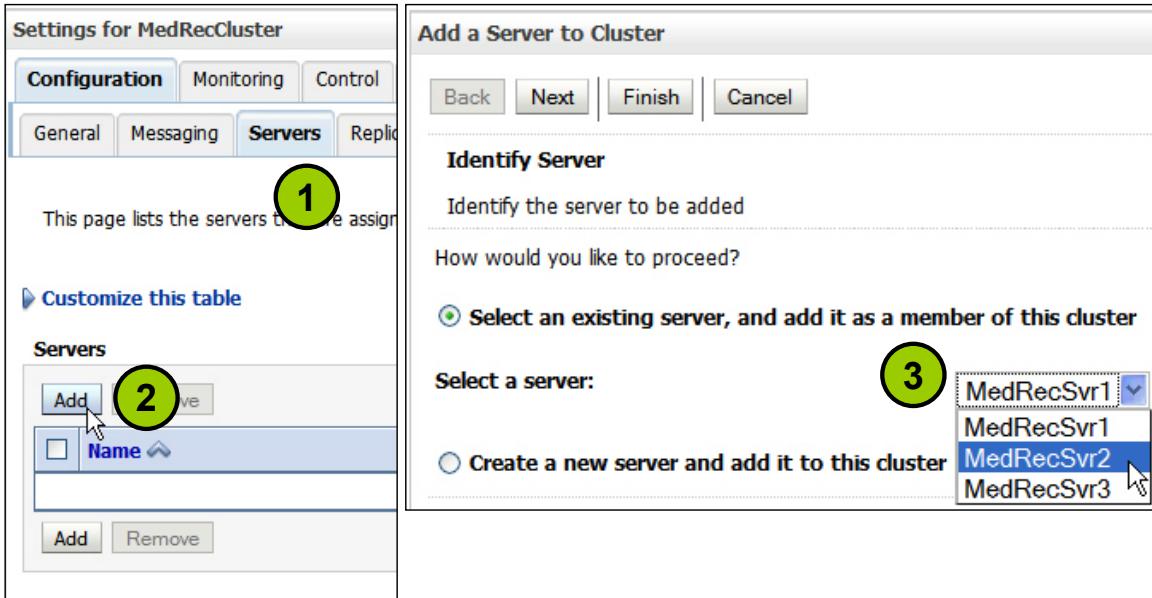
Copyright © 2009, Oracle. All rights reserved.

Configuring Cluster Communication

When you configure multicast mode of communication, you may want to set up the following parameters using the Advanced configuration:

- Multicast Send Delay:** The amount of time (between 0 and 100 milliseconds) to delay sending message fragments over multicast to avoid operating system–level buffer overflow
- Multicast TTL:** The number of network hops (between 1 and 255) that a cluster multicast message is allowed to travel. 1 restricts the cluster to one subnet.
- Multicast Buffer Size:** The multicast socket send or receive buffer size (at least 64 kilobytes)
- Idle Periods Until Timeout:** The maximum number of periods that a cluster member waits before timing out a member of a cluster
- Enable Data Encryption:** The option to enable encryption of data exchanges between servers in a cluster

Adding Cluster Members: Option 1



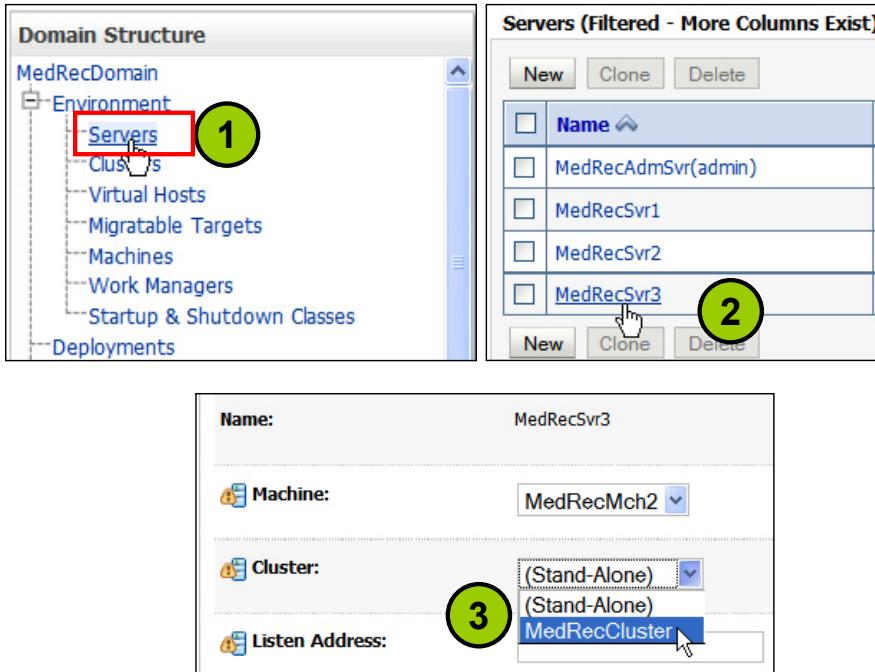
ORACLE

Copyright © 2009, Oracle. All rights reserved.

Adding Cluster Members: Option 1

1. In the Administration Console, expand Environment, and then click Clusters. Select the cluster to which you want to assign the servers. Finally, click the **Configuration > Servers** tab.
2. Click **Add**.
3. To add an existing server to the cluster, select the **Select an existing server, and add it as a member of this cluster** option, and then select a server from the list. To create a new server as part of a cluster, select the **Create a new server and add it to this cluster** option.

Adding Cluster Members: Option 2



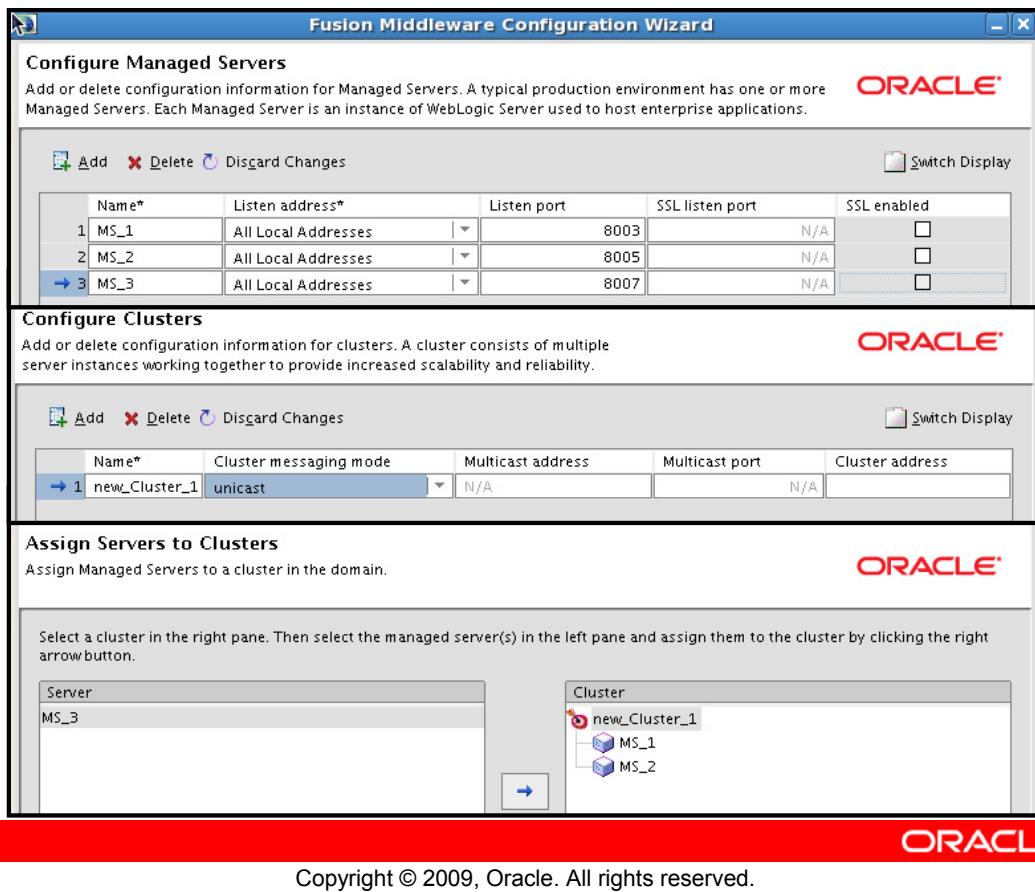
ORACLE®

Copyright © 2009, Oracle. All rights reserved.

Adding Cluster Members: Option 2

1. In the left pane of the Console, select **Environment** > **Servers**.
2. Select an existing server or create a new one. Confirm that the Configuration > General tab is displayed.
3. Specify whether or not this server will be a stand-alone server or will belong to a cluster.

Creating a Cluster with the Configuration Wizard

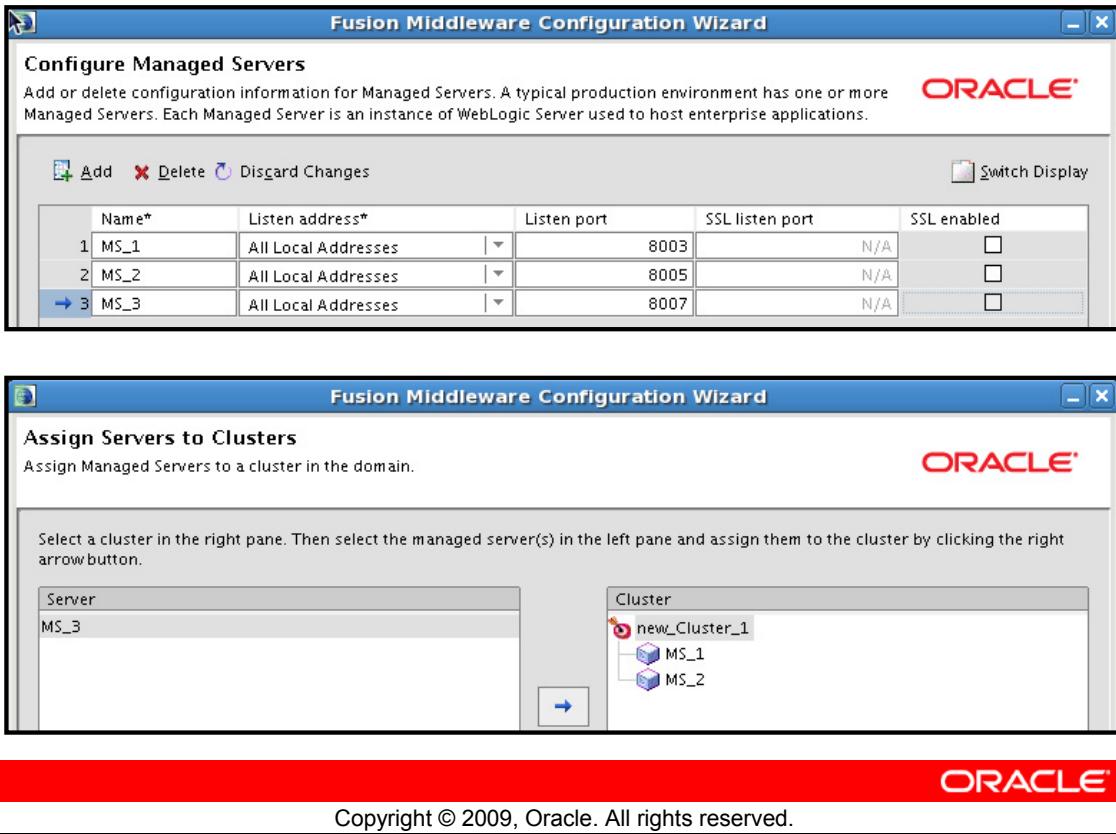


Copyright © 2009, Oracle. All rights reserved.

Creating a Cluster with the Configuration Wizard

You can also create and configure a cluster by using the Configuration Wizard. This is especially useful when creating a domain and you have already planned to configure clusters in the domain.

Clusters and the Configuration Wizard



Clusters and the Configuration Wizard

To group managed servers into clusters while creating a new domain, you can perform the following tasks in the Configuration Wizard:

- Add or delete clusters, or change the configuration of existing clusters.
- Assign the managed servers to a cluster in the domain.
- Optionally, use a managed server as an HTTP proxy for each cluster within the domain.

Clusters and WLST

```
connect('myuser','mypass','myhost:7001')
edit()
startEdit()
cd('/')
cmo.createCluster('HRWebCluster')
cd('/Clusters/HRWebCluster')
cluster = getMBean('/Clusters/HRWebCluster')
cd('/Servers/serverA')
cmo.setCluster(cluster)
cd('/Servers/serverB')
cmo.setCluster(cluster)
cd('/Servers/serverC')
cmo.setCluster(cluster)
activate()
disconnect()
exit()
```

Create a new cluster.

Assign cluster members.

ORACLE

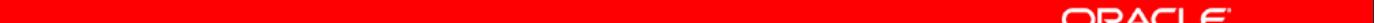
Copyright © 2009, Oracle. All rights reserved.

Clusters and WLST

The example in the slide demonstrates the creation of a new cluster by using the WebLogic Scripting Tool (WLST). After you create a new ClusterMBean, update each ServerMBean and assign the ClusterMBean to it.

Creating a Cluster Using the Cluster MBean

- The Cluster MBean is used to create a cluster by using Ant or command-line tools.
- Configuring the cluster from the command line requires the combined use of Cluster and Server MBeans.
- To create new clusters within a domain, use:
 - `weblogic.management.configuration.ClusterMBean`



ORACLE

Copyright © 2009, Oracle. All rights reserved.

Creating a Cluster Using the Cluster MBean

It is possible to create a complete cluster from the command-line script. This is more complex than the Administration Console, but it can provide greater flexibility in making small changes to the cluster. Use the Cluster MBean to create new cluster instances. You will notice that the MBean provides all the attributes needed to configure a cluster, and the attributes and operations are provided by the Administration Console.

The configuration of clusters usually involves a coordinated effort between the Cluster MBean and Server MBeans for the servers that join and participate in a cluster. For more information about Cluster MBean, visit <http://e-docs.bea.com/wls/docs103/wlsmbeanref/core/index.html>.

Synchronization When Starting Servers in a Cluster

The screenshot shows two separate windows, MedRecSvr2 and MedRecSvr3, both titled "MedRecSvr". Each window has a menu bar with File, Edit, View, Terminal, Tabs, and Help. The MedRecSvr2 window displays log messages from May 12, 2009, at 2:29:14 PM EDT, indicating the start of an asynchronous replication service with a remote cluster address of "null". The MedRecSvr3 window displays similar log messages, starting at 2:32:48 PM EDT, showing the server state changing to RESUMING, and listing various ports and protocols it is listening on (iiop, t3, ldap, snmp, http). Both windows have scroll bars on the right side.

```

MedRecSvr2
File Edit View Terminal Tabs Help
<May 12, 2009 2:29:14 PM EDT> <Notice> <Cluster> <BEA-000162> <Starting "async" replication service with remote cluster address "null">
<May 12, 2009 2:29:14 PM
] is now listening on fe00:0:0:0:a8a1:b0ff:fe00:305:7025 for protocols iiop, t3, CLUSTER-BROADCAST, ldap, snmp, http.>
<May 12, 2009 2:29:14 PM
] is now listening on 0.0.0.0:10216 for protocols iiop, t3, CLUSTER-BROADCAST, ldap, snmp, http.>
<May 12, 2009 2:29:14 PM
] is now listening on 10.216.4.16:7025 for protocols iiop, t3, CLUSTER-BROADCAST, ldap, snmp, http.>
<May 12, 2009 2:29:14 PM
] is now listening on 10.216.99.112:7025 for protocols iiop, t3, CLUSTER-BROADCAST, ldap, snmp, http.>
<May 12, 2009 2:29:14 PM
] is now listening on 127.0.0.1:7025 for protocols iiop, t3, CLUSTER-BROADCAST, ldap, snmp, http.>
<May 12, 2009 2:32:48 PM EDT> <Notice> <WebLogicServer> <BEA-000365> <Server state changed to RESUMING>
<May 12, 2009 2:32:48 PM EDT> <Notice> <Cluster> <BEA-000162> <Starting "async" replication service with remote cluster address "null">
<May 12, 2009 2:32:48 PM EDT> <Notice> <Server> <BEA-002613> <Channel "Default[2]>
] is now listening on fe00:0:0:0:a8a1:b0ff:fe00:305:7025 for protocols iiop, t3, CLUSTER-BROADCAST, ldap, snmp, http.>
<May 12, 2009 2:32:48 PM EDT> <Notice> <Server> <BEA-002613> <Channel "Default[5]>
] is now listening on 127.0.0.1:7025 for protocols iiop, t3, CLUSTER-BROADCAST, ldap, snmp, http.>
<May 12, 2009 2:32:48 PM EDT> <Notice> <Server> <BEA-002613> <Channel "Default[1]>
] is now listening on 10.216.4.16:7025 for protocols iiop, t3, CLUSTER-BROADCAST, ldap, snmp, http.>
<May 12, 2009 2:32:48 PM EDT> <Notice> <Server> <BEA-002613> <Channel "Default[4]>
] is now listening on 0:0:0:0:0:0:1:7025 for protocols iiop, t3, CLUSTER-BROADCAST, ldap, snmp, http.>
<May 12, 2009 2:32:48 PM EDT> <Notice> <Server> <BEA-002613> <Channel "Default[3]>
] is now listening on fe00:0:0:0:a8a0:b0ff:fe00:305:7025 for protocols iiop, t3, CLUSTER-BROADCAST, ldap, snmp, http.>
<May 12, 2009 2:32:48 PM EDT> <Notice> <Server> <BEA-002613> <Channel "Default">
is now listening on 10.216.99.112:7025 for protocols iiop, t3, CLUSTER-BROADCAST, ldap, snmp, http.>
<May 12, 2009 2:32:48 PM EDT> <Notice> <WebLogicServer> <BEA-000330> <Started WebLogic Managed Server "MedRecSvr3" for domain "MedRecDomain" running in Producti

```

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Synchronization When Starting Servers in a Cluster

To start an Oracle WebLogic Server instance that participates in a cluster, you use the same procedure as you would for starting any managed server. You identify the administration server that the instance should use. All the configuration information for the server is obtained from the configuration repository that is associated with the administration server.

If clustered server instances do not have open sockets for peer-to-peer communication, failed servers may also be detected via the Oracle WebLogic Server heartbeat. All the server instances in a cluster use multicast or unicast to broadcast regular server heartbeat messages to the other members of the cluster. Each heartbeat message contains data that uniquely identifies the server that sends the message. Servers broadcast their heartbeat messages at regular intervals of 10 seconds. In turn, each server in a cluster monitors the multicast or unicast address to ensure that the heartbeat messages of all peer servers are being sent.

If a server that is monitoring the multicast or unicast address misses three heartbeats from a peer server (that is, if it does not receive a heartbeat from the server for 30 seconds or longer), the monitoring server marks the peer server as “failed.” It then updates its local JNDI tree, if necessary, to retract the services that were hosted on the failed server. Thus, servers can detect failures even if they have no sockets open for peer-to-peer communication.

Synchronization When Starting Servers in a Cluster (continued)

When you start a managed server in a cluster, the server instance identifies the other running server instances in the cluster by listening for heartbeats, after a warm-up period specified by the MemberWarmupTimeoutSeconds parameter in ClusterMBean. The default warm-up period is 30 seconds.

Configuring OHS as Proxy Server

- To effectively use the load balancing and failover features, you should configure a proxy.
- You can configure OHS as the proxy by:
 - Including configuration directives in `httpd.conf`
 - Creating another file with directives and setting an include directive in `httpd.conf`
- The `WebLogicCluster` directive is the most important `mod_wl_ohs` for a cluster.
- You specify the list of host names of the managed servers with their ports in the `WebLogicCluster` directive.
- If you add or remove members to or from this list, you may have to restart OHS.

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Configuring OHS as Proxy Server

To effectively use the load balancing and failover features of the cluster, you should configure a proxy. Because OHS is already enabled with `mod_wl_ohs`, you can easily configure Oracle HTTP Server as the proxy server for the cluster. You can edit the `httpd.conf` file of the OHS instance and do one of the following:

- Set the `mod_wl_ohs` configuration directives in the `httpd.conf` file
- Create a configuration file such as the `mod_wl_ohs.conf` file with necessary configuration directives and set an appropriate include directive in `httpd.conf`

A typical `mod_wl_ohs.conf` file looks like this:

```
$ cat mod_wl*conf
LoadModule weblogic_module
"${ORACLE_HOME}/ohs/modules/mod_wl_ohs.so"
<IfModule mod_weblogic.c>
    WebLogicCluster wls1.com:7021,wls2.com:7021,wls3.com:7021
    ErrorPage http://myerrorpage.mydomain.com
    MatchExpression *.jsp
</IfModule>
<Location /medrec>
    SetHandler weblogic-handler
</Location>
$
```

Starting and Stopping OHS Manually

- To give effect to configuration changes to `httpd.conf`, you should restart OHS.
- The processing life cycle for OHS is managed by Oracle Process Manager and Notification Server (OPMN).
- The command-line interface to OPMN is `opmnctl`.
- To restart OHS, use the following command:

```
$> ./opmnctl restartproc process-type=OHS
```

- You can also stop, and then start OHS.

```
$> ./opmnctl stopproc process-type=OHS
$> ./opmnctl startproc process-type=OHS
```

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Starting and Stopping OHS Manually

Oracle HTTP Server is managed by OPMN, which manages the Oracle Application Server processes. You can use `opmnctl` to start, stop, and restart Oracle HTTP Server.

You can include the path (`<INSTANCE_HOME>/opmn/bin`) to the `opmnctl` location or change the directory to before using the `opmnctl` commands. `INSTANCE_HOME` is the location where the Web Tier instance containing this OHS instance has been configured. For example, in the class room environment, `opmnctl` is available in `/u01/app/oracle/instances/bin`.

- To start the Oracle HTTP Server process in the local instance:
`$> ./opmnctl startproc process-type=OHS`
- To stop the Oracle HTTP Server process:
`$> ./opmnctl stopproc process-type=OHS`
- To determine the state of Oracle HTTP Server:
`$> ./opmnctl status`
- To restart Oracle HTTP Server:
`$. ./opmnctl restartproc process-type=OHS`

Verifying Access Through OHS

Get the port on which OHS is running by using:

```
$> ./opmnctl status -l
Processes in Instance: wtinst
-----+-----+-----+-----+
-----+-----+
ias-component | process-type| pid |status |      uid |
memused |     uptime | ports
-----+-----+-----+-----+
-----+-----+
ohsa        | OHS          | 8614 | Alive | 1775979054 |
348736 | 0:00:29 | https:8889,https:4443,http:8888
```



Copyright © 2009, Oracle. All rights reserved.

Verifying Access Through OHS

You can verify that you are able to access the applications deployed to a cluster through OHS by directing your request to the port on which OHS is listening for requests. You can get the HTTP Listen port of OHS using the `opmnctl status -l` command. In the slide, OHS is running (HTTP) on port 8888.

Now, you can try to make a request to this port and see that the application is accessible.

Quiz

Which of the following is NOT an available configuration attribute associated with Oracle WebLogic Cluster?

1. Messaging mode
2. Multicast TTL
3. Multicast port
4. Broadcast server



Copyright © 2009, Oracle. All rights reserved.

Answer: 4

Remember that although clusters support a messaging mode for broadcast communication (unicast or multicast), there is no attribute called broadcast server.

Summary

In this lesson, you should have learned how to:

- Prepare your environment for a cluster
- Create and configure a cluster
- Add servers to a cluster
- Start up and shut down clustered servers



Copyright © 2009, Oracle. All rights reserved.

Practice 16 Overview: Configuring Clusters

This practice covers the following topics:

- Creating a cluster
- Assigning two servers to the cluster
- Verifying the port and status of Oracle HTTP Server



Copyright © 2009, Oracle. All rights reserved.

Practice 16 Overview: Configuring Clusters

See Appendix A for the complete steps to do the practice.

17

Managing Clusters

ORACLE®

Copyright © 2009, Oracle. All rights reserved.

Objectives

After completing this lesson, you should be able to do the following:

- Deploy applications to a cluster
- Describe the replication of a session state in a cluster
- Configure replication groups
- Configure in-memory replication
- Configure JDBC replication
- Configure file replication
- Configure a multitier cluster for EJB applications



Copyright © 2009, Oracle. All rights reserved.

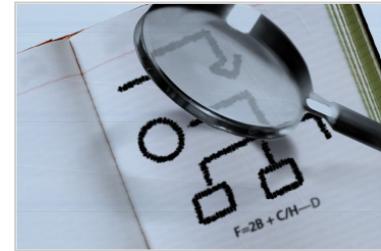
Objectives

Scenario

You deploy the application that you are using to evaluate the HTTP session failover feature. Configure Oracle HTTP Server to load balance between two managed servers in a cluster. Verify that the session failover happens appropriately.

Road Map

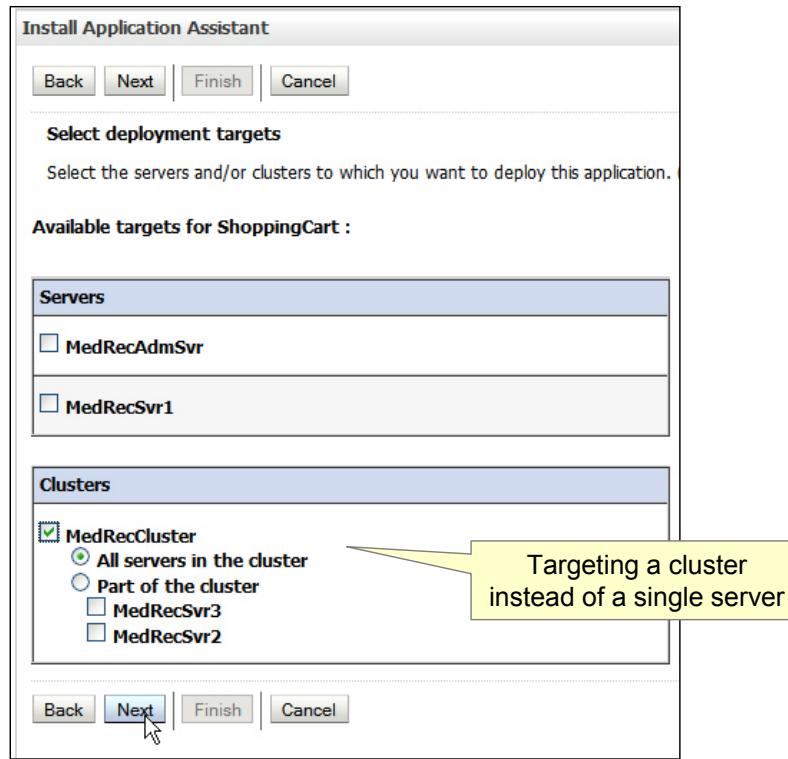
- Deploying applications
 - Selecting a cluster as the target
 - Two-phase deployment
 - Production redeployment
- HTTP session management
- EJB clustering
- Troubleshooting a cluster



ORACLE®

Copyright © 2009, Oracle. All rights reserved.

Deploying Applications to a Cluster



ORACLE®

Copyright © 2009, Oracle. All rights reserved.

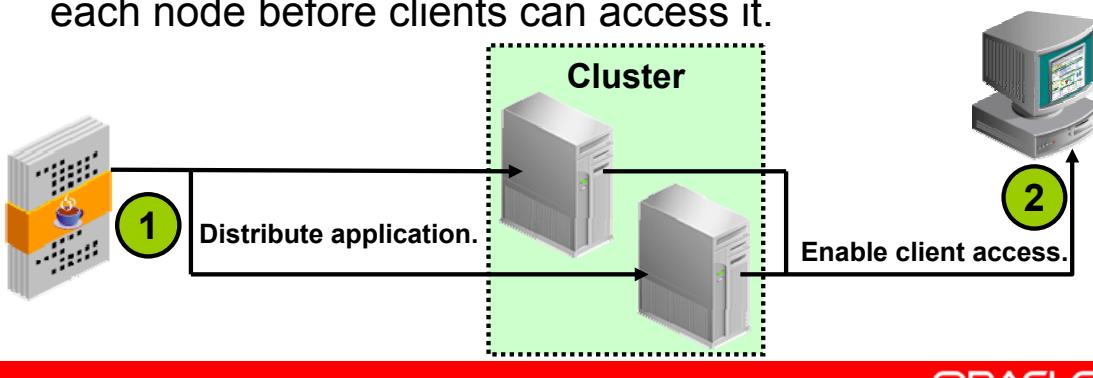
Deploying Applications to a Cluster

Regardless of the deployment tool that you use, when you initiate the deployment process, you specify the components to be deployed and the targets to which they will be deployed. The main difference between the way you deploy an application to a normal server and a cluster lies in your choice of the target. When you intend to deploy an application to the cluster, you select the target from the list of clusters and not from the list of servers.

Ideally, all servers in a cluster should be running and available during the deployment process. Deploying applications when some members of the cluster are unavailable is not recommended.

Two-Phase Deployment

- Applications are deployed using two-phase deployment (TPD).
 - Phase 1: Application components and modules are distributed to the server.
 - Phase 2: The application is deployed if phase 1 is successful and client access is permitted.
- This ensures that an application is available and active on each node before clients can access it.



ORACLE

Copyright © 2009, Oracle. All rights reserved.

Two-Phase Deployment

When deploying applications to a cluster, they must be packaged into a .war, .ear, or .jar file. WebLogic clusters use the concept of two-phase deployment.

- **Phase 1:** During the first phase of deployment, application components are distributed to the target server instances and the planned deployment is validated to ensure that the application components are successfully deployed. During this phase, user requests to the application being deployed are not allowed. If failures are encountered during the distribution and validation processes, the deployment is aborted on all server instances, including those on which the validation succeeded. Files that have been staged are not removed; however, container-side changes performed during the preparation are reverted.
- **Phase 2:** After the application components are distributed to targets and validated, they are fully deployed on the target server instances, and the deployed application is made available to the clients. If a failure occurs during this process, deployment to that server instance is canceled. However, a failure on one server of a cluster does not prevent successful deployment on other clustered servers.

If a cluster member fails to deploy an application, it fails at startup in order to ensure cluster consistency, because any failure of a cluster-deployed application on a managed server causes the managed server to abort its startup.

The two-phase commit feature enables you to avoid situations in which an application is successfully deployed on one node and not on the other.

Considerations for Deploying to Cluster

- It would be good to have all the servers in the cluster running before an application is deployed to a cluster.
- If phase 2 fails on one server, the application is still deployed to other servers in the cluster.
- Do not change cluster membership while deploying applications to the cluster.
- Oracle WebLogic Server allows partial deployment of applications to a partitioned server by default.
- You can configure Oracle WebLogic Server to disallow partial deployments by using the `enforceClusterConstraints` tag.

The red horizontal bar spans most of the width of the slide, positioned above the copyright notice.

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Considerations for Deploying to Cluster

When you deploy an application to a cluster, you should run all the servers in the cluster. If a server is unavailable when the application is deployed, WebLogic switches to a relaxed deployment model. In this model, deployments continue to all other nodes. Deployment completes on the partitioned server after it becomes reachable. When the unavailable server becomes available, it may experience a performance hit as the deployment restarts on that server.

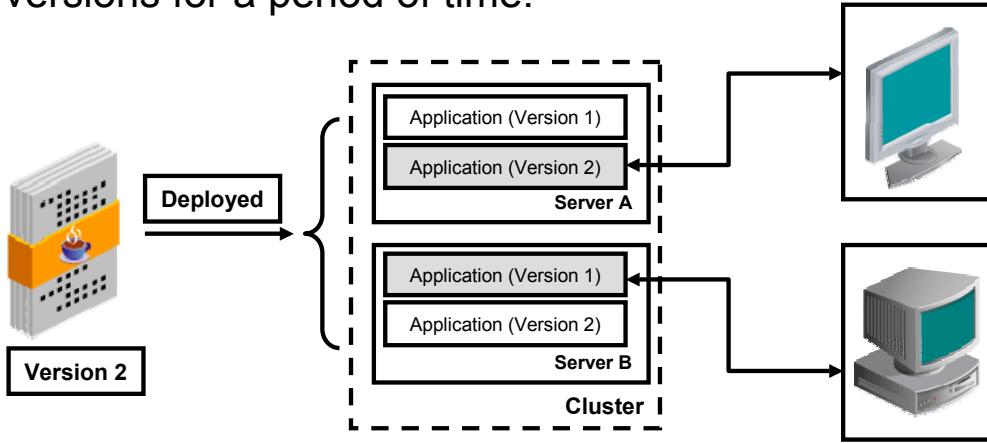
It is possible that even though a server is running, it cannot be reached by the administration server of the domain. Such an unreachable server is called a partitioned server. Oracle WebLogic Server allows deployment to such partitioned server. This is also referred to as partial deployment. One potential problem with partial deployment is that during the synchronization with other members of the cluster—when other servers in the cluster reestablish communications with the previously partitioned server instance—the user requests to the deployed applications and the attempts to create secondary sessions on that server instance may fail causing inconsistencies in cached objects.

You can configure Oracle WebLogic Server to disallow relaxed or partial deployments by using the `enforceClusterConstraints` tag with `weblogic.Deployer`.

Production Redeployment in a Cluster

When you use production redeployment of an application in a cluster, each server instance in the cluster retires the old version when the work is complete on that server.

- Therefore, different servers may be running different versions for a period of time.



ORACLE

Copyright © 2009, Oracle. All rights reserved.

Production Redeployment in a Cluster

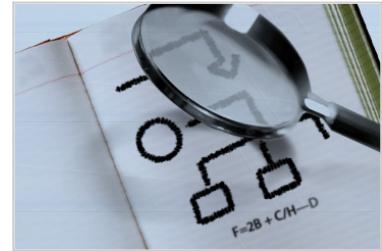
Production redeployment enables you to update and redeploy an application in a production environment without stopping the application or otherwise interrupting the application's availability to clients. You are saved the tasks of scheduling application down time, setting up redundant servers to host new application versions, manually managing client access to multiple application versions, and manually retiring older versions of an application.

The slide shows a cluster that contains Server A and Server B. Both servers initially run version 1 of the application. When version 2 of the application is deployed to the cluster, it is deployed to both servers in the cluster. However, because different clients are using the application on different servers, version 1 may be retired at different points. If the clients have completed using the application on Server A, any new requests are to version 2 of the application. On Server B, the client may still be interacting with version 1.

In a WebLogic Server cluster, each clustered server instance retires its local deployment of the retiring application version when the current workload is completed. This means that an application version may be retired on some clustered server instances before it is retired on other servers in the cluster. However, in a cluster failover scenario, client requests that are failed over are always handled by the same application version on the secondary server, if the application version is still available. If the same application version is not available on the secondary server, the failover does not succeed.

Road Map

- Deploying applications to clusters
- HTTP session management
 - HTTP session failover
 - Replication groups
 - In-memory replication
 - Persistent replication
- EJB session replication
- Troubleshooting a cluster

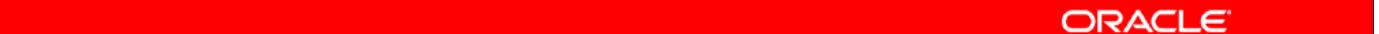


ORACLE®

Copyright © 2009, Oracle. All rights reserved.

HTTP Session Failover

- Web applications use HTTP sessions to track information in server memory for each client.
- By default, when a client fails over to another server in the cluster, its session information is lost.
- Oracle WebLogic Server supports several *Session Replication* strategies to recover sessions from failed servers:
 - In-memory replication
 - JDBC replication
 - File replication
- Replication is configured for each Web application within its `weblogic.xml` file.

ORACLE

Copyright © 2009, Oracle. All rights reserved.

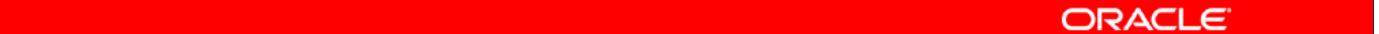
HTTP Session Failover

Web application components, such as servlets and JavaServer Pages (JSPs), maintain data on behalf of clients using an `HttpSession` instance that is available on a per-client basis. To provide high availability of Web applications, shared access to one `HttpSession` object must be provided. `HttpSession` objects can be replicated within Oracle WebLogic Server by storing their data using in-memory replication, file system persistence, or in a database.

In a cluster, the load-balancing hardware or the proxy plug-in in Web Server redirects the client requests to any available server in the Oracle WebLogic Server cluster. The cluster member that serves the request obtains a replica of the client's HTTP session state from the available secondary server in the cluster.

HTTP Session State Replication

- Session persistence is configured using the `<session-descriptor>` element in the `weblogic.xml` deployment descriptor file.
 - Each persistence method has its own set of configurable parameters.
- You should also configure access to the cluster through a collection of Web servers with identically configured proxy plug-ins or load-balancing hardware.
- Machine definition is one of the factors that WebLogic takes into account when it chooses another server as its backup for session information.

ORACLE

Copyright © 2009, Oracle. All rights reserved.

HTTP Session State Replication

Load balancing for servlet and JSP HTTP session states can be accomplished using separate load-balancing hardware or by using the built-in load-balancing capabilities of a WebLogic proxy plug-in.

For clusters that use a bank of Web servers and WebLogic proxy plug-ins, the proxy plug-ins provide only a round-robin algorithm for distributing requests to the servlets and JSPs in a cluster.

Clusters that use a hardware load-balancing solution can use any load-balancing algorithm that the hardware supports, including advanced load-based balancing strategies that monitor the utilization of individual machines.

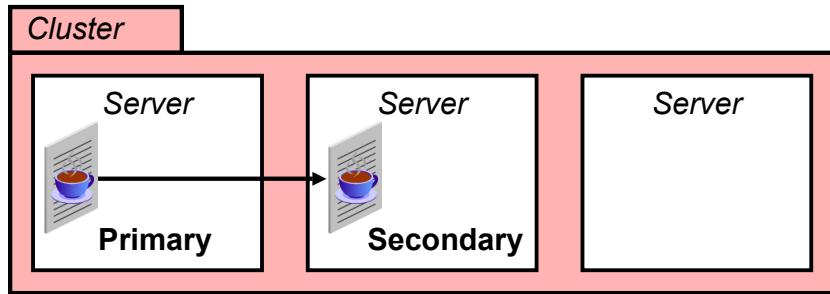
Note: This release of Oracle WebLogic Server provides Asynchronous HTTP Session Replication (AsyncRep) to improve cluster performance.

AsyncRep gives you the option to choose asynchronous session replication to the secondary server. It also provides the ability to throttle the maximum size of the queue that batches up session objects before the batched replication takes place.

AsyncRep is used to specify the asynchronous replication of data between a primary server and a secondary server. In addition, this option enables the asynchronous replication of data between a primary server and a remote secondary server located in a different cluster according to the cluster topology of MAN.

HTTP Session: In-Memory Replication

- Each user's session always exists on two servers:
 - Primary
 - Secondary
- Every update to the primary session is automatically replicated on the secondary server, either synchronously (default) or asynchronously (batch).



ORACLE®

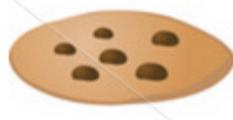
Copyright © 2009, Oracle. All rights reserved.

HTTP Session: In-Memory Replication

Using in-memory replication, Oracle WebLogic Server copies a session state from one server instance to another. The primary server creates a primary session state on the server to which the client first connects and a secondary replica on another Oracle WebLogic Server instance in the cluster. The replica is kept up-to-date so that it can be used if the server that hosts the Web application fails.

In-Memory Replication and Proxy Servers

- Oracle WebLogic Server uses nonpersistent cookies to track the primary and secondary servers for each client.
- Subsequent requests from the same client must be directed to the same primary server by the proxy.
- The server that is being failed over to automatically assumes the role of the primary server.



ORACLE

Copyright © 2009, Oracle. All rights reserved.

In-Memory Replication and Proxy Servers

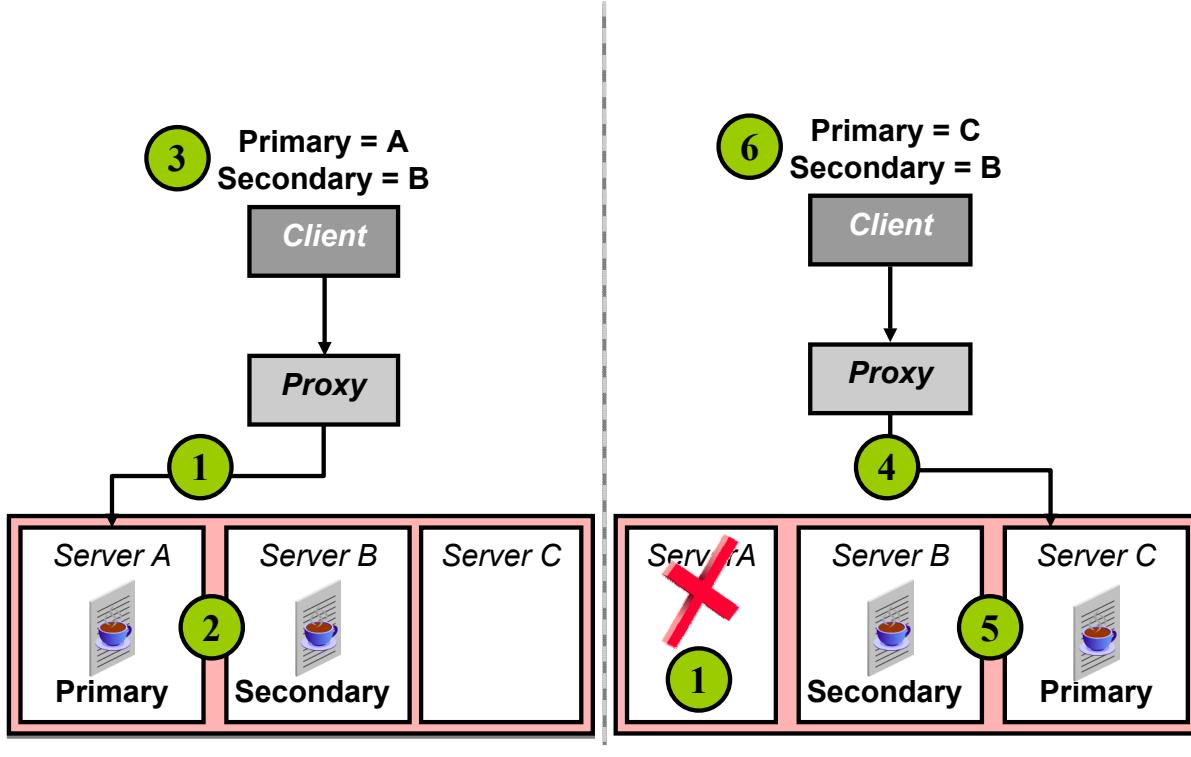
To use in-memory replication for HTTP session states, you must access the Oracle WebLogic Server cluster using either a collection of Web servers with identically configured WebLogic proxy plug-ins or a load-balancing hardware.

The WebLogic proxy plug-in maintains a list of Oracle WebLogic Server instances that host a clustered servlet or JSP, and forwards HTTP requests to these instances using a round-robin strategy.

Oracle WebLogic Server uses client-side cookies to keep track of the primary and secondary servers that host the client's servlet session state. If client browsers have disabled the cookie usage, Oracle WebLogic Server can also keep track of the primary and secondary servers using URL rewriting. With URL rewriting, both locations of the client session state are embedded into the URLs that are passed between the client and the proxy server. To support this feature, you must ensure that URL rewriting is enabled on the Oracle WebLogic Server cluster.

To support direct client access via the load-balancing hardware, the Oracle WebLogic Server replication system allows clients to use secondary session states regardless of the server to which the client fails over. Oracle WebLogic Server uses client-side cookies or URL rewriting to record the primary and secondary server locations. However, this information is used only as a history of the servlet session state location. When accessing a cluster via the load-balancing hardware, clients do not use the cookie information to actively locate a server after a failure.

In-Memory Replication: Example



ORACLE

Copyright © 2009, Oracle. All rights reserved.

In-Memory Replication: Example

The graphic in the slide depicts a client accessing a Web application that is hosted in a cluster. All client requests are forwarded to the Oracle WebLogic Server cluster via a proxy, such as `HttpClusterServlet` or a Web server plug-in.

To provide failover services for the Web application, the primary server replicates the client's session state to a secondary server in the cluster. This ensures that a replica of the session state exists even if the primary server fails (for example, due to a network failure).

In the example in the slide, initially Server A is the primary server and Server B is configured as the secondary server.

Initially, Server A is the primary server and Server B is configured as the secondary server, whereby Server A replicates the session state to Server B.

If the primary server (Server A) fails, the proxy sends the request to another member of the cluster, say Server C.

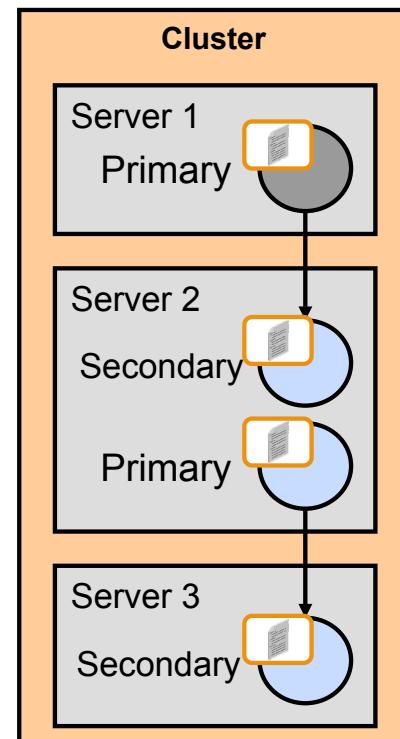
Because Server C is not secondary, it gets the session information from Server B that hosts the replica of the session state.

Now that Server C is serving the request, it becomes the primary and Server B remains secondary.

In the HTTP response, the proxy updates the client's cookie to reflect the new primary and secondary servers to account for the possibility of subsequent failovers.

In-Memory Replication

- WLS can replicate:
 - HttpSession objects
 - Stateful session EJBs
- Session objects exist on only two servers.
- Secondary:
 - The server is determined by the replication group and machine definition.
 - The object is created immediately after the primary object is created.
- Primary failure makes the backup object the primary object.



ORACLE

Copyright © 2009, Oracle. All rights reserved.

In-Memory Replication

Web application components, such as servlets and JSPs, maintain data on behalf of clients using an HttpSession instance that is available on a per-client basis.

To provide high availability of Web applications, shared access to one HttpSession object must be provided. HttpSession objects can be replicated within Oracle WebLogic Server by storing their data with in-memory replication, file system persistence, or in a database.

With in-memory replication, replicated objects are not accessible on all server instances in the cluster. Rather, when an object is created, it is called the *primary object*. On another server instance, a *backup object* is created. In the event of a failure of the primary object, the backup object is promoted as the primary object. If a failover occurs, another backup object is created.

This is optimal because replication of object data must occur only between the primary and backup objects (rather than the entire cluster).

Requirements for In-Memory Replication

- Subsequent requests from the same client must have access to the same primary object.
- To use in-memory replication for the HTTP session state, clients must access the cluster using one of these:
 - The load-balancing hardware (WLS aware)
 - Oracle HTTP Server with the `mod_wl_ohs` module
 - A collection of Web servers, or a single Web server, with WebLogic proxy plug-ins (configured identically)
 - Oracle WebLogic Server configured with `HttpClusterServlet`

The red horizontal bar spans most of the page width, centered below the main content area.

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Requirements for In-Memory Replication

Proxy Requirements

The WebLogic proxy plug-ins maintain a list of Oracle WebLogic Server instances that host a clustered servlet or JSP and forward HTTP requests to these instances by using a simple round-robin strategy.

The supported Web servers and proxy software include:

- Oracle HTTP Server with the `mod_wl_ohs` module configured
- Oracle WebLogic Server with `HttpClusterServlet`
- Netscape Enterprise Server with the Netscape (proxy) plug-in
- Apache with the Apache Server (proxy) plug-in
- Microsoft Internet Information Server with the Microsoft-IIS (proxy) plug-in

Load Balancer Requirements

If you choose to use load-balancing hardware instead of a proxy plug-in, you must use hardware that supports Secure Sockets Layer (SSL) persistence and passive cookie persistence. Passive cookie persistence enables Oracle WebLogic Server to write cookies through the load balancer to the client. The load balancer, in turn, interprets an identifier in the client's cookie to maintain the relationship between the client and the primary Oracle WebLogic Server that hosts the HTTP session state.

Configuring In-Memory Replication

1. Configure the proxy server (if applicable).
2. Optionally, define replication groups or machines, or both.
3. Specify the persistence type in the `weblogic.xml` deployment descriptor; the options include:
 - replicated
 - replicated-if-clustered
 - async-replicated
 - async-replicated-if-clustered

```
...
<session-descriptor>
    <persistent-store-type>replicated</persistent-store-type>
</session-descriptor>
...
```



Copyright © 2009, Oracle. All rights reserved.

Configuring In-Memory Replication

Set the persistent store method to one of the following options:

- **memory**: Disables persistent session storage
- **replicated**: Enables replication of session data across the clustered servers, and the session data is not persistent
- **replicated_if_clustered**: Replicates the in-effect persistent-store-type if the Web application is deployed on a clustered server; otherwise, the default is memory
- **async-replicated**: Enables asynchronous session replication in an application or a Web application
- **async-replicated-if-clustered**: Enables asynchronous session replication in an application or Web application when deployed to a cluster environment. If deployed to a single server environment, the session persistence/replication defaults to in-memory. This allows testing on a single server without deployment errors.
- **file**: Uses file-based persistence
- **async-jdbc**: Enables asynchronous JDBC persistence for HTTP sessions in an application or a Web application
- **jdbc**: Uses a database to store persistent sessions
- **cookie**: Stores all session data in the user's browser

Configuring In-Memory Replication (continued)

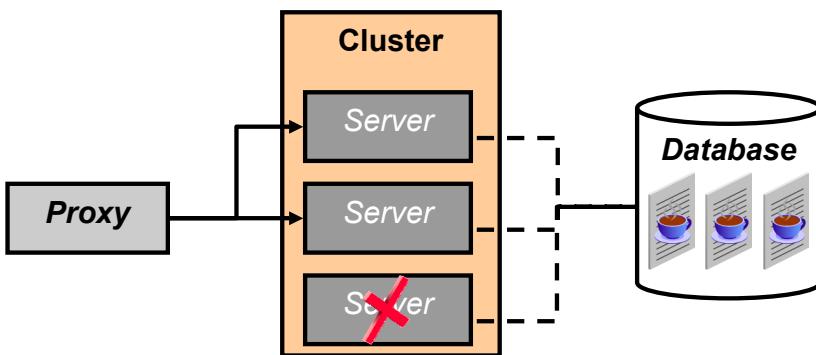
Cookie-based session persistence provides a stateless solution for session persistence by storing all session data in a cookie in the user's browser. Cookie-based session persistence is most useful when you do not need to store large amounts of data in the session. Cookie-based session persistence can simplify management of your Oracle WebLogic Server installation because clustering failover logic is not required. Because the session is stored in the browser, and not on the server, you can start and stop Oracle WebLogic Servers without losing sessions.

Note that cookies can persist only string data and there is no security on data because cookies are passed to and from the browser in clear text.

In the `<session-param>` element of `weblogic.xml`, set the `PersistentStoreType` attribute to `cookie`. Optionally, set a name for the cookie using the `PersistentStoreCookieName` attribute. The default is `WLCOOKIE`.

HTTP Session: Replication Using JDBC

- HTTP sessions can be persisted to a database using a common JDBC data source.
- The required Data Definition Language (DDL) file is available in the documentation.
- All members of the cluster have access to any client's session for failover purposes (no primary or secondary).



ORACLE

Copyright © 2009, Oracle. All rights reserved.

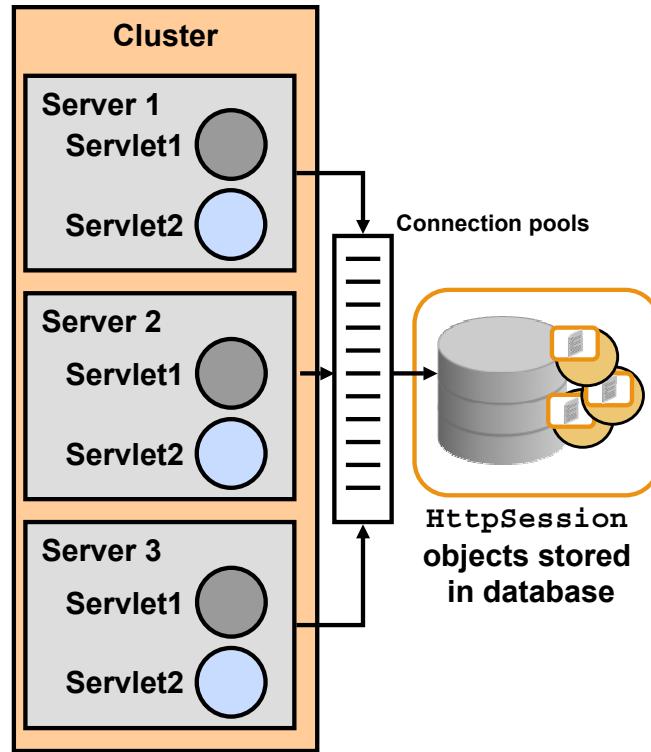
HTTP Session: Replication Using JDBC

With persistent Java Database Connectivity (JDBC) replication, a database is configured for storing `HttpSession` objects. After the database is configured, each server instance in a cluster uses an identical connection pool to share access to the database.

Whenever a Web application creates or uses a session object, the WebLogic Web container stores the session data persistently in the database. When a subsequent client request enters the cluster, any server in the cluster can handle the request. Each server in the cluster has identical access to the persistent store where it can look up the information needed to satisfy the client's request. This technique provides good failover capability because any server in the cluster can resolve a client's request, but there is a significant performance reduction due to the many database synchronizations required in a large Web-based system.

HTTP Session Replication Using JDBC

- All server instances have access to all sessions.
- Subsequent requests from the same client can be handled by any server.
 - Great failover capability
 - Significant performance reduction
- Changing session objects causes (slow) database synchronization.



ORACLE

Copyright © 2009, Oracle. All rights reserved.

HTTP Session Replication Using JDBC

Whenever a servlet creates or uses a session object, the servlet stores the session data persistently in the database. When a subsequent client request enters the cluster, any server in the cluster can handle the request. Each server in the cluster has identical access to the persistent store where it can look up the information needed to satisfy the client's request. This technique provides for good failover capability because any server in the cluster can resolve a client's request, but there is a significant performance reduction due to the many database synchronizations required in a large Web-based system.

Session persistence is not used for storing long-term data between sessions. That is, you should not rely on a session still being active when a client returns to a site at some later date. Instead, your application should record long-term or important information in a database.

You should not attempt to store long-term or limited-term client data in a session. Instead, your application should create and set its own cookies on the browser. Examples of this include an auto-login feature where the cookie lives for a long period or an auto-logout feature where the cookie expires after a short period of time. Here, you should not attempt to use HTTP sessions; instead, you should write your own application-specific logic.

Note that even though it is legal (according to the HTTP Servlet specification) to place any Java object in a session, only those objects that are serializable are stored persistently by Oracle WebLogic Server.

Configuring JDBC Replication

1. Create the required table in the database.
2. Create a JDBC data source that has read/write privileges for your database.
3. Configure JDBC session persistence in the `weblogic.xml` deployment descriptor.

```
...
<session-descriptor>
    <persistent-store-type>jdbc</persistent-store-type>
    <persistent-store-pool>MyDataSource</persistent-store-pool>
</session-descriptor>
...
```



Copyright © 2009, Oracle. All rights reserved.

Configuring JDBC Replication

Set up a database table named `wl_servlet_sessions` for JDBC-based persistence. The connection pool that connects to the database needs to have read/write access for this table. Create indexes on `wl_id` and `wl_context_path` if the database does not create them automatically. Some databases create indexes automatically for primary keys.

Set the `persistent-store-type` parameter in the `session-descriptor` element in the `weblogic.xml` deployment descriptor file to `j dbc`.

Set a JDBC connection pool to be used for persistence storage with the `persistent-store-pool` parameter in the `session-descriptor` element in the `weblogic.xml` deployment descriptor file. Use the name of a connection pool that is defined in the Oracle WebLogic Server Administration Console.

You can use the `j dbc-connection-timeout-secs` parameter to configure the maximum duration that the JDBC session persistence should wait for a JDBC connection from the connection pool, before failing to load the session data.

To prevent multiple database queries, Oracle WebLogic Server caches recently used sessions. Recently used sessions are not refreshed from the database for every request. The number of sessions in cache is governed by the `cache-size` parameter in the `session-descriptor` element of the Oracle WebLogic Server-specific deployment descriptor, `weblogic.xml`.

JDBC Persistent Table Configuration

A database table named WL_SERVLET_SESSIONS must exist with read/write access:

| | Column Head | Column Data Type |
|--------------------|-------------------|-------------------------------|
| Primary key | WL_ID | char, 100 variable width char |
| | WL_CONTEXT_PATH | |
| | WL_CREATE_TIME | numeric, 20 digits |
| | WL_IS_VALID | char, 1 character |
| | WL_SESSION_VALUES | BLOB, very large |
| | WL_ACCESS_TIME | numeric, 20 digits |
| | WL_IS_NEW | numeric, 20 digits |

Copyright © 2009, Oracle. All rights reserved.

JDBC Persistent Table Configuration

In the database that is mapped to the session persistence connection pool, you must configure a single table, WL_SERVLET_SESSIONS, which holds the values of all active session objects. The user specified with access to this table needs read/write/insert/delete access on the table to effectively manage the objects. The table requires the following eight columns:

- **WL_ID:** The session ID is used as the database primary key along with WL_CONTEXT_PATH. This is a variable-width alphanumeric data type of up to 100 characters.
- **WL_CONTEXT_PATH:** This is the context. This column is used with WL_ID as the primary key. This is a variable-width alphanumeric data type of up to 100 characters.
- **WL_IS_NEW:** This value is True as long as the session is classified in the “new” state by the Servlet engine. This is a single char column.
- **WL_CREATE_TIME:** This is the time when the session was originally created. This is a numeric column, 20 digits.
- **WL_IS_VALID:** This parameter is True when the session is available to be accessed by a servlet. It is used for concurrency purposes. This is a single char column.
- **WL_SESSION_VALUES:** This is the actual session data. It is a BLOB column.
- **WL_ACCESS_TIME:** This indicates the time when the session was last accessed. This is a numeric column, 20 digits.
- **WL_MAX_INACTIVE_INTERVAL:** This indicates the number of seconds between client requests before the session is invalidated. A negative time value indicates that the session should never time out. This is an integer column.

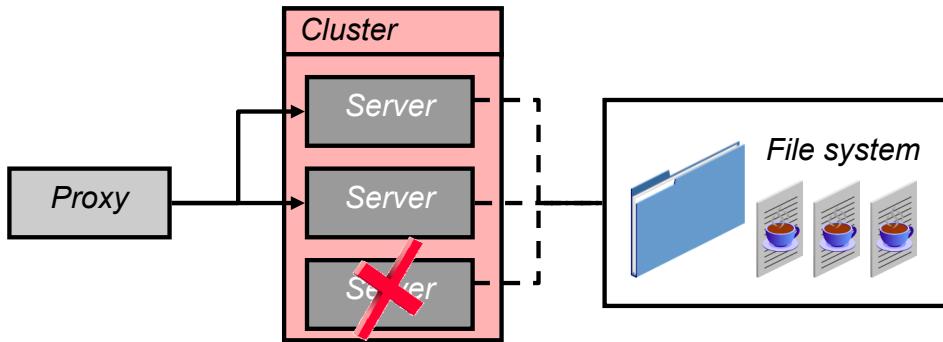
JDBC Persistent Table Configuration (continued)

The following is an example SQL statement to create this table, for Oracle Database:

```
create table wl_servlet_sessions ( wl_id VARCHAR2(100) NOT NULL,
wl_context_path VARCHAR2(100) NOT NULL, wl_is_new CHAR(1),
wl_create_time NUMBER(20), wl_is_valid CHAR(1), wl_session_values
LONG RAW, wl_access_time NUMBER(20), wl_max_inactive_interval
INTEGER, PRIMARY KEY (wl_id, wl_context_path) );
```

HTTP Session Replication Using File

File replication is similar to JDBC replication, but it persists sessions to a highly available file system.



ORACLE

Copyright © 2009, Oracle. All rights reserved.

HTTP Session Replication Using File

The session state may also be stored in a file. For file-based persistence:

- You must create the directory in which to store the file
- The file must have the appropriate access privileges

Configuring File Replication

1. Create a folder shared by all servers on the cluster on a highly available file system.
2. Assign read/write privileges to the folder.
3. Configure file session persistence in the `weblogic.xml` deployment descriptor.

```
...
<session-descriptor>
  <persistent-store-type>file</persistent-store-type>
  <persistent-store-dir>/mnt/wls_share</persistent-store-dir>
</session-descriptor>
...
```



Copyright © 2009, Oracle. All rights reserved.

Configuring File Replication

In the `weblogic.xml` deployment descriptor file, set the `persistent-store-type` parameter in the `session-descriptor` element to `file`.

Set the directory where Oracle WebLogic Server stores the sessions using the `persistent-store-dir` parameter. You must create this directory and make sure that appropriate access privileges are assigned to the directory.

Ensure that you have enough disk space to store the number of valid sessions multiplied by the size of each session. You can find the size of a session by looking at the files created in the location indicated by the `persistent-store-dir` parameter. Note that the size of each session can vary as the size of serialized session data changes.

Each server instance has a default persistent file store that requires no configuration. Therefore, if no directory is specified, a default store is automatically created in the `<server-name>\data\store\default` directory. However, the default store is not shareable among clustered servers.

Other options for `<persistent-store-type>`:

memory: When you use memory-based storage, all session information is stored in memory and is lost when you stop and restart Oracle WebLogic Server. To use memory-based, single-server, nonreplicated persistent storage, set the `PersistentStoreType` attribute in the `<session-param>` element of the `weblogic.xml` file to `memory`.

Configuring File Replication (continued)

cookie: Cookie-based session persistence provides a stateless solution for session persistence by storing all session data in a cookie in the user's browser. Cookie-based session persistence is most useful when you do not need to store large amounts of data in the session.

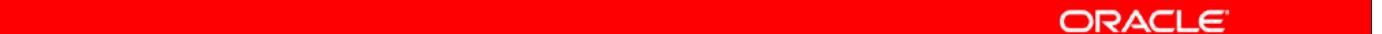
Cookie-based session persistence can simplify management of your Oracle WebLogic Server installation because clustering failover logic is not required. Because the session is stored in the browser, and not on the server, you can start and stop Oracle WebLogic Servers without losing sessions. But remember that cookies can persist only string data and that there is no security on the data as cookies are passed to and from the browser in clear text.

To set up cookie-based session persistence:

- In the `<session-param>` element of `weblogic.xml`, set the `PersistentStoreType` attribute to `cookie`
- Optionally, set a name for the cookie using the `PersistentStoreCookieName` attribute. The default is `WLCOOKIE`.

Replication Groups

- A replication group is a logical grouping of related servers in a cluster.
- WLS enables you to determine where to put backup objects using replication groups.
- WLS attempts to:
 - Send backup objects to a preferred secondary replication group, if it is configured
 - Send backup objects to a different machine
 - Avoid sending backup objects to servers in the same replication group

The red bar spans most of the width of the slide, centered horizontally.

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Replication Groups

By default, Oracle WebLogic Server attempts to create replicas of certain services on a machine other than the one that hosts the primary service.

Oracle WebLogic Server enables you to further control where the secondary states are placed by using replication groups. A replication group is a preferred list of clustered instances to use for storing session state replicas. When you configure a server instance that participates in a cluster, you can assign the server instance membership in a replication group. You can also assign a preferred secondary replication group to be considered for replicas of the primary HTTP session states that reside on the server.

When a client attaches to a cluster and creates an instance of a service, that service instance is automatically replicated in Oracle WebLogic Server (such as an HttpSession or a stateful session EJB). Oracle WebLogic Server instance that hosts the primary object honors the preferred secondary replication group if it is configured. Otherwise, a secondary on a remote machine is chosen for replication before trying to replicate to the local server.

An administrator can configure replication groups to operate such that secondary objects for replicated services always reside on different hardware. In earlier versions of Oracle WebLogic Server, the cluster would ensure that a replicated service exists on a different machine. However, because one computer can host multiple IP addresses and thus multiple machines, a replicated instance might not be protected from a general hardware failure. The creation of replication groups solves this issue.

Replication Groups

- Replication groups:
 - Represent a subset of servers within a cluster
 - Help to determine the placement of secondary sessions (for example, avoid replicating within the same room)
 - Are not explicitly defined in the console-like machines and clusters
- WLS attempts to:
 - Send secondary sessions to servers that are assigned to the *preferred secondary replication group* of the primary server
 - Avoid sending secondary sessions to servers that are assigned to the same replication group as the primary server



Copyright © 2009, Oracle. All rights reserved.

Replication Groups (continued)

By default, Oracle WebLogic Server attempts to create session state replicas on a machine other than the one that hosts the primary session state. You can further control where secondary states are placed using replication groups. A replication group is a preferred list of clustered servers to be used for storing session state replicas.

Using the Oracle WebLogic Server Console, you can define unique names for machines that host individual server instances. These machine names can be associated with the new Oracle WebLogic Server instances to identify where the servers reside in your system.

Machine names are used to indicate servers that run on the same machine. For example, you would assign the same machine name to all server instances that run on the same machine or the same server hardware.

If you are not running multiple Oracle WebLogic Server instances on a single machine, you need not specify the Oracle WebLogic Server machine names. Servers without a machine name are treated as though they reside on separate machines.

When you configure a clustered server instance, you can assign the server to a replication group and a preferred secondary replication group for hosting replicas of the primary HTTP session states that are created on the server.

Configuring Replication Groups

Select each server in a cluster and assign each a pair of replication groups.

The screenshot shows the 'Settings for dizzy1' page in the Oracle WebLogic Server Administration Console. The 'Cluster' tab is selected. The page includes fields for 'Replication Group', 'Preferred Secondary Group', 'Cluster Weight', and 'Interface Address', each with a description and a 'More Info...' link. A 'Save' button is at the bottom.

| Setting | Value | Description |
|----------------------------|----------------------------------|--|
| Replication Group: | <input type="text"/> | Defines preferred clustered instances considered for hosting replicas of the primary HTTP session states created on the server. More Info... |
| Preferred Secondary Group: | <input type="text"/> | Defines secondary clustered instances considered for hosting replicas of the primary HTTP session states created on the server. More Info... |
| Cluster Weight: | <input type="text" value="100"/> | The proportion of the load that this server will bear, relative to other servers in a cluster. More Info... |
| Interface Address: | <input type="text"/> | The IP address of the NIC that this server should use for multicast traffic. More Info... |

ORACLE

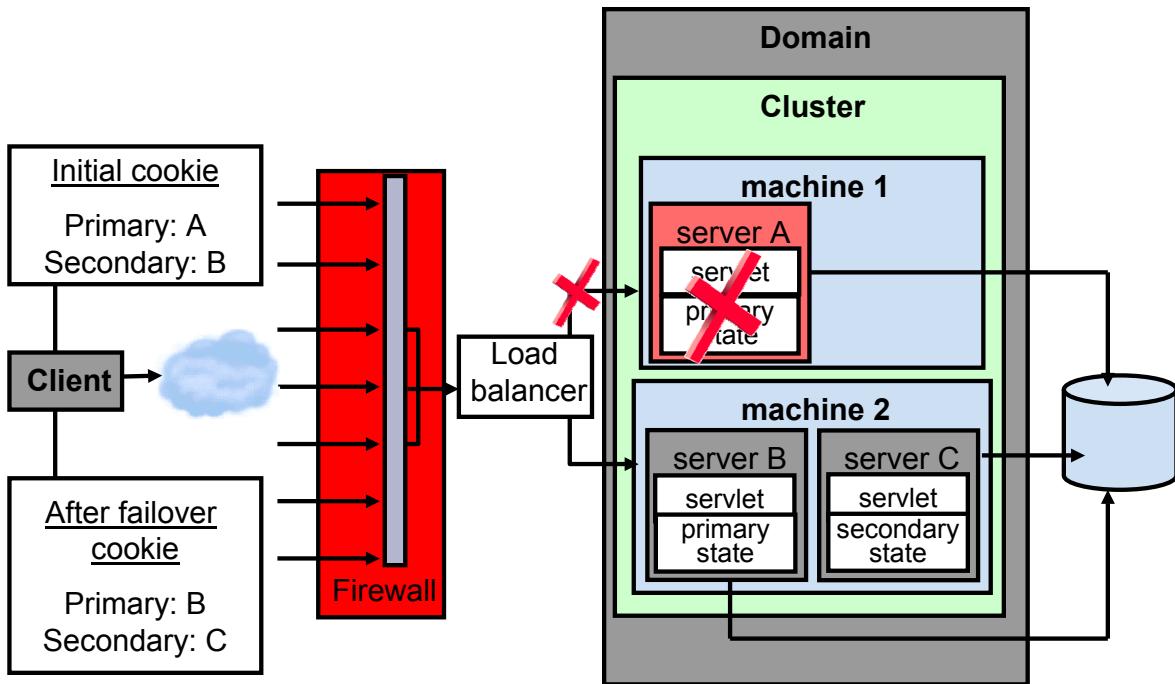
Copyright © 2009, Oracle. All rights reserved.

Configuring Replication Groups

If a cluster hosts servlets or stateful session EJBs, you might want to create replication groups of the Oracle WebLogic Server instances to host the session state replicas.

- **Replication Group:** The name of the replication group to which the server belongs. It is recommended that you group together all servers that have a relationship with one another (for example, servers that run on the same machine). For greater flexibility, you can define a different replication group for each server.
- **Preferred Secondary Group:** The name of the replication group to use to host the replicated HTTP session states for the server. You should select a secondary group in which all servers run on a different machine than the replication group's servers. For greater flexibility, you can select a secondary replication group that contains a single server running on a different machine.

Failover with Replication Groups



ORACLE

Copyright © 2009, Oracle. All rights reserved.

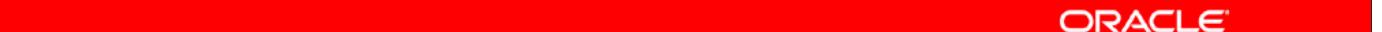
Failover with Replication Groups

When the client first makes a request, it is sent to server A. Because the application uses either an HTTP session or a stateful session bean, the client becomes pinned to a server. A cookie is written to the client machine stating that the primary state is stored on server A and the secondary on server B. Where the secondary state is stored is chosen based on machines and replication groups.

When server A fails, requests from the client go to any available server in the cluster. If it did not host the secondary, it will sync the session from the old secondary and become the new primary.

HTTP State Management Best Practices

- Create WLS machines if you are replicating the state across servers on different physical machines.
- Use replication groups to define the failover strategy.
- Choose the most appropriate replication strategy depending on the application needs and architecture.
- Use the ServerDebugConfig MBean to track session replication problems.
- Ensure that objects placed in replicated sessions are serializable.

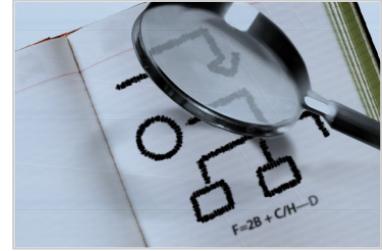


ORACLE

Copyright © 2009, Oracle. All rights reserved.

Road Map

- Deploying applications to clusters
- HTTP session management
- EJB session replication
 - EJB clustering deployment descriptors
 - Configuring stateless session beans
 - Configuring stateful session beans
- Troubleshooting a cluster



ORACLE®

Copyright © 2009, Oracle. All rights reserved.

Configuring EJB Clustering in Deployment Descriptors

- Clustering of EJB based on version 2 are configured in the application-specific deployment descriptors.
- When using clustering based on EJB version 3.0, you can use the deployment plans to implement clustering.

A snippet from `weblogic-ejb-jar.xml`:

```
...
<stateless-clustering>
    <stateless-bean-is-clusterable>True
    </stateless-bean-is-clusterable>

    <stateless-bean-load-algorithm>random
    </stateless-bean-load-algorithm>
    ...
</stateless-clustering>
...
```

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Configuring EJB Clustering in Deployment Descriptors

When using applications based on EJB 2.x, the cluster parameters are configured in the `weblogic-ejb-jar.xml` or `weblogic-cmp-rdbms-jar.xml` deployment descriptor files. Therefore, WebLogic administrators should discuss with their EJB development team the impact of the clustering features.

EJBs that are based on the 3.0 specification can be configured using annotations and can be configured using deployment plans. However, EJB 3.0 also supports all 2.x WebLogic-specific EJB features, but such features must continue to be configured as per the 2.x WebLogic-specific EJB features in deployment descriptor files.

Configuring EJB Clustering Using the Administration Console

The screenshot shows two panels. On the left is the 'Domain Structure' panel for 'MedRecDomain', with 'Clusters' highlighted (circled in green). A callout '1' points to this selection. On the right is the 'Settings for MedRecCluster' panel, showing the 'General' tab selected. A callout '2' points to the 'General' tab. A callout '3' points to the 'Number Of Servers In Cluster Address:' field, which contains the value '3'. A note says: 'Default for all EJBs, if not overridden' and 'Required only if Address is a single DNS name'. Another note says: 'Optionally, override the default server addresses and ports used by stubs.' The Oracle logo is at the bottom.

Copyright © 2009, Oracle. All rights reserved.

Configuring EJB Clustering Using the Administration Console

As an administrator, you configure the default EJB cluster settings for your domain using the following steps:

1. Select **Environment > Clusters** within the Domain Structure panel of the console. Then select a specific cluster.
2. On the **General** tab, update any of the fields described as follows:
 - Default Load Algorithm:** The algorithm used for load balancing between replicated services, such as EJBs, if none is specified for a particular service. The *round-robin* algorithm cycles through a list of Oracle WebLogic Server instances in order. *Weight-based* load balancing improves on the round-robin algorithm by taking into account a preassigned weight for each server. In *random* load balancing, requests are routed to servers at random.
 - Cluster Address:** The address used by EJB clients to connect to this cluster. This address may be either a DNS host name that maps to multiple IP addresses or a comma-separated list of single address host names or IP addresses.
 - Number Of Servers In Cluster Address:** The number of servers listed from this cluster when generating a cluster address automatically

Configuring Clusterable Stateless Session EJBs

- The WLS-specific deployment descriptor has a tag for configuring stateless session EJB clustering parameters.
- A snippet from a typical `weblogic-ejb-jar.xml` file:

```
...
<stateless-session-descriptor>
<!-- Other Tags As Appropriate Here... -->
  <stateless-clustering>
    <stateless-bean-is-clusterable>True</stateless-bean-is-
clusterable>
    <stateless-bean-load-algorithm>random</stateless-bean-
load-algorithm>
    <stateless-bean-call-router-class-
name>beanRouter</stateless-bean-call-router-class-name>
    <stateless-bean-methods-are-idempotent>True</stateless-
bean-methods-are-idempotent>
  </stateless-clustering> ...
...
```



Copyright © 2009, Oracle. All rights reserved.

Clusterable Stateless Session Beans

The `<stateless-clustering>` tag specifies options that determine how WebLogic Server replicates stateless session EJB instances in a cluster. When `<stateless-bean-is-clusterable>` is True, the EJB can be deployed from multiple WebLogic Servers in a cluster. Calls to the home stub are load-balanced between the servers on which this bean is deployed, and if a server hosting the bean is unreachable, the call fails over to another server hosting the bean.

The `<stateless-bean-call-router-class-name>` tag specifies the name of a custom class to use for routing bean method calls. This class must implement `weblogic.rmi.cluster.CallRouter()`. If specified, an instance of this class is called before each method call. The router class has the opportunity to choose a server to route to based on the method parameters. The class returns either a server name or null, which indicates that the current load algorithm should select the server.

Set `<stateless-bean-methods-are-idempotent>` to True only if the bean is written such that repeated calls to the same method with the same arguments have exactly the same effect as a single call. This allows the failover handler to retry a failed call without knowing whether the call actually completed on the failed server. Setting this property to True makes it possible for the bean stub to recover from any failure as long as another server hosting the bean can be reached.

Clusterable EJBs: Idempotent Methods

Example of an idempotent method snippet in `weblogic-ejb-jar.xml`:

```
...
<!-- LAST TAG inside <weblogic-ejb-jar.xml> -->
<idempotent-methods>
    <method> <!-- can be repeated -->
        <ejb-name>exampleSession</ejb-name>
        <method-intf>Remote</method-intf>
        <method-name>processUser</method-name>
        <method-params>
            <method-param>java.lang.String</method-param>
        </method-params>
    </method>
</idempotent-methods>
...
</weblogic-ejb-jar>
```



Copyright © 2009, Oracle. All rights reserved.

Clusterable EJBs: Idempotent Methods

Note the usage of the `<method>` tag. A quick review of the `<method>` tag syntax is as follows:

- `<ejb-name>` specifies the name of the EJB that hosts the methods.
- `<method-intf>` is optional. It specifies the single interface where the methods are located. The value could be either Home, Remote, Local, or LocalHome. If it is skipped, the methods of all interfaces are specified.
- `<method-name>` is either a single method name or a “`*`” (all methods).
- `<method-params>` is optional and contains a list of `<method-param>` tags. If multiple methods have the same name with different parameters, you can specify the method signature. This cannot be used with the “`*`”.

Stateful Session Beans

- Each stateful session EJB is unique.
- All calls on a remote stub must be directed to the server that contains the EJB.

A stateful session EJB is “pinned” to the server that it is created on. Its remote stub must also be pinned to the same server.

ORACLE®

Copyright © 2009, Oracle. All rights reserved.

Configuring Clusterable Stateful Session EJBs

- The WLS-specific deployment descriptor has a tag for configuring stateful session EJB clustering parameters.
- The replication type for EJBs is InMemory or None.

```
<stateful-session-descriptor>
    <stateful-session-clustering>
        <home-is-clusterable> true      </home-is-clusterable>
        <home-load-algorithm> random   </home-load-algorithm>
        <home-call-router-class-name> common.QARouter
        </home-call-router-class-name>
        <replication-type>     InMemory </replication-type>
    </stateful-session-clustering>
</stateful-session-descriptor>
```



Configuring Clusterable Stateful Session EJBs

All the tags in `<stateful-session-clustering>` are optional.

Because an instance of a stateful session EJB is connected to a single client, invocations can be sent only to a single server, not a cluster. But the home stub invocations are stateless and thus can be clustered.

`<home-is-clusterable>` indicates if the home stub is clustered.

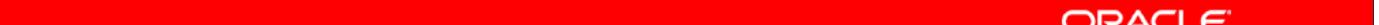
`<home-load-algorithm>` declares what load balancing algorithm to use.

The `<home-call-router-class-name>` tag here is similar to the `<stateless-bean-call-router-class-name>` tag shown previously, except that the router class acts only on the home stub.

The `<replication-type>` tag is used to indicate whether or not your stateful session EJB is replicated to a secondary server in the cluster. The value of the `<replication-type>` tag can be `InMemory` or `None`.

ReadWrite Versus Read-Only

- There are two types of entity beans to consider:
 - Read/write
 - Read-only
- For read/write entity beans, load balancing and failover occur only at the home level.
- For read-only entity beans, the replica-aware stub:
 - Load balances on every call
 - Does not automatically fail over in the event of a recoverable call failure



ORACLE

Copyright © 2009, Oracle. All rights reserved.

ReadWrite Versus Read-Only

There are two types of entity beans to consider: read/write entities and read-only entities.

- **Read/Write Entities:** When a home finds or creates a read/write entity bean, it obtains an instance on the local server and returns a stub pinned to that server. Load balancing and failover occur only at the home level. Because it is possible for multiple instances of the entity bean to exist in the cluster, each instance must read from the database before each transaction and write on each commit.
- **Read-Only Entities:** When a home finds or creates a read-only entity bean, it returns a replica-aware stub. This stub load-balances on every call but does not automatically fail over in the event of a recoverable call failure. Read-only beans are also cached on every server to avoid database reads.

Entity Bean Cluster-Aware Home Stubs

- Entity beans can have cluster-aware home stubs that have knowledge of the EJB Home objects on all WLS instances in the cluster.
- The `home-is-clusterable` deployment element in the `weblogic-ejb-jar.xml` file determines whether a home stub is cluster-aware.
- An example of setting an entity EJB home stub as cluster-aware:

```
<entity-clustering>
    <home-is-clusterable>True</home-is-clusterable>
    <home-load-algorithm>random</home-load-algorithm>
    <home-call-router-class-name>beanRouter
    </home-call-router-class-name>
</entity-clustering>
```

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Entity Bean Cluster-Aware Home Stubs

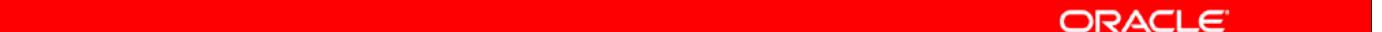
In an Oracle WebLogic Server cluster, the server-side representation of the Home object can be replaced by a cluster-aware “stub.” The cluster-aware home stub has knowledge of the EJB Home objects on all the Oracle WebLogic Servers in the cluster. The clustered home stub provides load balancing by distributing EJB lookup requests to available servers. It can also provide failover support for lookup requests because it routes those requests to available servers when other servers have failed.

All EJB types—stateless session, stateful session, and entity EJBs—can have cluster-aware home stubs. Whether or not a cluster-aware home stub is created is determined by the `home-is-clusterable` deployment element in `weblogic-ejb-jar.xml`.

When `home-is-clusterable` is True, the EJB can be deployed from multiple Oracle WebLogic Servers in a cluster. Calls to the home stub are load-balanced between the servers on which this bean is deployed. If a server that hosts the bean is unreachable, the call automatically fails over to another server that hosts the bean.

EJB Best Practices

- Set pool and cache sizes in accordance with anticipated load and execute threads per server.
- Understand that cache sizes equally affect all nodes in the cluster.
- Mark bean methods that can be called multiple times with impunity as idempotent in their deployment descriptors.



ORACLE

Copyright © 2009, Oracle. All rights reserved.

EJB Best Practices

Design Idempotent Methods: It is not always possible to determine when a server instance failed with respect to the work it was doing at the time of failure. For instance, if a server instance fails after handling a client request but before returning the response, there is no way to tell that the request was handled. A user that does not get a response retries, resulting in an additional request.

Failover for Remote Method Invocation (RMI) objects requires that methods be *idempotent*. An idempotent method is one that can be repeated with no negative side effects.

To configure idempotence, at bean level, set `stateless-bean-methods-are-idempotent` in `weblogic-ejb-jar.xml` to True. At method level, set `idempotent-methods` in `weblogic-ejb-jar.xml`.

Quiz

Select all valid values for the persistent store type element in weblogic.xml.

1. file
2. replicated
3. unicast
4. async-replicated-if-clustered
5. jdbc
6. async-wan



Copyright © 2009, Oracle. All rights reserved.

Answers: 1, 2, 4, 5

Quiz

Which two Oracle WebLogic Server features can be used to control the destination servers that are used for in-memory replication?

1. Web service
2. Replication group
3. Data source
4. Node Manager
5. Machine



Copyright © 2009, Oracle. All rights reserved.

Answers: 2, 5

Remember that clustered servers use machine and replication group boundaries to select destinations for replicated sessions.

Quiz

Which of the following terms is NOT associated with in-memory replication?

1. Cookie
2. Secondary
3. Session
4. Schema
5. Primary
6. Synchronous



Copyright © 2009, Oracle. All rights reserved.

Answer: 4

By default, in-memory replication involves both the synchronous creation of secondary copies of primary sessions and the tracking of these primary and secondary copies with cookies.

Quiz

Which types of replication configuration are allowed for EJBs?

1. JDBC
2. File
3. InMemory
4. None



Copyright © 2009, Oracle. All rights reserved.

Answers: 3, 4

Other types of replication are available for the replication of HTTP sessions only.

Summary

In this lesson, you should have learned how to:

- Deploy applications to a cluster
- Describe session state in a cluster
- Configure replication groups
- Configure in-memory replication
- Configure JDBC replication
- Configure file replication



Copyright © 2009, Oracle. All rights reserved.

Practice 17: Overview Managing Clusters

This practice covers the following topics:

- Defining a cluster as a target for new applications
- Retargeting existing applications to a cluster
- Deploying an application to a cluster
- Setting up in-memory session replication



Copyright © 2009, Oracle. All rights reserved.

Security Concepts and Configuration

18

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Objectives

After completing this lesson, you should be able to do the following:

- Use the WLS security architecture
- Configure security realms
- Configure users and groups
- Configure roles
- Configure policies
- Configure protection for:
 - Web application resources
 - EJBs



Copyright © 2009, Oracle. All rights reserved.

Objectives

Scenario

The Medical Records department has decided to explore the use of the security features provided by Oracle WebLogic Server to protect the application and other resources deployed in the Oracle WebLogic Server domain. You create users, groups, simple authentication, and authorization policies and understand the working of these policies in protecting a typical application.

Road Map

- Security overview
 - Oracle Platform Security Services
 - Oracle WLS Security
 - Oracle WLS Security Models
 - Introduction to WLS Security components
- Users and groups
- Protecting application resources

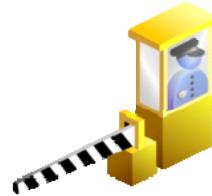


ORACLE

Copyright © 2009, Oracle. All rights reserved.

Introduction to Oracle WebLogic Security Service

- Security is a challenge in environments with diverse applications and Web-based services.
- This requires established and well-communicated security policies and procedures.
- You can use Oracle WebLogic Server as a comprehensive and flexible security infrastructure to protect applications.



ORACLE

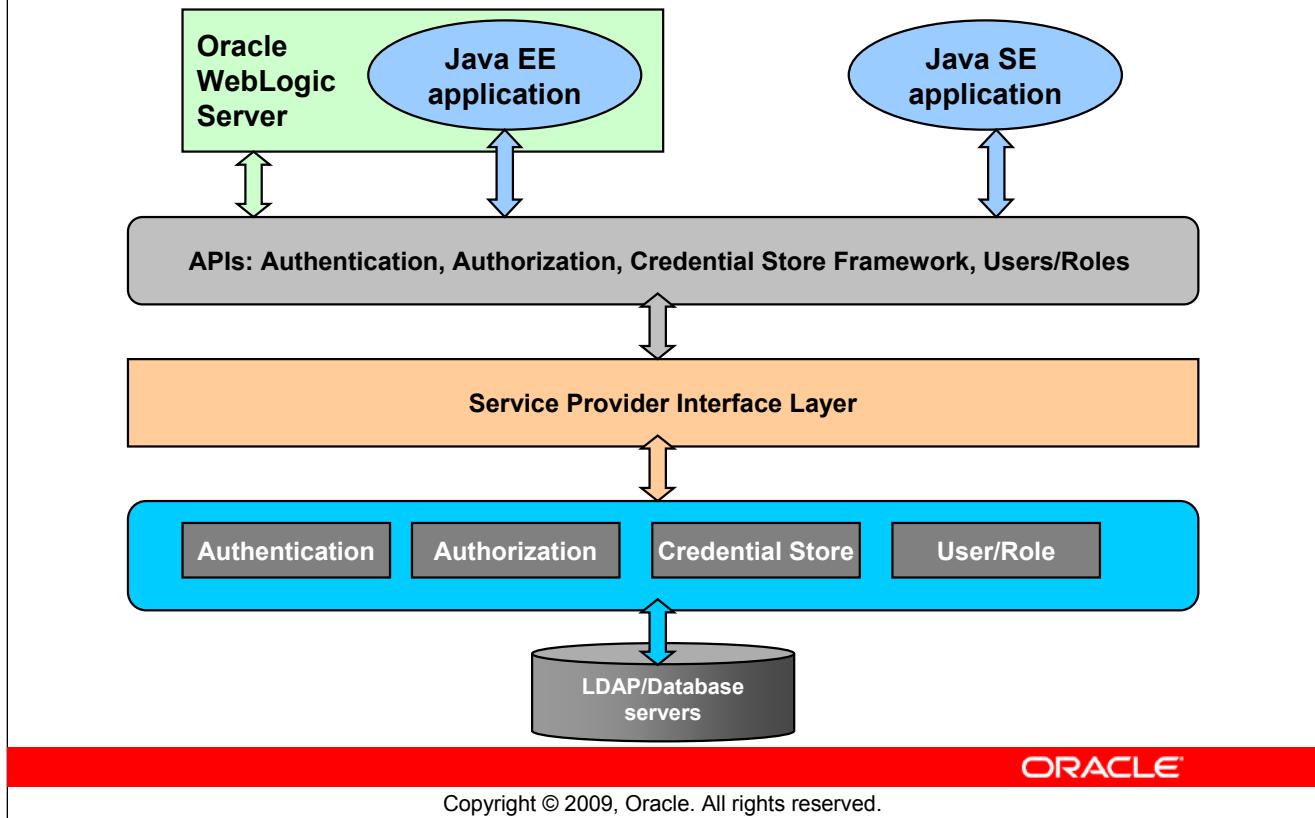
Copyright © 2009, Oracle. All rights reserved.

Introduction to Oracle WebLogic Security Service

Deploying, managing, and maintaining security is a challenge for organizations that provide new and expanded services to customers using the Web. To serve a worldwide network of Web-based users, an organization must address the fundamental issues of maintaining the confidentiality, integrity, and availability of the system and its data. Challenges to security involve every component of the system. Security across the infrastructure requires vigilance as well as established and well-communicated security policies and procedures.

Oracle WebLogic Server includes a security architecture that provides a comprehensive, flexible security infrastructure designed to address the security challenges of making applications available on the Web. WebLogic security can be used standalone to secure WebLogic Server applications or as part of an enterprise-wide, security management system that represents a best-in-breed, security management solution.

Oracle Platform Security Services



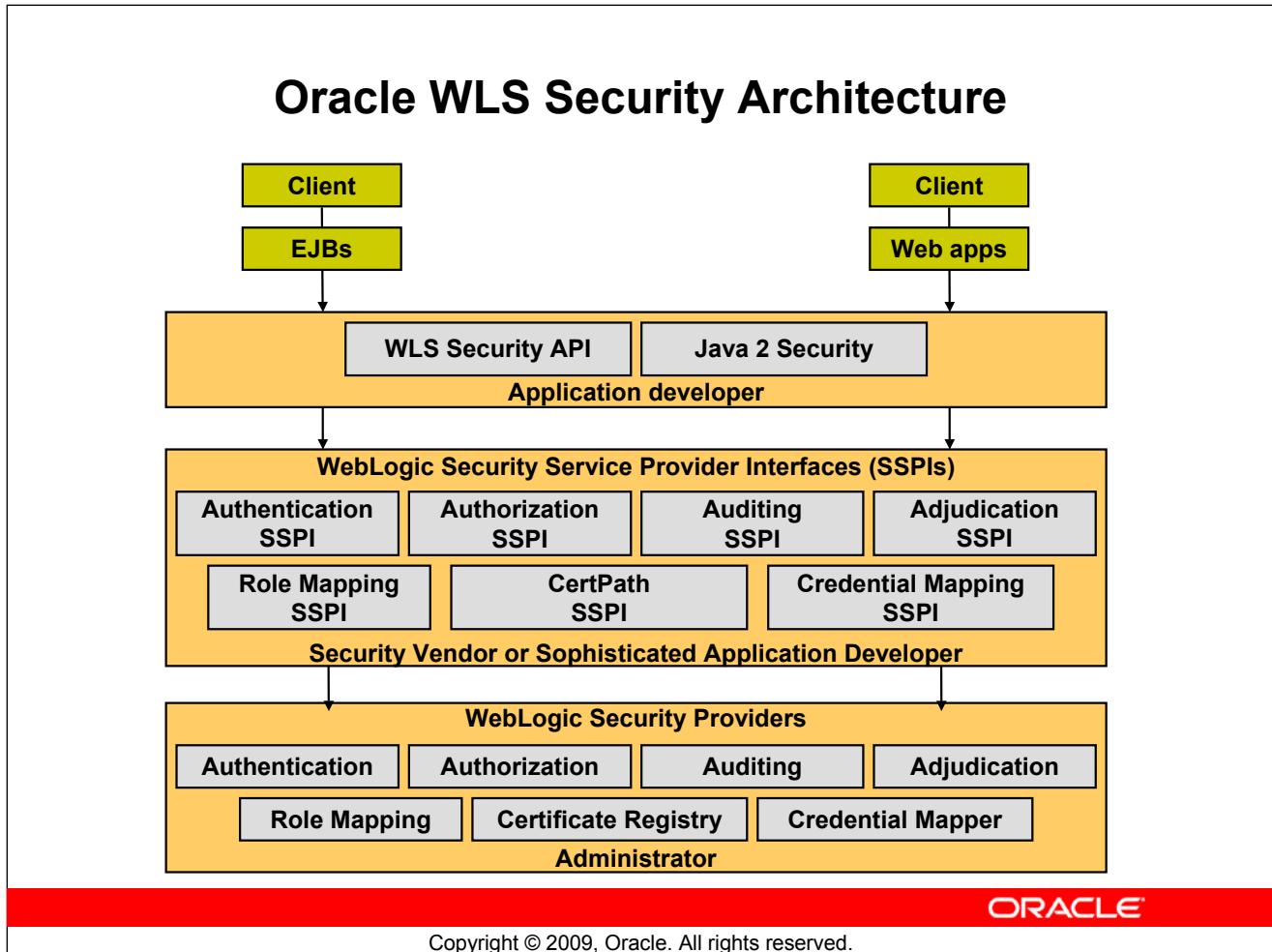
Oracle Platform Security Services

Oracle Platform Security Services (OPSS) is a security framework that runs on Oracle WebLogic Server. It combines the security features of the WebLogic Server and the Oracle Application Server to provide application developers, system integrators, security administrators, and independent software vendors with a comprehensive security platform framework for Java SE and Java EE applications. OPSS offers abstraction layer APIs that insulate developers from security and identity management implementation details.

- Developers can invoke the services provided by OPSS directly from the development environment (Oracle JDeveloper) using wizards.
- Administrators can configure the services of OPSS before and after the application is deployed into the Oracle WebLogic Server using Enterprise Manager pages, the Oracle WebLogic Administration Console, or command-line utilities.

OPSS provides security services to both the Oracle WebLogic Server and to the application deployed on it. Out of the box, Oracle WebLogic Server comes with a part of OPSS referred to as Common Security Services (CSS) that provides security services to the Oracle WLS components. This lesson explains the use of the CSS part of OPSS.

The complete OPSS is available when you install and use other components of Fusion Middleware such as Oracle SOA 11g Suite or Oracle WebCenter 11g Suite, or Oracle JDeveloper Suite. In such installations, you can configure and use OPSS fully. For further information about OPSS, refer to the *Oracle Fusion Middleware Security Guide*.

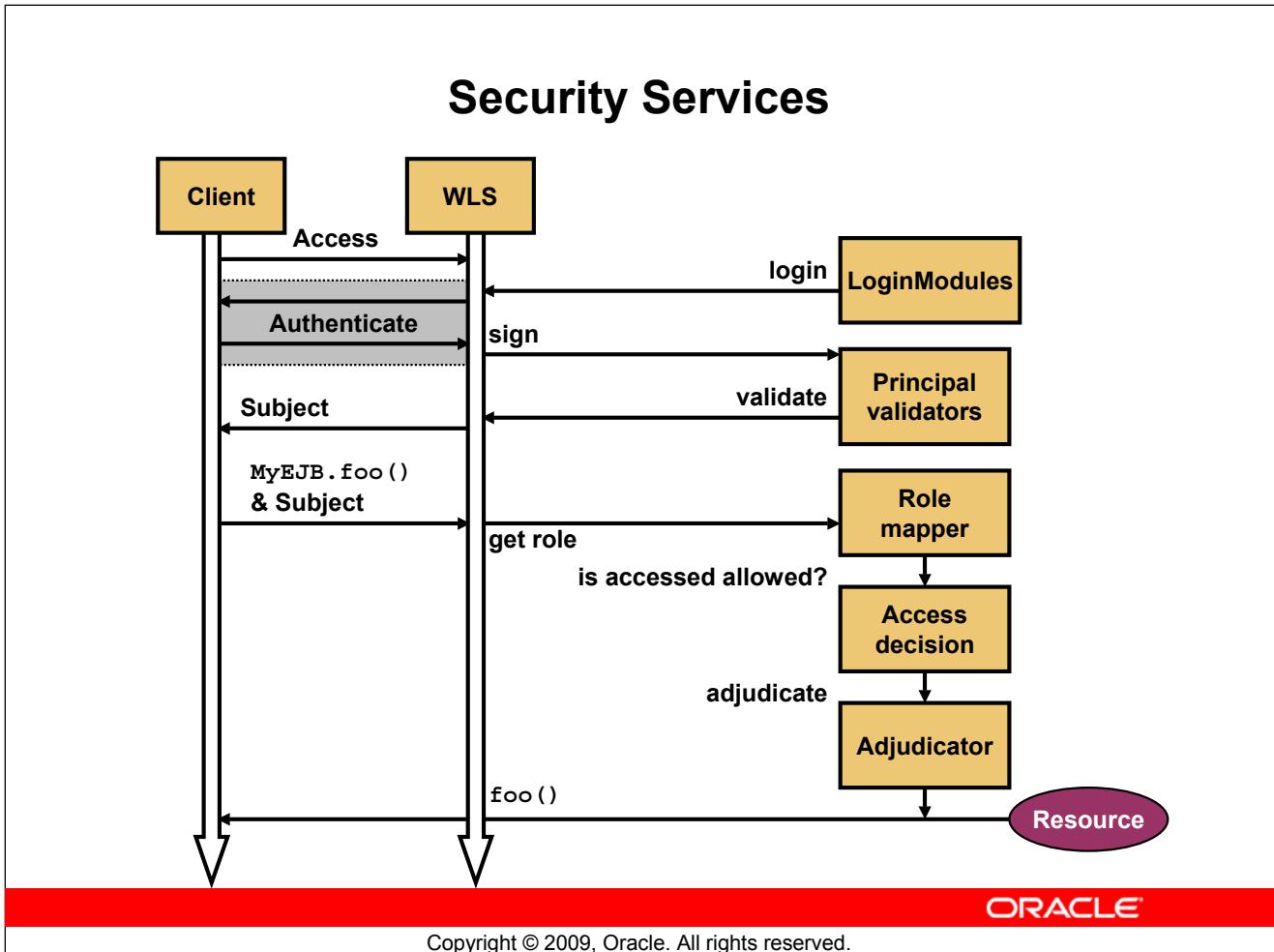


Oracle WLS Security Architecture

The WebLogic Security Service consists of:

1. A set of Security Service Provider Interfaces (SSPIs) for developing new security services that can be plugged into the Oracle WebLogic Server environment. SSPIs are available for Authentication, Authorization, Auditing, Role Mapping, Certificate Lookup and Validation, and Credential Mapping
2. A set of WebLogic security providers. These security providers are the Oracle implementation of the SSPIs and are available by default in the Oracle WebLogic Server product. The WebLogic security providers include Authentication, Authorization, and Auditing
3. A set of Application Programming Interfaces (APIs) that allow application developers to specify authorization information that is used when Oracle WebLogic Server acts as a client and to obtain information about the Subject and Principals used by Oracle WebLogic Server
4. J2SE 5.0 security packages, including Java Secure Socket Extensions (JSSE), Java Authentication and Authorization Service (JAAS), Java Security Manager, Java Cryptography Architecture and Java Cryptography Extensions (JCE), and Java Authorization Contract for Containers (JACC)

For more information, refer to the *Oracle Fusion Middleware Understanding Security for Oracle WebLogic Server* documentation.



Security Services

In a simple authentication, a user (or a client application), also referred to as the *subject*, attempts to log in to a system with a username/password combination. Oracle WebLogic Server establishes trust by validating that user's username and password. A principal represents the subject and the subject's features or properties. A subject can contain multiple principals. When the user (subject) enters the name and password, these properties and any other related information are encapsulated into the principal.

The validation of a principal is performed by the principal validator. After successfully proving the subject's identity, an authentication context is established, which allows an identified user or system to be authenticated to other entities.

During the authorization process, Oracle WebLogic Server determines whether a given subject can perform a given operation on a given resource, and returns the result of that decision to the client application. This process requires the use of access decisions, an adjudication provider, and possibly multiple role mapping providers.

Roles are obtained from the Role Mapping providers and input to the Access Decisions. The Access Decisions are then consulted for an authorization result. If multiple Access Decisions are configured and return conflicting authorization results (such as PERMIT and DENY), an Adjudication provider is used to resolve the contradiction by returning a final decision.

Overview of Security Concepts

- Authentication providers handle identity information and make it possible to associate with users, groups, or roles.
- Identity assertion providers map a valid token to an Oracle WebLogic Server user.
- An authorization provider is a process that is used to control the interactions between users and resources based on user identity.
- The adjudication provider weighs the results that multiple access decisions return to determine the final decision.
- The credential mapping process is initiated when application components access the authentication mechanism of a legacy system to obtain a set of credentials.
- Auditing provides a trail of activity. The auditing provider is used to log activity before and after security operations.

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Overview of Security Concepts

Authentication is the mechanism to answer the question “Who are you?” using credentials such as username/password combinations to determine whether the caller is acting on behalf of specific users or system processes. In WLS, authentication providers prove the identity of users or system processes and transport and make identity information available to the components of a system (via subjects) when needed.

A LoginModule authenticates the password and stores principals into the subject. There is a one- to-one relationship between an authentication provider and a LoginModule. Each authentication provider’s LoginModule store principals into the same subject.

Authorization answers the question “What can you access?” based on user identity or other information. Oracle WebLogic Server provides an authorization provider to limit the interactions between users and WebLogic resources to ensure integrity, confidentiality, and availability.

Authorization providers use access decision components to answer the question “Is access allowed?” Can a subject perform an operation on a WebLogic resource with specific parameters in an application? The result is PERMIT, DENY, or ABSTAIN.

Oracle WebLogic Server provides an auditing provider to collect, store, and distribute information about requests and the outcome of those requests for nonrepudiation. You can configure multiple auditing providers in a security realm, but none are required.

Confidentiality

- Oracle WebLogic Server supports the Secure Sockets Layer (SSL) protocol to secure the communication between the clients and the server.
- The SSL client authentication allows a server to confirm a user's identity by verifying that a client's certificate and public ID are valid and are issued by a Certificate Authority (CA).
- The SSL server authentication allows a user to confirm a server's identity by verifying that the server's certificate and public ID are valid and are issued by a CA.

The red horizontal bar spans most of the width of the slide, centered below the title.

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Confidentiality

Oracle WebLogic Server supports the SSL protocol to enable secure communication between the applications that are connected through the Web. By default, WebLogic Server is configured for one-way SSL authentication where the managed server is enabled with a digital certificate. Using the Administration Console, you can configure Oracle WebLogic Server for two-way SSL authentication where the client and server are both enabled with digital certificates to securely establish their identity.

To use SSL, you would require a private key, a digital certificate containing the matching public key, and a certificate signed by at least one trusted Certificate Authority (CA) to verify the data embedded in the digital certificate. For intermediate authorities, you may need to install the root-trusted CA's certificate.

SSL server authentication allows a user to confirm a server's identity, through an SSL-enabled client software using standard techniques of public key cryptography, to verify that a server's certificate and public ID are valid and have been issued by a CA that is listed in the client's list of trusted CAs. For example, when sending a credit card number, you may want to check the receiving server's identity.

Confidentiality (continued)

SSL client authentication allows a server to confirm a user's identity to verify that a client's certificate and public ID are valid and have been issued by a CA that is listed in the server's list of trusted CAs. For example, if a bank sends the account information to a customer, this check may be essential.

The SSL protocol includes two subprotocols: the SSL record protocol, which defines the format that is used to transmit data, and the SSL handshake protocol to exchange a series of messages between an SSL-enabled server and an SSL-enabled client when the SSL connection is established.

Credential Mapping

- The credential mapping process is used when application components access the authentication mechanism of an external system to obtain a set of credentials.
- The requesting application passes the subject as part of the call and information about the type of credentials required.
- Credentials are returned to the security framework, which is then passed to the requesting application component.
- The application component uses the credentials to access the external system.



ORACLE

Copyright © 2009, Oracle. All rights reserved.

Credential Mapping

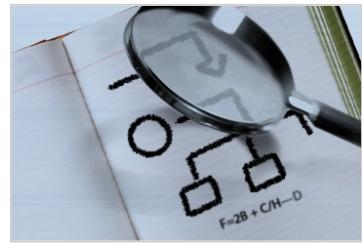
A credential map is a mapping of credentials used by Oracle WebLogic Server to credentials that are used in a legacy (or a remote) system to connect to a given resource in that system. Credential maps allow Oracle WebLogic Server to log in to a remote system on behalf of a subject that has already been authenticated.

A credential mapping provider of WLS can handle several different types of credentials, such as username/password, Kerberos tickets, and public key certificates. Credential mappings can be set in deployment descriptors or through the Administration Console.

You can configure multiple credential mapping providers in a security realm. The security framework makes a call to each credential mapping provider to determine whether it contains the type of credentials requested by the container. The framework accumulates and returns all the credentials as a list.

Road Map

- Security overview
- Users and groups
 - Security realms
 - Embedded LDAP
 - Configuring users, groups, and roles
- Protecting application resources

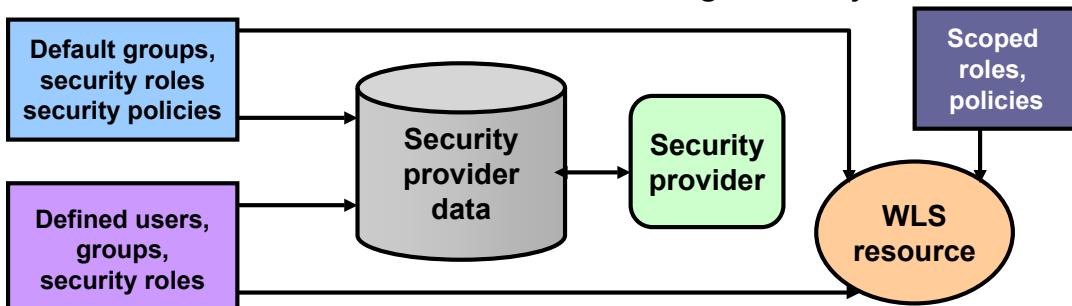


ORACLE

Copyright © 2009, Oracle. All rights reserved.

Security Realms

- A security realm is a collection of system resources and security service providers.
- A valid user must be authenticated by the authentication provider in the security realm.
- Only one security realm can be active at a given time.
- A single security policy can be used in any realm.
- Administration tasks include creating security realms.



ORACLE

Copyright © 2009, Oracle. All rights reserved.

Security Realms

A security realm is a mechanism for protecting Oracle WebLogic Server resources, such as authenticators, adjudicators, authorizers, auditors, role mappers, and credential mappers. Oracle WebLogic Server resources in a domain are protected under only one security realm and by a single security policy in that security realm. A user must be defined in a security realm in order to access any resources belonging to that realm. When a user attempts to access a particular Oracle WebLogic Server resource, Oracle WebLogic Server tries to authenticate the user and then authorize the user action by checking the access privileges that are assigned to the user in the relevant realm.

Security Model Options for Applications

| Security Model | Location of Users, Roles, and Policies | Security Checks Performed |
|---|---|--|
| Deployment Descriptor Only (Java EE standard) | Deployment descriptors: <ul style="list-style-type: none">• web.xml and weblogic.xml• ejb-jar.xml and weblogic-ejb-jar.xml | Only when clients request URLs or EJB methods that are protected by a policy in the deployment descriptor |
| Custom Roles | Role mappings from a role mapping provider that you configure for the security realm Policies are defined in the web.xml and ejb-jar.xml deployment descriptors. | Only when clients request URLs or EJB methods that are protected by a policy in the deployment descriptor. |
| Custom Roles and Policies | Role mappings and authorization from providers that you configure for the security realm | For all URLs and EJB methods in the application |
| Advanced | This model is fully flexible. You can import security data from deployment descriptors into the security provider databases to provide a baseline. | Configurable |

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Security Model Options for Applications

You choose a security model when you deploy each Web application or EJB, and your choice is immutable for the lifetime of the deployment. If you want to use a different model, you must delete and redeploy the Web application or EJB.

The Java EE platform already provides a standard model for securing Web applications and EJBs. In this standard model, you define role mappings and policies in the deployment descriptors of Web application or EJB.

Because this Java EE standard can be too inflexible for some environments, WebLogic Server offers a choice of other, more flexible models in addition to supporting the Java EE standard.

Security Model Options for Applications (continued)

Deployment Descriptor Only model

- This is the standard Java EE model and is therefore a widely known technique for adding declarative security to Web applications and EJBs.
- It uses only roles and policies defined by a developer in the Java EE deployment descriptor (DD) and the WebLogic Server DD.
- It requires the security administrator to verify that the security principals (groups or users) in the deployment descriptors exist and are mapped properly in the security realm.
- With this model, EJBs and URL patterns are not protected by roles and policies of a broader scope (such as a policy scoped to an entire Web application). If an EJB or URL pattern is not protected by a role or policy in the DD, then it is unprotected: anyone can access it.
- This model is appropriate if developers and security administrators can closely coordinate their work, both upon initial deployment of the Web application or EJB and upon subsequent redeployments.

Custom Roles model

- The model enables team members to focus on their areas of expertise. Web application and EJB developers need only to declare which URL patterns or EJB methods should be secured. Then the security administrator creates role mappings that fit within the existing hierarchy of roles and principals for a given realm.
- If a developer changes policies in a deployment descriptor, WebLogic Server recognizes the change as soon as you redeploy the Web application or EJB. If an administrator changes role mappings, the changes take effect immediately without requiring a redeployment.

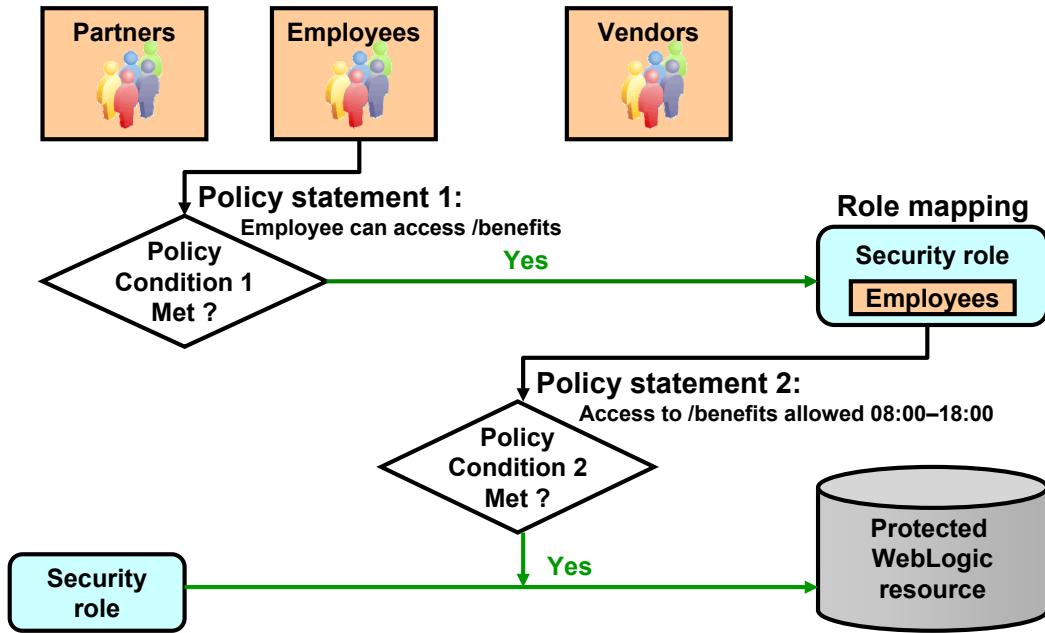
Custom Roles and Policies model

- This security model offers unified and dynamic security management. Instead of requiring developers to modify multiple deployment descriptors when organizational security requirements change, administrators can modify all security configurations from a centralized graphical user interface.
- Users, groups, security roles, and security policies can all be defined using the Administration Console. As a result, the process of making changes based on updated security requirements becomes more efficient.
- This model is appropriate if you require only that entire Web applications or EJBs be secured, but is less appropriate if you require fine-grained control of a large number of specific URL patterns or EJB methods.

Advanced model

- WebLogic Server provides this model primarily for backwards compatibility with releases prior to 9.0.

How WLS Resources Are Protected



ORACLE

Copyright © 2009, Oracle. All rights reserved.

How WLS Resources Are Protected

The following steps provide an overview of the process of granting access to a WLS resource:

- As an administrator, before creating security policies and roles, you can create users and groups and statically assign users to groups that represent organizational boundaries.
- Then create a security role based on your established business procedures. The security role consists of one or more conditions that specify the circumstances under which a particular user, group, or other role should be granted the security role.
- At run time, the WebLogic Security Service compares the groups against the role conditions to determine whether users in the group should be dynamically granted a security role. This process of comparing groups to roles is called role mapping.
- Then, you create a security policy based on your established business procedures. The security policy consists of one or more policy conditions that specify the circumstances under which a particular security role should be granted access to a WebLogic resource.
- At run time, the WebLogic Security Service uses the security policy to determine whether access to the protected WebLogic resource should be granted. Only users who are members of the group that is granted the security role can access the WebLogic resource.

Users and Groups

- Users are entities that use WLS, such as:
 - Application end users
 - Client applications
 - Other Oracle WebLogic Servers
- Groups are:
 - Logical sets of users
 - More efficient for managing a large number of users



ORACLE

Copyright © 2009, Oracle. All rights reserved.

Users and Groups

Users are entities that can be authenticated in a security realm. A user can be a person, such as an application end user, or a software entity, such as a client application, or other instances of WebLogic Server. As a result of authentication, a user is assigned an identity or principal. Each user is given a unique identity within the security realm.

Users may be placed into groups that are associated with security roles or be directly associated with security roles.

When users want to access WebLogic Server, they present proof material (such as a password or a digital certificate) to the authentication provider configured in the security realm. If WebLogic Server can verify the identity of the user based on that username and credential, WebLogic Server associates the principal assigned to the user with a thread that executes code on behalf of the user. Before the thread begins executing code, however, WebLogic Server checks the security policy of the WebLogic resource and the principal (that the user has been assigned) to make sure that the user has the required permissions to continue.

A person can be defined as both an individual user and a group member. Individual-access permissions override any group member-access permissions. Oracle WebLogic Server evaluates each user by first looking for a group and testing whether the user is a member of the group and then looking for the user in the list of defined users.

Configuring New Users

The screenshot shows the Oracle WebLogic Server Administration Console. On the left, there is a navigation pane with tabs: Configuration, Users and Groups, Roles and Policies, and Credentials. The 'Users and Groups' tab is selected. Within this tab, the 'Users' sub-tab is selected. A table lists users with columns for Name and Description. A 'New' button is highlighted with a cursor. On the right, a modal dialog box titled 'Create a New User' is open. It contains fields for Name (set to 'John Doe'), Description (set to 'User for Benefits application'), Provider (set to 'DefaultAuthenticator'), Password (a masked password), and Confirm Password (a masked password). At the bottom of the dialog are 'OK' and 'Cancel' buttons, with 'OK' being highlighted.

Configuring New Users

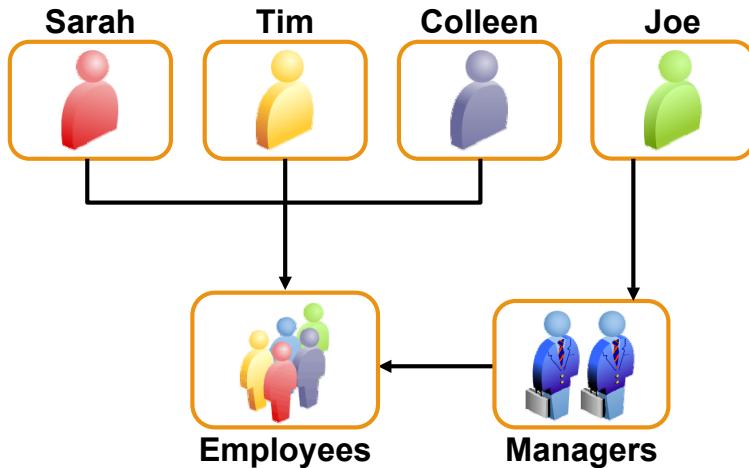
To configure a new user, perform the following steps:

1. Access Security Realms and select your security realm in the Realms Table on the Summary of Security Realms page.
2. Click the **Users and Groups** > **Users** tab for your realm. Click New in the Users table.
3. Enter the necessary details in the Create a New User dialog box and click OK.

Groups

WLS provides the flexibility to organize groups in various ways:

- Groups can contain users.
- Groups can contain other groups.



ORACLE

Copyright © 2009, Oracle. All rights reserved.

Groups

Groups can be organized in arbitrary ways, thereby providing greater flexibility. In this example, all the users (Sarah, Tim, Colleen, and Joe) are members of the Employees group. Joe is also a member of the Managers group. All Managers are also Employees.

Managing groups is more efficient than managing large numbers of users individually. For example, an administrator can specify permissions for 50 users at one time if those 50 users belong to the same group. Usually, group members have something in common. For example, a company may separate its sales staff into two groups: Sales Representatives and Sales Managers. This is because staff members have different levels of access to the Oracle WebLogic Server resources depending on their job descriptions.

Oracle WebLogic Server can be configured to assign users to groups. Each group shares a common set of permissions that govern its member users' access to resources. You can mix group names and usernames whenever a list of users is permitted.

Configuring New Groups

The screenshot shows the Oracle WebLogic Server Administration Console interface. On the left, there is a navigation tree with nodes like 'Configuration', 'Users and Groups', 'Roles and Policies', 'Credential Mappings', 'Providers', and 'Migration'. Under 'Users and Groups', the 'Groups' tab is selected. A sub-menu 'Customize this table' is open, showing options for 'Groups', 'New', and 'Delete'. A modal dialog box titled 'Create a New Group' is displayed over the main page. It contains fields for 'Name' (set to 'Supervisor'), 'Description' (set to 'Supervisors for benefits application'), and 'Provider' (set to 'DefaultAuthenticator'). At the bottom of the dialog are 'OK' and 'Cancel' buttons, with the 'OK' button being highlighted.

Settings for myrealm

Configuration **Users and Groups** Roles and Policies Credential Mappings Providers Migration

Users Groups

Customize this table

Groups

New Delete

Create a New Group

OK Cancel

What would you like to name your new Group?

* Name: Supervisor

How would you like to describe the new Group?

Description: Supervisors for benefits application

Please choose a provider for the group.

Provider: DefaultAuthenticator

OK Cancel

ORACLE

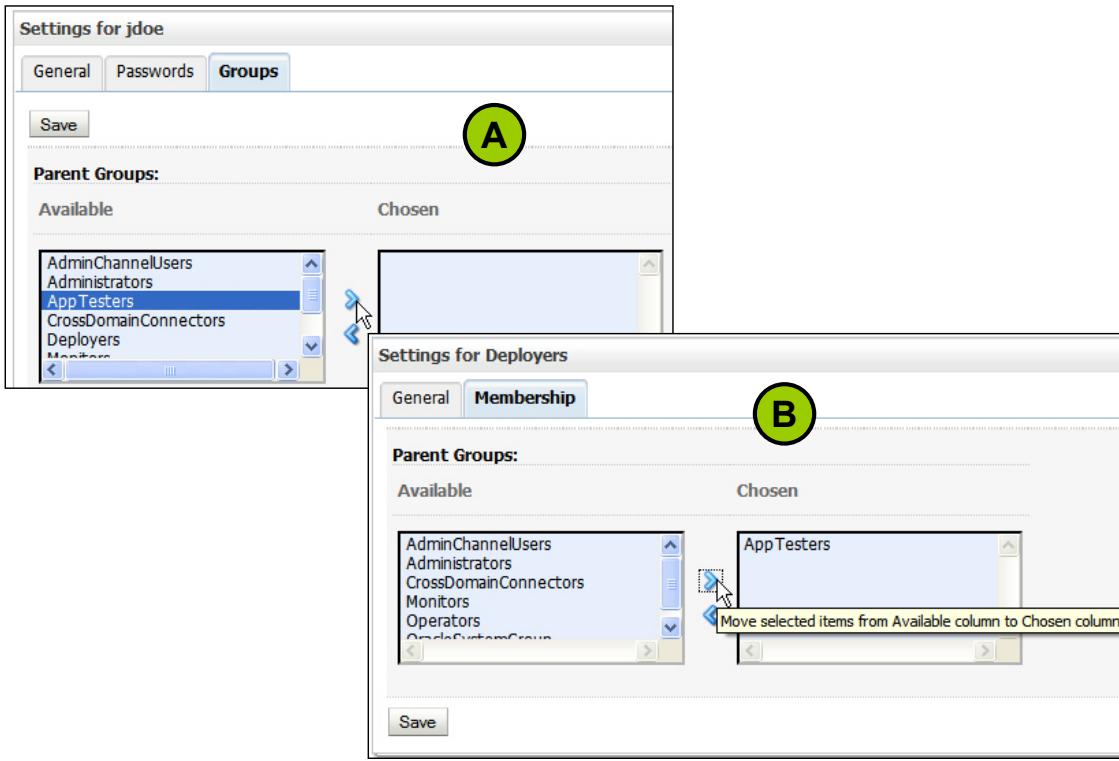
Copyright © 2009, Oracle. All rights reserved.

Configuring New Groups

To configure a new group, perform the following steps:

1. Access Security Realms and select your security realm in the Realms Table on the Summary of Security Realms page.
2. Click the **Users and Groups > Groups** tab for your realm. Click New in the Groups table.
3. Enter the necessary details in the Create a New Group dialog box and click OK.

Configuring Group Memberships



ORACLE

Copyright © 2009, Oracle. All rights reserved.

Configuring Group Memberships

Each group has two types of membership.

- You can configure a user to be a member of a group as follows:
 - Navigate to the Users subtab under the Users and Groups tab of the security realm.
 - Select the user for whom you want to configure the group membership.
 - Click the Groups tab on the “Settings for the *<user>*” page.
 - Select the group from the Available list and click > to move it to the Chosen list. Then click Save.
- You can configure a group to be a member of another group as follows:
 - Navigate to the Groups subtab under the Users and Groups tab of the security realm.
 - Select the Group you want to configure as a child of another group.
 - Click the Membership tab on the “Settings for the *<group>*” page.
 - Select the parent group from the Available list and click > to move it to the Chosen list. Then click Save.

Road Map

- Security overview
- Users and groups
- Roles and policies
 - Security roles
 - Security policies
 - Defining policies and roles
 - Protecting Web resources
 - Protecting other resources



ORACLE

Copyright © 2009, Oracle. All rights reserved.

Security Roles

- A role refers to a set of permissions granted to a user or group.
- A role differs from a group; a group has static membership, whereas a role is conditional.
- A user and group can be granted multiple roles.
- The two types of roles are global-scoped roles and resource-scoped roles.
- The global roles that are available by default are Admin, Operator, Deployer, Monitor, AppTester, and Anonymous.
- Roles defined in deployment descriptors can be inherited.
- You can manage role definitions and assignments without editing deployment descriptors or redeploying the application.

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Security Roles

A security role is a privilege granted to users or groups based on specific conditions. Similar to groups, security roles allow you to restrict access to WebLogic resources for several users simultaneously. However, unlike groups, security roles:

- Are evaluated and granted to users or groups dynamically, based on conditions such as username, group membership, or the time of day
- Can be scoped to specific WebLogic resources within an application in a WebLogic Server domain (unlike groups, which are always scoped to an entire WebLogic Server domain)

Granting a security role to a user or a group confers the defined access privileges to that user or group as long as the user or group is “in” the security role. Multiple users or groups can be granted a single security role. A role definition is specific to a security realm.

A role can be defined as global or scoped.

WLS defines a set of default global roles for protecting all the WebLogic resources in a domain. A scoped role protects a specific resource, such as a method of an EJB or a branch of the JNDI tree. Most roles are scoped.

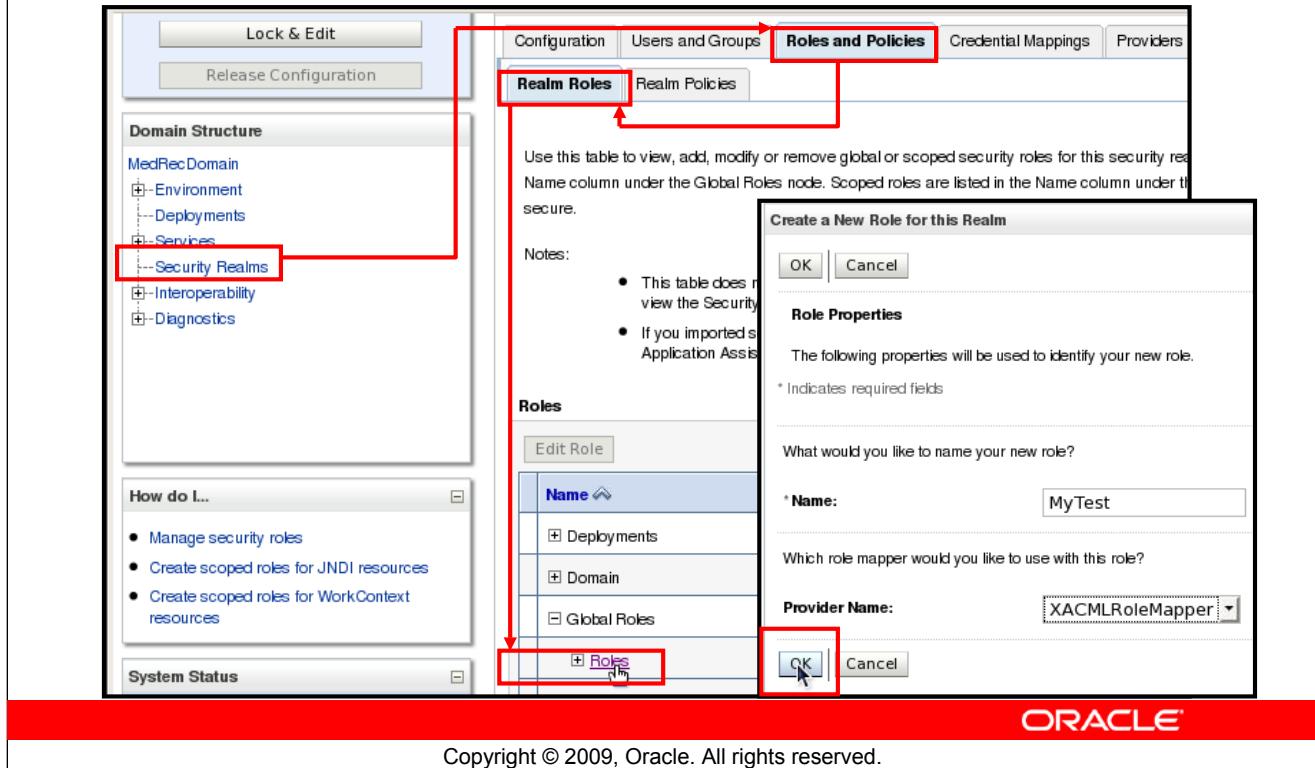
Note that by default no security role is enforced and therefore all the resources can be accessed by any user.

Security Roles (continued)

Default Global Roles Provided by Oracle WebLogic Server

- **Admin** can display and modify all resource attributes and perform start and stop operations. By default, users in the Administrators group are granted the Admin role. You can change this association or add other group associations.
- **Operator** can display all resource attributes. Users can start, suspend, resume, and stop resources. By default, users in the Operators group are granted the Operator role. You can change this association or other group associations.
- **Deployer** can display all resource attributes. Users can deploy applications, EJBs, and other deployable modules. By default, users in the Deployers group are granted the Deployer role. You can change this association or other group associations.
- **Monitor** can display all resource attributes. Users can modify the resource attributes and operations that are not restricted to the other roles. By default, users in the Monitors group are granted this role. You can change this association or other group associations.
- **AppTester** can test the versions of applications that are deployed to Administration mode.
- **Anonymous:** All users are granted this global role.

Configuring the Global Security Role



Configuring the Global Security Role

To create a global security role:

- In the left pane of the Administration Console, select Security Realms. On the Summary of Security Realms page, select the name of the realm in which you want to create the role (for example, myrealm).
- On the Settings page, click the Roles and Policies tab. Then click the Roles subtab. The Roles page organizes all the domain's resources and corresponding roles in a hierarchical tree control.
- In the Roles table, in the Name column, expand the Global Roles node. In the Name column, select the name of the Roles node.
- In the Global Roles table, click New. On the Create a New Role for this Realm page, enter the name of the global role in the Name field.
- If you have more than one role mapper configured for the realm, from the Provider Name list, select the role mapper you want to use for this role. Click OK to save your changes.
- In the Global Roles table, select the role. In the Role Conditions section, click Add Conditions.
- On the Choose a Predicate page, in the Predicate List, select a condition.
- The next steps depend on the condition that you chose. After you complete the conditions, click Finish.

Security Policies

- Security policies implement parameterized authorization.
- Security policies comprise rules and conditions.
- Users and groups that adhere to the security policy are granted access to resources protected by the policy.
- Security policies follow a hierarchy. The policy of a narrower scope overrides that of a broader scope.
- When you install Oracle WebLogic Server, some default root-level policies are provided.

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Security Policies

Oracle WebLogic Server provides security policies and roles as two mechanisms that are used together to control access to or protect resources. The security realm that Oracle WebLogic Server provides stores policies in the embedded LDAP server.

You can create a root-level policy that applies to all instances of a specific resource type. For example, you can define a root-level policy that applies to all JMS resources in your domain.

You can also create a policy that applies to a specific resource instance. If the instance contains other resources, the policy will apply to the included resource as well. For example, you can create a policy for an entire Enterprise Archive (EAR), an EJB JAR containing multiple EJBs, a particular EJB within that JAR, or a single method within that EJB.

The policy of a narrower scope overrides the policy of a broader scope. For example, if you create a security policy for an EAR and a policy for an EJB that is in the EAR, the EJB will be protected by its own policy and will ignore the policy for the EAR.

Policy Conditions

- Policy conditions are the essential components of a policy.
- The WebLogic Server authorization provides three kinds of built-in policy conditions in the Administration Console:
 - Basic policy conditions
 - Date and Time policy conditions
 - Context Element policy conditions

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Policy Conditions

To determine who can access a resource, a policy contains one or more conditions. The most basic policy simply contains the name of a security role or a principal. For example, a basic policy might simply name the global role Admin. At run time, the WebLogic Service interprets this policy as “allow access if the user is in the Admin role.”

You can create more complex conditions and combine them using the logical operators AND and OR (which is an inclusive OR). You can also negate any condition, which would prohibit access under the specified condition.

WebLogic Server by default provides three kinds of conditions:

- **Basic:** This can be used to allow or deny access to every one or specific users, groups or roles.
- **Date and Time:** When you use any of the date and time conditions, the security policy grants access to all users for the date or time you specify, unless you further restrict the users by adding one of the other conditions.
- **Context Element:** You can use the context element conditions to create security policies based on the value of HTTP Servlet Request attributes, HTTP Session attributes, and EJB method parameters. WebLogic Server retrieves this information from the ContextHandler object and allows you to define policy conditions based on the values. When using any of these conditions, it is your responsibility to ensure that the attribute or parameter/value pairs apply to the context in which you are using them.

Protecting Web Applications

To protect a Web application with declarative security, perform the following steps:

1. Define the roles that should access the protected resources.
2. Determine the Web application resources that must be protected.
3. Map the protected resources to roles that should access them.
4. Map roles to users or groups in the WLS security realm.
5. Set up an authentication mechanism.

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Protecting Web Applications

If you are using the DD Only or Custom Roles security model for the deployment of a Web application, you cannot use the Administration Console to modify its security policies. You have to define your security details using the deployment descriptors.

Specifying Protected Web Resources

Protection for Web resources are defined based on URL patterns.

Example URL patterns:

| URL Pattern | Role Name |
|--------------|--|
| /* | Some role name (for example, director) |
| /*.jsp | employee |
| /EastCoast/* | east-coaster |

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Specifying Protected Web Resources

URL patterns provide a flexible way to define security for a single resource or a group of resources.

- In the Administration Console, navigate to your domain > Deployments and click the Web application in the Deployments table.
- On the Settings for *<the application>* page, navigate to Security > Application Scope > URL Patterns (subtab). Click New in the Stand-Alone Web Application URL Pattern Scoped Roles table.
- Specify the URL pattern (for example, /managers/*) and then specify the name: director. This is the name that the application has been configured to use in securing its resources. Leave the provider name as XACMLRoleMapper and click OK.
- In the Stand-Alone Web Application URL Pattern Scoped Roles table, you should now see the URL pattern created.
- Click URL Pattern and click Add Conditions. Choose the appropriate predicate from the Predicate List and click Next. On the next page, enter the appropriate conditions and values and then click Add.
- Click Finish. On the next page, click Save.

Defining Policies and Roles for Other Resources

You can define roles and policies for other resources, such as JDBC and JMS.

The screenshot shows the 'Settings for testSample' page with the 'Security' tab selected. A green circle labeled '1' highlights the 'testSample' data source in the 'Data Sources' table. A green circle labeled '2' highlights the 'Policies' tab in the top navigation bar. A green circle labeled '3' highlights the 'Add Conditions' button in the 'Policy Conditions' section. A green circle labeled '4' highlights the 'Choose a Predicate' section. A green circle labeled '5' highlights the 'Edit Arguments' section where time ranges are specified.

Copyright © 2009, Oracle. All rights reserved.

Defining Policies and Roles for Other Resources

Defining roles and policies for other resources is similar to defining roles and policies for the Web resources. For all of them, you need to define policy conditions and policy statements. For some resources, you can also define methods or actions that are allowed for that resource. For instance, for servers, you may define restrictions on actions such as boot, shutdown, lock, and unlock.

The following steps illustrate how you can define a policy for the testSample JDBC data source:

1. In the Administration Console, navigate to Services > JDBC > Data Sources. In the Data Sources table, click the data source for which you want to define policy.
2. On the Settings for <resource> (testSample) page, select Security > Policies.
3. Click Add Conditions in the Policy Conditions section.
4. Select the appropriate choice from the Predicate List and click Next.
5. Specify the appropriate conditions and click Finish.

Embedded LDAP Server

- In WLS, users, groups, and authorization information are stored in an embedded LDAP server.
- Several properties can be set to manage the LDAP server, including:
 - Credentials
 - Backup settings
 - Cache settings
 - Replication settings

The red bar spans the width of the slide content area.

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Embedded LDAP Server

The embedded LDAP server is used as a storage mechanism with the Oracle WebLogic Server authentication, authorization, role mapping, and credential mapping providers.

Information from these providers is stored and updated in the administration server and replicated to all the managed servers in the domain. The read operations performed by the Oracle WebLogic Server security providers (when running on a managed server) access the local replicated embedded LDAP server. The write operations access the master embedded LDAP server on the administration server and any updates are replicated to all the managed servers in the domain. If the administration server is not running, operations by the Oracle WebLogic Server security providers that write to the embedded LDAP server (for example, adding new users, groups, or roles, or adding resources) are not possible.

Configuring an Embedded LDAP

The screenshot shows the 'Settings for MedRecDomain' window with the 'Security' tab selected. Below it, the 'Embedded LDAP' sub-tab is also selected. The configuration interface is divided into two main sections:

- Left Panel (Credential Configuration):**
 - Credential:** A password field containing '.....'.
 - Confirm Credential:** A password field containing '.....'.
 - Backup Hour:** A dropdown menu set to '23'.
 - Backup Minute:** A dropdown menu set to '5'.
 - Backup Copies:** A dropdown menu set to '7'.
- Right Panel (Advanced Cache Configuration):**
 - Cache Enabled:** A checked checkbox.
 - Cache Size:** A text input field set to '32'.
 - Cache TTL:** A text input field set to '60'.
 - Refresh Replica At Startup:** An unchecked checkbox.
 - Master First:** An unchecked checkbox.
 - Timeout:** A text input field set to '0'.
 - Anonymous Bind Allowed:** An unchecked checkbox.

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Configuring an Embedded LDAP

- **Credential:** The credential (usually password) that is used to connect to the embedded LDAP server. This password is encrypted. The default is null.
- **Backup Hour:** The hour at which to back up the embedded LDAP server. Minimum is 0, Maximum is 23, and Default is 23.
- **Backup Minute:** The minute at which to back up the embedded LDAP server. This attribute is used with the `BackupHour` attribute to determine the time at which the embedded LDAP server is backed up. Minimum is 0, Maximum is 59, and Default is 05.
- **Backup Copies:** The number of backup copies of the embedded LDAP server. Minimum is 0, Maximum is 65534, and Default is 7.
- **Cache Enabled:** Whether or not a cache is used for the embedded LDAP server. The default is True.
- **Cache Size:** The size of the cache (in KB) that is used with the embedded LDAP server. Minimum is 0 and Default is 32.
- **Cache TTL:** The time-to-live (TTL) of the cache in seconds. Minimum is 0 and Maximum is 60.

Configuring an Embedded LDAP (continued)

- **Refresh Replica At Startup:** Whether or not a managed server should refresh all replicated data at boot time. This is useful if you made a large number of changes when the managed server was not active and you want to download the entire replica instead of having the administration server push each change to the managed server. The default is false.
- **Master First:** The connections to the master LDAP server should always be made instead of connections to the local replicated LDAP server. The default is false.

Configuring Authentication

Configure how a Web application determines the security credentials of users:

- **BASIC**: The Web browser displays a dialog box.
- **FORM**: Use a custom HTML form.
- **CLIENT-CERT**: Request a client certificate.

Configure the authentication using the `<login-config>` element:

```
:<login-config>
    <auth-method>BASIC, FORM, or CLIENT-CERT</auth-method>
    <form-login-config>
        <form-login-page>login.jsp</form-login-page>
        <form-error-page>badLogin.jsp</form-error-page>
    </form-login-config>
</login-config>
```

Copyright © 2009, Oracle. All rights reserved.

Configuring Authentication

Configure how users will be authenticated in your Web application using the `<login-config>` element. J2EE provides three types of authentication:

- **BASIC**: A Web browser is used to display a dialog box with fields for a username and password.
- **FORM**: A specified HTML page, JSP, or servlet is used to display an HTML form with the username and password text fields. The generated form must conform to a set of specifications. Use the `<form-login-config>` element to specify the resource that contains the form. The `<form-error-page>` element defines the JSP, servlet, or HTML file to display if the user's credentials are invalid.
- **CLIENT-CERT**: WebLogic Server may receive digital certificates as part of Web Services requests, two-way SSL, or other secure interactions. To validate these certificates, WebLogic Server includes a Certificate Lookup and Validation (CLV) framework, whose function is to look up and validate X.509 certificate chains. The key elements of the CLV framework are CertPathBuilder and CertPathValidators. The CLV framework requires one and only one active CertPathBuilder which, given a reference to a certificate chain, finds the chain and validates it, and zero or more CertPathValidators which, given a certificate chain, validates it.

Authentication Examples

BASIC authentication



FORM-based authentication



ORACLE

Copyright © 2009, Oracle. All rights reserved.

Authentication Examples

Oracle WebLogic Server supports three types of authentication for Web browsers:

- BASIC
- FORM

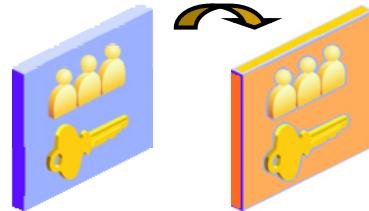
With BASIC authentication, the Web browser displays a dialog box in response to a WebLogic resource request. The login screen prompts the user for a username and password. The slide shows a typical login screen.

When using FORM authentication with Web applications, you provide a custom login screen that the Web browser displays in response to a Web application resource request and an error screen that displays if the login fails. The login screen can be generated using an HTML page, JSP, or servlet. The benefit of FORM-based login is that you have complete control over these screens. You can design them to meet the requirements of your application or enterprise policy or guideline.

The login screen prompts the user for a username and password.

Migrating Security Data

- You can export users and groups, security policies, security roles, or credential maps between security realms or domains.
- It is useful, for example, in transitioning from development to QA to production.
- You can use migration constraints (key/value pairs) to specify the export/import options.
- Currently, the system supports migrating only security data between the WLS security providers.



ORACLE

Copyright © 2009, Oracle. All rights reserved.

Migrating Security Data

Oracle WebLogic Server security realms persist different kinds of security data—for example, users and groups (for the WebLogic authentication provider), security policies (for the XACML authorization provider), security roles (for the XACML role mapping provider), and credential maps (for the WebLogic credential mapping provider).

When you configure a new security realm or a new security provider, you may prefer to use the security data from your existing realm or provider, rather than re-create all the users, groups, policies, roles, and credential maps. Several WebLogic security providers support security data migration. This means that you can export security data from one security realm and import it into a new security realm. You can migrate security data for each security provider individually or migrate security data for all the WebLogic security providers simultaneously (that is, security data for an entire security realm).

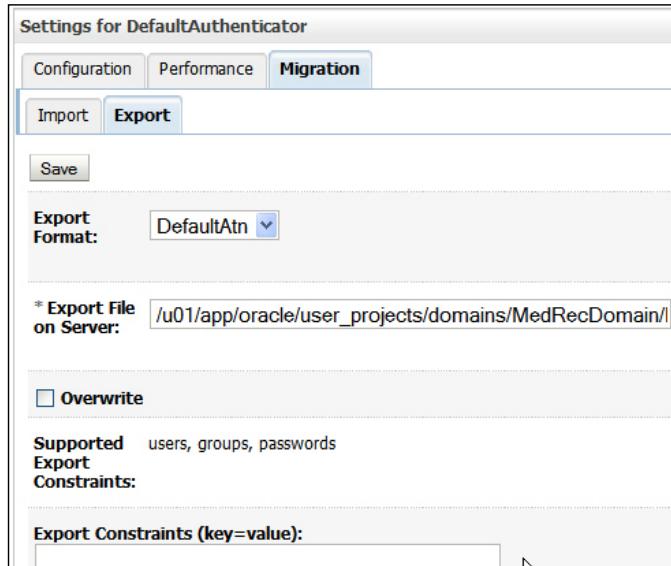
Note that you can migrate security data from one provider to another only if the providers use the same data format. You migrate security data through the WebLogic Administration Console or by using the WebLogic Scripting Tool (WLST).

Migrating Security Data (continued)

Migrating security data may be helpful when you:

- Transition from development to production mode
- Copy production mode security configurations to security realms in the new Oracle WebLogic Server domains
- Move data from one security realm to a new security realm in the same Oracle WebLogic Server domain, where one or more of the default WebLogic security providers are replaced with new security providers

Exporting the WLS Default Authenticator Provider



ORACLE

Copyright © 2009, Oracle. All rights reserved.

Exporting the WLS Default Authenticator Provider

To export security data from a security provider to a file, perform the following steps:

1. In the left pane, select **Security Realms** and then select the name of the realm that you are configuring (for example, `myrealm`).
2. Select the type of provider from which you want to export the security data (for example, authentication).
3. Select the security provider from which you want to export the security data.
4. Select **Migration > Export**.
5. Specify the directory and file name in which to export the security data in the **Export File on Server** field. The directory must exist.
Note: The directory and file into which you export the security data should be carefully protected with operating system security because they contain secure information about your deployment.
6. Optionally, define a specific set of security data to be exported in the **Export Constraints (key=value)** box.
7. Click **Save**.
Note: After the data is exported from the security provider, it can be imported at any time.

Importing into a Different Domain

Settings for DefaultAuthenticator

Configuration Performance Migration

Import Export

Save

Import Format: DefaultAtn

* Import File on Server: /u01/app/oracle/user_projects/domains/MedRecDomain/

Supported Import Constraints: None

Import Constraints (key=value):

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Importing into a Different Domain

You can export the security data from a security provider into a file and then import the data into a different security provider. As an alternative, you can export the security data from all the security providers in a realm and then import that data into another security realm. To import security data into a security provider, perform the following steps:

1. In the left pane of the Administration Console, select **Security Realms**.
2. Select the name of the security realm into which the security data is to be imported (for example, **myrealm**).
3. Select Providers and then the type of provider into which the security data is to be imported (for example, **Providers > Authentication**).
4. Select the security provider into which the security data is to be imported and select **Migration > Import**.
5. Specify the directory and file name of the file that contains the exported security data in the **Import File on Server** field.
6. You can restrict the imported security parameters by specifying the Import Constraints.
7. Click Save.

Summary

In this lesson, you should have learned how to:

- Use the WLS security architecture
- Configure users, groups, and roles
- Configure roles
- Configure policies
- Configure protection for:
 - Web application resources
 - EJBs
- Configure security realms



Copyright © 2009, Oracle. All rights reserved.

Practice 18: Overview Configuring Security for WLS Resources

This practice covers the following topics:

- Creating new users using the Administration Console
- Creating groups of employees and managers
- Assigning groups to users
- Configuring groups-to-role mapping
- Defining resources that are protected by the security you have configured
- Verifying that the security protection that you enabled is working

ORACLE

Copyright © 2009, Oracle. All rights reserved.

19

Protecting Against Attacks

ORACLE®

Copyright © 2009, Oracle. All rights reserved.

Objectives

After completing this lesson, you should be able to do the following:

- Describe the process of configuring Secure Sockets Layer (SSL)
- Use the `keytool` utility to configure keys and obtain digital certificates
- Configure SSL for the WLS server
- Configure countermeasures for some Web-based attacks such as:
 - Man in the middle
 - Denial of service
 - Large buffer
 - Connection starvation



Copyright © 2009, Oracle. All rights reserved.

Road Map

- Protecting the transport layer
 - Secure Sockets Layer (SSL)
 - keytool
 - Certificates
 - Configuring SSL
- Protecting against attacks



ORACLE®

Copyright © 2009, Oracle. All rights reserved.

What Is SSL?

Secure Sockets Layer (SSL) is a protocol that enables:

- Connection security through encryption
- A server to authenticate to a client
- A client to authenticate to a server (optional)
- Data integrity such that the data that flows between a client and server is protected from tampering by a third party



Copyright © 2009, Oracle. All rights reserved.

What Is SSL?

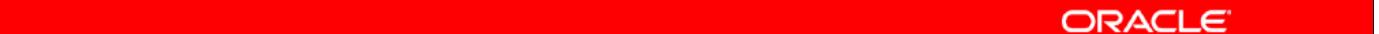
The SSL protocol offers security to applications that are connected through a network. Specifically, the SSL protocol provides the following:

- A mechanism that the applications can use to authenticate each other's identity
- Encryption of the data that is exchanged by the applications
- Data integrity, whereby the data that flows between a client and a server is protected from tampering by a third party

When the SSL protocol is used, the target always authenticates itself to the initiator. Optionally, if the target requests it, the initiator can authenticate itself to the target. Encryption makes the data that is transmitted over the network intelligible only to the intended recipient. An SSL connection begins with a handshake during which time the applications exchange digital certificates, agree on the encryption algorithms to be used, and generate the encryption keys to be used for the remainder of the session.

Trust and Identity

- SSL and keystore are configured independently.
- For the purpose of backward compatibility, this release of Oracle WebLogic Server supports private keys and a trusted WebLogic Keystore provider.
- Identity:
 - Private key and digital certificate (can now be looked up directly from the keystore, not necessarily as a stand-alone file outside the keystore)
- Trust:
 - Certificates of trusted certificate authorities

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Trust and Identity

For demonstration purposes, you can use the following out of the box:

<WL_HOME>\server\lib\DemoIdentity.jks as identity, and
<WL_HOME>\server\lib\DemoTrust.jks or
<JAVA_HOME>\jre\lib\security\cacerts for trust.

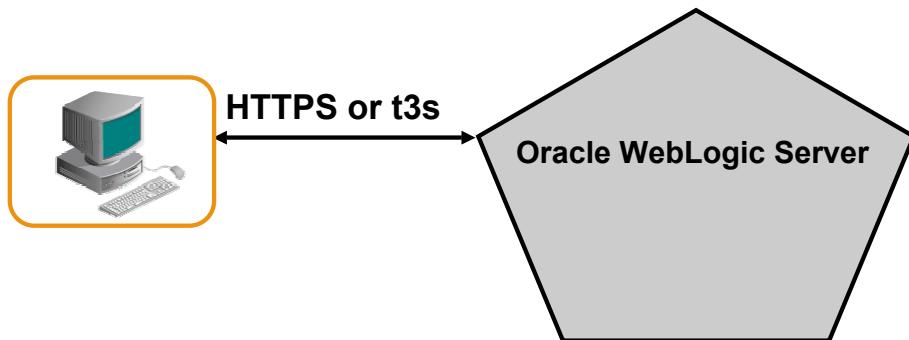
By default the Node Manager and server SSL use DemoTrust.jks for trust.

To create identity and trust for a server, perform the following steps:

1. Obtain digital certificates, private keys, and trusted CA certificates from the CertGen utility, Sun Microsystems' keytool utility, or a reputable vendor such as Entrust or VeriSign. You can also use the digital certificates, private keys, and trusted CA certificates provided by the Oracle WebLogic Server kit. The digital certificates, private keys, and trusted CA certificates in the demonstration should be used only in a development environment.
2. Store the private keys, digital certificates, and trusted CA certificates. Private keys and trusted CA certificates are stored in a keystore.
Note: The preferred keystore format is Java KeyStore (JKS). Oracle WebLogic Server supports private keys and trusted CA certificates that are stored in files or in the WebLogic Keystore provider only for the purpose of backward compatibility.
3. Configure the identity and trust keystores for Oracle WebLogic Server in the Oracle WebLogic Server Administration Console.

Using an SSL Connection

- WLS uses SSL to secure HTTP and t3 communication.
- To use SSL, clients access WLS via the HTTPS or t3s protocols.
 - `https://localhost:7002/orderStock`
 - `t3s://localhost:7002/useCreditCard`



Copyright © 2009, Oracle. All rights reserved.

Using an SSL Connection

The use of SSL is signified in the protocol scheme of the URL to specify the location of Oracle WebLogic Server. SSL communications between Web browsers and Oracle WebLogic Server are encapsulated in HTTPS packets for transport. For example:

`https://myserver.com:7002/mypage.html`

Oracle WebLogic Server supports HTTPS with Web browsers that support SSL version 3. Java clients connect to Oracle WebLogic Server with the SSL protocol tunnel over Oracle's multiplexed t3 protocol. For example:

`t3s://myserver.com:7002`

Java clients running in Oracle WebLogic Server can also establish either t3s connections to other Oracle WebLogic Servers, or HTTPS connections to other servers that support the SSL protocol, such as Web servers or secure proxy servers. Browsers connect securely to Oracle WebLogic Server by specifying the appropriate protocol (that is, HTTPS) in the requested URL, whereas Java clients have a variety of options available to them when setting up secure connections. Java clients can use the SSL libraries in Oracle WebLogic Server to provide the SSL socket or, alternatively, they can use an SSL provider such as Sun Microsystems' Java Secure Socket Extension (JSSE) as the SSL socket.

Using an SSL Connection (continued)

When using Oracle WebLogic Server's SSL libraries, the client can create an HTTPS connection directly by using WLS-specific classes, such as `weblogic.net.http.HttpsURLConnection`. This connection can then be used to send and receive secure data as with any other `java.net.HttpURLConnection`.

Clients can also use JNDI to set up an SSL connection (for example, to an EJB). This can be done by specifying a t3s connection within `PROVIDER_URL` and "strong" as the `SECURITY_AUTHENTICATION` type when populating the Hashtable object that is used to create the JNDI `InitialContext`.

Finally, clients can use another SSL provider's implementation to set up a secure connection with WLS. Sun Microsystems' JSSE implementation is a popular choice. JSSE has been integrated into the Java 2 SDK, Standard Edition (J2SDK), v 1.4. It is a collection of Java packages that allow for secure Internet communications. It is a Java implementation of the SSL and Transport Layer Security (TLS) protocols that allow for encryption, authentication (both server and client), and message integrity. After the client imports the proper packages and initializes the JSSE service, it uses the standard `java.net.HttpURLConnection` to create a secure connection.

Enabling Secure Communication

- With SSL, data is encrypted using a negotiated symmetric session key.
- A public key algorithm is used to negotiate the symmetric session key.
- In SSL, digital certificates are used to provide a trusted public key.



ORACLE

Copyright © 2009, Oracle. All rights reserved.

Enabling Secure Communication

Under normal, non-Internet circumstances, data is sent between two parties. Each party has the same key and can decipher the data. Such situations in which both parties use the same key to encrypt and decrypt the data are termed *symmetric key encryption*. The problem with symmetric key encryption is that anyone can potentially see anything that is transmitted over the Internet by intercepting its key as it is being transferred.

When you use public key/private key encryption, the public key is freely available and can be transferred across the Internet. Anyone can use the public key. Data is encrypted with the public key, but can be decrypted only with the private key, which is held privately in secure storage. Though the two keys are mathematically linked, it is statistically impossible to generate the private key programmatically, thus ensuring data security.

Typically, anyone who wants to send an encrypted message obtains a digital certificate from a trusted source known as a *Certificate Authority* or CA. The CA issues a digital certificate containing the applicant's public key and identification information. The digital certificate is then encrypted by the CA whose own public key is publicly available. The receiver of the message uses the CA's public key to decode the digital certificate attached to the message, verifies it, and then obtains the sender's public key and identification information that is held within the certificate. With this information, the recipient can send an encrypted reply, which only the originator can decrypt.

Enabling Secure Communication (continued)

Using the sender's public and private keys, a symmetric key is established between the communicating parties and eventually secure communication is achieved using a symmetric algorithm. The symmetric key is valid only for the duration of the connection, thus making symmetric key guessing very difficult.

Oracle WebLogic Server SSL Requirements

To enable Oracle WebLogic Server SSL, you must perform the following steps:

1. Obtain an appropriate digital certificate.
2. Install the certificate.
3. Configure SSL properties.
4. Configure two-way authentication (if desired).
 - SSL impacts performance.



ORACLE®

Copyright © 2009, Oracle. All rights reserved.

Oracle WebLogic Server SSL Requirements

There are a number of steps to configure Oracle WebLogic Server to use SSL. You must first obtain a valid certificate from a CA such as VeriSign, Inc. You must then install the certificate as well as the certificates of one or more certificate authorities that you trust. In addition, you can configure Oracle WebLogic Server to support mutual authentication by adding several additional property entries. These steps are covered in detail in the following slides. It is important, however, to remember that enabling security has a performance penalty. Packets need to be encrypted and tunneled out over the network. Also CPU cycles are expended for encryption and decryption. However, when security is required, the performance penalty is usually worth it.

keytool Utility

- keytool is a standard J2SE SDK utility for managing:
 - The generation of private keys and the corresponding digital certificates
 - Keystores (databases) of private keys and the associated certificates
- The keytool utility can display certificate and keystore contents.
- Specify an algorithm different from DSA when generating digital keys using keytool.



ORACLE

Copyright © 2009, Oracle. All rights reserved.

keytool Utility

The Sun Microsystems' keytool utility can also be used to generate a private key, a self-signed digital certificate for Oracle WebLogic Server, and a Certificate Signing Request (CSR). Submit the CSR to a certificate authority to obtain a digital certificate for Oracle WebLogic Server.

You can use the keytool utility to:

- Update the self-signed digital certificate with a new digital certificate
- Obtain trust and identity when using Oracle WebLogic Server in a production environment

For more information about Sun's keytool utility, see the keytool Key and Certificate Management Tool description at <http://java.sun.com/j2se/1.5.0/docs/tooldocs/windows/keytool.html>.

Note: When you use the keytool utility, specify an algorithm different from the default Digital Signature Algorithm (DSA) such as RSA because Oracle WebLogic Server does not support DSA.

Obtaining a Digital Certificate: keytool Examples

Generate a new self-signed digital certificate:

```
keytool -genkey -alias dwkey -keyalg RSA -keysize 512
        -keystore dw_identity.jks
```

Generate a Certificate Signing Request:

```
keytool -certreq -v -alias dwkey -file
        dw_cert_request.pem
        -keypass dwkeypass -keystore dw_identity.jks
        -storepass dwstorepass
```

Import a signed certificate reply from a CA:

```
keytool -import -alias dwkey -file dw_cert_reply.pem
        -keypass dwkeypass -keystore dw_identity.jks
        -storepass dwstorepass
```

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Obtaining a Digital Certificate: keytool Examples

In the given form of the command, keytool -genkey prompts for the remaining information that it requires (for example, the X.500 Distinguished Name). You can specify all the required information on the command line. For example:

```
keytool -genkey -v -alias dwkey -keyalg RSA -keysize 512 -keypass
dwkeypass -validity 365 -keystore dw_identity.jks -storepass
dwstorepass
```

- **genkey:** Generates a public key and an associated private key and wraps the public key into a self-signed certificate, which is stored as a single-element certificate chain. This certificate chain and the private key are stored in a new keystore entry that is identified by an alias.
- **alias:** Enables access to all keystore key and trusted certificate entries. A unique alias is specified when you add an entity to the keystore using the genkey or import command to add a certificate or certificate chain to the list of trusted certificates.
- **keyalg:** Specifies the algorithm to be used to generate the key pair
- **keysize:** Specifies the size of each key to be generated
- **sigalg:** Specifies the algorithm that should be used to sign the self-signed certificate; this algorithm must be compatible with keyalg.

Obtaining a Digital Certificate: `keytool` Examples (continued)

Certificates from CAs are not always completely compatible. Most of the major CAs allow you to specify the server vendor to ensure compatibility. For other CAs, specify either the X.509 or PKCS#7 format for the certificate that you receive in response to a CSR that you submit. The JDK's `keytool` utility can import X.509 v1, v2, and v3 certificates, as well as the PKCS#7 formatted certificate chains into a keystore for use by WLS.

Configuring Keystores

The screenshot shows two panels of the Oracle WebLogic Server Administration Console.

Left Panel (Keystores Configuration):

- Header tabs: General, Cluster, Services, **Keystores**, SSL, Federation Services.
- Buttons: Overload, Health Monitoring, Server Start.
- Save button.
- Description: Keystores ensure the secure storage and management of private keys and lets you view and define various keystore configurations. These settings help protect transmissions.
- Keystores dropdown: Demo Identity and Demo Trust.
- Identity:**
 - Demo Identity Keystore: /u01/app/oracle/product/fmw/11.1.0/wlservr_10.3/server/lib/Demoidentity.jks
 - Type: jks
 - Passphrase: [REDACTED]
- Trust:**
 - Demo Trust Keystore: /u01/app/oracle/product/fmw/11.1.0/wlservr_10.3/server/lib/DemoTrust.jks
 - Type: jks
 - Java Standard Trust Keystore: /u01/app/oracle/product/fmw/11.1.0/jrockit_160_05_R27.6.2-20/jre/lib/security/cacerts
 - Type: jks
 - Java Standard Trust Keystore: [REDACTED]
 - Passphrase: [REDACTED]
 - Confirm Java Standard Trust: [REDACTED]

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Configuring Keystores

Keystores ensure the secure storage and management of private keys and trusted CAs. WebLogic Server is configured with a default identity keystore (`DemoIdentity.jks`) and a default trust keystore (`DemoTrust.jks`). In addition, WebLogic Server trusts the CA certificates in the `JDK cacerts` file. This default keystore configuration is appropriate for testing and development purposes. However, these keystores should not be used in a production environment.

After you configure identity and trust keystores for a WebLogic Server instance, you can configure its SSL attributes. These attributes include information about the identity and trust location for particular server instances.

For purposes of backward compatibility, with WebLogic Server, you can store private keys and trusted certificates authorities in files or in the WebLogic Keystore provider. If you use either of these mechanisms for identity and trust, select the Files or Keystore Providers (Deprecated) option on the Configuration: SSL page.

Configuring SSL for an Oracle WebLogic Server

The screenshot shows the Oracle WebLogic Server Administration Console interface. On the left, the 'Change Center' section displays a 'Lock & Edit' button being clicked. Below it, the 'Domain Structure' shows a 'MedRecDomain' node with sub-nodes: Environment, Servers, Clusters, and Virtual Hosts. In the center, the 'Settings for MedRecSrv1' page is open under the 'Configuration' tab. The 'SSL' tab is selected. On the right, a detailed configuration dialog box is open, titled 'Identity and Trust'. It shows 'Keystores' selected in the dropdown. Under the 'Identity' section, 'Private Key' is set to 'from Demo Identity Keystore' and 'Location' is 'DemoIdentity'. Under the 'Trust' section, 'Trusted' is set to 'from Demo Trust Keystore and Java Standard Trust', 'Certificate' is 'Keystore', and 'Authorities' is listed. A 'Save' button at the bottom of the dialog is also being clicked.

Configuring SSL for an Oracle WebLogic Server

You configure SSL through the Administration Console:

1. Navigate to the server instance and click Lock & Edit.
2. Click the Configuration > SSL tab.
3. Enter the Keystore information and click Save.

Identity and Trust Locations: Indicates where SSL should find the server's identity (certificate and private key) as well as the server's trust (trusted CAs). If set to Keystores, SSL retrieves the identity and trust from the server's key store (that is configured on the server). The "Files or keystore providers" option is meant for use with older versions of WLS and is deprecated.

For a more secure deployment, Oracle recommends saving private keys in a keystore.

Road Map

- WLS Security Architecture overview
- Users and groups
- Protecting application resources
- Protecting communications
- Protecting against attacks
 - Types of attacks
 - Protecting against man-in-the-middle attacks
 - Protecting against denial of service (DoS) attacks
 - Protecting against large buffer attacks
 - Protecting against connection starvation



ORACLE®

Copyright © 2009, Oracle. All rights reserved.

Protecting Against Attacks

WLS can help protect applications against several attacks:

- Man-in-the-middle attacks
- Denial of service (DoS) attacks
- Large buffer attacks
- Connection starvation attacks



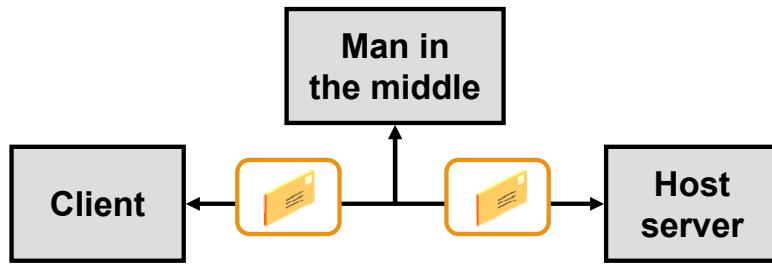
Copyright © 2009, Oracle. All rights reserved.

Protecting Against Attacks

In the following pages, attacks and countermeasures are described in detail.

Man-in-the-Middle Attacks

- In the “man-in-the-middle” attack, a third party poses as a destination host intercepting messages between the client and the real host.
- Instead of issuing the real destination host’s SSL certificate, the attacker issues his or her own hoping that the client would accept it as being from the real destination host.



ORACLE

Copyright © 2009, Oracle. All rights reserved.

Man-in-the-Middle Attacks

When you use SSL, servers that do not use a certificate signed by a trusted CA are vulnerable to the “man-in-the-middle” attacks.

If a client accepts the attacker’s certificate, the “man-in-the-middle” can decrypt and forward the traffic to and from the real destination host and monitor it.

Man-in-the-Middle: Countermeasures

- The “man-in-the-middle” attacks can be resisted by using a Hostname Verifier.
- A Hostname Verifier validates that the host to which an SSL connection is made is the intended or authorized party.
- WLS provides a Hostname Verifier by default.
- A custom Hostname Verifier can be created by implementing the `weblogic.security.SSL.HostnameVerifier` interface.



Copyright © 2009, Oracle. All rights reserved.

Man-in-the-Middle: Countermeasures

A Hostname Verifier is useful when an Oracle WebLogic Server or a WebLogic client acts as an SSL client to another application server. It prevents the “man-in-the-middle” attacks.

By default, Oracle WebLogic Server, as a function of SSL handshake, compares the common name in `SubjectDN` of the SSL server’s digital certificate with the host name of the SSL server that is used to initiate the SSL connection. If these names do not match, the SSL connection is dropped. The dropping of the SSL connection is caused by the SSL client, which validates the host name of the server against the digital certificate of the server.

If anything but the default behavior is desired, you can either turn off host name verification or register a custom Hostname Verifier. Turning off host name verification leaves Oracle WebLogic Server vulnerable to the “man-in-the-middle” attacks.

Note: Turn off host name verification when you use the demo digital certificates that are shipped with Oracle WebLogic Server. You can turn off host name verification in the following ways:

- In the Administration Console, select the Hostname Verification Ignored attribute under the SSL tab on the Server node.
- On the command line of the SSL client, enter the following argument:
`-Dweblogic.security.SSL.ignoreHostnameVerification=true`

Man-in-the-Middle: Countermeasures (continued)

- To use a custom Hostname Verifier, create a class that implements the `weblogic.security.SSL.HostnameVerifier` interface and define the methods that capture information about the server's security identity.
 - In the Administration Console, define the class for your Hostname Verifier in the Hostname Verifier attribute (an Advanced option under the Configuration > SSL tab for the server).
 - On the command line, enter the following argument:
`-Dweblogic.security.SSL.HostnameVerifier=hostnameVerifier` where `hostnameVerifier` is the name of the class that implements the custom Hostname Verifier.

Configuring a Hostname Verifier

The screenshot shows the 'Settings for MedRecSrv1' page in the Administration Console. The 'Configuration' tab is selected, and within it, the 'SSL' tab is selected. Under the 'SSL' tab, the 'Advanced' section is expanded. In the 'Hostname Verification:' dropdown, 'Custom HostnameVerifier' is selected. Below this, there is a 'Custom HostnameVerifier:' field which is empty. At the bottom of the 'Advanced' section, there is a checkbox labeled 'Use Server Certs' which is checked. The Oracle logo is visible at the bottom right of the page.

Configuring a Hostname Verifier

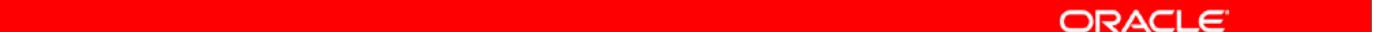
To configure a custom Hostname Verifier, perform the following steps:

1. If you have not already done so, in the Change Center of the Administration Console, click **Lock & Edit**.
2. In the left pane of the Console, expand **Environment** and select **Servers**.
3. Click the name of the server for which you want to configure a Hostname Verifier.
4. Select **Configuration > SSL** and click **Advanced** at the bottom of the page.
5. Select the appropriate Hostname Verifier in Hostname Verification.
6. Enter the name of the implementation of the `weblogic.security.SSL.HostnameVerifier` interface in the **Custom Hostname Verifier** field.
7. Click **Save**.
8. To activate these changes, in Change Center of the Administration Console, click **Activate Changes**.

Note: Not all changes take effect immediately; some require a restart of the server.

Denial of Service Attacks

- DoS attacks are attempts by attackers to prevent legitimate users of a service from using that service.
- There are three basic types of attack:
 - Consumption of scarce, limited, or nonrenewable resources
 - Destruction or alteration of configuration information
 - Physical destruction or alteration of network components



ORACLE

Copyright © 2009, Oracle. All rights reserved.

Denial of Service Attacks

Denial of service (DoS) attacks can disable your computer or your network. Depending on the nature of your enterprise, this can effectively disable your organization.

Some DoS attacks can be executed with limited resources against a large, sophisticated site. This type of attack is sometimes called an “asymmetric attack.” For example, an attacker with an old PC and a slow modem may be able to disable much faster and more sophisticated machines or networks.

Examples include attempts to:

- “Flood” a network, thereby preventing legitimate network traffic
- Disrupt connections between two machines, thereby preventing access to a service
- Prevent a particular individual from accessing a service
- Disrupt service to a specific system or person

Denial of Service Attacks: Countermeasures

Harden WLS against DoS attacks by:

- Filtering incoming network connections
- Configuring consumable WLS resources with the appropriate threshold and quotas
- Limiting access to configuration information and configuration tools
- Limiting access to back up configuration files
- Preventing unauthorized access by protecting passwords against password-guessing attacks



Copyright © 2009, Oracle. All rights reserved.

Denial of Service Attacks: Countermeasures

You can also use tools such as Oracle Adaptive Access Manager (OAAM) that can effectively prevent unauthorized accesses.

Filtering Network Connections

- WLS can be configured to accept or deny network connections based on the origin of the client.
- This feature can be used to restrict the:
 - Location from which connections to WLS are made
 - Type of connection made—that is, allow only SSL connections and reject all others
- To filter network connections, create a class that implements the `ConnectionFilter` interface and install it using the Administration Console.

The red bar spans the width of the slide content area.

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Filtering Network Connections

To configure connection filtering in the server, create a `ConnectionFilterImpl` class that implements the `weblogic.security.net.ConnectionFilter` interface (minimum requirement) and the `ConnectionFilterRulesListener` interface (optional). Use the Administration Console to install the class in Oracle WebLogic Server so that the server examines requests as they occur, and then accepts or denies them.

When a Java client or a Web browser client tries to connect to Oracle WebLogic Server, Oracle WebLogic Server constructs a `ConnectionEvent` object and passes it to the `accept()` method of your connection filter class. The `ConnectionEvent` object includes the remote IP address (in the form of `java.net.InetAddress`), the remote port number, the port number of the local Oracle WebLogic Server, and a string specifying the protocol (HTTP, HTTPS, t3, t3s, or IIOP).

The connection filter class (`ConnectionFilterImpl`) examines the `ConnectionEvent` object and either accepts the connection by returning or denies the connection by throwing a `FilterException`.

Connection Filter

The screenshot shows the 'Settings for MedRecDomain' page in the Oracle WebLogic Administration Console. The 'Security' tab is selected, and within it, the 'Filter' tab is active. A tooltip indicates that clicking the 'Lock & Edit' button will modify the settings on this page. A 'Save' button is present. The main content area describes how to define connection filter settings for the domain, mentioning the 'Connection Logger Enabled' checkbox and the 'Connection Filter' field. The 'Connection Filter Rules' section is also shown.

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Connection Filter

Using the Administration Console, access the domain (top) node in the navigation panel.

1. Click the **Security > Filter** tab.
2. After adding Filter Rules, click **Save**.
3. In the Connection Filter field, specify the connection filter class to be used in the domain.
 - To configure the default connection filter, specify `weblogic.security.net.ConnectionFilterImpl`.
 - To configure a custom connection filter, specify the class that implements the network connection filter. This class must also be present in CLASSPATH for Oracle WebLogic Server.

Excessive Resource Consumption

- Denial of service can come from consuming server-side resources used by Web applications:
 - Intentionally generating errors that will be logged, consuming disk space
 - Sending large messages, many messages, or delaying delivery of messages in an effort to cripple JMS
 - Disrupting network connectivity through “connection starvation”
 - Consuming system memory through “large buffer attacks”
- The effect of these attacks can be reduced by setting the appropriate quotas and threshold values.



ORACLE

Copyright © 2009, Oracle. All rights reserved.

Excessive Resource Consumption

The Oracle WebLogic Server resources can be vulnerable to abuse. A malicious piece of code can consume all the available database connections or cripple a service such as JMS by sending many large messages or delaying the delivery of messages.

You can reduce the effect of these attacks by using the Administration Console to set reasonable quotas and threshold values for each resource. You can also set the size of the log files and their rotation values to limit the amount of disk space that is consumed.

Large Buffer Attacks

- Individuals can try and bring down a Web site by sending a large buffer of data, which starves the system of memory.
- Administrators can combat this attack by setting a threshold for incoming data.

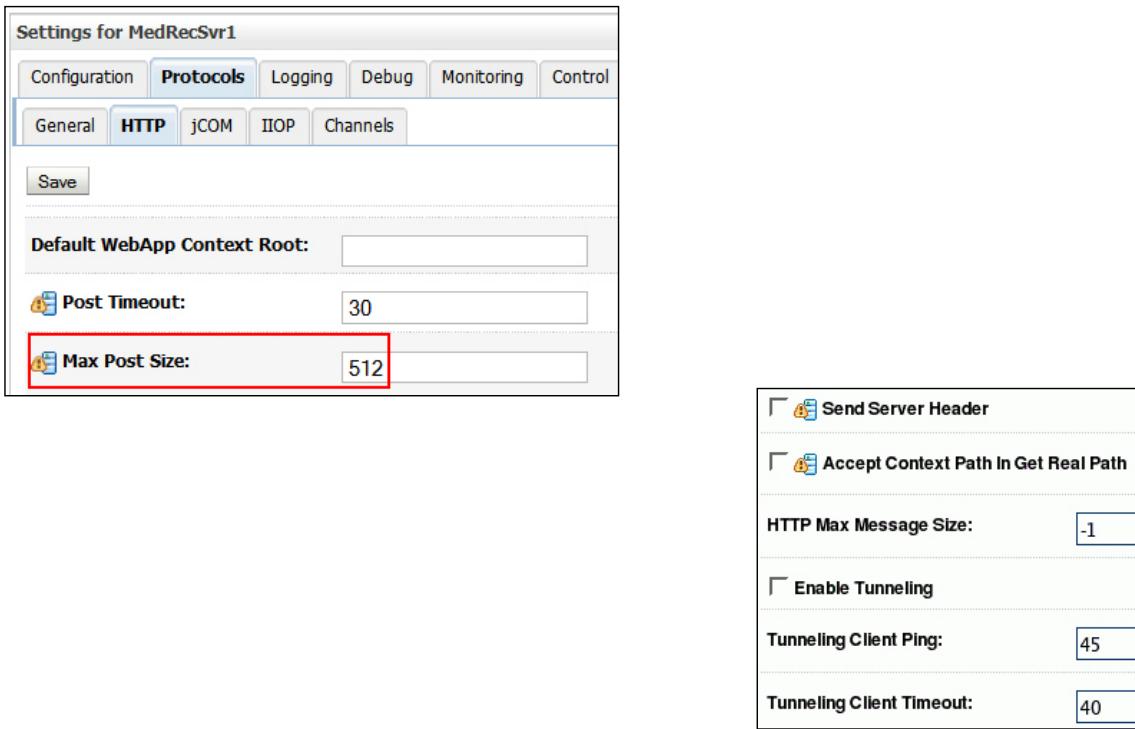


Copyright © 2009, Oracle. All rights reserved.

Large Buffer Attacks

Hackers try to bring down a Web site in a variety of ways. One particular way is referred to as large buffer attacks because hackers send large buffers of data to the server that starves the server of memory. Oracle WebLogic Server allows administrators to set a limit to the amount of HTTP data that can be posted to their servers. Administrators can use the Administration Console to manage this threshold. Any requests that exceed this threshold are denied access to the server.

Setting the Post Size



Copyright © 2009, Oracle. All rights reserved.

ORACLE

Setting the Post Size

The Max Post Size parameter determines the size of a data buffer that a server allows for reading HTTP POST data in a servlet request. A value less than 0 (such as -1) indicates an unlimited size.

To set the threshold of the request sizes that can be posted to the server, perform the following steps:

1. In the left pane, select the server that you want to set the limit on.
2. Click the Protocols > HTTP tab in the right pane.
3. Set Max Post Size. This is the threshold amount for the incoming requests. In this example, the maximum amount of data sent is 512 KB.
4. After you have finished entering your information, click Save to save your changes.

Similarly, HTTP Max Message Size limits the number of bytes allowed in messages that are received over the HTTP protocol. If you configure custom network channels for this server, each channel can override this maximum message size. This maximum message size helps guard against a DoS attack in which a caller attempts to force the server to allocate more memory than is available, thereby keeping the server from responding quickly to other requests.

Note: You need to restart the server after making these modifications.

Connection Starvation

- Individuals can try and take down a Web site by sending small, incomplete messages that cause the server to wait.
- Administrators can combat this attack by setting a threshold.
- Connections time out while waiting for the remainder of the data if they have reached the threshold set by the administrator.

The red bar spans most of the width of the slide, centered horizontally.

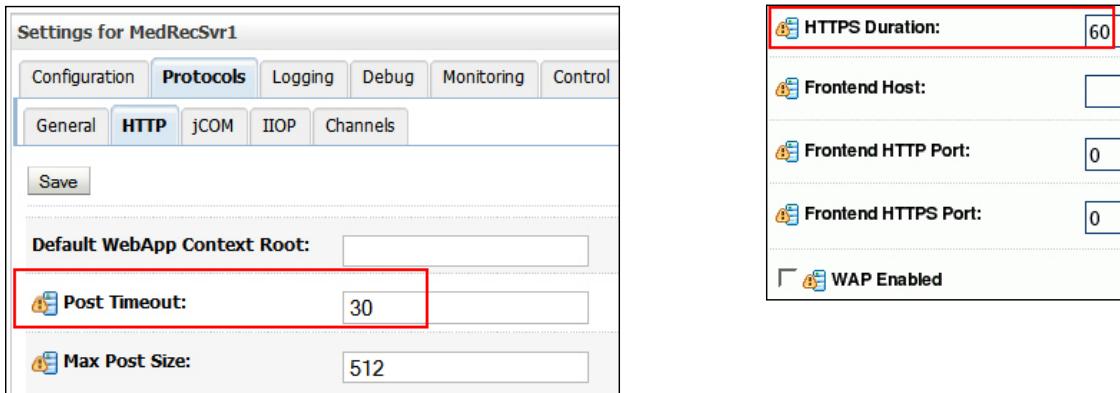
ORACLE

Copyright © 2009, Oracle. All rights reserved.

Connection Starvation

Another way that individuals can try and harm a Web site is by sending small, incomplete messages to the server. The server then waits for the completion of the message, in effect unduly burdening the server. Oracle WebLogic Server enables administrators to set a threshold for the time Oracle WebLogic Server will wait for the completion of the message. The administrator sets the time-out feature in the Administration Console and any connections that are still waiting for the completion of the message longer than this limit are canceled.

Connection Starvation



ORACLE

Copyright © 2009, Oracle. All rights reserved.

Connection Starvation (continued)

To set the threshold of Post Timeout and Max Post Time, perform the following steps:

1. In the left pane, select the server that you want to set the limit on.
2. In the right pane, click the Protocols tab.
3. Click the HTTP tab.
4. Set Post Timeout, which is the maximum amount of time that Oracle WebLogic Server waits for the next packet.
5. After you have finished entering your information, click Apply to save your changes.

Note: You need to restart the server after making these modifications.

Similarly you can also set the amount of time this server waits before closing an inactive HTTPS connection by using the HTTPS Duration parameter. The value you specify is in seconds. The value you specify is in seconds. The default of 60 seconds may be very large for some applications.

You specify the number of seconds during which to keep the HTTPS active before timing out the request.

User Lockout

- Individuals attempt to hack into a computer using various combinations of usernames and passwords.
- Administrators can protect against this security attack by setting the lockout attributes.
- The administrator can unlock a locked user using the console.

```
<May 2, 2009 2:42:36 PM EDT> <Notice> <Security> <BEA-090078> <User john Doe in security realm myrealm has had 5 invalid login attempts, locking account for 30 minutes.>
```

ORACLE

Copyright © 2009, Oracle. All rights reserved.

User Lockout

Password guessing is a common type of security attack. In this type of attack, a hacker attempts to log in to a computer by using various combinations of usernames and passwords. Oracle WebLogic Server provides a set of attributes to protect passwords and user accounts in a security realm.

Configuring User Lockout

The screenshot shows the 'User Lockout' configuration page for the 'myrealm' security realm. The 'Lockout Enabled' checkbox is checked. The configuration includes:

- Lockout Threshold:** 5
- Lockout Duration:** 30
- Lockout Reset Duration:** 5
- Lockout Cache Size:** 5
- Lockout GC Threshold:** 400

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Configuring User Lockout

The User Lockout feature enables you to prevent attack from hackers using a compromised user account. The User Lockout attributes apply to the security realm and all its security providers. If you are using an authentication provider that has its own mechanism for protecting user accounts, disable the Lockout Enabled attribute.

- **Lockout Threshold:** The maximum number of consecutive invalid login attempts before the account is locked out. For example, with the setting of 1, the user is locked out on the second consecutive invalid login. Minimum is 1 and the default is 5.
- **Lockout Duration:** The number of minutes that a user account is locked out. Minimum is 0 and the default is 30.
- **Lockout Reset Duration:** The number of minutes within which consecutive invalid login attempts cause the user account to be locked out. Minimum is 1 and the default is 5.
- **Lockout Cache Size:** The number of invalid login records that the server places in a cache. The server creates one record for each invalid login. Minimum is 0 and the default is 5.
- **Lockout GC Threshold:** The maximum number of invalid login records that the server keeps in memory. If the number of invalid login records is equal to or greater than this value, the server's garbage collection purges the records that have expired.

If a user lockout security event occurs on one node of a cluster, the other nodes in the cluster are notified of the event and the user account is locked on all the nodes in the cluster. This prevents a hacker from systematically breaking into all the nodes in a cluster.

Unlocking Users

Settings for MedRecDomain

Configuration Monitoring Control **Security** Web Service Security Notes

General Filter **Unlock User** Embedded LDAP Roles Policies

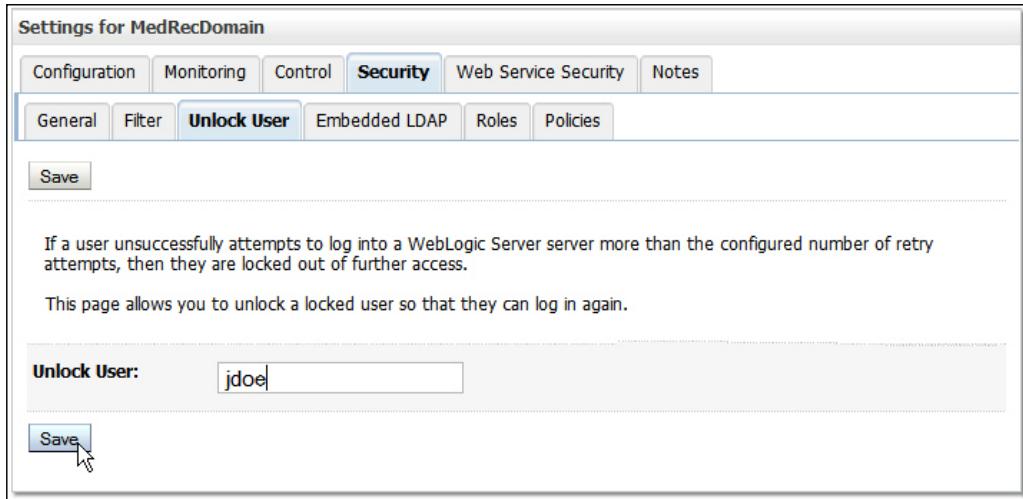
Save

If a user unsuccessfully attempts to log into a WebLogic Server server more than the configured number of retry attempts, then they are locked out of further access.

This page allows you to unlock a locked user so that they can log in again.

Unlock User:

Save



ORACLE®

Copyright © 2009, Oracle. All rights reserved.

Unlocking Users

If a user unsuccessfully attempts to log in to a WebLogic Server more than the configured number of retry attempts, they are locked out of further access. The Unlock User page allows you to unlock a locked user so that they can log in again.

Note: If a user account becomes locked and you delete the user account and add another user account with the same name and password, the User Lockout attribute will not be reset—that is, the added user may remain in the lockout status.

Protecting the Administration Console

- You can configure a separate administration port for all administration traffic.
- You can change the context path of the console.
- You can disable the console (application).

The screenshot shows the 'Settings for MedRecDomain' configuration page. The 'Configuration' tab is selected. Under the 'General' tab, there is a section titled 'Enable Administration Port' with an input field for 'Administration Port' set to 9002. Another section titled 'Console Enabled' has a checked checkbox. The 'Console Context Path' is set to 'console'.

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Protecting the Administration Console

By configuring a separate administration port for administration tasks, you do not expose the administration ports to other application ports. Before you enable an administration port, you ensure that all the servers in the domain are configured with SSL.

Similarly, you can reconfigure the context path of the console so that it does not remain the generally known /console.

Finally, in a production environments where you are less likely to make configuration changes regularly, you can disable the console application.

Quiz

The Hostname Verifier is one measure for combating this type of attack:

1. Large buffer
2. Connection starvation
3. Man in the middle
4. User lockout



Copyright © 2009, Oracle. All rights reserved.

Answer: 3.

Quiz

To counter connection starvation attacks, you can set:

1. Max Post Size
2. Post Timeout
3. Hostname Verifier
4. User lockout



Copyright © 2009, Oracle. All rights reserved.

Answer: 2.

Summary

In this lesson, you should have learned how to:

- Describe the process of configuring SSL
- Use the `keytool` utility to configure keys and obtain digital certificates
- Configure SSL for the WLS server
- Configure countermeasures for some Web-based attacks such as:
 - Man in the middle
 - Denial of service
 - Large buffer
 - Connection starvation



Copyright © 2009, Oracle. All rights reserved.

Practice 19: Overview

This practice covers the following topics:

- Using keytool to generate an identity keystore that contains a private key and a self-signed public certificate
- Configuring keystores using the Administration Console
- Configuring SSL for a managed server



Copyright © 2009, Oracle. All rights reserved.

20

Backup and Recovery Operations

ORACLE®

Copyright © 2009, Oracle. All rights reserved.

Objectives

After completing this lesson, you should be able to:

- Recommend a backup and recovery strategy
- Perform a full offline backup and recovery
- Perform an online and offline domain backup
- Perform an offline domain recovery
- Perform an Instance Home backup and recovery



Copyright © 2009, Oracle. All rights reserved.

Objectives

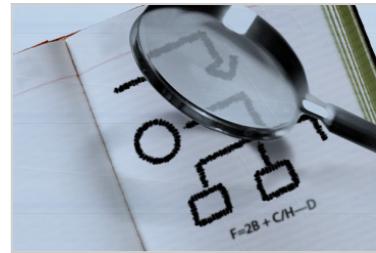
Scenario

As the middleware administrator, you need to plan a reasonable backup strategy that balances risk against inconvenience. Backing up once a month is too infrequent, whereas once an hour is too frequent, so what is the right balance? Given that you will do far more backups than recoveries, a plan that favors backup by shortening the time to create the backups at the expense of lengthening and complicating the recovery might be worth trying. Given that different backup strategies cause different kinds of recoveries, you plan to time how long it takes to do a recovery to help create service-level agreements (SLA).

Note the distinction between *restore* and *recover*: restore is a pure file system copy operation, whereas recovery is restore plus some extra operations.

Road Map

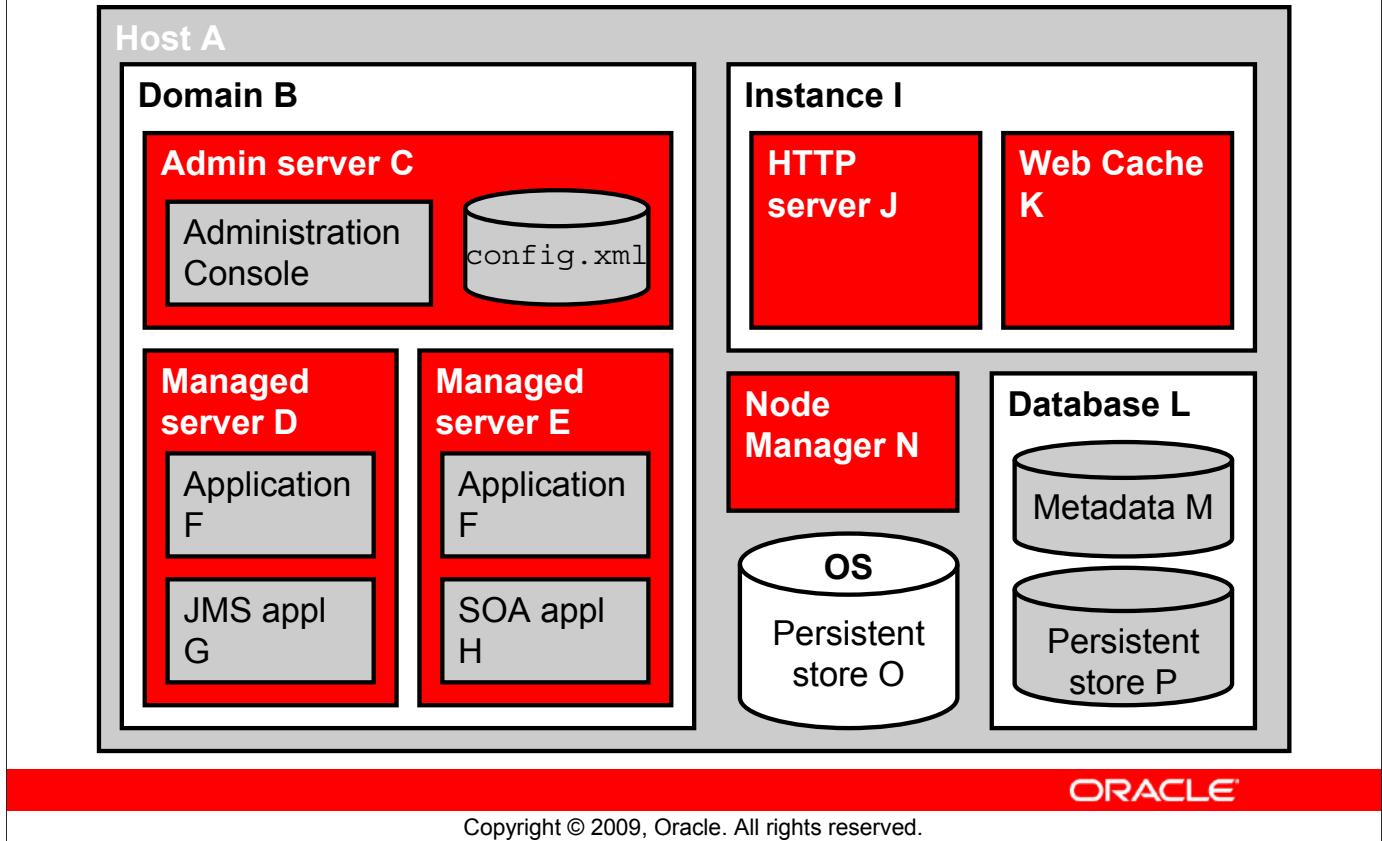
- Backup
 - Full
 - Incremental
 - Online
 - Offline
- Recovery



ORACLE®

Copyright © 2009, Oracle. All rights reserved.

Review of Terms and Components



Review of Terms and Components

- Host:** The computer may have redundant CPUs, RAID disks, and/or other hardware failover features.
- Domain:** WebLogic Server is an example of a system component domain, and Oracle HTTP Server and Oracle Web Cache are system component domains as well. In this case, a WebLogic Server domain consists of at least one administration server and zero or more managed servers. These servers are Java components.
- Administration server:** This server is required at the initial start of a managed server, but not thereafter. The administration server contains the master config.xml file, which is copied to managed servers at various times (startup, configuration changes, and so on).
- Managed server:** This server runs Java EE applications. The server may be part of a cluster.
- Managed server:** This server is the SOA server. SOA requires a metadata repository on a database (created by the Repository Creation Utility [RCU]) or in a plain file.
- Application:** An application may be deployed on a cluster or on several stand-alone servers.
- JMS application:** Messages can be stored either in a database persistent store or in a plain file persistent store.
- SOA application:** SOA requires a metadata repository, either in a database or a plain file.

Review of Terms and Components (continued)

- I. **Instance:** This is similar to a domain, but it contains system components. A system component is a non-Java component managed by the Oracle Process Manager and Notification (OPMN) server.
- J. **Oracle HTTP Server:** Based on the Apache 2.2.10 infrastructure, this includes modules developed specifically by Oracle. The features of single sign-on, clustered deployment, and high availability enhance the operation of Oracle HTTP Server.
- K. **Web Cache:** Because this is a *cache*, there is no permanent data to back up or recover, but there are configuration files and logs. Because there is no live data that you care about, the backup can be performed online. There is no need to worry about consistency or run-time artifacts.
- L. **Database:** Assuming this is an Oracle database, the backup tool is Recovery Manager (RMAN), capable of performing online backups and automated recovery to either any point in time or a complete recovery. A Flashback log (if configured) also provides a rolling recovery window. If the environment permits offline backup, after the processes are all stopped, a simple OS copy of all files will work. RMAN tasks are typically performed by the DBA and are outside the scope of this course.
- M. **Metadata:** Required for SOA, this is created by RCU. Use the database tools for backup and recovery.
- N. **Node Manager:** One per host, the Node Manager can autorestart managed servers that fail.
- O. **Persistent store:** This is an OS file that could contain JMS transactions.
- P. **Persistent store:** This is a database schema that could contain JMS transactions. Use the database tools for backup and recovery.

Static artifacts: The program binaries do not change very often. Their backup schedule might be only after patches, monthly, or even longer.

Run-time artifacts: These objects change frequently, even multiple times per second in the case of logs. Configuration objects may change several times per day, though typically they remain unchanged for long periods of time.

Persistent stores: These objects may change very frequently, even hundreds of times per second, depending on the volume of data traffic. A high-performance solution may be required so as to not lose data.

Homes: Oracle, Middleware, WebLogic

You can set up the disk in any way you like; this is only a portion of one suggested layout:

```
/u01/app/oracle ..... <ORACLE_BASE>
  /instances/config/OHS ..... <ORACLE_INSTANCE>
  /oradata
  /oraInventory
  /product
    /db/11.1.0/orcl ..... <ORACLE_HOME> one of many
    /fmw/11.1.0 ..... <MW_HOME>
      /jrockit_160_xxx ..... <JAVA_HOME>
      /webtier
      /wlserver_10.3 ..... <WL_HOME>
  /user_projects
    /applications
    /domains
```

Back up each of the homes.

Copyright © 2009, Oracle. All rights reserved.

Homes: Oracle, Middleware, WebLogic

This is the layout of the disks in the lab. Each of the homes can be a point for starting an incremental backup. Starting from <ORACLE_BASE> would be a full offline backup.

Oracle home: There can be several simultaneous <ORACLE_HOME>s. An Oracle home contains installed files necessary to host a specific product. Shown is the home for the database. Not shown might be a <SOA_ORACLE_HOME> and a <WC_ORACLE_HOME> (Web Cache). For example, the SOA Oracle home contains a directory that contains binary and library files for Oracle SOA Suite. The WebLogic Server home also consists of its installed files.

An Oracle home resides within the directory structure of the Middleware home. Each Oracle home can be associated with multiple system component domains or Oracle WebLogic Server domains. The WebLogic Server Home directory is a peer of Oracle Home directories. In order to keep all the multiple Oracle homes from conflicting with each other, they should be defined only in the scripts that start a particular process, not globally defined in a .profile nor using the source command.

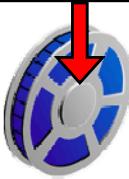
Middleware home: The Middleware home consists of the Oracle WebLogic Server home and one or more Oracle homes, such as SOA home and Web Cache home.

Instance home: The Instance would contain the Oracle HTTP Server and Web Cache configuration files.

Understanding Backup and Recovery

Backup

- Scheduled
- At least weekly (to capture logs)
- Different tools for different components



Recovery

- Unscheduled (usually)
- At least annually (if only to test procedures)
- Not necessarily the reverse of backup, may be new tools



- Protects against failures of hardware, software, power, environmental disasters, accidental and malicious changes, and more
- Guarantees a point of recovery, minimizes loss of business availability, insures an SLA, may satisfy legal requirements
- May impact business
- May be hardware and software

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Understanding Backup and Recovery

Commonly, the terms “backup” and “recovery” imply the use of secondary media to copy some data for later use. That kind of backup and recovery involves an offline or cold storage of the data such that if an outage occurs, then some process (human or automated) requires some time to get the system back up and running. Alternatively, “redundancy” and “failover” are additional means by which to back up and recover the data in more of an online or warm or hot storage mode, thus reducing, or even eliminating the switchover time. If an outage occurs with redundancy and failover implemented, it is often undetectable by the user. The following are different forms of backup and recovery:

- Redundant disks in a SCSI array
- Multiple servers configured on multiple machines in a cluster with an application deployed on the cluster
- The ability to cancel all pending changes to a configuration
- The architecture of the Oracle 11g Database with inherent transaction logging

In addition to those very significant features, a media backup plan is essential. The most common problem that requires a backup and recovery is when a person who is authorized to make changes accidentally commits a wrong change. Usually, the mistake is realized within seconds and all that is needed is a mechanism that will enable the user to go back to a very recent version of the configuration. A more serious problem is when there is a complete loss of

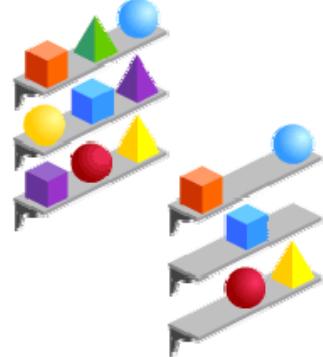
Understanding Backup and Recovery (continued)

the computer hosting the WebLogic component. There is no single point of failure in the Fusion Middleware architecture, but there may be an impact in service (for example, no configuration changes can be made while the administration server is down).

Backup and recovery policies may impact your business both financially and in terms of availability (required maintenance windows).

Types of Backups

- Online
 - Nondisruptive
 - Possibly inconsistent
 - Can be tricky, especially for database
- Offline
 - Requires *all* processes be stopped
 - Very easy
- Full
 - Easier to recover
 - Slower to create
- Incremental
 - Harder to recover
 - Faster to create



ORACLE

Copyright © 2009, Oracle. All rights reserved.

Types of Backups

Online

If your environment requires 24x7 availability, you have no choice but to perform an online backup. Different components require different tools to perform online (also known as hot or inconsistent) backups. Inconsistent is not bad in itself; it just means that if the backup takes an hour to complete and you start at 1:00 AM, the files at 1:02 AM will be in a different state than those backed up at 1:00 AM. To accommodate this, there needs to be some kind of online transaction log recording the changes occurring from 1:00 AM until 2:00 AM. This log needs to be incorporated into the recovery, and the logs themselves get backed up at a different time (usually, after they rotate).

Offline

If you can afford to shut down the entire middleware tier (application servers, database, Web servers, and so on) for maintenance during some regularly scheduled time, an offline (also known as cold or consistent) backup is very simple. Using OS tools such as TAR or ZIP, the backup is guaranteed to be consistent. Make sure you preserve file permissions on UNIX systems.

Types of Backups (continued)

Full

After the initial installation, or after a major set of patches, a full backup should be performed. Often, this is done before going live, so the system is offline. It is very difficult (if not impossible) to perform a full backup online. If there is a complete loss of a host (for example, a disaster such as a fire or flood), recovery is simple; just copy the backup files to the new bare-metal host and boot. Name the backup so as to include the date—for example, `my_full_backup_2009_03_30.tar`—and keep several generations of them in case you accidentally capture “the problem” in the most recent backup.

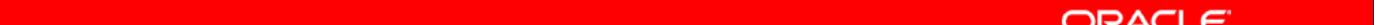
Incremental

Considering that the executable files and the configuration files are usually backed up separately, most backups are partially incremental. Backing up only changes may require several sets of backups recovered in order to perform a full recovery. RMAN can help automate this for databases, especially if the backups are kept online (on disk as opposed to tape).

You can make an incremental backup at the functional level. For example, you can make a WebLogic backup from `<WL_HOME>`, make an instance backup from `<ORACLE_INSTANCE>`, make a database backup from `<ORACLE_HOME>`, and so on. Within WebLogic, make a backup of all domains and then make backups of individual domains. The disadvantage of doing this is that the backup process will take a long time, but the advantage is that the recovery process can be greatly simplified. Alternatively, if you do not make so many different kinds of incremental backups, the backup procedure will complete faster, but now you have complicated and lengthened your potential recovery time. It is a delicate tradeoff balancing storage space versus time versus complexity.

Backup Recommendations

- After the initial domain is created (offline)
- Scheduled backups (online)
- After the component or the cluster changes (online)
- Before application deployment (online)
- Before patches or upgrades (offline)
- Database backups (online) for:
 - LDAP
 - Persistent stores
 - SOA repository



ORACLE

Copyright © 2009, Oracle. All rights reserved.

Backup Recommendations

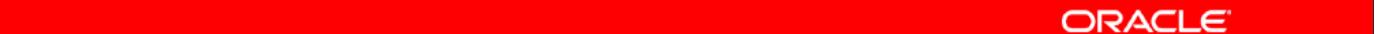
The initial software installation and most patches and upgrades require the servers to be offline anyway, so before and after the patches and upgrades is a good time to perform backups.

Many of the online configuration backups can be automatic by enabling configuration archive (discussed in the following slides).

The database should be in archivelog mode and then backed up with RMAN. In addition, the database should be configured with redundant critical files (for example, control files) and multiplexed critical logs (for example, redo logs). As an added high availability measure, the database can be made completely redundant by using RAC.

Limitations and Restrictions for Backing Up Data

- You should not be adding users or changing permissions while backing up the Lightweight Directory Access Protocol (LDAP).
- Online persistent stores by nature are going to be an inconsistent backup.
 - Database backups can accommodate inconsistencies.
 - File-based stores and OS copies cannot accommodate online backup.
- HTTP session states and cookies information may be lost.
 - In-memory replication may lose the state.
 - JDBC replication of the HTTP session state solves this problem.

The red bar spans most of the width of the slide, centered horizontally.

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Limitations and Restrictions for Backing Up Data

None of these restrictions apply to offline backups; they apply only to online backups. In many cases, the WebLogic Server has the option to be configured to use either database or file storage for information. Choosing database is always a safer option, but you pay for it with complexity and perhaps a speed penalty. If you have a database anyway, and the DBA is backing up the database anyway, then some additional WebLogic Server files should not be any additional effort, so it is worth the security to specify database storage when possible over OS file storage. For files such as configuration XML; application JARs, WARs, or EARs; and properties files; database storage is not an option.

Performing a Full Offline Backup

1. Shut down all processes:
 - Stop WebLogic via the Administration Console.
 - Shut down the database.
 - Stop the Listener and the Node Managers.
 - Stop the Enterprise Manager and the emAgent.
 - Stop Web Cache and HTTP server via OPMN.
2. Perform the backup via OS tools:
 - If using TAR, make sure that you keep permissions.
 - If using ZIP, make sure that you include empty directories.
3. Test the backup by performing recovery on another computer:
 - Ideally, use an alternate computer in an alternate data center.
 - Time the recovery for SLA input.
4. Store the backup offsite.

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Performing a Full Offline Backup

Shutting Down

Stop all deployed applications so that you can shut down all servers. Verify the Node Managers and emAgent PIDs. Stop the Node Managers and emAgents. In SQL*Plus (`sqlplus / as sysdba`), issue `shutdown immediate`. This may take a while (despite the name, it is not “immediate”). Stop the database listeners and the Enterprise Manager console. Stop all OPMN-managed utilities (for example, OHS and Web Cache).

Performing Backup

In the lab, all the product and configuration information is stored in `/u01/app/oracle`. Signed on as `root`, from the `root` directory, use the appropriate operating system backup utilities (for example, `tar` or `winzip`):

```
tar -zcvpf mybackup1.tar /u01/app/oracle /etc/ora* etc/hosts
```

There may be more sophisticated options to exclude `/tmp/` files and to include parts of other applications, but this will do as a start. The sequence number 1, 2, 3, might be replaced with the `date_time` in the name of the TAR file. If the directories are backed up from the root, you do not need to worry about where to recover them to; that information will be part of the backup.

Performing a Full Backup (continued)

Testing Restore

Make sure that the first backup you took is successful. You need to do this very early in the life of the system while there is no urgent need. Many administrators will tell you unfortunate stories of how they found out only during an emergency that several tapes that they dutifully preserved were blank for some simple overlooked reason. Later, you need to perform scheduled recoveries to make sure that the processes are all still working.

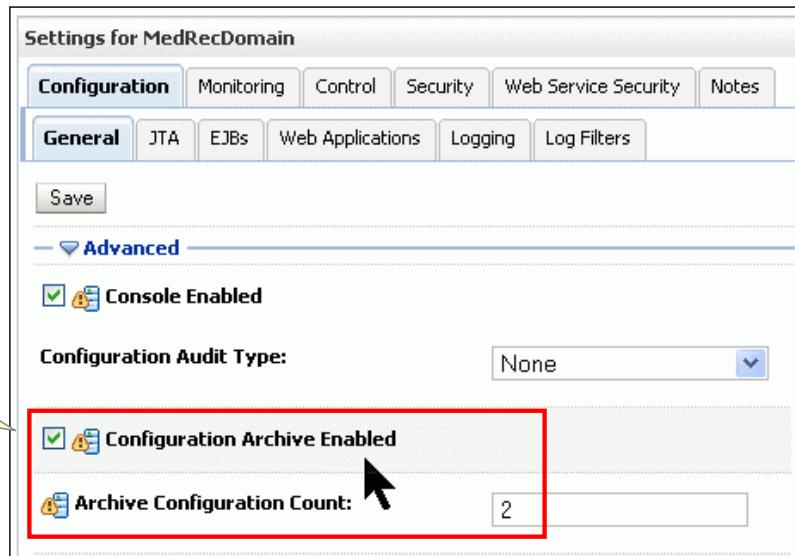
From the `root` directory, signed on as the `root` user, enter:

```
tar -zxvpf mybackup1.tar
```

Because you have signed on as `root`, it is vital to make sure that the `-p` switch is used in the `tar` command to preserve the original owners and group permissions (for example, `oracle` and `oinstall` versus `root`). Restart all processes to test the recovery and make sure that it is complete.

Backing Up a Domain Configuration

- Enable autobackup of configuration.
- Check new JAR files and directories.



ORACLE

Copyright © 2009, Oracle. All rights reserved.

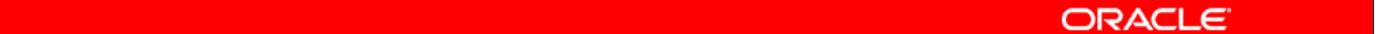
Backing Up a Domain Configuration

In Domain > Configuration > General > Advanced, you can enable autobackup at the domain level. Each startup of the administration server creates two files: config-booted.jar and config-original.jar in the domain directory. In addition, each saved change of the configuration file makes a backup named configArchive/config-n.jar, where n is a sequential number. Archive Configuration Count limits the number of retained configuration JARs, so that in the example shown, there are never more than two kept: the most recent backup and the one immediately before that. Older backups are automatically deleted. If you made a series of mistakes, this provides a very easy way to return to a previous recent configuration. However, be aware that a typical configuration change requires clicking Activate Changes a few times, and each one then cycles the stored JARs. You may want a higher number such as 10 or 20 for the count. An example from the MedRecDomain directory:

```
[oracle@edvmr1p0]# cd
/u01/app/oracle/user_projects/domains/MedRecDomain
[oracle@edvmr1p0]# ll conf*
drwxr-x--- 11 oracle oinstall 4096 Mar 23 16:51 config
drwxr----- 2 oracle oinstall 4096 Mar 25 08:58 configArchive
-rw-r----- 1 oracle oinstall 12328 Mar 25 08:54 config-booted.jar
-rw-r----- 1 oracle oinstall 12328 Mar 25 08:54 config-original.jar
[oracle@edvmr1p0]# ll configArchive/
-rw-r----- 1 oracle oinstall 12339 Mar 25 08:59 config-2.jar
-rw-r----- 1 oracle oinstall 12328 Mar 25 09:03 config-3.jar
```

Backing Up an Instance Home

- Stop the Web tier (Oracle HTTP Server and Oracle Web Cache):
 - opmnctl stopall
 - opmnctl status
- Copy the Instance home:
 - As the superuser, change to the root directory.
 - Execute `tar -zcvpf myinstance1.tar $ORACLE_INSTANCE`.
- Restart all services:
 - opmnctl startall
 - opmnctl status

The red bar spans most of the width of the slide, centered horizontally.

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Backing Up an Instance Home

There may be more sophisticated ways of not backing up the `/tmp/` files, but this is a good start. There is no facility for performing an online backup of the Instance home. After creating the backup, store a copy offsite. A sample Instance home might contain the following directories and files:

```
[oracle@edvmr1p0]$ ll instance2
total 32
drwx----- 4 oracle oinstall 4096 Mar 26 11:38 auditlogs
drwx----- 2 oracle oinstall 4096 Mar 26 11:37 bin
drwx----- 5 oracle oinstall 4096 Mar 26 11:38 config
drwx----- 3 oracle oinstall 4096 Mar 26 11:37 diagnostics
drwx----- 3 oracle oinstall 4096 Mar 26 11:37 OHS
drwx----- 2 oracle oinstall 4096 Mar 26 11:37 tmp
drwx----- 3 oracle oinstall 4096 Mar 26 11:38 WebCache
-rw----- 1 oracle oinstall      9 Mar 26 11:38 webcacheAdmin1621.txt
```

Creating a Record of Installations

Create a record for your Oracle Fusion Middleware product installation. The record must contain:

- For each host:
 - Names and addresses
 - OS information
- For each installation:
 - Installation type, host, owner name and number, group name and number, environment profile and type of shell, directory structure, mount points, full path for Oracle home, and port numbers used by the installation

Store it offsite.



ORACLE

Copyright © 2009, Oracle. All rights reserved.

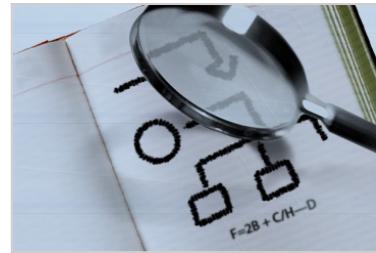
Creating a Record of Installations

You should maintain an up-to-date record of your Oracle Fusion Middleware installations in hard copy and in electronic form. You need this information in the event that you must restore and recover your installations to a new disk or host. The electronic form should be stored on a system that is completely separate from your Oracle Fusion Middleware that is being backed up. Your hardware and software configuration record should include:

- The following information for each host in your environment:
 - Host name, virtual host name (if any), domain name, IP address, hardware platform, and operating system release level and patch information
- The following information for each Oracle Application Server installation in your environment:
 - Installation type (for example, Infrastructure, or Java EE and Web Cache), host on which the installation resides, username, user ID number, group name, group ID number, environment profile, and type of shell for the operating system user that owns the Oracle home (/etc/passwd and /etc/group entries), directory structure, mount points, and full path for Oracle home, and port numbers used by the installation
 - For Oracle Database, the database version, patch level, base language, character set, global database name, and SID

Road Map

- Backup
- Recovery



ORACLE®

Copyright © 2009, Oracle. All rights reserved.

Directories to Restore

- Binaries
 - Be mindful of preserving group ownership and permissions.
 - This should be read-only for most users.
- Configurations
 - If the last configuration *caused* the problem, recover to a point in time prior to that.
- Logs are:
 - Not required for recovery
 - Created if they do not exist
- Data
 - Database restores data within tablespaces, not directories.
 - RMAN *restore* brings data up to the last backup, then *recover* brings data up to a later point in time.

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Directories to Restore

In most cases, recovery is performed offline. If you think that only one or two files are missing, you may be tempted to recover only those individual files from the system. But, instead, you should always recover whole directories because there may be other files that are related to these files.

If the directories were backed up from the root, you do not need to worry about where to recover them to. The full path information will be provided to the operating system because it is contained in the backup. Restore them as the `root` user, from the `root` directory, and they will go back to their correct hierarchies. Do not forget the `-p` switch in the `tar` or `jar` command to get the original owner and group information correct.

Recovery After Disaster

- Possible causes of failure:
 - Data loss
 - User error
 - Malicious attack
 - Corruption of data
 - Media failure
 - Application failure
- Recovery depends on the cause:
 - Repair
 - Replace
 - Relocate



ORACLE®

Copyright © 2009, Oracle. All rights reserved.

Recovery After Disaster

If the problem was caused by a minor configuration error, the administrator may be able to reverse the steps and remove the problem without a formal recovery. If the problem requires replacing hardware, restoring full backups is a simple procedure. Recovery is complicated when you need to relocate some functions to an existing machine. According to the old configuration (and backups), the functions must be routed to the old name and address of A, but now according to the new configuration, the functions need to be routed to the new name and address of B.

Recovery of Homes

This applies to recovering a Middleware home, Oracle home, or Instance home after data loss or corruption:

1. Stop all processes.
2. Make a new full offline backup as a checkpoint (which can be reused).
3. Change directory to the affected home.
4. Use OS copy, untar, or unzip commands for the directories affected.
5. Make a new full offline backup (especially if you have been performing incremental backups up until this point).
6. Restart all processes.



Copyright © 2009, Oracle. All rights reserved.

Recovery of Homes

Make sure that all Fusion Middleware software is stopped so that this is an offline recovery. The most important rule in problem resolution is: “Do not make the problem worse.” By performing the two extra backups, you guarantee that you can at least put everything back to the way it was before you tried to help.

Assume that the last good backup was sequence number 9. As an example, here is how to recover a damaged Instance home:

In the Administration Console, shut down all servers including the administration server:

```
opmnctl stopall
```

In SQL*Plus, shut down the database cleanly, that is, using “immediate”:

```
lsnrctl stop
tar -zcvpf mycheckpoint.tar /u01/app/oracle
tar -zxvpf myinstance09.tar
tar -zcvpf myfullbackup10.tar /u01/app/oracle
lsnrctl start
```

In SQL*Plus, start the database:

```
opmnctl startall
```

Start the administration server.

From the Administration Console, start the managed servers.

Recovery of a Managed Server

- If the software crashes, the Node Manager will automatically restart it.
- If the files are damaged, you can recover the files in their original places and restart the software.
- If the computer is damaged, perform either of the following:
 - Restore the files on a new host with the old computer name by using the following OS commands—for example, `copy`, `cp`, `tar`, or `unzip`.
 - Restore the files on another host with a different host name by using templates to extend the domain.

The red bar spans most of the page width, with the Oracle logo centered on it.

ORACLE®

Copyright © 2009, Oracle. All rights reserved.

Recovery of a Managed Server

The original pack command that created the remote managed server can be used to re-create it in a recovery. The significant configuration and application files are stored at the administration server, so when the managed server comes back, it will first refresh all its configuration information and redeploy all its applications from the administration server.

Recovery of the Administration Server Configuration

Managed Server Independence (MSI) reduces the urgency to fix the outage.

The screenshot shows the 'Settings for MedRecSvr1' configuration page. The 'Tuning' tab is selected. A yellow callout box labeled 'Enabled by default' points to the 'Managed Server Independence Enabled' checkbox, which is checked. The 'Period Length' field is set to 60000 milliseconds. The 'Idle Periods Until Timeout' field is set to 4. A note indicates that a value of 0 means heartbeats are turned off.

ORACLE

Copyright © 2009, Oracle. All rights reserved.

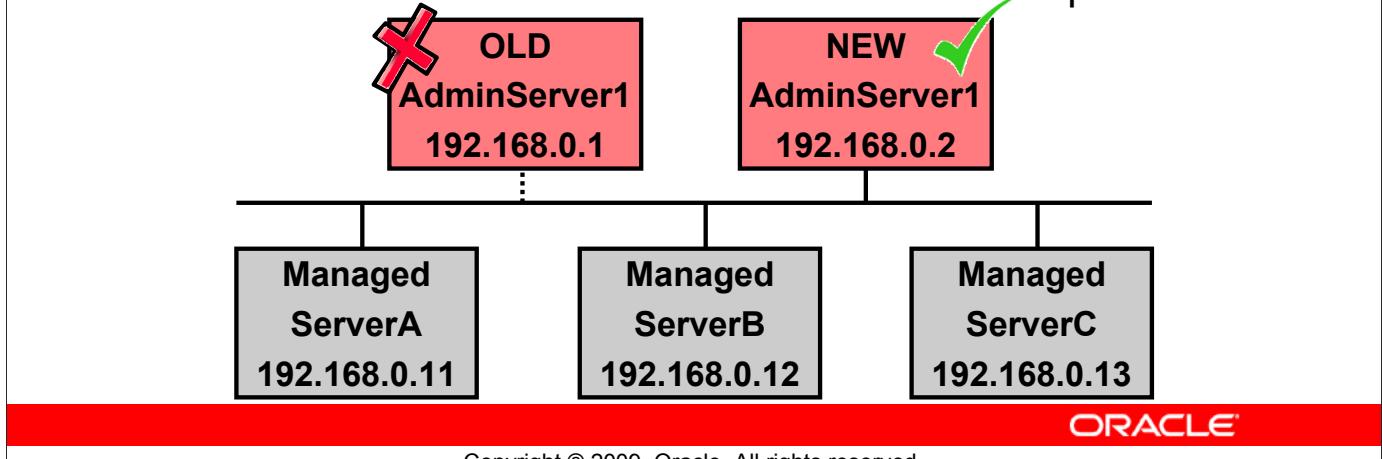
Recovery of the Administration Server Configuration

The administration server is required only for making changes to the active configuration; it is not required for the normal operation of the managed servers as long as the managed servers are in Managed Server Independence Enabled mode, which is the default. This allows you time to recover the administration server without any service outages. As shown in the screenshot, the heartbeat detected between the administration server and the managed servers is, by default, a one-minute period. After four minutes of not hearing from the administration server, the managed servers become independent. After the administration server is fixed, the heartbeats start up again and the managed servers deactivate their independence, but MSI is still enabled for a future event. These times can all be changed to suit your particular environment.

Restarting an Administration Server on a New Computer

Oracle WebLogic Server allows the creation of a backup of the administration server as follows:

1. Install Oracle WebLogic Server on a backup computer.
2. Copy the application files to a backup computer.
3. Copy the configuration files to a backup computer.
4. Restart the administration server on a new computer.



Copyright © 2009, Oracle. All rights reserved.

Restarting an Administration Server on a New Computer

If a hardware crash prevents you from restarting the administration server on the same computer, you can recover the management of the running managed servers as follows:

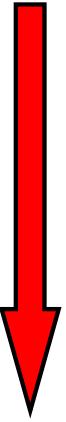
1. Install the Oracle WebLogic Server software on the new computer designated as the replacement administration server.
2. Make your application files available to the new administration server by copying them from backups or by using a shared disk. Your files must be available in the same relative location on the new file system as on the file system of the original administration server.
3. Make your configuration and security files available to the new administration computer by copying them from backups or by using a shared disk. These files are located under the directory of the domain being managed by the administration server.
4. Restart the administration server on the new computer.

When the administration server starts, it communicates with the already running managed servers via a Node Manager and informs the servers that the administration server is now running on a different IP address.

Note: You cannot have two administration servers at the same time, both claiming ownership of the same managed servers. This is not a warm standby; this must be a cold standby. The original administration server must be stopped or dead for the backup administration server to contact the managed servers.

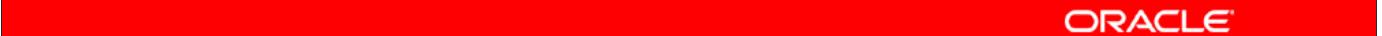
Recovery of a Cluster

If you accidentally lost a member of a cluster or a whole cluster, you can use several ways to recover it.

 **+ Most preferable way to recover**

- Undo the changes in the Change Center.
- Reenter the configuration changes that you made.
- Use the configuration archive to go back one or two versions.
- Recover the configuration.
- Recover the domain.
- Recover WebLogic.
- Perform a full recovery.

- Least preferable way to recover

 ORACLE

Copyright © 2009, Oracle. All rights reserved.

Recovery of a Cluster

All the methods require stopping the cluster itself using WebLogic Scripting Tool (WLST) or the Administration Console. The first two methods do not require stopping any other processes, which means that it can be an online recovery. The remaining methods require stopping all processes and performing an offline recovery.

Restoring OPMN-Managed Components to a New Computer

1. Use the methods described earlier to recover the files as though this was the same host.
2. Update the registration of the Oracle instance with the administration server using:
`updateinstanceregistration`
3. Update the registration of the component with the administration server using:
`updatecomponentregistration`
4. Edit the `targets.xml` file for Fusion Middleware Control.
5. Edit `emd` files for Enterprise Management Agent.
6. Restart the EM Agent.



Copyright © 2009, Oracle. All rights reserved.

Restoring OPMN-Managed Components to a New Computer

The syntax for the command to update the instance registration is:

```
opmnctl updateinstanceregistration -adminHost new_host
```

This command updates OPMN's `instance.properties` file with the new host name.

The syntax for the command to update component registration on the new host depends on the components that you are updating. For example, to update the registration for Oracle Virtual Directory, use the following command:

```
opmnctl updatecomponentregistration -Host new_host
    -Port nonSSLPort
    -componentName ovd1
    -componentType OVD
```

For the `targets.xml` file located in

`<MW_HOME>/user_projects/domains/domain_name/servers/AdminServer/sysman/state`, change the host name to the new host name.

To recover the EM Agent, edit the following files to change the host name:

`<ORACLE_INSTANCE>/EMAGENT/emAgnt_instname/sysman/emd/targets.xml`
`<ORACLE_INSTANCE>/EMAGENT/emAgnt_instname/sysman/config/emd.properties`

If the component is Web Cache, you also need to edit the host name in:

`<ORACLE_INSTANCE>/config/WebCache/webcache_name/webcache.xml`

Quiz

What mode must the Middleware software be in to perform a full backup?

1. Online
2. Offline
3. Either online or offline
4. Neither. A full backup is technically impossible.



Copyright © 2009, Oracle. All rights reserved.

Answer: 2

To be consistent, the Middleware software must be completely stopped.

Quiz

What is another name for an inconsistent backup?

1. Hot
2. Cold
3. Either online or offline
4. Broken. If it is inconsistent, there is something wrong with it.



Copyright © 2009, Oracle. All rights reserved.

Answer: 1

In a hot backup, the files are inconsistent—that is, some files may have different time stamps and need to be reconciled via a transaction log.

Quiz

When making a TAR backup in UNIX, what is a key point to remember?

1. Make it from the lowest directory possible, as far from root as practical.
2. Make sure that you perform the backup signed on as the owner of the Middleware Home directory.
3. Make sure that you preserve the original owner, group, and permissions.
4. Make sure that all Middleware processes are stopped.

ORACLE

Copyright © 2009, Oracle. All rights reserved.

Answer: 3

In TAR, use the -p option to preserve the permissions.

Quiz

The configuration archive is enabled by default.

1. True
2. False



Copyright © 2009, Oracle. All rights reserved.

Answer: 2

You need to enable the configuration archive by selecting Domain > Configuration > General > Advanced.

Quiz

What happens if you have a backup administration server?

1. You are allowed to have only one administration server. If it fails, the managed servers run in MSI mode until your one administration server comes back.
2. It runs simultaneously with the primary administration server in a load-sharing mode.
3. It can run in a warm standby keeping itself in sync with the main administration server.
4. It must be in cold standby and you have to sync it with the main administration server manually.



Copyright © 2009, Oracle. All rights reserved.

Answer: 4

You can have only one administration server at a time; the backup administration server must be cold.

Summary

In this lesson, you should have learned how to:

- Recommend a backup and recovery strategy weighing convenience against risk
- Perform a full offline backup and recovery of all components using OS copy tools
- Perform an online domain backup and recovery of the configuration
- Perform an Instance home backup and recovery for Oracle HTTP Server and Web Cache



Copyright © 2009, Oracle. All rights reserved.

Practice 20 Overview: Backing Up and Restoring Configuration and Data

This practice covers the following topics:

- Backing up an Oracle WebLogic domain
- Backing up an Oracle HTTP Server installation
- Restoring an Oracle WebLogic domain
- Restoring an Oracle HTTP Server installation



Copyright © 2009, Oracle. All rights reserved.

Practice 20 Overview: Backing Up and Restoring Configuration and Data

See Appendix A for the complete steps to do the practice.

Appendix A

Practices and Solutions

Table of Contents

| | |
|--|----|
| Practices for Lesson 1 | 5 |
| Practice 1-1: Connecting to the Classroom Grid | 6 |
| Practices for Lesson 2 | 11 |
| Practices for Lesson 3 | 12 |
| Practice 3-1: Installing Oracle WebLogic Server | 13 |
| Practice 3-2: Navigating the WLS Installation Directories | 14 |
| Practices for Lesson 4 | 16 |
| Practice 4-1: Creating a Minimal Domain from the Beginning..... | 17 |
| Practice 4-2: Creating a Functional Domain..... | 20 |
| Practices for Lesson 5 | 23 |
| Practice 5-1: Extending Domains by Using Templates | 24 |
| Practices for Lesson 6 | 26 |
| Practice 6-1: Getting Familiar with the Administration Console | 27 |
| Practice 6-2: Making Configuration Changes..... | 30 |
| Practice 6-3: Using WLST | 31 |
| Practices for Lesson 7 | 33 |
| Practice 7-1: Managing Managed Servers by Using the Administration Console | 34 |
| Practice 7-2: Adding Managed Servers by Using WLST | 37 |
| Practices for Lesson 8 | 39 |
| Practice 8-1: Adding Machines and Assigning Servers..... | 40 |
| Practice 8-2: Connecting to the Node Manager..... | 41 |
| Practice 8-3: Starting Managed Servers by Using the Node Manager | 45 |
| Practices for Lesson 9 | 46 |
| Practice 9-1: Configuring Logging Parameters | 47 |
| Practice 9-2: Examining Log Entries | 48 |
| Practices for Lesson 10 | 49 |
| Practices for Lesson 11 | 50 |
| Practice 11-1: Deploying Libraries | 51 |
| Practice 11-2: Deploying Applications | 54 |
| Practice 11-3: Performing Life Cycle Management of Applications | 56 |
| Practice 11-4: Enabling OHS as the Front End of Applications..... | 58 |
| Practices for Lesson 12 | 62 |
| Practice 12-1: Redeploying Unversioned Applications | 63 |
| Practice 12-2: Redeploying Versioned Applications | 66 |
| Practices for Lesson 13 | 69 |
| Practice 13-1: Creating JDBC Modules..... | 70 |
| Practice 13-2: Deploying JDBC Modules..... | 72 |
| Practice 13-3: Testing JDBC Modules | 73 |
| Practice 13-4: Creating JDBC Modules by Using Scripts | 75 |
| Practices for Lesson 14 | 76 |
| Practice 14-1: Configuring JMS Resources and Deploying the JMS Application | 77 |
| Practices for Lesson 15 | 82 |
| Practices for Lesson 16 | 83 |
| Practice 16-1: Initiating Clusters | 84 |

| | |
|---|-----|
| Practices for Lesson 17 | 90 |
| Practice 17-1: Targeting Applications to a Cluster | 91 |
| Practice 17-2: Configuring Session Replication by Using In-Memory Structures..... | 93 |
| Practices for Lesson 18 | 97 |
| Practice 18-1: Managing Users and Groups | 98 |
| Practice 18-2: Securing WebLogic Server Resources | 101 |
| Practices for Lesson 19 | 105 |
| Practice 19-1: Configuring Keystores..... | 106 |
| Practices for Lesson 20 | 109 |
| Practice 20-1: Backing Up the Configuration..... | 110 |
| Practice 20-2: Enabling Autobackup of config.xml | 111 |
| Practice 20-3: Performing Recovery..... | 112 |

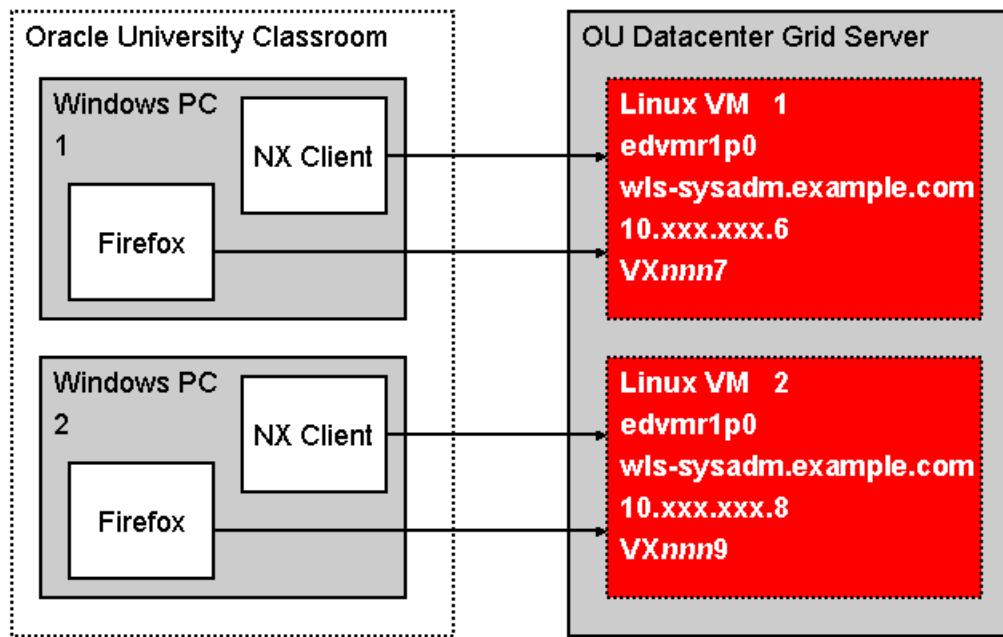
Practices for Lesson 1

Lab Familiarity

The lab uses a virtual machine grid to host your Linux environment. To use the lab environment, you use a client on the local PC to access the desktop on the remote PC. The instructor will give you the host names and IP addresses to be used by your team. The key tasks are:

- Logging on to the local PC
- Configuring the local client
- Starting the NoMachine client
- Logging on to the remote PC
- Arranging the remote desktop

Big Picture:



Although all the host names are identical, the numbered designators (including host aliases) are unique. There is no obvious correlation between the PC number and the VX number and the IP addresses.

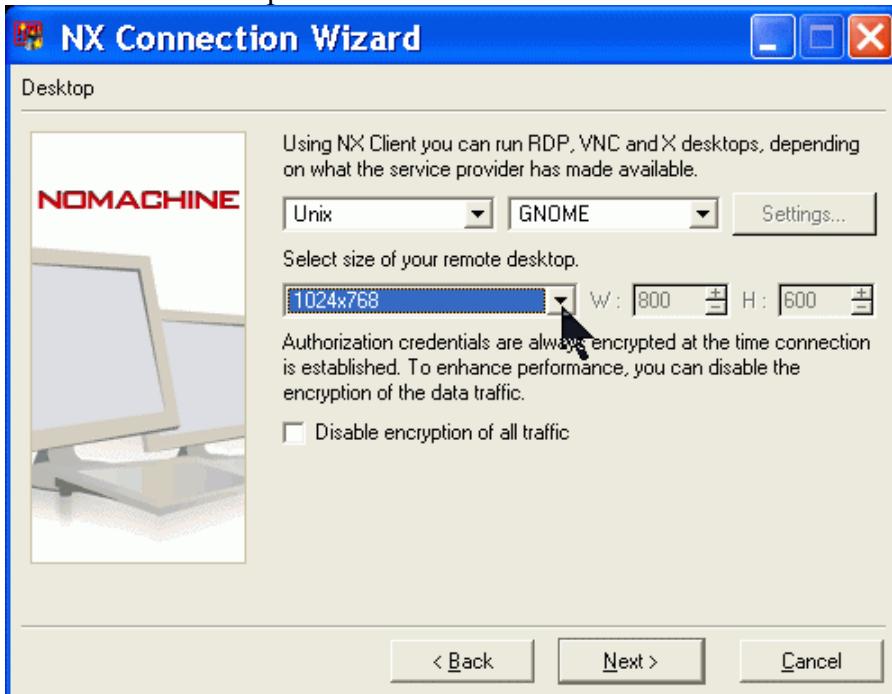
Practice 1-1: Connecting to the Classroom Grid

In this practice, you configure the remote desktop client software so that you can access and operate the remote Linux desktop.

- 1) Make sure that the local Windows PC is powered on. It should automatically log you in.
- 2) Your instructor will assign virtual machines for each lab team. Write down the following information:

| Field | Value |
|------------------------|-------|
| Host name | |
| Host IP address | |
| Username | |
| User password | |

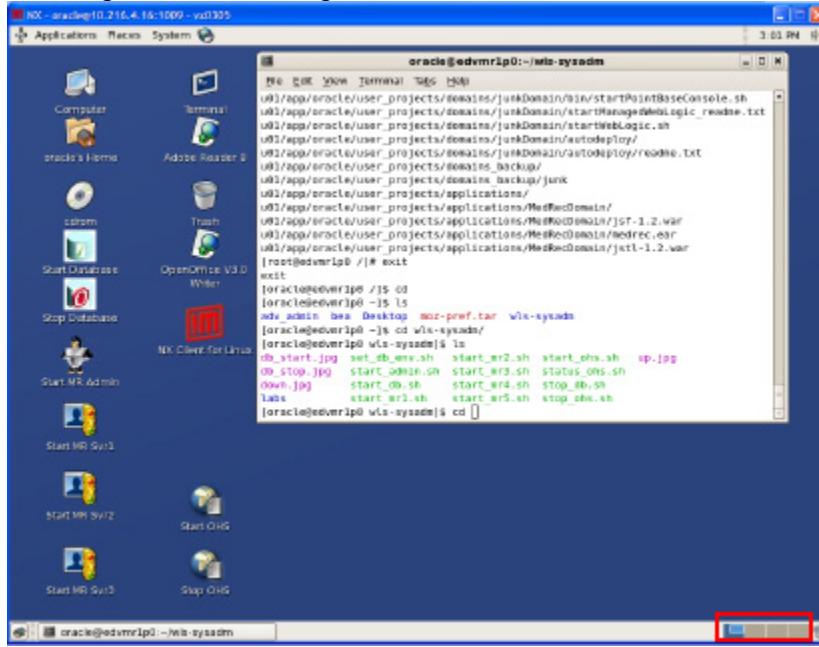
- 3) From your Windows Start menu, select **Start > All Programs > NX Client for Windows > NX Connection Wizard**. The Welcome page appears.
- 4) Click **Next**.
- 5) On the Session page, in the **Session** field, enter any name to identify this session—for example, **WLS-Labs** or your own name.
- 6) In the **Host** field, enter the host name given to you by the instructor. Leave all the other values and settings as the defaults. Click **Next**.
- 7) On the Desktop page, select **GNOME** in the second drop-down list. Change the size of the remote desktop to **1024x768**. Click **Next**.



- 8) Click **Finish**. This starts the client.

Practice 1-1: Connecting to the Classroom Grid (continued)

- 9) Enter your username and password as given to you by your instructor. The drop-down menu should have the Session name you picked earlier—for example, **WLS-Labs**. Click **Login**. A series of screens appear as the client contacts the remote desktop.
- 10) After you have connected to the remote desktop, you may want to set up the remote desktop for your labs.
- You will notice four palettes at the bottom right of your remote machine. Each of them represents a desktop on the remote machine.



- For your convenience, you can invoke different applications to full screen on each desktop appropriately.
 - You can invoke a Gnome terminal on the first desktop by using the Terminal icon on the desktop.

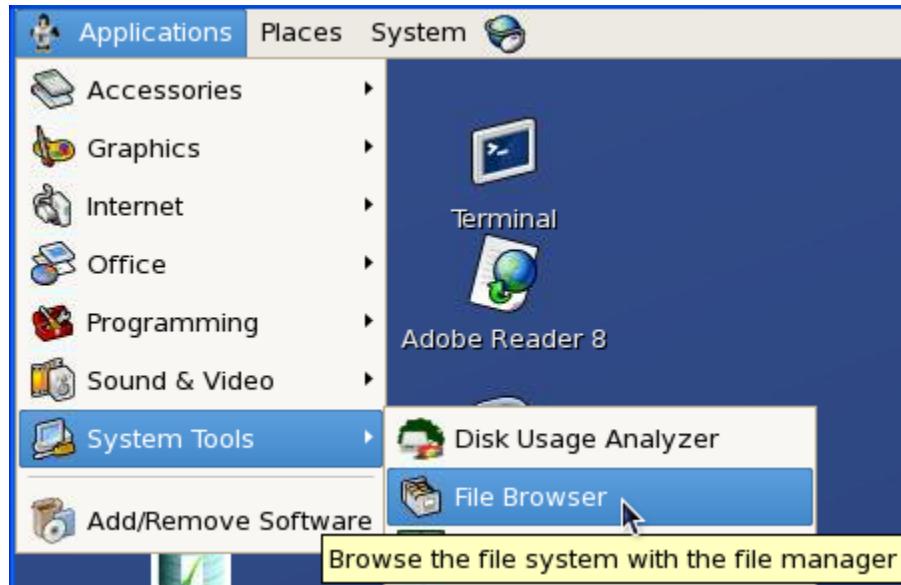


- Invoke a Web browser on the second desktop by using the icon on the menu bar.



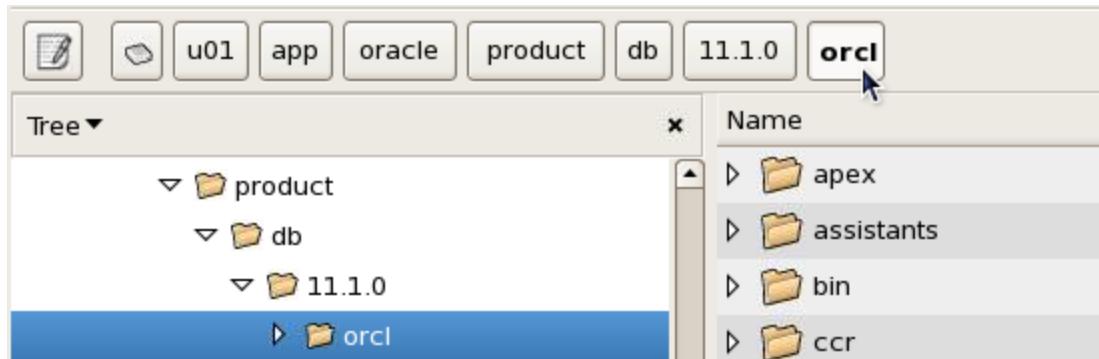
Practice 1-1: Connecting to the Classroom Grid (continued)

- iii) Invoke a file browser in the third desktop by using Applications > System Tools > File Browser.



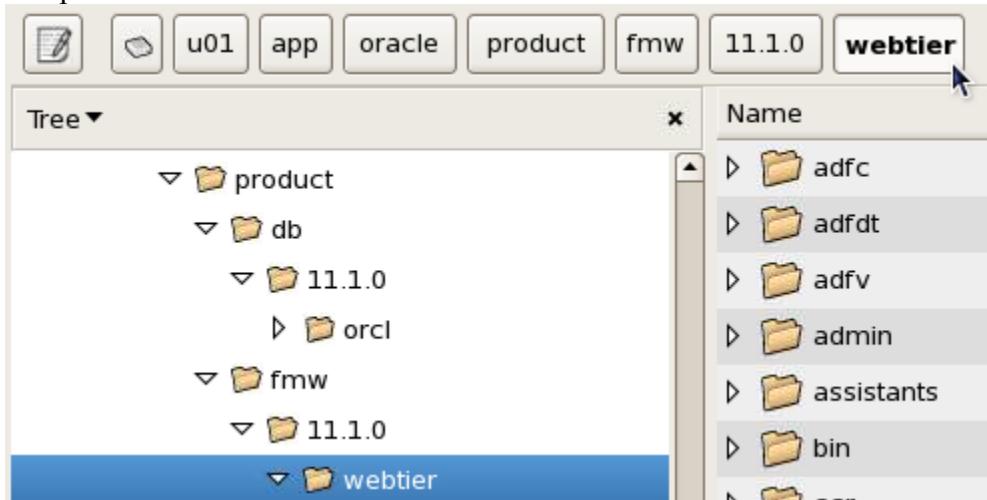
11) Oracle Database 11g and Oracle HTTP Server 11g have already been installed and configured in your remote machine. Using the File Browser desktop on the remote machine, navigate through the installation and configuration directories.

- a) On the desktop, with File Browser, navigate the File System tree to the /u01/app/oracle/product/db/11.1.0/orcl folder. This folder is the ORACLE_HOME folder for the database. The database executables are in this folder.

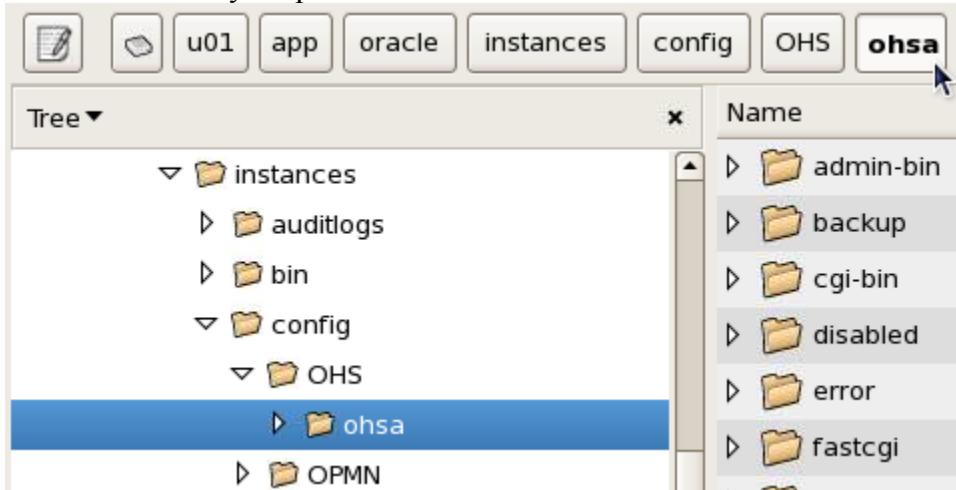


Practice 1-1: Connecting to the Classroom Grid (continued)

- b) Now navigate to /u01/app/oracle/product/fmw/11.1.0/webtier. This folder contains the installed binaries for the Oracle Fusion Middleware Web Tier components.



- c) Navigate to the /u01/app/oracle/instances/config/OHS/ohsa folder. This folder contains the Oracle HTTP Server configured in this machine. You will be using this OHS later in your practices.



12) Using the Gnome terminal session, perform the following steps to get familiar with the scripts that you will use in the practices.

- a) The wls-sysadm folder contains all the scripts and applications that you will use in the practices for this course. Navigate to the wls-sysadm subfolder in your \$HOME folder (/home/oracle) and list the files in this folder:

Practice 1-1: Connecting to the Classroom Grid (continued)

```
$> cd /home/oracle/wls-sysadm
$ wls-sysadm> ls
```

```
oracle@edvmr1p0:~/wls-sysadm
File Edit View Terminal Tabs Help
[oracle@edvmr1p0 ~]$ cd /home/oracle/wls-sysadm/
[oracle@edvmr1p0 wls-sysadm]$ ls
db_start.jpg  set_db_env.sh  start_mr3.sh  status_ohs.sh  up.jpg
db_stop.jpg   start_admin.sh  start_mr4.sh  stop_adm.sh
down.jpg      start_db.sh    start_mr5.sh  stop_db.sh
labs          start_mr1.sh   start_nm.sh   stop_mr1.sh
old_labs      start_mr2.sh   start_ohs.sh  stop_ohs.sh
[oracle@edvmr1p0 wls-sysadm]$
```

- b) Execute the `ps` command to check whether the database is running. If it is not running, use the `start_db.sh` script to start the database.

```
$> ps -ef | grep pmon
```

- i) If the preceding command returns two rows in response as follows, the database is running.

```
[oracle@edvmr1p0 wls-sysadm]$ ps -ef | grep pmon
oracle  24542     1  0 15:30 ?        00:00:00 ora_pmon_orcl
oracle  24584  8614  0 15:31 pts/0    00:00:00 grep pmon
[oracle@edvmr1p0 wls-sysadm]$
```

- ii) If the preceding command does not return the `ora_pmon_orcl` row, you need to start the database using the `start_db.sh` script in the `wls-sysadm` subfolder:

```
$ wls-sysadm> ./start_db.sh
```

13) Close the NX Client window. Click **Disconnect**. This allows you to resume where you left off the next time. Note that if you click **Terminate**, you may have to set up your remote desktop environment (in the preceding step 10) again.

14) In later labs, you will be able to use the local Web browser as well as the remote Web browser for accessing the WebLogic Server Administration Console.

Practices for Lesson 2

There are no practices for Lesson 2.

Practices for Lesson 3

Installing Oracle WebLogic Server 11g

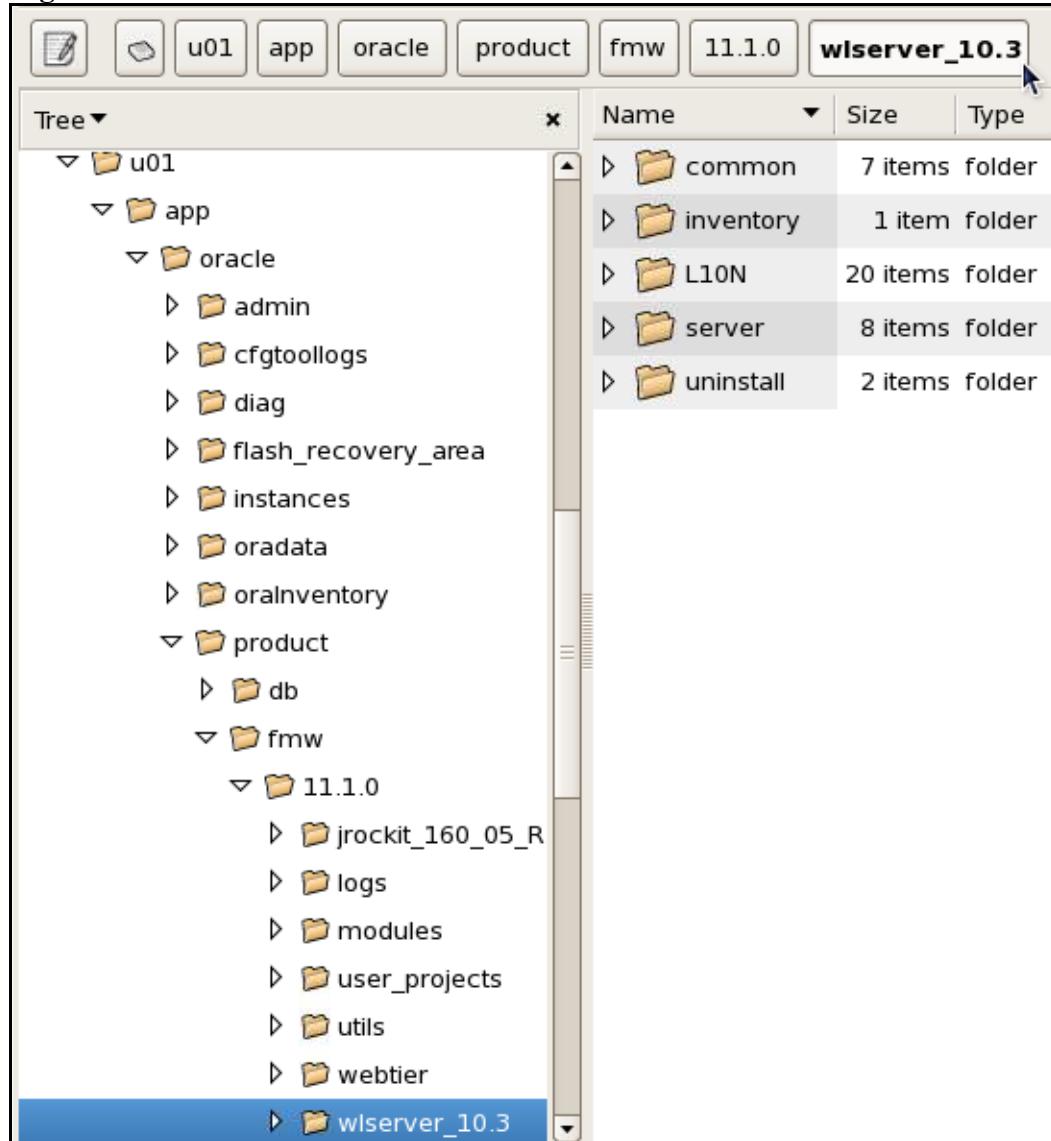
As the administrator of middle-tier computing for The Example Corp, you install Oracle WebLogic Server on a single Linux machine using many of the default options to test the basic functionality of simple configurations.

The key tasks in this practice session are:

- Installing Oracle WebLogic Server with JRockit as the Java Virtual Machine
- Navigating the installed WebLogic Server folder structure

Successful completion of this practice is essential for performing subsequent practices.

Big Picture:



Practice 3-1: Installing Oracle WebLogic Server

In this practice, you install Oracle WebLogic Server version 10.3.1 into an existing directory structure that contains, among other things, Oracle Database 11g. You install Oracle WebLogic Server in this session but configure a domain later on.

- 1) Log on to your remote Linux desktop as the **oracle** user.
- 2) Open the desktop with the terminal window and navigate to the /modules/stage/WLS folder that contains the Oracle WebLogic Server installable.
- 3) Run the Linux 32-bit installer by entering (be mindful of the leading dot which you need to enter):

```
$> ./wls1031_ccjk_linux32.bin
```

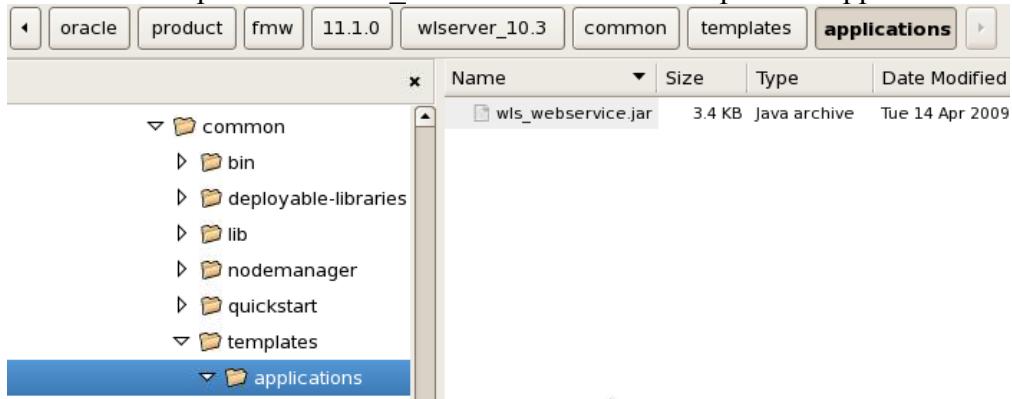
- 4) Use the following table for installing Oracle WebLogic Server:

| Step | Screen/Page Description | Choices or Values |
|-------------|---|---|
| a. | Welcome | Click Next. |
| b. | Choose Middleware Home Directory | Select “Create a new Middleware Home.” In Middleware Home Directory, enter /u01/app/oracle/product/fmw/11.1.0, or browse to it because it already exists. Click Next. When the Warning dialog box indicating that the directory is nonempty appears, click Yes to proceed. |
| c. | Register for Security Updates | Even though you would register in real life, for the lab, deselect the check box to opt out of the security updates. In the “Are you sure?” dialog box, click Yes. Click Next. |
| d. | Choose Install Type | Select Custom. Click Next. |
| e. | Choose Products and Components | Do not select Server Examples. Click Next. |
| f. | JDK Selection | Select only Oracle JRockit 1.6.0_05 and click Next. |
| g. | Choose Product Installation Directories | Accept the defaults. Click Next. |
| h. | Installation Summary | Make sure that only one JDK (JRockit) is present. Click Next. The progress bar appears and displays the progress from 0 to 100%. |
| i. | Installation Complete | Deselect Run Quickstart and click Done. |

Practice 3-2: Navigating the WLS Installation Directories

In this practice, you locate the key directories that are used in the later labs. You do not have to do anything with these files; just make a note of where you find them.

- 1) Note the WL_HOME (the location where WLS is installed) and explore some of the important folders and files in your WLS installation.
 - a) Using the remote desktop with File Browser, navigate to the folder (u01 > app > oracle > product > fmw > 11.1.0 > wlserver_10.3) where you have installed WLS.
 - b) Locate the templates in <WL_HOME> common > templates > applications.



- c) Similarly, locate common > bin in <WL_HOME> and view the list of configuration scripts.
- 2) Using the remote desktop with Gnome Terminal, view the `setWLSEnv.sh` script to see which environment variables are set. Then run the script and verify that the variables are appropriately set.

Practice 3-2: Navigating the WLS Installation Directories (continued)

- a) Look at the comments in the `setWLSEnv.sh` file in `/u01/app/oracle/product/fmw/11.1.0/wlserver_10.3/server/bin`.
- ```
[oracle@edvmrlp0 WLS]$ cd /u01/app/oracle/product/fmw/11.1.0/wlserver_10.3/server/bin
[oracle@edvmrlp0 bin]$ more setWLSEnv.sh
#!/bin/sh
#####
This script is used to set up your environment for development with WebLogic
Server. It sets the following variables:
#
WL_HOME - The root directory of your WebLogic installation
JAVA_HOME - Location of the version of Java used to start WebLogic
Server. This variable must point to the root directory of a
JDK installation and will be set for you by the installer.
See the Oracle Fusion Middleware Supported System Configuration
page
(http://www.oracle.com/technology/software/products/ias/files/
fusion_certification.html)
for an up-to-date list of supported JVMs on your platform.
PATH - Adds the JDK and WebLogic directories to the system path.
CLASSPATH - Adds the JDK and WebLogic jars to the classpath.
#
Other variables that setWLSEnv takes are:
#
PRE_CLASSPATH - Path style variable to be added to the beginning of the
CLASSPATH
POST_CLASSPATH - Path style variable to be added to the end of the
CLASSPATH
```

- b) Run the following script: (Use `source` to ensure that the variables are set for the entire session and not just within the script shell itself.)

```
$> source ./setWLSEnv.sh
```

- c) Verify the values of the new environment variables by entering:

```
echo $WL_HOME
echo $MW_HOME
echo $JAVA_HOME
echo $ANT_HOME
```

```
[oracle@edvmrlp0 bin]$ echo $WL_HOME
/u01/app/oracle/product/fmw/11.1.0/wlserver_10.3
[oracle@edvmrlp0 bin]$ echo $MW_HOME
/u01/app/oracle/product/fmw/11.1.0
[oracle@edvmrlp0 bin]$ echo $JAVA_HOME
/u01/app/oracle/product/fmw/11.1.0/jrockit_160_05_R27.6.2-20
[oracle@edvmrlp0 bin]$ echo $ANT_HOME
/u01/app/oracle/product/fmw/11.1.0/modules/org.apache.ant_1.7.0
[oracle@edvmrlp0 bin]$ █
```

This makes navigating the directories much faster and less prone to typographical errors. You use this script to set your environment variables for every lab from this point forward. It needs to be done only once per session.

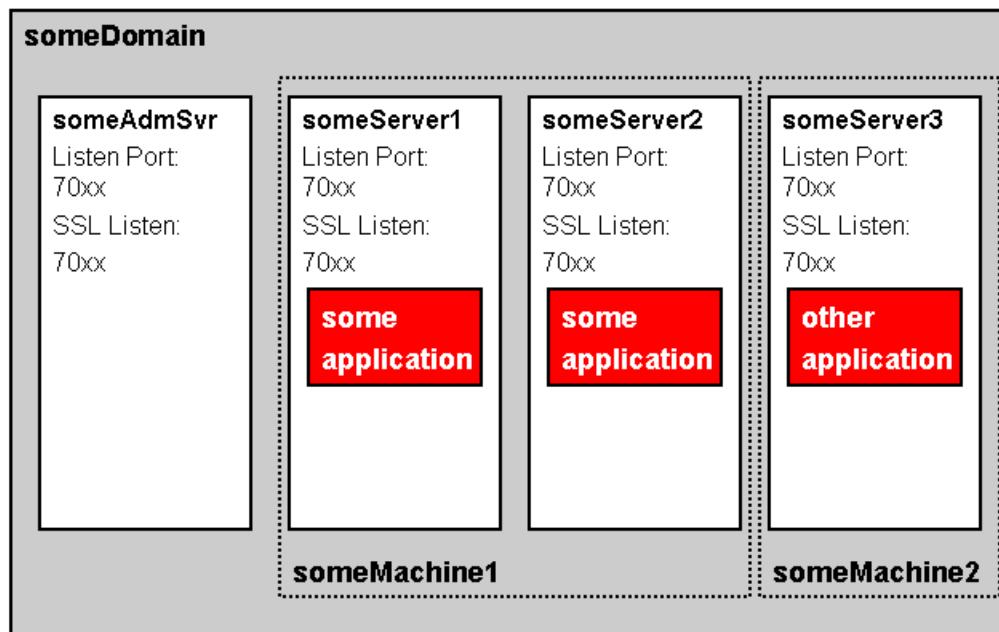
## Practices for Lesson 4

### Configuring a Domain

As the administrator of the middleware, you name the domains and servers. The application is a Medical Records system for a doctor's office, so you decide on a "MedRec" prefix for most names. The application is from a software company named Avitek, so you will see that name appear on Web page banners. This system uses Web clients and a back-end database. Your first task is to create the total application environment, a "domain." The domain references the database, but does not include the database. All domains require some common elements, so if the creation of a domain can also make those other pieces (servers of various sorts), then you choose time-saving procedures. Besides, you can always come back later and either modify the servers created at this time or create other servers at a later date. The key tasks are:

- Creating a minimal domain from scratch
- Creating a domain to support a particular application template
- Starting the administration server
- Stopping the administration server

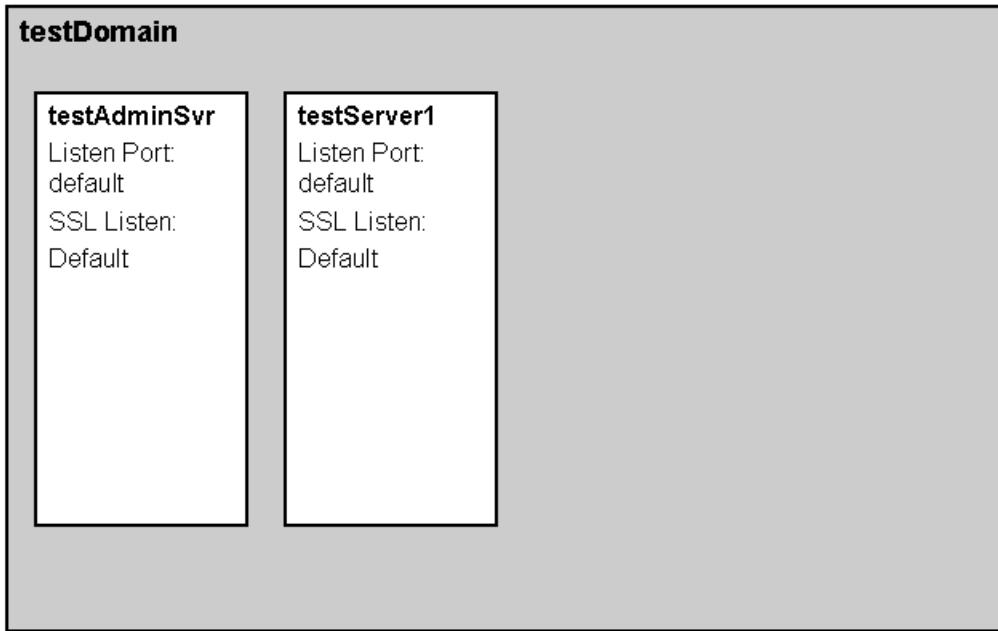
### Big Picture:



## **Practice 4-1: Creating a Minimal Domain from the Beginning**

In this practice, you make a simple domain named `test` as an experiment. After you prove that this works, you will not use this `test` domain any more. The purpose of this lab is to see the Configuration Wizard screens for the first time and their default values.

### **Big Picture:**



- 1) Using the Configuration Wizard, configure a domain with the following parameters:

| Screen/Page Description             | Choices or Values                                   |
|-------------------------------------|-----------------------------------------------------|
| Domain Name                         | <code>testDomain</code>                             |
| Location                            | <code>/u01/app/oracle/user_projects/domains/</code> |
| Administrative User name / password | <code>weblogic/Welcome1</code>                      |
| Start Mode / JDK                    | Production Mode/JRockit                             |
| Name of Administration Server       | <code>testAdminSvr</code>                           |
| Managed Servers                     | <code>testServer</code>                             |

- a) In a gnome-terminal session of the remote machine, navigate to the common binaries subfolder of your WebLogic Server installation and run the configuration assistant:

```
$> cd
/u01/app/oracle/product/fmw/11.1.0/wlserver_10.3/common/bin
$>./config.sh
```

- b) Specify the following values on the Configuration Wizard pages. Note that most values are case-sensitive:

## **Practice 4-1: Creating a Minimal Domain from the Beginning (continued)**

|    | Screen/Page Description                        | Choices or Values                                                                                                                                                                                                                                                   |
|----|------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| a. | Welcome                                        | Select “Create a new WebLogic domain.” Click Next.                                                                                                                                                                                                                  |
| b. | Select Domain Source                           | Do not select any other component. Basic WebLogic Server Domain is already selected. Click Next.                                                                                                                                                                    |
| c. | Specify Domain name and Location               | In Domain name, enter <b>testDomain</b> . All names are case-sensitive.<br>In the Domain location, change it to /u01/app/oracle/user_projects/domains. The idea is to separate configuration data from the executables. If the directory does not exist, create it. |
| d. | Configure Administrator User name and Password | User name: <b>weblogic</b><br>User Password: <b>Welcome1</b><br>Confirm password: <b>Welcome1</b><br>Description: (leave the default)<br>Click Next.                                                                                                                |
| e. | Configure Server Start Mode and JDK            | Select Production Mode. Click Next.                                                                                                                                                                                                                                 |
| f. | Select Optional Configuration                  | Select Administration Server and Managed Servers, Clusters and Machines. Click Next.                                                                                                                                                                                |
| g. | Configure the Administration Server            | Change the name to <b>testAdminSvr</b> . Click Next.                                                                                                                                                                                                                |
| h. | Configure Managed Servers                      | Click Add. Change the name to <b>testServer1</b> . Click Next.                                                                                                                                                                                                      |
| i. | Configure Clusters                             | There will not be any clusters on this simple domain. Click Next.                                                                                                                                                                                                   |
| j. | Configure Machines                             | There will not be any machines on this simple domain. Click Next.                                                                                                                                                                                                   |
| k. | Configuration Summary                          | Notice the two servers that you renamed. Everything should have a prefix of test. See the following screenshot. Click Create.                                                                                                                                       |
| l. | Creating Domain                                | After the domain is created successfully, click Done.                                                                                                                                                                                                               |



## ***Practice 4-1: Creating a Minimal Domain from the Beginning (continued)***

- 2) View the configuration details of the domain that you have created:
  - a) Navigate to the folder of the domain that you just created. List the files and folders just created:

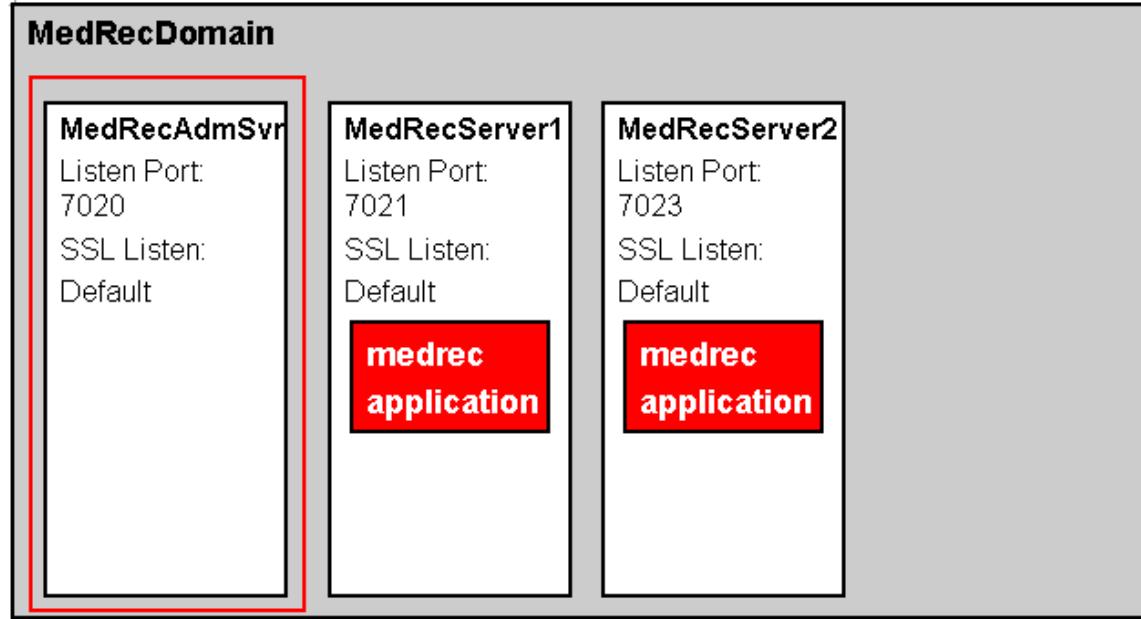
```
$> cd /u01/app/oracle/user_projects/domains/testDomain
$> ls -l
```
  - b) View config.xml in the config subfolder. This file contains the specifications for the domain that you just configured. Look for the names testDomain, testAdminSvr, and testServer1. Note the listen ports and the encrypted password values.
- 3) Start and stop the administration server for the domain.
  - a) Navigate to the test domain folder and run ./startWebLogic.sh to start the administration server. The username is weblogic and the password is Welcome1. (Note the “one” at the end of the password is a number and not the letter L.) The password will not be displayed. Make sure that the last message is <Server started in RUNNING mode>.
  - b) From a new terminal session, stop the administration server by running stopWebLogic.sh from the bin folder of the domain.

```
$> cd /u01/app/oracle/user_projects/domains/testDomain/bin
$> ./stopWeblogic.sh
```
  - c) You have now finished creating the test domain, and you will not need it for the rest of the class.

## Practice 4-2: Creating a Functional Domain

In this practice, you create a domain to support the Medical Records (Medrec) application. You create the domain and later extend it using the application template. **Successful completion of this lab is a prerequisite for the remaining labs.**

**Big Picture:**



- 1) Using the Configuration Wizard, configure a domain with the following parameters:

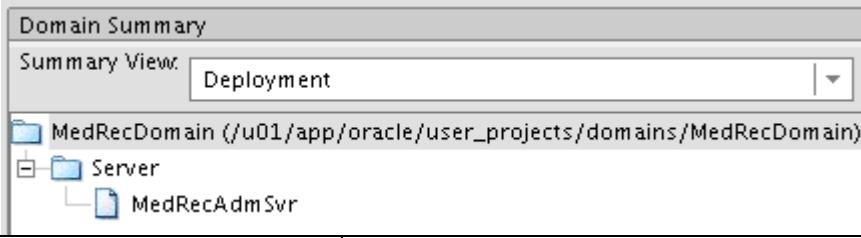
| Screen/Page Description             | Choices or Values                      |
|-------------------------------------|----------------------------------------|
| Domain Name                         | MedRecDomain                           |
| Location                            | /u01/app/oracle/user_projects/domains/ |
| Administrative User name / password | weblogic/Welcome1                      |
| Start Mode / JDK                    | Production Mode/JRockit                |
| Administration Server               | MedRecAdmSvr Port 7020                 |
| Managed Servers                     |                                        |
| Machines                            |                                        |

- a) In a gnome terminal session desktop of the remote machine, navigate to the common binaries subfolder of your WebLogic Server installation and run the configuration assistant:

```
$> cd $WL_HOME/common/bin
$> ./config.sh
```

## **Practice 4-2: Creating a Functional Domain (continued)**

- 2) Specify the following values on the Configuration Wizard pages:

| <b>Step</b> | <b>Screen/Page Description</b>                 | <b>Choices or Values</b>                                                                                                                                                     |
|-------------|------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| a.          | Welcome                                        | Select “Create a new WebLogic domain.” Click Next.                                                                                                                           |
| b.          | Select Domain Source                           | Do not select any other component. Basic WebLogic Server Domain is already selected. Click Next.                                                                             |
| c.          | Specify Domain Name and Location               | In Domain name, enter <b>MedRecDomain</b> . All names are case-sensitive.<br>In the Domain location, change it to <b>/u01/app/oracle/user_projects/domains</b> . Click Next. |
| d.          | Configure Administrator User name and Password | User name: <b>weblogic</b><br>User Password: <b>Welcome1</b><br>Confirm password: <b>Welcome1</b><br>Description: (leave the default)<br>Click Next.                         |
| e.          | Configure Server Start Mode and JDK            | Select Production Mode. Click Next.                                                                                                                                          |
| f.          | Select Optional Configuration                  | Select Administration Server and click Next.                                                                                                                                 |
| g.          | Configure the Administration Server            | Change the name to <b>MedRecAdmSvr</b> . Change the Listen port to <b>7020</b> . Leave SSL disabled. Click Next.                                                             |
| h.          | Configuration Summary                          | Notice the administration server. Everything should have a prefix of MedRec. Click Create.                                                                                   |
| i.          |                                                |                                                                                          |
| j.          | Creating Domain                                | After the domain is created successfully, click Done.                                                                                                                        |

- 3) Navigate to the domain that you just created. List the files and folders just created:

```
$> cd /u01/app/oracle/user_projects/domains/MedRecDomain
$> ls -l
```

## **Practice 4-2: Creating a Functional Domain (continued)**

- 4) View the config.xml file in the config subfolder. This file contains the specifications for the domain that you just configured. Look for the names MedRecDomain and MedRecAdminSvr. Note the listen ports and the encrypted values.

```

<credential-encrypted>{AES}3sAW0MKV2z0Rwg0RM2S4Z43810maTzDRSQZAuf+eiUg7Qa96l
MjcroZ/D5nF0codMF5AgKzFlQiur54J1Qn4cw3nWWwa5c6VsyeJpOCMVhSOAyYmPRsvxDSJeY9IQfi<
/credential-encrypted>
<node-manager-username>6P4E19Av3m</node-manager-username>
<node-manager-password-encrypted>{AES}2ClrMZY5P8mk0GNGoDeYwAGTaDRbsTMuSaG9Ej
5siKw=</node-manager-password-encrypted>
</security-configuration>
<server>
<name>MedRecAdmSvr</name>
<listen-port>7020</listen-port>
<listen-address/>
</server>
<production-mode-enabled>true</production-mode-enabled>
<embedded-ldap>
<name>MedRecDomain</name>
<credential-encrypted>{AES}P0ySlj4JE26oqWbTSRIRgIA5H5SdzyrlqzVcImKkuVgB25KFM
U/5ieKrHSnbr29f</credential-encrypted>
</embedded-ldap>
<configuration-version>10.3.1.0</configuration-version>
<admin-server-name>MedRecAdmSvr</admin-server-name>
</domain>
```

- 5) Start the administration server to test that it starts correctly.
- You can use the Start MR Admin icon. Alternatively, in your gnome terminal session, navigate to the domain folder (`/u01/app/oracle/user_projects/domains/MedRecDomain`) and execute the `./startWebLogic.sh` command.
  - Enter the username `weblogic` and password `Welcome1` when prompted. The password will not be displayed, and the password is case-sensitive.
  - Make sure that the last message is  
`<Server started in RUNNING mode>`.
- 6) Stop the administration server process by using the `stopWebLogic.sh` script in the bin subfolder of your domain. You are going to add more functions to this domain during the rest of the labs.

```
$> cd
/u01/app/oracle/user_projects/domains/MedRecDomain/bin
$> ./stopWebLogic.sh
```

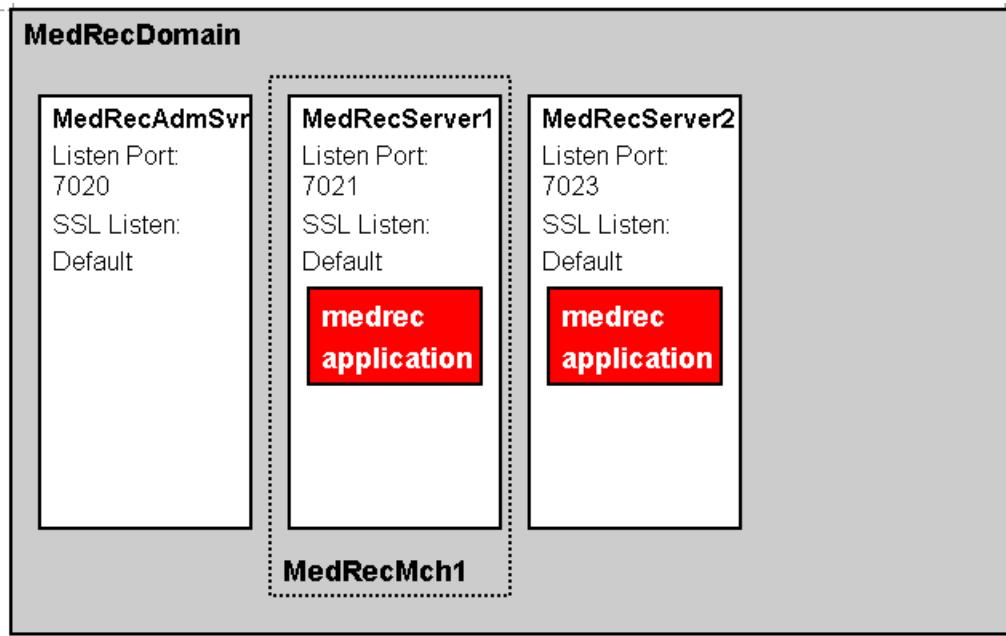
## Practices for Lesson 5

### Extending Domains Using Templates

The application programmers have created a custom application template for you to do this. This will be similar to the previous lab titled “Creating a Minimal Domain from the Beginning,” but with more options selected. Although it is possible to create a domain template, you only have an application template. So you need to create the domain itself and then come back and extend the domain with the application template in the next lab. The key tasks are:

- Identifying the domain to be extended
- Identifying the template with which to extend the domain
- Running the Configuration Wizard to do the extension

### Big Picture:



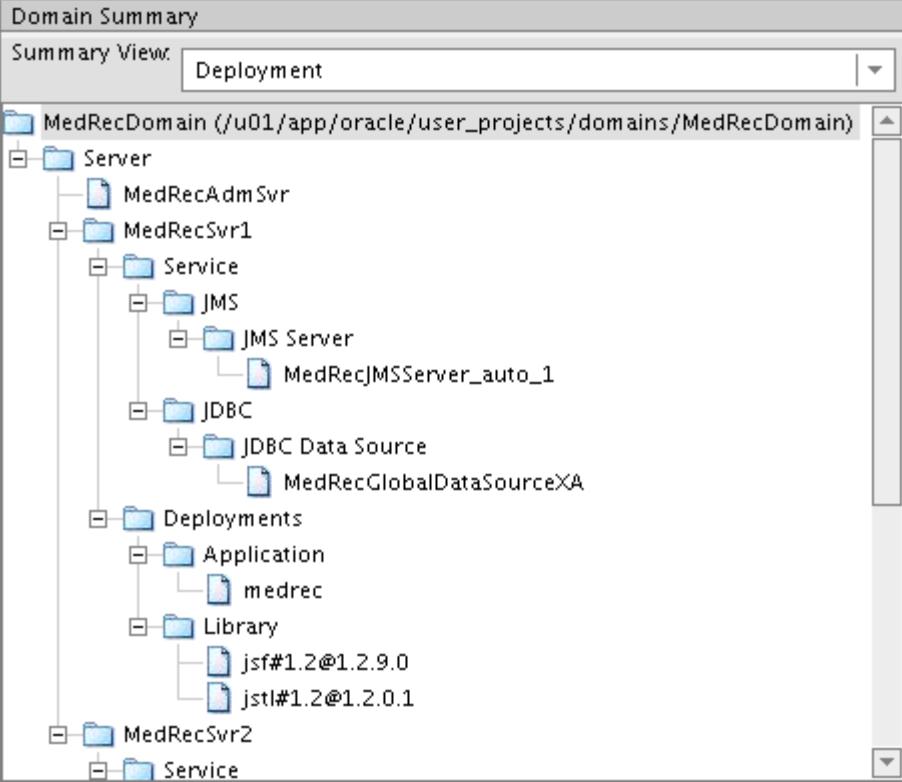
## Practice 5-1: Extending Domains by Using Templates

In this practice, you extend the existing domain with an application template. The domain itself (for example, the administration server and managed servers) does not change; this simply adds more functions as specified by the application developer.

- 1) Navigate to \$WL\_HOME/common/bin. Start the Configuration Wizard the same way you did in the previous lab, but this time for the application part of the domain. Enter ./config.sh.
- 2) Specify the following values in the Configuration Wizard pages:

| Step | Screen/Page Description                                                                                                                                                                                                                                                              | Choices or Values                                                                                                                                                                                    |
|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| a.   | Welcome                                                                                                                                                                                                                                                                              | Select “Extend an existing WebLogic domain.” Click Next.                                                                                                                                             |
| b.   | Select a WebLogic Domain Directory                                                                                                                                                                                                                                                   | Navigate to /u01/app/oracle/user_projects/domains/MedRecDomain. Note that the valid targets have a blue jar icon on the folders. Click Next.                                                         |
| c.   | Select Extension Source                                                                                                                                                                                                                                                              | Select “Extend my domain using an existing extension template.” Enter /home/oracle/wls-sysadm/labs/Lab05/MedRecResources.jar as the location (or you can browse to it). Click Next.                  |
|      | <input checked="" type="radio"/> Extend my domain using an existing extension template<br>Template location: <input type="text" value="/home/oracle/wls-sysadm/labs/Lab05/MedRecResources.jar"/> <input type="button" value="Browse"/>                                               |                                                                                                                                                                                                      |
|      | If you get Conflict Detected                                                                                                                                                                                                                                                         | Select “Keep existing component.” Select “Apply this selection if further conflicts are detected.” Click OK.                                                                                         |
| d.   | Specify Domain name and Location                                                                                                                                                                                                                                                     | Change the Application location to /u01/app/oracle/user_projects/applications. Click Next.                                                                                                           |
|      | Domain name: <input type="text" value="MedRecDomain"/><br>Domain location: <input type="text" value="/u01/app/oracle/user_projects/domains"/><br>Application location: <input type="text" value="/u01/app/oracle/user_projects/applications"/> <input type="button" value="Browse"/> |                                                                                                                                                                                                      |
| e.   | Configure JDBC Data Sources                                                                                                                                                                                                                                                          | Accept the defaults and click Next. Ensure that the database is running.                                                                                                                             |
| f.   | Test JDBC Data Sources                                                                                                                                                                                                                                                               | If the test is successful, click Next.                                                                                                                                                               |
| g.   | Select Advanced Configuration                                                                                                                                                                                                                                                        | These are the same screens you saw earlier when you created MedRecDomain, so there is no need to do anything additional. Click Next.                                                                 |
| h.   | Configuration Summary                                                                                                                                                                                                                                                                | The Deployment view shows all the applications and libraries that have been deployed. The administrator did not have to know anything about them; it was all included in the template. Click Extend. |

## Practice 5-1: Extending Domains by Using Templates (continued)

| Step | Screen/Page Description                                                             | Choices or Values                               |
|------|-------------------------------------------------------------------------------------|-------------------------------------------------|
|      |  |                                                 |
| i.   | Creating Domain                                                                     | When the Progress bar reaches 100%, click Done. |

- 3) Use the Start MR Admin icon on the desktop to start the administrative server of MedRecDomain. Enter the username (weblogic) and password (Welcome1) in the Admin Server terminal window. Make sure that the last message is <Server started in RUNNING mode>.
- a) Stop the administration server by running `stopWebLogic.sh` from the bin folder of the domain.

```
$> cd /u01/app/oracle/user_projects/domains/MedRecDomain/bin
$> ./stopWeblogic.sh
```

## Practices for Lesson 6

### Using the Administration Console and WLST

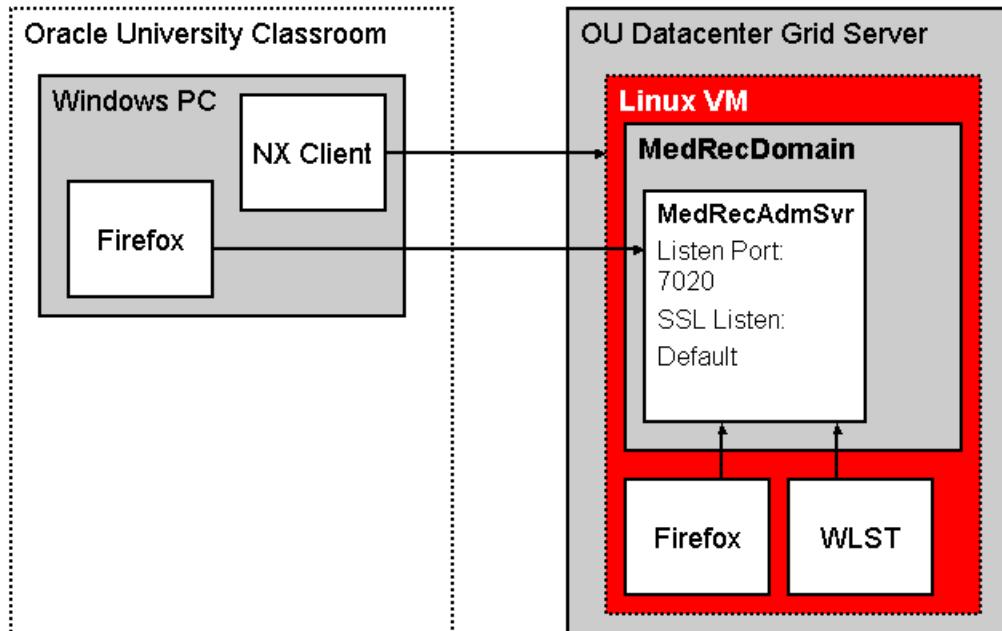
There are two main interfaces to configure the managed servers:

- Web-based graphical user interface, namely Administration Console
- Command-line interface, namely WebLogic Scripting Tool (WLST)

The key tasks covered in this practice include:

- Signing on to the Administration Console
- Making configuration changes using the Administration Console
- Invoking WLST and connecting to a domain
- Making configuration changes using WLST

#### Big Picture:



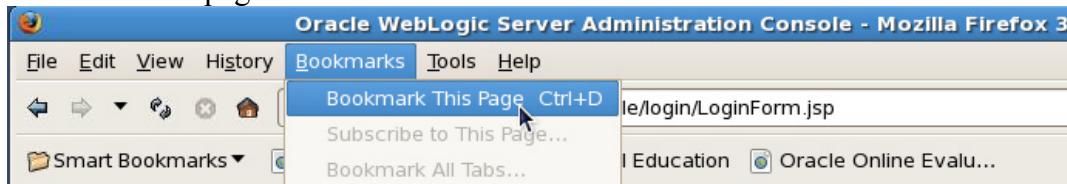
## Practice 6-1: Getting Familiar with the Administration Console

In this practice, you navigate the Administration Console using a Web browser. The main skills you learn are the terminology and the shortcuts.

- 1) Use the Start MR Admin icon on the desktop to start the administration server of MedRecDomain. Enter the username (`weblogic`) and password (`Welcome1`) in the Admin Server terminal window. Make sure that the last message is `<Server started in RUNNING mode>`.
- 2) In the Web browser, access the URL:

`http://wls-sysadm:7020/console`

- a) Bookmark this page.



- b) Log on with `weblogic` as the username and `Welcome1` as the password. If the browser offers to remember the password, click Yes.
- 3) In a gnome terminal session, use the `/sbin/ifconfig` command to find the IP address assigned to the Ethernet adapter 0. It should be a private address in the form `10.x.y.z`. For example:

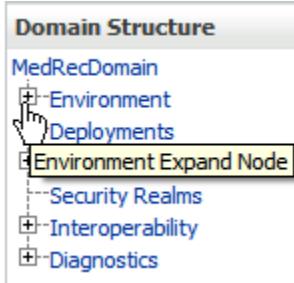
```
[oracle@edvmr1p0 /]$ /sbin/ifconfig ↗
eth0 Link encap:Ethernet HWaddr AA:A0:B0:00:03:05
 inet addr:10.216.4.16 Bcast:10.216.7.255 Mask:255.255.252.0
 inet6 addr: fe80::a8a0:b0ff:fe00:305/64 Scope:Link
 UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
 RX packets:269277 errors:0 dropped:0 overruns:0 frame:0
 TX packets:122027 errors:0 dropped:0 overruns:0 carrier:0
 collisions:0 txqueuelen:1000
 RX bytes:85294647 (81.3 MiB) TX bytes:21652134 (20.6 MiB)

eth1 Link encap:Ethernet HWaddr AA:A1:B0:00:03:05
```

- 4) On the Windows (local) desktop, open a Web browser (you can use Internet Explorer or Firefox) and access the URL `http://10.x.y.z:7020/console`, where `10.x.y.z` is the address you found in the previous step. Log on with `weblogic` as the username and `Welcome1` as the password. If the browser offers to remember the password, click Yes. Bookmark this page. Note that two people can sign on at the same time with the same username.

## Practice 6-1: Getting Familiar with the Administration Console (continued)

- 5) The main navigation is by expanding the plus or collapsing the minus icons in the Domain Structure on the left of the browser.



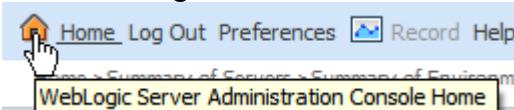
For example, click next to Environment (it will turn into ) , then click Servers, and you should see that the MedRecAdmSvr is RUNNING. If it were not running, you would not have any Administration Console! Note that the New, Clone, and Delete options are disabled. Alternatively, you can collapse all levels, click Environment (still showing ) , and then click Servers in the table at the right under Summary of Environment. This is one way to display the Summary of Servers table.

- 6) After you have gone to several pages in the Administration Console, you can see the breadcrumbs on the top showing the items you have navigated to. It looks like:

[Home](#) > [Summary of Servers](#) > [Summary of Environment](#) > [Summary of Servers](#)

It is historical, not hierarchical. The same menu item could be in there multiple times. Clicking one of those entries will take you back to that item, but with refreshed data.

- 7) Click WebLogic Server Administration Console Home.



The Home page gives yet another way to get to the same Summary of Servers page. Click Servers in the middle of the Home page.

- 8) You can refresh the entire Web page just as you would with any browser, or you can set some tables to autorefresh. On the Summary of Servers table, there is a cycle symbol that will make the table refresh repeatedly. Click the symbol. While refreshing, the cycle icon spins and the last refresh date/time is displayed. Click it again to make it stop.

- 9) In the Domain Structure, click MedRecDomain. There are two levels of tabs shown.

The screenshot shows the Administration Console interface with two main panels. On the left, the 'Domain Structure' sidebar shows 'MedRecDomain' expanded, with 'Environment' and 'Deployments' visible. A 'Lock & Edit' button is highlighted in this section. On the right, the 'Settings for MedRecDomain' page is displayed. It has a top navigation bar with tabs: Configuration, Monitoring, Control, Security, Web Service Security, Notes, General, JTA, EJBs, Web Applications, Logging, and Log Filters. A message box says 'Click the Lock & Edit button in the Change Center to modify the settings on this page.' A 'Save' button is at the bottom. Both the 'Lock & Edit' button in the sidebar and the message box on the right both have a red box around them.

## ***Practice 6-1: Getting Familiar with the Administration Console (continued)***

If you shrink the browser window so that the tabs would be impacted, they wrap to the next line and there is a blue bar to separate the upper- and the lower-level tabs. As you select different upper tabs, the lower tabs change. The Notes tab enables you to document configuration changes.

- 10) Scroll down in the Settings for MedRecDomain > Configuration > General. At the bottom of many pages is an Advanced toggle  **Advanced** . By clicking it, you can see an additional set of configuration parameters. Clicking it again will hide the advanced options.

## Practice 6-2: Making Configuration Changes

In this practice, you make changes to the active configuration. At this point, the change will be trivial. You are going to change the Administration Servers Logging Rotation file size from 5000 to 5001.

- 1) Change the Administration Servers Logging Rotation file size from 5000 to 5001.
  - a) Navigate to the Summary of Servers table from the previous practice and click MedRecAdmSvr (admin).
  - b) Click the Logging tab. Note that the Rotation file size 5000 is disabled.
  - c) In the Change Center, click Lock & Edit. Note that now the Rotation file size 5000 (as well as all the other options) becomes enabled. If you made a mistake, you can click Release Configuration, and it goes back the way it was, similar to Cancel. The Lock must be done before most configuration changes.
  - d) Change the Rotation file size 5000 to **5001**. Click Save.
  - e) In the Change Center, click Activate Changes.
  - f) Note the Messages panel at the top. It indicates that no restarts are necessary. Some changes may be effective only when the server is restarted. In this case, nothing needed to be restarted. Note that Activate Changes also releases the lock.



- 2) In the Change Center, click Lock & Edit again. Change the Rotation file size 5001 back to **5000**. Click Save. Do *not* activate anything.
- 3) In the Change Center, click "View changes and restarts." Here, you can selectively undo changes. Click Undo All Changes.

## Practice 6-3: Using WLST

In this practice, you use the WebLogic Scripting Tool (WLST) to get some status information from the running domain. Because it is a script, it can be saved and run over and over again as well as scheduled to run at specified times (such as the cron job in Linux).

- 1) Create a WLST script to change the value for the log rotation file size from 5001 to 5002.

- a) Open a terminal session on the Linux desktop. To ensure that the environment variables are set, run the `setWLSEnv.sh` script from your `WL_HOME/server/bin` folder.

```
$ cd
/u01/app/oracle/product/fmw/11.1.0/wlserver_10.3/server/bin
$ source ./setWLSEnv.sh
```

- b) Invoke WLST as follows: (Remember that the Java executables are case-sensitive.)

```
$ java weblogic.WLST
```

- c) Connect to the running administration server by entering:

```
wls:/offline> connect('weblogic','Welcome1','t3://wls-
sysadm:7020')
```

- d) Browse using UNIX-like commands. You should see the administration server and the two managed servers:

```
wls:/MedRecDomain/serverConfig> cd('Servers')
wls:/MedRecDomain/serverConfig/Servers> ls()
dr-- MedRecAdmSvr
dr-- MedRecSvr1
dr-- MedRecSvr2
```

```
wls:/MedRecDomain/serverConfig/Servers>
```

- e) Back up a level and see what else is at the same level as Servers. You will see several items. Scroll to look at the list. Items flagged with a leading “d” are directories that you can cd (change dir) to. Items flagged with “r” are readable attributes that you can view:

```
wls:/MedRecDomain/serverConfig/Servers> cd ('../')
wls:/MedRecDomain/serverConfig> ls()
dr-- AdminConsole
dr-- AppDeployments
dr-- BridgeDestinations
dr-- Clusters
dr-- CustomResources
dr-- DeploymentConfiguration
...

```

- f) Get the status of MedRecServer2 Startup Mode. It should say RUNNING. You can retrieve other information from this server.

### Practice 6-3: Using WLST (continued)

```
wls:/MedRecDomain/serverConfig> cd ('Servers/MedRecSvr2')
wls:/MedRecDomain/serverConfig/Servers/MedRecSvr2>
get('StartupMode')
'RUNNING'
wls:/MedRecDomain/serverConfig/Servers/MedRecSvr2>
```

- 2) Change the value of the Rotation file size from 5001 to **5002** using WLST.

- a) Use the following WLST commands sequence: (Note the use of directory paths.)

```
edit()
startEdit()
cd('/Servers/MedRecAdmSvr/Log/MedRecAdmSvr')
get('FileMinSize')
cmo.setFileMinSize(5002)
get('FileMinSize')
save()
activate()
disconnect()
exit()
```

If you disconnected before saving, the change is not committed. The steps—`save`, `activate`, `disconnect`, and `exit`—are common to all configuration scripts.

```
wls:/MedRecDomain/serverConfig/Servers/MedRecSvr2> edit()
Location changed to edit tree. ...

wls:/MedRecDomain/edit> startEdit()
Starting an edit session ...
wls:/MedRecDomain/edit !>
cd('/Servers/MedRecAdmSvr/Log/MedRecAdmSvr')
wls:/MedRecDomain/edit/Servers/MedRecAdmSvr/Log/MedRecAdmSv
r !> get('FileMinSize')
5001
wls:/MedRecDomain/edit/Servers/MedRecAdmSvr/Log/MedRecAdmSv
r !> cmo.setFileMinSize(5002)
wls:/MedRecDomain/edit/Servers/MedRecAdmSvr/Log/MedRecAdmSv
r !> get('FileMinSize')
5002
wls:/MedRecDomain/edit/Servers/MedRecAdmSvr/Log/MedRecAdmSv
r !> save()
Saving all your changes ...

wls:/MedRecDomain/edit/Servers/MedRecAdmSvr/Log/MedRecAdmSv
r !> activate()
Activating all your changes, this may take a while ...

wls:/MedRecDomain/edit/Servers/MedRecAdmSvr/Log/MedRecAdmSv
r > disconnect()
Disconnected from weblogic server: MedRecAdmSvr
wls:/offline> exit()
```

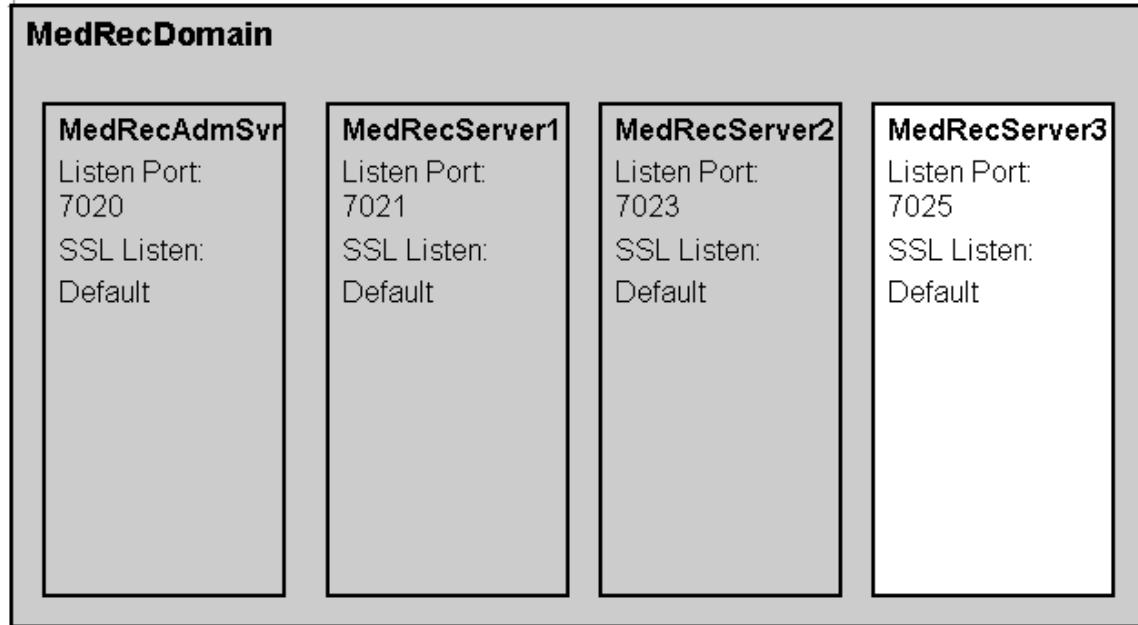
## Practices for Lesson 7

### Configuring Servers

The key tasks covered in this practice include:

- Creating and deleting managed servers
- Starting and stopping managed servers
- Monitoring managed servers

### Big Picture:



## Practice 7-1: Managing Managed Servers by Using the Administration Console

In this practice, you create, delete, start, stop, and monitor managed servers using the Web browser interface. You currently have two managed servers. You add a third managed server (which you will keep). Then you add a fourth managed server (which you will not keep) and delete the fourth server.

- 1) Using the Administration Console, create a new managed server with the following properties:

| Step | Property Name         | Choices or Values        |
|------|-----------------------|--------------------------|
| a.   | Server Name           | <b>MedRecSvr3</b>        |
| b.   | Server Listen Address | (leave it blank)         |
| c.   | Server Listen Port    | <b>7025</b>              |
| d.   | Cluster               | <b>None (Standalone)</b> |

- a) Log in to the Administration Console at `http://wls-sysadm:7020/console` using **weblogic** as the username and **Welcome1** as the password.
- b) In the Change Center, click Lock & Edit.
- c) In the Domain Structure, navigate to Environment > Servers.
- d) In the Servers table, click New.
- e) Specify the following values in the Create a New Server pages:

| Screen/Page Description | Choices or Values                                                                                                                                          |
|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Server Properties       | Server Name: <b>MedRecSvr3</b><br>Server Listen Address: (leave blank)<br>Server Listen Port: <b>7025</b><br>Select No, stand-alone server.<br>Click Next. |
| Review Choices          | Click Finish.                                                                                                                                              |

By leaving Listen Address blank, you can use any name in a URL that resolves to the same host regardless of the IP address.

- f) In the Change Center, click Activate Changes.
- 2) Similarly, create the MedRecSvr4 managed server using the Administration Console with the following properties:

| Property Name         | Choices or Values        |
|-----------------------|--------------------------|
| Server Name           | <b>MedRecSvr4</b>        |
| Server Listen Address | (leave it blank)         |
| Server Listen Port    | <b>7027</b>              |
| Cluster               | <b>None (Standalone)</b> |

- a) In the Change Center, click Lock & Edit.
- b) In the Servers table, click New.

## **Practice 7-1: Managing Managed Servers by Using the Administration Console (continued)**

- c) Specify the following values on the Create a New Server pages:

| Screen/Page Description | Choices or Values                                                                                                                                              |
|-------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Server Properties       | Server Name: <b>MedRecSvr4</b><br>Server Listen Address: <b>wls-sysadm</b><br>Server Listen Port: <b>7027</b><br>Select No, stand-alone server.<br>Click Next. |
| Review Choices          | Click Finish.                                                                                                                                                  |

- d) In the Change Center, click Activate Changes.
- 3) Delete the MedRecSvr4 managed server.
- a) In the Change Center, click Lock & Edit.
  - b) Select the check box next to MedRecSvr4. Click Delete. Click Yes in the “Are you sure?” dialog box.
  - c) In the Change Center, click Activate Changes.
- 4) The managed servers cannot be started from the Administration Console because the Node Manager has not been configured yet. To enable you to start the different servers in MedRecDomain in their own terminal sessions, shell scripts have been created in the /home/oracle/wls-sysadm folder.
- a) View the following scripts and use them to start MedRecSvr1, MedRecSvr2, and MedRecSvr3. You can also use the desktop icons for starting these servers.
- ```
$> /home/oracle/wls-sysadm/start_mr1.sh
$> /home/oracle/wls-sysadm/start_mr2.sh
$> /home/oracle/wls-sysadm/start_mr3.sh
```
- b) Each server session prompts for the username and password. Enter the values, and then each session should eventually indicate that the corresponding server is running as the message <Server started in RUNNING mode> appears.

Practice 7-1: Managing Managed Servers by Using the Administration Console (continued)

- c) Back in the Administration Console, refresh the Summary of Servers table and verify that MedRecAdmSvr, MedRecSvr1, MedRecSvr2, and MedRecSvr 3 are all RUNNING, and Health is OK.

| Servers (Filtered - More Columns Exist) | | | | | | |
|--|---------|---------|---------|--------|-------------------------------------|--|
| Click the Lock & Edit button in the Change Center to activate all the buttons on this page. | | | | | | |
| | | New | Clone | Delete | Showing 1 to 4 of 4 Previous Next | |
| Name | Cluster | Machine | State | Health | Listen Port | |
| MedRecAdmSvr(admin) | | | RUNNING | ✓ OK | 7020 | |
| MedRecSvr1 | | | RUNNING | ✓ OK | 7021 | |
| MedRecSvr2 | | | RUNNING | ✓ OK | 7023 | |
| MedRecSvr3 | | | RUNNING | ✓ OK | 7025 | |

- 5) Shut down MedRecSvr2 from the command line and MedRecSvr3 by using the Administration Console
- In your gnome terminal, navigate to the bin folder of your domain and run stopManagedWebLogic.sh as follows:
- ```
$> cd /u01/app/oracle/user_projects/domains/MedRecDomain/bin
$> ./stopManagedWebLogic.sh MedRecSvr2
```
- In the Administration Console, access the Summary of Servers page. Click the Control tab.
  - Select MedRecSvr3. Click Shutdown, and then from the drop-down menu, select Force Shutdown Now. Acknowledge you want to do this by clicking Yes.
  - Check the tab running the process for MedRecSvr3. It should have stopped.
  - Refresh the Summary of Servers table and now MedRecSvr1 should still be RUNNING, and MedRecSvr2 and MedRecSvr3 should both be SHUTDOWN.

| Servers (Filtered - More Columns Exist)       |         |          |                       |  |
|-----------------------------------------------|---------|----------|-----------------------|--|
| Start Resume Suspend ▾ Shutdown ▾ Restart SSL |         |          |                       |  |
| Showing 1 to 4 of 4 Previous   Next           |         |          |                       |  |
| Server                                        | Machine | State    | Status of Last Action |  |
| MedRecAdmSvr(admin)                           |         | RUNNING  | None                  |  |
| MedRecSvr1                                    |         | RUNNING  | None                  |  |
| MedRecSvr2                                    |         | SHUTDOWN | None                  |  |
| MedRecSvr3                                    |         | SHUTDOWN | TASK COMPLETED        |  |

## **Practice 7-2: Adding Managed Servers by Using WLST**

In this practice, you add a fourth managed server by using WLST. Using the Administration Console, verify that the server is created. Finally, delete the server by using WLST.

- 1) Create a managed server with the following properties:

| <b>Property Name</b>  | <b>Choices or Values</b> |
|-----------------------|--------------------------|
| Server Name           | <b>MedRecSvr4</b>        |
| Server Listen Address | <b>(leave it blank)</b>  |
| Server Listen Port    | <b>7027</b>              |
| Cluster               | <b>None (Standalone)</b> |

- a) In your gnome terminal session, ensure that the environment variables have been set. You make a quick check using the following command:

```
$> env | grep JAVA
JAVA_USE_64BIT=
JAVA_OPTIONS= -Xverify:none
JAVA_VENDOR=Oracle
JAVA_HOME=/u01/app/oracle/product/fmw/11.1.0/jrockit_160_05
_R27.6.2-20
JAVA_VM=-jrockit
```

If JAVA parameters do not appear, you can execute the `setWLSEnv.sh` script to set the environment variables.

- b) Enter the following code to create the managed server:

```
java weblogic.WLST
connect('weblogic','Welcome1','t3://wls-sysadm:7020')
edit()
startEdit()
cmo.createServer('MedRecSvr4')
cd('/Servers/MedRecSvr4')
cmo.setListenAddress('wls-sysadm')
cmo.setListenPort(7027)
activate()
disconnect()
exit()
```

- 2) In the Administration Console, in Domain Structure, navigate to MedRecDomain > Environment > Servers, and you should see MedRecSvr4.

**Servers(Filtered - More Columns Exist)**

| <input type="checkbox"/> | Server              | Machine | State    | Status of Last Action |
|--------------------------|---------------------|---------|----------|-----------------------|
| <input type="checkbox"/> | MedRecAdmSvr(admin) |         | RUNNING  | None                  |
| <input type="checkbox"/> | MedRecSvr1          |         | RUNNING  | None                  |
| <input type="checkbox"/> | MedRecSvr2          |         | SHUTDOWN | None                  |
| <input type="checkbox"/> | MedRecSvr3          |         | SHUTDOWN | TASK COMPLETED        |
| <input type="checkbox"/> | MedRecSvr4          |         | SHUTDOWN | None                  |

- 3) Delete MedRecSvr4 by using WLST.

- a) In the gnome terminal session, enter the following code to delete the MedRec4 managed server:

```
java weblogic.WLST
connect('weblogic','Welcome1','t3://wls-sysadm:7020')
edit()
startEdit()
cd('/Servers')
ls()
delete('MedRecSvr4')
activate()
disconnect()
exit()
```

- 4) Back in the Administration Console, in Domain Structure, navigate to MedRecDomain > Environment > Servers. You should see that MedRecSvr4 is deleted.

**Servers(Filtered - More Columns Exist)**

| <input type="checkbox"/> | Server              | Machine | State    | Status of Last Action |
|--------------------------|---------------------|---------|----------|-----------------------|
| <input type="checkbox"/> | MedRecAdmSvr(admin) |         | RUNNING  | None                  |
| <input type="checkbox"/> | MedRecSvr1          |         | RUNNING  | None                  |
| <input type="checkbox"/> | MedRecSvr2          |         | SHUTDOWN | None                  |
| <input type="checkbox"/> | MedRecSvr3          |         | SHUTDOWN | TASK COMPLETED        |

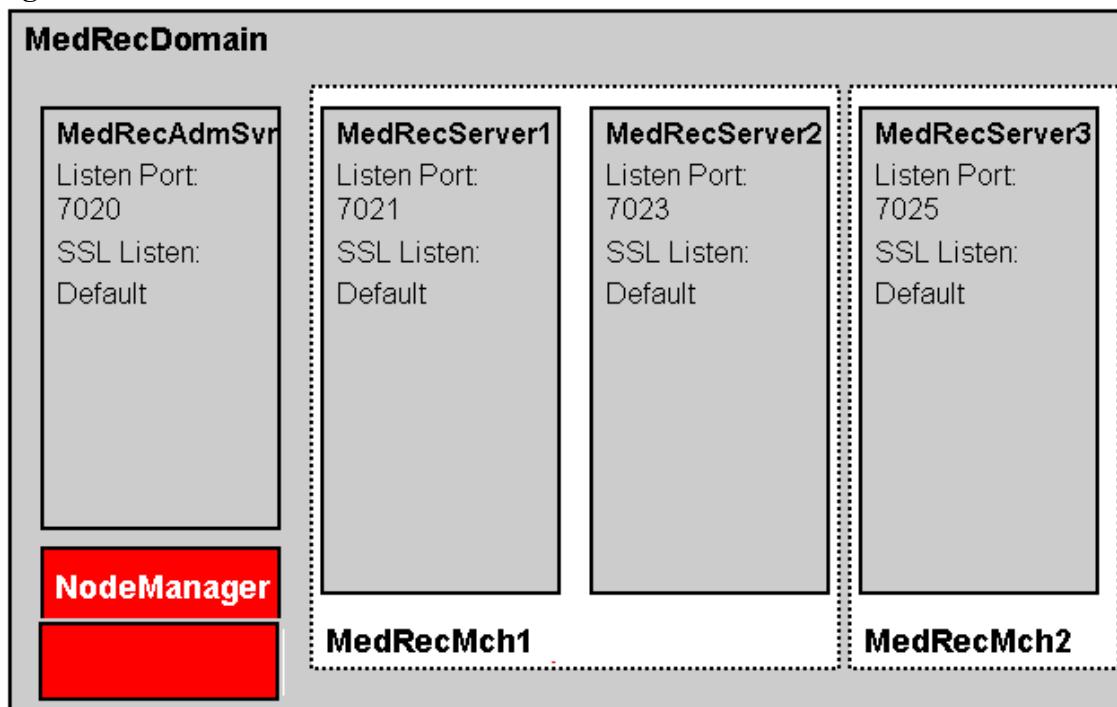
## Practices for Lesson 8

### Configuring the Node Manager

The Node Manager operates on machines. A machine is a logical construct that contains a group of servers. Usually, there is only one machine per computer, but for the lab, you will have two machines in the same computer. One of the main purposes of the Node Manager is to allow you to remotely start a managed server via the Administration Console. The key tasks are:

- Adding two machines and assigning servers to them
- Connecting to the Node Manager
- Connecting the Node Manager to the managed servers
- Invoking the Node Manager to start a managed server via the command line
- Invoking the Node Manager to start a managed server via the Administration Console

### Big Picture:



## Practice 8-1: Adding Machines and Assigning Servers

In this practice, you add two machines to the domain: MedRecMch1 and MedRecMch2. Then you assign managed servers to these machines. Note that the servers have to be stopped before they can be added to a machine. Generally, you do not assign the administration server to any machine. Later on, you will manage the servers using the Node Manager.

- 1) In MedRecDomain, create two machines: MedRecMch1 and MedRecMch2. Assign MedRecSvr1 and MedRecSvr2 to MedRecMch1, and MedRecSvr3 to MedRecMch2.
  - a) If the MedRecAdmSvr server is not running, start it by using the Start MR Admin icon on the desktop. If any of the managed servers are running, stop them by using the Console or `stopManagedWebLogic` as covered in the previous lab.
  - b) In the Administration Console, in the Change Center, click Lock & Edit. You will need to click Lock & Edit each time before any change to the configuration.
  - c) In Domain Structure, navigate to MedRecDomain > Environment > Machines. Click New.
  - d) In Name, enter **MedRecMch1**. In Machine OS, select Unix from the drop-down menu. Click OK.
  - e) On Summary of Machines, click MedRecMch1. Click the Configuration > Servers tab. Click Add.
  - f) On the Identify Server page, from the “Select a server” drop-down list, select MedRecSvr1 and click Next.
  - g) On the Summary of Machines page, click Add. On the Identify Server page, from the “Select a server” drop-down list, select MedRecSvr2 and click Finish.
  - h) In the Change Center, click Activate Changes. This adds the servers to a new machine.
  - i) Using steps d through f, create the **MedRecMch2** machine and assign **MedRecSvr3** to it. Click Lock & Edit as necessary.
  - j) In the Administration Console, refresh the “Summary of Servers” table. Now all managed servers should be associated with a machine.

**Servers (Filtered - More Columns Exist)**

Click the **Lock & Edit** button in the Change Center to activate all the buttons on this page.

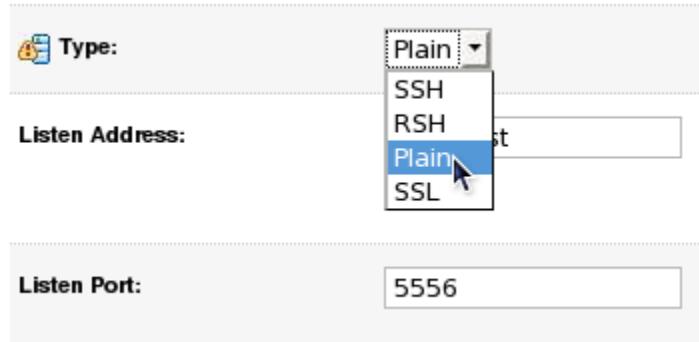
|                          | Name                | Cluster | Machine    | State    | Health | Listen Port |
|--------------------------|---------------------|---------|------------|----------|--------|-------------|
| <input type="checkbox"/> | MedRecAdmSvr(admin) |         |            | RUNNING  | ✓ OK   | 7020        |
| <input type="checkbox"/> | MedRecSvr1          |         | MedRecMch1 | SHUTDOWN |        | 7021        |
| <input type="checkbox"/> | MedRecSvr2          |         | MedRecMch1 | SHUTDOWN |        | 7023        |
| <input type="checkbox"/> | MedRecSvr3          |         | MedRecMch2 | SHUTDOWN |        | 7025        |

## Practice 8-2: Connecting to the Node Manager

In the classroom environment, the administration server and the managed servers are all on the same physical computer. This would not normally be the case. Because the servers would probably be on different computers, the administration server needs to know the JAVA\_HOME and CLASSPATH on the *managed servers*' machines, which may be different from its own classpath and Java home.

In this practice, you start the Node Manager for the MedRecMch1 machine. This must be done using the command line.

- 1) By default, the Node Manager communicates with the administration server over Secure Sockets Layer (SSL) connections. However, the administration server is not yet configured with secured connections. So reconfigure the Node Manager to use plain communication.
  - a) In the Change Center, click Lock & Edit. Then navigate to MedRecDomain > Environment > Machines. Click MedRecMch1.
  - b) Click the Node Manager tab and select Plain from the Type drop-down list. Set Listen Address to wls-sysadm and click Save.



- c) Click Activate Changes.
- d) Similarly, set up the Node Manager to use plain communication for MedRecMch2.
- 2) Set up the startup parameters \$JAVA\_HOME and \$CLASSPATH to enable the Node Manager to start up appropriately.
  - a) In the gnome terminal that has been set up with `setWLSEnv.sh`, get the values for \$JAVA\_HOME and \$CLASSPATH.

```
$> echo $JAVA_HOME
$> echo $CLASSPATH
```

## Practice 8-2: Connecting to the Node Manager (continued)

Both these pieces of information will be pasted into the browser window.

```
[oracle@edvmr1p0 /]$ echo $JAVA_HOME
/u01/app/oracle/product/fmw/11.1.0/jrockit_160_05_R27.6.2-20
[oracle@edvmr1p0 /]$ echo $CLASSPATH
/u01/app/oracle/product/fmw/11.1.0/patch_wls1031/profiles/default/sys_manifest_cl
asspath/weblogic_patch.jar:/u01/app/oracle/product/fmw/11.1.0/jrockit_160_05_R27.
6.2-20/lib/tools.jar:/u01/app/oracle/product/fmw/11.1.0/utils/config/10.3.1.0/con
fig-launch.jar:/u01/app/oracle/product/fmw/11.1.0/wlserver_10.3/server/lib/weblog
ic_sp.jar:/u01/app/oracle/product/fmw/11.1.0/wlserver_10.3/server/lib/weblogic.ja
r:/u01/app/oracle/product/fmw/11.1.0/modules/features/weblogic.server.modules_10.
3.1.0.jar:/u01/app/oracle/product/fmw/11.1.0/wlserver_10.3/server/lib/webservices
.jar:/u01/app/oracle/product/fmw/11.1.0/modules/org.apache.ant_1.7.0/lib/ant-all.
jar:/u01/app/oracle/product/fmw/11.1.0/modules/net.sf.antcontrib_1.0.0.0_1-0b2/li
b/ant-contrib.jar:
[oracle@edvmr1p0 /]$
```

- b) In the Change Center of the Administration Console, click Lock & Edit. In Domain Structure, navigate to MedRecDomain > Environment > Servers. Click MedRecSrv1. Navigate to the Configuration > Server Start tabs.
- c) Copy and paste the \$JAVA\_HOME information from the gnome terminal session window into the Java Home field on the Settings for MedRecSrv1 page.



- d) Copy and paste the \$CLASSPATH information from the Linux terminal session window into the Class Path field on the Settings for MedRecSrv1 page.

- e) In User Name, enter **weblogic**. In Password and Confirm Password, enter **Welcome1**. Click Save.

|           |                                     |
|-----------|-------------------------------------|
| User      | weblogic                            |
| Name:     |                                     |
| Password: | *****                               |
| Confirm:  | *****                               |
| Save      | <input type="button" value="Save"/> |

## **Practice 8-2: Connecting to the Node Manager (continued)**

- f) Repeat steps b, c, and d for MedRecSrv2.
  - g) In Change Center, click Activate Changes.
- 3) Create the nodemanager.properties file in the \$WL\_HOME/common/nodemanager folder for starting the Node Manager.
- a) The file you need to edit does not exist yet. You can verify this by listing the files in \$WL\_HOME/common/nodemanager.
  - b) Start the Node Manager by navigating to \$WL\_HOME/server/bin and using the following command:
- ```
$> ./startNodeManager.sh wls-sysadm 5556.
```
- Wait until it says <Secure socket listener started on port 5556>.
- Press Ctrl + C to stop the Node Manager. This step created the nodemanager.properties file.
- c) In the gnome terminal session, navigate to the \$WL_HOME/common/nodemanager folder and edit nodemanager.properties. Set the SecureListener parameter to **false** and StartScriptEnabled to **true**:

```

9 LogLevel=INFO
10 DomainsFileEnabled=true
11 StartScriptName=startWebLogic.sh
12 ListenAddress=localhost
13 NativeVersionEnabled=true
14 ListenPort=5556
15 LogToStderr=true
16 SecureListener=false
17 LogCount=1
18 StopScriptEnabled=false
19 QuitEnabled=false
20 LogAppend=true
21 StateCheckInterval=500
22 CrashRecoveryEnabled=false
23 StartScriptEnabled=true
24LogFile=/u01/app/oracle/product/fmw/11.1.0/wlserver_10.3/
...

```

- 4) Start the Node Manager because you have reconfigured it using plain mode of communication.
- a) In a gnome terminal session, go to \$WL_HOME/server/bin and start the Node Manager using the following command:
- ```
$> gnome-terminal --title "Node Manager" -e
"./startNodeManager.sh wls-sysadm 5556"
```

## **Practice 8-2: Connecting to the Node Manager (continued)**

This brings up the Node Manager in a separate window. Now it should say  
<Plain socket listener started on port 5556>.

- 5) It is possible for a single Node Manager to manage multiple domains on a single computer. List the contents of nodemanager.domains to see the domains that could be supported from this Node Manager. At a terminal session, type:  
`more $WL_HOME/common/nodemanager/nodemanager.domains`  
At least MedRecDomain should be in the list, there may be other domains as well.

## Practice 8-3: Starting Managed Servers by Using the Node Manager

In this practice, you start one managed server via the Administration Console and the other managed server via WLST. Lastly, you kill one of the servers to simulate an accident (for example, a power outage) and the Node Manager will restart the managed server automatically.

- 1) Using the Administration Console, start the MedRecSvr1 and MedRecSvr2 servers.
  - a) In the Administration Console, navigate to MedRecDomain > Environment > Servers and click the Control tab.
  - b) Select MedRecSvr1 and click Start. Click Yes to start the server. The State changes from UNKNOWN to STARTING. You can refresh the table by clicking the cycle icon or by selecting MedRecDomain > Environment > Servers.
  - c) Note that you no longer have a terminal session displaying the server log. Note that the Node Manager terminal output indicated that it was creating several directories and files for the managed server MedRecSvr1.

- 2) Using WLST, start the MedRecSvr2 server.

```
$> java weblogic.WLST
wlst/connect('weblogic','Welcome1','wls-sysadm:7020')
cd('/Servers')
ls()
start('MedRecSvr2')
exit()
```

You should see a series of dots as a progress bar, and then the message:  
Server with name MedRecSvr2 started successfully

- 3) Verify that the MedRecSvr1 and MedRecSvr2 servers are started properly by viewing their State in the Administration Console. The State should be RUNNING and Health should be OK.

|  | Name                | Cluster | Machine    | State    | Health | Listen Port |
|--|---------------------|---------|------------|----------|--------|-------------|
|  | MedRecAdmSvr(admin) |         |            | RUNNING  | OK     | 7020        |
|  | MedRecSvr1          |         | MedRecMch1 | RUNNING  | OK     | 7021        |
|  | MedRecSvr2          |         | MedRecMch1 | RUNNING  | OK     | 7023        |
|  | MedRecSvr3          |         | MedRecMch2 | SHUTDOWN |        | 7025        |

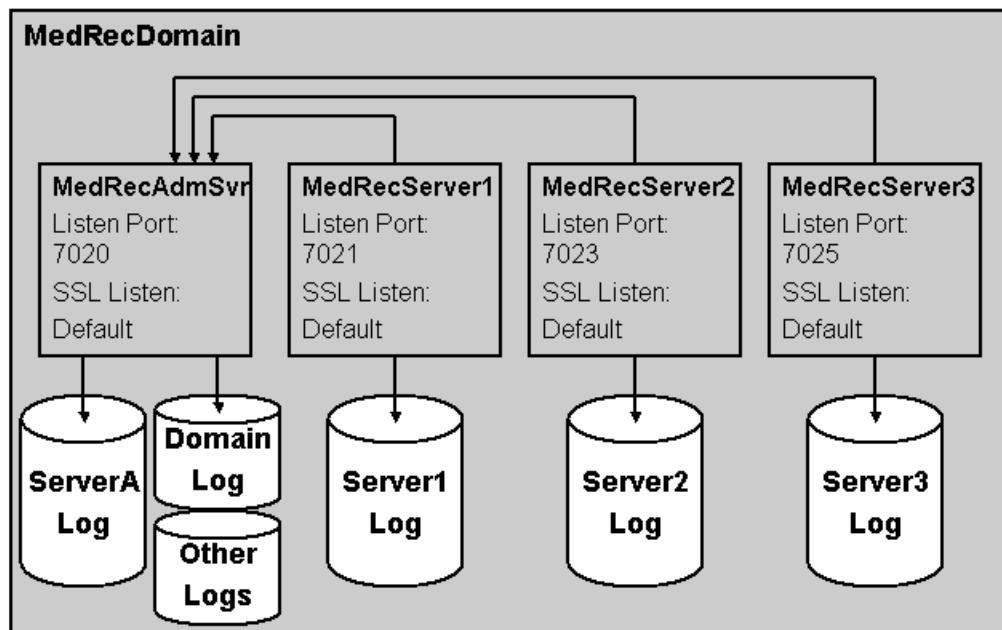
## Practices for Lesson 9

### Configuring Logging

Each server (both the administration and managed varieties) generates logs of activity. In addition to the server logs, there are HTTP logs, JMS logs, JDBC logs, and application logs. The server logs are stored locally and some of the information can also be forwarded to a domain log at the administration server. You filter the traffic from MedRecSvr3 to only send more severe JDBC errors. Lastly, you look at the kinds of logs available and look at the domain log in particular. The key tasks are:

- Configuring logging parameters
- Examining log entries

### Big Picture:



## Practice 9-1: Configuring Logging Parameters

In this practice, you configure MedRecSvr1 to forward only severe JDBC errors by the use of filters. The default configuration for MedRecSvr2 will forward all errors. The filters are created at the domain side, but applied at the server side.

- 1) Start the Administration Server and the Node Manager, if they have not already been started. Then start the MedRecSvr1 and MedRecSvr2 servers.
  - a) Use the Start MR Admin icon to start the administration server.
  - b) Use the `start_nm.sh` script in a gnome terminal to start the Node Manager.
  - c) Using the Administration Console, start MedRecSvr1 and MedRecSvr2.
- 2) In the Administration Console, in Change Center, click Lock & Edit. In Domain Structure, navigate to MedRecDomain. In the tabs, navigate to Configuration > Log Filters. Click New.
- 3) Enter the following values on the Create a New Log Filter pages:

| Step | Screen/Page Description                                                                                                                             | Choices or Values                                                              |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------|
| a.   | Log Filter Properties                                                                                                                               | Name: <b>SevereJDBC</b> . Click OK.                                            |
| b.   | Log Filters                                                                                                                                         | Click SevereJDBC.                                                              |
| c.   | Config Log Filter Expressions                                                                                                                       | Click Add Expressions.                                                         |
| d.   | Add Expression (By default Notice and Warning, messages are also logged. Using this filter you restrict the amount and kind of messages forwarded.) | Message Attribute: SEVERITY<br>Operator: =<br>Value: <b>ERROR</b><br>Click OK. |
| e.   | Config Log Filter Expressions                                                                                                                       | Click Add Expressions.                                                         |
| f.   | Add Expression                                                                                                                                      | Message Attribute: SUBSYSTEM<br>Operator: =<br>Value: <b>JDBC</b><br>Click OK. |
| g.   | Save expressions                                                                                                                                    | Click Save.                                                                    |

Note that the two expressions are conjugated with “OR” by default. You can change the conjugation to “AND” to get only ERROR messages from the JDBC subsystem.

- 4) In Change Center, click Activate Changes.
- 5) Click Lock & Edit again and navigate to MedRecDomain > Servers. In the Servers table, click MedRecSvr1 and navigate tabs to Logging > General. Click Advanced.
- 6) Note that there are four sections that look similar. In the “Domain log broadcaster” section, change Filter to SevereJDBC. The other three filters should say None.



- 7) Click Save, and then click Activate Changes.

## **Practice 9-2: Examining Log Entries**

In this practice, you look at the log files using the Administration Console and the raw file itself.

- 1) Using the Administration Console, view ServerLog, and DomainLog for MedRecDomain, and note the significant properties.
  - a) In the Domain Structure, navigate to MedRecDomain > Diagnostics > Log Files. Sort the list by type and list the types of logs that you find:

| Type of Logs | Type of Logs |
|--------------|--------------|
|              |              |
|              |              |
|              |              |
|              |              |

- b) To examine ServerLog for MedRecSvr1, select ServerLog for MedRecSvr1 and click View. It is possible that ServerLog will not display anything. Does that mean that it is empty?
  - c) Similarly, examine DomainLog for MedRecAdmSvr. For the first entry, select it and click View for a different format. There is no additional information (though columns from the table may have been suppressed). The View is mostly for a better format or layout of the message.
- 2) View the log files in a text editor and note the significant properties.
  - a) In a terminal session, navigate to  
 /u01/app/oracle/user\_projects/domains/MedRecDomain/servers/  
 MedRecAdmSvr/logs.
  - b) Examine the domain log. Most of the messages are related to the starting and stopping of the servers and are of the severity “Notice.” Still, there seems to be many more lines in the file than on the Administration Console. Why?
- 3) In the Administration Console, change the time interval for logging and verify the logs.
  - a) Back in the Administration Console, view the Domain Log again. Click “Customize this table.” Change the Filter Time Interval from the default “Last 5 minutes” to “Last 1 week(s).” Click Apply.
  - b) Now there should be several entries to look at.
  - c) See if this time filter change altered the number of entries shown on ServerLog as well.

## **Practices for Lesson 10**

There are no practices for Lesson 10.

## Practices for Lesson 11

### Deploying Applications

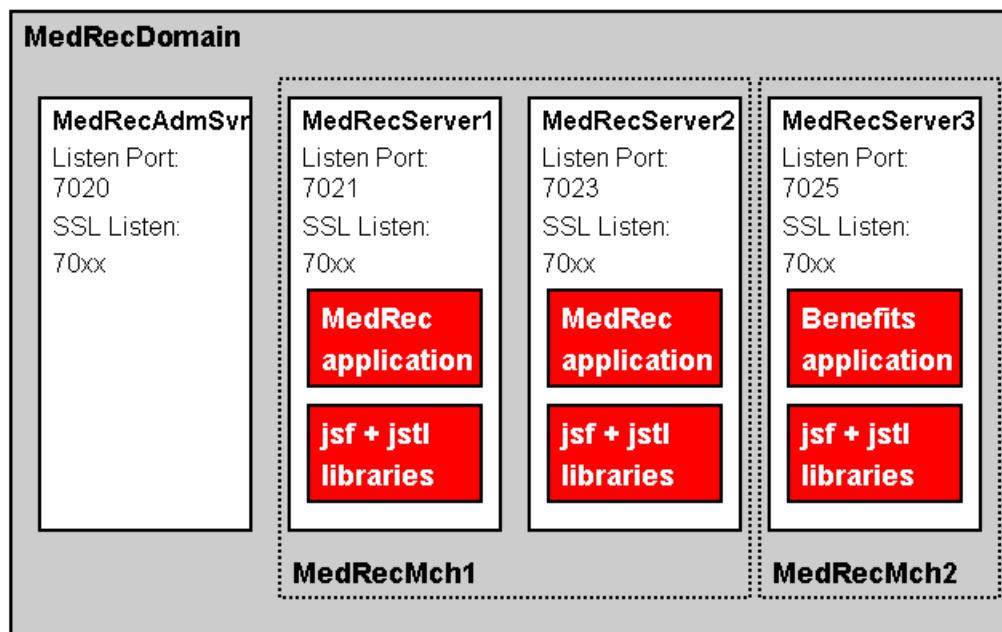
You are going to deploy two applications: Benefits and Medrec.

Benefits is very simple and does not use any extra services, and Medrec requires two libraries and uses many services, such as JMS and JDBC. These services are covered in later labs. For the moment, you just install them as a “black box.”

The key tasks are:

- Deploying prerequisite libraries
- Deploying applications
- Starting and stopping applications
- Testing applications
- Deleting applications
- Front-ending applications with a Web server, such as Oracle HTTP Server

### Big Picture:



## Practice 11-1: Deploying Libraries

Often, many applications use the same set of library classes. These classes might be ones that you write or ones that are part of a larger framework, such as JavaServer Faces.

Rather than duplicate those classes in each application, you can store the common classes in a library that many applications can use. In this practice, you deploy two libraries on all the servers that will be used by (at least) the Medrec application (and potentially by other applications as well).

- 1) Start up the administration server and the Node Manager if they have not already been started. Then start the MedRecSvr1 and MedRecSvr2 servers.
  - a) Use the Start MR Admin desktop icon to start the administration server.
  - b) Use the `start_nm.sh` script in a gnome terminal to start the Node Manager.
  - c) Using the Administration Console, start MedRecSvr1 and MedRecSvr2.
- 2) Clear up any deployments in MedRecDomain. (If nothing is deployed, skip the rest of this step.)
  - a) In the Administration Console, navigate to MedRecDomain > Deployments.
  - b) In Change Center, click Lock & Edit. In Deployments, select all the deployments by selecting the check box at the top next to Name. That should select everything.
  - c) Select Stop > Force Stop Now. Click Yes to stop everything. Note the message that the libraries will not be stopped.
  - d) Select the check box at the top next to Name again, and click Delete. Click Yes to delete everything.
  - e) Click Activate Changes. You should now have a clean set of servers to work with.
- 3) Deploy the two required library files for the MedRec application using the following steps:
  - a) In Change Center, click Lock & Edit. In Deployments, click Install.
  - b) In the Install Application Assistant, complete the following steps for deploying a JSF library class:

| Step | Screen/Page Description                                                                                                                                                                                                                                                                                                                                                                            | Choices or Values                                                                                           |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------|
| a.   | Locate deployment to install and prepare for deployment                                                                                                                                                                                                                                                                                                                                            | Navigate to <code>/home/oracle/wls-sysadm/labs/Lab11</code> . Select <code>jsf-1.2.war</code> . Click Next. |
|      | <p><b>Path:</b> <code>/home/oracle/wls-sysadm/labs/Lab11/jsf-1.2.war</code></p> <p><b>Recently Used Paths:</b> (none)</p> <p><b>Current Location:</b> <code>wls-sysadm / home / oracle / wls-sysadm / labs / Lab11</code></p> <p><input checked="" type="radio"/> <b>jsf-1.2.war</b><br/> <input type="radio"/> <b>jstl-1.2.war</b></p> <p><b>Back</b> <b>Next</b> <b>Finish</b> <b>Cancel</b></p> |                                                                                                             |

## Practice 11-1: Deploying Libraries (continued)

| Step | Screen/Page Description                                                                                                                                                                                                                                                  | Choices or Values                                                                                                    |
|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------|
| b.   | Choose targeting style                                                                                                                                                                                                                                                   | Note that the Assistant knows that jsf is a library, and not an application. Click Next.                             |
| c.   | Select deployment targets                                                                                                                                                                                                                                                | Select the MedRecSvr1 server as the target. Click Next.                                                              |
|      |                                                                                                                                                                                        |                                                                                                                      |
| d.   | Optional Settings                                                                                                                                                                                                                                                        | You can override anything in the deployment plans or you can accept the defaults. Click Next to accept the defaults. |
|      | The dashed lines at the bottom of the page in the Source accessibility section are not aligned with the options in an obvious way. For instance, "Use the defaults" is the "Recommended selection" even though it appears to be associated with "Copy this application." |                                                                                                                      |
| e.   | Review your choices                                                                                                                                                                                                                                                      | Select "Yes, take me to the deployment's configuration screen." Click Finish.                                        |
| f.   | Settings for jsf                                                                                                                                                                                                                                                         | You can use the Notes to document who did the implementation, when, why, and so forth.                               |
| g.   | Final step                                                                                                                                                                                                                                                               | Click Save. Click Activate Changes.                                                                                  |

- c) In Domain Structure, click Deployments. In Change Center, click Lock & Edit. Click Install.
- d) In the Install Application Assistant, complete the following steps for deploying the JSTL library class:

| Step | Screen/Page Description                                 | Choices or Values                                                                         |
|------|---------------------------------------------------------|-------------------------------------------------------------------------------------------|
| a.   | Locate deployment to install and prepare for deployment | Navigate to /home/oracle/wls-sysadm/labs/Lab11. Select jstl-1.2.war. Click Next.          |
| b.   | Choose targeting style                                  | Note that the Assistant knows that jstl is a library, and not an application. Click Next. |
| c.   | Select deployment targets                               | Select MedRecSvr1. Click Next.                                                            |
| d.   | Optional Settings                                       | You have already seen the screens here and you need not change anything. As a shortcut,   |

**Practice 11-1: Deploying Libraries (continued)**

| Step | Screen/Page Description | Choices or Values                                                                                                                                    |
|------|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------|
|      |                         | click Finish to accept the defaults.                                                                                                                 |
| e.   | Settings for jstl       | If you are not going to add any notes, then you do not have to save. Note at the bottom of the page that no applications reference this library yet. |
| f.   | Final step.             | Click Activate Changes.                                                                                                                              |

- 4) On the Deployments page, you should now see jsf and jstl deployed and are active. Unlike application deployments, libraries do not need to be started; they are already started. To verify this, select jsf and select Start > Servicing all requests. A warning message appears indicating that this is not necessary and will be ignored.

## Practice 11-2: Deploying Applications

In this practice, you deploy two applications: MedRec and Benefits. Deploy MedRec on MedRecSrv1 and Benefits on MedRecSrv2.

- 1) Deploy a Java EE application named MedRec on the MedRecSrv1 server.
  - a) In the Administration Console, in Domain Structure, navigate to Deployments. In Change Center, click Lock & Edit. In Deployments, click Install.
  - b) In the Install Application Assistant, complete the following steps for deploying Medrec applications:

| Step | Screen/Page Description                                 | Choices or Values                                                                                                                                                        |
|------|---------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| a.   | Locate deployment to install and prepare for deployment | Navigate to /home/oracle/wls-sysadm/labs/Lab11. Select medrec.ear. Click Next.                                                                                           |
| b.   | Choose targeting style                                  | Note that the Assistant knows that Medrec is an application versus a library. This is different from jsf and jstl that you created previously. Click Next.               |
| c.   | Select deployment targets                               | Select MedRecSrv1. Click Next.                                                                                                                                           |
| d.   | Optional settings                                       | You have already seen the screens here and you need not change anything. As a shortcut, click Finish to accept the defaults. Note that State is distribute Initializing. |
| e.   | Change Center                                           | Click Activate Changes. Note that State is now Prepared.                                                                                                                 |

- 2) Optionally, you can click the plus next to Medrec and see the pieces of the EAR that were deployed. Click the minus sign to shrink it.
- 3) Deploy the Benefits application on MedRecSrv2.
  - a) In the Administration Console, in Domain Structure, navigate to Deployments. In Change Center, click Lock & Edit. In Deployments, click Install.
  - b) In the Install Application Assistant, complete the following steps for the Benefits deployment:

| Step | Screen/Page Description                                 | Choices or Values                                                                                                                             |
|------|---------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| a.   | Locate deployment to install and prepare for deployment | Navigate to /home/oracle/wls-sysadm/labs/Lab11. Select benefits.war. Click Next.                                                              |
| b.   | Choose targeting style                                  | Note that the Assistant knows that Benefits is an application versus a library. Click Next.                                                   |
| c.   | Select deployment targets                               | Select MedRecSrv2. You are allowed to have more than one application per server, but for this lab, you are spreading them around. Click Next. |
| d.   | Optional settings                                       | You have already seen the screens here and you need not change anything. As a shortcut,                                                       |

**Practice 11-2: Deploying Applications (continued)**

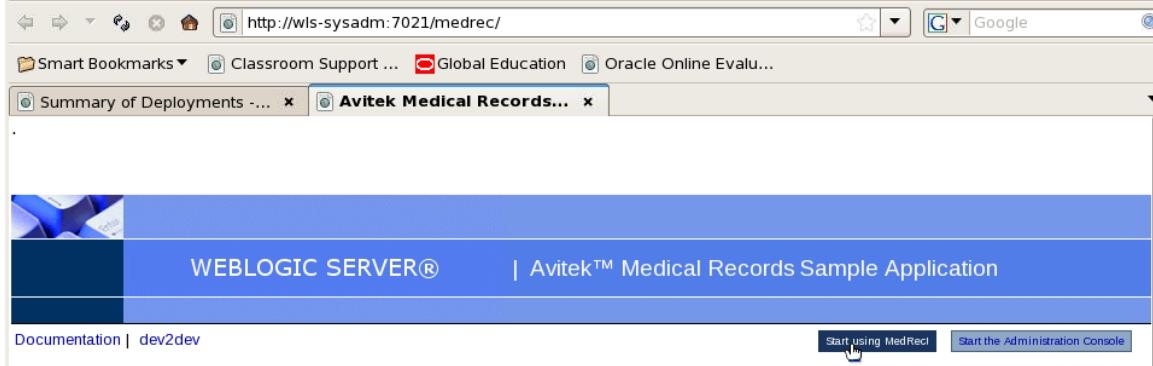
| Step | Screen/Page Description | Choices or Values                                                                    |
|------|-------------------------|--------------------------------------------------------------------------------------|
|      |                         | click Finish to accept the defaults. Note that the State is distribute Initializing. |
| e.   | Change Center           | Click Activate Changes. Note that State is now Prepared.                             |

- 4) Optionally, you can click jsf and observe that at the bottom, there is now a list of dependent applications that reference this library. Medrec requires the jsf library, whereas Benefits does not.

## **Practice 11-3: Performing Life Cycle Management of Applications**

In this practice, you start, stop, update (refresh/redeploy), and delete applications. There is a more advanced way of doing the update, which is covered in the next lab.

- 1) Start the two applications that you have deployed.
  - a) In Deployments, select benefits and medrec.
  - b) Select Start > Servicing all requests. Click Yes to start both deployments. State should now be Active for all deployments.
- 2) Test these applications by using their individual URLs.
  - a) Open a new browser tab and use the URL:  
<http://wls-sysadm:7021/medrec>
  - b) It displays the application on MedRecSrv1. You should see a welcome page from Avitek Medical Records Sample Application.
  - c) To test the application, click “Start using MedRec.”



- d) Click Login under Administrator. In email, enter `admin@avitek.com` with a password of `Welcome1`. Click Submit.



[View Pending Requests](#)  
Approve/Deny patient registration requests.

Do not spend more than a minute here because several key components are not implemented yet. Bookmark this URL.

### **Practice 11-3: Performing Life Cycle Management of Applications (continued)**

- e) To test the Benefits application, open another browser tab and enter the following URL: `http://wls-sysadm:7023/benefits`. You should see Welcome to MedRec Black and have the option to view several HR-related pages.
- 3) Monitor the two applications by using the Administration Console.
  - a) In the Administration Console, in Summary of Deployments, click the Monitoring tab. You should see both Medrec and Benefits running with some sessions.
- 4) Suppose a new version of the Benefits application is released. Deploy the new version of the Benefits application.

If the server was in development mode and autodeploy was active (which is not the case), then this change of time stamp would be enough to trigger a redeployment. For your environment, you must explicitly redeploy.

- a) Go to a Linux terminal session and navigate to `~/wls-sysadm/labs/Lab11`, and enter `touch benefits.war` to change the time stamp to now.
- b) In the Administration Console, in Change Center, click Lock & Edit.
- c) In Domain Structure, navigate to MedRecDomain > Deployments. Select benefits and click Update.
- d) Click Finish.

If you had the old and new WAR files in the `/old` and `/new` directories, you could change the paths here. In your case, nothing has moved. State is now deploy Initializing.

- e) In Change Center, click Activate Changes. State is now Active.

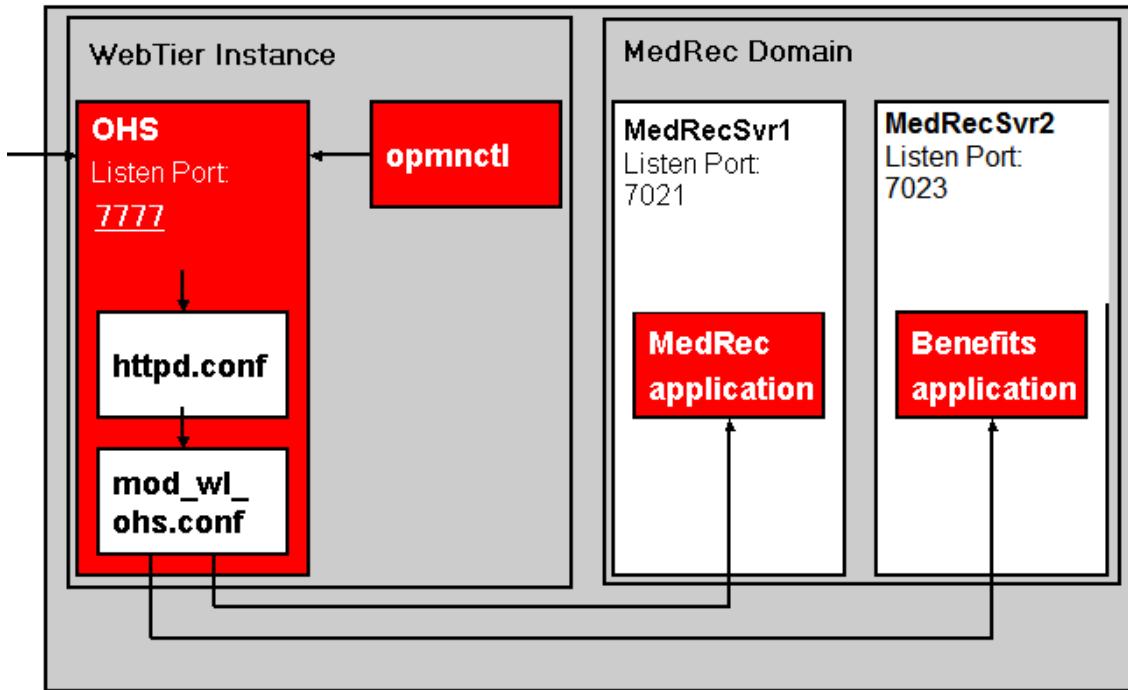
This was a rather ungraceful way of performing an update. Perhaps more graceful would be to stop the Benefits application first and then update. You can choose Force Stop Now or when work completes, depending on your situation.

## Practice 11-4: Enabling OHS as the Front End of Applications

In this practice, you redirect requests for the Benefits and Medrec applications via a Web server. Instead of the explicit URLs used before (which will still work), all references to those applications will be routed through Oracle HTTP Server. By configuring `mod_wl_ohs.conf`, you will be redirected to the following addresses:

- `http://wls-sysadm:7777/medrec` to `http://wls-sysadm:7021/medrec`
- `http://wls-sysadm:7777/benefits` to `http://wls-sysadm:7023/benefits`

In real life, the OHS, medrec, and benefits servers may be on different hosts.



- 1) Start Oracle HTTP Server that is installed and configured, and note the port number for Oracle HTTP Server.
  - a) Click the Start OHS icon on your desktop.
  - b) Run the `status_ohs.sh` script in the `~/wls-sysadm` folder and note the port beside http in the last line:

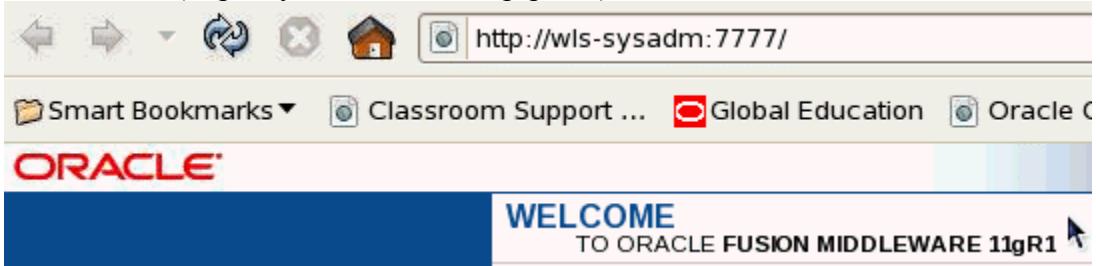
```
[oracle@edvmr1p0 wls-sysadm]$./status_ohs.sh

Processes in Instance: instance1
-----+-----+-----+-----+-----+
ias-component | process-type | pid | status |
uid | memused | uptime | ports
-----+-----+-----+-----+
ohsa | OHS | 2381 | Alive | 412
494714 | 348740 | 0:00:20 | https:9999,https:4443,http:7777

[oracle@edvmr1p0 wls-sysadm]$]
```

## **Practice 11-4: Enabling OHS as the Front End of Applications (continued)**

- c) Access OHS (<http://<your server>:<http-port>>) in the browser:



- 2) Reconfigure mod\_wl\_ohs.conf to route Benefits requests to MedRecSrv2 and Medrec requests to MedRecSrv1
- In a gnome terminal session, change directory to the OHS instance configuration folder (/u01/app/oracle/instances/config/OHS/ohsa).
  - Copy the mod\_wl\_ohs.conf file to mod\_wl\_ohs.bak. Then edit mod\_wl\_ohs.conf so that it appears as in the following screenshot:

```
NOTE : This is a template to configure mod_weblogic.

LoadModule weblogic_module "${ORACLE_HOME}/ohs/modules/mod_wl_ohs.so"

This empty block is needed to save mod_wl related configuration from EM to this
file when changes are made at the Base Virtual Host Level
<IfModule mod_weblogic.c>
WebLogicHost <WEBLOGIC_HOST>
WebLogicPort <WEBLOGIC_PORT>
Debug ON
WLLogFile /tmp/weblogic.log
MatchExpression *.jsp
</IfModule>

<Location /benefits>
 WebLogicHost wls-sysadm
 WebLogicPort 7023
 SetHandler weblogic-handler
PathTrim /weblogic
ErrorPage http://WEBLOGIC_HOME:WEBLOGIC_PORT/
</Location>
<Location /medrec>
 WebLogicHost wls-sysadm
 WebLogicPort 7021
 SetHandler weblogic-handler
PathTrim /weblogic
ErrorPage http://WEBLOGIC_HOME:WEBLOGIC_PORT/
</Location>
```

If you do not like editing the file, you can copy the mod\_wl\_ohs.conf file from the ~/wls-sysadm/labs/Lab12 folder to /u01/app/oracle/instances/config/OHS/ohsa.

- Verify accessing Medrec and Benefits applications through OHS.
- Stop and start OHS to give effect to the changed mod\_wls\_ohs configuration by using the Stop OHS and Start OHS icons on the desktop.

## **Practice 11-4: Enabling OHS as the Front End of Applications (continued)**

- b) Then verify accessing the Medrec application through OHS.

The screenshot shows a web browser window with the URL <http://wls-sysadm:7777/medrec/> in the address bar. The page title is "Avitek™ Medical Records Sample Application". The content area lists several holidays: New Year Day Monday, January 1, 2001; Memorial Day Monday, May 28, 2001; Independence Day Wednesday, July 4, 2001; Labor Day Monday, September 3, 2001; Thanksgiving Day Thursday, November 22, 2001; Day Following Thanksgiving Friday, November 23, 2001; Floating Holiday Monday, December 24, 2001; Christmas Day Tuesday, December 25, 2001; and Holiday Break Wednesday, December 26, 2001 thru Monday, December 31, 2001. At the bottom right of the content area is a blue button labeled "Start using MedRec!" with a hand cursor icon pointing at it.

- c) Then verify accessing the Benefits application through OHS at <http://wls-sysadm:7777/benefits>

The screenshot shows a web browser window with the URL <http://wls-sysadm:7777/benefits> in the address bar. The page title is "MedRec Red Vacation Schedule". The content area lists the same set of holidays as the previous screenshot. At the bottom left is a blue link labeled "Back To Home Page".

Note that you are accessing two different back-end application servers using a single front-end OHS.

- 4) Now delete the Benefits application.
  - a) In Change Center, click Lock & Edit. Then navigate to the Deployments page.
  - b) Select the Benefits application and try to delete it while it is still in an active state. This will fail. You are warned that you cannot delete an application while it is running (active).
  - c) Select benefits again, select Stop > Force Stop Now, and click Yes.
  - d) Select benefits now and click Delete. Click Yes to delete Benefits.

### ***Practice 11-4: Enabling OHS as the Front End of Applications (continued)***

- e) In the Change Center, click Activate Changes. Benefits should now be deleted.
- 5) Suppose you want to remove jsf. (Note that a library cannot be stopped)
  - a) In Change Center, click Lock & Edit. Select jsf and click Delete. (This will fail.) The warning message tells you that you cannot do that because one or more applications still reference the library. The Delete is ignored.
  - b) The only way to remove jsf is to remove Medrec first, or to remove both jsf and Medrec at the same time.

## Practices for Lesson 12

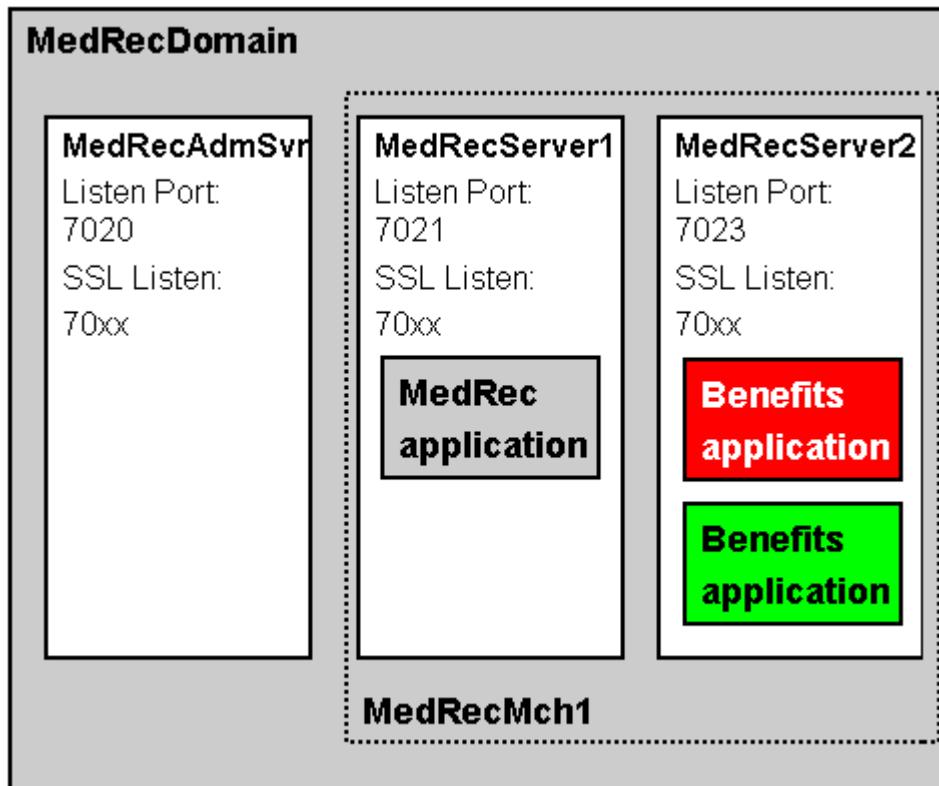
### Advanced Deployment for the Web

The Benefits application is targeted for MedRecServer2. As the developers worked on that application, it has gone through four versions. Rather than number them 1, 2, 3, 4, or 1.1, 1.2, 1.3, 1.4, the developers chose to version according to a color scheme. The version chronology is Black, then Blue, then Green, and then Red. You can tell which application is which because the title on the pages says “MedRec *Color*” where *Color* is either Black, Blue, Green, or Red. The first two iterations, Black and Blue, do not have versioning enabled in the manifest. The last two iterations, Green and Red, do have versioning enabled in the manifest.

The key tasks in this lab are the following:

- First, you deploy the Black version.
- Then upgrade to the Blue version. In this step, Blue version replaces the Black application in the middle of what you were doing and this can cause data inconsistencies
- Then you undeploy (unversioned) Blue to deploy the versioned Green.
- Finally, you deploy the upgrade to Red and verify that the two versions coexist appropriately.

### Big Picture:



## Practice 12-1: Redeploying Unversioned Applications

In this practice, you deploy two iterations of the benefits package: the Black and the Blue. (They are referred to as iterations so as to not imply any kind of version control.) A potential complication to the lab is that the browser will try to be helpful and cache pages it thinks it has seen before. This may cause the old iteration to show even after the new iteration has been replaced. You may have to close the Web browser to clear its cache.

- 1) Open a Linux terminal session and make sure that the environment variables are set.

```
cd /u01/app/oracle/product/fmw/11.1.0/wlserver_10.3/server/bin
source ./setWLSEnv.sh
```

- 2) Navigate to ~/wls-sysadm/labs/Lab12. and copy benefits.war.Black to benefits.war.
- 3) You cannot redeploy an application with the same name from two different locations such as /Lab11 and /Lab12, so you need to undeploy the old benefits that was from /Lab 11. Type all on one line (with no line breaks):

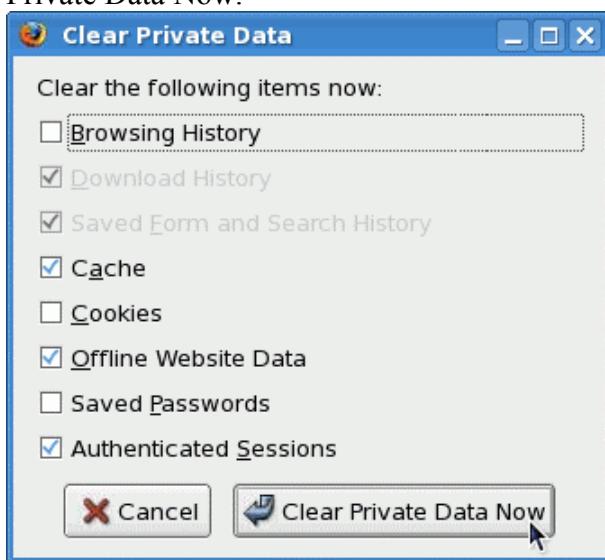
```
java weblogic.Deployer -adminurl t3://wls-sysadm:7020
 -username weblogic -password Welcome1 -name benefits
 -undeploy
```

- 4) Deploy the application using WLST by entering the following without any line breaks:

```
java weblogic.Deployer -adminurl t3://wls-sysadm:7020
 -username weblogic -password Welcome1 -name benefits
 -deploy benefits.war -targets MedRecSvr2
```

When you use the Deployer, it sets up Edit and Activate internally.

- 5) Open a Web browser and clear its cache by going to Tools > Clear Private Data. Deselect Browsing History and select Cache and Authenticated Sessions. Click Clear Private Data Now.



## **Practice 12-1: Redeploying Unversioned Applications (continued)**

- 6) Test the application by using the Web browser. Use the URL  
<http://wls-sysadm:7023/benefits>.



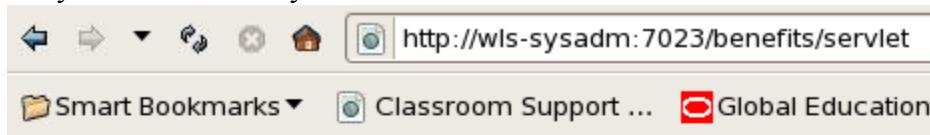
### Welcome To MedRec Black

Select What Benefits You Would Like To See

- View Vacation Schedule
- View Health Care Options
- View Vision Options
- View Dental Options

[Get Information](#)

- Note that the title contains “Black” in a black font.
- View all the pages by selecting all the check boxes and clicking Get Information. They too should all say “MedRec Black” and be in a black font.



### MedRec Black Health Benefits

Blue Cross and Blue Shield  
 Mathew Thornton Health Plan

[Back To Home Page](#)

- Go back to the Benefits home page and clear the cache again. Leave this page displayed.
- Update the Benefits application to use blue fonts.
  - In the gnome terminal session, copy the Blue benefits onto the current Benefits application by entering:  
`cp benefits.war.Blue benefits.war`
  - In a tab of the browser, access the Administration Console.
  - In Domain Structure, navigate to MedRecDomain > Deployments.
  - In Change Center, click Lock & Edit.

## Practice 12-1: Redeploying Unversioned Applications (continued)

- e) In the Deployments table, select benefits and click Update.

| Deployments                         |                   |        |        |                 | Showing 1 to 3 of 3          |
|-------------------------------------|-------------------|--------|--------|-----------------|------------------------------|
|                                     | Name              | State  | Health | Type            |                              |
| <input checked="" type="checkbox"/> | benefits          | Active | OK     | Web Application | <a href="#">View Details</a> |
| <input type="checkbox"/>            | jsf(1.2,1.2.9.0)  | Active |        | Library         | <a href="#">View Details</a> |
| <input type="checkbox"/>            | jstl(1.2,1.2.0.1) | Active |        | Library         | <a href="#">View Details</a> |

- f) The changes are all internal to the WAR file, so you do not need to change any paths. Click Finish.
- g) The State is deploy Initializing. During this time, the application is unavailable (but do not try it). In Change Center, click Activate Changes. State changes to Active.
- 8) Verify that the changes have become effective.
- In the Benefits tab of the browser, clear the cache again, and then select Vacation or Vision or any of the other pages.
  - You see that they should now say, “MedRec Blue.” If it *still* says the wrong color, the state of the servlet is completely confused, which is why you need to do versioning in the first place. This can be fixed or avoided by stopping the Benefits application, deleting it, activating changes, and then installing it fresh, and then restarting it.



### Welcome To MedRec Blue

Select What Benefits You Would Like To See

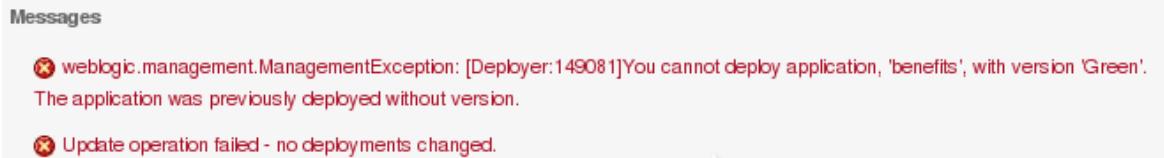
- View Vacation Schedule
- View Health Care Options
- View Vision Options
- View Dental Options

[Get Information](#)

## Practice 12-2: Redeploying Versioned Applications

In this practice, you deploy two versions of the benefits package: the Green version and the Red version. They are versioned using the `manifest.mf` file that has a WebLogic version line in it. This allows existing sessions using the Green version to complete gracefully while new sessions connect to the Red version. When the Green version is completely quiet for some time, it will change its state to retired.

- 1) Attempt to update the Benefits application to use green fonts without stopping the blue version. When it fails, stop and delete the blue version of the Benefits application.
  - a) In the gnome terminal session, copy the Blue benefits onto the current Benefits application by entering:  
`cp benefits.war.Green benefits.war`
  - b) In a tab of the browser, access the Administration Console.
  - c) In Domain Structure, navigate to MedRecDomain > Deployments.
  - d) In Change Center, click Lock & Edit.
  - e) Select benefits and click Update to update the existing Blue version with the new Green version without stopping the application. It will fail.
  - f) Nothing about the paths has changed. Click Finish. Note that you cannot replace a nonversioned application with a versioned one. You need to delete the old application. Click Cancel.



- g) Select benefits and then select Stop > Force Stop Now. Click Yes to stop the Blue version. State changes to Prepared.
- h) Select benefits and click Delete. Click Yes to delete the Blue benefits deployment. Click Activate Changes.
- 2) Now deploy the green version of the Benefits application to MedRecSvr2.
  - a) In Change Center, click Lock & Edit. In the Deployments table, click Install.
  - b) Select `benefits.war` (in `/home/oracle/wls-sysadm/labs/Lab12`, which is the Green version) and click Next.
  - c) The Install Application Assistant knows that this is an application (versus a library), so click Next.
  - d) Select MedRecSvr2 as the target and click Next.

## Practice 12-2: Redeploying Versioned Applications (continued)

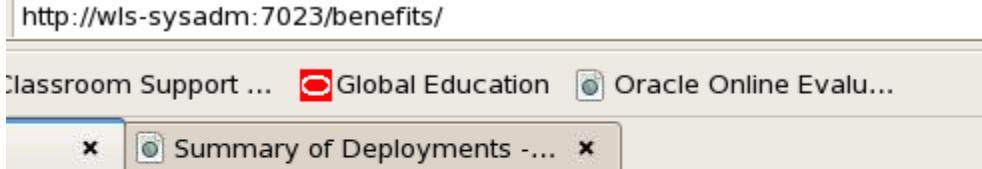
- e) Note that Archive Version (from the manifest.mf file) displays Green. You can change it here if you want to. Accept the defaults and click Finish.

**General**

What do you want to name this deployment?

|                  |          |
|------------------|----------|
| Name:            | benefits |
| Archive Version: | Green    |
| Deployment Plan  |          |
| Version:         |          |

- f) In Change Center, click Activate Changes. Note on the Deployments table that this is the (Green) version.
- g) Select benefits and then select Start > Servicing all requests. Click Yes to start the deployment.
- 3) In the Web tab, <http://wls-sysadm:7023/benefits> should now show Green.



- 4) Now update the application with the Red version, and this time, you can install the new version of the application while the previous version is running, and then retire the older version.
- In the Linux terminal session, copy the Red (final) benefits onto the current Benefits application  

```
$> cp benefits.war.Red benefits.war
```
  - In the Administration Console tab, in Change Center, click Lock & Edit. In Deployments, click Install.
  - Select benefits.war (which is the Red version) and click Next.
  - The Install Application Assistant knows that this is an application, so click Next.
  - Select MedRecSrv2 as the target and click Next.

## Practice 12-2: Redeploying Versioned Applications (continued)

- f) Note that Archive Version (from the manifest.mf file) displays Red. Accept the defaults and click Finish.

**General**

What do you want to name this deployment?

Name:

Archive Version:  

Deployment Plan

Version:

- g) In Change Center, click Activate Changes. Note that there are now two versions of the Benefits application: the older Green version (Active) and the newer Red version (Prepared).

**Deployments**

|                          |                                                                                                     |  |          | Install                                                                                  | Update | Delete | Start ▾ | Stop ▾ |
|--------------------------|-----------------------------------------------------------------------------------------------------|--|----------|------------------------------------------------------------------------------------------|--------|--------|---------|--------|
|                          | Name                                                                                                |  | State    | Health                                                                                   |        |        |         |        |
| <input type="checkbox"/> |  benefits (Green) |  | Active   |  OK |        |        |         |        |
| <input type="checkbox"/> |  benefits (Red)  |  | Prepared |  OK |        |        |         |        |

- h) Select the benefits (Red) and then select Start > Servicing all requests. Click Yes to start the deployment. The older Green version changes State from Active to Retired.

|                          | Name                                                                                                 | State   | Health                                                                                   |
|--------------------------|------------------------------------------------------------------------------------------------------|---------|------------------------------------------------------------------------------------------|
| <input type="checkbox"/> |  benefits (Green) | Retired |                                                                                          |
| <input type="checkbox"/> |  benefits (Red)   | Active  |  OK |

- i) Click Retired to see the time that the state changed.
- 5) Start a new Web browser on a different PC. (It cannot be a tab in an existing Web browser.) If your previous browser was on the Windows desktop, start one on the Linux desktop; or if you started the previous browser on the Linux desktop, start a Web browser on the Windows desktop. Access the benefits URL <http://VXnnnn:7023/benefits>, where nnnn is the number the instructor gave you of your Linux host. It should now show the new “MedRec Red” while the other Web browser continues to show the old “MedRec Green.”

## Practices for Lesson 13

### Configuring JDBC

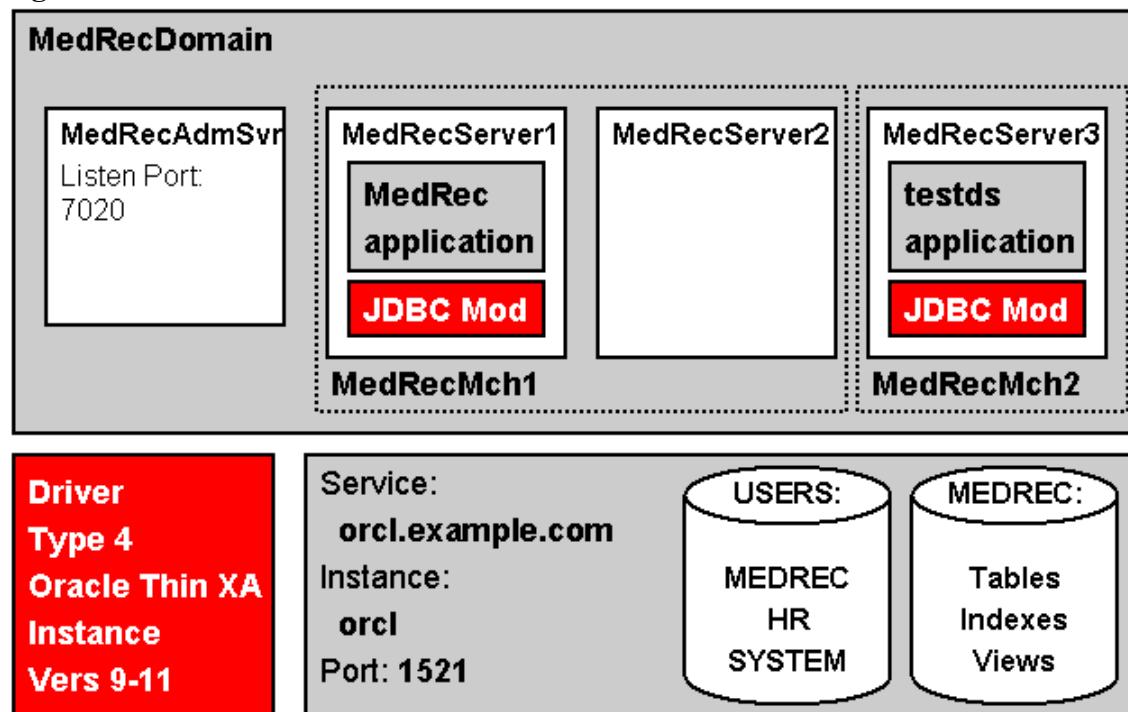
Your developers have written a small program that will assist them in testing their JDBC connections. The program prompts for a data source, a table name, and a user/password, and then it dumps the table to the browser by issuing

```
SELECT * FROM table_name;
```

to the data source. The schema you can query against is the MEDREC schema, and the tables are ADMINISTRATORS, PATIENTS, PHYSICIANS, PRESCRIPTIONS, and RECORDS. The key tasks are:

- Creating JDBC modules (via GUI and WLST)
- Deploying JDBC modules
- Testing JDBC modules

### Big Picture:



## Practice 13-1: Creating JDBC Modules

In this practice, you create the data source for project XYZ under department ABC. The name does not need to match anything in the application; it is a parameter passed into the application (that would be unusual in production).

- 1) Using the Administration Console, create a data source that can be used by the applications deployed in your domain. Ensure that the data source connection pools have a minimum of 5 and a maximum of 25 connections with increments of 5. Also, the connection pool should check for a need to increase every three minutes, and check for a need to shrink every ten minutes.
  - a) Sign on to the Administration Console, and in Change Center, click Lock & Edit.
  - b) In Domain Structure, navigate to MedRecDomain > Services > JDBC > Data Sources. Then click New.
  - c) Enter the following values in the Create a New JDBC Data Source pages:

| Step | Screen/Page Description      | Choices or Values                                                                                                                                                                                                                                   |
|------|------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| a.   | JDBC Data Source Properties  | Name: <b>testSample</b><br>JNDI Name: <b>abc.xyz.testSample</b><br>Database Type: <b>Oracle</b><br>Database Driver: <b>Oracle's (Thin XA) for Instance connections; Versions: ... 11</b><br>Look at the other database choices and then click Next. |
| b.   | Transaction Options          | Because you selected an XA driver, there is nothing to do here. Click Next.                                                                                                                                                                         |
| c.   | Connection Properties        | Database Name: <b>orcl</b><br>Host Name: <b>wls-sysadm</b><br>Port: <b>1521</b><br>Database User Name: <b>weblogic</b><br>Password and Confirm Password: <b>Welcome1</b><br>Click Next.                                                             |
| d.   | Test Database Connection     | Click Test Configuration. Messages should say, "Connection test succeeded." Click Finish.                                                                                                                                                           |
| e.   | Summary of JDBC Data Sources | Click testSample.                                                                                                                                                                                                                                   |
| f.   | Settings for testSample      | Navigate to the Configuration > Connection Pool tab.                                                                                                                                                                                                |

### **Practice 13-1: Creating JDBC Modules (Continued)**

| <b>Step</b> | <b>Screen/Page Description</b>  | <b>Choices or Values</b>                                                                                                 |
|-------------|---------------------------------|--------------------------------------------------------------------------------------------------------------------------|
| g.          | Configuration > Connection Pool | Initial Capacity: <b>5</b><br>Maximum Capacity: <b>25</b><br>Capacity Increment: <b>5</b><br>Click Save. Click Advanced. |
| h.          | Advanced                        | Test Frequency: <b>180</b><br>Shrink Frequency: <b>600</b><br>Click Save.                                                |
| i.          | Change Center                   | Click Activate Changes.                                                                                                  |

## Practice 13-2: Deploying JDBC Modules

In this practice, you deploy the data source module you just created to all servers and an application `testds` to MedRecSvr3 to test the data sources.

- 1) Deploy the data source module you just created to all servers and an application `testds` to MedRecSvr3 to test the data sources.
  - a) If MedRecSvr3 has not yet been started, start it using the Start MR Svr3 desktop icon.
  - b) In the Administration Console, in Change Center, click Lock & Edit.
  - c) In Domain Structure, navigate to MedRecDomain > Deployments. Click Install.
  - d) Specify the following values on the Install Application Assistant pages:

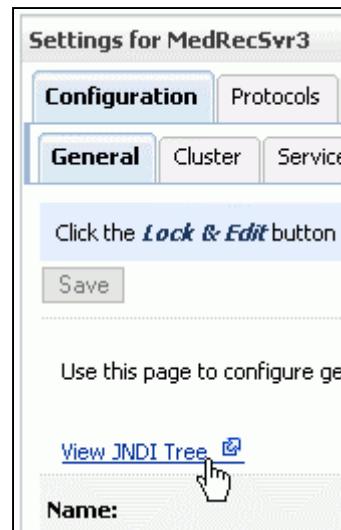
| Step | Screen/Page Description                                 | Choices or Values                                                                                        |
|------|---------------------------------------------------------|----------------------------------------------------------------------------------------------------------|
| a.   | Locate deployment to install and prepare for deployment | Path: <code>/home/oracle/wls-sysadm/labs/Lab13</code><br>Select <code>testds.war</code> .<br>Click Next. |
| b.   | Choose targeting style                                  | Accept the default of “Install this deployment as an application.” Click Next.                           |
| c.   | Select deployment targets                               | Select MedRecSvr3. Click Next.                                                                           |
| d.   | Optional Settings                                       | Accept all the defaults. Click Finish.                                                                   |
| e.   | Change Center                                           | Click Activate Changes.                                                                                  |
| f.   | Summary of Deployments                                  | Select <code>testds</code> (check box). Click Start and select “Servicing all requests.”                 |
| g.   | Start Deployments                                       | Click Yes.                                                                                               |

- e) In Domain Structure, navigate to MedRecDomain > Services > JDBC > Data Sources and click `testSample`.
- f) Click the Targets tab.
- g) In Change Center, click Lock & Edit.
- h) Select MedRecSvr3. Click Save.
- i) In Change Center, click Activate Changes.
- j) In Domain Structure, navigate to MedRecDomain > Services > JDBC > Data Sources to view your deployment status in Summary of Data Sources. Note that it is now associated with MedRecSvr3.

### Practice 13-3: Testing JDBC Modules

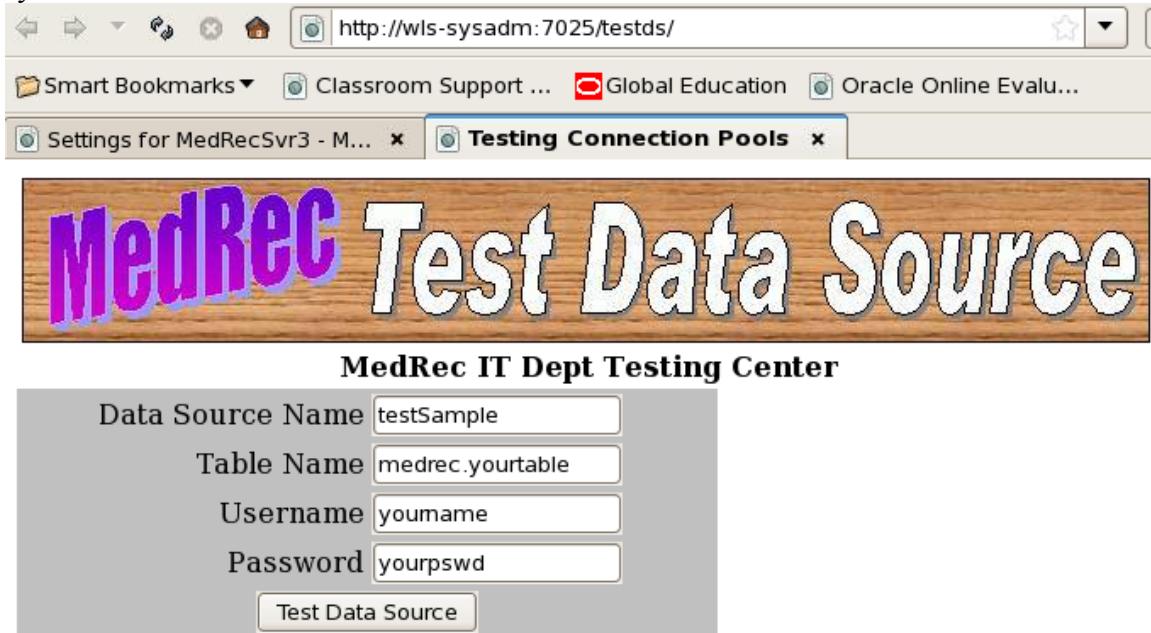
In this practice, you test your data sources and test the tables that are accessible through it.

- 1) In Domain Structure, navigate to MedRecDomain > Services > JDBC > Data Sources. In Summary of Data Sources, click testSample.
- 2) Click the Monitoring > Testing tabs.
- 3) Select MedRecSvr3. Click Test Data Source. The message at the top of the table should say, “Test of testSample on MedRecSvr3 was successful.”
- 4) In Domain Structure, navigate to MedRecDomain > Environment > Servers. Click MedRecSvr3.
- 5) Click View JNDI Tree. It opens up a new browser window or tab. (JNDI is a little hard to find on the screen. See the screen capture on the right. **Hint:** The browser Find command, Ctrl + F, can be helpful for busy screens. It highlights the word you are looking for—in this case, JNDI—making it easier to spot.)
- 6) Click the plus to expand abc. Click the plus to expand xyz. Click testSample. There is nothing to modify in the Overview tab, but there are some options in the Security tab.
- 7) Close the JNDI Tree Browser window or tab.



### Practice 13-3: Testing JDBC Modules (continued)

- 8) Open a new browser window or tab and access the URL <http://wls-sysadm:7025/testds>.



- 9) In the application, try using the following values (it will fail):

| Field Name       | Value                                                              |
|------------------|--------------------------------------------------------------------|
| Data Source Name | testSample                                                         |
| Table Name       | medrec.patients or<br>medrec.physicians or<br>medrec.prescriptions |
| Username         | weblogic                                                           |
| Password         | Welcome1                                                           |

Note the error message.

```
javax.naming.NameNotFoundException: Unable to resolve
'testSample'. Resolved ''; remaining name 'testSample'
```

- 10) Click Back in the browser (to save typing) and change the Data Source Name to **abc.xyz.testSample** and click Test Data Source. It should work.
- 11) Click Back in the browser (to save typing) and change the Data Source Name to **ABC.XYZ.testSample** and click Test Data Source. It should fail because the JNDI name is case-sensitive.
- 12) Click Back in the browser (to save typing) and change the Username to medrec and password to Welcome1. This is allowed from the database's point of view, but the testDS application will fail to authenticate medrec. You will get an error message: "Error: User: medrec, failed to be authenticated."

## Practice 13-4: Creating JDBC Modules by Using Scripts

In this practice, you create the JDBC module for the Medrec application. It is possible to have this module precreated and deployed as part of the application template that extended the domain, or you can create it separately after deploying the application by itself. Rather than use the GUI as you did for the testds application, this time the JDBC module will be created using the WLST script.

- 1) In the Administration Console, in Domain Structure, navigate to MedRecDomain > Services > JDBC > Data Sources.
- 2) If MedRecGlobalDataSourceXA already exists, delete it because you are going to add it using a script. In Change Center, click Lock & Edit, select MedRecGlobalDataSourceXA, and click Delete. Click Yes to delete the data source. Click Activate Changes.
- 3) Go to a Linux terminal session and make sure that the environment variables are set by running `setWLSEnv.sh`. Navigate to `~/wls-sysadm/labs/Lab13`.
- 4) Look at the contents of the `createDataSource.py` script. You should recognize all the commands from the screens you just completed for testds. Should you be able to sit down with a blank editor and write that? Probably not (yet). Should you be able to modify it to suit your purposes? Probably yes. Should you be able to run a Record session  to capture those steps from the GUI? Absolutely.
- 5) You can run the script by entering  
`java weblogic.WLST createDataSource.py`.
- 6) Verify that the script worked by going back to the Administration Console and checking that the new data source is there.
  - a) In the Administration Console, in Domain Structure, navigate to MedRecDomain > Services > JDBC > Data Sources.
  - b) Verify that MedRecGlobalDataSourceXA has been recreated.

## Practices for Lesson 14

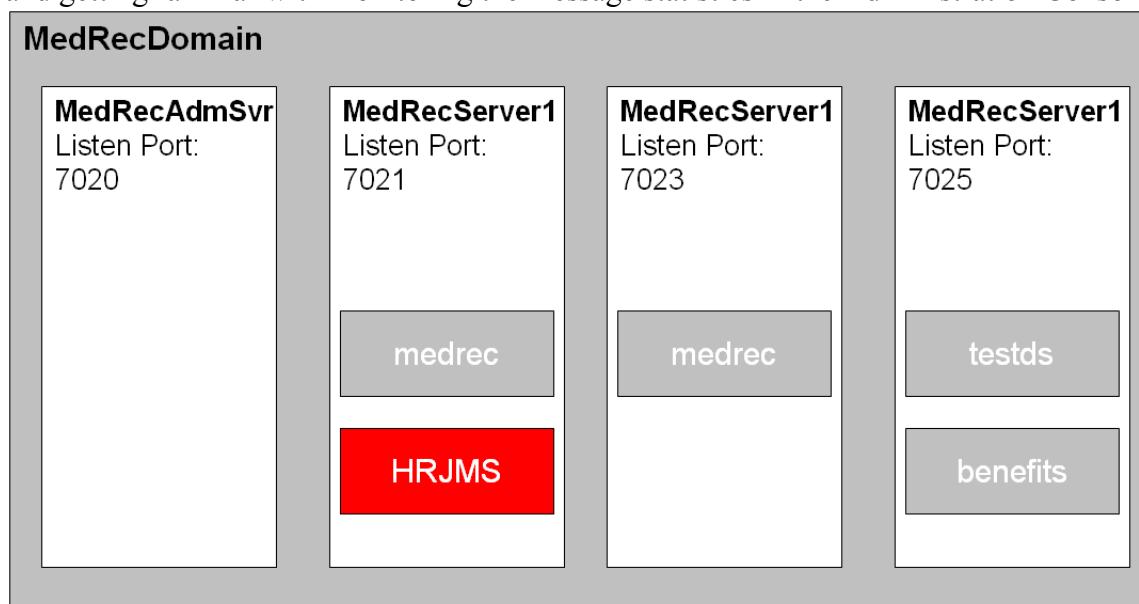
A JMS server implements the JMS infrastructure on a WebLogic server. Destinations (queues or topics) are targeted to a WebLogic server when the JMS server is targeted to the WebLogic server.

In this practice, you configure:

- JMS server
- JMS module
- Queue
- Topic

You then post messages to the queue and topic and monitor them in the Administration Console.

Right now, you will not have any consumers; you will simply be posting the messages and getting familiar with monitoring the message statistics in the Administration Console.



## Practice 14-1: Configuring JMS Resources and Deploying the JMS Application

In this practice, you configure a JMS server, a queue, and a topic. You then post messages to the queue and topic and monitor them in the Administration Console.

- 1) Verify that JMS Servers and Modules have been already created.
  - a) Ensure that the `orcl` database, `MedRecAdmSvr`, and `MedRecSrv1` are running.
  - b) Navigate to `MedRecDomain > Services > Messaging > JMS Servers`. You should see two JMS Servers. These were configured when you extended the domain with a template.

**JMS Servers(Filtered - More Columns Exist)**

Click the **Lock & Edit** button in the Change Center to activate all the buttons on this page.

| JMS Servers(Filtered - More Columns Exist)                                                                                                                               |                        |                  |            |                |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------|------------------|------------|----------------|
| Click the Lock & Edit button in the Change Center to activate all the buttons on this page.                                                                              |                        |                  |            |                |
| <input type="button" value="New"/> <input type="button" value="Delete"/> Showing 1 to 2 of 2 <input type="button" value="Previous"/> <input type="button" value="Next"/> |                        |                  |            |                |
|                                                                                                                                                                          | Name                   | Persistent Store | Target     | Current Server |
| <input type="checkbox"/>                                                                                                                                                 | MedRecJMSServer_auto_1 |                  | MedRecSrv1 | MedRecSrv1     |
| <input type="checkbox"/>                                                                                                                                                 | MedRecJMSServer_auto_2 |                  | MedRecSrv2 | MedRecSrv2     |

Showing 1 to 2 of 2

- c) Navigate to `MedRecDomain > Services > Messaging > JMS Modules`. Note that a JMS module has also been created.

**JMS Modules**

Click the **Lock & Edit** button in the Change Center to activate all the buttons on this page.

| JMS Modules                                                                                                                                                              |            |        |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|--------|
| Click the Lock & Edit button in the Change Center to activate all the buttons on this page.                                                                              |            |        |
| <input type="button" value="New"/> <input type="button" value="Delete"/> Showing 1 to 1 of 1 <input type="button" value="Previous"/> <input type="button" value="Next"/> |            |        |
|                                                                                                                                                                          | Name       | Type   |
| <input type="checkbox"/>                                                                                                                                                 | MedRec-jms | System |

Showing 1 to 1 of 1

- d) Click `MedRec-jms` to see the resources (Queues/Topics) created.

|                          | Name                            | Type  | JNDI Name                                   |
|--------------------------|---------------------------------|-------|---------------------------------------------|
| <input type="checkbox"/> | PatientNotificationQueue_auto_1 | Queue | com.bea.medrec.jms.PatientNotificationQueue |
| <input type="checkbox"/> | PatientNotificationQueue_auto_2 | Queue | com.bea.medrec.jms.PatientNotificationQueue |
| <input type="checkbox"/> | RecordToCreateQueue_auto_1      | Queue | com.bea.medrec.jms.RecordToCreateQueue      |
| <input type="checkbox"/> | RecordToCreateQueue_auto_2      | Queue | com.bea.medrec.jms.RecordToCreateQueue      |
| <input type="checkbox"/> | WSRMDefaultQueue_auto_1         | Queue | weblogic.wsee.DefaultQueue                  |
| <input type="checkbox"/> | WSRMDefaultQueue_auto_2         | Queue | weblogic.wsee.DefaultQueue                  |

## **Practice 14-1: Configuring JMS Resources and Deploying the JMS Application (continued)**

- 2) Configure a JMS Server with the name **HRJMSServer** and no persistent store.
  - a) Ensure that the `orcl` database, `MedRecAdmSvr`, and `MedRecSvr1` are running.
  - b) Navigate to `MedRecDomain > Services > Messaging > JMS Servers`. Then click **Lock & Edit** to configure a resource.
  - c) Click **New** under the JMS Servers table and specify the following properties:  
Name: **HRMSServer**, and Persistent Store: **(none)**

The screenshot shows a configuration dialog for a JMS Server. At the top, there is a field labeled "Name" with the value "HRMSServer". Below it, a note says "Specify persistent store for the new JMS Server." Under the "Persistent Store:" label, there is a dropdown menu set to "(none)".

- d) Click **Next** and target the JMS server to `MedRecSvr1`. Click **Finish**.
- e) Click **Activate Changes** and confirm that all changes have been activated.
- 3) Configure a JMS module and add a queue and a topic to the JMS module according to the following specifications:

| Resource              | Parameter            | Choices or Values |
|-----------------------|----------------------|-------------------|
| <b>JMS Module</b>     | Name                 | HRModule          |
|                       | Descriptor File Name | HRModule          |
|                       | Target               | MedRecSvr1        |
|                       |                      |                   |
| <b>Sub Deployment</b> | Name                 | HRSubDeployment   |
|                       | Targets              | HRJMSServer       |
|                       |                      |                   |
| <b>Queue</b>          | Name                 | HRQueue           |
|                       | JNDI Name            | HRQueue           |
|                       | Template             | None              |
|                       | Target               | HRJMSServer       |
|                       |                      |                   |
| <b>Topic</b>          | Name                 | HRTopic           |
|                       | JNDI Name            | HRTopic           |
|                       | Template             | None              |
|                       | Target               | HRJMSServer       |
|                       |                      |                   |

- a) Navigate to `MedRecDomain > Services > Messaging > JMS Modules` in the Administration Console. Click **Lock & Edit** to enable configuring resources.
- b) Click **New** in the JMS Modules table and specify Name: `HRModule` and Descriptor File Name: `HRModule`, and click **Next**.
- c) Select `MedRecSvr1` as the target managed server.

## **Practice 14-1: Configuring JMS Resources and Deploying the JMS Application (continued)**

- d) Click Next and select “Would you like to add resources to this JMS system module?” and then click Finish.
- e) On the Settings for HRModule page, click the Subdeployments tab. In the Subdeployments table, click New to create a subdeployment.
- f) Enter HRSUBDeployment as the subdeployment name and click Next.
- g) On the Targets page, select the HRJMServer as the target under the JMS Servers table. Click Finish.

### **Targets**

Please select targets for the Subdeployment

| Servers                             |
|-------------------------------------|
| <input type="checkbox"/> MedRecSvr1 |

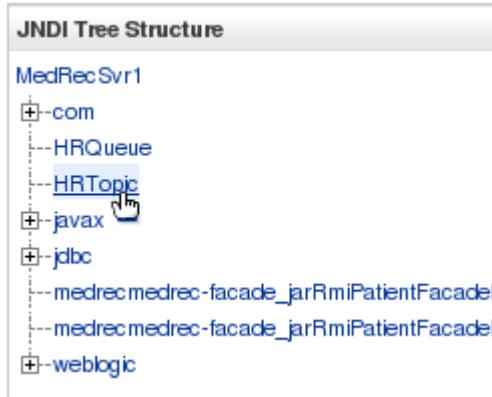
  

| JMS Servers                                     |
|-------------------------------------------------|
| <input checked="" type="checkbox"/> HRJMSServer |
| <input type="checkbox"/> MedRecJMServer_auto_1  |

- h) Click the Configuration tab. In the Summary of Resources table on the Settings for HRModule page, click New to configure a new JMS queue for the JMS module.
- i) On the Create a New JMS System Module Resource page, under the heading “Choose the type of resource you want to create,” select Queue and click Next.
- j) In JMS Destination Properties, specify the parameters—Name: HRQueue, JNDI Name: HRQueue, Template: None—and click Next.
- k) Select HRSUBDeployment from the subdeployments list. Click Finish.
- l) In the Summary of Resources table on the Settings for HRModule page, click New to configure a new JMS topic for the JMS module.
- m) On the Create a New JMS System Module Resource page, under the heading “Choose the type of resource you want to create,” select Topic. Click Next.
- n) In JMS Destination Properties, specify the parameters—Name: HRTopic, JNDI Name: HRTopic, Template: None. Click Next.
- o) Select HRSUBDeployment from the subdeployments list. Click Finish.

## **Practice 14-1: Configuring JMS Resources and Deploying the JMS Application (continued)**

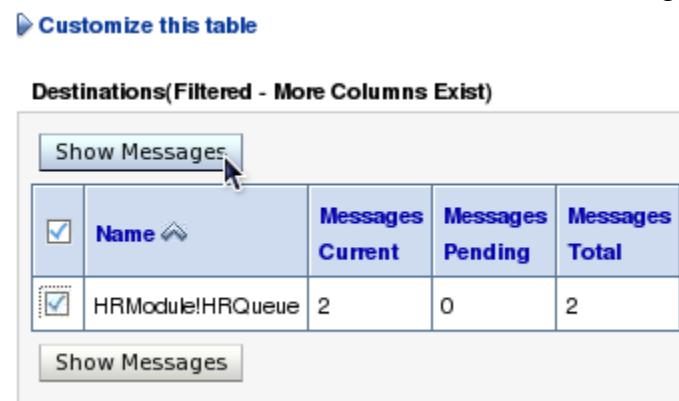
- p) Activate the changes. You should be able to see JNDI entries on the MedRecSvr1 managed server called HRQueue and HRTopic.



- 4) Deploy the Web application messaging.war, which you use to post messages to the queue or the topic.
  - a) Navigate to MedRecDomain > Deployments. Click Lock & Edit.
  - b) Select Install, navigate to /home/oracle/wls-sysadm/labs/Lab14, and select messaging.war. Click Next and accept all the defaults and click Next again. Target the application to MedRecSvr1.
  - c) Click Next and accept all the defaults. Click Finish. Click Activate Changes.
  - d) Start the application by selecting the check box against the application name under the Deployments table. Select Start > Servicing all requests.
  - e) Click OK to confirm starting the application.
- 5) Verify that the Web application has deployed correctly by navigating to http://wls-sysadm:7021/messaging in a Web browser and posting messages to either the queue or the topic using the deployed Web application.
  - a) If not already open, open a new Web browser tab or window and navigate to http://wls-sysadm:7021/messaging.
  - b) Using the application, post a few messages to the queue and to the topic. **Do not post any message to the distributed queue.**
  - c) In the Administration Console window or tab, navigate to MedRecDomain > Services > Messaging > JMS Modules. In the JMS Modules table, click HRModule. On the Summary of Resources page, click HRQueue, and then click the Monitoring tab. This will show the number of messages that have been posted into HRQueue.

## Practice 14-1: Configuring JMS Resources and Deploying the JMS Application (continued)

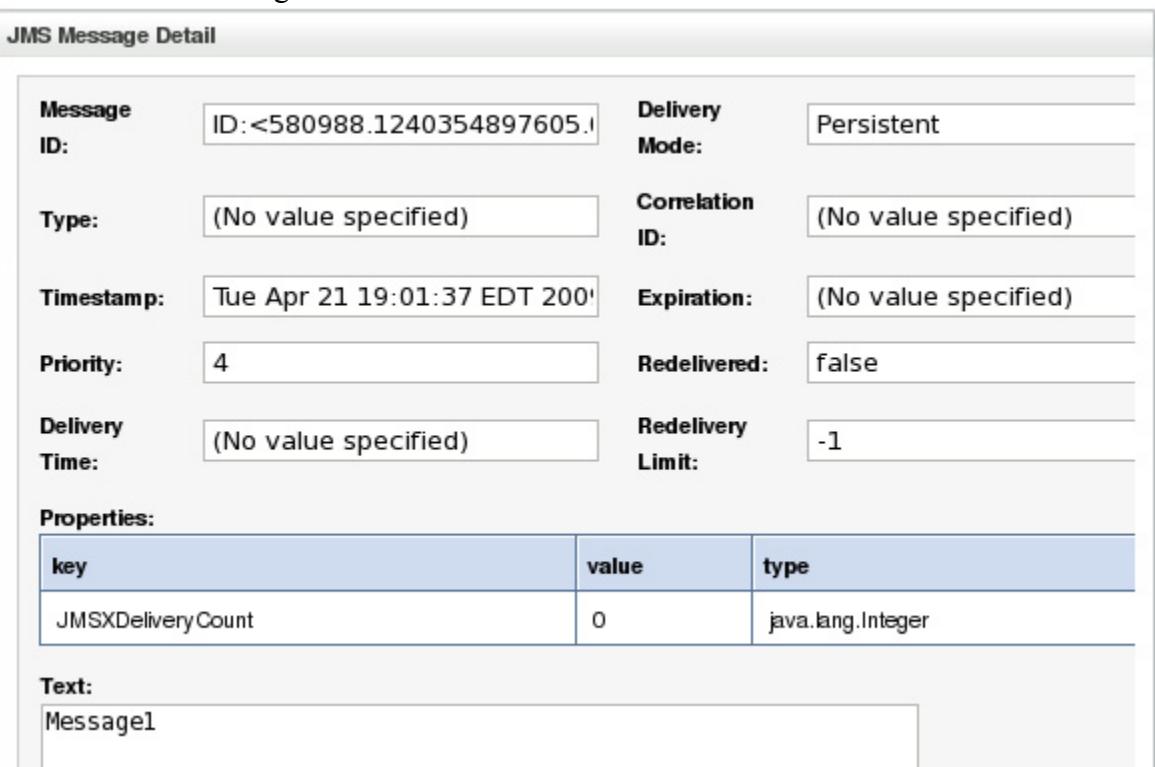
- d) Select HRModule!HRQueue and click Show Messages.



The screenshot shows a table titled "Destinations(Filtered - More Columns Exist)". A row for "HRModule!HRQueue" is selected, indicated by a blue border around the entire row. The table has columns: Name, Messages Current, Messages Pending, and Messages Total. The "Name" column header is sorted in ascending order. The "Show Messages" button at the bottom left of the table is highlighted with a red box.

| Destinations(Filtered - More Columns Exist) |                  |                  |                  |                |
|---------------------------------------------|------------------|------------------|------------------|----------------|
|                                             | Name             | Messages Current | Messages Pending | Messages Total |
| <input checked="" type="checkbox"/>         | HRModule!HRQueue | 2                | 0                | 2              |
| <a href="#">Show Messages</a>               |                  |                  |                  |                |

- e) At the bottom of the Summary of Messages page, click the Message link in the table to see the message details.



The screenshot shows the "JMS Message Detail" page. It displays a table of message properties. The message ID is <580988.1240354897605.1. The properties listed are: Type (No value specified), Timestamp (Tue Apr 21 19:01:37 EDT 2001), Priority (4), Delivery Time (No value specified), Delivery Mode (Persistent), Correlation ID (No value specified), Expiration (No value specified), Redelivered (false), and Redelivery Limit (-1). Below the table, there is a section for "Properties" with a table showing a single entry: key (JMSXDeliveryCount) with value (0) and type (java.lang.Integer). At the bottom, there is a "Text" section containing the message text: "Message1".

|                       |                              |                          |                      |
|-----------------------|------------------------------|--------------------------|----------------------|
| <b>Message ID:</b>    | ID:<580988.1240354897605.1   | <b>Delivery Mode:</b>    | Persistent           |
| <b>Type:</b>          | (No value specified)         | <b>Correlation ID:</b>   | (No value specified) |
| <b>Timestamp:</b>     | Tue Apr 21 19:01:37 EDT 2001 | <b>Expiration:</b>       | (No value specified) |
| <b>Priority:</b>      | 4                            | <b>Redelivered:</b>      | false                |
| <b>Delivery Time:</b> | (No value specified)         | <b>Redelivery Limit:</b> | -1                   |
| <b>Properties:</b>    |                              |                          |                      |
| key                   | value                        | type                     |                      |
| JMSXDeliveryCount     | 0                            | java.lang.Integer        |                      |
| <b>Text:</b>          |                              |                          |                      |
| Message1              |                              |                          |                      |

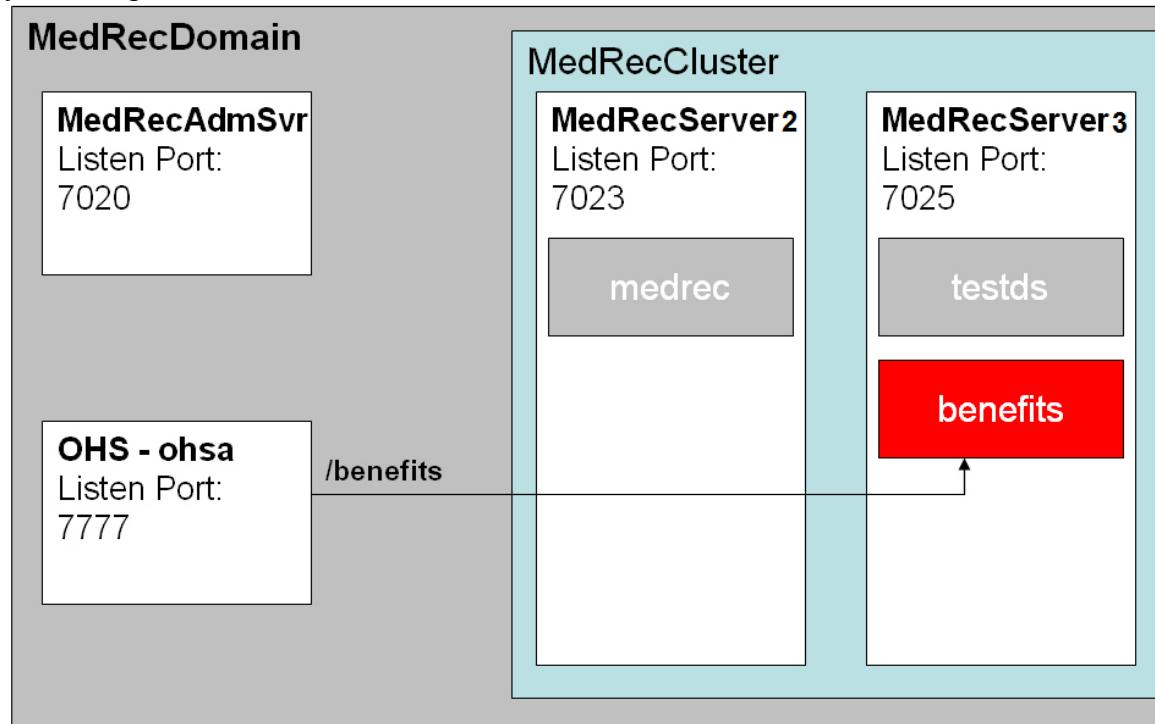
**Note:** In the topic (unlike the queue), messages do not appear to be getting stored. This is because you do not have any durable subscribers registered for this topic.

## **Practices for Lesson 15**

No practices for this lesson

## Practices for Lesson 16

In this practice, you create a cluster and assign two servers to the cluster. You also make the preliminary check on the port and status of Oracle HTTP Server. In the next practice, you configure Oracle HTTP Server to function as the Web tier front end for the cluster.



## Practice 16-1: Initiating Clusters

In this practice, you create a cluster that uses the default Unicast messaging mode and assign two managed servers to the cluster.

- 1) Create a new cluster with the following properties:

| Parameter      | Choices or Values         |
|----------------|---------------------------|
| Name           | MedRecCluster             |
| Messaging Mode | Unicast (default)         |
| Servers        | MedRecSrv2 and MedRecSrv3 |

- a) Ensure that MedRecAdmSvr is running. If the MedRecSrv2 and MedRecSrv3 managed servers are already running, stop them.

```
$> /home/oracle/wls-sysadm/start_adm.sh
```

- b) Log in to the Administration Console, and navigate to MedRecDomain > Environment > Clusters. Click Lock & Edit.

- c) Create a new cluster with the name and properties listed in the preceding table. Then click OK.

Home > Summary of Clusters

Create a New Cluster

OK Cancel

**Cluster Properties**

The following properties will be used to create your new Cluster.  
\* Indicates required fields

What would you like to name your new Cluster?

\* Name:

Clusters use messaging for sharing session, load balancing and failover, JMS, and other information between cluster members. Clusters can use either Unicast or Multicast messaging. Multicast is a simple broadcast technology that enables multiple applications to subscribe to a given IP address and port number and listen for messages, but requires hardware configuration and support. Unicast does not have these requirements. What messaging mode should this cluster use?

**Messaging Mode:**

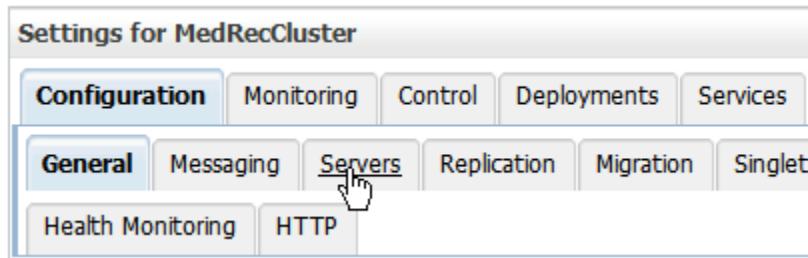
Unicast Broadcast Channel:

Multicast Address:

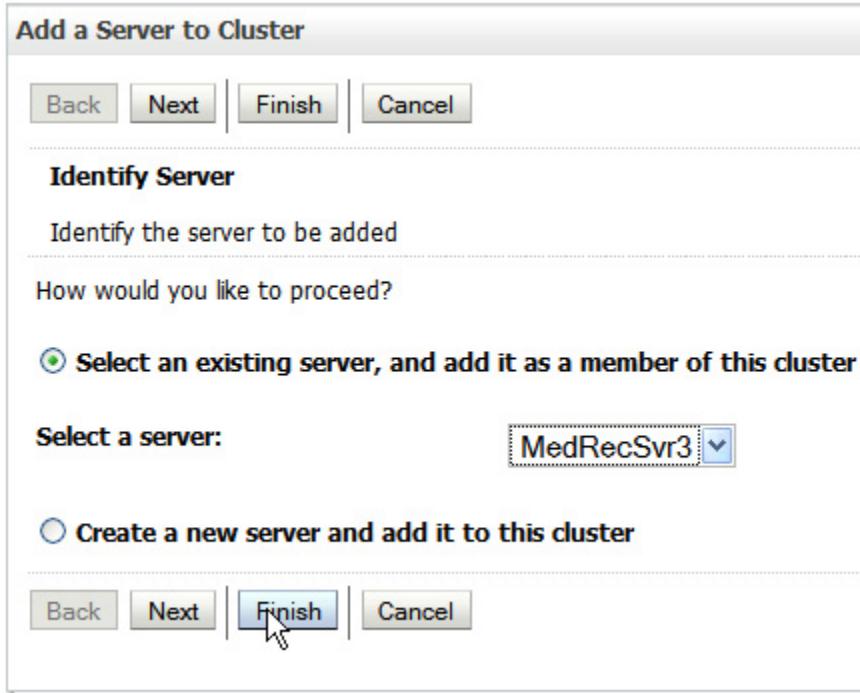
Multicast Port:

## Practice 16-1: Initiating Clusters (continued)

- d) Go back to MedRecCluster and navigate to the Configuration > Servers tab.



- e) Click Add, then select an existing server MedRecSvr2, and click Next. Again click Add and select MedRecSvr3, and then click Finish. Click Activate changes.



## Practice 16-1: Initiating Clusters (continued)

- f) Navigate to MedRecDomain > Environment > Servers and view the list of servers. Note that MedRecSrv2 and MedRecSrv3 are now part of the MedRecCluster.

| Servers (Filtered - More Columns Exist)                                                                |                     |               |            |                                     |        |             |
|--------------------------------------------------------------------------------------------------------|---------------------|---------------|------------|-------------------------------------|--------|-------------|
| Click the <b>Lock &amp; Edit</b> button in the Change Center to activate all the buttons on this page. |                     |               |            |                                     |        |             |
|                                                                                                        | New                 | Clone         | Delete     | Showing 1 to 4 of 4 Previous   Next |        |             |
| <input type="checkbox"/>                                                                               | Name                | Cluster       | Machine    | State                               | Health | Listen Port |
| <input type="checkbox"/>                                                                               | MedRecAdmSvr(admin) |               |            | RUNNING                             | OK     | 7020        |
| <input type="checkbox"/>                                                                               | MedRecSrv1          |               | MedRecMch1 | SHUTDOWN                            |        | 7021        |
| <input type="checkbox"/>                                                                               | MedRecSrv2          | MedRecCluster | MedRecMch1 | SHUTDOWN                            |        | 7023        |
| <input type="checkbox"/>                                                                               | MedRecSrv3          | MedRecCluster | MedRecMch2 | SHUTDOWN                            |        | 7025        |

- 2) Start MedRecSrv3. Wait for it to come up. Then start MedRecSrv2. Watch each server as it tries to synchronize with other servers in the cluster and finally joins the cluster.
- Start the MedRecSrv3 server by using the `start_mr3.sh` shell script in the `/home/oracle/wls-sysadm` folder:
  - Watch the server start up in another terminal window. At some point, you should see it start listening for cluster announcement and waiting to synchronize with other servers in the cluster. Because the other servers have not started yet, there is nothing for it to synchronize with yet.
- ```
<Notice> <Cluster> <BEA-000197> <Listening for
announcements from cluster using unicast cluster messaging>
<Notice> <Cluster> <BEA-000133> <Waiting to synchronize
with other running members of MedRecCluster.>
```
- Start the MedRecSrv2 server by using the `start_mr2.sh` shell script in the `/home/oracle/wls-sysadm` folder. (You could have started using the Administration Console, but using the command line, you can see some startup messages that help understand the startup process of a clustered server.)
 - Watch the MedRecSrv2 server start up in the terminal window. As it starts, it will synchronize with MedRecSrv3, which is the other server in the cluster, and will download the cluster JNDI tree.

```
<Notice> <Cluster> <BEA-000133> <Waiting to synchronize
with other running members of MedRecCluster.>
<Notice> <Cluster> <BEA-000142> <Trying to download cluster
JNDI tree from server MedRecSrv3.>
<Notice> <Cluster> <BEA-000164> <Synchronized cluster JNDI
tree from server MedRecSrv3.>
```

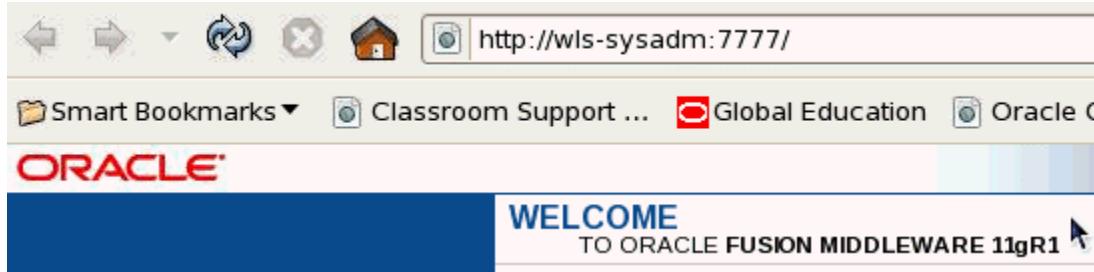
Practice 16-1: Initiating Clusters (continued)

- 3) Start Oracle HTTP Server, verify that OHS is running, and find its HTTP listen port.
- View the `start_ohs.sh` script in the `/home/oracle/wls-sysadm` folder and note the relevant OPMNCTL command that is used to start OHS. Then run the script to start OHS. (You can also use the Start OHS icon on the desktop to start OHS.)
 - Run the `status_ohs.sh` script in the `/home/oracle/wls-sysadm` folder, and note the number beside `http:` in the response. For example, in the following case, OHS is running and its HTTP listen port is 7777:

```
$> status_ohs.sh

Processes in Instance: wtinst
-----+-----+-----+-----+
-----+-----+-----+
ias-component | process-type | pid | status |
uid          | memused    | uptime | ports
-----+-----+-----+-----+
-----+-----+-----+
ohsa         | OHS        | 12462 | Alive   |
1775979059 | 348732    | 0:01:41 |
https:8889,https:4443,http:7777
```

- Access the URL `http://wls-sysadm:7777` in your Web browser.



- Shut down OHS, and then configure `mod_wl_ohs` to enable routing requests to MedRecCluster.
 - View the `stop_ohs.sh` script in the `/home/oracle/wls-sysadm` folder and note the relevant OPMNCTL command that is used to stop OHS and other WebTier components. Then run the script to stop OHS and the WebTier components. (You could also use the Stop OHS icon on the desktop to stop OHS.)
 - In a gnome terminal session, change directory to the OHS instance configuration folder (`/u01/app/oracle/instances/config/OHS/ohsa`). Copy the `mod_wl_ohs.conf` file to `mod_wl_ohs.bak16`. Then edit `mod_wl_ohs.conf` so that it appears as in the following screenshot:
- ```
$> cd /u01/app/oracle/instances/config/OHS/ohsa
$> cp mod_wl_ohs.conf mod_wl_ohs.bak16
```

## Practice 16-1: Initiating Clusters (continued)

```
$> gedit mod_wl_ohs.conf
NOTE : This is a template to configure mod_weblogic.

LoadModule weblogic_module "${ORACLE_HOME}/ohs/modules/mod_wl_ohs.so"

This empty block is needed to save mod_wl related configuration from EM to this
file when changes are made at the Base Virtual Host Level
<IfModule mod_weblogic.c>
 WebLogicCluster wls-sysadm:7023,wls-sysadm:7025
</IfModule>

<Location /benefits>
 SetHandler weblogic-handler
</Location>
```

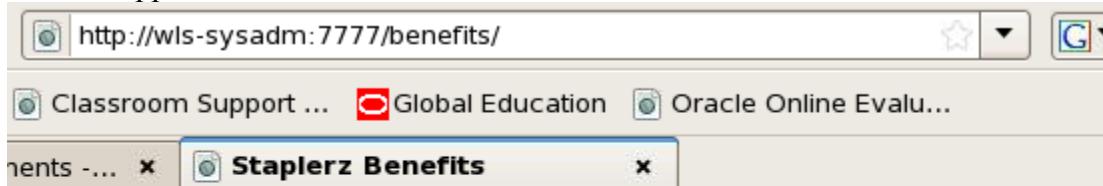
**Note:** To simplify this task, you can copy the mod\_wl\_ohs.conf file in the /home/oracle/wls-sysadm/labs/Lab16 folder to /u01/app/oracle/instances/config/OHS/ohsa:

```
$> cp /home/oracle/wls-sysadm/labs/Lab16/mod_wl_ohs.conf
/u01/app/oracle/instances/config/OHS/ohsa
```

- 5) Start OHS and verify that you can access the Benefits application through OHS (port 7777).
  - a) Use the scripts available in the /home/oracle/wls-sysadm folder and start OHS.

```
$> /home/oracle/wls-sysadm/start_ohs.sh
```

  - b) Access the Benefits (red) application through OHS  
(URL: http://wls-sysadm:7777/benefits) and note that you can access the Benefits application.



Welcome To MedRec Red

Select What Benefits You Would Like To See

- 6) Stop the MedRecSvr2 server, clear the browser cache, and try to access the Benefits application through OHS. What happens?

### **Practice 16-1: Initiating Clusters (continued)**

- a) You will not be able to access the application because even though you have created a cluster, the application was not targeted to cluster and, therefore, only MedRecSvr2 was serving requests to the Benefits application.



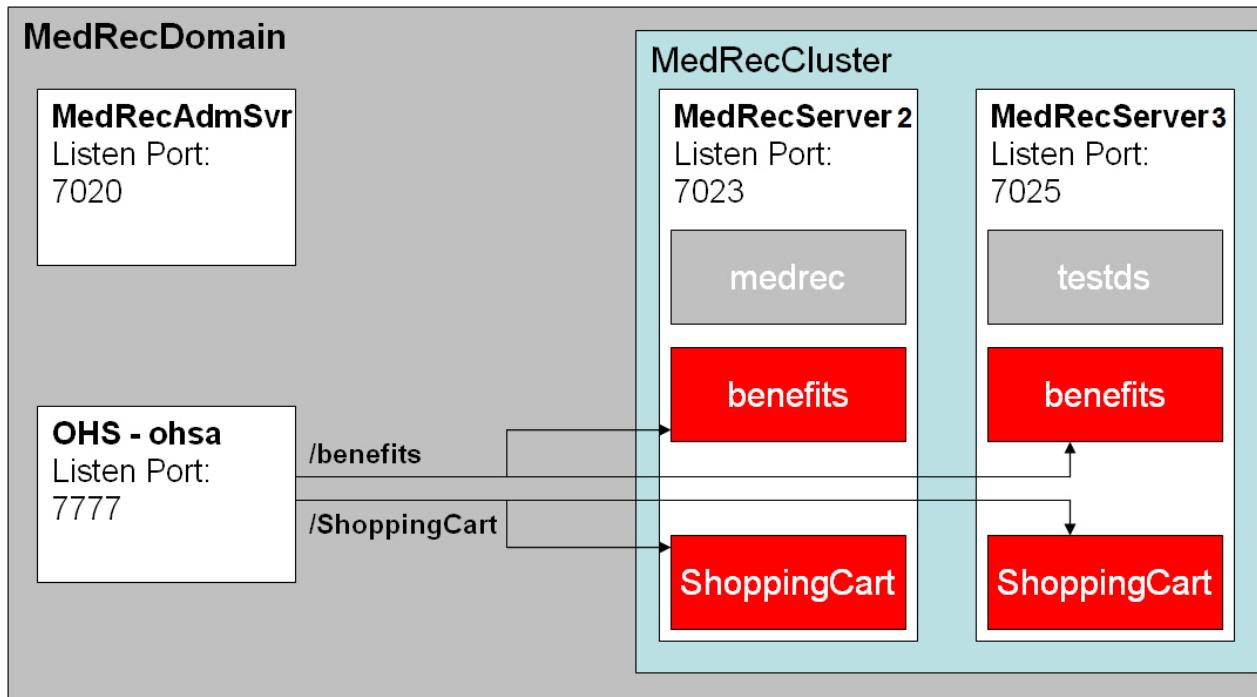
**Error 404--Not Found**

**From RFC 2068 *Hypertext Transfer Protocol -- HTTP/1.1***

## Practices for Lesson 17

In this exercise, you perform the following tasks:

- Retarget applications to a cluster
- Set up in-memory session replication
- Deploy an application to a cluster
- Set up in-memory session replication



## Practice 17-1: Targeting Applications to a Cluster

In this practice, you retarget an application to the cluster and see that it can be accessed using OHS as long as one server in the cluster is able to serve the request.

- 1) Ensure that MedRecSrv2 is started up. Then retarget the Benefits application to MedRecCluster (instead of just MedRecSrv3 server). Verify the access through OHS even after stopping MedRecSrv2—the server that initially was serving requests to the Benefits application.
  - a) Start up the MedRecSrv2 server using either the desktop icon or the `start_mr2.sh` script in the `~/wls-sysadmin` folder.
  - b) In the Administration Console, navigate to `MedRecDomain > Deployments`.
  - c) In the Deployments table, select `benefits (Red)` and select `Stop > Force Stop Now`. Click `Yes` in confirmation.
  - d) Click `Lock & Edit` in Change Center. Then click `benefits (Red)` in the Summary of Deployments table.
  - e) Click the Targets tab. In the Clusters section, select “All servers in the cluster” and click `Save`.
  - f) Click `Activate Changes` in Change Center. Navigate back to the Summary of Deployments page and note that the application is in the Active state.

| Deployments              |                  |          |        |                 |                  |  |
|--------------------------|------------------|----------|--------|-----------------|------------------|--|
|                          | Name             | State    | Health | Type            | Deployment Order |  |
| <input type="checkbox"/> | benefits (Green) | Prepared | OK     | Web Application | 100              |  |
| <input type="checkbox"/> | benefits (Red)   | Active   | OK     | Web Application | 100              |  |

- g) In another Browser window, access the Benefits application through OHS (URL: `http://wls-sysadm:7777/benefits`). Note that you are able to access the Benefits application.

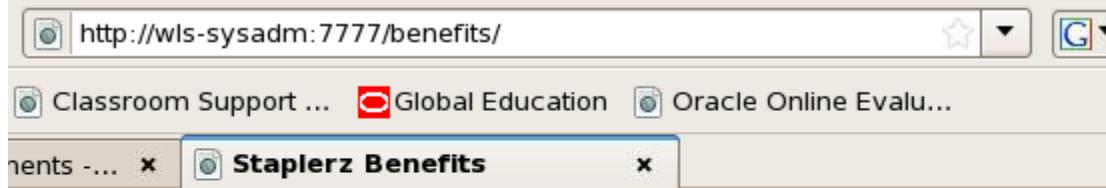


Welcome To MedRec Red

Select What Benefits You Would Like To See

### **Practice 17-1: Targeting Applications to a Cluster (continued)**

- h) Stop the MedRecSvr2 server, clear the browser cache, and try to access the Benefits application through OHS. You can continue to access the Benefits application because the application has been targeted to the cluster.



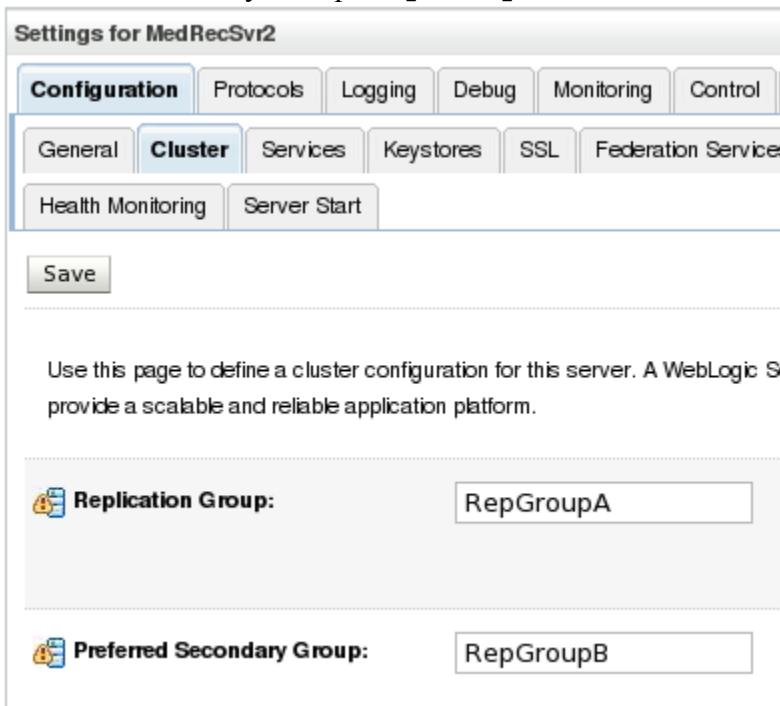
## Practice 17-2: Configuring Session Replication by Using In-Memory Structures

You have set up a cluster with a proxy server and retargeted a simple Web application to the cluster. In this practice, you deploy a new session-enabled application to the cluster. The application uses sessions using shopping-cart information. You configure session replication using the cluster.

- 1) Create the following replication groups:

| Server     | Replication Group | Preferred Secondary |
|------------|-------------------|---------------------|
| MedRecSrv2 | RepGroupA         | RepGroupB           |
| MedRecSrv3 | RepGroupB         | RepGroupA           |

- a) Stop the MedRecSrv2 and MedRecSrv3 servers, if they are running.
- b) Navigate to MedRecDomain > Environment > Servers > MedRecSrv2 > Configuration > Cluster. Click Lock & Edit to enable reconfiguration.
- c) Set the following properties:
  - i) Replication Group: RepGroupA
  - ii) Preferred Secondary Group: RepGroupB



- d) Click Save and then click Activate Changes.
  - e) Using the preceding steps, set up replication properties for MedRecSrv3 so that the Replication Group is RepGroupB and Preferred Secondary is RepGroupA.
  - f) Start the MedRecSrv2 and MedRecSrv3 servers.
- 2) Verify that the Shopping Cart application can use in-memory session replication.

## Practice 17-2: Configuring Session Replication by Using In-Memory Structures (continued)

- a) Look at /home/oracle/wls-sysadm/labs/Lab17/In-Memory/ShoppingCart/WEB-INF/weblogic.xml in an XML editor or text editor and note that the session-descriptor element is set.

```
<session-descriptor>
 <timeout-secs>300</timeout-secs>
 <invalidation-interval-secs>60</invalidation-interval-secs>

 <persistent-store-type> replicated_if_clustered
 </persistent-store-type>
</session-descriptor>
```

- 3) Package and deploy the ShoppingCart Web application.

- a) Change directory to /home/oracle/wls-sysadm/labs/Lab17/In-Memory/ShoppingCart and package the Web application into a .war file by using the jar command:

```
$>jar -cf ./ShoppingCart.war *
```

(This step has already been done for you and the resultant ShoppingCart.war is placed in the HOME/wls-sysadm/labs/Lab17/In-Memory folder.)

- b) Deploy the ShoppingCart.war application from the /home/oracle/wls-sysadm/labs/Lab17/In-Memory folder.



- c) Target the application to all servers in MedRecCluster.



- d) Activate your changes and start the application to serve all requests.

## Practice 17-2: Configuring Session Replication by Using In-Memory Structures (continued)

- 4) Configure the new application in OHS.

- a) Edit the `mod_wl_ohs.conf` file in the `/u01/app/oracle/instances/config/OHS/ohsa` folder and include the `<Location>` element for `/ShoppingCart`. Finally, it should appear as follows:

```
NOTE : This is a template to configure mod_weblogic.

LoadModule weblogic_module "${ORACLE_HOME}/ohs/modules/mod_wl_ohs.so"

This empty block is needed to save mod_wl related configuration from EM to this
file when changes are made at the Base Virtual Host Level
<IfModule mod_weblogic.c>
 WebLogicCluster wls-sysadm:7023,wls-sysadm:7025
</IfModule>

<Location /benefits>
 SetHandler weblogic-handler
</Location>

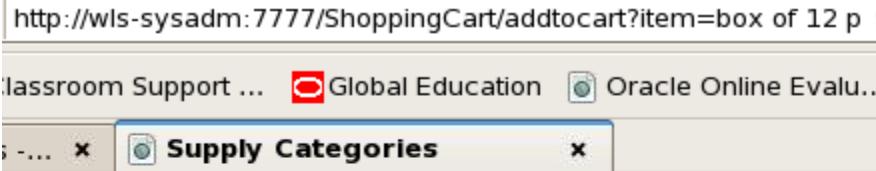
<Location /ShoppingCart>
 SetHandler weblogic-handler
</Location>
```

- b) Restart OHS using `stop_OHS.sh` and `start_ohs.sh` scripts.
- 5) Test the in-memory session replication by accessing the `ShoppingCart` application and adding a few items to cart. Identify which server is active for the request and shut down that server. Note that the request fails over to the other running server.

- a) Open a Web browser and navigate to:

`http://wls-sysadm:7777/ShoppingCart`

- b) Select Go Shopping and add an item to your shopping cart.



### Dizzyworld Store

added new element  
box of 12 pens (blue)  
[Back To Home Page](#)

- c) Go back to the home page and view the items in your shopping cart.

## Practice 17-2: Configuring Session Replication by Using In-Memory Structures (continued)

- d) Check the gnome terminal session for MedRecSrv2 and MedRecSrv3. You will notice messages indicating addition of items in the server that is handling the request.

```
<Apr 22, 2009 1:36:37 AM EDT> <Notice> <WebLogicServer> <BEA-000330> <Started WebLogic Managed Server "MedRecSrv3" for domain "MedRecDomain" running in Production Mode
<Apr 22, 2009 1:36:40 AM EDT> <Notice> <WebLogicServer> <BEA-000365> <Server state changed to RUNNING>
<Apr 22, 2009 1:36:40 AM EDT> <Notice> <WebLogicServer> <BEA-000360> <Server started in RUNNING mode>
within welcome.jsp
within shoppingcart.jsp
within shopping cart servlet
added new element: box of 12 pens (blue)
```

- e) To simulate a server failure, kill the server instance handling your requests by entering CTRL-C in the terminal window of that server.
- f) Back in the application browser, continue shopping and add something else to your shopping cart.
- g) View the shopping cart. All the items you added to the cart should be in the cart.



### Dizzyworld Store Shopping Cart

| Item                  | Price |
|-----------------------|-------|
| box of 12 pens (blue) | 4.99  |
| 3 mechanical pencils  | 8.99  |

- h) Check the server consoles to see which server is now handling the request.
- i) Restart the server that was killed.

## Practices for Lesson 18

You need to create users and groups in your security realm to enable appropriate authentication for some applications.

In this case, you:

- Create new users using the Administration Console
- Create groups of employees and managers
- Assign groups to users
- Configure groups-to-role mapping
- Define resources that are protected by the security you have configured
- Verify that it is working

You use the `timeoff.war` Web application in this lab.

You configure security so that only users in a specific group can make requests on the URL pattern `/managers/*`.

## Practice 18-1: Managing Users and Groups

In this practice, you create a few users in your domain. These users are to be authenticated into the WebLogic Server environment. Each user is an employee of the company and belongs to the `employees` group. Additionally, some users belong to a group called `managers`.

- 1) Create two groups in the security realm of your environments. Then create users and assign users to these groups as per the following table:

| User   | Password | Groups              |
|--------|----------|---------------------|
| John   | Welcome1 | Administrators      |
| Joe    | Welcome1 | employees, managers |
| Ted    | Welcome1 | employees,          |
| Mary   | Welcome1 | Employees, managers |
| Albert | Welcome1 | employees           |

- a) In the Administration Console, navigate to Security Realm. Click `myrealm` in the Realms table and click the Users and Group tab.
- b) Click the Groups subtab and click New.
- c) On the Create a New Group page, specify the details for the employee group as shown in the following screenshot and click OK:

**Create a New Group**

**Group Properties**

The following properties will be used to identify your new Group.

\* Indicates required fields

What would you like to name your new Group?

**\* Name:** employees

How would you like to describe the new Group?

**Description:** Group of Employees

Please choose a provider for the group.

**Provider:** DefaultAuthenticator

OK Cancel

- d) Similarly, create a new group called `managers`.

## Practice 18-1: Managing Users and Groups (continued)

- e) Then click the Users subtab and click New to create new users. The screen for creating the user Mary is shown here. Similarly, create other users as stated in the table at step 1.

**Create a New User**

**User Properties**

The following properties will be used to identify your new User.  
\* Indicates required fields

What would you like to name your new User?

\* **Name:**

How would you like to describe the new User?

**Description:**

Please choose a provider for the user.

**Provider:**

The password is associated with the login name for the new User.

**Password:**

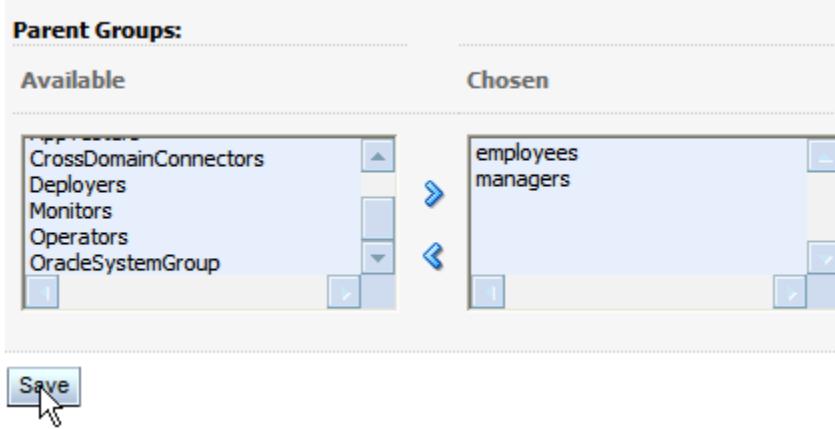
**Confirm Password:**

**OK** **Cancel**

- f) Click each username in the Users table, on the “Settings for myrealm” page, and click the Groups subtab.

### Practice 18-1: Managing Users and Groups (continued)

- g) Select the groups from the Available list and click to assign a group to the user and click Save. For example, group assignment for Mary is shown here:



- h) Similarly, assign groups to other users as per the table in step 1.

## Practice 18-2: Securing WebLogic Server Resources

In this practice, you deploy a Web application and secure it using policies defined in the deployment descriptor.

- 1) Deploy the `timeoff.war` Web application and configure security settings for the Web application by selecting the following option while deploying the application:

| Description   | Choices or Values                                                                                                     |
|---------------|-----------------------------------------------------------------------------------------------------------------------|
| Custom Roles  | Use roles that are defined in the Administration Console; use policies that are defined in the deployment descriptor. |
| New Role:     |                                                                                                                       |
| URL Pattern   | /managers/*                                                                                                           |
| Name          | Director                                                                                                              |
| Provider Name | XACMLRoleMapper                                                                                                       |

- If not already running, start the MedRecAdmSvr and MedRecSrv1 servers. To save on resources, you can stop other servers, if they are running.
- Using the Administration Console, deploy the `timeoff.war` Web application located in the `/home/oracle/wls-sysadm/labs/Lab18` folder. Target the application to MedRecSrv1.
- On the Optional Settings page, in the Security section, select “Custom Roles: Use roles that are defined in the Administration Console; use policies that are defined in the deployment descriptor.”

### Security

What security model do you want to use with this application?

**DD Only: Use only roles and policies that are defined in the deployment descriptors.**

**Custom Roles: Use roles that are defined in the Administration Console; use policies that are defined in the deployment descriptor.**

**Custom Roles and Policies: Use only roles and policies that are defined in the Administration Console.**

## Practice 18-2: Securing WebLogic Server Resources (continued)

- d) Under Source Accessibility, select “Copy this application onto every target for me.”

**Source accessibility**

How should the source files be made accessible?

Use the defaults defined by the deployment's targets

Recommended selection.

Copy this application onto every target for me

- e) Click Finish and activate your changes.
- f) Navigate to MedRecDomain > Deployments and click timeoff in the Deployments table.
- g) Navigate to Security > URL Patterns (subtab). Click New in the Standalone Web Application URL Patterns and Roles table.
- h) Specify the URL Pattern: /managers/\* Name: director, Provider Name: XACMLRoleMapper and click OK.

Create a New Stand-Alone Web Application URL Pattern Scoped Role

**OK** | **Cancel**

**Create a New Role URL Pattern**

The following property will be used to identify your new Role URL pattern.

What would you like to name your new Role URL pattern?

**URL Pattern:**

What would you like to name your new Role?

**Name:**

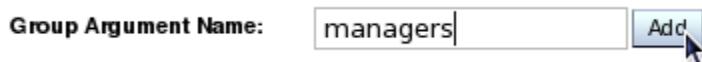
What Role Mapper would you like to select?

**Provider Name:**

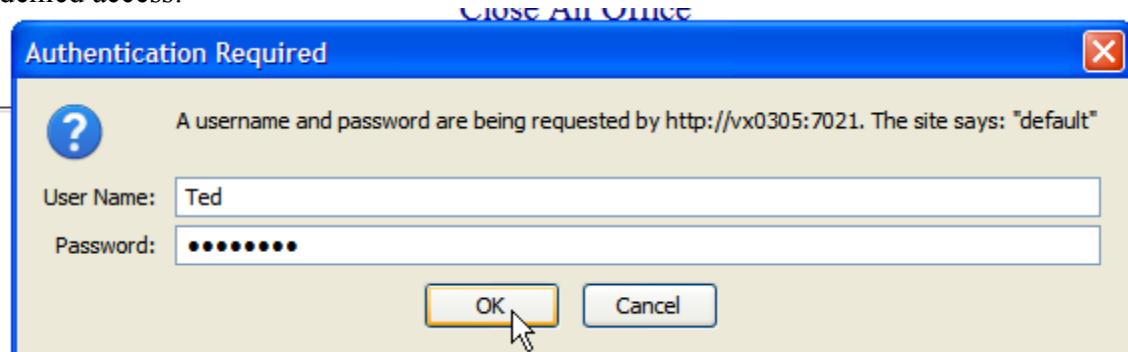
- i) In the Standalone Web Application URL Patterns and Roles table, you should now see the URL pattern created and assigned to the director role.
- j) Click director and click Add Conditions. Select Group from the Predicate list and click Next.

## Practice 18-2: Securing WebLogic Server Resources (continued)

- k) On the next screen, enter managers as Group Argument Name and click Add.



- l) Click Finish. On the next page, click Save.
- 2) Start the timeoff application and verify the policy you have configured.
- In the Administration Console, navigate to the Deployments page. In the Deployments table, select the timeoff application and then select Start > Servicing all requests. Click Yes when prompted.
  - Using another browser window or tab, navigate to the following URL:  
`http://wls-sysadm:7021/timeoff`
  - Try closing the office by clicking Close An Office. (You may need to enter the username and password. Use the values specified in step 1 of Section 18-1)
  - Log on as different users created in the previous lab. For example, Ted will be denied access:



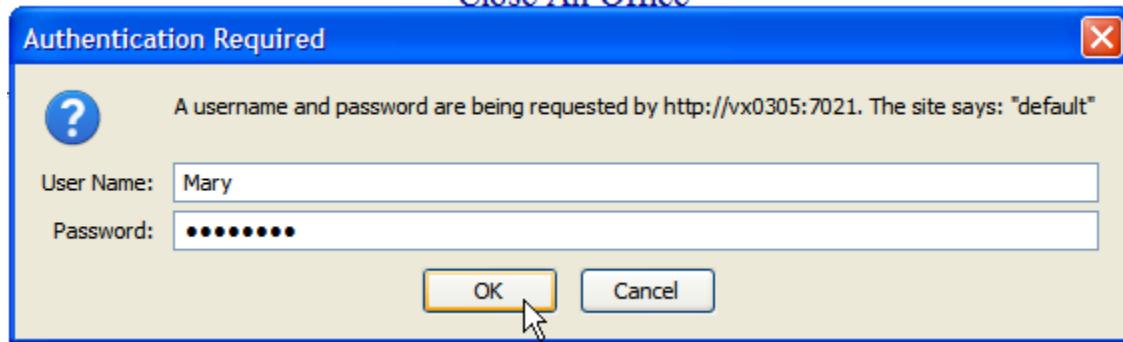
## Error 403--Forbidden

**From RFC 2068 *Hypertext Transfer Protocol -- HTTP/1.1*:**

### 10.4.4 403 Forbidden

## **Practice 18-2: Securing WebLogic Server Resources (continued)**

- e) Joe or Mary will be granted access.



## **Dizzyworld Office Closing Form**

A screenshot of a web-based form titled "Dizzyworld Office Closing Form". It has a grey header bar. Below it are two input fields: "Date" and "Reason", both with empty text boxes. At the bottom is a blue "Submit Form" button.

[Back To Home Page](#)

**Note:** Clear the previous cached authentication information before logging on as another user.

## Practices for Lesson 19

Many applications need the security of communicating over the Secure Socket Layer (SSL). This provides secure communications between the server and the client, or between two servers. Your company has decided to configure SSL for ensuring secure communications between a server and the client.

In this lab, you configure SSL and the keystores for the MedRecSv1 managed server in MedRecDomain.

In this practice, you perform the following tasks:

- Using keytool to generate an identity keystore that contains a private key and a self-signed public certificate
- Configuring keystores in the Administration Console
- Configuring SSL for a managed server

## Practice 19-1: Configuring Keystores

In this practice, you generate a key, self-signed certificate, and identity keystore.

- 1) Using the Java keytool utility, create a key and copy the key to your domain folder.

- a) In your gnome terminal session, ensure that JAVA\_HOME and the related environment variables have been set. (If they have not been set, run the setWLSEnv.sh script.)

```
$> env | grep JAVA
JAVA_USE_64BIT=
JAVA_OPTIONS= -Xverify:none
JAVA_VENDOR=Oracle
JAVA_HOME=/u01/app/oracle/product/fmw/11.1.0/jrockit_160_05_R2
7.6.2-20
JAVA_VM=-jrockit
```

- b) Navigate to the Lab19 subfolder under the /home/oracle/wls-sysadm/labs folder. Then run the keytool command as follows (all in one line). You can use the genkey.sh script in this folder for convenience.

```
$> cd /home/oracle/wls-sysadm/labs/Lab19
$> keytool -genkey -v -alias MRkey -keyalg RSA -keysize 512
 -dname "CN=wls-sysadm"
 -keypass MRkeypass -validity 365
 -keystore MR_identity.jks -storepass MRstorepass
```

- c) Copy the key file you generated to your domain folder.

```
$> cp MR_identity.jks
/u01/app/oracle/user_projects/domains/MedRecDomain/
```

- d) Generate a Certificate Signing Request (CSR) using the key you have created. (You can use certreq.sh instead of entering the keytool command.)

```
$> keytool -certreq -v -alias MRkey -file MR_cert_request.pem
 -keypass MRkeypass -storepass MRstorepass
 -keystore MR_identity.jks
```

- e) Copy the CSR you generated to your domain folder.

```
$> cp MR_cert_request.pem
/u01/app/oracle/user_projects/domains/MedRecDomain/
```

- f) In the Administration Console, navigate to MedRecDomain > Environment > Servers > MedRecSrv1 > Configuration > Keystores. In Change Center, click Lock & Edit.

- g) On the Keystores page, specify the following properties and click Save.

| Description                             | Choices or Values                       |
|-----------------------------------------|-----------------------------------------|
| Keystores                               | Custom Identity and Java Standard Trust |
| Custom Identity Keystore                | MR_identity.jks                         |
| Custom Identity Keystore Type           | JKS                                     |
| Custom Identity Keystore Passphrase     | MRstorepass                             |
| Java Standard Trust Keystore Passphrase | changeit                                |

## **Practice 19-1: Configuring Keystores (continued)**

- 2) Configure MedRec Svr1 with SSL. Verify accessing the timeoff application by using HTTPS.
- In the Administration Console, navigate to MedRecDomain > Environment > Servers > MedRecSvr1 > Configuration > SSL.
  - On the SSL page, specify the following properties and click Save.
    - Identity and Trust Locations: Keystores
    - Private Key Alias: MRkey
    - Private Key Passphrase: MRkeypass
  - Navigate to MRDomain > Environment > Servers > MedRecSvr1 > Configuration > General.
  - Select the check box next to SSL Listen Port Enabled and set the SSL Listen Port as 7022. Then click Save.
  - Click Activate Changes. Then stop the MedRecSvr1 server.
  - Start the MedRecSvr1 server using the desktop icon or the script.
  - In another browser window or tab, access the URL: <https://wls-sysadm:7022/timeoff>. You may receive an error or warning.

### **Secure Connection Failed**

wls-sysadm:7022 uses an invalid security certificate.

The certificate is not trusted because it is self signed.

(Error code: sec\_error\_ca\_cert\_invalid)

- This could be a problem with the server's configuration, or it could be someone trying to impersonate the server.
- If you have connected to this server successfully in the past, the error may be temporary, and you can try again later.

[Or you can add an exception...](#)

- h) Click the link to add an exception and click Add Exception

You should not add an exception if you are using an internet connection that you do not trust completely or if you are not used to seeing a warning for this server.

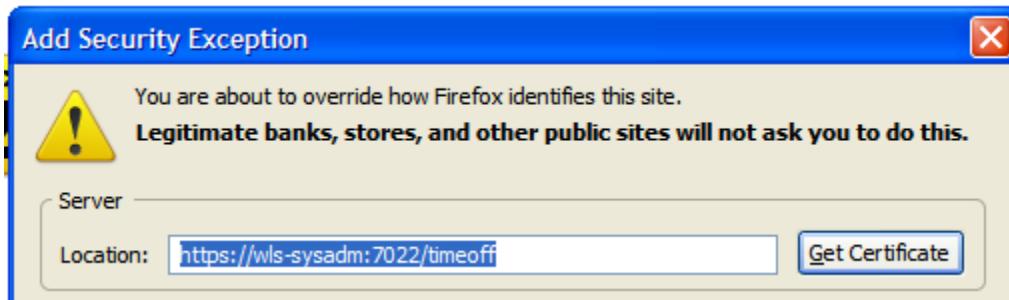
[Get me out of here!](#)

[Add Exception...](#)

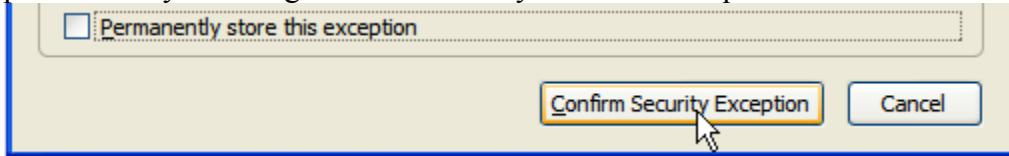


### Practice 19-1: Configuring Keystores (continued)

- i) Then click Get Certificate to add the server certificate to your browser.



- j) Click Confirm Security Exception. In this box, you can also make this exception permanent by selecting the “Permanently store this exception” check box.



- k) Now, you can access the application on MedRecSrv1.

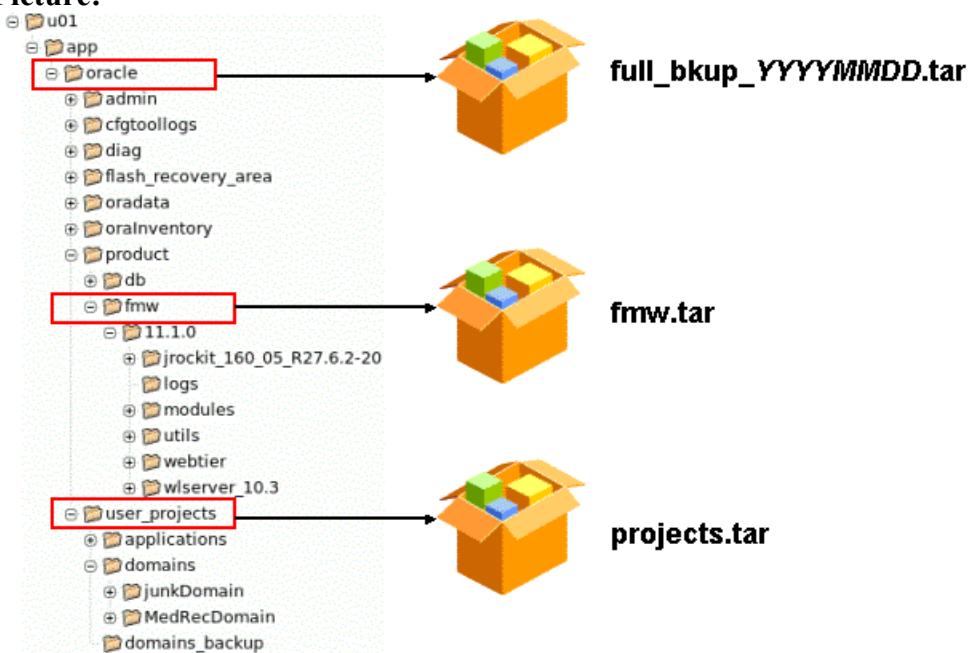
## Practices for Lesson 20

### Backup and Recovery

A full backup would obviously include the database. The procedures for doing hot (online, inconsistent) backups of the database use RMAN and are beyond the scope of this course. For this lab, you perform only backups of the Fusion Middleware components. Because of the nature of the lab environment, you are not going to do a full backup of Fusion Middleware, but only an incremental backup of the Middleware configuration components. Lastly, you enable the autobackup of the config.xml files in Change Center of the Administration Console. The key tasks are:

- Stopping everything
- Performing a full, cold (offline, consistent) backup
- Performing an incremental, cold backup
- Simulating a failure
- Stopping everything (all FMW recoveries are performed cold)
- Restoring the affected components
- Restarting everything

### Big Picture:



## Practice 20-1: Backing Up the Configuration

In this practice, you stop all middleware processes and take a cold backup of the `user_projects` directory. In real life, you would back up the `fmw` directory and the database as well. A backup of the `fmw` directory is the same process except that it takes five times longer and does not demonstrate anything that you will not see on the shorter directory. A backup of the database is longer still and involves a tool called RMAN. The DBA usually handles that process (in fact, it is usually automatic). So in this lab, you will not back up the database.

- 1) Stop all the servers using any method you have learned. For example, you could change directory to  
`/u01/app/oracle/user_projects/domains/MedRecDomain/bin/` and run `./stopManagedWebLogic.sh MedRecSvr1` to stop `MedRecSvr1`. Similarly, stop `MedRecSvr2` and `MedRecSvr3`. To stop the administration server, run `./stopWebLogic.sh`.
- 2) Switch to the `root` user. As `root`, go to the `root` (`/`) directory and run the `tar` command as follows:  
`tar -czpvf /projects.tar /u01/app/oracle/user_projects/*` where `c` is create, `z` is zipped, `p` is preserve permissions, `v` is verbose messages, and `f` is file name. The resulting file should be about 51 MB.
- 3) Test the TAR file by using the `-t` option of the `tar` command  
`tar -ztf /projects.tar`. You really want to test it now, so that there are no problems later.
- 4) Restart the administration server to make sure that it is functioning properly.

## Practice 20-2: Enabling Autobackup of config.xml

In this practice, you enable the Administration Console to save a copy of config.xml each time it is changed. The number of copies to save on a rolling basis is configurable. The default of one copy is too few; ten is a reasonable number of copies because the file is relatively small.

- 1) Observe the state of the directories and files *before* the configuration archive is enabled. In a terminal session, change to directory /u01/app/oracle/user\_projects/domains/MedRecDomain/ and note the *absence* of the following files and directories: config.original.jar, config.booted.jar, and configArchive.
- 2) In the Administration Console, click Lock & Edit in Change Center.
- 3) In Domain Structure, click MedRecDomain > Configuration > Advanced.
- 4) Select Configuration Archive Enabled. Change the Archive Configuration Count to 10 and click Save. Ten is not a fixed number; anything more than two will work; more than 50 is excessive; ten is a good middle ground balancing risk and convenience against space consumed.
- 5) In Change Center, click Activate Changes. Note that this requires a restart of all WebLogic components. Click “View changes and restarts.” Click the Restart Checklist tab. From here, you can stop all resources. Use the top-left box to select all resources and stop them. Click Stop. Click Yes to shut down the servers.
- 6) Restart the administration server. For the purpose of this lab, there is no need to start any other servers.
- 7) Make a change to a managed server. For example, adding Notes to the settings documenting what you are doing is a good idea. Click Save and Activate Changes.
- 8) Go back to the terminal session and note the *presence* of the following files and directories: config.original.jar, config.booted.jar, and configArchive. Note how many files are in the configArchive directory.
- 9) Go back to the Administration Console and make a few more changes (add more Notes to more settings), making sure that you click Activate Changes in between each one. Note now how many files are in the configArchive directory. Are any of the time stamps changing on config.original.jar or config.booted.jar?

### **Practice 20-3: Performing Recovery**

In this practice, you simulate a media failure and then recover from it.

- 1) While the administration server is running, delete the directory that contains the configurations. As root, enter  
`rm -rf /u01/app/oracle/user_projects/domains/*.`  
This deletes all configuration files.
- 2) In the terminal session associated with the administration server, the standard out error messages should show a series of errors that say <The file could not be found in the webapp directory or war.>. In a few minutes, the server should shut itself down. You will see the following messages:  
<JVM called WLS shutdown hook. The server will force shutdown now> <Server shutdown has been requested by <WLS Kernel>>  
To make sure that WebLogic Server shut down cleanly, enter:  
`ps -ef | grep java`  
`ps -ef | grep MedRec`  
This ensures that nothing is left as a zombie (a process that won't be killed properly).
- 3) As root, from the root directory, restore the configuration files. Even though you lost only part of /user\_projects, restore the whole TAR file because there may be interdependencies. Enter `tar -zxvpf /projects.tar`.
- 4) Restart the administration server to ensure that it is functioning properly.

---

## Glossary

---



## Glossary/Acronyms

|                 |                                                                                   |
|-----------------|-----------------------------------------------------------------------------------|
| <b>ACID</b>     | Atomicity, consistency, isolation, durability (for DB transactions)               |
| <b>ACK</b>      | acknowledgement-based                                                             |
| <b>AsyncRep</b> | Asynchronous HTTP Session Replication                                             |
| <b>API</b>      | Application programming interface                                                 |
| <b>BPEL</b>     | Business Process Execution Language                                               |
| <b>CA</b>       | Certificate Authority (issues SSL certificates)                                   |
| <b>CGI</b>      | Common Gateway Interface                                                          |
| <b>CLI</b>      | Command-line interface (as opposed to GUI)<br>Call-level interface (part of JDBC) |
| <b>CLV</b>      | Certificate Lookup and Validation                                                 |
| <b>CSR</b>      | Certificate Signing Request                                                       |
| <b>CSS</b>      | Common Security Services                                                          |
| <b>CMO</b>      | Current Management Object (part of WLST)                                          |
| <b>COM</b>      | Component Object Model (see also DCOM and jCOM)                                   |
| <b>CORBA</b>    | Common Object Request Broker Architecture                                         |
| <b>DBA</b>      | Database administrator                                                            |
| <b>DCOM</b>     | Distributed COM                                                                   |
| <b>DD</b>       | deployment descriptor                                                             |
| <b>DDL</b>      | Data Definition Language (part of SQL)                                            |
| <b>DMZ</b>      | Demilitarized Zone (network between firewalls)                                    |
| <b>DNS</b>      | domain name server                                                                |
| <b>DoS</b>      | denial of service                                                                 |
| <b>DSA</b>      | Digital Signature Algorithm                                                       |
| <b>DTD</b>      | document type definition                                                          |
| <b>EAR</b>      | Enterprise Archive                                                                |
| <b>EE</b>       | Enterprise Edition (as opposed to Standard Edition [SE])                          |
| <b>EJB</b>      | Enterprise JavaBeans                                                              |
| <b>FIFO</b>     | First in, first out (queuing)                                                     |
| <b>FMW</b>      | Fusion Middleware                                                                 |
| <b>GMD</b>      | guaranteed message delivery                                                       |
| <b>GUI</b>      | Graphical user interface (as opposed to CLI)                                      |
| <b>HTML</b>     | Hypertext Markup Language                                                         |
| <b>HTTP</b>     | Hypertext Transfer Protocol                                                       |
| <b>IOP</b>      | Internet Inter-ORB Protocol                                                       |
| <b>IIS</b>      | Internet Information Server                                                       |

|                  |                                                                                                             |
|------------------|-------------------------------------------------------------------------------------------------------------|
| <b>J2SDK</b>     | Java 2 SDK Standard Edition                                                                                 |
| <b>JAAS</b>      | Java Authentication and Authorization Service                                                               |
| <b>JACC</b>      | Java Authorization Contract for Containers                                                                  |
| <b>JAR</b>       | Java Archive                                                                                                |
| <b>Java EE</b>   | Java Platform Enterprise Edition                                                                            |
| <b>Java SE 6</b> | Java Platform Standard Edition 6                                                                            |
| <b>JAX-WS</b>    | Java API for XML-Based Web Services                                                                         |
| <b>JAZN</b>      | Java AuthoriZatioN                                                                                          |
| <b>JCA</b>       | Java EE Connector Architecture                                                                              |
| <b>JCE</b>       | Java Cryptography Extensions                                                                                |
| <b>jCOM</b>      | WLS Java-to-COM bridge                                                                                      |
| <b>JDBC</b>      | Java Database Connectivity                                                                                  |
| <b>JDK</b>       | Java Development Kit                                                                                        |
| <b>JKS</b>       | Java KeyStore                                                                                               |
| <b>JMS</b>       | Java Message Service                                                                                        |
| <b>JMX</b>       | Java Management Extensions                                                                                  |
| <b>JNDI</b>      | Java Naming and Directory Interface                                                                         |
| <b>JPA</b>       | Java Persistence API                                                                                        |
| <b>JPS</b>       | Java Platform Security                                                                                      |
| <b>JRF</b>       | Java Required Files (also known as Portability Layer [PL]; Oracle Web services stack for SOA and WebCenter) |
| <b>JRMP</b>      | Java Remote Method Protocol                                                                                 |
| <b>JSSE</b>      | Java Secure Socket Extensions                                                                               |
| <b>JSP</b>       | JavaServer Pages                                                                                            |
| <b>JSR</b>       | Java Specification Request                                                                                  |
| <b>JTA</b>       | Java Transaction API                                                                                        |
| <b>JVM</b>       | Java Virtual Machine                                                                                        |
| <b>JWS</b>       | Java Web Service                                                                                            |
| <b>LDAP</b>      | Lightweight Directory Access Protocol                                                                       |
| <b>LDIF</b>      | LDAP Data Interchange Format (readable text)                                                                |
| <b>LVC</b>       | Live Virtual Class                                                                                          |
| <b>MIME</b>      | Multipurpose Internet Mail Extensions                                                                       |
| <b>MSI</b>       | Managed Server Independence                                                                                 |
| <b>NES</b>       | Netscape Enterprise Server                                                                                  |
| <b>NIC</b>       | Network Interface Card (usually an Ethernet adapter)                                                        |
| <b>O/R</b>       | object-relational                                                                                           |
| <b>OAAM</b>      | Oracle Adaptive Access Manager                                                                              |

|              |                                                                                                                                                                         |
|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>OASIS</b> | Organization for the Advancement of Structured Information Standards                                                                                                    |
| <b>ODBC</b>  | Open Database Connectivity                                                                                                                                              |
| <b>OHS</b>   | Oracle HTTP Server                                                                                                                                                      |
| <b>OID</b>   | Oracle Internet Directory                                                                                                                                               |
| <b>OOTB</b>  | Out Of The Box (by default; without modification)                                                                                                                       |
| <b>OPMN</b>  | Oracle Process Manager and Notification Server                                                                                                                          |
| <b>OPSS</b>  | Oracle Platform Security Services                                                                                                                                       |
| <b>ORB</b>   | Object Request Broker                                                                                                                                                   |
| <b>OS</b>    | Operating system (examples: Windows, Linux)                                                                                                                             |
| <b>OTN</b>   | Oracle Technology Network                                                                                                                                               |
| <b>PAM</b>   | Pluggable Authentication Module                                                                                                                                         |
| <b>PKI</b>   | Public Key Infrastructure                                                                                                                                               |
| <b>PTP</b>   | Point To Point                                                                                                                                                          |
| <b>QoS</b>   | Quality of Service                                                                                                                                                      |
| <b>RAC</b>   | Real Application Clusters (for multihost databases)                                                                                                                     |
| <b>RAR</b>   | Resource Adapter Archive                                                                                                                                                |
| <b>RBAC</b>  | Role-Based Access Control (part of JPS)                                                                                                                                 |
| <b>RCU</b>   | Repository Creation Utility                                                                                                                                             |
| <b>RMAN</b>  | Recovery Manager (for the database)                                                                                                                                     |
| <b>RMI</b>   | Remote Method Invocation                                                                                                                                                |
| <b>RSH</b>   | Remote Shell (as opposed to SSH)                                                                                                                                        |
| <b>SAF</b>   | store-and-forward                                                                                                                                                       |
| <b>SAML</b>  | Security Assertion Markup Language                                                                                                                                      |
| <b>SDK</b>   | Software development kit (programming tools)                                                                                                                            |
| <b>SHA</b>   | Secure Hash Algorithm                                                                                                                                                   |
| <b>SLA</b>   | service-level agreements                                                                                                                                                |
| <b>SNMP</b>  | Simple Network Management Protocol                                                                                                                                      |
| <b>SOA</b>   | Service-Oriented Architecture                                                                                                                                           |
| <b>SOAP</b>  | Simple Object Access Protocol                                                                                                                                           |
| <b>SQL</b>   | Structured Query Language                                                                                                                                               |
| <b>SSH</b>   | Secure Shell (as opposed to RSH)                                                                                                                                        |
| <b>SSL</b>   | Secure Sockets Layer                                                                                                                                                    |
| <b>SSPI</b>  | Security Services Provider Interface                                                                                                                                    |
| <b>T3</b>    | An optimized, proprietary communications protocol used to transport data between WebLogic Server and other Java programs, including clients and other WebLogic Servers. |
| <b>T3S</b>   | T3 protocol using SSL                                                                                                                                                   |

|              |                                                                      |
|--------------|----------------------------------------------------------------------|
| <b>TAR</b>   | Tape Archive                                                         |
| <b>TCP</b>   | Transmission Control Protocol                                        |
| <b>TLS</b>   | Transport Layer Security                                             |
| <b>TTL</b>   | time-to-live                                                         |
| <b>URL</b>   | Uniform Resource Locator                                             |
| <b>VM</b>    | Virtual Machine                                                      |
| <b>WAN</b>   | Wide Area Network                                                    |
| <b>WAR</b>   | Web Archive                                                          |
| <b>WLDF</b>  | WebLogic Diagnostics Framework                                       |
| <b>WLS</b>   | WebLogic Server                                                      |
| <b>WLST</b>  | WebLogic Scripting Tool                                              |
| <b>WSIT</b>  | Web Services Interoperability Technologies                           |
| <b>XA</b>    | X/Open Distributed Transaction Processing (part of two-phase commit) |
| <b>XACML</b> | eXtensible Access Control Markup Language                            |
| <b>XML</b>   | Extensible Markup Language                                           |

---

# Index

---



**A**

Apache 1-10, 5-5, 5-15, 9-2, 9-10-11, 10-43, 15-15, 15-21, 15-25, 17-15, 20-5  
 API 2-7, 2-13-14, 2-18-19, 2-22, 6-14, 9-8, 9-11, 11-18, 12-30, 13-4, 13-7, 14-11, 15-5,  
 16-9  
 autodeploy 4-34, 10-2, 10-9, 10-11, 10-18, 10-29, 10-30, 10-48

**B**

Backup 18-31, 18-32, 20-1, 20-3, 20-7-8, 20-11, 20-13-14, 20-18, I-7  
 BEA 3-18, 3-19, 3-20-21, 3-23, 4-42-43, 5-9, 5-14, 7-8, 9-13-14, 9-17

**C**

CA 18-9, 18-10, 19-5, 19-8, 19-10, 19-12, 19-14, 19-18  
 cache 2-25, 4-27, 4-46, 8-16, 8-18, 11-21, 11-22, 11-23, 11-31, 11-32,  
 17-20, 17-40, 18-32, 19-32, 20-5  
 CCI 2-22  
 cluster 2-26, 4-5-9, 4-16-19, 4-31, 6-11, 6-16, 6-33, 6-35, 7-8, 7-14-15, 7-18, 7-31,  
 8-7-8, 8-41, 9-17, 9-23-26, 10-4, 10-10, 10-13, 10-43, 12-4, 13-22, 13-24, 14-18,  
 14-20-21, 14-23-24, 14-29, 14-38, 15-3-4, 15-2-9, 15-11, 15-13-24, 15-26-39, 16-1-  
 22, 16-24-27, 17-2-16, 17-18-19, 17-24, 17-26-29, 17-31-34, 17-37-40, 17-45-46,  
 19-32, 20-4, 20-7, 20-11, 20-25, I-4-5  
 clusters 4-11, 4-14, 6-5, 8-5, 8-7-8, 9-23, 12-15, 13-12  
 Coherence 3-5, 14-36  
 Commons 9-5, 9-8, 9-11  
 context root 10-19, 11-9  
 CORBA 2-18

**D**

DMZ 10-35, 15-16, 15-20, 15-22  
 DNS 2-14, 4-9, 4-17, 7-4, 7-19, 8-10, 8-13, 11-13, 13-18, 15-20, 16-4, 16-6-7, 16-12,  
 17-33  
 DTDs 11-32

**E**

EAR 4-34, 11-7, 11-27, 11-32, 11-33-35, 12-26-27, 12-36, 14-19-21, 18-26  
 Eclipse 2-9, 12-8, 12-22  
 EIS 2-22  
 EJB 2-12, 2-14, 2-23, 2-30, 3-5, 4-17, 4-34, 11-3, 11-10, 11-17-31, 11-33-35, 11-38,  
 12-4, 12-27, 12-30, 15-3-6, 15-8-11, 15-17, 15-19, 15-34, 15-38, 16-7, 17-2-3,  
 17-8, 17-26, 17-31, 17-32-37, 17-39-40, 18-14-15, 18-23, 18-26-27, 19-7, I-3  
 extend 1-8-9, 4-10, 4-12, 4-21, 5-4-5, 5-7, 5-12, 5-17, 6-23, 6-26, 20-22

**F**

FMW 1-10, 4-11, 7-24

**G**

GUI 3-2-4, 3-6, 3-13-14, 4-5, 5-6, 7-7, 13-36, I-5

**H**

heartbeat 15-29, 15-31, 16-20, 20-23

HTML 1-8, 2-8, 2-9, 2-10, 2-11, 2-24, 3-23, 3-24, 6-9, 10-35, 10-38, 10-40, 10-41, 11-4, 11-5, 11-26, 15-16, 18-34, 18-35

HTTP 1-10, 1-12, 2-7, 2-8, 2-23, 2-24, 2-26, 2-30, 3-22, 4-11, 4-15, 4-18, 4-19, 4-45, 5-15, 6-6, 6-8, 7-14, 8-7, 9-4, 9-6, 9-10, 10-34-40, 10-42, 11-4, 11-11, 11-13-14, 11-31, 11-41, 12-25, 12-27, 12-30, 15-3-4, 15-6-8, 15-15-16, 15-18-19, 15-21-27, 15-34, 16-2, 16-4, 16-17, 16-22, 16-23-24, 16-27, 17-2-3, 17-8-13, 17-15-16, 17-18-19, 17-23, 17-26-31, 17-44, 18-27, 19-6, 19-24, 19-27-28, 19-30, 20-4-6, 20-12-13, 20-16, 20-32-33, I-4

**I**

IIOP 2-7, 2-23, 19-24

**J**

JAAS 2-20, 18-6

Jakarta 9-11

JAR 2-11, 3-15, 3-16, 3-17, 3-22, 4-33-34, 4-38-39, 5-4, 5-8, 5-12, 6-16, 6-26, 6-32, 7-25, 10-6, 10-31, 10-47, 11-6, 11-8, 11-27, 11-33-34, 18-26, 20-15, I-2

JCA 2-22, 4-38, 12-26, 12-30

JCP 2-6

JDBC 1-8, 2-7, 2-13-14, 2-17, 2-21-22, 3-16-17, 4-11-12, 4-14, 4-20-25, 5-5, 5-13, 7-21, 9-4, 9-6, 10-5, 10-22, 11-26, 11-28-29, 11-32, 12-5, 12-26, 12-30, 13-1-7, 13-9-14, 13-16-20, 13-22, 13-24, 13-26-27, 13-29-30, 13-32, 13-35-36, 14-7, 14-9, 14-16, 14-36-37, 14-39-40, 15-4, 15-6, 15-15-16, 17-2, 17-9, 17-16, 17-18-23, 17-44-45, 18-30, 20-12, I-7

JDeveloper 0-6, 2-9, 5-2, 5-11, 5-12, 5-21, 12-8, 12-22, 18-5

JMS 2-7, 2-14, 2-19, 3-5, 4-11-12, 4-14, 4-21, 4-26-27, 4-31, 5-5, 5-13, 7-21, 9-4, 9-6, 9-19, 10-5, 10-22, 11-26, 11-28-29, 11-32, 12-26, 12-30, 12-39, 14-1-4, 14-7-33, 14-35-42, 14-44-53, 15-4, 15-6, 15-15, 15-33, 18-26, 18-30, 19-26, 20-4-5, I-3, I-7

JMX 2-21, 4-7, 6-3, 6-14, 6-17, 6-18, 6-26, 6-35, 6-36, 6-39, 6-44, 8-14, 10-31, 16-9

**J**

JNDI 2-7, 2-14, 2-15, 2-16, 2-17, 4-22, 11-21, 11-23, 12-4, 12-30, 13-2, 13-5, 13-12, 13-15-16, 13-23-24, 13-34, 14-9-10, 14-22-24, 14-26, 14-30, 14-42, 15-29, 15-31, 15-35-36, 16-20, 18-23, 19-7  
 JPA 2-13, 11-18, 11-20  
 JRMP 2-23  
 JSTL 2-10  
 JTA 2-18, 14-11, 14-23, 15-5  
 JTS 2-18, 9-4, 12-30  
 JWS 11-14  
 Jython 1-9, 5-13, 6-22-25, 6-30, 10-26

**L**

LDAP 2-14, 4-29, 4-51, 6-16, 11-15, 13-34, 18-5, 18-12, 18-26, 18-31-33, 20-11-12  
 log4j 4-38, 4-39, 9-2, 9-8, 9-11

**M**

MBean 1-9, 6-27, 6-35-36, 6-39, 7-5-6, 7-14, 8-9, 9-7, 10-32, 13-24, 16-19, 17-30  
 MIME 2-8, 10-36, 10-38, 11-10-11, A-10  
 MSI 7-2, 7-3, 7-23, 7-26-29, 7-35-36, 8-5, 8-18, 20-23, 20-31  
 multicast 4-16, 4-17, 9-25-26, 15-29-33, 15-35-36, 16-4, 16-6, 16-10-11, 16-13, 16-20, 16-25

**N**

NIC 9-26, 9-27

**O**

ODBC 13-4  
 OHS 1-10, 1-12, 2-26, 5-15, 10-42, 10-43-44, 15-25-26, 16-22-24, 20-6, 20-13, 20-16  
 OID 1-11, 1-12  
 OPMN 1-9, 1-12, 4-45, 10-44, 16-23, 20-5, 20-13, 20-26  
 opmnctl 10-44, 16-23, 16-24, 20-16, 20-21, 20-26  
 OPSS 18-5

**P**

PAM 2-20  
 plug-in 2-26, 10-43, 15-3-4, 15-7, 15-21, 15-23-26, 17-9-10, 17-12-13, 17-15  
 Plug-In 15-23  
 proxy 1-10, 2-25-27, 3-16-17, 4-11, 4-18-19, 6-6, 10-43, 15-3-4, 15-7, 15-13, 15-16, 15-21-25, 15-27, 16-4-5, 16-17, 16-22, 17-9-10, 17-12-13, 17-15-16, 17-18, 19-6

**R**

RAC 13-9, 13-18, 20-11  
 RAR 2-22, 4-34, 11-25, 11-27, 12-30  
 RCU 5-15, 5-16, 5-17, 20-4, 20-5  
 Recovery 14-7, 20-1, 20-3, 20-5, 20-7, 20-8, 20-18, 20-20-23, 20-25, I-7  
 reverse proxy 1-10, 2-25  
 RMAN 0-6, 20-5, 20-10, 20-11, 20-19  
 RMI 2-7, 2-14, 2-23, 4-7, 12-25, 12-30, 15-4, 15-6, 15-9-10, 15-34, 15-36, 17-40

**S**

SAML 4-30  
 SCSI 20-7  
 setDomainEnv 3-20, 3-21, 4-39, 4-42, 4-43, 5-10, 6-29, 7-8-9, 7-33  
 SOA 1-3, 1-4, 1-5, 1-7, 1-11, 1-12, 5-2, 5-11, 5-12, 5-15-17, 5-21, 18-5, 20-4-6, 20-11  
 SOAP 1-3, 2-23, 2-24, 11-14  
 SSL 4-9, 4-11, 4-15, 6-6, 6-27, 6-29, 6-34, 6-36, 7-4, 7-7, 7-15, 7-19, 8-13, 8-21, 8-23,  
 8-29, 8-34, 9-26, 10-23, 10-40, 17-15, 18-9-10, 18-34, 19-2-8, 19-10, 19-14-15,  
 19-18-21, 19-24, 19-34, 19-37, 19-38

**T**

tar 10-5, 20-10, 20-13, 20-14, 20-16, 20-19, 20-21, 20-22  
 template 0-9, 1-12, 3-16, 4-10-12, 4-20-21, 4-24-25, 4-32, 4-42, 4-44, 4-53, 5-2-8,  
 5-11-14, 5-19-22, 6-26, 6-28, 6-32-33, 7-24-25, 12-6, 12-8, 12-11, 14-26

**U**

unicast 4-16, 15-29, 15-30, 15-33, 15-35, 15-36, 16-10, 16-20, 16-25,  
 17-41  
 URL 4-15, 4-22, 4-24, 7-4, 7-11, 10-2, 10-41, 11-11, 11-12, 11-13,  
 11-16, 12-34, 13-5, 13-13, 13-20, 13-25, 13-26, 15-24, 16-7, 17-12, 18-15,  
 18-29, 19-6, 19-7

**W**

WAR 2-11, 4-34, 11-4, 11-5, 11-7, 11-8, 11-27, 11-34, 11-35, 12-27,  
 12-30  
 Web Cache 1-10, 1-12, 4-45, 5-15, 15-25, 20-4-6, 20-13, 20-16, 20-17, 20-26, 20-32  
 WLDF 6-8, 7-20, 7-21, 9-25, 10-22

**W**

WLS 1-5, 2-6, 2-18, 2-23, 3-7, 3-9-10, 3-23, 4-2, 4-44, 5-15, 6-12, 6-14, 6-23, 6-26, 6-31, 6-33, 7-21, 7-24, 7-31-32, 8-4, 8-10, 8-13, 8-21, 9-1-2, 9-11, 9-13-14, 9-16, 9-26, 9-29-30, 10-8, 10-32, 10-36, 10-43, 11-6, 11-18, 11-20, 11-28, 11-32, 12-18, 12-27, 12-33, 12-36, 13-9, 13-14, 13-25, 14-9, 14-10, 14-13, 14-15-16, 14-22, 15-24, 15-26-27, 17-14, 17-15, 17-26-27, 17-30, 17-34, 17-37, 17-39, 18-2-3, 18-5-8, 18-11, 18-13, 18-16-17, 18-19, 18-23, 18-28, 18-31, 18-36, 18-38, 18-40, 18-41, 19-2, 19-6-7, 19-13, 19-15-17, 19-19, 19-23-24, 19-37, I-7

WLST 1-9, 3-16, 4-2, 4-10, 4-38, 4-40, 4-41, 5-4-7, 5-13, 5-20, 6-1, 6-3, 6-14, 6-18, 6-22-23, 6-25-32, 6-34, 6-36-37, 6-39, 6-43-45, 7-2, 7-4-7, 7-14-15, 7-34, 8-2, 8-5, 8-9, 8-14-15, 8-39, 8-40, 8-44, 9-5, 9-13, 10-2, 10-8-9, 10-25-27, 10-32, 10-48, 12-6, 12-39, 13-6, 13-16, 13-24, 13-36, 14-19, 16-8, 16-9, 16-18, 18-36, 20-25, I-6

**X**

XA 4-23, 13-9, 13-16, 13-17, 14-22, 14-23

XML 1-3, 1-8, 2-24, 3-4, 3-16, 3-17, 4-8, 5-14, 6-16-18, 6-20-21, 6-32, 6-35, 9-5, 9-11, 11-4, 11-6, 11-7, 11-18, 11-21, 11-29, 11-32, 12-4, 12-37, 13-6, 13-14, 14-9, 14-20-21, 20-12, I-3

**Z**

zip 10-5, 11-25

