

*If you want Job...
Download more materials.
----- VISIT you should join in DURGA*

<http://ameerpetmaterials.blogspot.in/>
<http://ameerpetmatbooks.blogspot.in/>
<http://satyajohnny.blogspot.in/>

Web logic

Server Administration



MASTER IN JAVA TRAINING



Software Solutions®

Weblogic Training

Introduction to Weblogic Application server

Brief History of Weblogic

- This session will explain about History of Weblogic.
- Weblogic Application server Versions.
- Weblogic Server Capabilities.
- Supported Open Standards.
- Comparison of Application Servers.

History

- Prior to co-founding WebLogic, Inc., in September 1995, Paul Ambrose and Carl Resnikoff had developed (pre-JDBC) Oracle, Sybase, and Microsoft SQL Server database-drivers for Java under the name dbKona, as well as a "three tier" server to permit applets to connect to these databases.
- This WebLogic 1.48 server had the name T3Server (a bastardization of "3-Tier Server"). Concurrently, Pitman and Pasker had worked on network-management tools written in Java. Pasker had written an SNMP stack in Java and a W32 native method for ICMP ping, while Pitman worked on applets to display the management data.
- The 1.48 server version had (among other hidden features) the ability to extend it by modifying a dispatcher and adding a handler for different types of messages. Pasker talked Ambrose into sending him the source code for the server, and Pasker extended it so that applets could make SNMP and PING requests on the network, and display the results.
- At this point, the founders worked together to pursue what eventually became the "Application Server".
- BEA Systems acquired WebLogic, Inc. in 1998, following which it became BEA WebLogic.
- Oracle acquired BEA in 2008, following which it became Oracle WebLogic.
- Oracle WebLogic Server Owned by Oracle Corporation, Oracle WebLogic consists of a JavaEE platform product family that includes:
 - a JavaEE application server → WebLogic Application Server
 - an enterprise portal → WebLogic Portal
 - an Enterprise Application Integration platform
 - a transaction server and infrastructure → WebLogic Tuxedo
 - a telecommunication platform → WebLogic Communication Platform
 - an HTTP web server

Application Server Versions

- WebLogic Server 11g (10.3.2) - Nov 2009
- WebLogic Server 11g (10.3.1) - Jul 2009
- WebLogic Server 10.3 - Aug 2008
- WebLogic Server 10.0 - Mar 2008
- WebLogic Server 9.2

- WebLogic Server 9.1
- WebLogic Server 9.0 - Nov 2006
- WebLogic Server 8.1 - Jul 2003
- WebLogic Server 7.0 - Jun 2002
- WebLogic Server 6.1
- WebLogic Server 6.0 - March 2001
- WebLogic Tengah 3.1 - Jun 1998
- WebLogic Tengah 3.0.1 - Mar 1998
- WebLogic Tengah 3.0 - Jan 1998
- WebLogic Tengah - Nov 1997

Capabilities

- Oracle WebLogic Server forms part of Oracle Fusion Middleware portfolio and supports Oracle, DB2, Microsoft SQL Server, MySQL Enterprise and other JDBC-compliant databases.
Oracle WebLogic Platform also includes:
 - Portal → which includes Commerce Server and Personalization Server
 - WebLogic Integration
 - WebLogic Workshop → an Eclipse IDE for Java, SOA and Rich Internet applications
 - JRockit → a custom JVM.
- WebLogic Server includes .NET interoperability and supports the following native integration capabilities:
 - Native enterprise-grade JMS messaging
 - Java EE Connector Architecture
 - WebLogic/Tuxedo Connector
 - COM+ Connectivity
 - CORBA connectivity
 - IBM WebSphere MQ connectivity
- Oracle WebLogic Server Process Edition also includes Business Process Management and Data Mapping functionality. WebLogic supports security policies managed by security administrators. The Oracle WebLogic Server Security Model includes:
 - Application business logic separated from security code
 - Complete scope of security coverage for all JavaEE and non-JavaEE components

Supported Open Standards

- JavaEE 1.3 & 1.4 & 5
- JPA 1.0
- JAAS
- XSLT and XQuery
- ebXML
- BPEL & BPEL-J
- JMX and SNMP
- **Native support for**
 - SOAP
 - WSDL

- UDDI
- WS-Security
- WSRP

Comparison of Application Servers

- Proprietary application servers provide system services in a well defined but proprietary manner. The application developers develop programs according to the specification of the application server. Dependence on a particular vendor is the drawback of this approach.
- An opposite but analogous case is the Java EE platform discussed below.
- Java EE application servers provide system services in a well defined, open, industry standard. The application developers develop programs according to the Java EE specification and not according to the application server.
- A Java EE application developed according to Java EE standard can be deployed in any Java EE application server making it vendor independent.
- This article compares the features and functionality of application servers, grouped by the hosting environment that is offered by that particular application server.
- **Apache:** Apache-based application server bundles integrate script languages (such as Perl, Python, PHP, Ruby, Tcl) into Apache web server.
- **BASIC Language**
- **Run BASIC** - An all-in-one BASIC scriptable appserver with automatic session and state management
- **C++ Language**
- **Barracuda Web Server** SDK Embeddable Application Server with C and Lua support.
- **Tuxedo** Based on the ATMI standard, is one of the original application servers.
- **G-WAN:** G-WAN low-resource server with ANSI C servlets (edit & run) faster than Microsoft IIS 7.0 ASP.Net C# and available on Windows and Linux
- **Tntnet** is a web application server for C++. Tntnet comes with a template engine, which allows the programmer to embed C++-code into HTML pages. Since these templates are not compiled at runtime, but like classic C++-programs in a separate compile-cycle, the resulting programs are native and therefore very fast. Tntnet is multithreaded and supports object lifetime through scoped variables.
- **Common-Lisp Language:** Core Server Common-Lisp Application Server for Linux.
- **Java EE**

Product	Vendor	Edition	Release Date	JavaEE Compatibility	Servlet Spec	JSP Spec	License
Tomcat	ASF	6.0.20	03 Jun 2009	with OpenEJB or EclipseLink	2.5	2.1	Apache License
WebLogic Server	Oracle Corporation	10.3	10 Apr 2007	5	2.5	2.1	Proprietary
WebObjects	Apple Inc.	5.3.2	07 Aug 2006	partial1			Proprietary
JBoss	Red Hat	5.1.0	23 May 2009	5	2.5	2.1	LGPL

<u>WebSphere Application Server</u>	IBM	7.0.0.0	26 Sep 2008	5	2.5	2.1	Proprietary
<u>WebSphere AS Community Edition</u>	IBM	2.1.1.3	20 August 2009	5	2.5	2.1	Free to use, embed and distribute IBM ILANWP

What is Web Server?

- Web Server Overview
- The primary function of a web server is to deliver web pages (HTML documents) and associated content (e.g. images, style sheets, JavaScript's) to clients. A client, commonly a web browser or web crawler, makes a request for a specific resource using HTTP and, if all goes well, the server responds with the content of that resource. The resource is typically a real file on the server's secondary memory, but this is not necessarily the case and depends on how the web server is implemented.
- While the primary function is to serve content, a full implementation of HTTP also includes a way of receiving content from clients. This feature is used for submitting web forms, including uploading of files.
- Many generic web servers also support server-side scripting (e.g. Apache HTTP Server and PHP). This means that the behavior of the web server can be scripted in separate files, while the actual server software remains unchanged. Usually, this functionality is used to create HTML documents on-the-fly as opposed to return fixed documents. This is referred to as dynamic and static content respectively.
- History of web servers
- In 1989 Tim Berners-Lee proposed to his employer CERN (European Organization for Nuclear Research) a new project, which had the goal of easing the exchange of information between scientists by using a hypertext system. As a result of the implementation of this project, in 1990 Berners-Lee wrote two programs:
 - a browser called Worldwide Web;
 - the world's first web server later known as CERN httpd, which ran on NeXTSTEP.
- Between 1991 and 1994 the simplicity and effectiveness of early technologies used to surf and exchange data through the World Wide Web helped to port them to many different operating systems and spread their use among lots of different social groups of people, first in scientific organizations, then in universities and finally in industry.
- In 1994 Tim Berners-Lee decided to constitute the World Wide Web Consortium to regulate the further development of the many technologies involved (HTTP, HTML, etc.) through a standardization process.
- Common features
 - Virtual hosting to serve many web sites using one IP address.
 - Large file support to be able to serve files whose size is greater than 2 GB on 32 bit OS.
 - Bandwidth throttling to limit the speed of responses in order to not saturate the network and to be able to serve more clients.

What is Application Server?

- An application server is a software framework dedicated to the efficient execution of procedures (scripts, routines, programs ...) for supporting the construction of applications. The term was created in the context of web applications. In these, the application server acts as a set of components accessible to the software developer through an API defined by the platform itself. These components are usually

performed in the same machine where the web server is running, and their main job is to support the construction of dynamic pages.

- Other uses of the term can refer to:

- the services that a server makes available
 - the computer hardware on which the services run

- **Java application servers**

- Following the success of the Java platform, the term application server sometimes refers to a J2EE or Java EE 5 application server. Some of the better-known Java Enterprise Edition application servers include:

- Apache Tomcat (Apache Software Foundation)
 - Tcat Server (MuleSoft)
 - WebSphere Application Server and WebSphere Application Server Community Edition (IBM)
 - Sybase Enterprise Application Server (Sybase Inc)
 - WebLogic Server (Oracle)
 - JBoss (Red Hat)
 - Apache Geronimo (Apache Software Foundation)
 - Oracle OC4J (Oracle)
 - SAP Netweaver AS (ABAP/Java) (SAP)
 - WebObjects (Apple Inc.)

- The web modules include servlets and JavaServer Pages. Business logic resides in Enterprise JavaBeans (EJB-3 and later). The Hibernate project offers an EJB-3 container implementation for the JBoss application server. Tomcat from Apache and JOnAS from ObjectWeb exemplify typical containers which can store these modules.
- A Java Server Page (JSP) (a servlet from Java) executes in a web container — the Java equivalent of CGI scripts. JSPs provide a way to create HTML pages by embedding references to the server logic within the page. HTML coders and Java programmers can work side by side by referencing each other's code from within their own. JavaBeans are the independent class components of the Java architecture from Sun Microsystems.

Content management system

- A Content Management System (CMS) is a collection of procedures used to manage work flow in a collaborative environment. These procedures can be manual or computer-based.
- The procedures are designed to:
 - Allow for large number of people to contribute to and share stored data
 - Control access to data, based on user roles. User roles are used to define each use as to what information they can view or edit
 - Aid in easy storage and retrieval of data
 - Reduce repetitive duplicate input
 - Improve the ease of report writing
 - Improve communication between users
- In a CMS, data can be defined as almost anything - documents, movies, pictures, phone numbers, scientific data, etc. CMSs are frequently used for storing, controlling, revising, and publishing documentation. Content that is controlled is industry-specific. (Entertainment content differs from the design of a fighter jet). There are various terms for systems (related processes) that do this. Examples

include: Web Content Management, Digital Asset Management, Digital Records Management, Electronic Content Management (and others). Synchronization of intermediate steps, and collation into a final product are common goals of each.

What is the difference between Web Server and Application server?

- (Webserver serves pages for viewing in web browser, application server provides exposes business logic for client applications through various protocols)
- Webserver exclusively handles http requests. Application server serves business logic to application programs through any number of protocols.
- Webserver delegation model is fairly simple, when the request comes into the webserver, it simply passes the request to the program best able to handle it (Server side program). It may not support transactions and database connection pooling.
- Application server is more capable of dynamic behavior than webserver. We can also configure application server to work as a webserver. Simply application server is a superset of webserver.
- Web Server serves static HTML pages or gifs, jpegs, etc., and can also run code written in CGI, JSP etc. A Web server handles the HTTP protocol. Eg of some web server are IIS or apache.
- An Application Server is used to run business logic or dynamically generated presentation code. It can either be .NET based or J2EE based (BEA WebLogic Server, IBM WebSphere, JBoss).
- A J2EE application server runs servlets and JSPs (in fact a part of the app server called web container is responsible for running servlets and JSPs) that are used to create HTML pages dynamically. In addition, J2EE application server can run EJBs - which are used to execute business logic.
- An Application server has a 'built-in' web server; in addition to that it supports other modules or features like e-business integration, independent management and security module, portlets etc.

Summary

- ↳ Covered overview of web and Applications servers.
- ↳ History Of Weblogic Application Server.
- ↳ Differences between Web and Application Server
- ↳ Content Management

Installing Oracle WebLogic Server Environment

Objectives

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

- Explain the motivation behind distributed systems
- List the major components of the J2EE specification
- Describe the terminology used throughout the course
- Describe the Oracle WebLogic Server architecture
- Install and run Oracle WebLogic Server

Road Map

- Distributed Architecture
 - J2EE Technologies
 - Oracle WebLogic Server Terminologies

- Setting Up an Oracle WebLogic Server Environment

Distributed Systems

- Distributed systems divide the work amongst several independent modules.
- The failure of a single module has less impact on the overall system, which makes the system more:
 - Available
 - Scalable
 - Maintainable

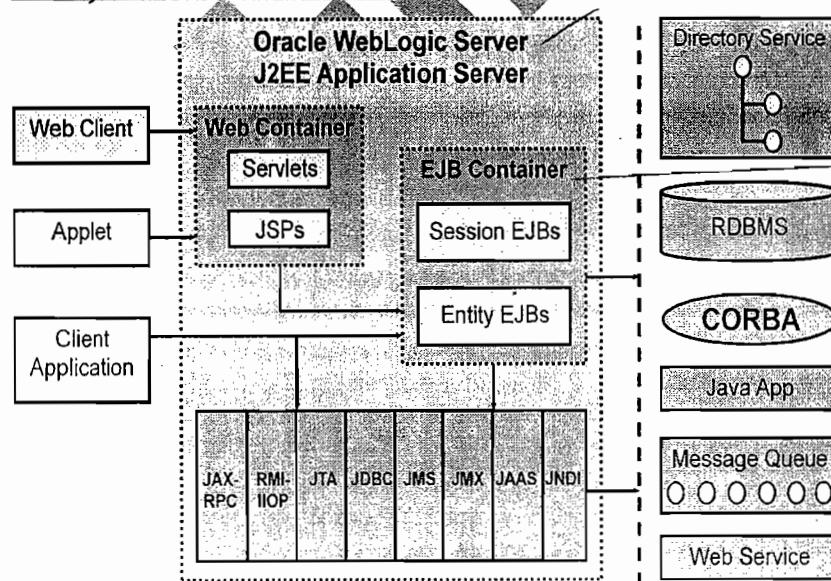
How Standards Help

- Many of the advantages of distributed systems come from standards.
- Standards:
 - Allow modularization of complex hardware and software
 - Allow a larger portion of the project costs to go toward solving business software needs

The J2EE Standard

- Java Platform 2 Enterprise Edition (J2EE) helps you to overcome distribution liabilities.
- Applications deployed with J2EE technologies are:
 - Standardized
 - Adherent to specification guidelines
 - Written in Java
 - Deployable in any compliant application server

The J2EE Architecture



Difference between web container & EJB container.

Java Servlets

- A servlet is a Java program that executes on the server, accepting client requests and generating dynamic responses.
- The most prevalent type of servlet is an HttpServlet that accepts HTTP requests and generates HTTP responses.
- Servlets:
 - Do not just generate HTML
 - Can also be used to generate other MIME types, such as images

JavaServer Pages (JSPs)

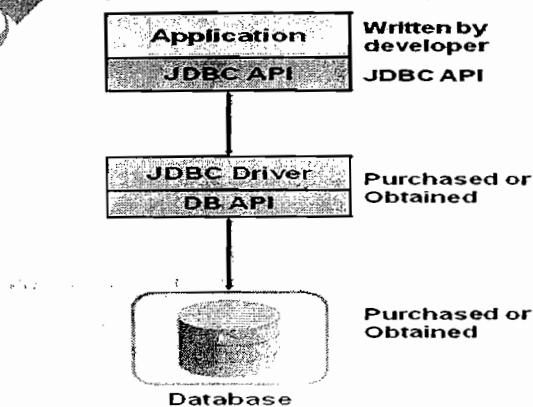
- Are HTML documents that are interwoven with Java
- Provide a dynamic response that is based on the client's request
- Provide for the separation of responsibilities between Web presentation and dynamic content
- Are portable (write once, run anywhere)
- Compile and run as servlets

Enterprise JavaBeans (EJBs)

- Are distributed components written in the Java programming language
- Provide distributable and deployable business services (logic) to clients
- Have well-defined interfaces
- Are reusable across application servers
- Execute within a container that provides management and control services
 - Oracle WebLogic Server 10.3 supports the EJB 3.0 specification

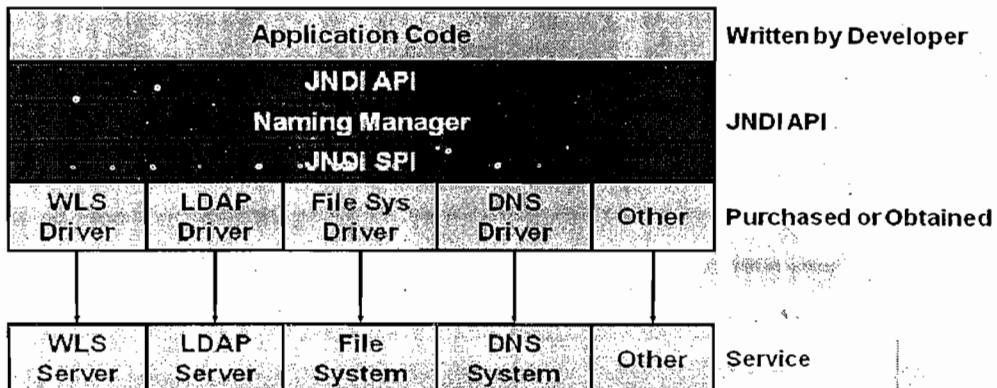
Java Database Connectivity (JDBC)

- The standard Java interface for accessing heterogeneous databases
- The specification (that defines four driver types for connecting to databases)



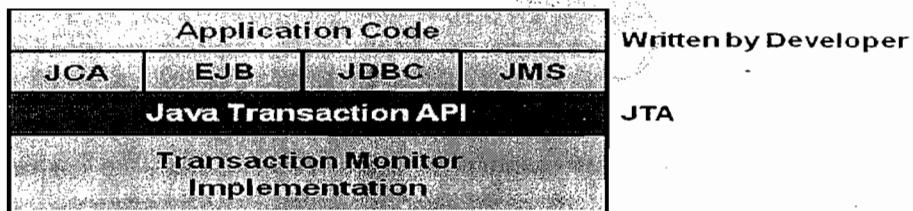
Java Naming and Directory Interface (JNDI)

- Java API for accessing naming and directory servers
- Built as a layer over DNS, LDAP, and so forth



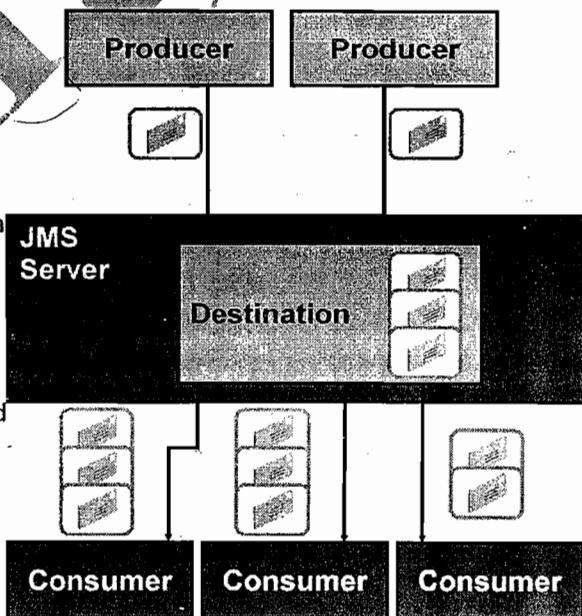
Java Transaction API (JTA)

- JTA is a standard Java API for demarcating transactions within a program.
- Oracle WebLogic Server supports local and distributed transactions.



Java Message Service (JMS)

- JMS is a Java API for accessing message-oriented middleware.
- The interface supports:
 - The point-to-point domain
 - The publish/subscribe domain
 - Guaranteed message delivery
 - Transactional participation
 - Dynamically configurable services
 - Application- or system-scoped resources
 - Interoperability with other messaging systems

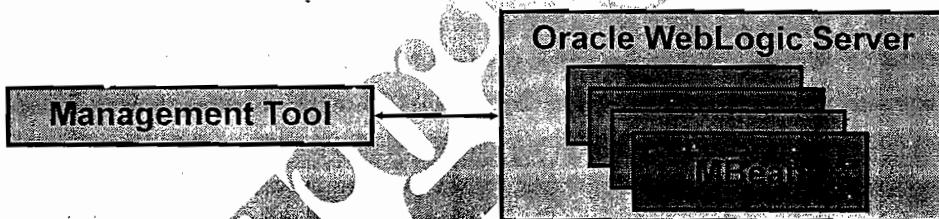


Java Authentication and Authorization

- Java Authentication and Authorization Service (JAAS) is a Java-based security management framework.
- JAAS supports:
 - Single sign-on
 - A Pluggable Authentication Module (PAM)
- JAAS enables flexible control over authorization whether it is based on:
 - Users
 - Groups
 - Roles

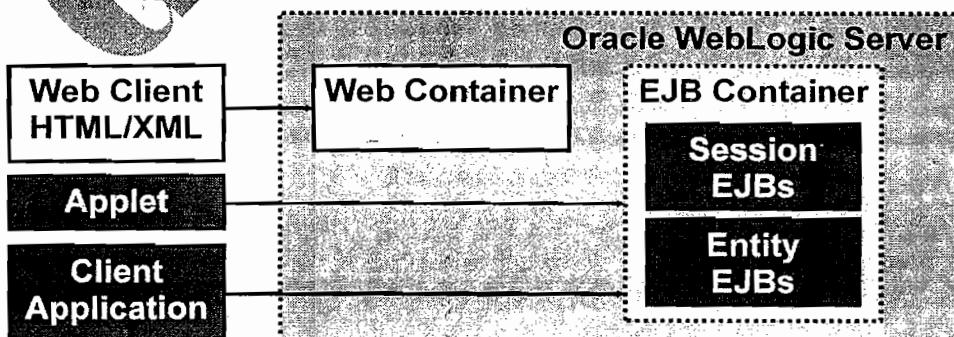
Java Management Extensions (JMX)

- Java Management Extensions (JMX):
 - Defines a standard infrastructure to manage a device from Java programs
 - Decouples the managed device from the management tools
- The specification describes MBeans, which are the building blocks of JMX.



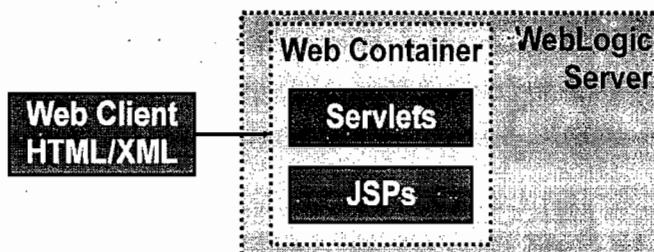
Client Application

- The client application interacts with WLS through JRMP/T3, IIOP, and COM.
- The types of clients include:
 - Stand-alone Java applications
 - Applets within a browser



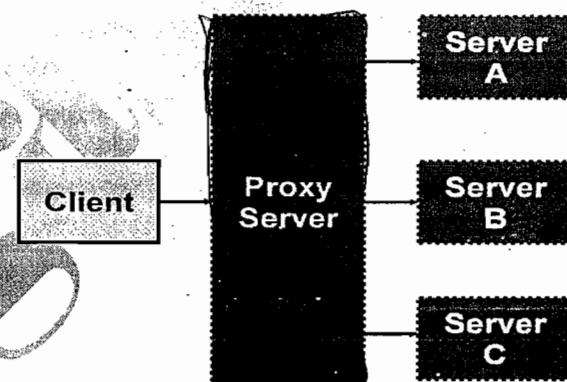
Web Client

- A Web client interacts with Oracle WebLogic Server via HTTP using servlets or JSPs.
- The types of Web clients include:
 - Browser
 - Web services (SOAP over HTTP)



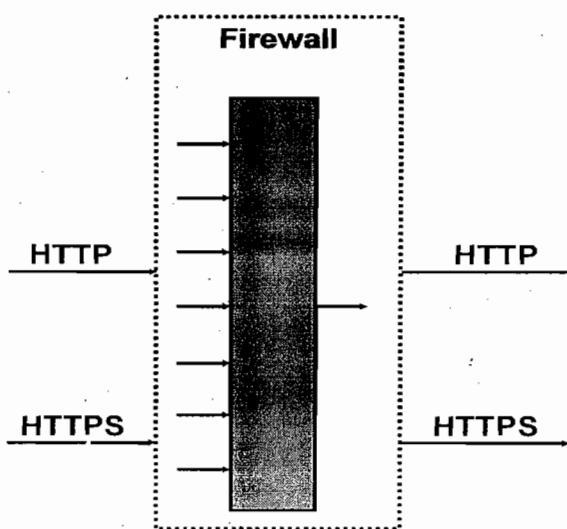
Proxy Server

- Forwards requests to other machines
- Can be used as a level of indirection and security
- Can be used to load balance a system



Firewalls

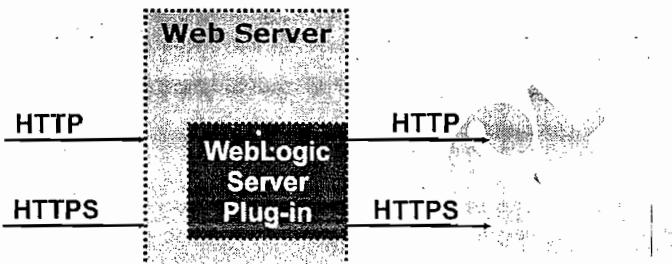
- Provide filtering, authorization, and authentication services
- Help keep out hackers
- Map port requests
- Can act as proxy servers
- Can decrease back-end network activity



Web Server

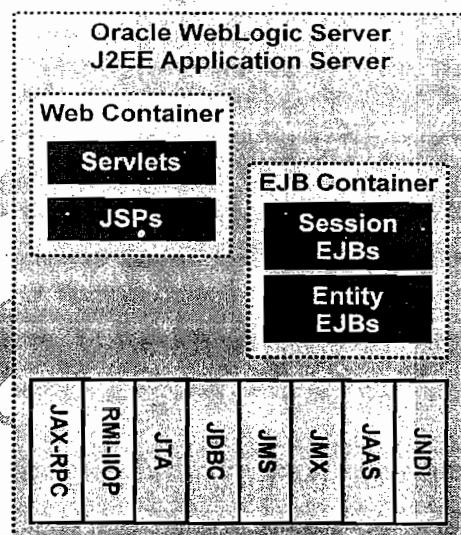
Web servers:

- Provide Web content
- Communicate via HTTP, FTP, and so forth
- Can handle CGI requests
- Proxy some requests to application servers

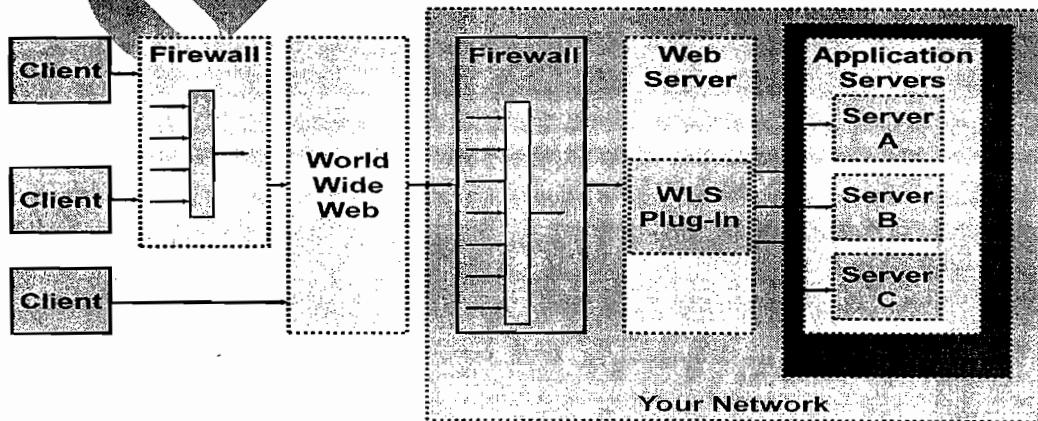


Application Servers

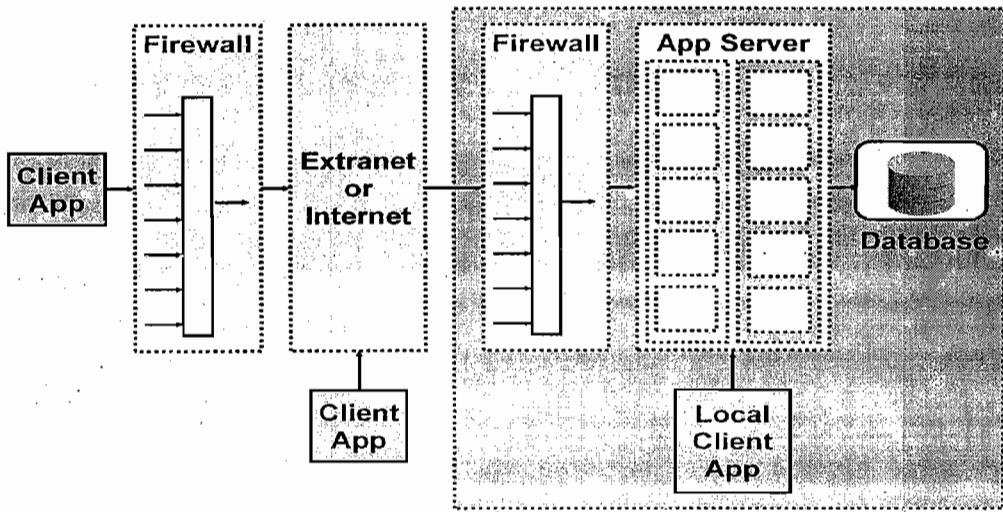
- Provide services that support the execution and availability of deployed applications
- Handle heavier processing chores than Web servers



Web Application Server Configuration

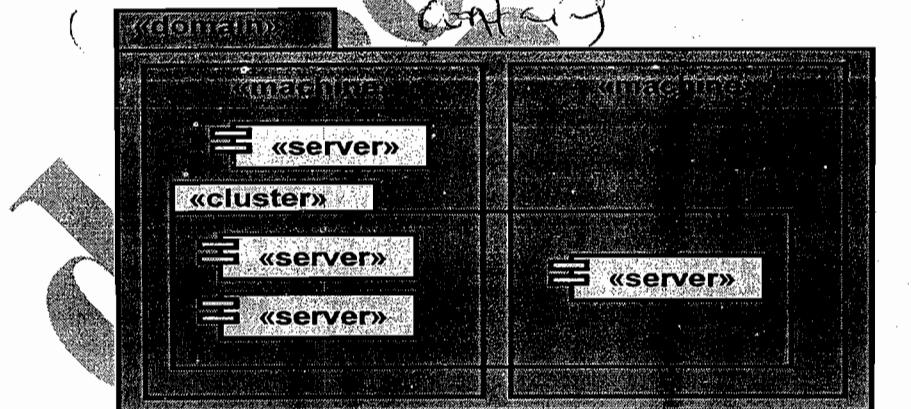


Application Server Configuration



Domain

- Is a logically related group of Oracle WebLogic Server resources that are managed as a single unit) is the combination of A.S., M.S., cluster, JMS, JDBC, JNDI components, that are managed as Single unit
- Provides one point of administration
- Can logically separate:
 - Development, test, and production applications
 - Organizational divisions



Why Use Domains

- A domain is an administration feature that:
 - Is transparent to applications
 - Can be configured and administered, for technical or business reasons, even after the applications are developed or are in production
- Oracle WebLogic Server domains can be used to separate:
 - Development, test, and production applications

- Administration and operational responsibilities
- Organizational or business divisions

Domain Restrictions

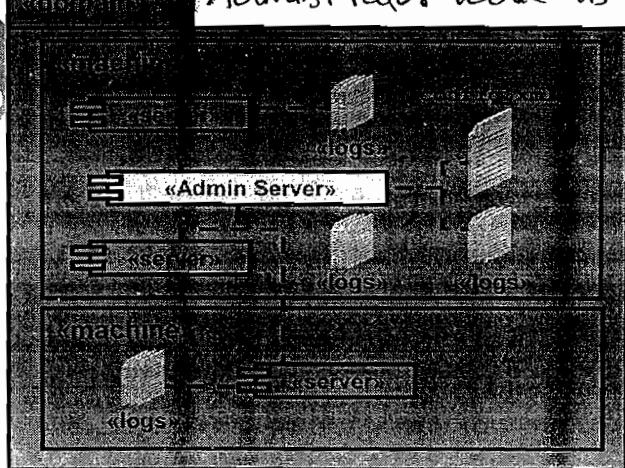
- Each domain requires its own Administration Server.
- A cluster cannot span multiple domains.
- The Managed Servers in a domain must run the same version of Oracle WebLogic Server.
- The Administration Server in a domain must run the same or higher version as Managed Servers in the domain.

Server

- A server is an instance of weblogic.Server executing in a Java Virtual Machine (JVM).
- A server:
 - Runs on a designated Oracle WebLogic Server machine
 - Has a dedicated amount of RAM
 - Is multithreaded
- Two types of servers:
 - Administration Server
 - Managed Server

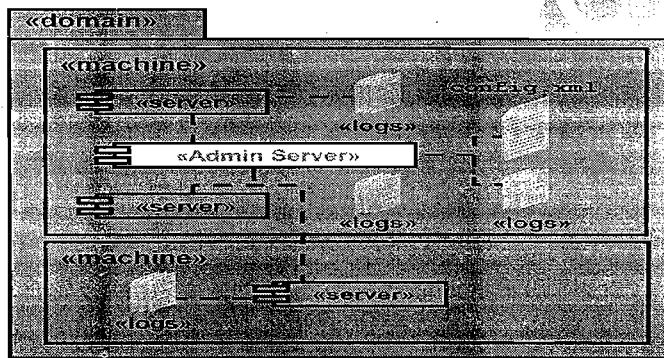
Administration Server

- Is the central point of control for a domain
- Stores the configuration information and logs for a domain
- Runs the WebLogic Administration Console
- *Any work is there I will be looked after by Admin. Server*



Managed Server

- It is any server in a domain that is not the Administration Server.
- It contacts the Administration Server for configuration information.
- It runs business applications in a production environment.
- It is independent of all other Managed Servers in the domain (unless they are in a cluster).
- You can have many Managed Servers in a domain.
- Individual Managed Servers are typically added for capacity and application isolation.

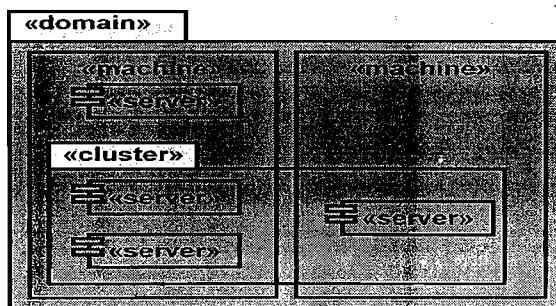


Interaction between the Administration Server and the Managed Server

- The Administration Server stores the master copy of the domain configuration, including the configuration for all Managed Servers in the domain
- Each Managed Server stores a local copy of the domain configuration file
- When a Managed Server starts, it connects to the Administration Server to synchronize the configuration
- When the configuration is changed, the Administration Server sends the changed configuration to the Managed Servers

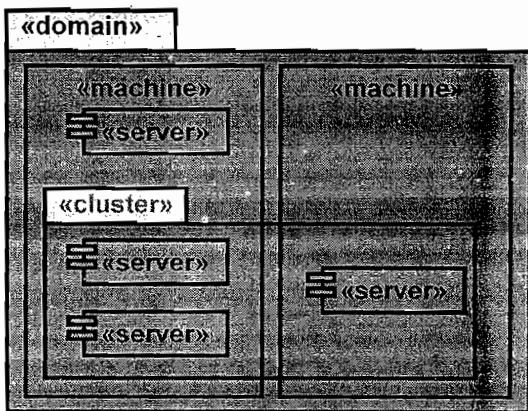
Machine

- Is a computer that hosts the Oracle WebLogic Server instances
- Runs a supported operating system platform
- Is used by Node Manager to restart a failed Managed Server



Cluster

- A cluster is a logical group of WLS servers.
- Oracle WebLogic Server clusters provide:
 - High Availability (Reliability)
 - Load balancing (Scalability)
- A cluster is transparent to a client.

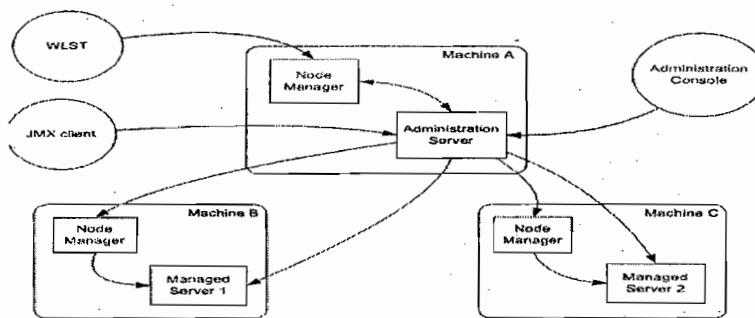


Cluster Guidelines

- A cluster cannot span domains.
- All servers in a cluster must also be in the same domain.
- All servers within a cluster must be at the same version level.
- Clustered servers can be on the same or different machines with the same or different operating systems.
- There can be multiple clusters in a domain.

Node Manager

- Is a utility or process running on a physical server that enables starting, stopping, suspending, or restarting the Administration and Managed Servers remotely.
- Is not associated with a domain:
 - Can start any server instances that reside on the same physical server
- Is required if you use the Administration Console to start servers
- Is required for Whole Server Migration and for some configurations of Automatic Server Migration
- Has the following versions:
 - Java-based
 - Script-based
- Access Node Manager using the Administration Console, the JMX utilities, or the WLST commands and scripts.

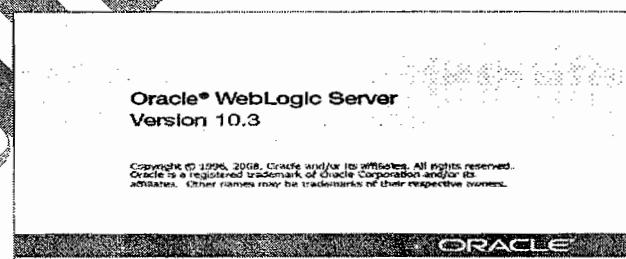


Road Map

- Distributed Architecture
 - J2EE Technologies
 - Oracle WebLogic Server Terminologies
- Setting Up an Oracle WebLogic Server Environment

Oracle WebLogic Server Installation

- ✓ You can install Oracle WebLogic Server in three different ways:
 - GUI mode
 - Console mode
 - Silent mode
- ✓ Oracle WebLogic Server supports a number of platforms including:
 - Windows 2000, 2003 Server, XP
 - Sun Solaris
 - HP-UX
 - Linux
- Set PATH, JAVA_HOME to point to JDK 6 update 5 or higher (Sun JDK 6 or JRockit Mission Control for Java 6)

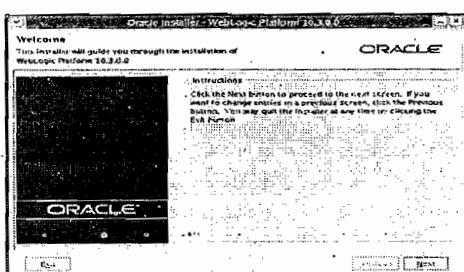


GUI Mode Installation

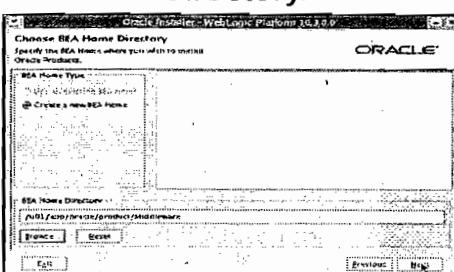
Start the Installer (net or package).

- server103_linux32.bin/server103_win32.exe

Read the Welcome screen.

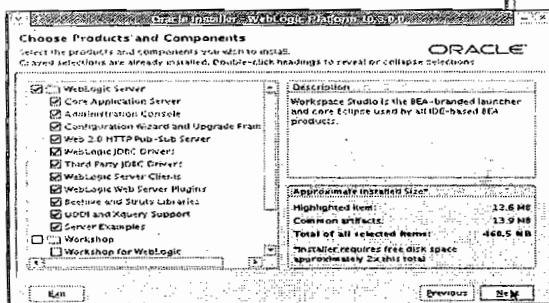
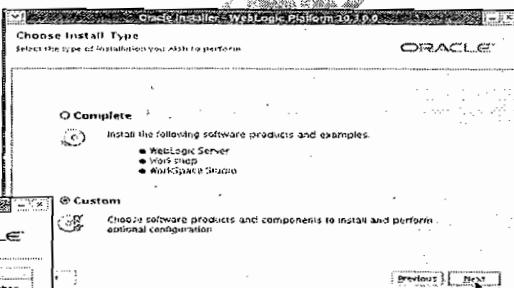


Choose the BEA Home directory.



Choosing an Installation Type and Products

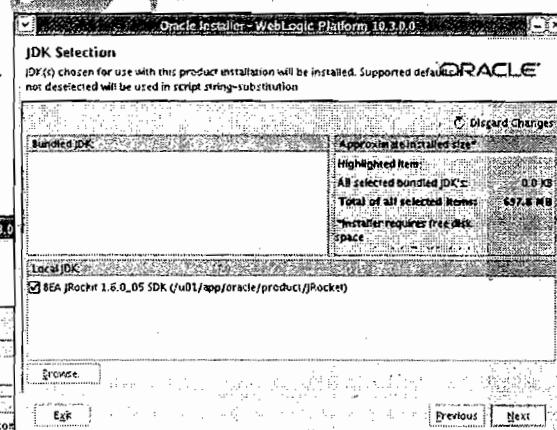
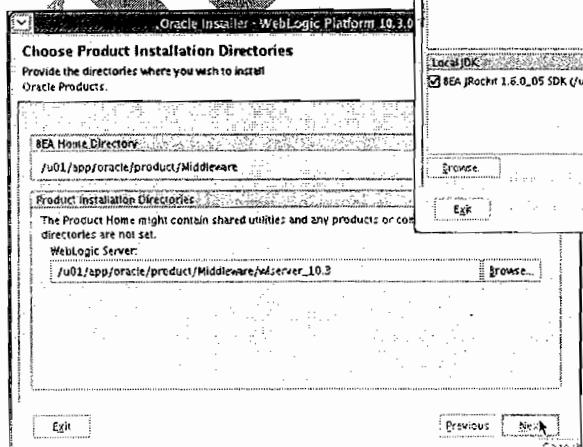
Select the installation type.



Select the products and components.

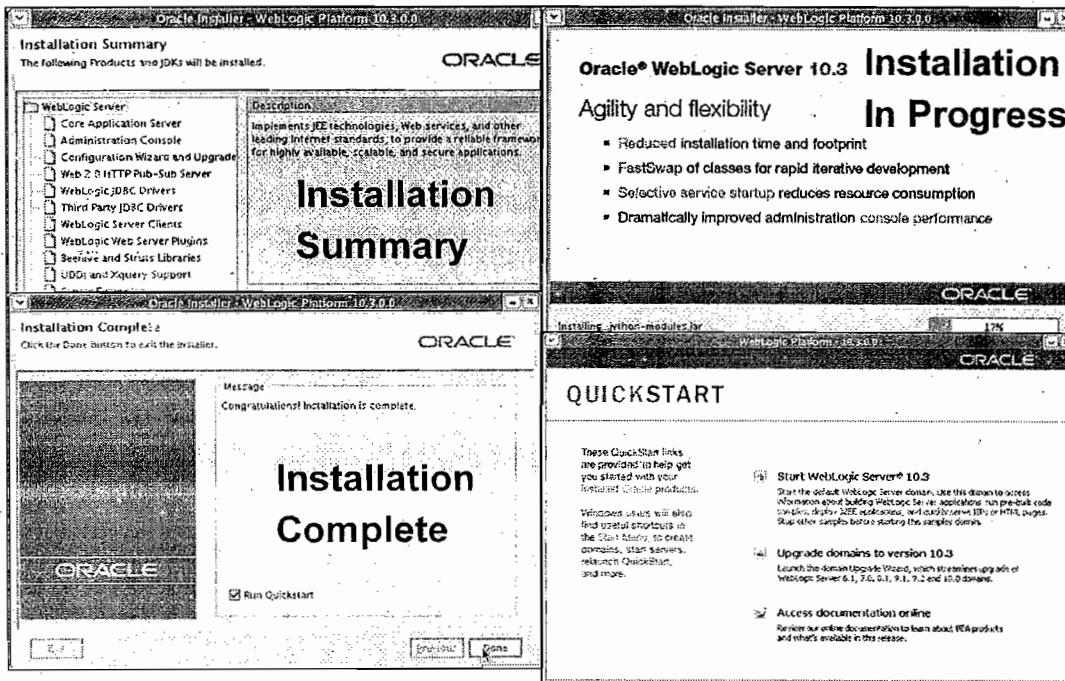
Choosing the JDK and Product Directory

Select the Java Development Kit (JDK)



Select the Oracle WebLogic Server Home.

Installation and Summary



Console and Silent Mode Installations

- Console mode:
 - server103_linux32.bin –mode=console
 - The installation steps are similar to the GUI-based installation.
- Silent mode:
 - Create a silent.xml file.
 - server103_linux32.bin –mode=silent –silent_xml=path_to_silent.xml

Postinstallation: Oracle Home

- /u01/app/oracle/product/Middleware: Oracle Home
- registry.dat/registry.xml: Record of all Oracle products
- utils: Additional or utility JAR files
- wlserver_10.3: Oracle WebLogic Server 10.3 Home
- logs: Installation logs
- modules: Modules (.jar) installed in Oracle Home

```
bash-3.00$ cd /u01/app/oracle/product/Middleware/
bash-3.00$ ls
logs  modules  registry.dat  registry.xml  utils  wlserver_10.3
bash-3.00$ cd wlserver_10.3/
bash-3.00$ ls
common  samples  server  uninstall
```

Oracle WebLogic Server Directory Structure

- **Common:** The files shared by the Oracle WebLogic Server 10.3 components, including the template JAR files used by the Configuration Wizard when creating domains
- **samples:** Sample code and resources
- **server:** Server software components (executables, database files, XML JAR files, alternative JDBC drivers, Oracle WebLogic Server JAR files, and plug-ins)
- **uninstall:** The code required to uninstall Oracle WebLogic Server 10.3

```
bash-3.00$ cd /u01/app/oracle/product/Middleware/
bash-3.00$ ls
logs modules registry.dat registry.xml utils wlserver_10.3
bash-3.00$ cd wlserver_10.3/
bash-3.00$ ls
common samples server uninstall
```

Samples Directory Structure

- **Server:** The source code for the sample domain examples that are installed with Oracle WebLogic Server
- **Domains:** The sample domain for the medrec application, examples application, and workshop application

```
bash-3.00$ pwd
/u01/app/oracle/product/Middleware/wlserver_10.3/samples
bash-3.00$ ls
domains server
bash-3.00$ cd domains/
bash-3.00$ ls
medrec wl server
bash-3.00$ cd ../server/
bash-3.00$ ls
docs examples medrec wls_samples_overview.html
bash-3.00$
```

Setting Environment Variables

- Run **setWLSEnv.sh** under **WL_HOME/server/bin** to set **WL_HOME**, **JAVA_HOME**, **PATH**, and **CLASSPATH**.
- **setWLSEnv.sh** makes a call to **commEnv.sh** under **WL_HOME/common/bin** to set **common environment variables**, such as **BEA_HOME**, **ANT_HOME**, and **POINTBASE_HOME**.
- Check the version of JDK:
 - **java -version**

```
bash-3.00$ java -version
java version "1.6.0_05"
Java(TM) SE Runtime Environment (build 1.6.0_05-b13)
BEA JRockit(R) (build R27.6.0-50_o-100423-1.6.0_05-20080626-2104-linux-ia32, com
piled mode)
bash-3.00$
```

Quiz

Which of the following statements is NOT true?

1. Managed servers in a domain must run the same version of Oracle WebLogic Server.

2. A domain comprises only the Administration Server, only the Managed Server, or the Administration and Managed Servers.
3. The Administration Server stores the configuration information and logs for a domain.
4. The Administration Server in a domain must run the same or higher version of Oracle WebLogic Server as the Managed Servers in the domain.

Answer: 2

Which of the following statements is NOT true?

1. There can be multiple clusters in a domain.
2. All the servers within a cluster must be at the same version level.
3. All the servers in a cluster must also be in the same domain.
4. A cluster can span multiple domains.

Answer: 4

The Administration Console is unavailable if the Administration Server is shut down.

1. True
2. False

Answer: 1

Which of the following statements is true?

1. There is one Node Manager for each machine.
2. There is one Node Manager for each domain.
3. There is one Node Manager for each cluster.

Answer: 1

Oracle WebLogic Server 10.3 is certified with JDK 1.6.

1. True
2. False

Answer: 1

Which JDK does Oracle WebLogic Server 10.3 come bundled with for Linux Operating System ?

1. Sun SDK 1.6
2. JRockit SDK 1.6
3. Both

Answer: 3

Net installer is preferred over the package installer if you want to install select components using only the custom option and have access to the Internet.

1. True
2. False

Answer: 1

The applications in the sample examples and medrec domains use the demo database named:

1. Oracle
2. PointBase
3. Sybase

4. SQL Server

Answer: 2

Summary

In this lesson, you should have learned about:

- The basic Oracle WebLogic Server terminology and concepts
- How to install Oracle WebLogic Server 10.3

Configuring Oracle WebLogic Server Environment

Objectives

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

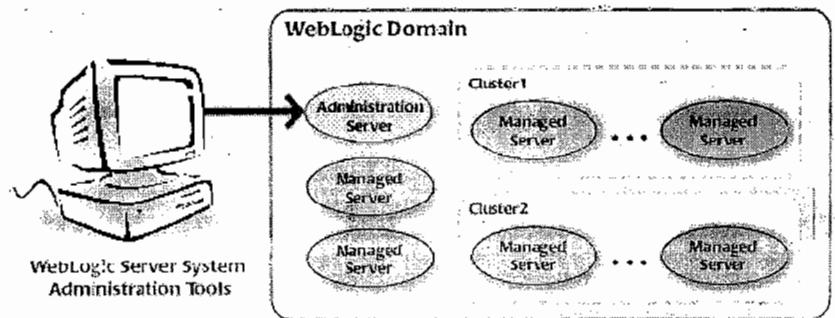
- Describe how the domain works
- Describe the domain directory structure
- Configure a domain
- Start or stop the Oracle WebLogic Server
- Configure Managed Servers
- Start Managed Servers
- Describe administration and Managed Server Independence
- Create a custom domain template
- Perform Console administration
- Perform command-line administration

Road Map

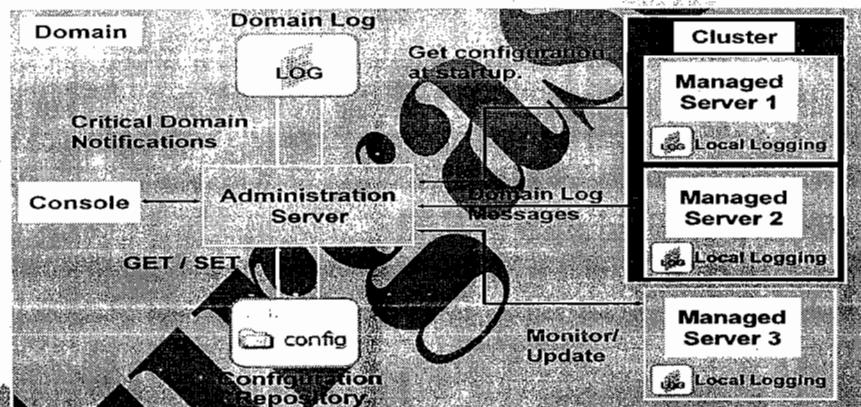
- Explaining How the Domain Works
- Describing the Domain Directory Structure
- Configuring a Domain
- Starting or Stopping the Oracle WebLogic Server
- Configuring Managed Servers and Machines
- Starting Managed Servers
- Explaining Administration and Managed Server Independence
- Creating a Custom Domain Template
- Performing Console Administration
- Performing Command-line Administration

Domain Overview

- Is the basic administration unit for Oracle WebLogic Server
- Always includes one Oracle WebLogic Server instance configured as an Administration Server
- May include optional Oracle WebLogic Server instances in a domain called Managed Servers
- May also include clusters of server instances that work together



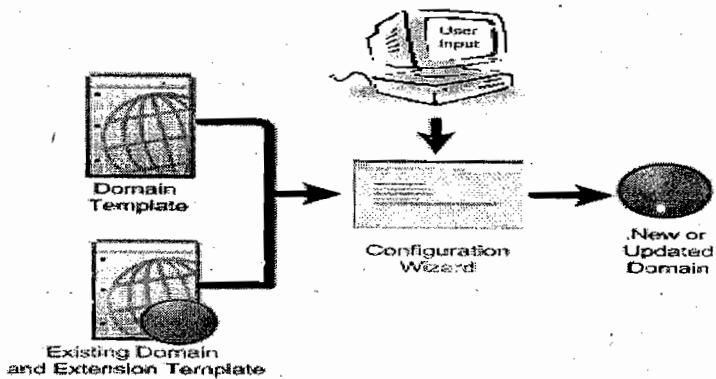
Domain Diagram



Configuring a Domain

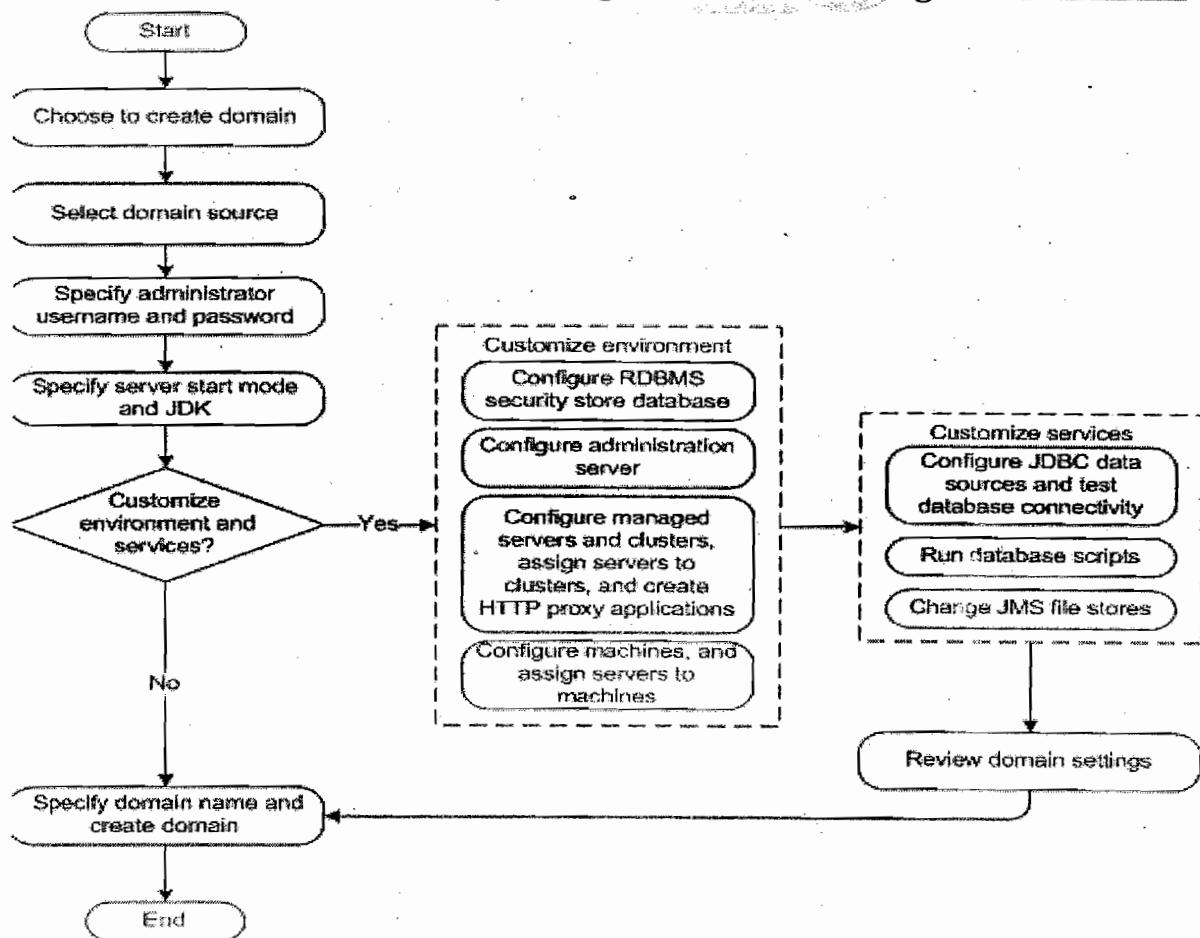
- After installation, configure a domain on which to develop and deploy applications.
- By creating a domain, you define a collection of resources, such as:
 - Managed Servers
 - Clusters
 - Database connections
 - Security services
 - J2EE applications
- Use the Configuration Wizard to create and configure domains.
- Common domain configurations:
 - Development or test
 - Production

Starting the Domain Configuration Wizard

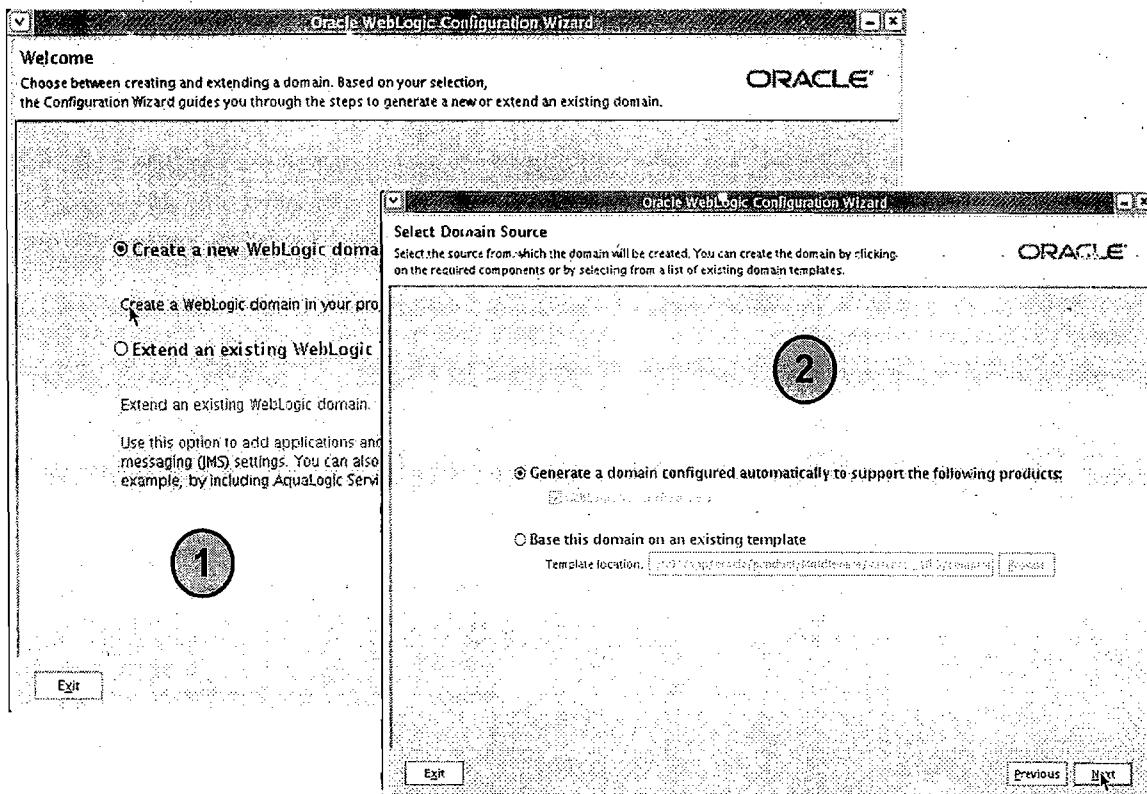


- Scripts are in the <WL_HOME>/common/bin directory
- Graphical mode:
 - config.sh
- Console mode:
 - config.sh -mode=console

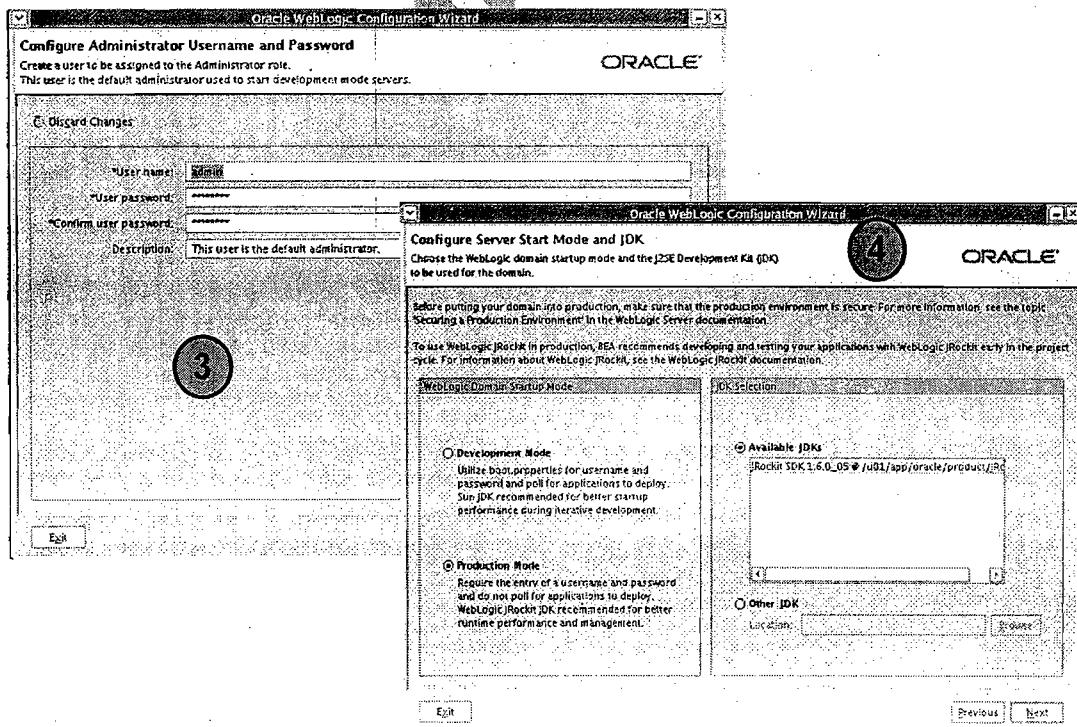
Flow Chart to Create a Domain by Using the Domain Configuration Wizard



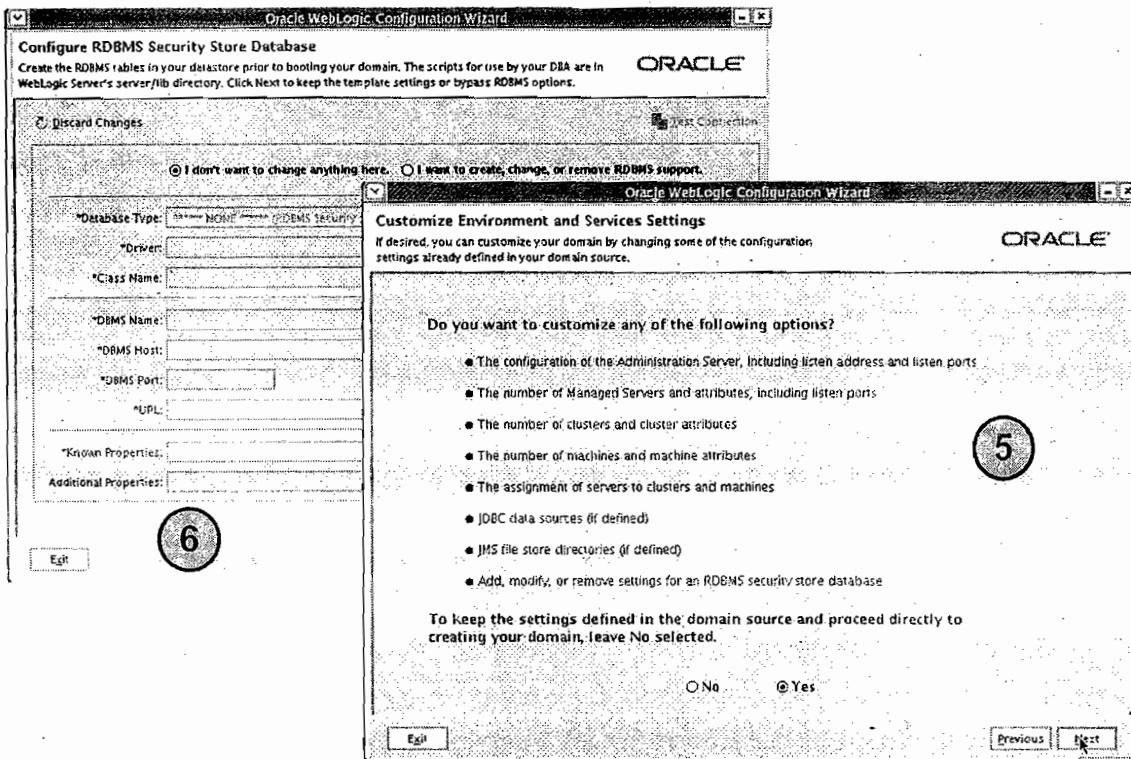
Creating a New WebLogic Domain and Selecting the Domain Source



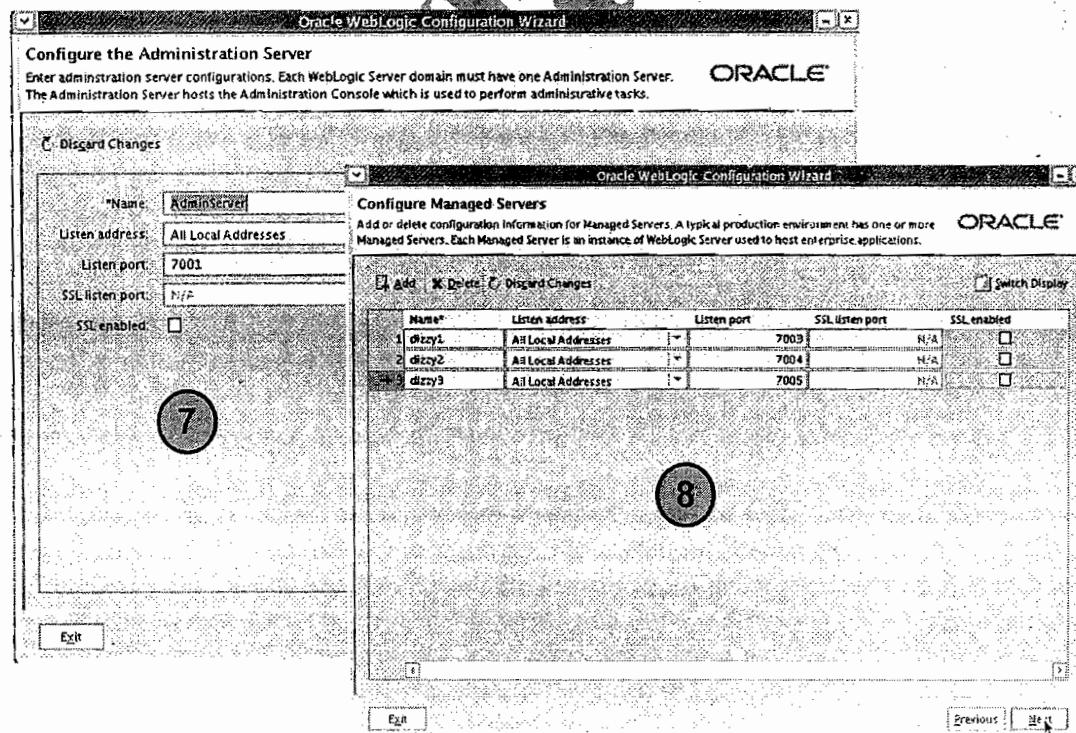
Configuring Administrator Settings, Start Mode, and JDK



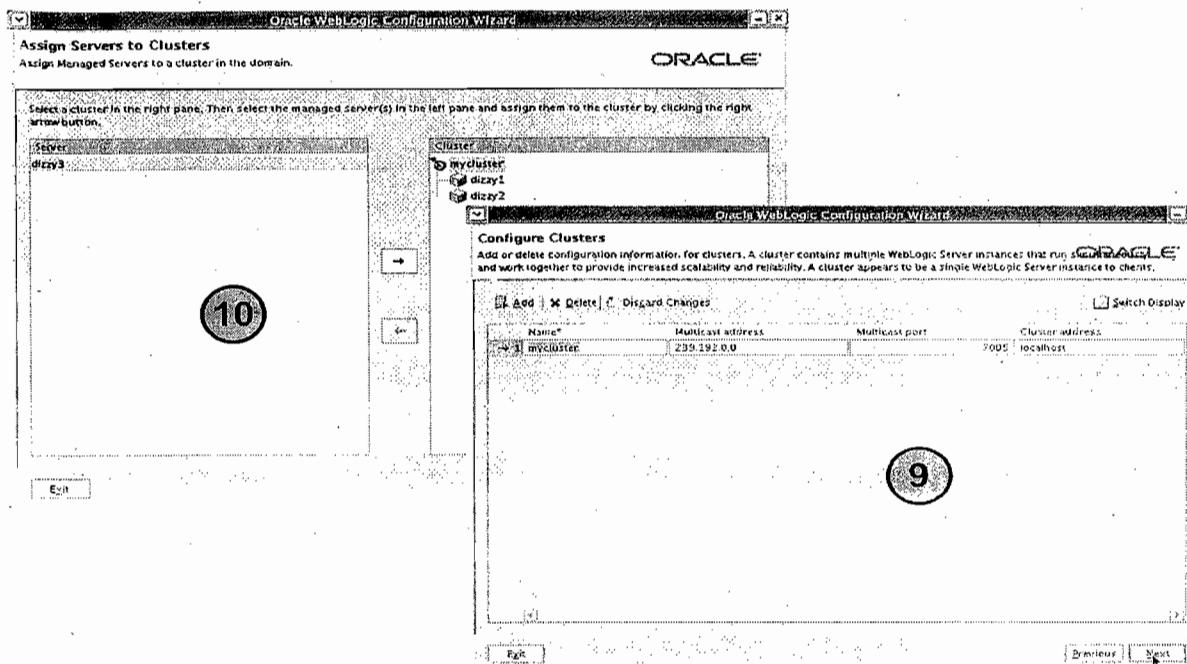
Customizing the Environment and RDBMS Security Store



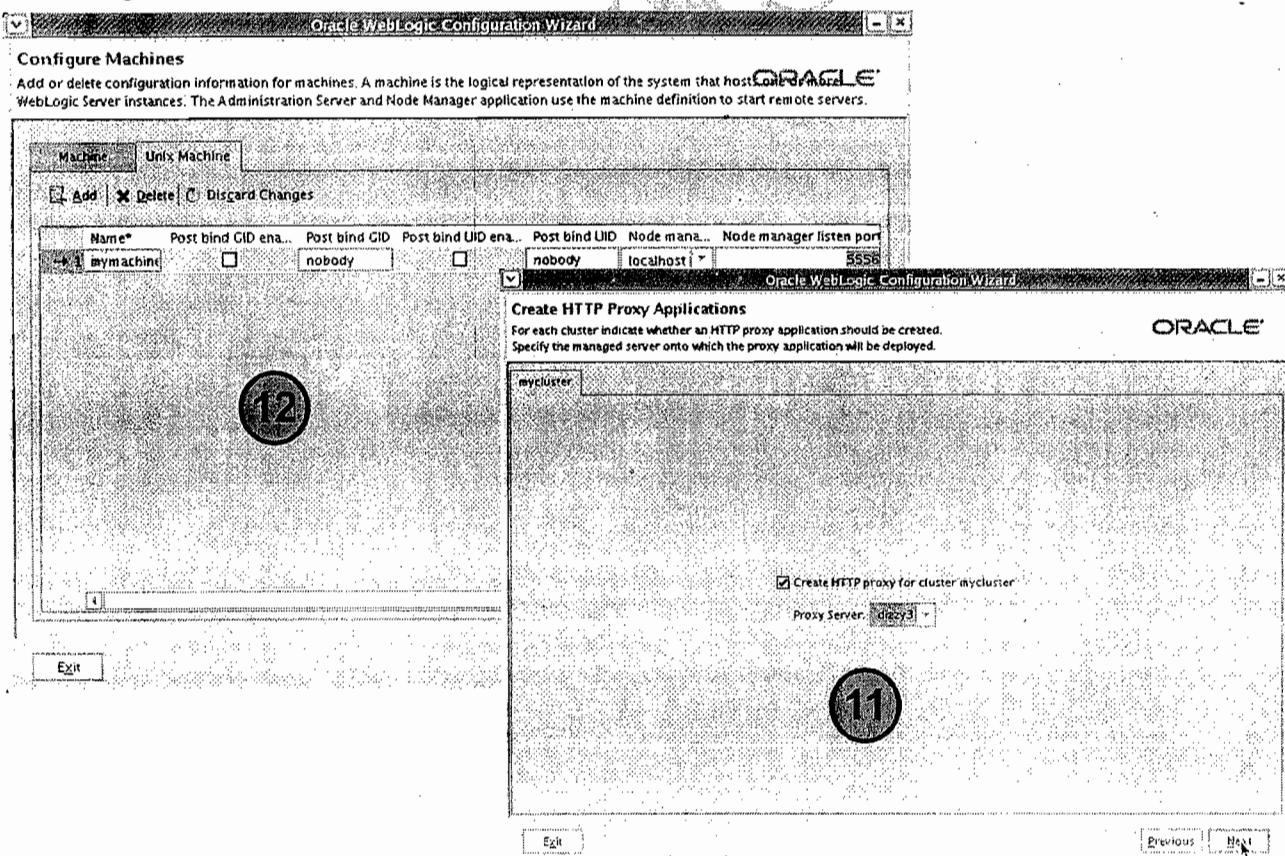
Configuring the Administration and Managed Servers



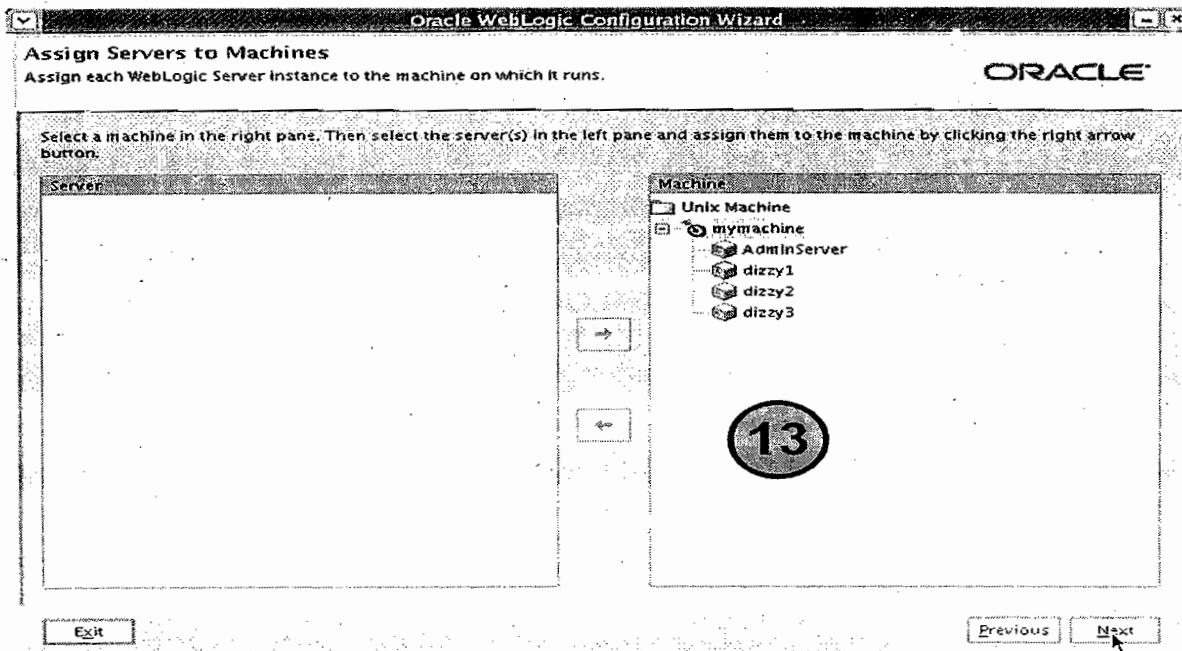
Configuring Clusters and Assigning Servers to Clusters



Creating an HTTP Proxy Application and Configuring Machines



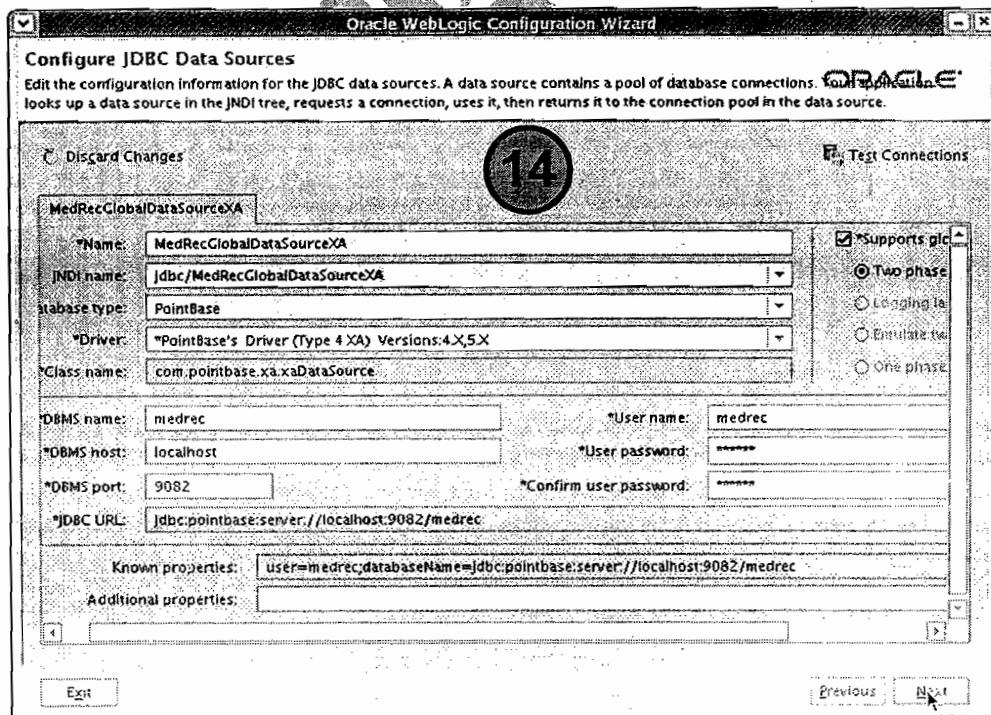
Assigning Servers to Machines and Configuring JDBC Data Sources



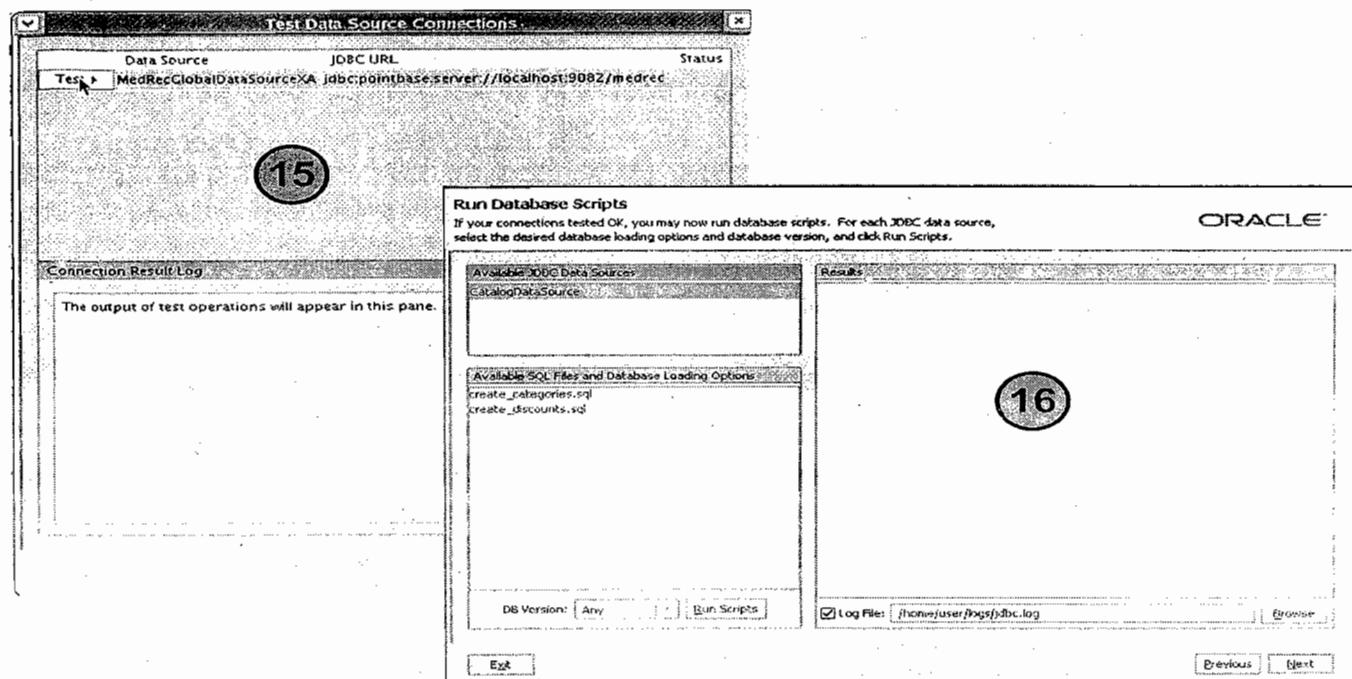
No databases are defined in the wls.jar template. Therefore, the Configure JDBC Data Sources page is not displayed.

Configuring JDBC Data Sources

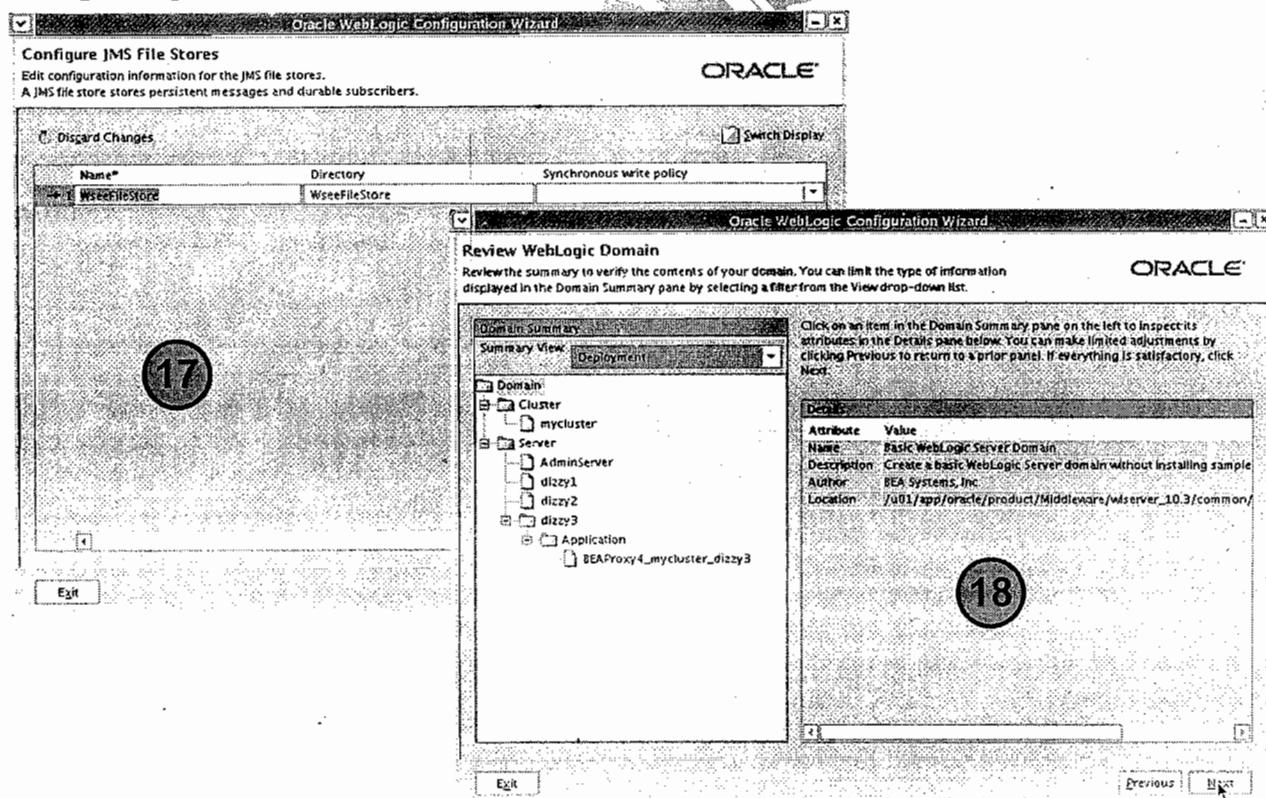
- Select the “Extend an existing domain using existing extension template” option.
- Choose a template that contains JDBC data sources and JMS store definitions.



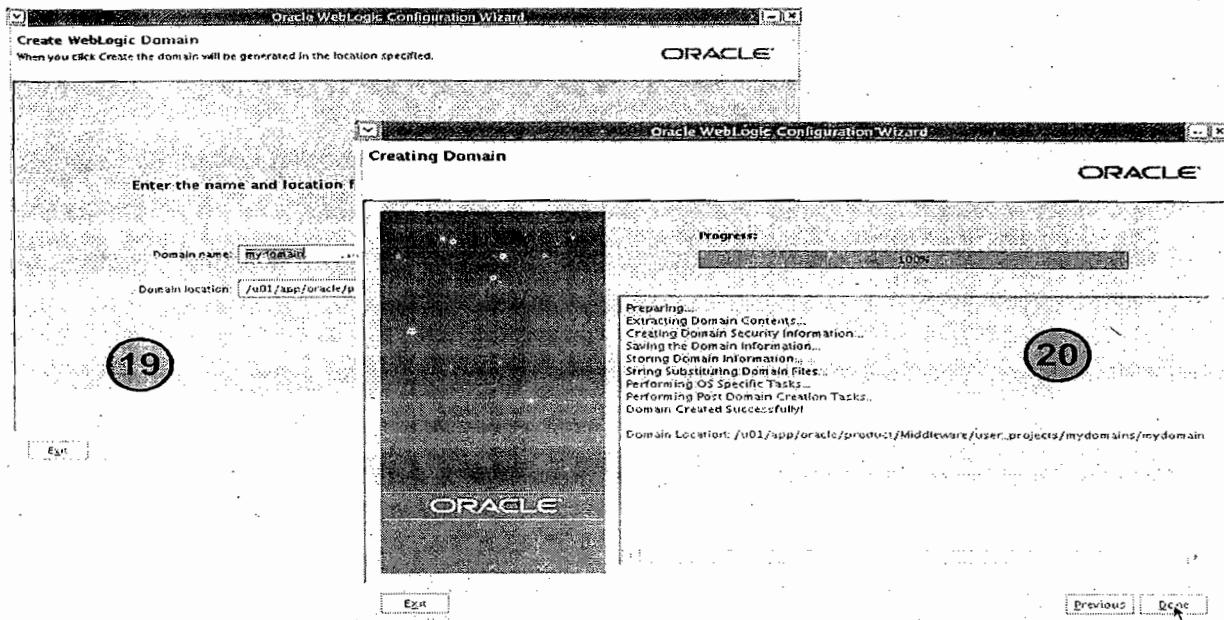
Testing Data Source Connections and Running Database Scripts



Configuring the JMS File Store and Reviewing the WebLogic Domain



Creating the WebLogic Domain



Domain Directory Structure

Directory	Description
domain-name	The name of this directory is the name of the domain.
autodeploy	In development mode, WLS automatically deploys any applications or modules that you place in this directory.
bin	The scripts for starting and stopping the Administration Server and the Managed Servers in the domain
config	The current configuration and deployment state of the domain; config.xml
console-ext	Console extensions
domain-info	Domain initialization information
lib	JAR files added to CLASSPATH of each server instance
pending	Domain configuration changes that have been requested, but not yet been activated
security	Domainwide security-related files
servers	One subdirectory for each server in the domain
server-name	The server directory for the WLS instance with the same name

JVM Run-Time Arguments

- Oracle WebLogic Server can be executed with most Java Virtual Machines such as Sun JVM or JRockit.
- Oracle WebLogic Server supports JDK 1.6.
- The syntax for running a virtual machine:
 - java options FullyQualifiedJavaClass ProgramOptions
- Some virtual machine options:
 - -Xms: The minimum size of the dynamic heap
 - -Xmx: The maximum size of the dynamic heap
 - -Dprop=val: An environment variable that is accessible by the program
 - -classpath CLASSPATH: The list of files or directories that contain the dependent classes

Oracle WebLogic Server Dependencies

To run Oracle WebLogic Server, configure the following environment variables:

- PATH to include all executable programs (including the Java interpreter)
- CLASSPATH to include dependencies

Configuring CLASSPATH

- The Oracle WebLogic Server CLASSPATH is configured by the Java system CLASSPATH environment variable.
- Files that must be in CLASSPATH:
 - WL_HOME/server/lib/weblogic.jar
 - Any additional service pack JAR files
- Files that can be in CLASSPATH:
 - WL_HOME/common/eval/pointbase/lib/pbclient57.jar
 - WL_HOME/common/eval/pointbase/lib/pbembedded57.jar
 - WL_HOME/server/lib/log4j.jar

Starting Oracle WebLogic Administration Server

Start the Administration Server by using the following:

- weblogic.Server (only in development)
- Start menu (only Windows)
- DOMAIN_DIR/bin/startWebLogic.sh
- WebLogic Scripting Tool (WLST) and Node Manager
- WLST without Node Manager

Starting Administration Server by Using the java weblogic.Server Command

- Run WL_HOME/server/bin/setWLSEnv.sh.
- Run java weblogic.Server.
- Optionally, run java weblogic.Server with these additional options:

java -server -Xms256m -Xmx512m -classpath "CLASSPATH"

- Dweblogic.Name=SERVER_NAME
- Dplatform.home=WL_HOME
- Dweblogic.management.username=WLS_USER
- Dweblogic.management.password=WLS_PW
- Djava.security.policy= WL_HOME\server\lib\weblogic.policy
- weblogic.Server

Starting Administration Server by Using startWebLogic.sh

Run DOMAIN_HOME/bin/startWebLogic.sh.

- Sets the environment by using setDomainEnv.sh
- Invokes java weblogic.Server

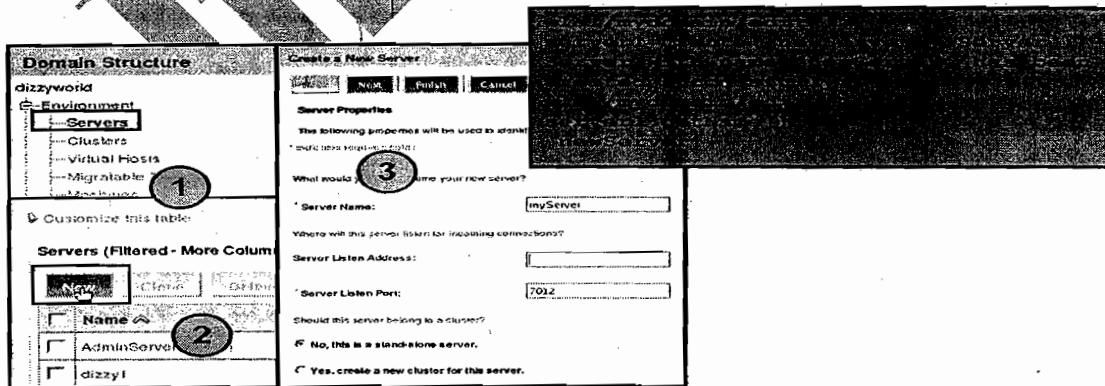
```
<Jul 20, 2008 2:42:41 PM GMT+07:00> <Info> <Management> <BEA-141107> <Version: WebLogic Server 10.3 Mon Jun 16 22:31:55 EDT 2008 1127403 >
<Jul 20, 2008 2:42:43 PM GMT+07:00> <Info> <Security> <BEA-090065> <Getting boot identity from user: >
Enter username to boot WebLogic server:admin
Enter password to boot WebLogic server:
<Jul 20, 2008 2:42:55 PM GMT+07:00> <Notice> <WebLogicServer> <BEA-000365> <Server state changed to STARTING>
<Jul 20, 2008 2:42:55 PM GMT+07:00> <Info> <WorkManager> <BEA-002900> <Initializ
```

```
1>
<Jul 20, 2008 2:43:16 PM GMT+07:00> <Notice> <WebLogicServer> <BEA-000365> <Server state changed to RUNNING>
<Jul 20, 2008 2:43:16 PM GMT+07:00> <Notice> <WebLogicServer> <BEA-000360> <Server started in RUNNING mode>
```

Configuring Managed Servers

You can configure Managed Servers by using the following:

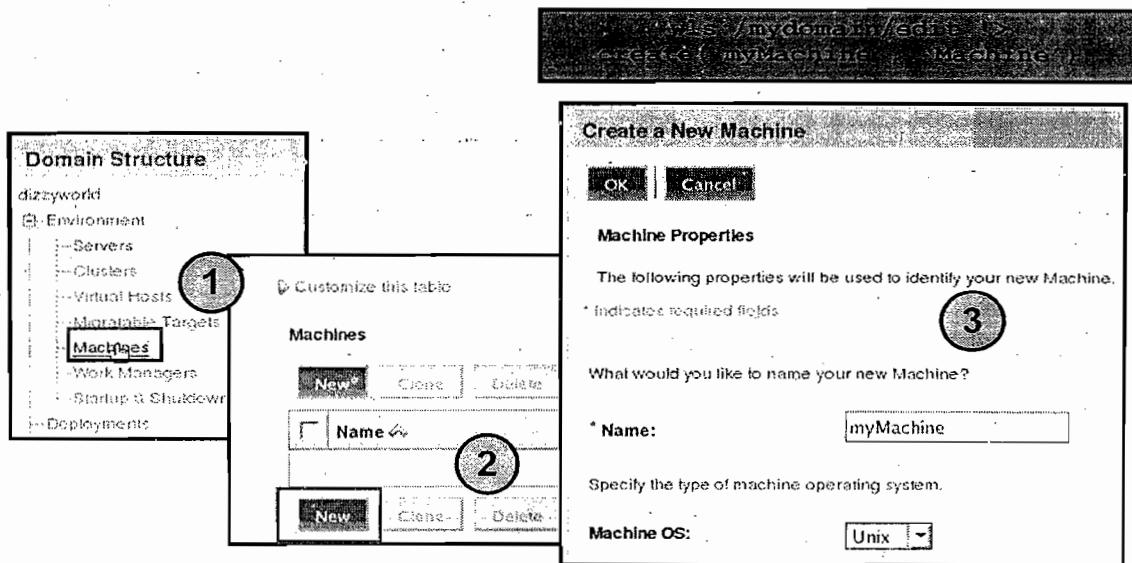
- Domain Configuration Wizard
- Administration Console
- Command Line (WLST)



Configuring Machines

You can configure machines by using the following:

- Domain Configuration Wizard
- Administration Console
- Command Line (WLST)



Starting Oracle WebLogic Managed Servers

Start Managed Servers by using:

- weblogic.Server
- DOMAIN_DIR/bin/startManagedWebLogic.sh
- Administration Console
- WLST and Node Manager

Starting a Managed Server by Using the java weblogic.Server Command

- Run WL_HOME/server/bin/setWLSEnv.sh.
- **Start Administration Server:**
 - java weblogic.Server
- **Start a Managed Server:**
 - java -Dweblogic.Name=managed_server_name -Dweblogic.management.server=url_Admin_Server weblogic.Server

Starting a Managed Server by Using startManagedWebLogic.sh

Run DOMAIN_HOME/bin/startManagedWebLogic.sh.

- Start the domain's Administration server.
- startManagedWebLogic.sh managed_server_name admin_url

```

bash-3.00$ cd /home/oracle/wls_sysadmin/work/domains/dizzyworld/bin
bash-3.00$ ls
nodemanager      setDomainEnv.sh          startWebLogic.sh
server_migration startManagedWebLogic.sh stopManagedWebLogic.sh
service_migration startPointBaseConsole.sh stopWebLogic.sh
bash-3.00$ ./startManagedWebLogic.sh dizzy1 http://edrsr31pl:7001

```

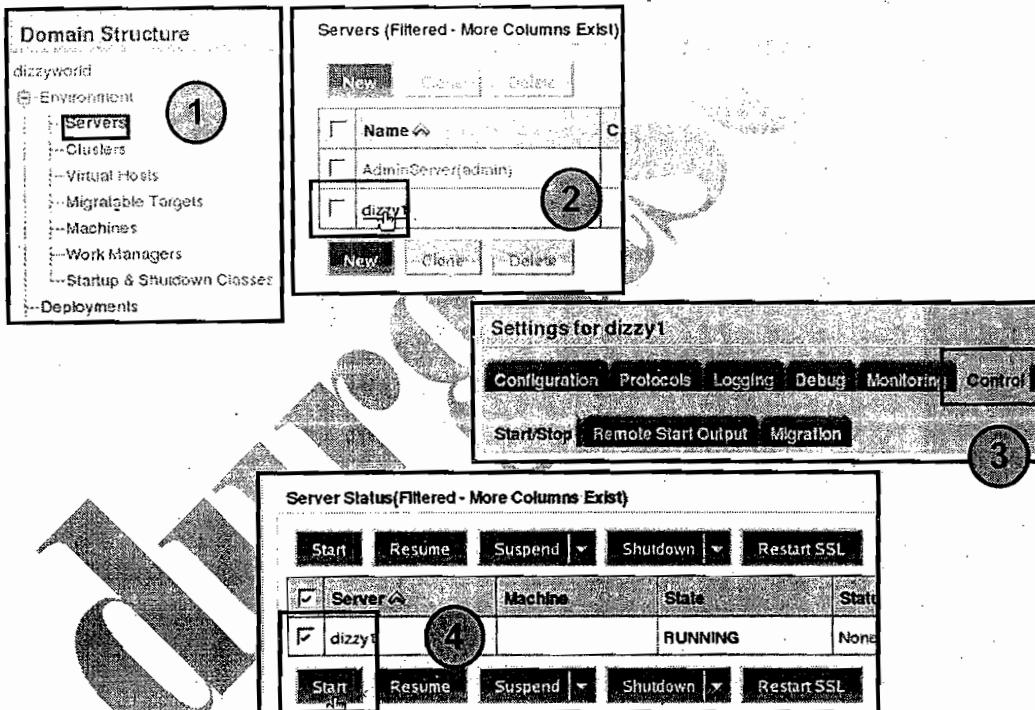
```

<Jul 21, 2008 7:04:50 PM GMT+07:00> <Info> <Security> <BEA-090065> <Getting boot
identity from user>
Enter username to boot WebLogic server:admin
Enter password to boot WebLogic server:

<Jul 21, 2008 7:05:17 PM GMT+07:00> <Notice> <WebLogicServer> <BEA-000330> <Star
ted WebLogic Managed Server "dizzy1" for domain "dizzyworld" running in Producti
on Mode>
<Jul 21, 2008 7:05:19 PM GMT+07:00> <Notice> <WebLogicServer> <BEA-000365> <Serv
er state changed to RUNNING>
<Jul 21, 2008 7:05:19 PM GMT+07:00> <Notice> <WebLogicServer> <BEA-000360> <Ser
ver started in RUNNING mode>

```

Starting a Managed Server by Using the Administration Console



Creating a Boot Identity File

- Create a file called `boot.properties` in the `<DOMAIN_HOME>\servers\<server_name>\security` directory that contains two lines:
 - `username=username`
 - `password=password`
- The first time you start the server, the server reads the Boot Identity file and overwrites it with an encrypted version of the username and password.
- Thereafter, the server remembers the credentials for subsequent startup cycles.

Managed Server Independence (MSI)

- By default, Managed Servers can function independently of the Administration Server.
- A Managed Server instance can start in MSI mode if the Administration Server is unavailable.
- Configure MSI mode from the Administration Console.
- To start a Managed Server in MSI mode, perform the following:
 - Ensure that the Managed Server's root directory contains the config subdirectory.
 - If the config subdirectory does not exist, copy it from the Administration Server's root directory.
 - Start the Managed Server at the command line or by using a script.
- Environment > Servers > Server_Name > Tuning > Advanced > Managed Server Independence Enabled check box
- If the Administration Server is unavailable at boot time, Managed Servers search for:
 - config.xml
 - SerializedSystemIni.dat
 - boot.properties(optional)
- Each Managed Server looks in its local config directory for config.xml, a replica of the domain's config.xml.
- You cannot change the configuration of the Managed Server that is running in MSI mode until it restores communication with the Administration Server.

When the Administration Server Is Down

- The Administration Server:
 - Can go down without affecting the operation of the Managed Servers
 - Can be restarted when the Managed Servers are still running
- When an Administration Server goes down:
 - The domain log entries are unavailable while it is down
 - Managed Servers can start in independent mode
 - The Administration Console and the management tools are unavailable

Restarting an Administration Server on a New Machine

- Oracle WebLogic Server allows the creation of a backup of the server as follows:
 - Install Oracle WebLogic Server on a backup machine.
 - Copy the application files to a backup machine.
 - Copy the configuration files to a backup machine.
 - Restart the Administration Server on a new machine.
- The new Administration Server contacts the Managed Servers and informs them that it is running on a new IP address.

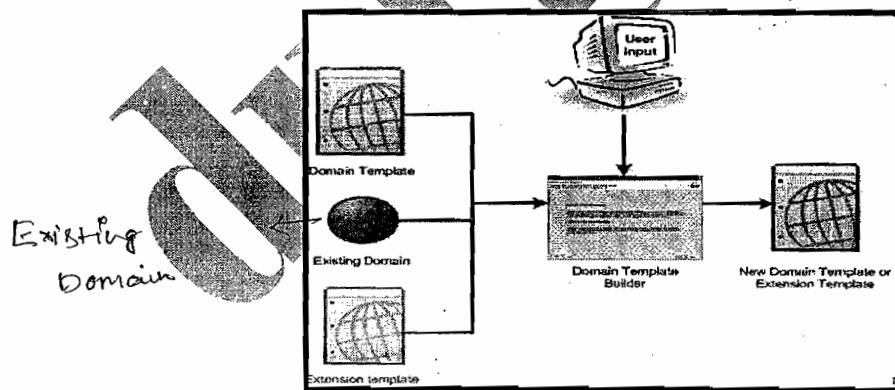
Running Multiple WLS Instances

- You can run multiple instances of WLS using different configurations on the same physical machine at the same time by either:
 - Assigning multiple IP addresses to a machine (multihoming) and defining each server to use a unique IP address
 - Specifying the same IP address but using different listen ports
- A multihomed machine:
 - Is a machine with multiple IP addresses
 - Can run a different WLS instance that is bound to each IP address
 - Can be used to configure a cluster on a single machine

Custom Domain Templates

- A domain template defines the full set of resources within a domain.
- Oracle provides sample templates for creating any platform domain.
- There are three ways to create domain templates:
 - WLST offline command line tool
 - pack command
 - Domain Template Builder
- Use the Domain Template Builder to create a domain template or an extension template.
- Using the Domain Template Builder:
 - Define a domain and replicate it across multiple projects
 - Distribute a domain packed with an application that has been developed to run in it

Domain Template Builder



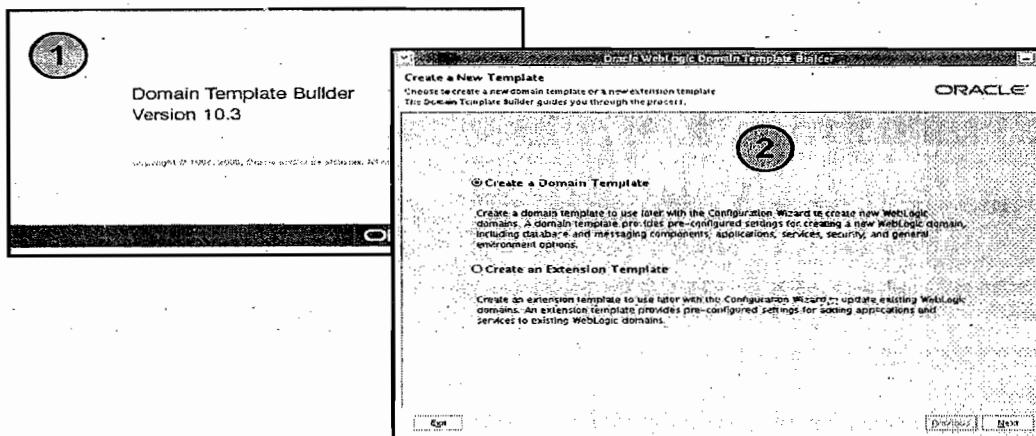
- Available in GUI mode and not console mode
- A stand-alone Java application to create custom domain and extension templates

Creating a Domain Template

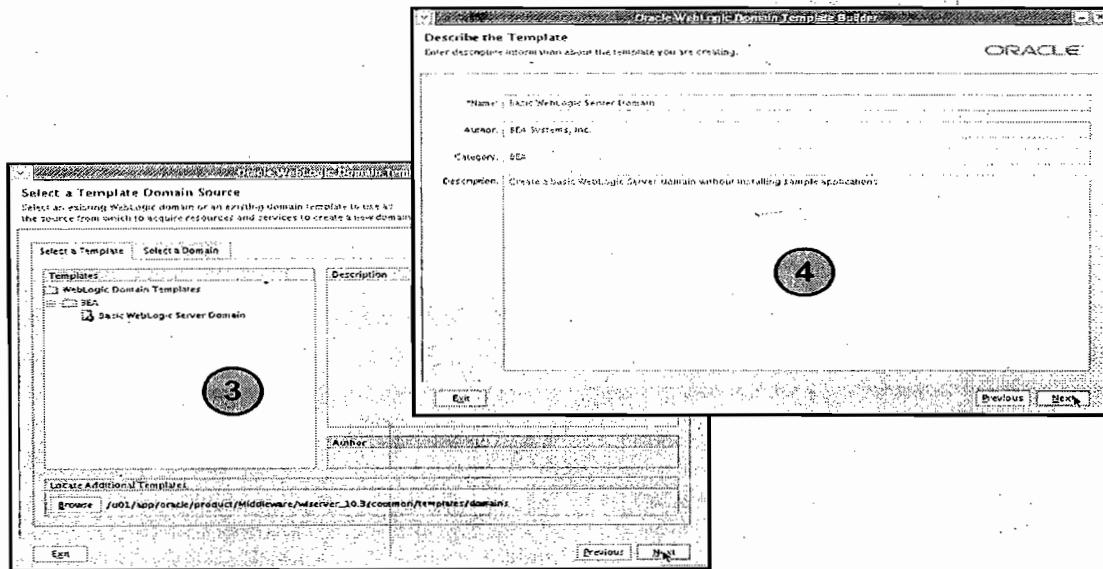
Start the Domain Template Builder:

- config_builder.sh under WL_HOME/common/bin

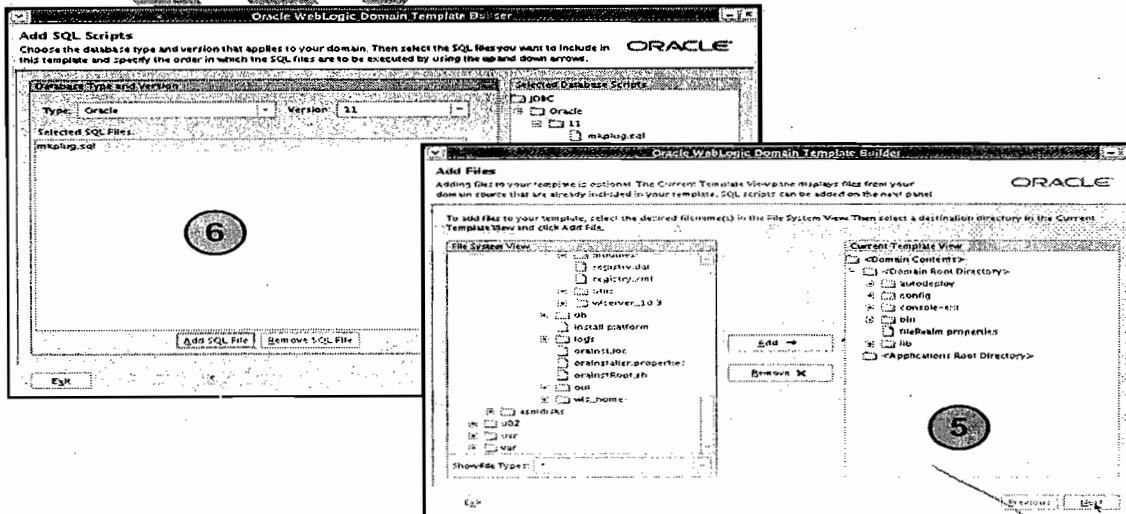
test later



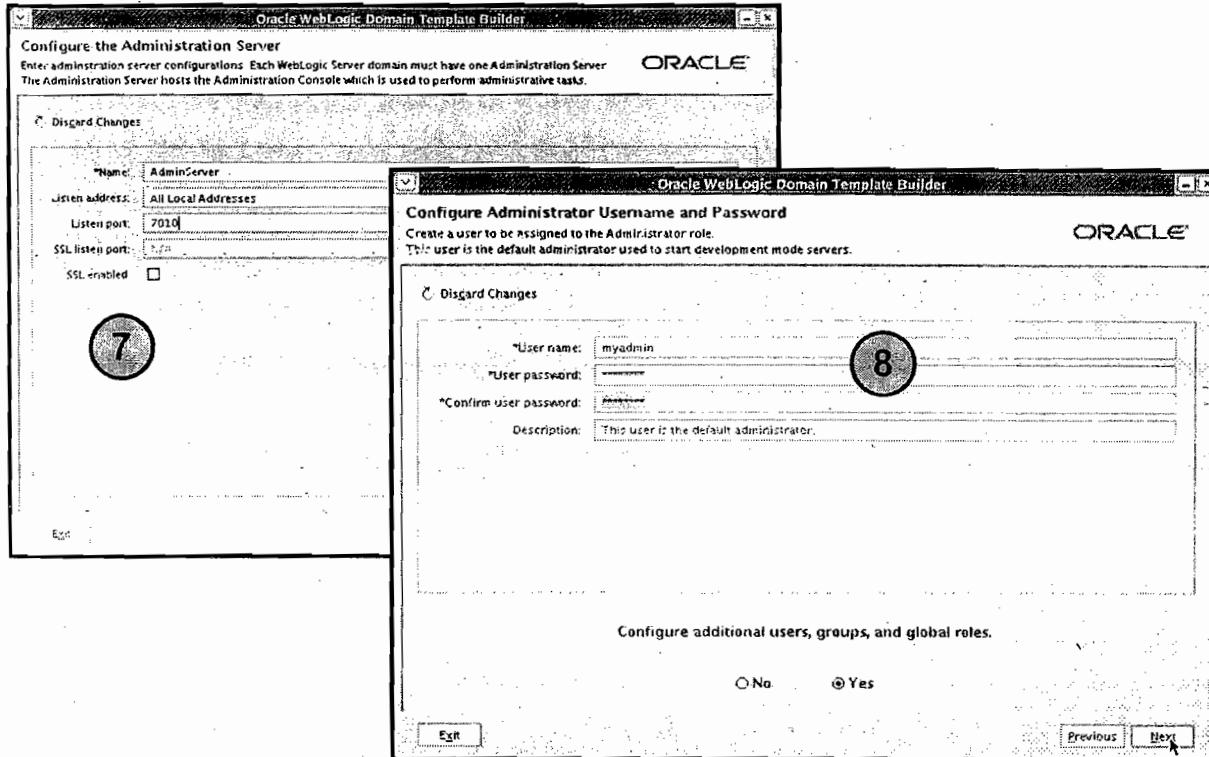
Selecting the Template Domain Source and Describing the Template



Adding Files and SQL Scripts to the Template



Configuring the Administration Server, Username, and Password



Configuring Users, Groups, and Roles

The image shows the Oracle WebLogic Domain Template Builder interface for configuring users, groups, and roles.

User Configuration (Top Left):

A table for managing users. It has columns for "Name*" and "Description". A circled "9" is in the top-right corner of this section.

Name*	Description
Administrators	Administrators can view and modify all resource attributes and start and stop servers.
Deployers	Deployers can view all resource attributes and deploy applications.
Operators	Operators can view and modify all resource attributes and perform server lifecycle operations.
Monitors	Monitors can view and modify all resource attributes and perform operations not restricted by roles.
AppTesters	AppTesters group.
CrossDomainConnectors	CrossDomainConnectors can make inter-domain calls from foreign domains.
AdminChannelUsers	AdminChannelUsers can access the admin channel.

Group Configuration (Top Right):

A table for managing groups. A circled "Group" is to the right of this section.

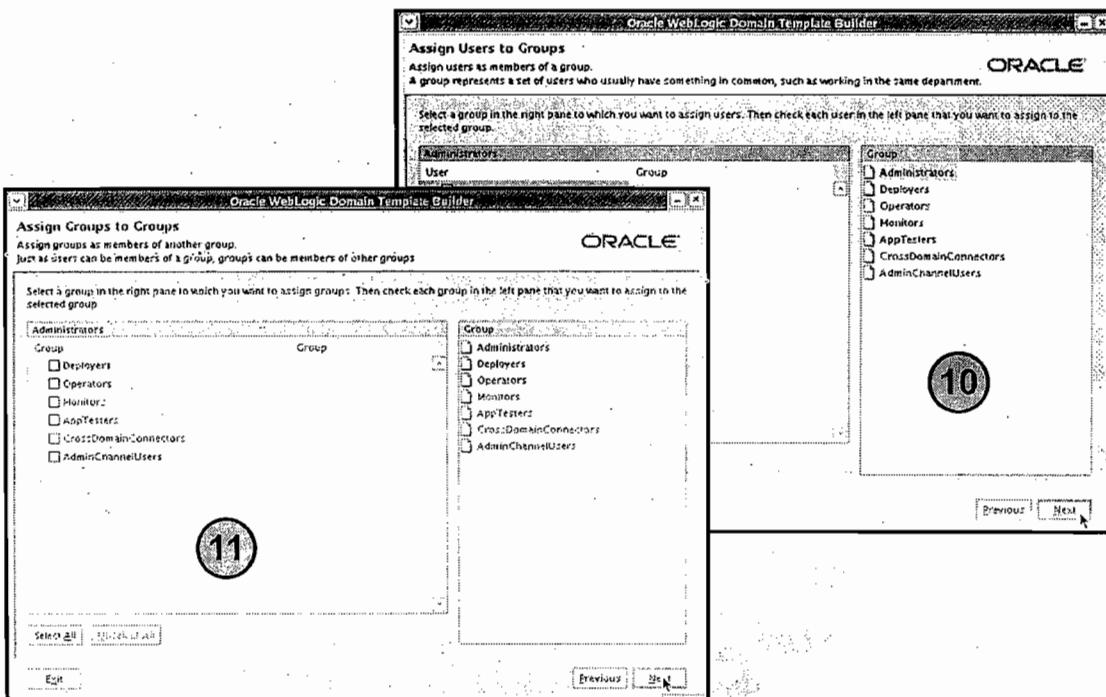
Name*	Description
Admin	built-in admin role
Deployer	View the server configuration and deploy applications.
Operator	Operators can view and modify all resource attributes and perform server lifecycle operations.
Monitor	Monitors can view and modify all resource attributes and perform operations not restricted by roles.
AppTester	AppTester role.
CrossDomainConnector	CrossDomainConnectors can make inter-domain calls from foreign domains.
AdminChannelUser	AdminChannelUsers can access the admin channel.
Anonymous	built-in anonymous role

Role Configuration (Bottom Left):

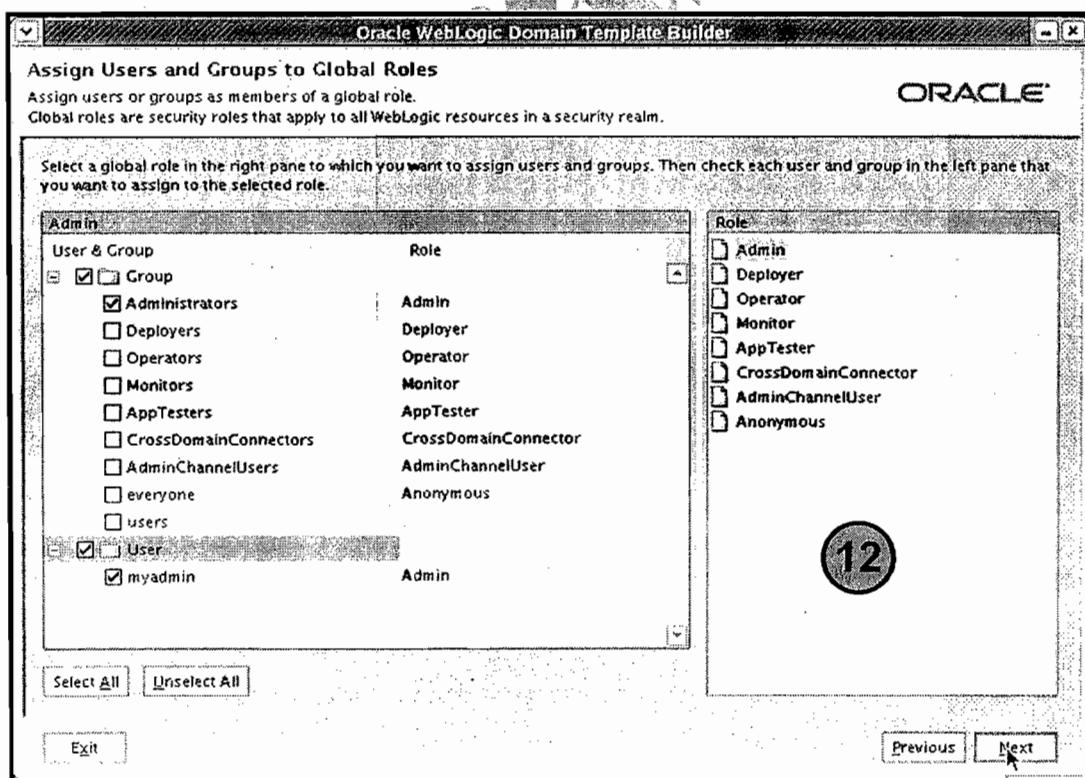
A table for managing roles. A circled "Role" is to the left of this section.

Name*	Description
Admin	built-in admin role
Deployer	View the server configuration and deploy applications.
Operator	Operators can view and modify all resource attributes and perform server lifecycle operations.
Monitor	Monitors can view and modify all resource attributes and perform operations not restricted by roles.
AppTester	AppTester role.
CrossDomainConnector	CrossDomainConnectors can make inter-domain calls from foreign domains.
AdminChannelUser	AdminChannelUsers can access the admin channel.
Anonymous	built-in anonymous role

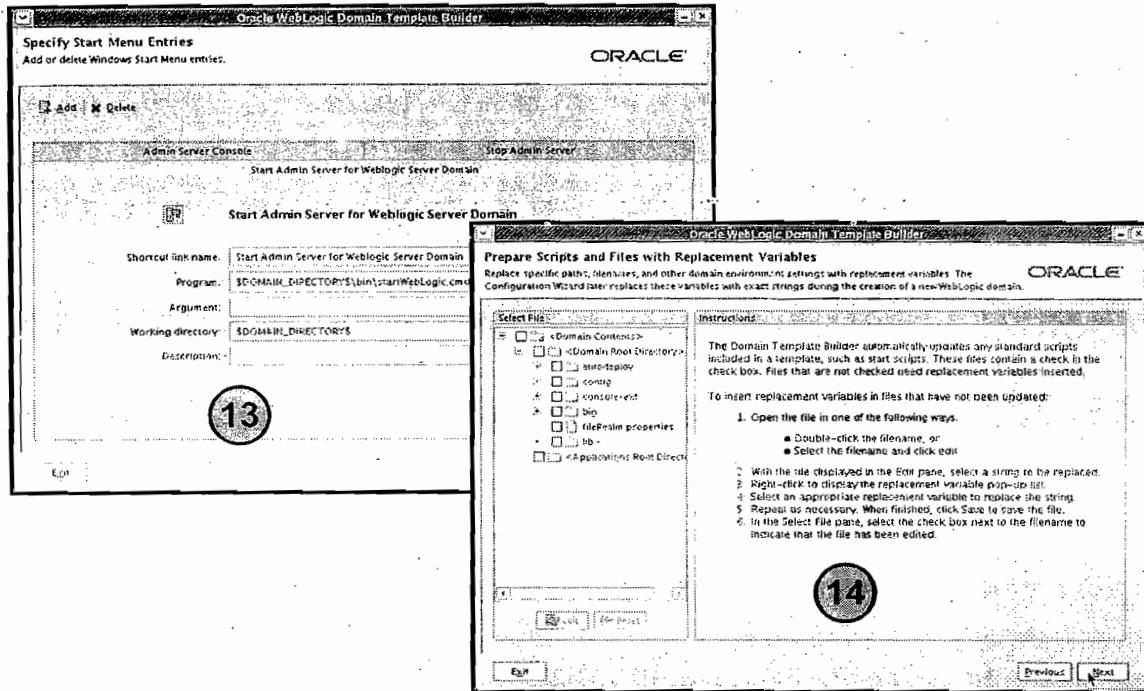
Assigning Users to Groups and Assigning Groups to Groups



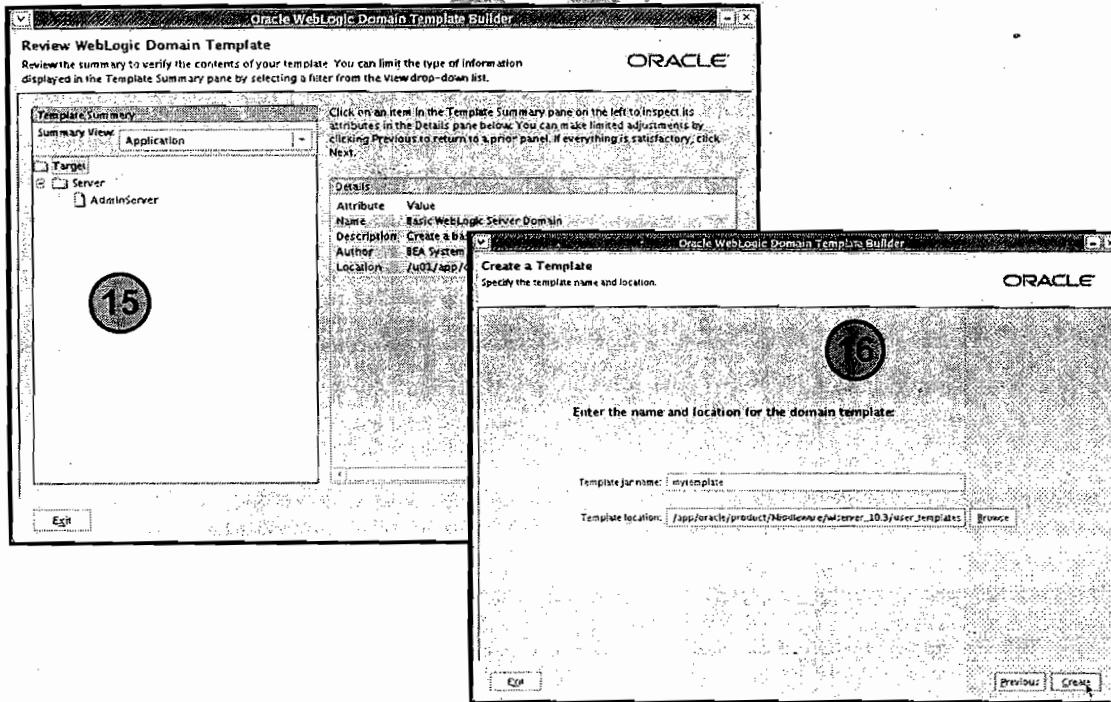
Assigning Users and Groups to Global Roles



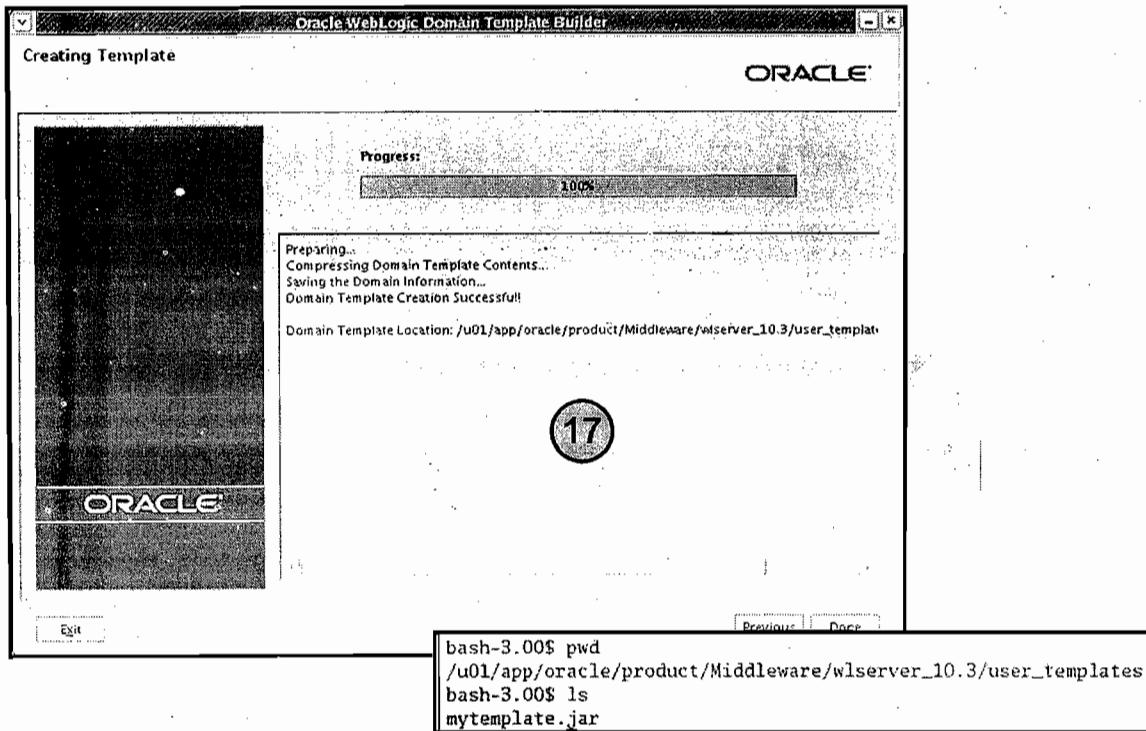
Start Menu Entries for Windows and Prepare Scripts and Files with Replacement Variables



Reviewing the WebLogic Domain Template and Creating a Template



Successful Template Creation



Using the Administration Console

Using the Administration Console, you can:

- Configure attributes of servers and their resources
- Deploy and secure applications
- Configure, collect, and view diagnostic information
- Start and shut down servers or perform other management actions

Accessing the Console

After starting the Administration Server, you can access the console in a browser of your choice.

```
http://[hostname]:[port]/console  
https://[hostname]:[secureport]/console
```

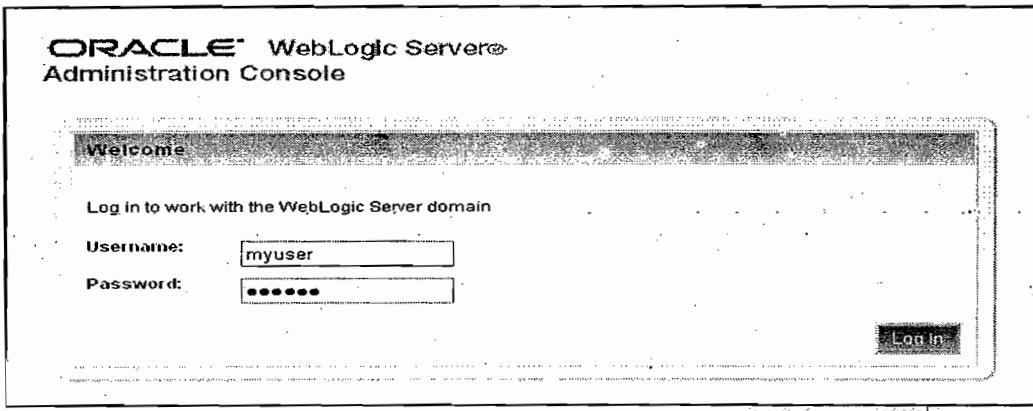
hostname = The name or IP address of the Administration Server
port = The port number that the Administration Server listens on
secureport = The SSL port number that the Administration Server listens on

```
http://[hostname]:[port]/console  
https://[hostname]:[secureport]/console
```

hostname = The name or IP address of the Administration Server
port = The port number that the Administration Server listens on
secureport = The SSL port number that the Administration Server listens on

Console Login

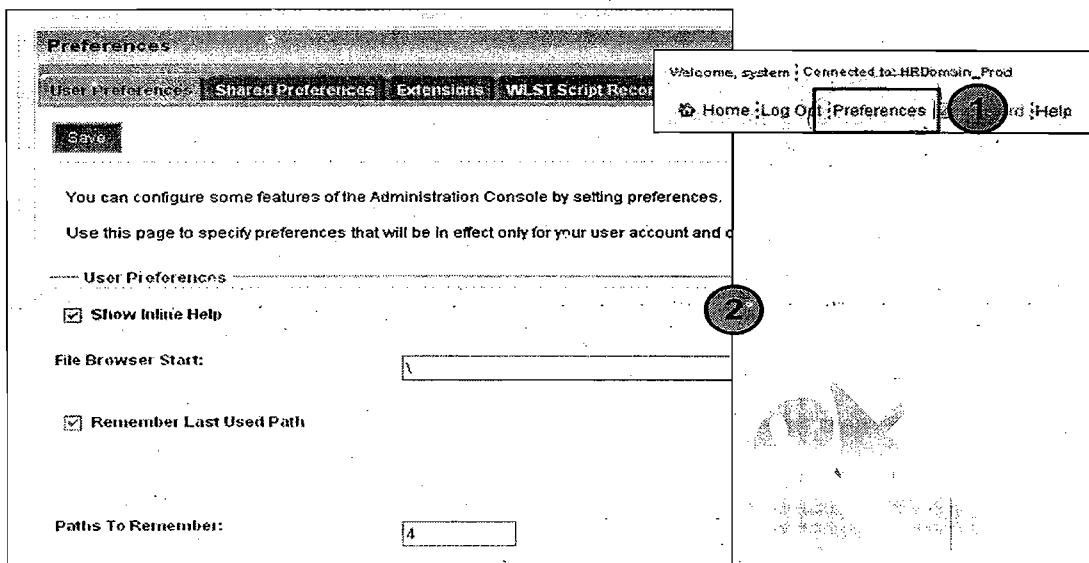
Enter the username and password that you set when creating your domain.



Basic Navigation

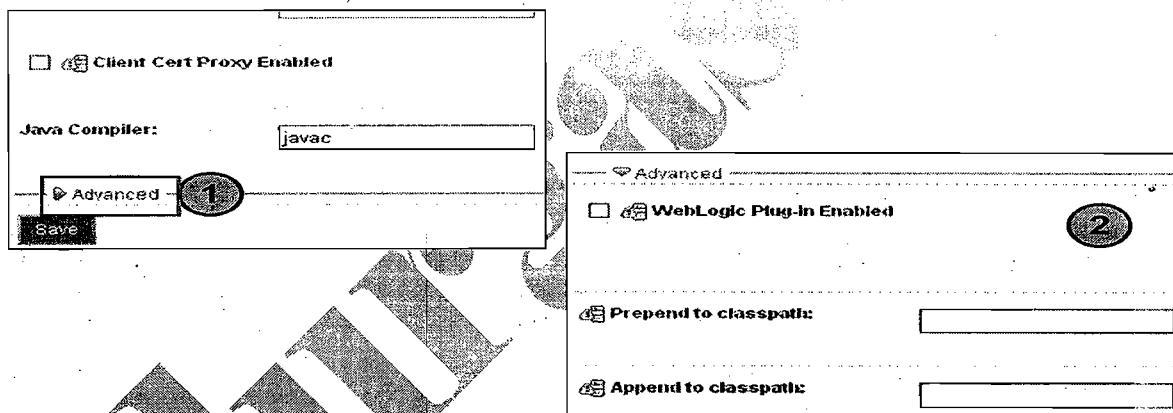
This screenshot displays the Oracle WebLogic Server Administration Console interface. On the left, there's a "Change Center" section with a "Lock & Edit" button and a "Domain Structure" tree view under "lastDomain". The tree includes Environment, Servers, Clusters, Virtual Hosts, Migratable Targets, Machines, Work Managers, Startup & Shutdown Classes, and Deployments. A circled "1" is placed near the "Environment" node. On the right, the main content area shows a "Settings for AdminServer" page. The top navigation bar includes "Home", "Log Out", "Preferences", "Logout", and "Help". Below it is a "Breadcrumb" trail: "Home > Summary of Servers > AdminServer". The main menu tabs are Configuration, Protocols, Logging, Monitoring, Control, and Deployments, with "Protocols" currently selected (circled "2"). Sub-tabs under Protocols include General, HTTP, JMX, IIOP, and Channels. A callout box points to the "HTTP" tab with the text: "Click the Lock & Edit button in the Change Center to modify this server's configuration. Web-based clients communicate with WebLogic Server using HTTP (HyperText Transfer Protocol). Use this page to define the HTTP settings for this server." At the bottom, there's a "Default WebApp Context Root:" input field and a "Returns the original Web application" link.

General Console Preferences



Advanced Console Options

In a screen's *Advanced* section, the console hides the options that are not frequently used.



Setting Basic Properties

Changing the "Standard out" severity threshold:

Settings for AdminServer

Configuration Protocols Logging **Debug** Monitoring Control Deployments Services Security Notes

General HTTP

Message destination(s)

Log file :

Severity level: **Debug**

Filter: **None**

Standard out :

Severity level: **Notice**

Filter: **None**

Domain log broadcaster :

Severity level: **Notice**

Advanced

Minimum severity to log: **Info**

Logger severity properties:

Logging implementation: **JDK**

Redirect stdout logging enabled

Shutting Down a Server

Domain Structure

HRDomain_Prod

- Environment
- Servers
- Clusters
- Virtual Hosts
- Migratable Targets
- Machines
- Work Managers
- Startup & Shutdown Classes
- Deployments

Summary of Servers

Configuration	Control
Start	Resume
Suspend	Shutdown
Restart SSL	Restart SSL
Server: AdminServer(admin)	When work completes: None
	Force Shutdown Now
3	4
Resume	Suspend
Shutdown	Restart SSL

XML Schema for config.xml

- The config.xml file adheres to an XML schema that can be used for validation.

- The config.xml file aggregates configuration information from other configuration files representing Oracle WebLogic Server subsystems, which adhere to their own XML schemas.
- The config.xml file is now located (by default) in the user_projects/domains/domain_name/config directory.
- Subsidiary configuration files are located in subdirectories.

Configuration Change Management

The change management features of WLS:

- Enable you to distribute configuration changes throughout a domain securely, consistently, and predictably
- Are the same, regardless of whether you are using:
 - The WLS Administration Console
 - The WebLogic Scripting Tool (WLST)
 - The Java Management Extension (JMX) APIs

Console Change Center

Pending Changes				
		Type	Description	Restart Required
<input type="checkbox"/>	Enabled	SSL attribute	Modified in ServerA from false to true	
<input type="checkbox"/>	GuardianEnabled	Domain attribute	Modified in HRDomain_Prod from false to true	Yes
<input type="checkbox"/>	ListenPort	Server attribute	Modified in ServerA from 7011 to 7013	
<input type="checkbox"/>	ListenPort	SSL attribute	Modified in ServerA from 7002 to 7014	

Domain Configuration Repository

Directory	Description
config	Root folder of domain configuration files
configCache	Cached configuration data
deployments	Staging area for deployment applications
diagnostics	Configuration modules for diagnostics framework
jdbc	Configuration modules for data services
jms	Configuration modules for messaging services
lib	Not currently used. See <domain>/lib.
nodemanager	Node Manager configuration files
security	Configuration modules for the security framework
startup	Any scripts to run as part of server startup
config.lok	Lock file used for change management
config.xml	Primary domain configuration file

Configuration Change Process

- Domain configuration is represented in two ways:
 - On the file system by a set of XML configuration files, including config.xml
 - At run time by a hierarchy of in-memory JMX objects
- The configuration change management process of Oracle WebLogic Server loosely resembles a database transaction.
- When you activate the changes, it is a two-phase process:
 - Each server determines whether it can accept the change.
 - If all servers are able to accept the change, they update their working configuration hierarchy and the change is completed.

Predictable Distribution of Domain Configuration Changes

- The change management features of WLS enable you to distribute configuration changes throughout a domain securely, consistently, and predictably.
- The change management process is the same, regardless of whether you use:
 - The WLS Administration Console
 - The new WebLogic Scripting Tool
 - JMX
- For change management, use the Change Center in the WLS Administration Console.

Configuration Management Architecture

SCJP EXAM OBJECTIVES

- »> Primitive Fundamentals
- »> Operators and Assignments
- »> Declarations and Access Control
- »> Flow Control
- »> Exception Handling
- »> Assertions
- »> OO Concepts
- »> Threads and Concurrency
- »> Fundamental Classes in `java.lang` package
 - »> Object Class
 - »> String Class
 - »> StringBuffer Class
 - »> StringBuilder Class
 - »> Wrapper Classes
- »> The Collections Frame Work and Generics
- »> File I/O & Serialization
- »> Garbage Collection
- »> 1.5 New Features
 - »> For
 - »> For-Each Loop
 - »> Co-varient return types
 - »> Var-Ag Methods
 - »> AutoBoxing & Un-Boxing
 - »> Static Imports
- »> 1.6 New Features
 - »> NavigableSet
 - »> NavigableMap

SCWCD EXAM OBJECTIVES

Section-1: The Servlet Technology Model

Section-2: The Structure and Deployment of Web Applications

Section-3: The Web Container Model

Section-4: Session Management

Section-5: Web Application Security

Section-6: The JavaServer Pages (JSP) Technology Model

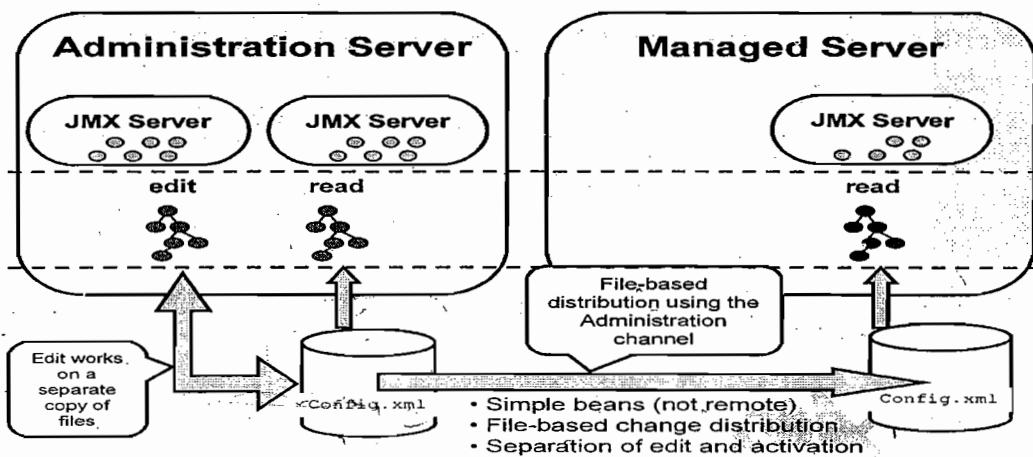
Section-7: Building JSP Pages Using the Expression Language (EL)

Section-8: Building JSP Pages Using Standard Actions

Section-9: Building JSP Pages Using Tag Libraries

Section-10: Building a Custom Tag Library

Section-11: Java EL Patterns



WebLogic Scripting Tool (WLST)

- The WLS command-line tools are useful:
 - For automating common administration activities
 - As an alternative to the Administration Console
 - When graphical tools are not supported
- WLST provides a command-line interface for:
 - Creating new WLS domains
 - Retrieving and updating WLS domain configurations
 - Deploying applications
 - Obtaining run-time server statistics

Jython

Jython is a Java implementation of the popular Python scripting language:

- Simple and clear syntax
- Indentation to structure code
- Interactive command mode
- Custom commands
- Integration with any existing Java libraries

```
list = ['ab','cd','ef']
if len(list) >= 3:
    for x in list:
        print x, len(x)
print 'done'
```

```
from java.util import ArrayList
list = ArrayList()
list.add('ab')
```

Using Jython

Jython can interpret commands in three ways:

- Interactive: Supply commands one at a time from a command prompt. Enter a command in the WLST console and view the response immediately.

- Batch: Provide a series of commands in a script file (.py) when you create a text file with the .py extension that contains a series of WLST commands.
- Embedded: Run the Jython interpreter within a Java class when you instantiate an instance of the WLST interpreter in your Java code and use it to run WLST commands.

WLST Modes

- Online mode:
 - Connected to a running server
 - Access to all WLS configuration and run-time attributes
 - Create and activate change sessions similar to the WLS console
- Offline mode:
 - Domain not running
 - Access to only persisted domain configuration (config.xml)
 - Create or update domains similar to using the Configuration Wizard

WLST Example

```
C:\>java weblogic.WLST
Initializing WebLogic Scripting Tool <WLST> ...
Welcome to WebLogic Server Administration Scripting Shell
Type help() for help on available commands
wls:/offline> connect<'system','weblogic','t3://localhost:7011'>
Connecting to weblogic server instance running at t3://localhost:7011 as usernam
e system ...
Successfully connected to Admin Server 'adminserver' that belongs to domain 'hum
anresources'.
Warning: An insecure protocol was used to connect to the server.
To ensure on-the-wire security, the SSL port or Admin port
should be used instead.
wls:/humanresources/serverConfig> cd<'Servers'>
wls:/humanresources/serverConfig/Servers> ls()
dr-- adminserver,
dr-- mainserver
wls:/humanresources/serverConfig/Servers> cd<'mainserver'>
wls:/humanresources/serverConfig/Servers/mainserver> get<'StartupMode'>
'RUNNING'
wls:/humanresources/serverConfig/Servers/mainserver> exit()
Exiting WebLogic Scripting Tool ...
```

WLST Command Requirements

- Use case-sensitive names and arguments of commands.
- Use arguments enclosed in single or double quotation marks.
- Precede the quoted string with r while specifying the backward slash (\) in the string.
 - Example: readTemplate(r'c:\mytemplate.jar')
- Do not use the following invalid characters in object names when you use WLST offline:
 - Period (.)
 - Forward slash (/)
 - Backward slash (\)
- Use the display help:
 - Example: wls:/mydomain/serverConfig> help('disconnect')

Running WLST Scripts

- Use the setWLSEnv script to initialize the PATH and CLASSPATH required for WLST.
 - If no script file is supplied, WLST runs in interactive mode.
- Use the execfile() command to run additional scripts.

```
>setWLSEnv.sh  
>java weblogic.WLST [scriptfile.py]  
  
To support SSL connection to a server:  
>java -Dweblogic.security.SSL.ignoreHostnameVerification=true  
-Dweblogic.security.TrustKeyStore=DemoTrust weblogic.WLST
```

Importing WLST as a Jython Module

- Invoke WLST:
c:\>java weblogic.WLST
wls:/offline>
- Use the writeIniFile command to convert the WLST definitions and method declarations to a .py file:
wls:/offline> writeIniFile("wl.py")
- Open a new command shell and invoke Jython directly by entering the following command:
c:\>java org.python.util.jython
- Import the WLST module into your Jython module by using the Jython import command:
>>>import wl
- Now you can use the WLST methods in the module. For example, to connect WLST to a server instance:
wl.connect('username','password')

General WLST Commands

help	Get help for a given WLST command.
exit	Quit WLST.
dumpVariables	Display all variables used by WLST.
dumpStack	Display the stack trace for the last error that occurred in WLST.
redirect/stopRedirect	Redirect all WLST output to a file.

Offline WLST Commands

createDomain	Create a domain by using a given template.
readDomain	Open an existing domain on the file system.
readTemplate	Open an existing domain template.
addTemplate	Apply a template file to the current domain.
updateDomain	Save changes to the current domain.
writeDomain	Save changes to the current domain to a specified directory.
writeTemplate	Save the current domain to a template file.
assign/unassign	Target applications or services to servers.
setOption	Configure domain creation options (domain name, Java home, start mode, and so on).

Creating a Domain: Example.

```

▶ readTemplate('mybasetemplate.jar')

▶ setOption('DomainName', 'mydomain')
  setOption('JavaHome', 'c:/myjdk')
  setOption('ServerStartMode', 'prod')

▶ writeDomain('c:/mydomains')
  closeTemplate()

▶ readDomain('c:/mydomains/mydomain')
  addTemplate('myjms.jar')
  addTemplate('myapps.jar')
  updateDomain()
  closeDomain()
  exit()

```

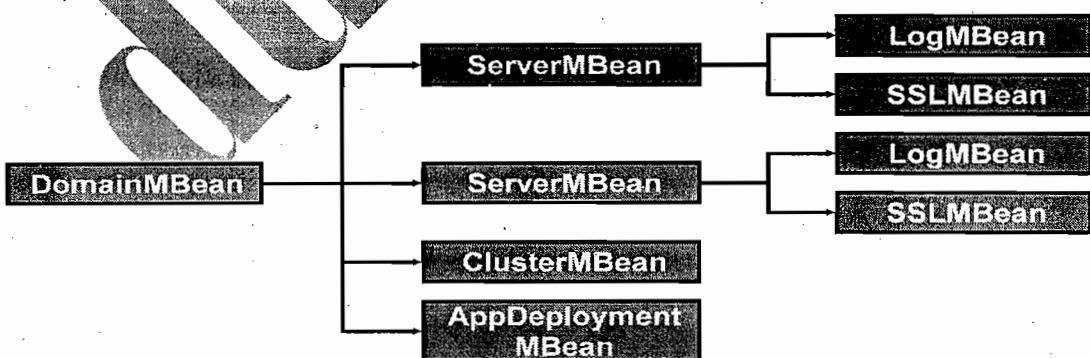
Online WLST Commands

connect	Connect to a server by using supplied credentials.
disconnect	Disconnect from the current server.
shutdown	Shut down servers.
start	Use the Node Manager to start servers.
startEdit	Begin a new change session.
stopEdit	Release the edit lock and discard any changes.
activate	Commit all changes in the current session.
showChanges	List all changes made in the current session.
isRestartRequired	Determine if any changes require a server restart.
deploy/redeploy	Deploy an application to servers.
undeploy	Shut down a running application on servers.

WebLogic JMX: Overview

JMX MBeans:

- Are Java objects found on the server
- Have attributes and operations
- Support the configuration, management, and monitoring of all types of server resources



Navigating JMX MBeans

- Use the cd, ls, and pwd commands to navigate the server configuration or run-time MBeans, similar to a file system.
- Use the get or set commands to read or update the MBean attributes.

The cmo variable refers to the current MBean.

```
>connect('myuser', 'mypass', 'localhost:7001')
>cd('Servers')
>ls()
dr-- AdminServer
dr-- ServerA
>cd('ServerA')
>ls()
dr-- Log
dr-- SSL
-r- ListenPort 7011
-r- StartupMode RUNNING
>cd('Log/ServerA/StdoutFilter')
```

Creating a Server: Example

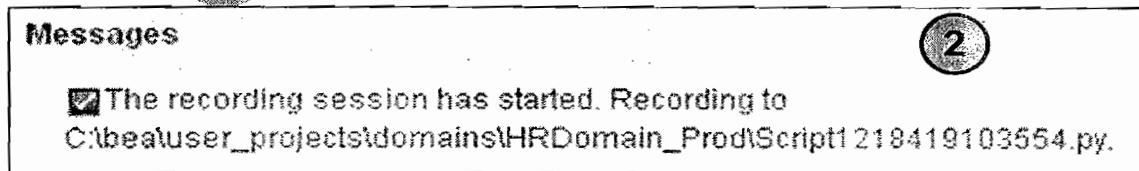
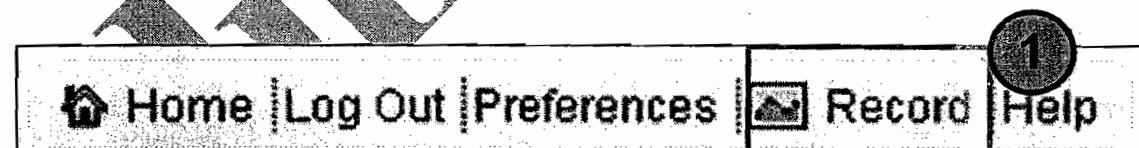
```
>servers = {'serverB':7021, 'serverC':7031}

>connect('myuser', 'mypass', 'localhost:7001')
edit()
startEdit()

for name, port in servers.items():
    cd('/Servers')
    create(name, 'Server')
    cd(name)
    set('ListenPort', port)

save()
activate()
disconnect()
exit()
```

Generating a WLST Script



Quiz

Which link would you click in the Console to add or remove columns to or from a monitoring page?

1. Customize this table
2. Change Monitoring View

3. Update Columns
4. Manage Preferences

Answer: 1

Which directory within a domain is used to maintain its configuration repository?

1. /Console
2. /cache
3. /config
4. /logs
5. /AdminServer

Answer: 3

The _____ panel in the Administration Console uses a tree to represent your domain resources.

1. Hierarchy
2. Preferences ?
3. Domain Structure
4. How do I ?
5. Change Center

Answer: 3

WLST communicates with Oracle WebLogic Server's _____ to retrieve and update resources on a running server.

1. Templates
2. Logs
3. MBeans
4. Scripts

Answer: 3

By using WLST's _____ mode, you can supply commands one at a time and get immediate feedback.

1. Management
2. Operational
3. Sequential
4. Template
5. Interactive

Answer: 5

Invoke the Domain Configuration Wizard by using _____:

1. config.sh under WL_HOME/common/bin
2. config_builder.sh under WL_HOME/common/bin

Answer: 1

When you create a new domain to automatically support Oracle WebLogic Server, which template is it based on by default?

1. wlst.jar
2. ws.jar

3. web_server.jar
4. server.jar
5. wls.jar

Answer: 5

In the Domain Configuration Wizard, you can configure server start mode as Development or Production. In this context, choose which of the following is NOT true?

1. The "Lock and Edit" option is not available in the Administration Console when you select the Development mode.
2. Sun SDK is generally recommended in Development mode and JRockit SDK in Production mode.
3. When you select the Production mode, a boot.properties file is created.
4. In Production mode, Oracle WebLogic Server does not poll for applications to deploy.

Answer: 3

Where are all users, groups, and roles stored by default?

1. Oracle Database
2. PointBase Database
3. Oracle Internet Directory
4. Administration Server LDAP Store

Answer: 4

What is the main configuration file for the domain called?

1. configuration.xml
2. wlsconfig.xml
3. wls.xml
4. config.xml

Answer: 4

Under the servers directory of WLS domain, there are subdirectories for Administration and Managed Servers. The servers directory contains one subdirectory for each WebLogic Server instance in the domain. If you do not see the subdirectory for each WebLogic Server instance in your domain, it means:

1. The WebLogic Server instance is not correctly configured.
2. The domain is unavailable.
3. The Administration Server is unable to communicate with the Managed Servers.
4. The WebLogic Server instance has not been started since it was created.

Answer: 4

To start the Administration Server by using java weblogic.Server, you must specify the –Dweblogic.management.server=url-for-Admin-Server option.

1. True
2. False

Answer: 2

Running startWebLogic.sh does which of the following:

1. Invokes java weblogic.Server
2. Starts the Managed Servers associated with the Administration Server
3. Sets the environment by using setDomainEnv.sh
4. Starts the Administration Server

Answer: 1, 3, 4

Which of the following options would you use to create a Managed Server?

1. Domain Configuration Wizard
2. Administration Console
3. Command Line (WLST)

Answer: 1, 2, 3

Both the Administration and Managed Servers can be started by using the Administration Console.

1. True
2. False

Answer: 2

You can use boot identity files to start the following without being prompted for the admin username and password.

1. Managed Servers
2. Administration Server
3. Both

Answer: 3

Which of the following is true when the Administration Server is down?

1. Domain log entries are unavailable.
2. Managed Servers can start in MSI mode.
3. The Administration Console and management tools are unavailable.
4. At boot time, Managed Servers read a local copy of config.xml, SerializedSystemIni.dat, and boot.properties (optional).
5. You cannot change the configuration of the Managed Servers that are running in MSI mode until communication with the Administration Server is restored.
6. The Node Manager can start the Managed Servers in MSI mode.

Answer: 1, 2, 3, 4, 5

Which of the following can you use to create a domain template:

1. The pack command
2. config_temp.sh
3. The Administration Console
4. config_builder.sh
5. The WLST command-line tool

Answer: 1, 4, 5

Summary

In this lesson, you should have learned to:

- Describe how the domain works
- Describe the domain directory structure
- Configure a domain
- Start or stop the Oracle WebLogic Server
- Configure Managed Servers
- Start Managed Servers
- Describe administration and Managed Server Independence
- Create a custom domain template
- Perform Console administration
- Perform command-line administration



Managing and Monitoring the Oracle WebLogic Server Environment

Objectives

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

- Describe machines and Node Manager
- Describe simple logging
- Use commands to get attributes from an MBean
- Use the network channels to add flexibility to your networking configuration

Road Map

- Remote Administration
 - Configuring Machines
 - Node Manager
 - Configuring Node Manager
- Logs and Monitoring
- Network Channels

Node Manager

Node Manager (NM):

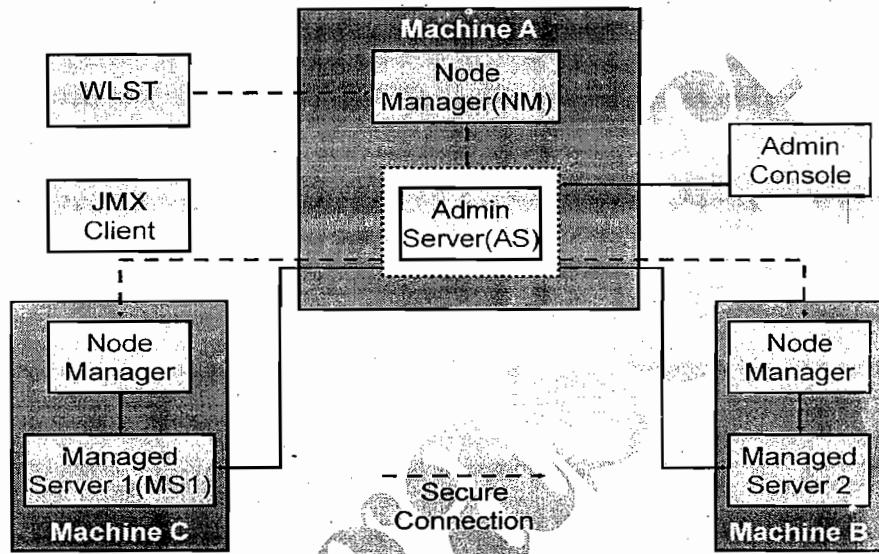
- Starts and stops Managed Servers remotely: server, domain, and cluster
- Available as either a Java-based or (for UNIX or Linux) a script-based process
- Monitors and acts on server health
- Runs on the same computers as the Managed Servers
- Can be run automatically in the background, as a Windows service or a UNIX daemon

What Node Manager Can Do

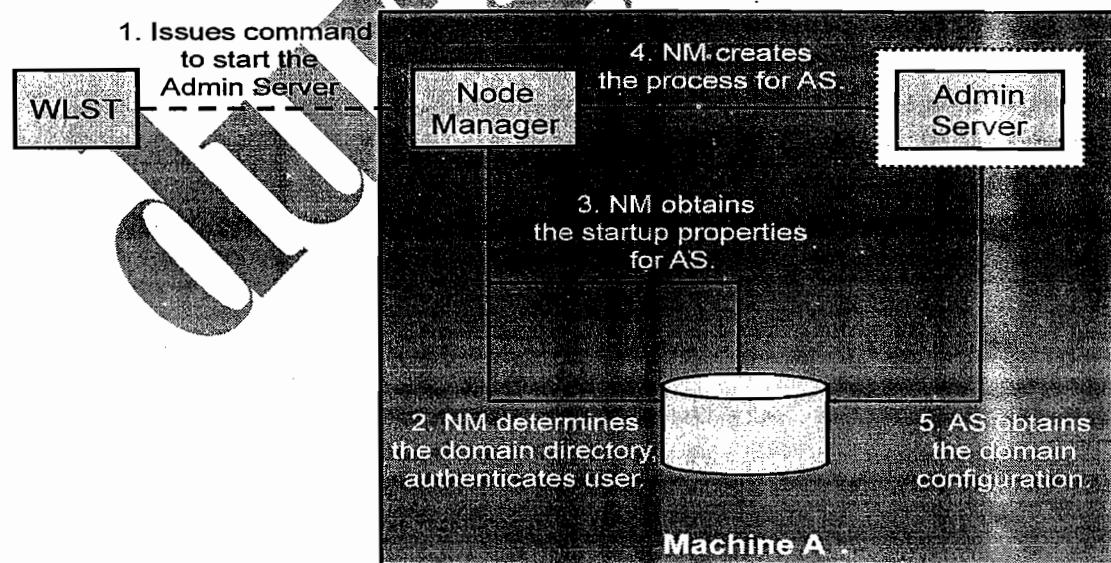
You can use Node Manager to:

- Start, shut down, and restart an Administration Server
- Start, shut down, suspend, and restart Managed Servers
- Automatically restart the Administration and Managed Servers on failure
- Monitor servers and collect log data

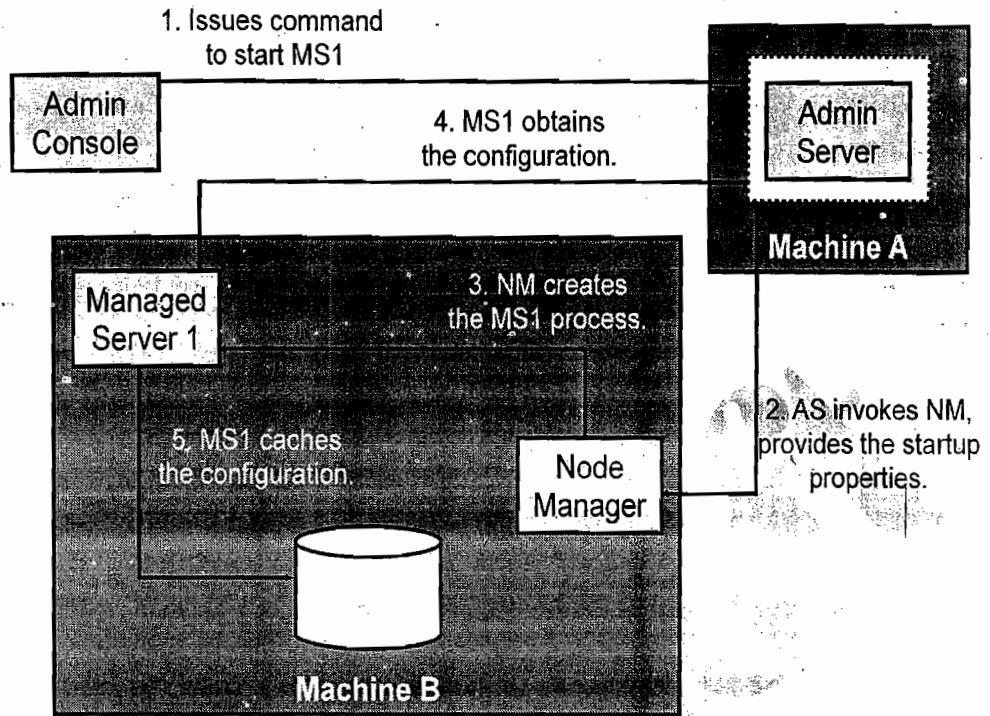
Node Manager Architecture



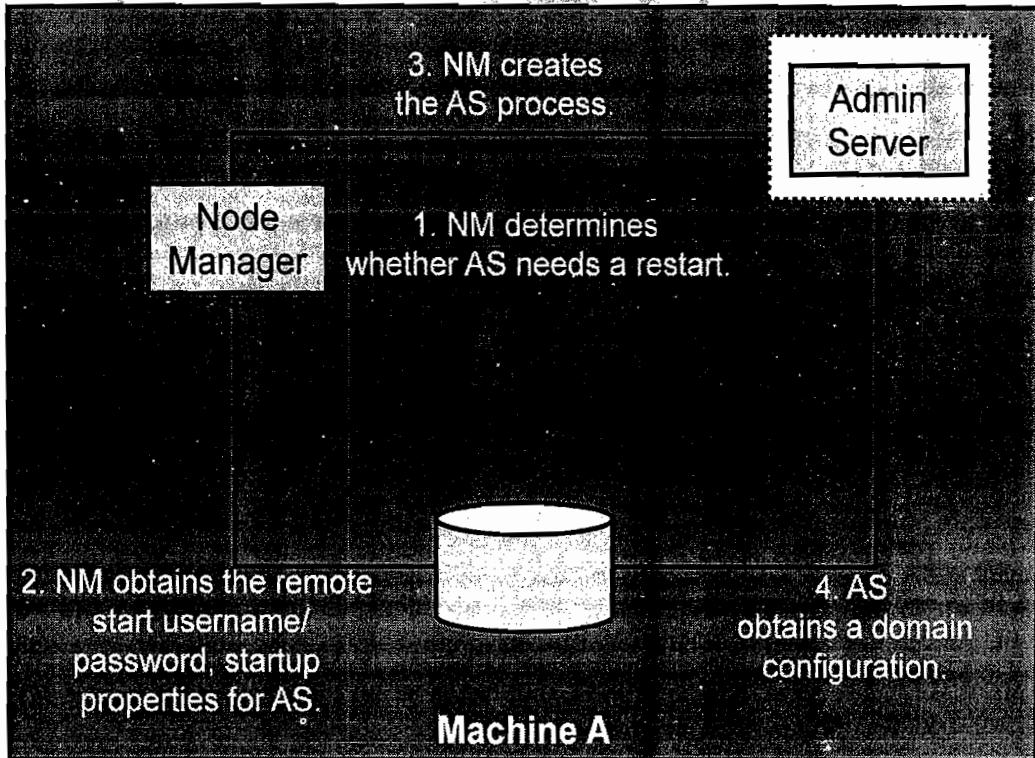
How Node Manager Starts an Administration Server



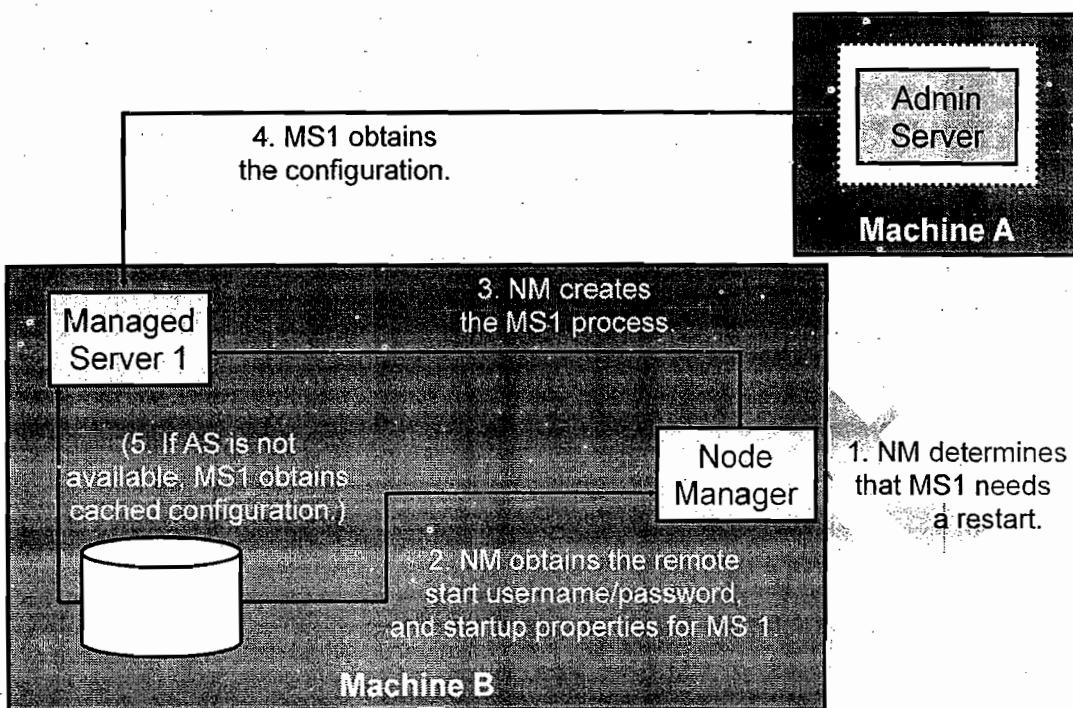
How Node Manager Starts a Managed Server



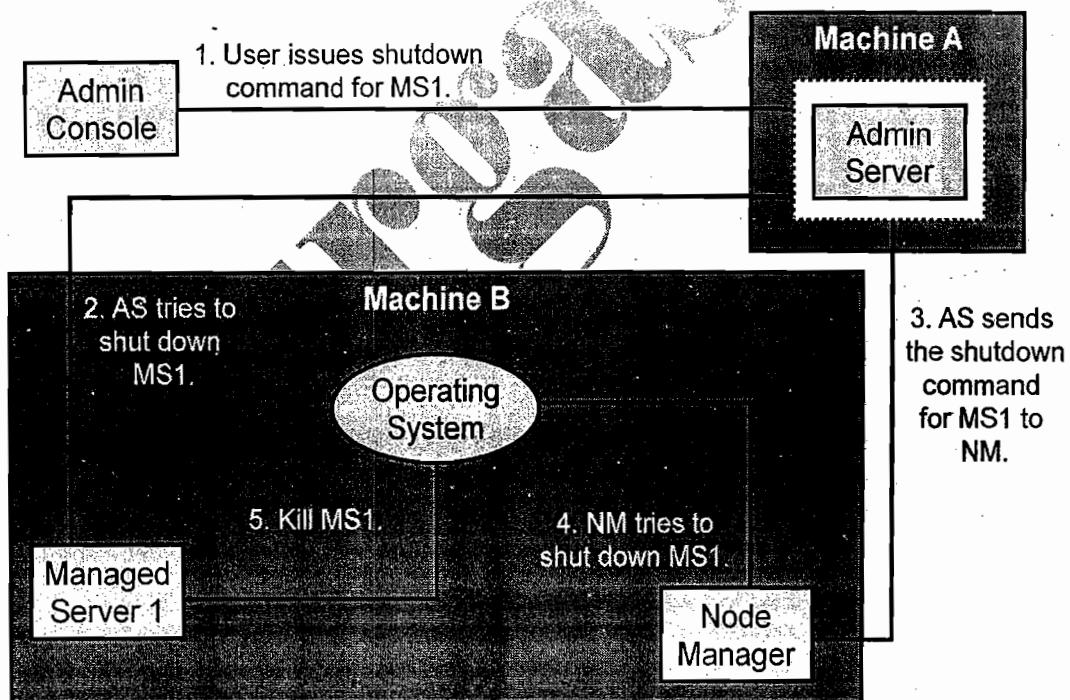
How Node Manager Restarts an Administration Server



How Node Manager Restarts a Managed Server



How Node Manager Shuts Down a Server Instance



Versions of Node Manager

- There are two versions of Node Manager:
 - Java-based Node Manager
 - Script-based Node Manager

- Java-based Node Manager runs within a Java Virtual Machine (JVM) process.
- Script-based Node Manager (used only for UNIX and Linux systems) does not have as much security, but provides the ability to remotely manage servers over a network using Secure Shell (SSH).

Node Manager Configuration

- Node Manager must run on each computer that hosts the WLS instances that you want to control with Node Manager.
- You should configure each computer as a machine in Oracle WebLogic Server, and assign each server instance, which is to be controlled by Node Manager, to the machine that the server instance runs on.
- Node Manager should run as an operating system service, so that it automatically restarts upon system failure or reboot.

Node Manager Default Behaviors

- After Oracle WebLogic Server is installed, Node Manager is “ready-to-run” if Node Manager and Administration Server are on the same machine.
- By default, the following behaviors are configured:
 - The Administration Console can use Node Manager to start the Managed Servers.
 - Node Manager monitors the Managed Servers that it started.
 - The automatic restart of Managed Servers is enabled.

Configuring Java-Based Node Manager

- It is recommended that you configure Node Manager to run as an operating system service.
- The configuration tasks for Java-based Node Manager include:
 - Reconfiguring the startup service for a Windows installation
 - Daemonizing Node Manager for UNIX systems
 - Configuring the Java-based Node Manager security
 - Reviewing nodemanager.properties
 - Configuring Node Manager on multiple machines

Starting Node Manager at System Startup

- It is recommended that you run Node Manager (NM) as:
 - A Windows service on Windows platforms and
 - A daemon on UNIX platforms
- Running NM during system startup allows it to restart automatically when the system is rebooted.
- Node Manager can be configured to start at boot time, as either of these:
 - A Windows service

- A UNIX daemon

Configuring Node Manager as a Windows Service

- Delete the Node Manager service using `uninstallNodeMgrSvc.cmd`.
- Edit `installNodeMgrSvc.cmd` to specify Node Manager's listen address and listen port.
- Run `installNodeMgrSvc.cmd` to reinstall Node Manager as a service, listening on the updated address and port.

Daemonizing Node Manager for UNIX Systems

- When configuring Node Manager to run as an `inetd` or `xinetd` service, the following considerations apply:
 - Ensure that `NodeManagerHome` and other system properties are defined.
 - If `xinetd` is configured with `libwrap`, you should add the `NOLIBWRAP` flag.
 - Ensure that the `hosts.deny` and `hosts.allow` files are configured correctly.
 - Depending on your network environment, additional configuration may be necessary.

Starting Oracle WebLogic Server at Boot

- Oracle WebLogic Server (WLS) can be configured to start at boot time, as either of these:
 - A Windows service
 - A UNIX daemon

Oracle WebLogic Server as a Windows Service

- The Configuration Wizard can register a domain's admin server as a service.
- To configure WebLogic Server as a Windows service with custom settings, you must edit and run this script:
 - `%WL_HOME%\server\bin\installService.cmd`
- The script sets up the admin server as a Windows service.

Setting Up Windows Service

- The `beasvc.exe` utility supports installing Java programs (for example, WLS) as Windows Services.
- Syntax:
 - `%WL_HOME%\server\bin\beasvc.exe -install -svcname displayName - javahome jdk1.6Home -execdir wlsInstallRoot -extrapath wlsBinDirectory -cmdline cmdToRun`
 - `displayName` The name to associate with the service
 - `jdk1.6Home` The path to where the JDK 1.6 is installed
 - `wlsInstallRoot` The root of the WLS install tree (`c:\bea\weblogic103`)
 - `wlsBinDirectory` The path to the WLS \bin directory

- cmdToRun The Java class to run as a service (weblogic.Server), including parameters

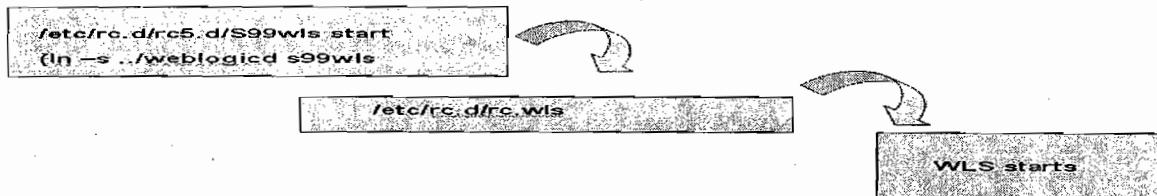
Oracle WebLogic Server as a UNIX Daemon

- The UNIX daemon is configured using an rc script.
- The rc script and the init process describe:
 - What is run, and when
 - How the init process reads /etc/inittab to determine the run level to enter
 - Which rc scripts are run as the machine enters and exits each run level

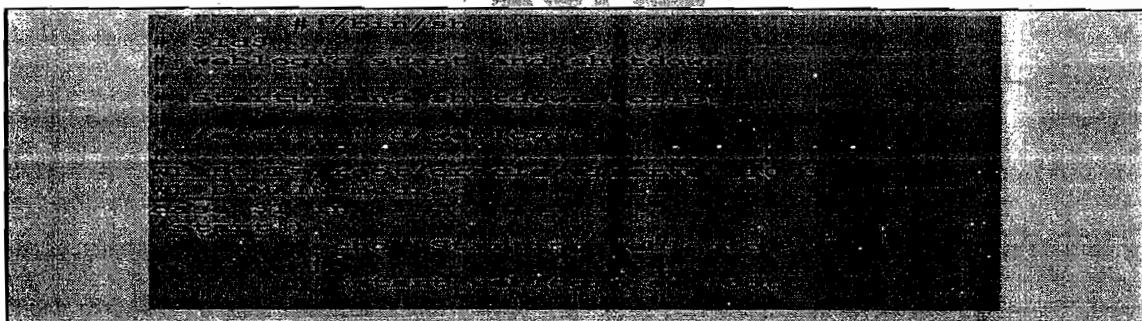
Oracle WebLogic Server at Boot

You can configure Oracle WebLogic Server using two files for boot control:

- /etc/rc.d/init.d/weblogicd
- /etc/rc.d/rc.wls



Example rc Scripts



Example rc Scripts

```

#
(cd ${WL_HOME}/samples/domains/${WL_DOMAIN}/bin;
./startWebLogic.sh >${WL_HOME}/logs/boot.log
2>&1)
{
  'stop')
echo "Stopping weblogic."
(cd ${WL_HOME}/samples/domains/${WL_DOMAIN}/bin;
./stopWebLogic.sh >${WL_HOME}/logs/stop.log 2>&1)
}
echo "Usage: $0 {start|stop}"
exit 1
;;
esac
  
```

Configuring Java-Based Node Manager Security

- Node Manager security relies on a one-way SSL connection between the client and server.
- WLST uses the nmConnect command to establish a connection to the Java Node Manager.
- The nmConnect command requires a username and password, which is verified against the nm_password.properties file.

Administration Console Node Manager Security

Settings for dizzyworld

Configuration Monitoring Control Security Web Service Security Notes

General Filter Unlock User Embedded LDAP Roles Policies

Default Realm: myrealm

Anonymous Admin Lookup Enabled

Cross Domain Security Enabled

Excluded Domain Names:

Security Interoperability Mode: default

Credential:

Confirm Credential:

NodeManager Username: oXe2hEc0tw

NodeManager Password:

Remote Server Start Security for the Java-Based Node Manager

The credentials for Managed Servers and the Administration Server are handled differently.

- Managed Servers – When you invoke Node Manager to start a Managed Server, it gets its remote username and password from the Administration Server.

- Administration Servers – When you invoke Node Manager to start an Administration Server, the remote start username comes from either the command-line or the boot.properties file.

Reviewing nodemanager.properties

- You can specify the properties for a Java-based Node Manager process either at the command line or in the nodemanager.properties file.
- Values supplied on the command line take precedence over those in the nodemanager.properties file.
- To configure Node Manager to use a start script, in the nodemanager.properties file:
 - Set the StartScriptEnabled property to true.
 - Set the StartScriptName property to the name of your script.

Configuring Node Manager on Multiple Machines

- Node Manager must be installed and configured on each machine on which there is a Managed Server.
- You can use the WLST nmEnroll command to copy the required domain and configuration information from one machine to another.
- nmEnroll([domainDir], [nmHome])
- This command downloads the following files from the Administration Server:
 - nm_password.properties, which contains the encrypted username and password that is used for server authentication
 - The SerializedSystemIni.dat file

Configuring Script-Based Node Manager

- The SSH Node Manager is a shell script, wlscontrol.sh, located in NM_HOME/common/bin.
- An executable SSH client must reside on each machine where Node Manager or the Node Manager client runs.
 - An SSH client is typically a standard part of a UNIX or Linux installation.
- The configuration tasks for a script-based Node Manager include:
 - Using SSH with the script-based Node Manager
 - Creating a Node Manager user
 - Configuring the script-based Node Manager security

Creating a Node Manager User

- Before you run Node Manager, you should create a dedicated UNIX user account for performing the Node Manager functions.

- You should add this user to all machines that will host the SSH Node Manager and to all machines that will host a Node Manager client, including the Administration Server.

Configuring Script-Based Node Manager Security

- The Node Manager SSH shell script relies on SSH user-based security to provide a secure trust relationship between users on different machines.
- Authentication is not required.
- You create a UNIX user account—typically one per domain—for running the Node Manager commands and scripts.
- A user logged in as this user can issue Node Manager commands without providing a username and password.

Additional Configuration Information

Other Node Manager configuration tasks include:

- Configuring a machine to use Node Manager
- Configuring the nodemanager.domains file
- Configuring the remote startup arguments
- Ensuring that the Administration Server address is defined
- Setting Node Manager environment variables

Configuring a Machine to Use Node Manager

A WLS machine resource maps a machine with the server instances that it hosts.

Settings for dizzyMachine1

Configuration **Monitoring** **Notes**

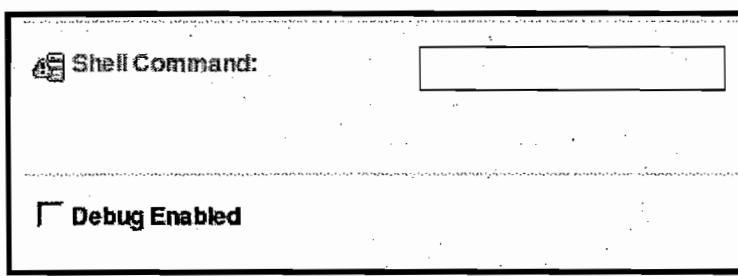
General **Node Manager** **Servers**

Save

This page allows you to define the Node Manager configuration for the console. Node Manager must be configured and running on the machine.

The settings defined on this page are used to configure communication with instances that control Managed Servers. This page does not control the managed servers.

Type:	SSL
Listen Address:	localhost
Listen Port:	5555
Node Manager Home:	[Empty Input Field]



Configuring the nodemanager.domains File

- The nodemanager.domains file specifies the domains that a Node Manager instance controls.
- When a user issues a command for a domain, Node Manager looks up the domain directory from this file.
- nodemanager.domains provides additional security by restricting Node Manager client access to the domains listed in this file.

Configuring Remote Startup Arguments

Settings for dizzy1

Configuration | Protocols | Logging | Debug | Monitoring | Control | Deployments | Services | Security | Notes

General | Clusters | Services | Keystores | SSL | Federation Services | Deployment | Migration | Tracing
Overload | Health Monitoring | Server Start |

Save

Node Manager is a WebLogic Server utility that you can use to start, suspend, shut down, and restart servers in normal or unexpected conditions. Use this page to configure the startup settings that Node Manager will use to start this server on a remote machine.

Java Home: <input type="text"/>	The Java home directory (path on the machine running Node Manager to use when starting this server). More Info...
Java Vendor: <input type="text"/>	The Java Vendor value to use when starting this server. For example, BEA, Sun, IBM etc. More Info...
BEA Home: <input type="text"/>	The BEA home directory (path on the machine running Node Manager) to use when starting this server.
Root Directory: <input type="text"/>	
Class Path: <code>/u01/app/oracle/product/Middleware/patch_xia1030/profilers/default</code>	
Arguments: <input type="text"/>	
Security Policy: <input type="text"/>	
User Name: <input type="text"/>	
Password: <input type="text"/>	
Confirm Password: <input type="text"/>	

Ensuring That the Administration Server Address Is Defined

You must define a listen address for each Administration Server that connects to the Node Manager process.

Settings for dizzy1

Configuration | Protocols | Logging | Debug | Monitoring | Control | Deployments | Services | Security | Notes

General | Cluster | Services | Keystores | SSL | Federation Services | Deployment | Migration | Tuning

Overload | Health Monitoring | Server Start

Save

Use this page to configure general features of this server such as default network communications.

Name: dizzy1 An alphanumeric name for this server instance. [More Info...](#)

Machine: dizzyMachine1 The WebLogic Server host computer (machine) on which this server is meant to run. [More Info...](#)

Cluster: (Stand-Alone) The cluster, or group of WebLogic Server instances, to which this server belongs. [More Info...](#)

Listen Address: The IP address or DNS name this server uses to listen for incoming connections. [More Info...](#)

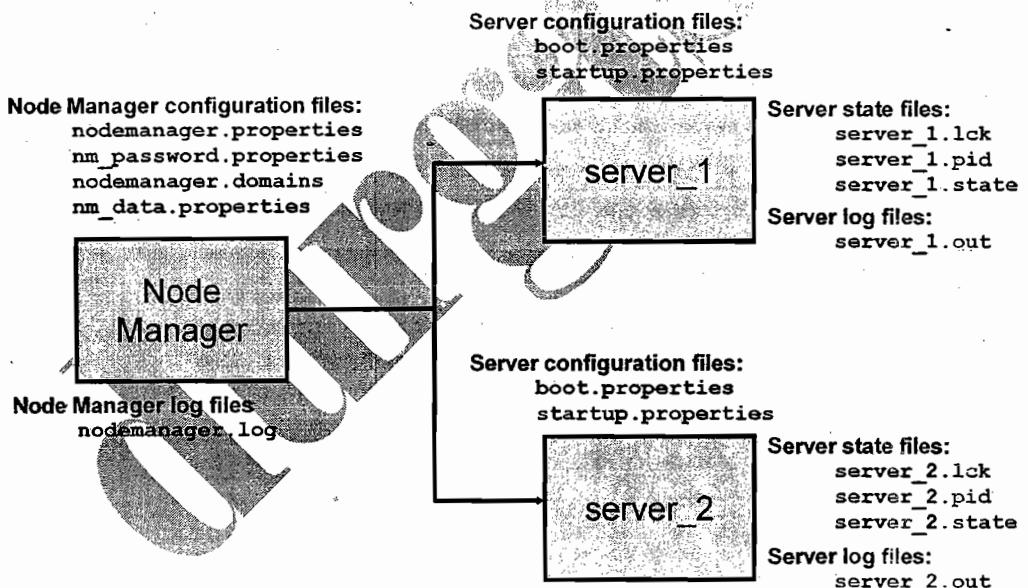
Listen Port Enabled Specifies whether this server can be reached

Setting Node Manager Environment Variables

Environment Variable	Description
JAVA_HOME	Root directory of the JDK that you are using for Node Manager. For example: set JAVA_HOME=c:\oracle\jdk1.6.0_05 Node Manager has the same JDK version requirements as Oracle WebLogic Server.
WL_HOME	Oracle WebLogic Server installation directory. For example: set WL_HOME=c:\oracle\wlserver_10.3
PATH	Must include the Oracle WebLogic Server bin directory and path to your Java executable. For example: set PATH=%WL_HOME%\server\bin;%JAVA_HOME%\bin;%PATH%

LD_LIBRARY_PATH (UNIX only)	<p>For HP-UX and Solaris systems, you must include the path to the native Node Manager libraries.</p> <p>Solaris example:</p> <pre>LD_LIBRARY_PATH:\$WL_HOME/server/native/solaris:\$WL_HOME/server/lib/solaris/oci816_8</pre> <p>HP-UX example:</p> <pre>SHLIB_PATH=\$SHLIB_PATH:\$WL_HOME/server/native/hpux11:\$WL_HOME/server/lib/hpux11/oci816_8</pre>
CLASSPATH	<p>You can set the Node Manager CLASSPATH either as an option on the Java command line that is used to start Node Manager or as an environment variable.</p> <p>Windows example:</p> <pre>set CLASSPATH=.;%WL_HOME%\server\lib\weblogic_sp.jar;%WL_HOME%\server\lib\weblogic.jar</pre>

Node Manager Configuration and Log Files



Oracle WebLogic Server Logs

Logs can aid in the discovery of:

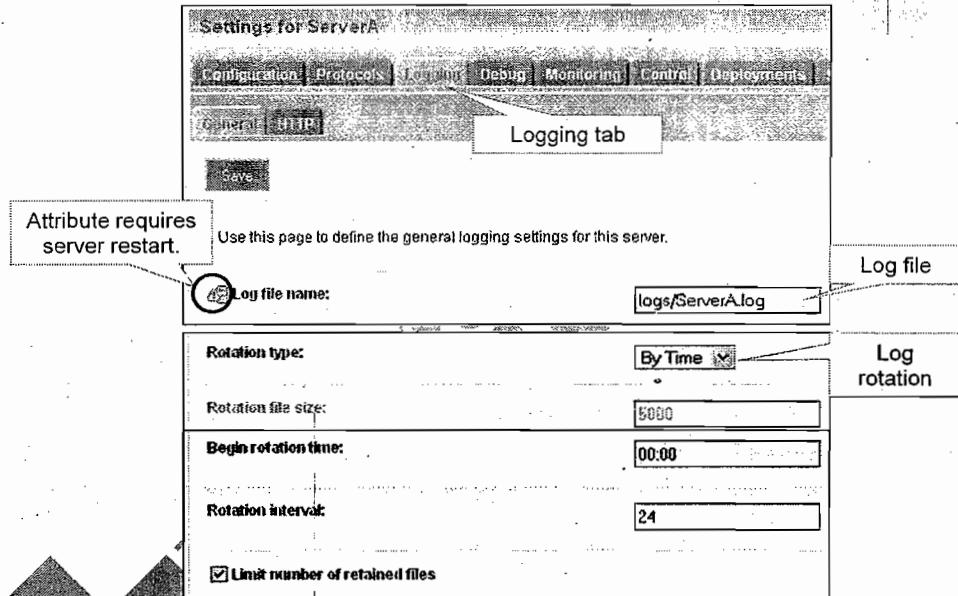
- Any problems encountered while servicing requests
- Activity by day and time interval
- The IP addresses of users accessing an application
- Frequently accessed resources

- The amount of data sent and received
- Performance statistics

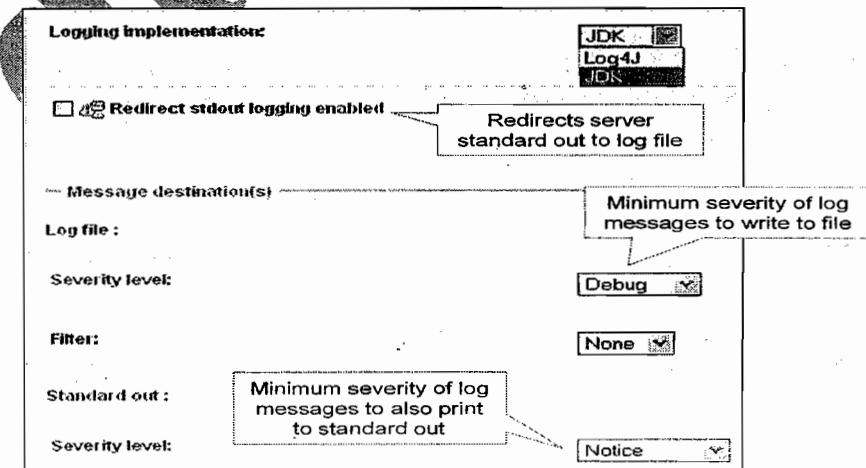
Server and Domain Logs

- A server log:
 - Logs all activity for a single server
 - Is stored in <serverName>/logs/<serverName>.log by default
- A domain log:
 - Logs key events for all servers in a domain
 - Is stored in <AdminServer>/logs/<domainName>.log by default
- These logs are independently configured.

Configuring Server Logging



Configuring Server Logging



Viewing Logs with the Console

The screenshot shows the Oracle WebLogic Server Administration Console. On the left, the 'Domain Structure' tree view is expanded to show 'Log Files'. A circled '1' is next to the 'Log Files' node. In the center, a 'Server Log' table displays two log entries. A circled '2' is next to the 'View' button at the bottom of the table. Below the table, a message says 'Showing 1 to 9 of 9 Previous | Next'. A circled '3' is next to the third log entry in the table.

ServerLog	Server Log	AdminServer
ServerLog	Server Log	serverA

Showing 1 to 9 of 9 Previous | Next

Server Log Entries(FILTERED - More Columns Exist)

Date	Subsystem	Severity	Message ID	Message
Jul 31, 2008 1:46:09 PM EDT	Server	Info	BEA-002622	The protocol "snmp" is now configured.
Jul 31, 2008 1:46:09 PM EDT	Server	Info	BEA-002622	The protocol "admin" is now configured.

Previous | Next

Message Attributes

Attribute	Description	Supported Outbound
Timestamp	The time and date when the message originated, in a format that is specific to the locale	X
Severity	The degree of impact or seriousness of the event reported by the message	X
Subsystem	The particular WLS subsystem that was the source of the message (Management, Security, EJB, RMI, JMS, and so on)	X
Catalog ID	The unique ID assigned to this type of event, to reference in the online documentation	X
Server Name	The WebLogic instance that generated the message	
Thread ID	The server thread that generated the message	
User ID	The current security context, if any	
Transaction ID	The current XA transaction context, if any	

Message Severity

Message Severity		Log Level
TRACE	Messages from the diagnostics framework	
DEBUG	Detailed internal messages (if debugging is enabled)	
INFO	Normal operations	
NOTICE	INFO message of greater importance	X
WARNING	Suspicious operation or configuration	X
ERROR	Error handling request, but no interruption in service	X
CRITICAL	System or service error that may cause temporary loss or degradation of service	X
ALERT	One or more services in an unusable state, requiring administrative attention	X
EMERGENCY	Entire server in an unusable state	X

Message Catalog

Use the WLS online message catalogs to obtain more information about a specific log message ID.

Index Of Messages By Message Range		
Messages In the Message Catalog are part of the WebLogic Server Internationalization and Localization packages.		
Range	Subsystem	Catalog
BEA-000100 - BEA-000199	Cluster	Cluster
BEA-000200 - BEA-000399	WebLogicServer	T3Srvr
BEA-000400 - BEA-000499	Socket	Socket
000500 - 000599	RJVM	RJVM
BEA-000600 - BEA-000699	Common	Common
BEA-000700 - BEA-000799	T3Misc	T3Misc
BEA-000800 - BEA-000899	Kernel	Kernel
BEA-000900 - BEA-000999	Net	Net
BEA-001000 - BEA-001999	JDBC	JDBC

```

[Start Admin Server for Weblogic Server Domain: C:\bea10\user_projects\domains\base_domain\bin]
up for pool "cgDataSource". Connections will not be tested. >
<Apr 19, 2007 12:19:53 PM EDT> <Warning> <JDBC> <BEA-001110> <No test table set
up for pool "cgDataSource-nonXA". Connections will not be tested.>
<Apr 19, 2007 12:19:53 PM EDT> <Warning> <JDBC> <BEA-001110> <No test table set
up for pool "samplesDataSource". Connections will not be tested.>
<Apr 19, 2007 12:20:07 PM EDT> <Notice> <Log Management> <BEA-170027> <The server
initialized the domain log broadcaster successfully. Log messages will now be
broadcasted to the domain log.>
<Apr 19, 2007 12:20:08 PM EDT> <Notice> <WebLogicServer> <BEA-000365> <Server st
ate changed to ADMIN>
<Apr 19, 2007 12:20:08 PM EDT> <Notice> <WebLogicServer> <BEA-000365> <Server st
ate changed to RESURING>
<Apr 19, 2007 12:20:08 PM EDT> <Notice> <Server> <BEA-002613> <Channel "Default"
is now listening on 192.168.1.2:7001 for protocols https, t3, ldap, snmp, http.>
<Apr 19, 2007 12:20:08 PM EDT> <Notice> <Server> <BEA-002613> <Channel "Default"
is now listening on 127.0.0.1:7001 for protocols https, ldap, snmp, http.>
<Apr 19, 2007 12:20:08 PM EDT> <Notice> <WebLogicServer> <BEA-000331> <Started W
ebLogic Admin Server "AdminServer" for domain "base_domain" running in Developme
ntMode>
<Apr 19, 2007 12:20:08 PM EDT> <Notice> <WebLogicServer> <BEA-000365> <Server st
ate changed to RUNNING>
<Apr 19, 2007 12:20:08 PM EDT> <Notice> <WebLogicServer> <BEA-000360> <Server st
arted in RUNNING mode>

```

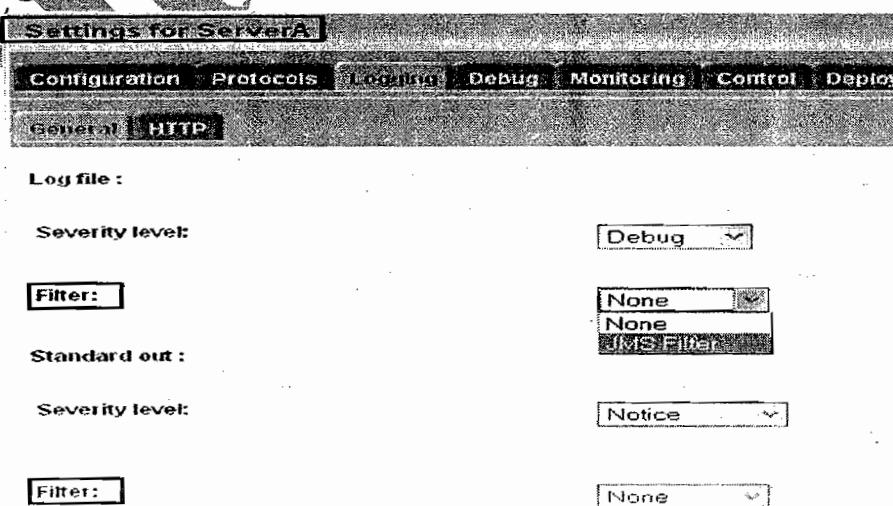
BEA-002613 *Notice: Channel "channel" is now listening on listenAddress:port for protocols protocols.*
 Description The server successfully started the listen thread and server socket.
 Cause None.
 Action None.

Log Filters

Log filters:

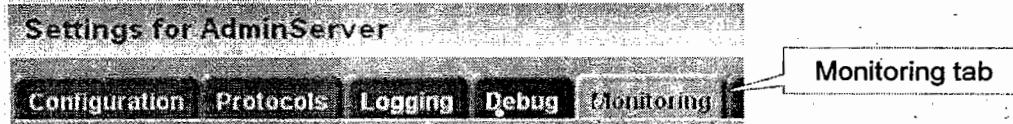
- Control the log messages that get published
- Are based on the values of message attributes
- Can be applied to different message destinations:
 - Server log file
 - Server memory buffer
 - Server standard out
 - Domain log file

Applying a Log Filter



Console Monitoring

The console offers many monitoring capabilities for servers, application deployments, and Java EE services.



Monitoring All Servers

This figure contains two screenshots. On the left, the "Domain Structure" page shows a tree view of a domain named "HRDomain_Prod". One node under "Servers" is highlighted with a red oval labeled "1". On the right, the "Servers (Filtered - More Columns Exist)" page shows a table of three servers: AdminServer (running, port 7001), ServerA (running, port 7011), and ServerB (shutdown, port 7021). The table includes columns for Name, Cluster, Machine, State, Health, and Listen Port. Buttons for New, Clone, and Delete are at the bottom.

Customizing the View for All Servers

This screenshot shows the "Servers" monitoring page with a customized view. At the top, there is a "Filter" section with "Filter by Column: Name" and a "Criteria:" input field. Below it, the "View" section allows customizing the displayed columns. A "Available" list includes: Listen Address, Listen Thread Start Delay Secs, Login Attempts While Locked, and Machine. A "Chosen" list includes: State, Health, Listen Port, Heap Size Current, and Locked Users Current Count. An arrow points from the "Available" list to the "Chosen" list. At the bottom, a table displays the filtered server data with the chosen columns: Name, State, Health, Listen Port, Heap Size Current, and Locked Users Current Count. The data rows are identical to the previous screenshot: AdminServer (running, port 7001), ServerA (running, port 7011), and ServerB (shutdown, port 7021).

Monitoring Individual Servers

The screenshot shows the 'Settings for AdminServer' page. At the top, there is a navigation bar with tabs: Configuration, Protocols, Logging, Debug, Monitoring (which is highlighted with a red circle containing the number 1), Control, and Deployment. Below the navigation bar is another row of tabs: General, Health, Channels, Performance, Threads, Timers, and Workload (with a red circle containing the number 2). The main content area displays runtime information about the server:

- State:** RUNNING
- Activation Time:** Mon Jul 07 09:27:39 EDT 2008
- WebLogic Version:** WebLogic Server 10.3 Mon Jun 16 22:31:55 EDT 2008 1127403
- Java Vendor:** Sun Microsystems Inc.
- Java Version:** 1.6.0_06
- OS Name:** Windows XP

Network Addressing Features

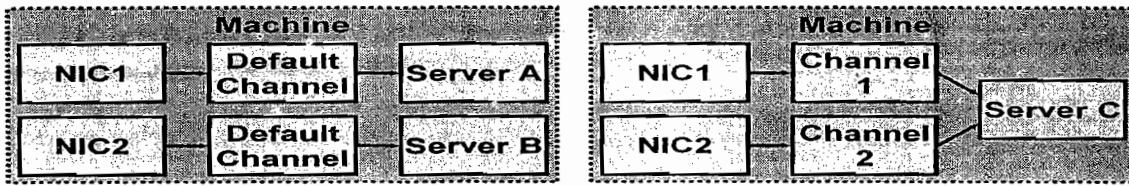
Adds flexibility to the networking configuration:

- Multiple NICs for a single WLS server
- Specific NICs or multiple port numbers on a NIC for specific WLS servers
- Ability to use multiple IP addresses with each server
- Ability to use a single IP address with multiple ports
- Ability to configure the cluster multicast port number independently of the port numbers used by the cluster members
- Multiple SSL configurations on one server

Network Addressing Features

Adds flexibility to networking configuration:

- Administration-traffic-only port
- Interoperability with previous WLS versions

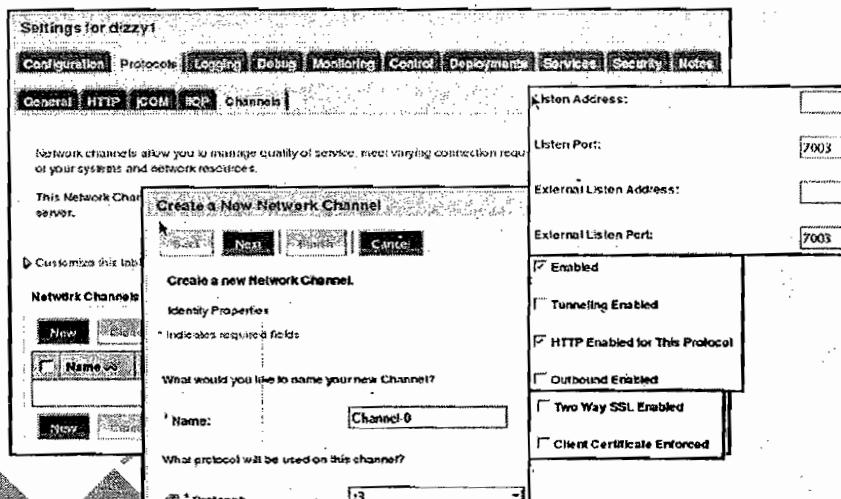


Network Channels

Network channels:

- Define the set of basic attributes of a network connection to WLS
- Can assign multiple channels to a single server (segment network traffic)
- Can prioritize internal (non-URL) connections
- Can separate incoming client traffic from internal server to server traffic in a domain
 - A “default” channel gets generated when a server is created.

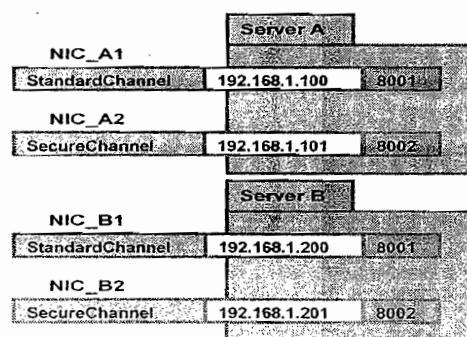
Configuring Network Channels



Using Channels: Example 1

Multiple NICs per server

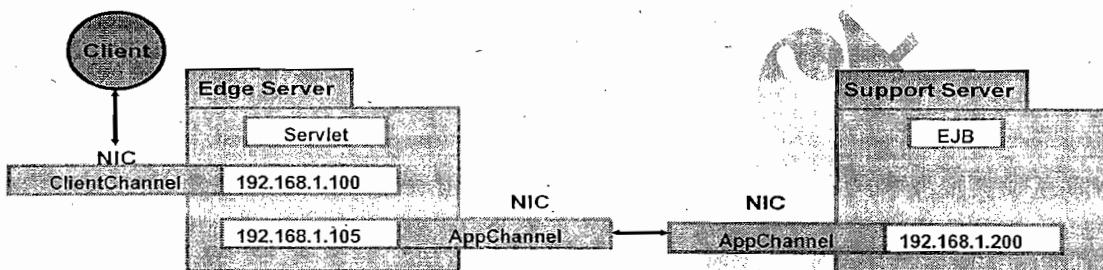
- Each server has two NICs.
- Each NIC has one channel; therefore, there are two channels per server.
- Types of channels
 - StandardChannel
 - Enables HTTP
 - Disables other protocols
 - SecureChannel
 - Enables HTTPS
 - Disables other protocols



Using Channels: Example 2

Separate internal and external traffic

- AppChannel is common between servers:
 - This is used for internal communication.
 - The OutgoingEnabled attribute is enabled.
- ClientChannel is used for external access:
 - Clients can connect only to public IP address 192.168.1.100.
 - The OutgoingEnabled attribute is disabled.

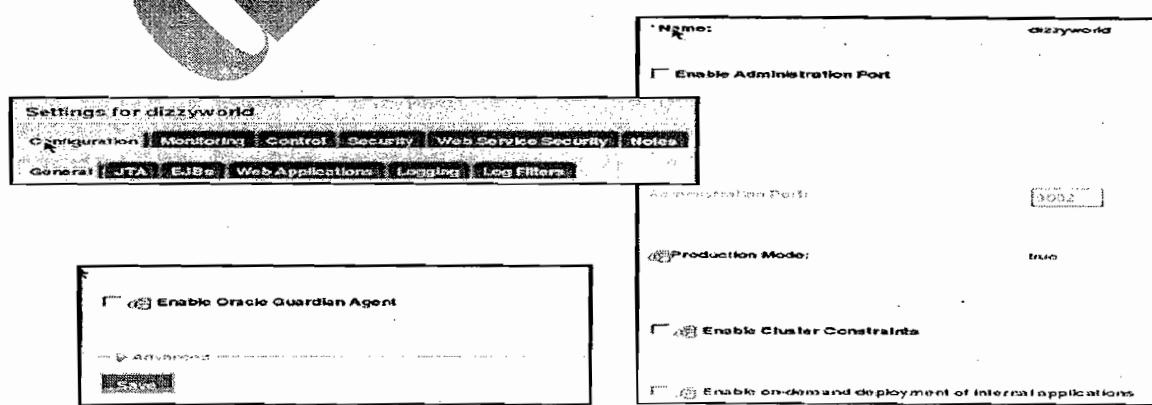


Administration Port

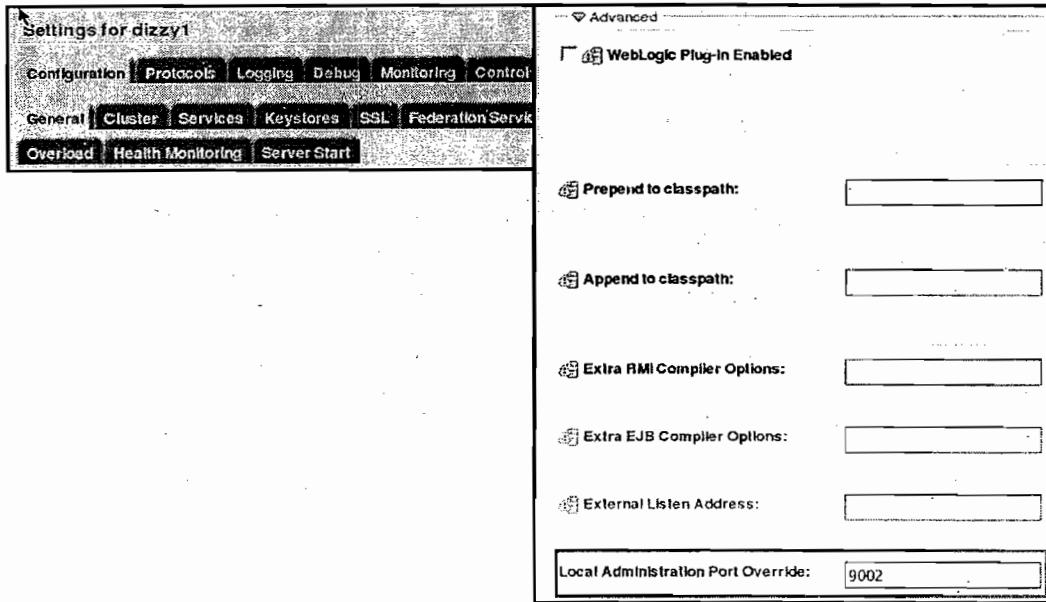
WLS allows the configuration of a dedicated administration port:

- Accepts only secure, SSL traffic:
 - All connections via the port require authentication by a server administrator.
- Generates an administration channel
- Specifies that channel settings are as the default channel except when:
 - A separate SSLListenPort value is defined
 - Non-SSL ListenPort is disabled
- Allows only administration traffic from the Administration Console, weblogic.Admin, and Managed Servers
- Enables you to start the server in Standby mode

Administration Port



Overriding the Administration Port



Quiz

Which is NOT a standard severity level for Oracle WebLogic Server log messages?

1. Debug
2. Transaction
3. Info
4. Notice
5. Error

Answer: 2

You can start a Managed Server using WLST and without using Node Manager.

1. True
2. False

Answer: 2

To start a Managed Server using the Administration Console, Node Manager must be configured on the machine where the Managed Server resides.

1. True
2. False

Answer: 1

Summary

In this lesson, you should have learned about:

- Using the Node Manager

- Monitoring domains and servers
- Configuring the network channels

Basic Deployment

Objectives

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

- Describe the Web server capabilities of Oracle WebLogic Server
- Use static and dynamic deployment
- Work with Oracle WebLogic Server servlets
- Define and work with enterprise applications

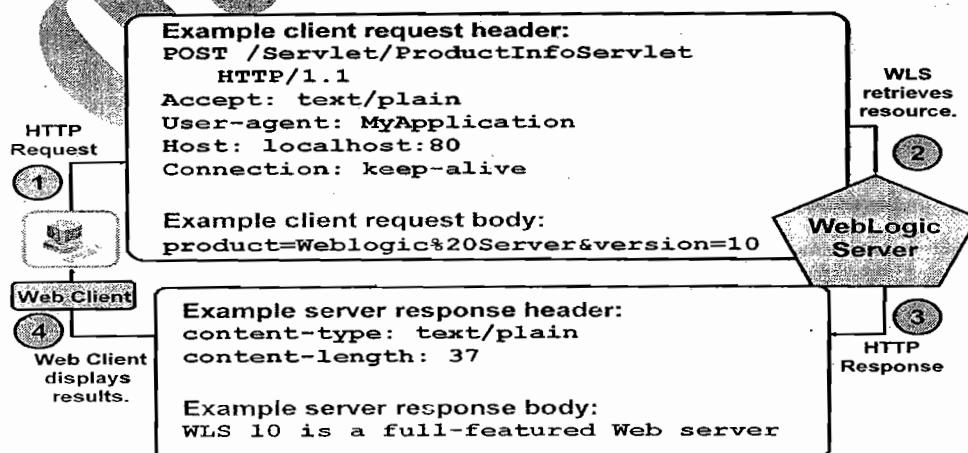
Road Map

- Web Servers
 - Web Servers Defined
 - HTTP
 - Static and Dynamic Content
- Web Applications
- EJB Applications
- Enterprise Applications
- Deployment

The Role of Web Servers

- Web servers are responsible for handling HTTP requests from clients.
- Web servers typically return:
 - Static content (HTML pages, graphics, and so on)
 - Dynamic content (Servlet, JSPs, CGIs, and so on)

A Typical Web Interaction



MIME Types

- Be sent to Adobe Acrobat.
- Application code can be directly executed. Multipurpose Internet Mail Extensions (MIME) is a protocol for identifying and encoding binary data.
- All HTTP response data is encoded with a MIME content type.
- Browsers interpret HTTP response data differently depending on the MIME type of the data:
 - HTML pages are parsed and displayed.
 - PDF documents can

HTTP Status Codes

HTTP status codes:

- Indicate to the client whether or not the request was successful
- Provide the client a reason for a failed request
- Are used by clients to provide alternate behavior

Indicating success:

The default status code is 200, which indicates success.

Reason for failure:

A status code of 404 tells the client the requested resource was not found.

Providing alternate behavior:

If a browser receives a 401 status code, the browser can prompt the user for an ID and password to login. WLS 10 is a full-featured Web server.

Static Content

- Static content documents are predefined on the server and do not change.
- Oracle WebLogic Server can be used to serve static content such as:
 - HTML documents
 - Images
 - PDF documents
- Oracle WebLogic Server can serve static documents:
 - Over standard HTTP
 - Through SSL using HTTPS

Dynamic Content

- Dynamic content documents may change based on the client's request.
- HTML documents can be created using:
 - Servlets
 - JSPs
 - Common Gateway Interface (CGI) programs

Java EE Web Applications

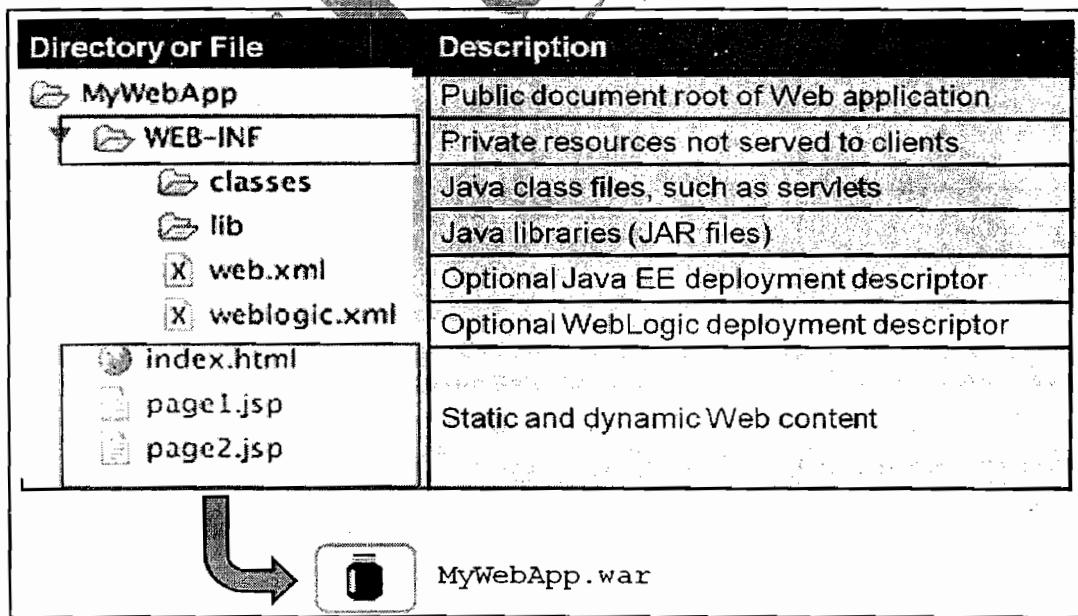
- Web Application:
 - Responds to client requests using the HTTP protocol
 - Typically implements an interactive Web site
- The contents of a Web application can include:
 - Java Servlets
 - JavaServer Pages (JSPs) for dynamic content
 - Static content (HTML, CSS, images, and so on)
 - Java classes and libraries
 - Client-side libraries (JavaScript, Java Applets, and so on)
 - XML deployment descriptors

Packaging Web Applications

Before you deploy an application package and register it with Oracle WebLogic Server, follow these steps to package a Web application:

1. Arrange the resources in a prescribed directory structure.
2. Develop the web.xml deployment descriptor (or copy as required).
3. Develop the weblogic.xml deployment descriptor (WLS-specific).
4. Archive the Web application into a .war file using Java Archive (JAR).
5. Deploy the Web application onto Oracle WebLogic Server.
6. Configure the Web application with the Oracle WebLogic Server Administration Console.

Web Application Structure



Web Application Structure

The structure of Web applications is defined by the servlet specification.

A Web application can be either:

- An archived file (.war file)
- An expanded directory structure

Directory/File	Description
MyWebApplication	Document root of Web application
META-INF	Information for archive tools (manifest)
WEB-INF	Private files that will not be served to clients
classes	Server-side classes such as servlets and applet
lib	The .jar files used by the Web application
web.xml	The Web application deployment descriptor
weblogic.xml	WLS-specific deployment descriptor

Configuring Web Applications

Web applications are configured using the web.xml and weblogic.xml deployment descriptors, which:

- Define the run-time environment
- Map URLs to servlets and JSPs
- Define application defaults such as welcome and error pages
- Specify J2EE security constraints
- Define work managers for applications
- Set the context root for the application

web.xml

The web.xml file is a deployment descriptor that is used to configure the following:

- Servlets and JSP registration
- Servlet initialization parameters
- JSP tag libraries
- MIME type mappings
- Welcome file list
- Error pages
- Security constraints and roles
- Resources
- EJB references

weblogic.xml

Using weblogic.xml, you can configure the following:

- The application's root context path

- Application logging
- Security role mappings
- Advanced session settings
- Session clustering
- References to shared libraries
- References to server resources (data sources, EJBs, and so on)
- Work managers and threading
- Virtual directories
- JSP compiler options

The weblogic.xml Deployment Descriptor

Example of the weblogic.xml deployment descriptor:

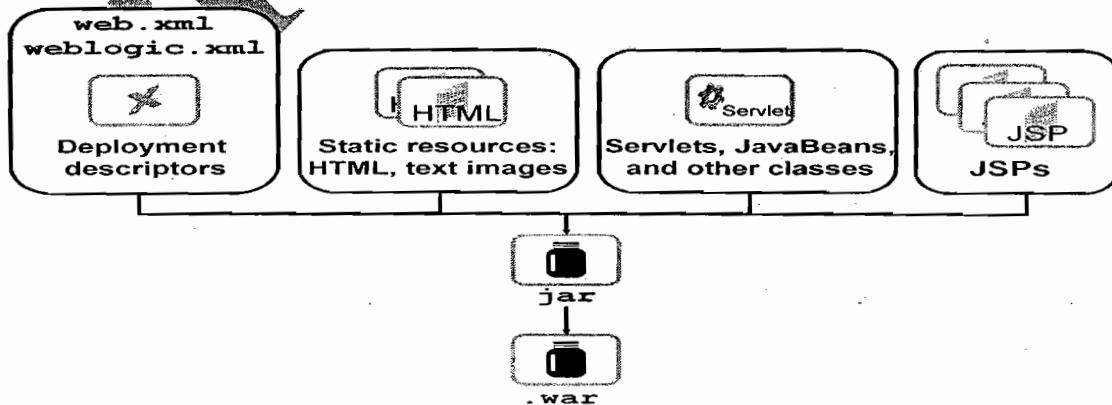
```
<?xml version='1.0' encoding='utf-8'?>
<weblogic-web-app
  xmlns="http://www.bea.com/ns/weblogic/100"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
</weblogic-web-app>
```

Web Application Archive

- Web application archives (.war files):
 - Are compressed files that contain directory structures that represent Web applications
 - Simplify the distribution and sharing of Web applications across a network
 - Can share common resources
 - Can be combined into larger applications
- For ease of development and debugging, Web applications are not archived until the end of production.

Web Application Archive

Web archives are created using the jar utility:



URLs and Web Applications

The URL that is used to reference a resource in a Web application must include the name of the Web application.

Accessing a resource in a Web application:
http://hostname:port/MyWebApplication/resource

Where:

Hostname	Host name mapped to virtual host or hostname:port
MyWebApplication	Name of the Web application; not necessary if this is the default Web application
resource	Static page, Servlet mapping, or JSP

Web Service Applications

A Web service application:

- Responds to HTTP client requests using the Simple Object Access Protocol (SOAP)
- Uses the same structure as a Java EE Web application
- Supports two additional deployment descriptors:
 - webservices.xml
 - weblogic-webservices.xml

Virtual Directory Mappings

Virtual directories:

- Can be used to refer to physical directories
- Enable you to avoid the need to hard code paths to physical directories
- Allow multiple Web applications to share common physical directories for specific requests such as images
- Decrease duplication of files across applications
- Are configured in weblogic.xml

Virtual Directory Mapping: Example

```
<virtual-directory-mapping>
    <local-path>c:/usr/gifs</local-path>
    <url-pattern>/images/*</url-pattern>
    <url-pattern>*.jpg</url-pattern>
</virtual-directory-mapping>
<virtual-directory-mapping>
    <local-path>c:/usr/common_jsp.sar</local-path>
    <url-pattern>*.jsp</url-pattern>
</virtual-directory-mapping>
```

Archive Versus Expanded Directory

- Archive Web applications if you are:
 - In production phase
 - Deploying to several machines
- Do not archive Web applications if:
 - In development or debugging phase
 - Application is being updated frequently
 - Deploying to a single machine (Administration Server)

EJB Applications

Enterprise JavaBeans (EJBs):

- Standardize the development and deployment of server-side distributed components
- Are annotated Java classes
- Are packaged with XML deployment descriptors
- Support the following capabilities:
 - Remote access over a network
 - Object-relational mapping via WLS or the Java Persistence API (JPA)
 - Transactions
 - Messaging integration
 - Dependency injection

Types of EJBs

EJB Type	Description	Example
Stateless Session	<ul style="list-style-type: none">• Do not maintain state• Are synchronous• Are maintained in memory	<ul style="list-style-type: none">• Check validity of stock symbol• Calculate billing of a phone call
Stateful Session	<ul style="list-style-type: none">• Offer conversational interaction• Maintain state for client• Are synchronous	<ul style="list-style-type: none">• Book a flight & car rental for travel• Manage a shopping cart
Entity	<ul style="list-style-type: none">• Represent persisted data• Are synchronous	<ul style="list-style-type: none">• Represent a player's statistics• Represent a stock's history

Do u want to become collection king

Attend

Collections

(Java.util package)

Mr. DURGA M.Tech

- » Collection » SortedSet**
- » List » NavigableSet**
- » Queue » Map**
- » Set » SortedMap**
- » NavigableMap**

3 Days 30 Hours Full Enjoy

COLLECTION FRAME WORK SYLLABUS

Collection

List

ArrayList

LinkedList

Vector

Stack

Set

HashSet

LinkedHashSet

SortedSet

NavigableSet

TreeSet

Queue

PriorityQueue

BlockingQueue

PriorityBlockingQueue

LinkedBlockingQueue

Map

HashMap

LinkedHashMap

WeakHashMap

IdentityHashMap

Hashtable

Properties

SortedMap

NavigableMap

TreeMap

Cursors

► Enumeration

► Iterator

► ListIterator

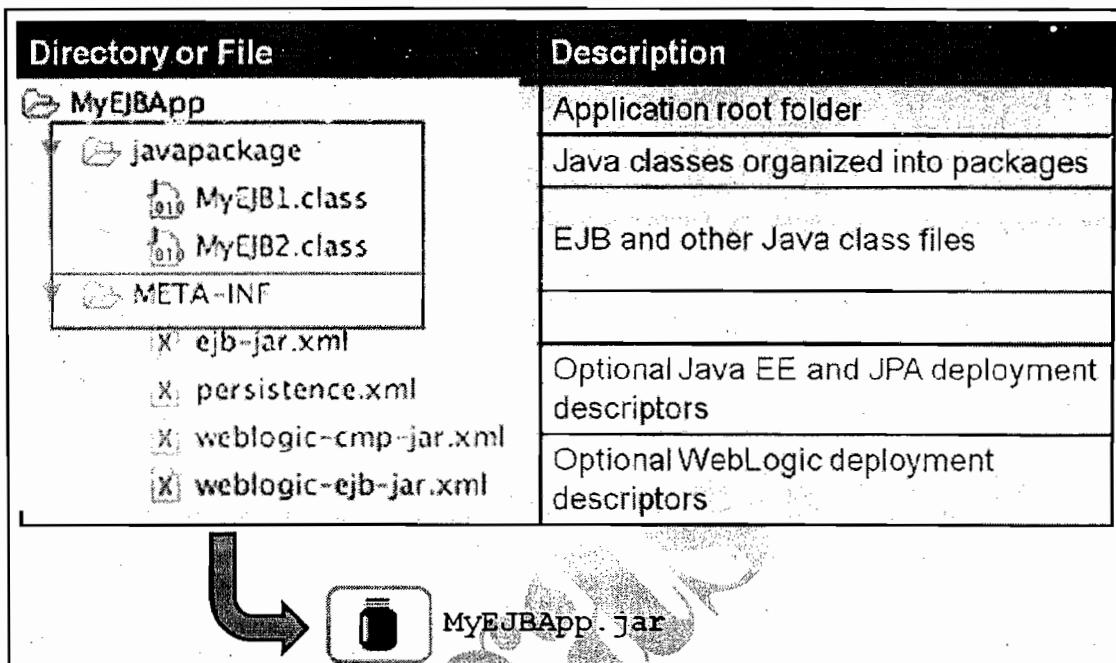
Comparable Vs Comparator

Collections Class

Arrays Class

Message-Driven	<ul style="list-style-type: none"> • Are asynchronous & stateless • Consume JMS messages 	<ul style="list-style-type: none"> • Store logging messages
-----------------------	--	--

EJB Application Structure



weblogic-ejb-jar.xml

Using weblogic-ejb-jar.xml, you can configure the following:

- Security role mappings
- Advanced security settings
- EJB clustering
- EJB pooling and caching
- Work managers and threading

EJB Administrator Tasks with WLS

The EJB administrator tasks include:

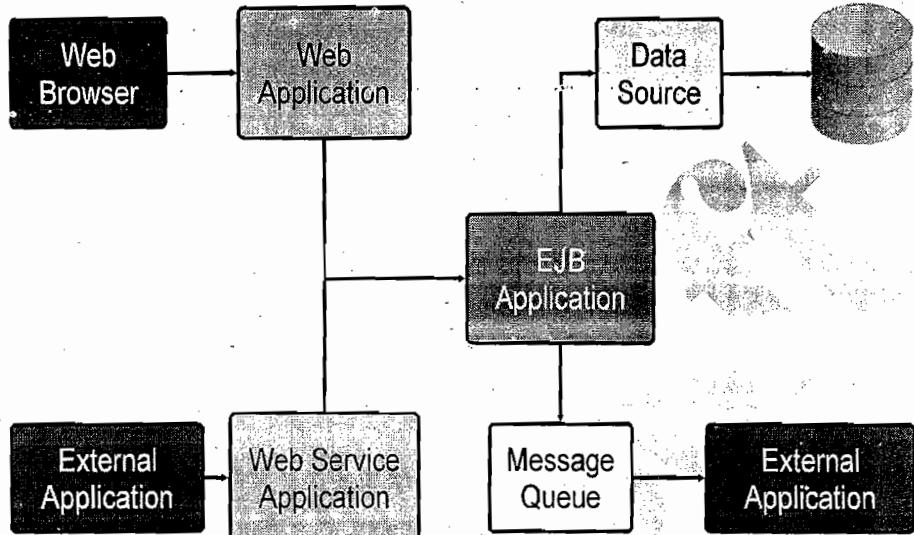
- Configuring and deploying
- Resolving JNDI and other infrastructure issues
- Monitoring EJB caches and pools

*What Is an Enterprise Application?

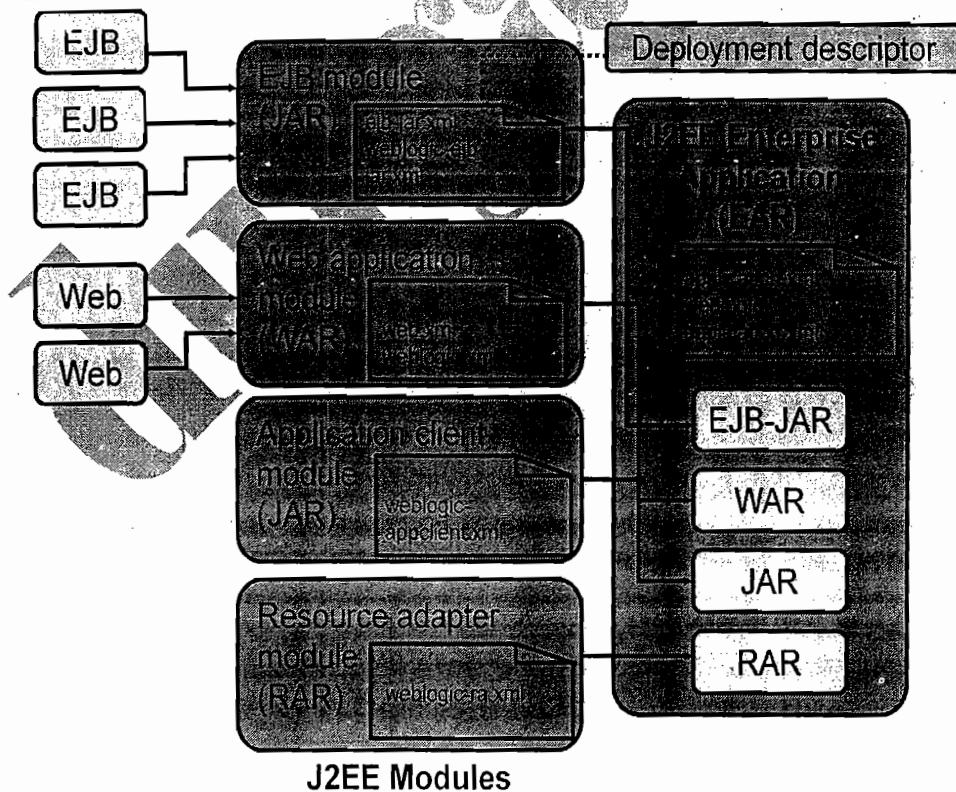
- An enterprise application is a grouping of several resources into one deployable unit that is packaged in an .ear file.
- These resources include:
 - Web applications (.war)

- EJB applications (.jar)
- Java applications (.jar)
- Resource adapters (.rar)

A Typical Java EE System



J2EE Enterprise Application



Java EE Enterprise Applications

An enterprise application:

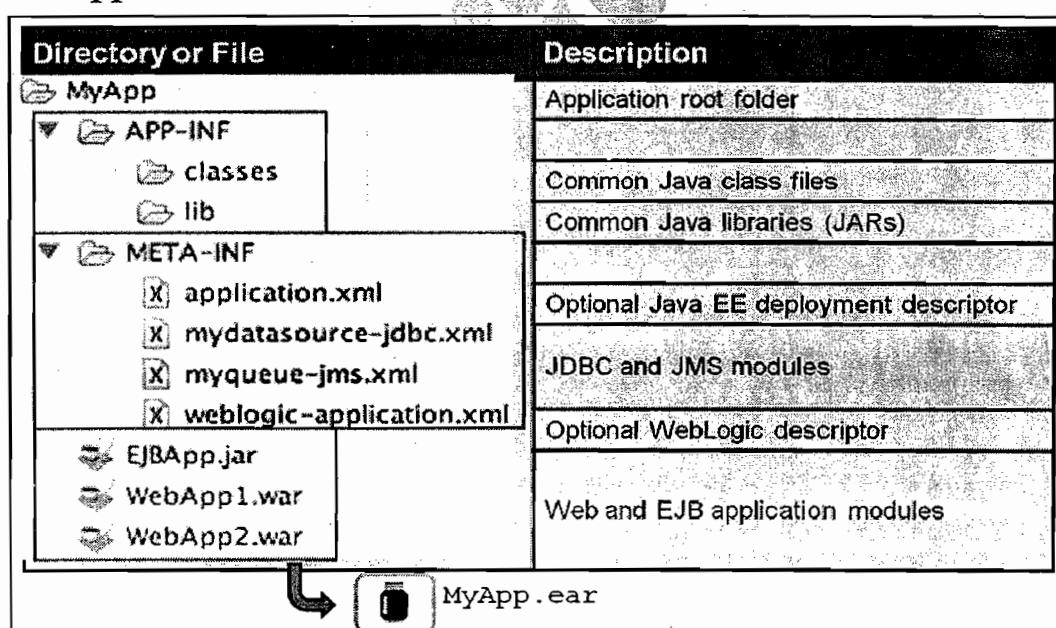
- Comprises one or more Java EE application modules:
 - Web applications
 - EJB applications
 - Other Java libraries (JARs)
- Allows related applications to be deployed as a unit
- Can include application-specific JDBC and JMS resources

Why Enterprise Applications?

Use enterprise applications to:

- Avoid namespace clashes
- Declare applicationwide security roles
- Deploy an application as one unit
- Share applicationwide EJB resources
- Configure local JDBC data sources
- Configure local JMS resources
- Configure local XML resources

Enterprise Application Structure



weblogic-application.xml

Using `weblogic-application.xml`, you can configure:

- References to shared libraries
- Work managers and threading

- Default EJB and Web application parameter values

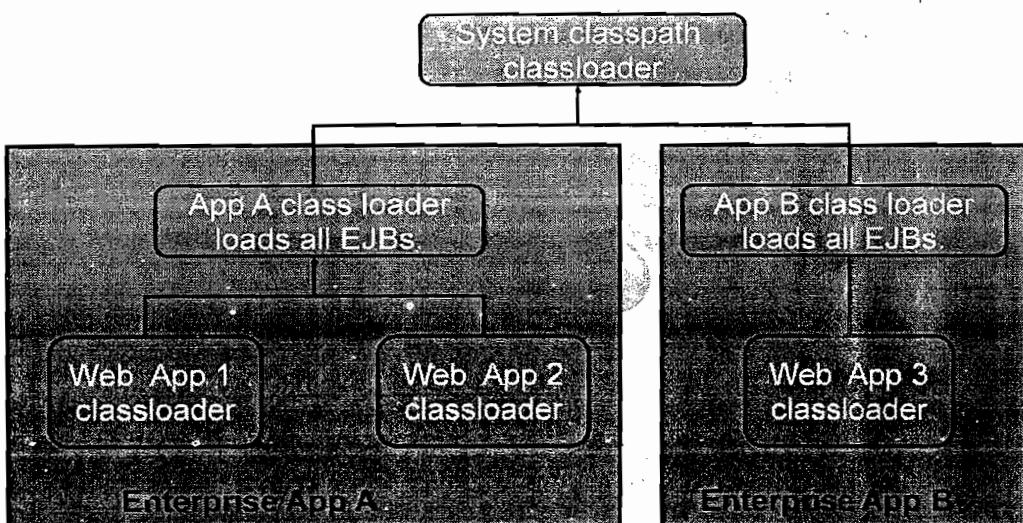
Configuring WLS-Specific Features

Configure enterprise-wide WLS-specific features with the `weblogic-application.xml` deployment descriptor:

- XML parsers
- XML entity mappings
- JDBC data sources
- JMS connection factories and destinations
- Security realms

WLS Application Classloader

Each application receives its own classloader hierarchy with the system class loader as its parent.



EAR Class Libraries

- Extending the J2EE specification, Oracle has added `APP-INF/lib` and `APP-INF/classes` to the standard J2EE EAR file structure.
- When an application is initialized, the paths extracted are prefixed to the application's classpath.
- Classes are added to the root classloader of the application.

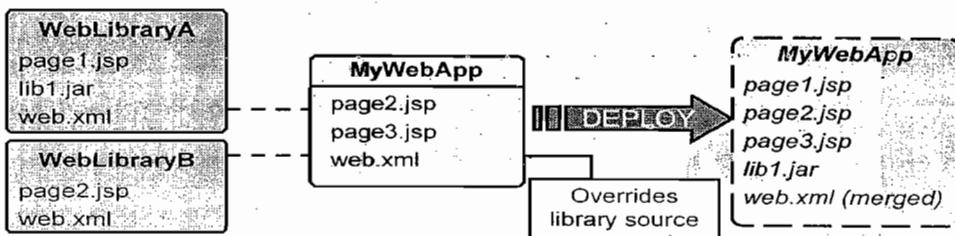
J2EE Library Support

- To make things easier, you can create a library of J2EE modules, package the modules into an enterprise application (EAR), and then deploy and register the module with the application container.
- Other applications can later use the modules as if they were packaged in their own EAR files.
- This allows for more reusability between the applications.

WebLogic Shared Java EE Libraries

A Shared Java EE library:

- Is a reusable portion of a Web or enterprise application
- Is referenced by other deployed applications
- Avoids duplicating source files among Java EE projects
- Can contain deployment descriptors that are merged with the application's descriptors



Shared Library References

- For Web applications, list the required shared libraries in `weblogic.xml`.
- For enterprise applications, list the required shared libraries in `weblogic-application.xml`.

Excerpts from `weblogic.xml`:

```
<library-ref>
    <library-name>ajax-tools-lib</library-name>
    <specification-version>1.5.0</specification-version>
    <implementation-version>2.0.0</implementation-version>
</library-ref>

<library-ref>
    <library-name>help-web-lib</library-name>
    <specification-version>1.5.0</specification-version>
    <implementation-version>1.1.0</implementation-version>
</library-ref>
```

Shared library name and version

Deployment Process: Overview

Deploying an application involves the following tasks:

- Preparing – Choosing whether to package the application as an archived file or keep it in an exploded directory
- Configuring – Creating a deployment plan to maintain the configuration changes without changing the deployment descriptors
- Deploying – Targeting and distributing the application to servers in an Oracle WebLogic Server domain

Deployment Methods

- WLS supports three deployment methods:
 - Auto-deployment
 - Console deployment
 - Command-line deployment

- You can deploy:
 - Enterprise, Web, and EJB applications
 - Web services
 - J2EE libraries
 - JDBC, JMS, and Diagnostic Framework modules
 - Resource adapters
 - Optional packages
 - Client application archives
- Applications and EJBs can be deployed:
 - In an archived file (.ear, .war, .jar)
 - In an exploded (open) directory format

Deployment Tools

Several methods are available to deploy the Oracle WebLogic Server applications and shared libraries, including:

- Administration Console
- WebLogic Scripting Tool (WLST)
- weblogic.Deployer Java class
- wldeploy Ant task
- Auto-deployment folder

Auto-Deployment by Copying Files

If Production Mode is OFF:

- You can install an application simply by copying it (manually or using the console) to the "autodeploy" directory of the domain
- The Administration Server monitors this directory for new, changed, or removed applications
- This configures, targets, and deploys the application only to the Administration Server

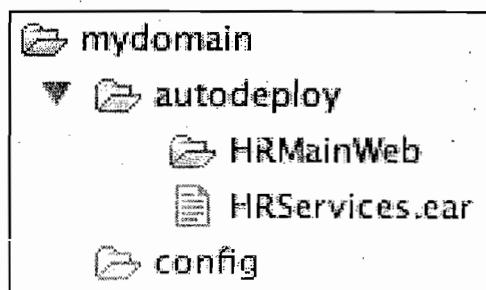
Location of Applications Directory:

\$BEA_HOME/user_projects/domains/domain_name/autodeploy

Auto-Deployment

- By default, the auto-deployment feature is enabled only if the domain is *not* running in production mode.
- When enabled:

- The Administration Server monitors its Autodeploy folder for new, updated, or removed applications
- Applications are targeted only to the Administration Server
- Developers can quickly test or experiment with an application



FastSwap and On-Demand Deployment

- WebLogic's FastSwap feature is:
 - Enabled using the WebLogic deployment descriptors
 - Available only if the domain is *not* running in production mode
 - Applicable only to Web applications that are *not* archived
- When enabled:
 - WebLogic automatically reloads the modified Java class files within applications
 - Developers can perform iterative development without an explicit redeployment
- On-demand deployment:

Excerpt from weblogic.xml:

```
<fast-swap>true</fast-swap>
```

Development Versus Production Modes

- An Administration Server starts using either:
 - The development mode, which turns auto-deployment on
 - The production mode, which turns auto-deployment off
- The Administration Server starts in the mode selected at domain creation time.
- The mode is set for all Oracle WebLogic Servers in a given domain.

Production Mode Flag

When Production Mode is disabled, applications can be dynamically deployed.

- An application poller is enabled in development mode.

Settings for dizzyworld

Configuration Monitoring Control Security Web Service Security Notes

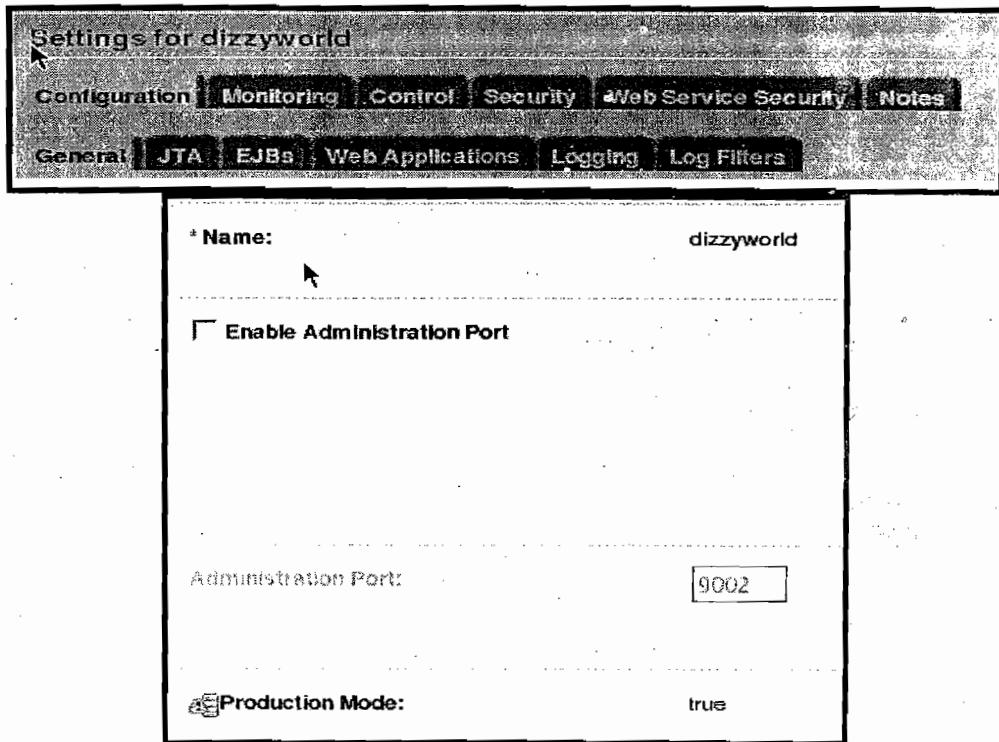
General JTA EJBs Web Applications Logging Log Filters

* Name: dizzyworld

Enable Administration Port

Administration Port: 9002

Production Mode: true



Console Deployment Method

Deploying with the console allows full administrator control:

- Installation from a location of your choice
- Manual configuration of application name
- Targeting of application to individual servers and/or clusters
- Configuring the application without targeting it
- Activating deployment when desired

Console Deployment Method

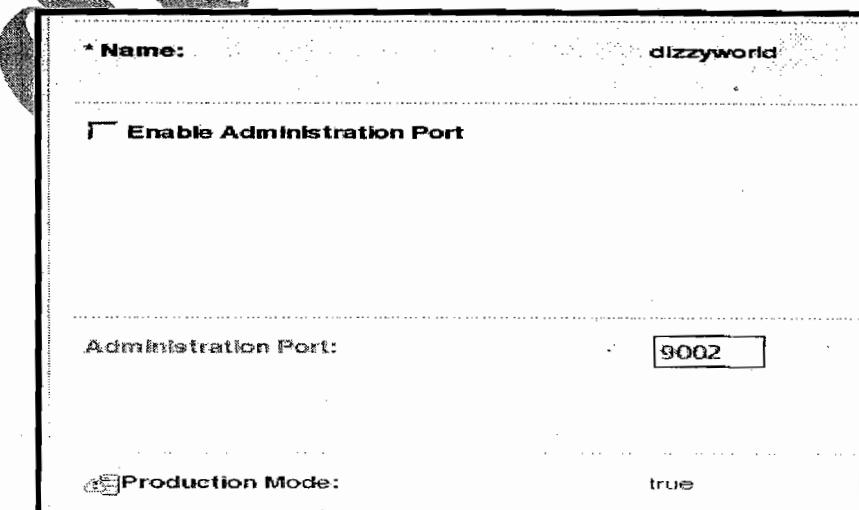
Best used with Production Mode

* Name: dizzyworld

Enable Administration Port

Administration Port: 9002

Production Mode: true



Preparing a New Application

The screenshot shows the Oracle WebLogic Server Administration Console. On the left, the Domain Structure tree includes HRWebDomain, Environment, Deployments (circled 1), Services, Security Realms, Interoperability, and Diagnostics. The Deployments page lists one deployment: MainHRWeb (State: Prepared, Health: OK). Below this, the Application Catalog displays:

Name	State	Health
MainHRWeb	Prepared	OK

Path: /home/oracle/bea/user_projects/myapplications/HRServices.ear
Recently Used Paths:
Current Location: localhost / home / oracle / bea / user_projects / myapplications

3 BenefitsWeb (open directory)
MainHRWeb (open directory)
HRServices.ear

Select an application folder or archive on the file system.

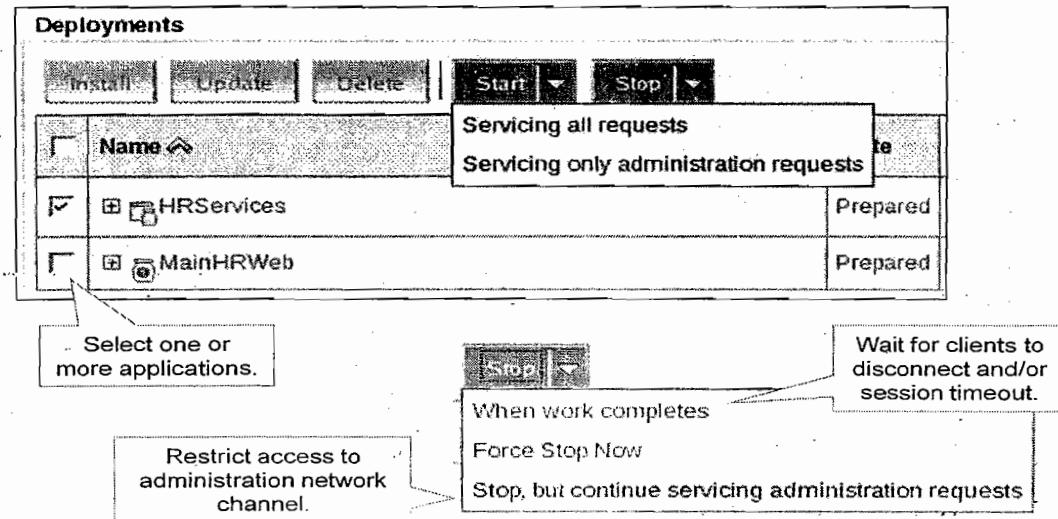
Preparing a New Application

The screenshot shows two dialog boxes from the Install Application Assistant:

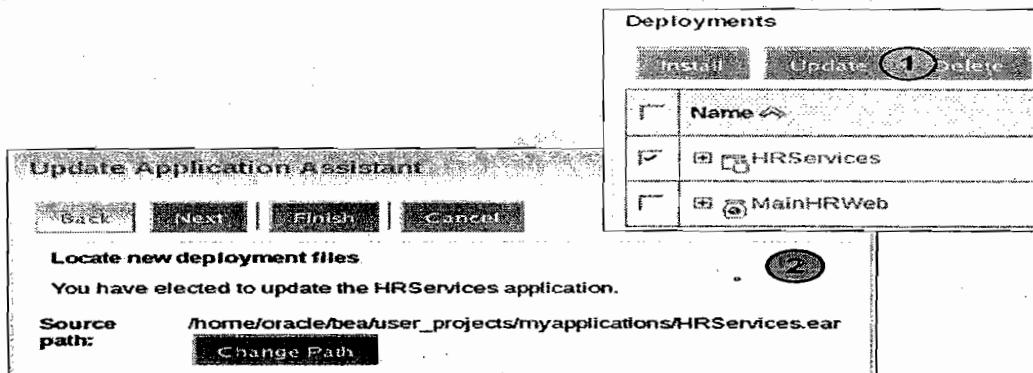
Install Application Assistant (Left):
Choose targeting style
1 Targets are the servers, clusters, and virtual hosts on which the application will run.
Install this deployment as an application (radio button selected)
The application and its components will be targeted to the following servers:
Install this deployment as a library
Application libraries are deployments that are available to multiple applications for running their referencing applications.
Install this deployment as an application, but target the components individually

Available targets for HRServices : (Right):
Servers
AdminServer (checkbox)
serverA (checkbox circled 2)
Deploy as a shared library. Target each EAR module separately.

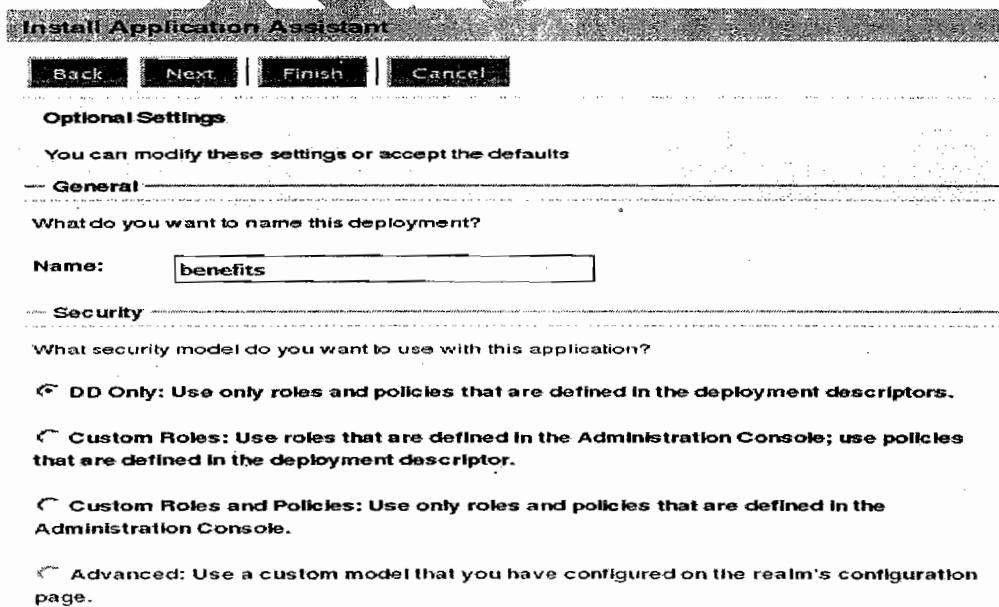
Deploying or Undeploying Applications



Redeploying an Application



Console Deployment



Console Deployment

Settings for physician

Overview | Deployment Plan | Configuration | Security | Targets | Control | Testing | Monitoring | Notes

Start/Stop | Initialize Caches:

Use this page to start or stop a deployed Enterprise application.

Enterprise Application

		Showing 1 to 1 of 1 Previous Next	
		State	Type
<input checked="" type="checkbox"/>	physician	distributing initializing	Enterprise Application
<input type="checkbox"/>	EJBs		
<input type="checkbox"/>	Modules		
<input type="checkbox"/>	Web Services		
		com.bea.physician.facade.broker.JaxRpcRecordCreationFacadeClientBroker	Web Service

Start | Stop

Showing 1 to 1 of 1 Previous | Next

Editing Deployment Descriptors

Settings for physician

Overview | Deployment Plan | Configuration | Security | Targets | Control | Testing | Monitoring | Notes

Application | Persistence | Workload | Instrumentation

Save

In this page, you define the configuration of the application deployment descriptor file that is associated with this Web application module.

Session cookies max age (in seconds):	1	The life span of the session cookie (in seconds) after which it expires.
Session Invalidation Interval (in seconds):	60	The time (in seconds) that WebLogic Server waits between doing a check and freeing up memory. More Info...
Session Timeout (in seconds):	3600	The amount of time (in seconds) that a session can remain inactive.
<input type="checkbox"/> Debug Enabled		Specifies whether to add JSP line numbers to generated class files.
Maximum In-memory Sessions:	1	The maximum number of sessions to retain in memory. More Info...
Monitoring Attribute Name:		The monitoring attribute. More Info...
<input type="checkbox"/> Index Directory Enabled		Specifies whether the target should automatically generate an index directory.
Index Directory Sort By:		Specifies the way in which index directories are sorted. More Info...
Servlet Reload Check (in seconds):	1	The amount of time (in seconds) that WebLogic Server waits to check for changes.
Resource Reload Check (in seconds):	0	The amount of time (in seconds) that WebLogic Server waits to check for changes.
<input type="checkbox"/> Session Monitoring Enabled		Specifies whether runtime listeners will be created for session tracking.
Minimum Native File Size:	0	The minimum native file size. More Info...

Monitoring an Application

The monitoring features that are available vary by application type.

The screenshot shows two panels. On the left, the 'Domain Structure' panel for 'HRWebDomain' is displayed, with 'Deployments' highlighted (circled with number 1). On the right, the 'Settings for MainHRWeb' panel shows tabs for Overview, Deployment Plan, Configuration, Security, Targets, Control, Testing, Monitoring (which is selected and highlighted with number 2), and Notes. Below these tabs is a link 'Customize this table'. Under the 'Monitoring' tab, there is a section titled 'Web Applications' showing a table with one row:

State	Sessions	Sessions High	Total Sessions	Servlets
Active	3	3	4	5

Application Testing

You can use the Administration Console to test a deployed application.

The screenshot shows the 'Settings for benefits' page. The 'Testing' tab is selected. A message states: 'Use this page to test that the deployment of the Web application component (WAR file) was successful.' Below this is a section titled 'Deployment Tests' with a table:

Name	Test Point	Comments
benefits		
default	http://127.0.0.1:7003/benefits	Default url on server dizzy1
welcome.html	http://127.0.0.1:7003/benefits/welcome.html	Welcome file welcome.html on server dizzy1

Application Update and Delete

- Using the console, you can either update or redeploy applications after configuration or component changes, or delete or undeploy them.
- All concurrent deployment activities are tracked by the Administration Server in a series of tasks:
 - Task progress and outcome can be queried for each application.
 - Reasons for failure are logged.

Application Update and Delete

The image consists of two vertically stacked screenshots of the Oracle WebLogic Server Administration Console.

Screenshot 1: Shows the "Deployments" table with one application named "benefits". The application is active, healthy, and a Web Application. A circled "1" is next to the table header.

Name	State	Health	Type	Deployment Order
benefits	Active	OK	Web Application	100

Screenshot 2: Shows the "Update Application Assistant" dialog. It displays the "Source path" as "/home/oracle/wls_systadm/work/domains-dizzyworld/servers/AdminServer/upload/benefits.war" and the "Deployment plan path" as "(No value specified)". A circled "2" is next to the dialog title, and a circled "3" is next to the "Change Path" button.

Screenshot 3: Shows the "Deployments" table again, identical to Screenshot 1, with the "benefits" application listed.

Command-Line Deployment

- The weblogic.Deployer utility allows you to perform deployment operations similar to those available in the console.
- weblogic.Deployer actions can also be scripted with the Ant task wldeploy.

```
weblogic Deployer Syntax:  
* java weblogic Deployer [options]  
  [-deploy|-undeploy|-redeploy|-start|-stop|-listapps]  
  [file(s)]
```

Deployment with weblogic.Deployer

Prepare and deploy a new application:

```
java weblogic Deployer -adminurl t3://adminserver:7001  
  -username myuser -password mypass -name HRServices  
  -source /usr/HRServices.ear -targets serverA -deploy
```

Redeploy an application:

```
java weblogic Deployer -adminurl t3://adminserver:7001  
  -username myuser -password mypass -name HRServices  
  -redeploy
```

Undeploy an application:

```
java weblogic Deployer -adminurl t3://adminserver:7001  
  -username myuser -password mypass -name HRServices  
  -undeploy
```

List all applications:

```
java weblogic Deployer -adminurl t3://adminserver:7001  
  -username myuser -password mypass -listapps
```

weblogic.Deployer: Examples

More weblogic.Deployer Examples:

To list all deployed applications:

```
java weblogic.Deployer -adminurl t3://localhost:7001  
-username system -password weblogic -listapps
```

To list all deployment tasks:

```
java weblogic.Deployer -adminurl http://localhost:7001  
-username system -password weblogic -listtask
```

To cancel a deployment task:

```
java weblogic.Deployer -adminurl http://localhost:7001  
-username system -password weblogic -cancel -id  
tag
```

The screenshot shows a Windows Command Prompt window titled 'C:\WINNT\System32\cmd.exe'. The command entered is 'c:\java weblogic.Deployer -adminurl t3://localhost:7001 -username system -password weblogic -listapps'. The output displayed is:

```
timeoff <DEPLOYED>  
messaging <DEPLOYED>  
benefits <DEPLOYED>  
payroll <DEPLOYED>  
retirement <DEPLOYED>  
Number of Applications Found : 5
```

Deploying Applications with WLST

WLST provides a number of deployment commands.

You can use these commands to:

- Deploy, undeploy, and redeploy applications and stand-alone modules to an Oracle WebLogic Server instance
- Update an existing deployment plan
- Start and stop a deployed application

Deploying an Application with WLST

```
Deploy an application (deployapp.py):  
##  
# WLST script for Deploying J2EE Application #  
##  
  
# Connect to the server  
print 'Connecting to server'  
connect('system','weblogic','t3://localhost:7001')  
  
appname = "mbeanlister"  
application = "c:/domains/dizzyworld/apps/mbeanlister"  
  
# Start deploy  
print 'Deploying application ' + appname  
deploy(appname, application, targets='myserver',  
       planPath='c:/myapps/plan/plan.xml')  
print 'Done Deploying the application ' + appname  
exit()
```

Deployment with WLST

Prepare and deploy a new application, or redeploy an existing one:

```
connect('myuser','mypass','t3://adminserver:7001')
name = "HRServices"
location = "/usr/myapplications/HRServices.ear"

deploy(name, location, targets='serverA')
```

Other WLST deployment commands:

```
distributeApplication(location, targets='serverA')
startApplication(name)
redeploy(name)
stopApplication(name)
listApplications()
```

Quiz

A _____ is a reusable Oracle WebLogic Server application that can be referenced by other deployed applications:

1. Java Library
2. Shared JavaEE Library
3. Web Library
4. Composite Library
5. Reference Library

Answer: 2

Which of the following is NOT a supported type of application in Oracle WebLogic Server?

1. Enterprise application
2. EJB application
3. Process application
4. Web service application
5. Web application

Answer: 3

Which of the following are valid Oracle WebLogic Server deployment descriptor files for configuring applications?

1. weblogic-webapp.xml
2. weblogic-ejb-jar.xml
3. weblogic.xml
4. weblogic-application.xml
5. weblogic-library.xml

Answer: 2, 3, 4

Summary

In this lesson, you should have learned about:

- Web server and Web applications

- Packaging and deploying Web applications
- Enterprise JavaBeans concepts
- EJB configuration
- Enterprise application concepts

Understanding JNDI

Objectives

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

- Describe the naming and directory services
- Describe the high-level architecture of JNDI
- Define the basic terminology
- View the JNDI tree in Oracle WebLogic Server

Road Map

Introduction to Java Naming and Directory Interface (JNDI)

- What Are Directory and Naming Services and How They Work
- The High-Level Architecture of JNDI
- Viewing the JNDI Tree via the Administration Console and the Command Line

* **What Is JNDI?**

- The Java Naming and Directory Interface is an API for uniformly addressing the different naming and directory services.
- This is a major step forward because:
 - Different services use vastly different naming schemes
 - Java applications can now navigate seamlessly across databases, files, directories, objects, and networks

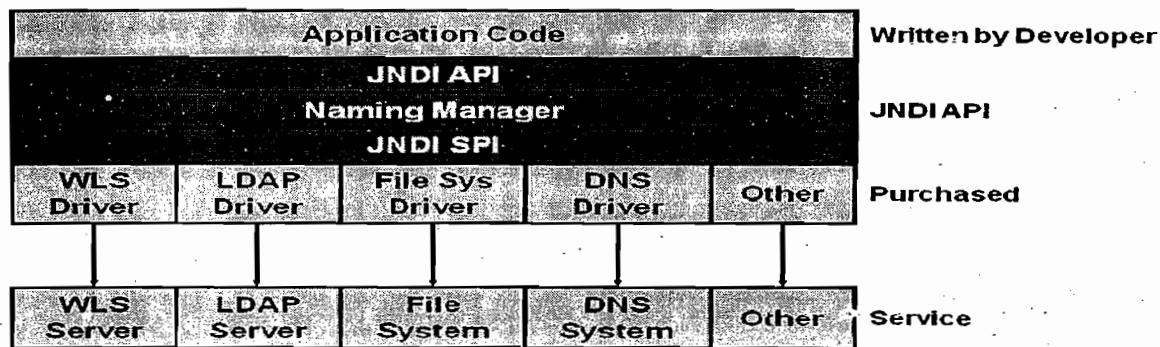
* **Why JNDI?**

In Oracle WebLogic Server, JNDI serves as a repository and lookup service for JNDI objects, including:

- Enterprise JavaBeans (EJB) home stubs
- JDBC DataSources
- JMS connection factories, queues, and topics
- Remote Method Invocation (RMI) stubs

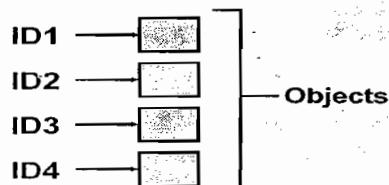
* **JNDI Structure**

Difference



Naming Service

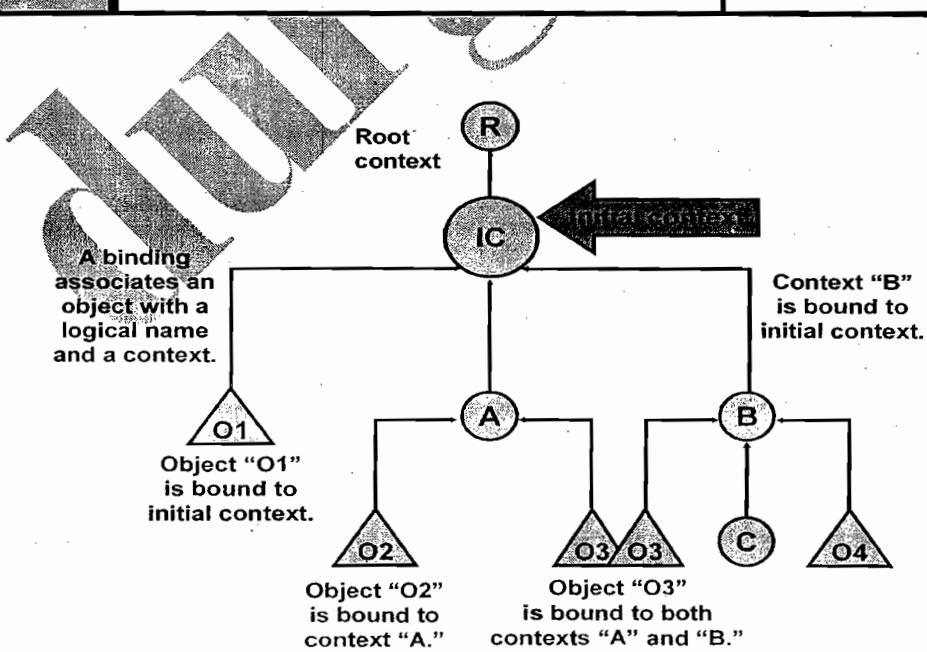
- A naming service provides a method for mapping identifiers to entities or objects.



- Naming service vocabulary:

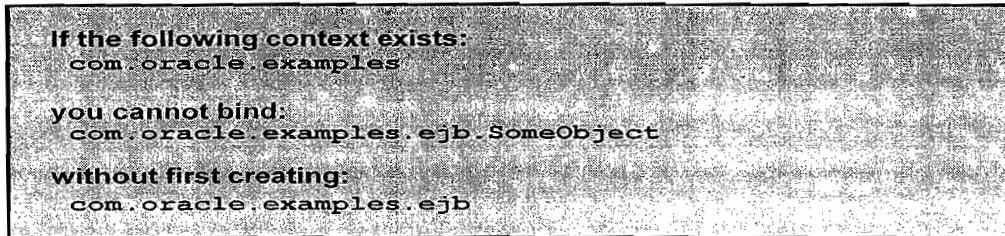
Term	Definition	Example
Binding	The association of an atomic name and an object	www.example.com is bound to 209.10.217.38.
Namespace	A set of unique names in a naming system	www.example.com/ products

JNDI Tree



Contexts and Subcontexts

- Subcontexts are referenced through the dot delimiters (.).
- The subcontexts must be created before objects are placed into them.
- Typically when objects are bound to a JNDI tree, subcontexts are automatically created based on the JNDI name.



JNDI for Administrators

Administrators must understand JNDI because it is their job to:

- Verify that objects are bound in the JNDI tree
- Set security on contexts within the JNDI tree

Viewing the JNDI Tree

The screenshot displays two main windows. On the left, the 'Domain Structure' window shows the 'Servers' section with one server listed: 'examplesServer(admin)'. This server is highlighted with a red circle labeled '1'. On the right, a detailed view of the 'examplesServer' is shown. This view includes a table with columns: Name, Cluster, Machine, State, Health, and Listen Port. The 'examplesServer(admin)' entry is highlighted with a red circle labeled '2'. Below the table, there are two tabs: 'Settings for examplesServer' (selected) and 'Settings for examples-dataSource-demoPool'. The 'examplesServer' tab has tabs for Configuration, Protocols, and Login. The 'examples-dataSource-demoPool' tab has tabs for Overview and Security. A note at the bottom of this panel says: 'This page displays details about this bound object.' On the far left, under 'JNDI Tree Structure', the tree view shows the structure of the 'examplesServer' object, with a red circle labeled '4' next to it. At the bottom center, there is a red circle labeled '3' next to the 'View JNDI Tree' button. On the right side of the detailed view, there is a summary table with the following data:

Binding	examples-dataSource-demoPool
Name:	
Class:	weblogic.jdbc.common.internal.RiniDataSource
Hash	29414270
Code:	
toString	weblogic.jdbc.common.internal.RiniDataSource@1c0d37e
Results:	

A red circle labeled '5' is positioned above the 'Name:' field in the summary table.

Listing the JNDI Contents

- 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","weblogic","t3://localhost:7001")
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--  sampleDataSource
                               weblogic.rmi.cluster.ClusterableRemoteObject
                               weblogic.rmi.cluster.ClusterableRemoteObject
                               weblogic.rmi.cluster.ClusterableRemoteObject
                               weblogic.rmi.cluster.ClusterableRemoteObject
```

Summary

In this lesson, you should have learned about:

- The naming and directory services
- The high-level architecture of JNDI
- The terminology used in the naming and directory services
- Viewing the JNDI tree in Oracle WebLogic Server

Setting Up JDBC

Objectives

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

- Describe the high-level architecture of JDBC
- List the four driver types and those provided by WLS
- Describe and configure data sources
- Use the Administration Console to manage the JDBC resources

Road Map

- Overview of JDBC
 - High-Level Architecture of JDBC and the Driver Model
 - Four Different Driver Types
 - Differences Between Two-Tier and Multitier Models
 - Drivers Provided by Oracle WebLogic Server
- Data Sources
- Monitoring and Testing Data Sources



What Is JDBC?

JDBC is an API for accessing databases in a uniform way.

JDBC provides:

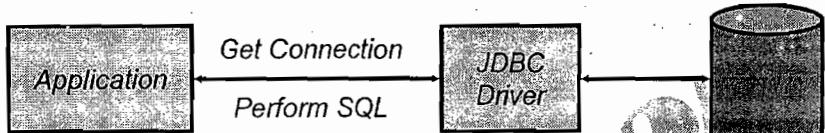
- Platform-independent access to databases
- Location transparency
- Transparency to proprietary database issues
- Support for both two-tier and multitier models for database access

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*

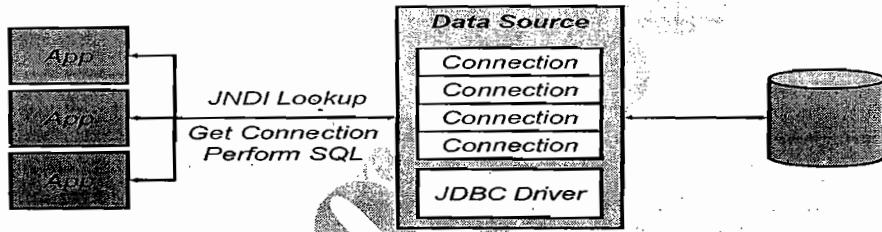
JDBC drivers are supplied by WLS or your database vendor.



* JDBC Data Sources

Data sources:

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

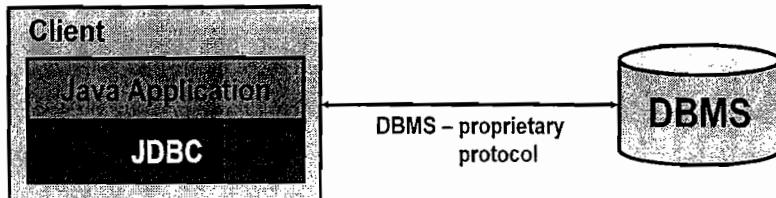


* Data Source Scope

- 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 EE enterprise applications
 - Accessible only by the containing application

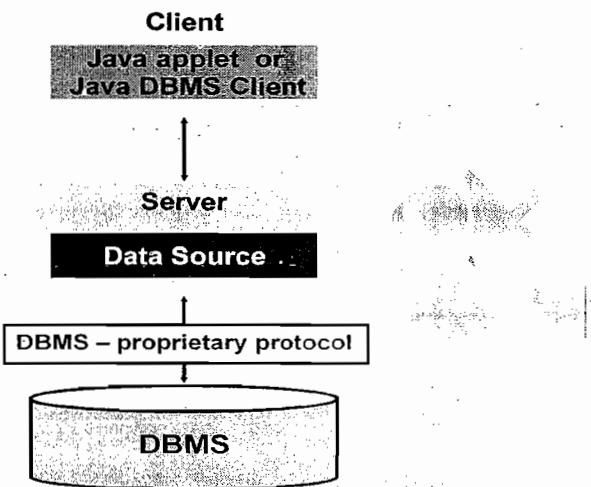
* Two-Tier Architecture

- In the two-tier model, a Java application communicates directly with the DBMS.
- A JDBC driver is needed that can communicate directly with the DBMS.
- This is a client/server configuration.

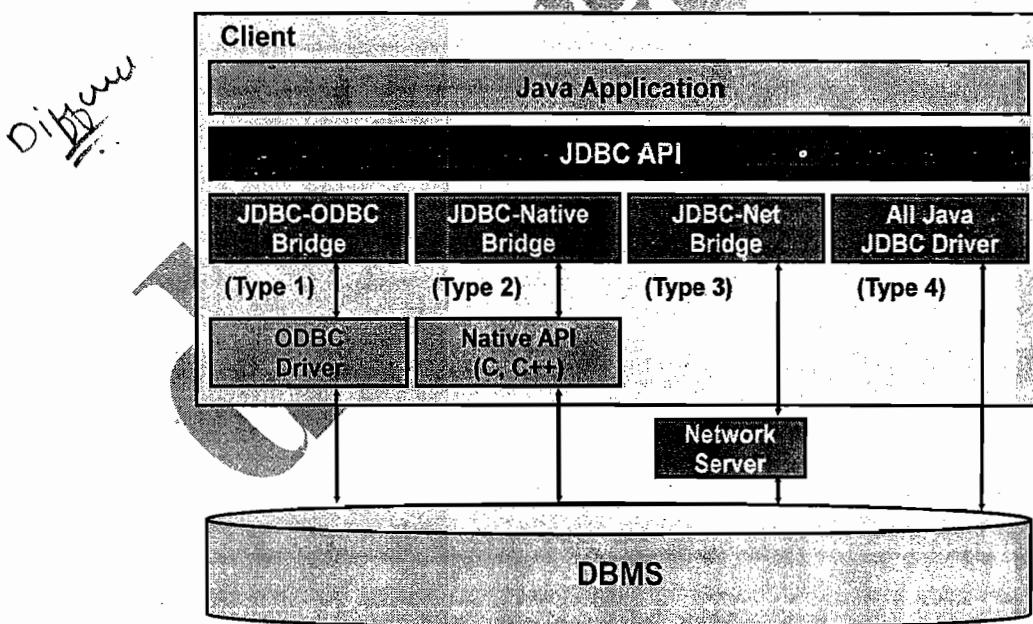


* Multitier Architecture

- In the multitier 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.

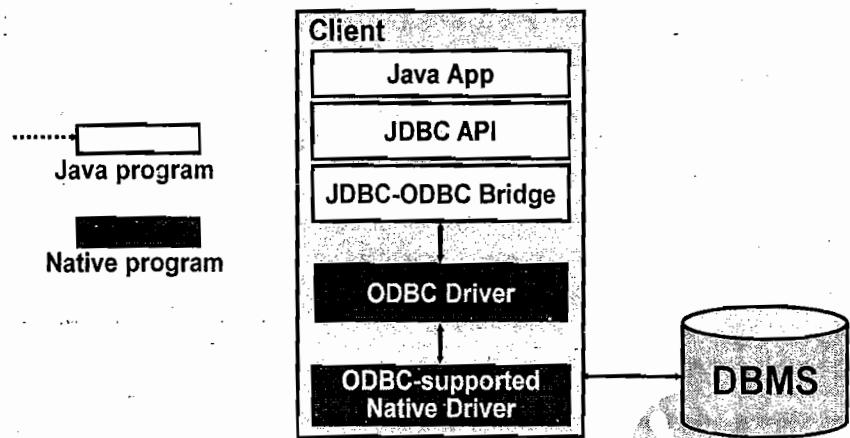


* JDBC Architecture



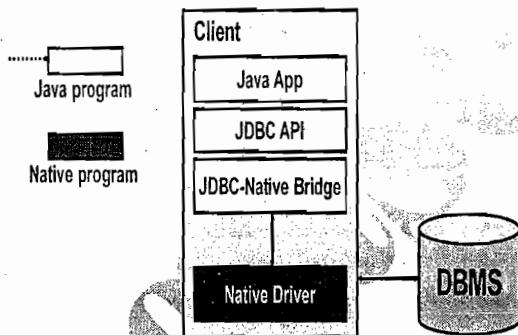
Type 1 Driver

- Is a JDBC-ODBC bridge
- Usually runs on Windows
- Requires an ODBC driver to be installed on the client machine

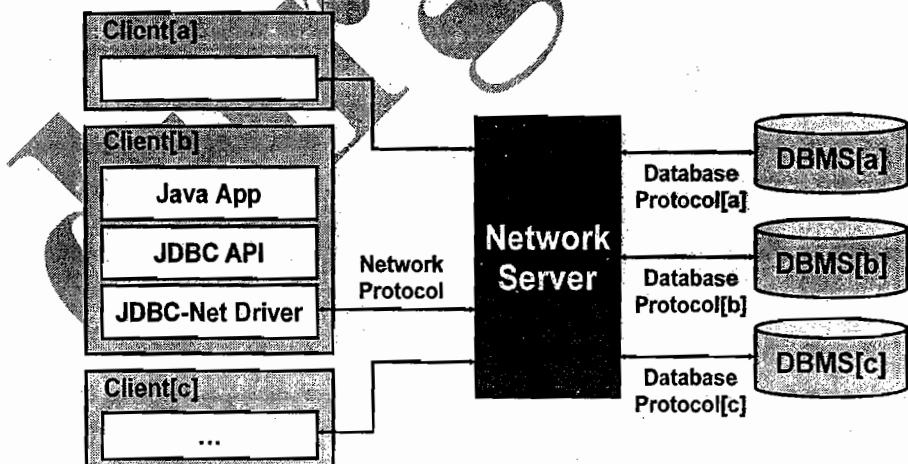


Type 2 Driver

- Requires a native driver to be already installed on the client machine
- Converts JDBC calls to native API calls of the database

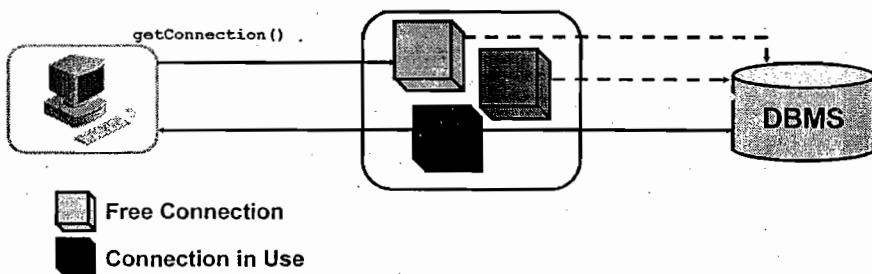


Type 3 Drivers



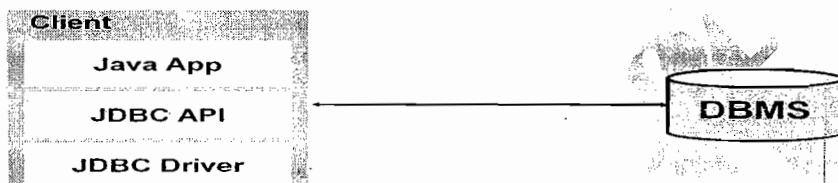
A network server can apply several techniques to boost performance:

- Connection pooling
- Load management
- Caching



Type 4 Drivers

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



WebLogic JDBC Drivers

- Oracle and third-party drivers are included in 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 server's classpath. →

Choosing the Correct Driver

- Choosing the correct driver can have significant impact on performance.
- For two-tier applications, use the Type 1, 2, or 4 driver that is specific to the DBMS that you use.
- For multi-tier applications, use:
 - A data source lookup in a client class
 - A Type 1, 2, or 4 driver on the server, specific to the DBMS you use
 - An XA driver where transaction support is required

* What Is a Data Source? It is anything.

- A data source object provides a way for a JDBC client to obtain a database connection from a connection pool.
- A data source:
 - Is stored in the Oracle WebLogic Server JNDI tree
 - Can support transactions
 - Is associated with a connection pool

(3)

What Is a Connection Pool?

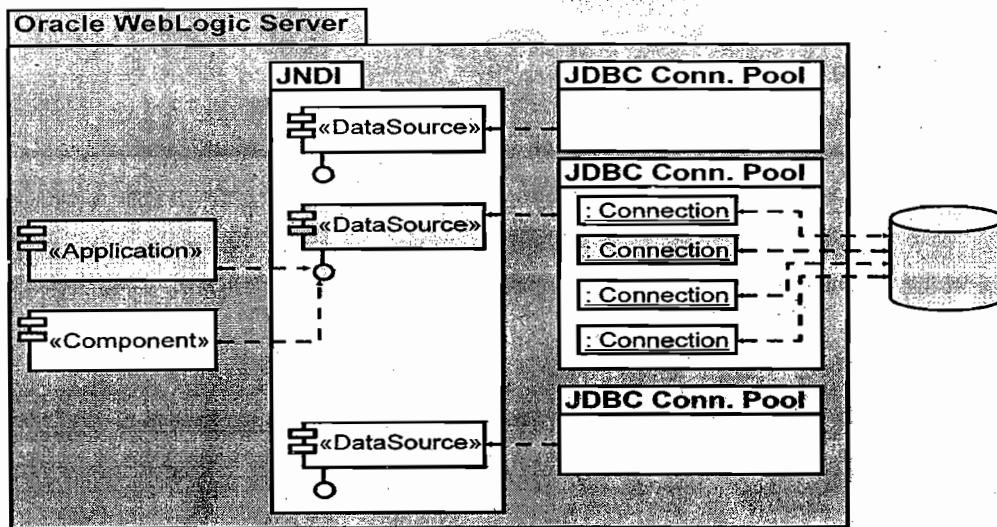
- A connection pool is a group of ready-to-use database connections associated with a data source. (~~as a pre-defined JDBC connections Maintained with Back end D.B.~~)
- Connection pools:
 - Are created at Oracle WebLogic Server startup
 - Can be administered using the Administration Console
 - Can be dynamically resized to accommodate increasing load

(4)

Benefits of Data Sources and Connection Pools

- The following are some advantages of this approach:
 - Time and overhead are saved by using an existing database connection.
 - Connection information is managed in one location in the Administration Console.
 - 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.

JDBC Data Source Architecture

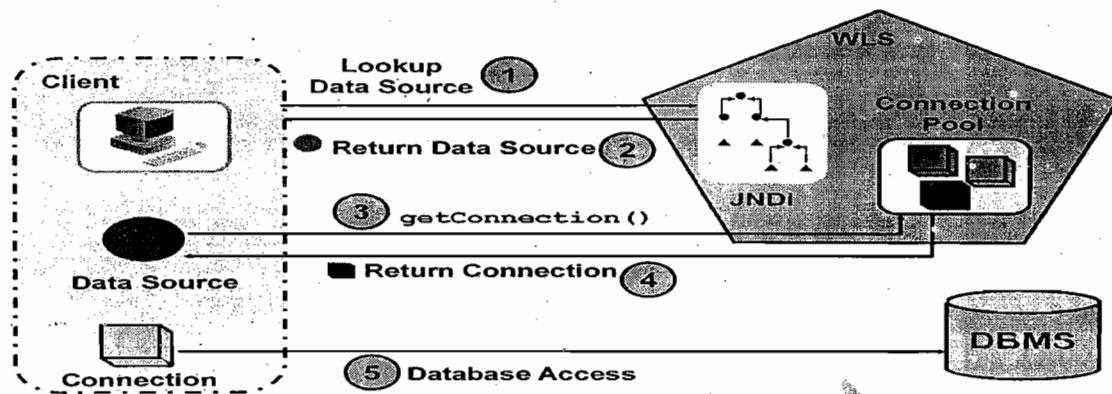


Modular Configuration and Deployment of JDBC Resources

- The JDBC configurations in WLS 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.
- JDBC application modules are a WLS-specific extension of J2EE modules and can be deployed either within a J2EE application or as stand-alone modules.

How Data Sources Are Used

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



Creating a Data Source

Step 1: Domain Structure

- HRWebDomain
- Environment
- Deployment
- Services
 - Messaging
 - JDBC
 - 1 Data Sources**
 - Multi Data Source
 - Data Source Pools
 - Persistent Stores

Create a New JDBC Data Source

What would you like to name your new JDBC data source?
Name: **HRDataSource**

What JNDI name would you like to assign to your new JDBC Data Source?
JNDI Name: **HRDataSource**

What database type would you like to select?
Database Type: **Oracle**

What database driver would you like to use to create database connections?
Driver: **Oracle's Driver (Thin XA) Versions:9.0.1,9.2,10.1,11**

Step 2: Create a New JDBC Data Source

What is the name of database you would like to connect to?
Database Name: **HRDATA**

What is the name or IP address of the database server?
Host Name: **db.mycompany.com**

What is the port on the database server used to connect to the database?
Port: **1521**

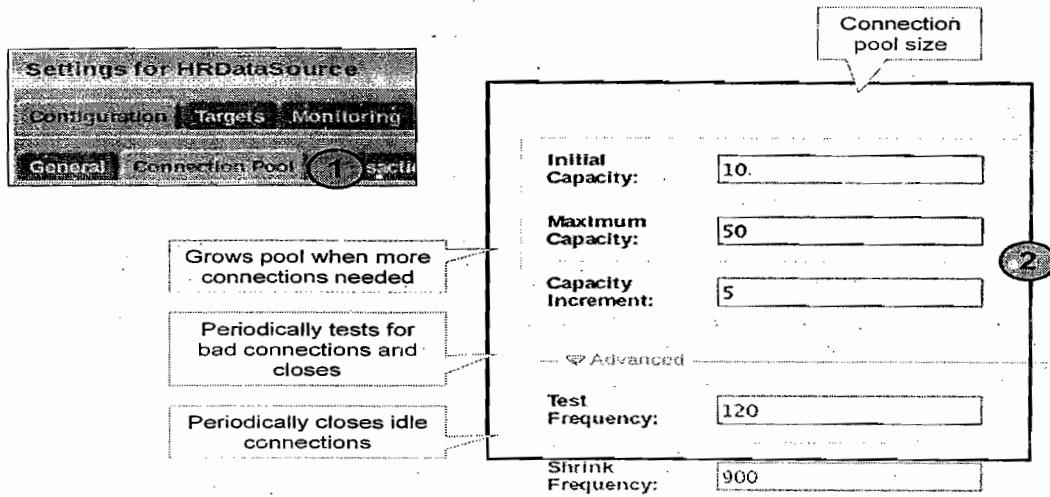
What database account user name do you want to use to create database connections?
User Name: **myuser**

What database account password do you want to use to create database connections?
Password: **password**

Create a New JDBC Data Source

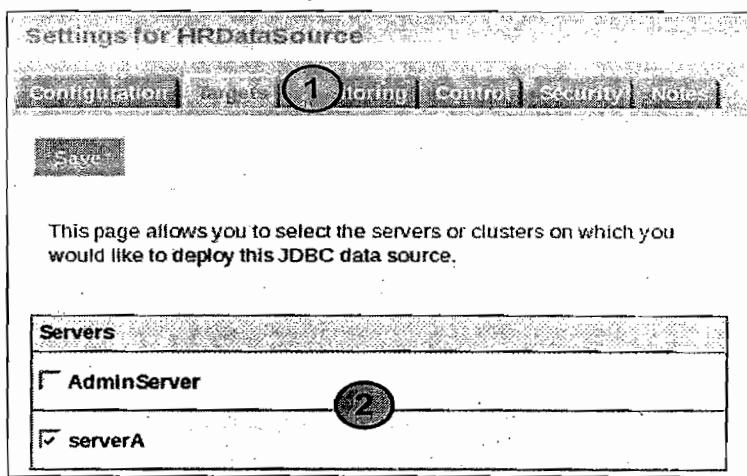
Test Configuration (2) Back Next Finish Cancel

Connection Pool Configuration



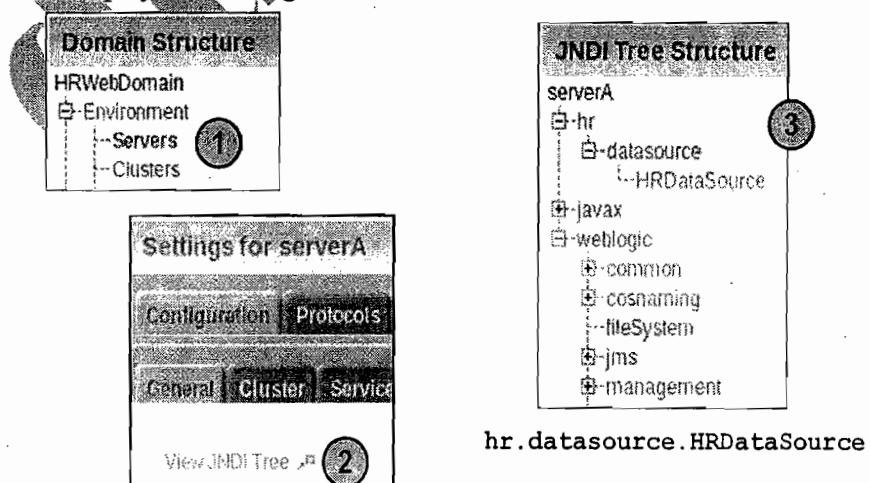
* Targeting a Data Source

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



* Viewing the Server JNDI Tree

Confirm data source deployment using the server's JNDI tree.



Connection Pool Checklist

- 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
- Ask your DBA for the maximum number of connections that your application will be allowed.

JDBC URLs

- Database locations are specified using a JDBC Uniform Resource Locator (URL).
- The syntax for a JDBC URL is:
 - `jdbc:<subprotocol>:<subname>`
 - Where `<subprotocol>` identifies the database connectivity mechanism
 - Where `<subname>` identifies the data source. The subname can vary depending on the subprotocol.

JDBC URL Examples

Example 1: This URL can be used to access a PointBase database:

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

The subprotocol is “`pointbase:server`;” the subname is a location of the PointBase database named “`HRDATABASE`.”

Example 2: This URL specifies that the “`oracle:thin`” subprotocol should be used to connect to an Oracle Database:

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

Connection Properties

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

Connection Properties

A partial list of connection properties for the supplied drivers:

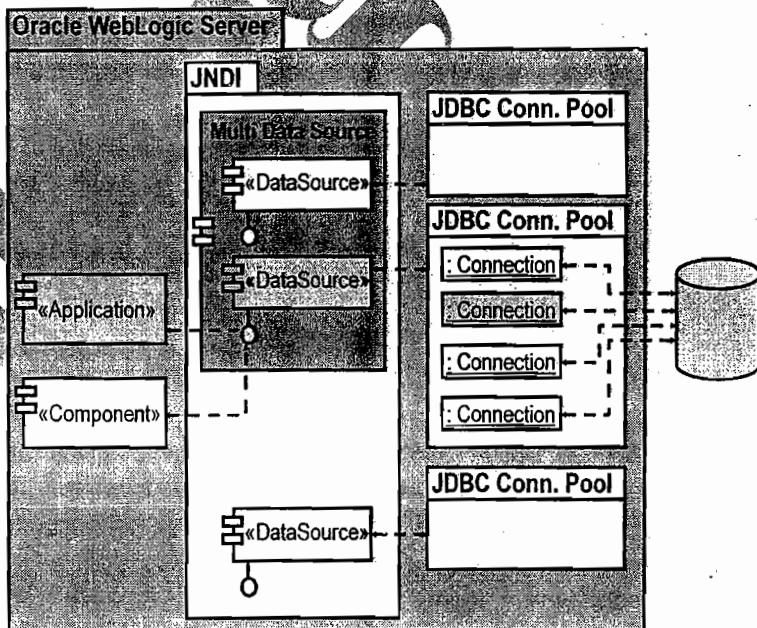
Driver	Some Connection Properties
Oracle	User, Password, ServerName, ServiceName, PortNumber

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

Multi Data Sources

- Multi data source:
 - Is an abstraction around a group of data sources
 - Determines which data source to use to satisfy the request depending on the algorithm selected in the multi data source configuration:
 - Load balancing or failover
 - Is bound to the JNDI tree
- XA support for multi data sources:
 - The WLS JDBC supports using multi data sources in XA transactions.
 - You can configure the data sources contained within the multi data source to use XA JDBC drivers.

Multi Data Sources



* Monitoring and Testing a Data Source

Monitor data source statistics.

Retest data source.

Customize this table

Showing 1 to 1 of 1 Previous | Next

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

Showing 1 to 1 of 1 Previous | Next

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

Answer: 2

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

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

Answer: 3, 5

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

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

Answer: 4

Summary

In this lesson, you should have learned about the following:

- JDBC high-level architecture
- Oracle WebLogic Server—provided JDBC driver types
- Data source definition and workings
- Connection pool definition and workings
- The management of JDBC resources using the Administration Console

Setting Up Java Message Service (JMS) Applications

Objectives

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

- Describe how Oracle WebLogic Server JMS is implemented
- Use the Administration Console to configure JMS administered objects
- Configure persistent messages
- Use the WLS Administration Console to monitor JMS

Road Map

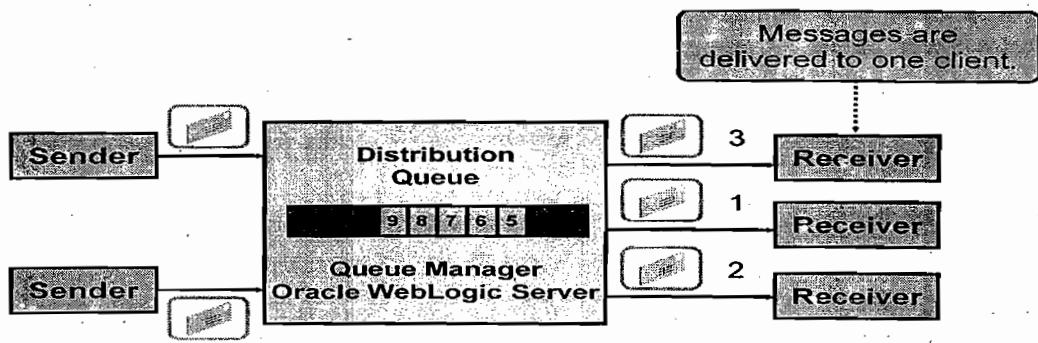
- Oracle WebLogic Server JMS Administration
 - Messaging Fundamentals
 - Point-to-Point (PTP) and Publish-Subscribe (Pub/sub) Domains
 - Configuring JMS Objects
 - Fine-Tuning WLS JMS
- Configuring Persistent Messaging
- Monitoring JMS in WLS

* **Message-Oriented Middleware**

- 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

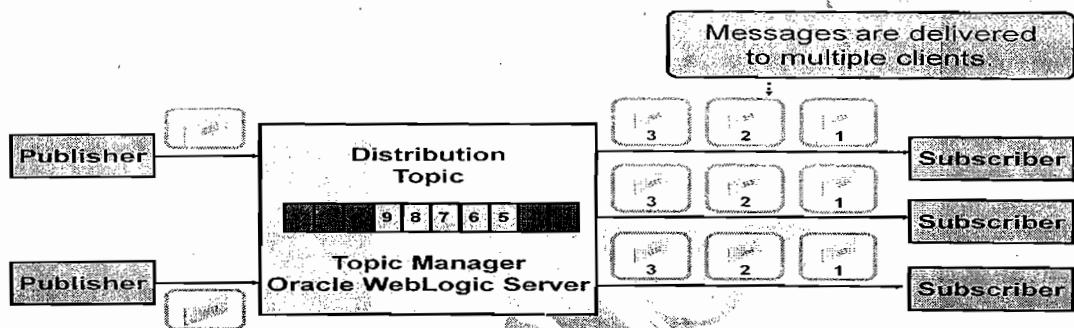
* **Point-to-Point Queue**

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



* Publish-Subscribe Topics

Publishing and subscribing to a topic decouples producers from consumers.

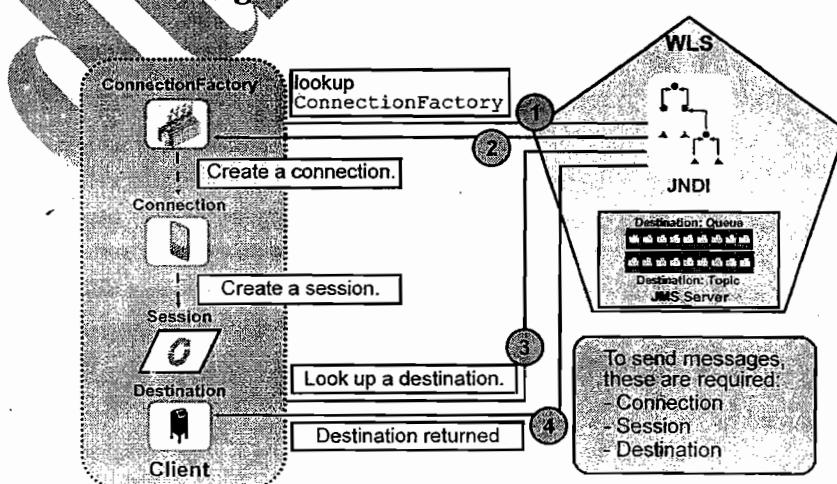


* Oracle WebLogic Server JMS Features

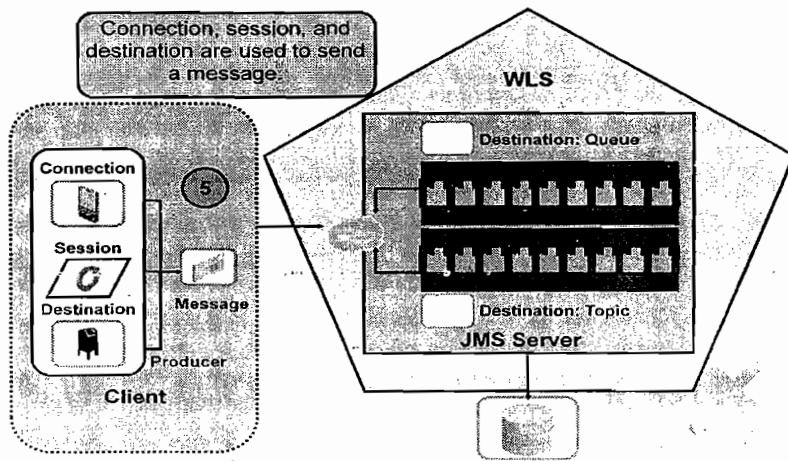
Oracle WebLogic Server JMS supports:

- PTP and pub/sub domains
- Guaranteed and transactional message delivery
- Durable subscribers
- Distributed destinations
- Recovery from failed servers

* JMS Architecture: Connecting

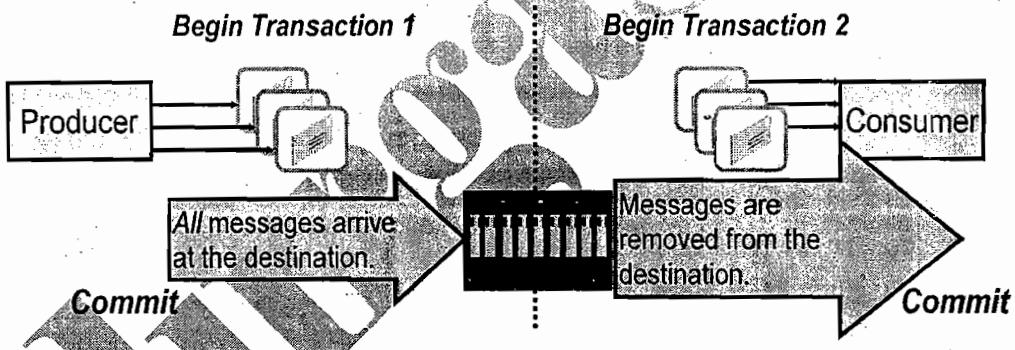


* JMS Architecture: Sending Messages



* 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.



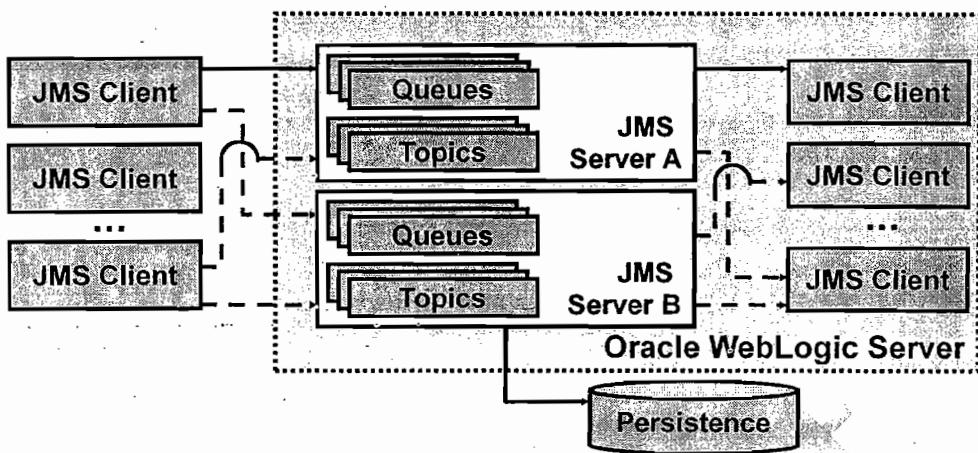
* Administrative Tasks

Administrative tasks include the following:

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

* Oracle WLS JMS Server

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



Creating a JMS Server

Change Center
View changes and restarts
Pending changes exist. They must be activated to take effect.
ACTIVE CHANGES
Undo All Changes

Summary of JMS Servers
This page summarizes the JMS servers that have been created in the current WebLogic Server domain.

1 Customize this table
JMS Servers (Filtered - More Columns)
Name Persistent
New Cancel

Create a New JMS Server
Back Next Finish Cancel
JMS Server Properties
The following properties will be used to identify your new JMS Server.
* Indicates required fields
2
What would you like to name your new JMS Server?
Name: JMServer-0
Specify persistent store for the new JMS Server.
Persistent Store: (none) Create a New Store

Targeting a JMS Server

Create a New JMS Server
Back Next Finish Cancel

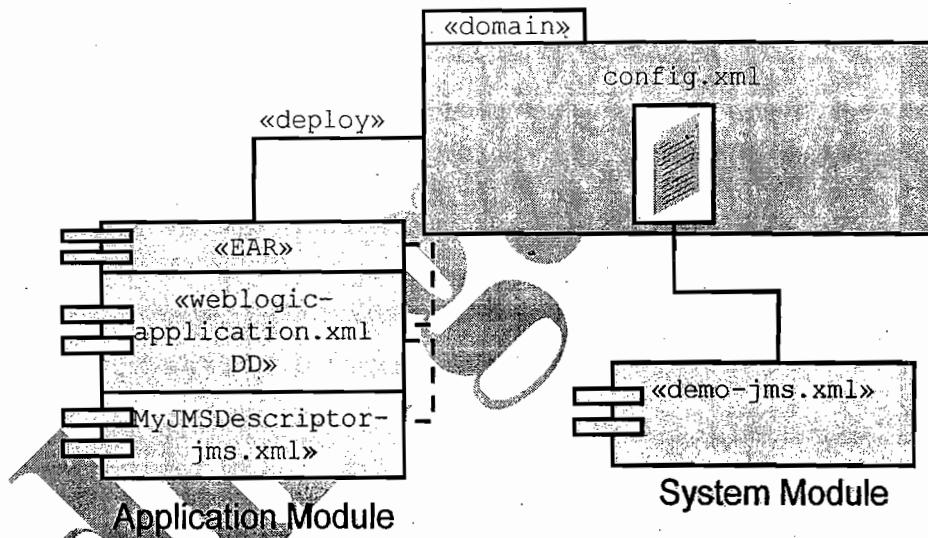
Select targets
Select the server instance or migratable target on which you would like to deploy this JMS Server.

Target: dizzy1
Back Next Finish Cancel

Configuring a JMS Server

* JMS Resources

JMS resources are managed either as system modules or as application modules.



* Modular JMS Resource Configuration and Deployment

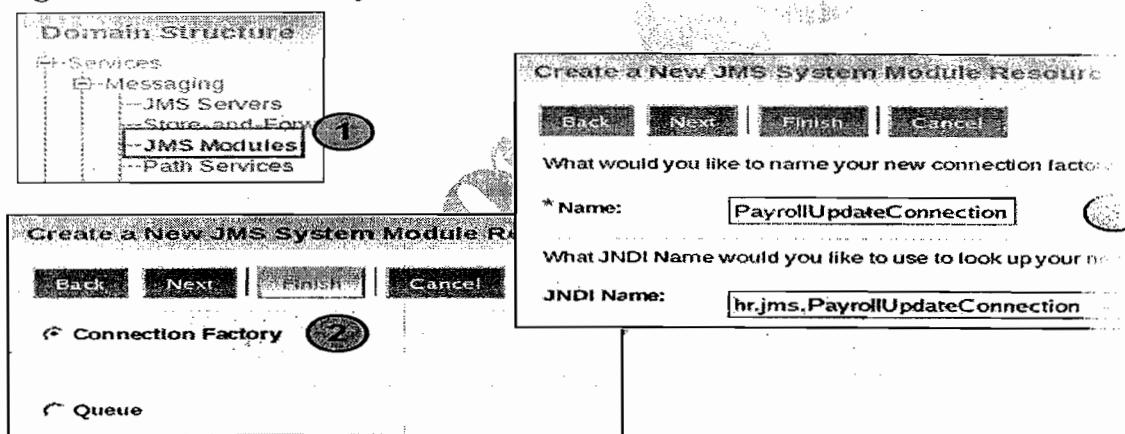
- JMS configurations in Oracle WebLogic Server are stored as JMS modules.
 - Defined by an XML file that conforms to the weblogic-jmsmd.xsd schema
 - Similar to standard J2EE modules
- An administrator can create and manage JMS modules as:
 - Global system resources
 - Global stand-alone modules
 - Modules packaged with an enterprise application
- An advantage of modular deployment is simplified migration 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

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

Creating a Connection Factory



Configuring a Connection Factory

Settings for PayrollUpdateConnection		Connection Factory Configuration tabs
<input checked="" type="radio"/> Configuration <input type="radio"/> Subdeployment <input type="radio"/> Notes		
<input checked="" type="radio"/> General <input type="radio"/> Default Delivery <input type="radio"/> Client <input type="radio"/> Transactions <input type="radio"/> Flow Control		
<input type="button" value="Save"/>		
Default Priority:		<input type="text" value="4"/>
Default Time-to-Live:		<input type="text" value="0"/>
Default Delivery Mode:		<input type="button" value="Persistent"/>

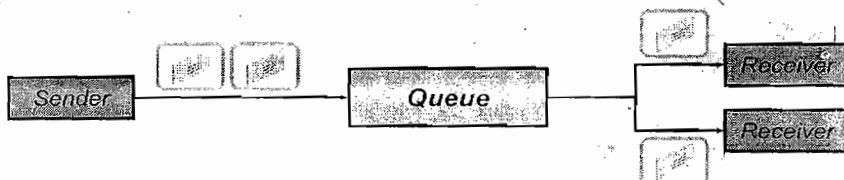
* Destination

- A destination is a lightweight object that is stored in JNDI.
- It is the target on a JMS server for sending or receiving messages.
- The JMS destination types are:
 - Queue
 - Topic

* Queue Destinations

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

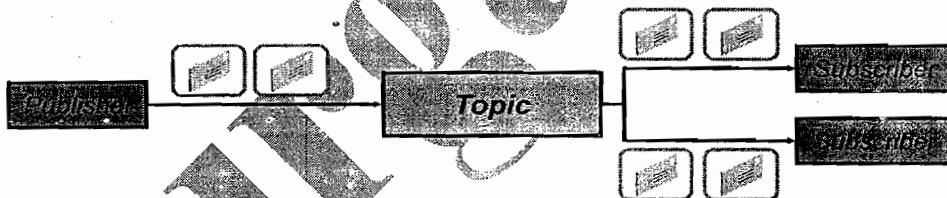
- Clients communicate with a *queue* destination.
- Messages are distributed to consumers in a serial fashion (first in, first out).
- Each message is delivered only to a single consumer.



* 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").



* Creating a Destination

The screenshot shows a software interface for creating a new JMS resource. At the top, a title bar says "Create a New JMS System Module Resource". Below it are buttons for "Back", "Next", "Finish", and "Cancel". A message says "Choose the type of resource you want to create." with two radio button options: "Queue" (unchecked) and "Topic" (checked).

On the left, there's a "Domain Structure" tree view with nodes like "Services", "Messaging", "JMS Servers", "Stores and Endpoints", "JMS Modules", and "Path Services". A circled "1" is next to the "JMS Modules" node.

At the bottom, a "Summary of Resources" table shows a row with a circled "2" and a "New" button. The table has columns for "Name" and "Type". One row is listed: "PayInquiryQueue" and "Queue". A circled "3" is next to the table.

Creating a Destination

Create a New JMS System Module Resource

Back | Next | Finish | Cancel

JMS Destination Properties

* Name: DepositUpdateTopic

JNDI Name: hr.jms.DepositUpdateTopic

Create a New JMS System Module Resource

Back | Next | Finish | Cancel

Subdeployments: PayrollJMServer ▾ Create a New subdeployment

Creating a Queue Destination

Settings for SystemModule-0

Configuration | Subdeployments | Targets | Security | Notes

This page displays general information about a JMS system module and its resources. It also lists existing resources and access existing resources.

Name:	SystemModule-0	The name of the module.
Descriptor File Name:	jms/system/module-0-jms.xml	The name of the descriptor file. More Info...

This page summarizes the JMS resources that have been created for this JMS system module. It lists destination definitions, connection factories, JMS templates, destination sort keys, destination quota, foreign servers, and store-and-forward parameters.

Customize this table

Summary of Resources

New | Delete

Name	Type	JNDI Name	Subdeployment	Targets
There are no items to display.				

Showing 0 to 0 of 0 | Previous | Next

Create a New JMS System Module Resource

Back | Next | Finish | Cancel

Choose the type of resource you want to create.

Use these pages to create resources in a JMS system module.

Depending on the type of resource you select, you are presented with different configuration options. For targetable resources, like stand-alone queues and topics, servers, and JMS SAF destinations, you can also proceed to associate targetable resources with subdeployments and the members to server resources.

Connection Factory

Queue

Create a New JMS System Module Resource

Back Next Finish Cancel

The following properties will be used to target your new JMS system module resource

Select the subdeployment you want to use. If you select (none), no targeting will occur.

Subdeployments: (none) Create a New Subdeployment

What targets do you want to assign to this subdeployment?

Targets:

3

4

JMS Destination Properties

The following properties will be used

* indicates required fields

Name: Queue-0

JNDI Name:

Template: None

Threshold and Quota

- A threshold and a quota can be set for the server and destination objects.
- 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
- A threshold is a limit that triggers message paging, flow control, and logged warnings, using:
 - Upper and lower values for the number of bytes
 - Upper and lower values for the number of messages

Configuring Thresholds and Quotas

Settings for Queue-0

Configuration Monitoring Control Security Subdeployment Notes

General Thresholds and Quotas Overrides Logging Delivery Failure

Thresholds

Bytes Threshold High: 9223372036854775807

Bytes Threshold Low: 9223372036854775807

Messages Threshold High: 9223372036854775807

Messages Threshold Low: 9223372036854775807

Quotas

Quota: None

Maximum Message Size: 2147483647

* **Durable Subscribers and Subscriptions**

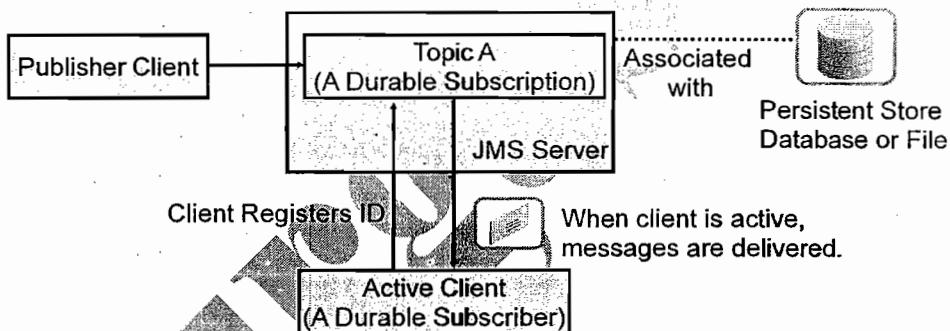
- Durable subscribers register durable subscriptions for guaranteed message delivery even if the subscribers are inactive.
- A subscriber is considered active if the Java object that represents it exists.
- By default, subscribers are nondurable.
- Administrators configure:
 - Where the messages are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

* **When to Use Persistent Messaging**

- Persistent messaging permits messages in memory to be written out to a persistent store.
- Configure persistent messaging if:
 - Development requires durable subscriptions (use durable subscribers in the application)
 - You require that in-progress messages persist across server restarts

* **How a Durable Subscription Works**

- If a subscriber client is active, messages are delivered normally.
- When the client becomes active again, its ID is used to retrieve and redeliver messages.



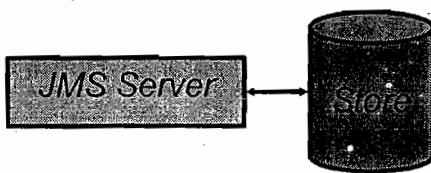
* **Configuring a Durable Subscription**

- 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:
 - A file store
 - A JDBC store (a connection pool)

* **Persistent Messaging**

- 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.
- The following types of JMS persistent stores are available:

- File system
- JDBC (requires an existing data source)



Creating a JMS Store

The screenshot shows two overlapping windows. The background window is titled "Domain Structure" and lists categories like Services, Messaging, JDBC, Persistent Stores, Foreign JNDI Provider, and Work Contexts. A circled "1" is next to the "Persistent Stores" category. The foreground window is titled "Create a New File Store" and contains fields for Name (HRFileStore), Target (serverA), and Directory (/mnt/share1/hr/wls/jms). A circled "2" is next to the "Create FileStore" button, and a circled "3" is next to the "OK" button.

Creating a JMS JDBC Store

- To configure JMS JDBC persistence, perform the following:
 - Create a JDBC DataSource.
 - Create a JMS 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.

The screenshot shows a window titled "Create a New JDBC Store". It includes fields for Name (JDBCStore-0), Target (AdminServer), Data Source ((None)), and Prefix Name. Buttons for OK and Cancel are visible at the top.

Assigning a Store to a JMS Server

Domain Structure

- HRWebDomain
 - Environment
 - Deployments
 - Services
 - Messaging
 - JMS Servers**
 - Store-and-Forward
 - JMS Modules

Settings for PayrollJMS Server

Configuration Logging Targets Monitoring

General Thresholds and Quotas Session Pools

Persistent Store:

(none) (none) **HREIS Store**

Persistent Connection Factory

Settings for ConnectionFactory-D

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, such as the default delivery mode, default time to live, etc.

Default Priority: 4 The default priority used for messages when a priority is not explicitly defined. [More Info...](#)

Default Time-to-Live: 0 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 time to live. [More Info...](#)

Default Time-to-Deliver: 0 The delay time, in milliseconds, between when a message is produced and when it is made visible on its destination. [More Info...](#)

Default Delivery Mode: Persistent The default delivery mode used for messages when a delivery mode is not explicitly

Configuring a Persistent Destination

Settings for Queue-D

Configuration Monitoring Control Security Subdeployment Notes

General Thresholds and Quotas Overrides Logging Delivery Failure

Overrides

Priority Override: -1

Time-to-Live Override: -1

Time-to-Deliver Override: -1

Delivery Mode Override: No-Delivery

Statistics for JMS Objects

Statistics are provided for the following JMS objects:

- JMS servers
- Connections
- Destinations

Monitoring JMS Servers

The screenshot shows the 'Monitoring' tab selected in the 'Settings for PayrollJMS Server' interface. A callout box highlights the 'Monitoring' tab with the number '1'. Another callout box highlights the 'Monitoring' link in the navigation bar with the number '2'. The main content area displays a table titled 'Statistics(Filtered - More Columns Exist)'. The table has columns: Name, Messages Current, Messages High, Bytes Current, and Session Pools Current. One row is present: PayrollJMS Server, 3, 3, 36, 0.

Name	Messages Current	Messages High	Bytes Current	Session Pools Current
PayrollJMS Server	3	3	36	0

Monitoring and Managing Destinations

The screenshot shows the 'Monitoring' tab selected in the 'Settings for PayrollJMS Server' interface. A callout box highlights the 'Monitoring' tab with the number '1'. Another callout box highlights the 'Monitoring' link in the navigation bar with the number '2'. The main content area displays a table titled 'Destinations(Filtered - More Columns Exist)'. The table has columns: Production, Consumption, Insertion, Name, Messages Current, Messages Pending, Messages High, Messages Received, and Destination Type. Two rows are listed: PayrollJMS!DepositUpdateTopic (Topic) and PayrollJMS!PayInquiryQueue (Queue).

Production	Consumption	Insertion	Name	Messages Current	Messages Pending	Messages High	Messages Received	Destination Type
			PayrollJMS!DepositUpdateTopic	0	0	0	4	Topic
			PayrollJMS!PayInquiryQueue	3	0	3	3	Queue

Quiz

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

1. Queue : Pub/Sub
2. Queue : P-to-P
3. Topic : Pub/Sub
4. Topic : P-to-P

Answer: 2, 3

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

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

Answer: 1, 2, 3

Summary

In this lesson, you should have learned about:

- Implementing Oracle WebLogic Server JMS
- Using the Administration Console to configure JMS administered objects
- Configuring persistent messages
- Using the WLS Administration Console to monitor JMS

Managing Transactions

Objectives

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

- Configure transactions using the console
- Monitor transactions using the console

Road Map

Configuring and Monitoring Transactions

- Configuring Transactions
- Using the Transaction Log
- Monitoring Transactions

What Is a Transaction?

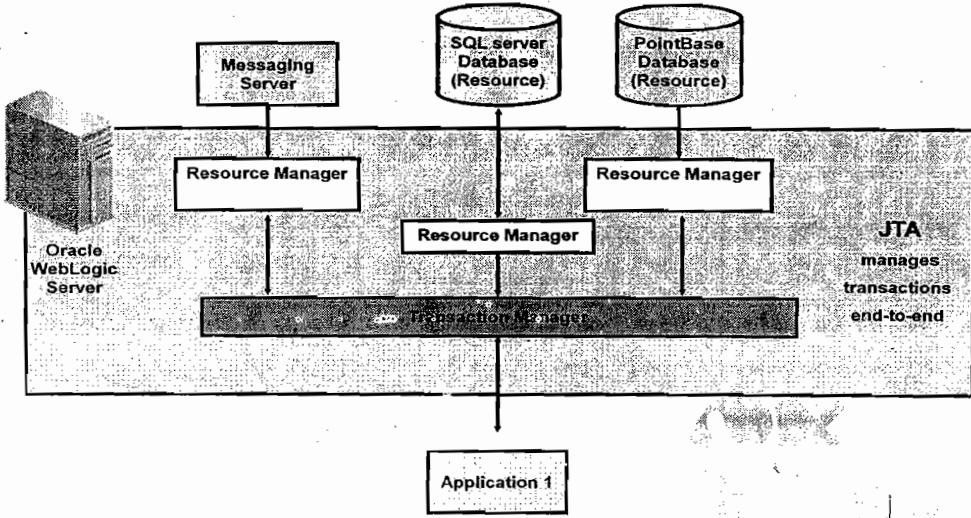
- A transaction is a mechanism to handle groups of operations as though they were one.
- Either all operations in a transaction occur or none occur at all.
- The operations involved in a transaction might rely on multiple servers and databases.

ACID Properties of a Transaction

A transaction is formally defined by the set of properties that is known by the acronym ACID.

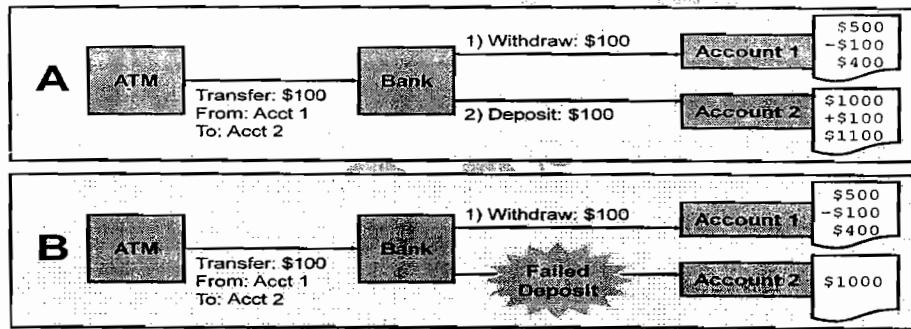
- Atomicity: A transaction is done or undone completely. In the event of a failure, all operations and procedures are undone, and all data rolls back to its previous state.
- Consistency: A transaction transforms a system from one consistent state to another consistent state.
- Isolation: Each transaction occurs independently of other transactions that occur at the same time.
- Durability: Completed transactions remain permanent, even during system failure.

Transaction Management



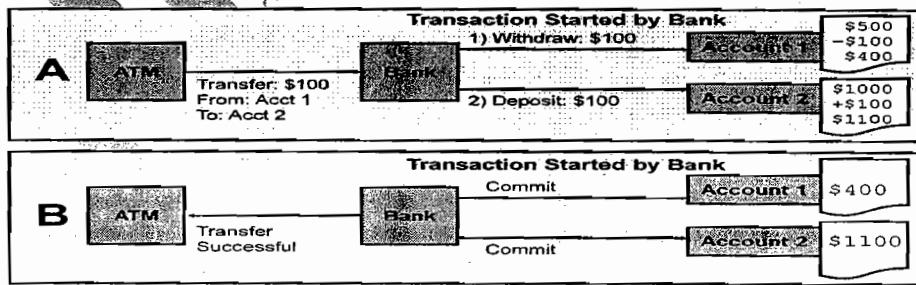
Transferring Without Transactions

- Successful transfer (A)
- Unsuccessful transfer (accounts are left in an inconsistent state) (B)



Successful Transfer with Transactions

- Changes within a transaction are buffered. (A)
- If a transfer is successful, changes are committed (made permanent). (B)



Unsuccessful Transfer with Transactions

- Changes within a transaction are buffered. (A)
- If a problem occurs, the transaction is rolled back to the previous consistent state. (B)

Do You Want to perfect every topic in Servlets

Attend

Servlets

(SCWCD Classes)

Mr. DURGA M.Tech

- ☛ Http Methods
- ☛ HttpServletRequest
- ☛ HttpServletResponse
- ☛ Servlet Life Cycle
- ☛ Web Application Directory Structure
- ☛ Web.xml
- ☛ WAR file
- ☛ ServletConfig
- ☛ ServletContext
- ☛ Servlet Scopes
- ☛ RequestDispatcher
- ☛ Session Management
- ☛ Filters
- ☛ Wrappers
- ☛ Listeners
- ☛ Web Security

Other than these... Nothing else in Servlets

SCWCD Servlets Syllabus

Section 1: The Servlet Technology Model

For each of the HTTP Methods (such as GET, POST, HEAD, and so on) describe the purpose of the method and the technical characteristics of the HTTP Method protocol, list triggers that might cause a Client (usually a Web browser) to use the method, and identify the HttpServlet method that corresponds to the HTTP Method.

Using the HttpServletRequest interface, write code to retrieve HTML form parameters from the request, retrieve HTTP request header information, or retrieve cookies from the request.

Using the HttpServletResponse interface, write code to set an HTTP response header, set the content type of the response, acquire a text stream for the response, acquire a binary stream for the response, redirect an HTTP request to another URL, or add cookies to the response.

Describe the purpose and event sequence of the servlet life cycle: (1) servlet class loading, (2) servlet instantiation, (3) call the init method, (4) call the service method, and (5) call destroy method.

Section 2: The Structure and Deployment of Web Applications

Construct the file and directory structure of a Web Application that may contain (a) static content, (b) JSP pages, (c) servlet classes, (d) the deployment descriptor, (e) tag libraries, (f) JAR files, and (g) Java class files; and describe how to protect resource files from HTTP access.

Describe the purpose and semantics of the deployment descriptor.

Construct the correct structure of the deployment descriptor.

Explain the purpose of a WAR file and describe the contents of a WAR file, how one may be constructed.

Section 3: The Web Container Model

For the ServletContext initialization parameters, write servlet code to access initialization parameters, and create the deployment descriptor elements for declaring initialization parameters.

For the fundamental servlet attribute scopes (request, session, and context), write servlet code to add, retrieve, and remove attributes, given a usage scenario, identify the proper scope for an attribute, and identify multi-threading issues associated with each scope.

Describe the Web container request processing model; write and configure a filter, create a request or response wrapper, and given a design problem, describe how to apply a filter or a wrapper.

Describe the Web container life cycle event model for requests, sessions, and web applications, create and configure listener classes for each scope life cycle, create and configure scope attribute listener classes, and given a scenario, identify the proper attribute listener to use.

Describe the RequestDispatcher mechanism, write servlet code to create a request dispatcher, write servlet code to forward or include the target resource, and identify and describe the additional request-scoped attributes provided by the container to the target resource.

Section 4: Session Management

Write servlet code to store objects into a session object and retrieve objects from a session object.

Given a scenario, describe the APIs used to access the session object, explain when the session object was created, and describe the mechanisms used to destroy the session object, and when it was destroyed.

Using session listeners, write code to respond to an event when an object is added to a session, and

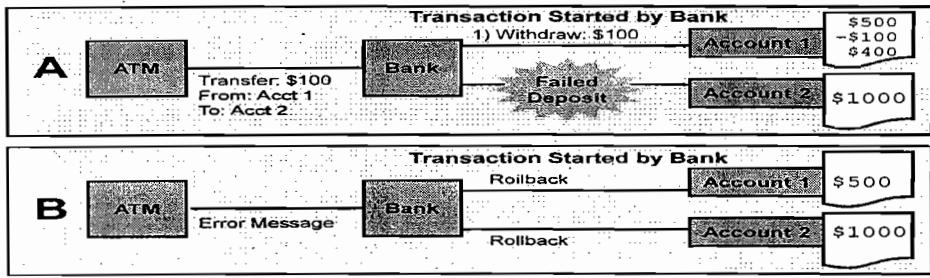
write code to respond to an event when a session object migrates from one VM to another.

Given a scenario, describe which session management mechanism the Web container could employ, how cookies might be used to manage sessions, how URL rewriting might be used to manage sessions, and write servlet code to perform URL rewriting.

Section 5: Web Application Security

Based on the servlet specification, compare and contrast the following security mechanisms: (a) authentication, (b) authorization, (c) data integrity, and (d) confidentiality.

In the deployment descriptor, declare a security constraint, a Web resource, the transport guarantee, the login configuration, and a security role.



Types of Transactions

- A local transaction deals with a single resource manager. Local transactions use the non-Extended Architecture (non-XA) interface between Oracle WebLogic Server and the resource manager.
- A distributed transaction coordinates or spans multiple resource managers.
- Global transactions can deal with multiple resource managers. Global transactions use the Extended Architecture (XA) interface between Oracle WebLogic Server and the resource managers.
 - You need to create non-XA or XA resources for local transactions. However, for global transactions, you need to create only XA resources.

The Two-Phase Commit Protocol

- The Two-Phase Commit (2PC) protocol uses two steps to commit changes within a distributed transaction.
 - Phase 1 asks the RMs to prepare to make the changes.
 - Phase 2 asks the RMs to commit and make the changes permanent or to roll back the entire transaction.
- A global transaction ID (XID) is used to track all the changes associated with a distributed transaction.

Extended Architecture Protocol

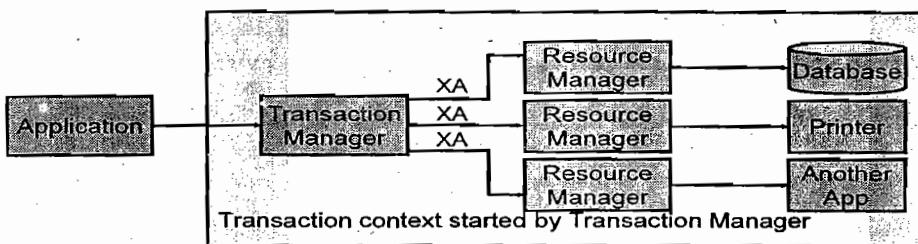
The Extended Architecture (XA) protocol:

- Is the interface that is used between WLS and the RMs
- Implements the 2PC protocol
- Allows programs to control the RMs that are involved in distributed transactions

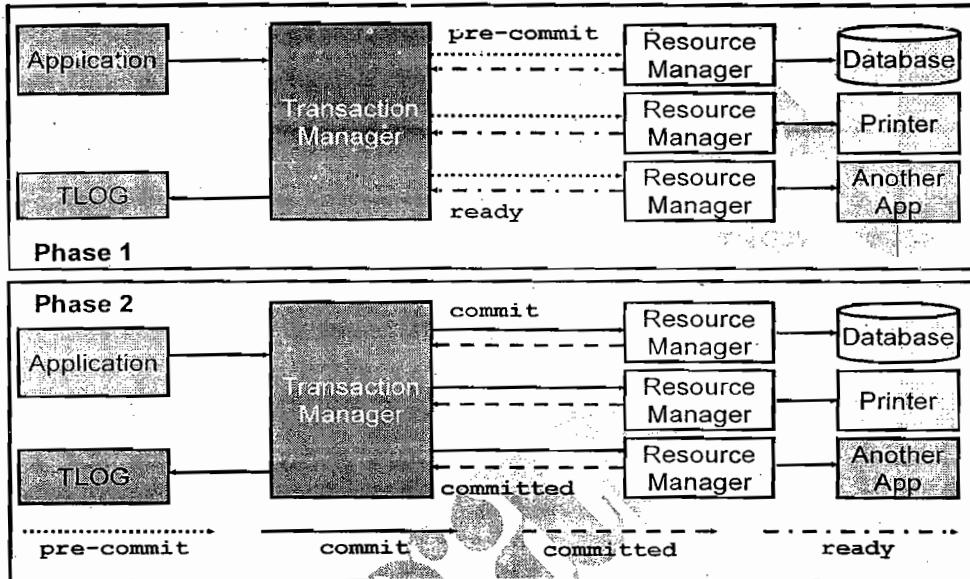


Transaction and Resource Managers

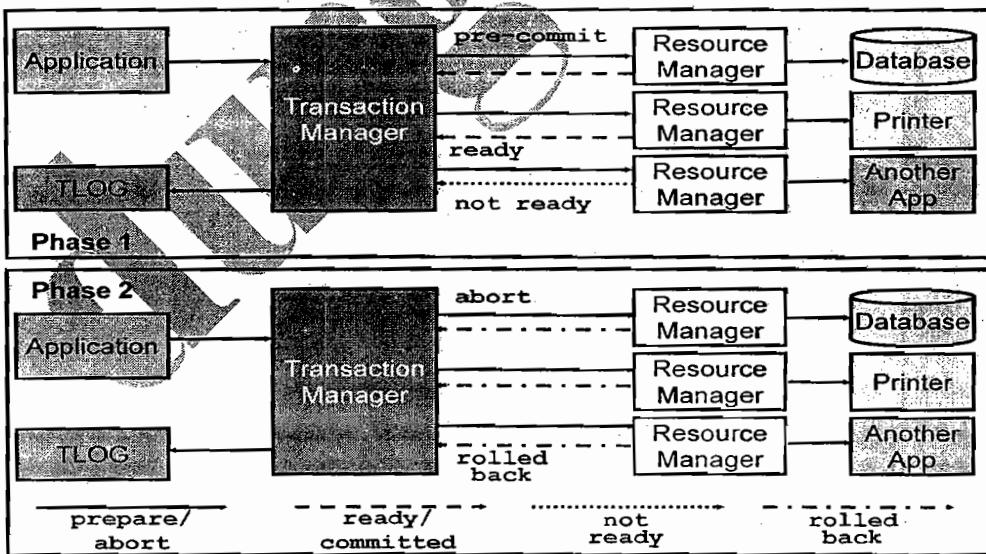
- A transaction manager coordinates multiple resource managers.
- The 2PC protocol is used to coordinate the transaction.
- The XA protocol implements 2PC.



Successful Two-Phase Commit



Unsuccessful Two-Phase Commit



Java Transaction API (JTA)

- WLS uses JTA to implement and manage transactions.
- WLS JTA provides the following support:
 - It creates a unique transaction identifier (XID).

- It supports an optional transaction name.
- It tracks objects involved in transactions.
- It notifies databases of transactions.
- It orchestrates 2PC using XA.
- It executes rollbacks.
- It executes automatic recovery procedures in the event of failure.
- It manages timeouts.

Configuring Transactions

The screenshot shows the Oracle WebLogic Server Administration Console. On the left, there is a tree view titled 'Domain Structure' with various service categories like Messaging, JDBC, Persistent Stores, etc. A red circle labeled '1' is drawn around the 'JTA' node in the tree. On the right, there is a configuration page for 'Java Transaction API (JTA)'. The top navigation bar has tabs: Configuration, Monitoring, Control, Security, Web Service Security, Notes, General, JTA, EJBs, Web Applications, Logging, Log Filters. A red circle labeled '2' is drawn around the 'JTA' tab. Below the tabs, a message says 'Use this page to define the Java Transaction API (JTA) configuration of this WebLogic'. There are several configuration fields with descriptions:

- Timeout Seconds:** 30. Description: The maximum transaction time limit for transactions in progress.
- Abandon Timeout Seconds:** 86400. Description: The maximum time after which a transaction will be abandoned if it fails to commit or rollback.
- Before Completion Iteration Limit:** 10. Description: The maximum number of iterations before completion for this WebLogic.
- Max Transactions:** 10000. Description: The maximum number of in-progress transactions in this WebLogic.
- Max Unique Name Statistics:** 1000. Description: The maximum number of unique names for statistics.

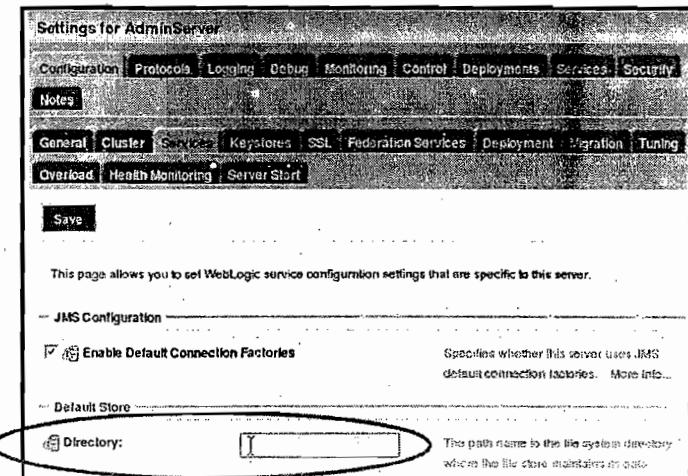
Configuring the Transaction Log

- Each server has a transaction log that stores information about committed transactions coordinated by the server that may not have been completed.
 - Oracle WebLogic Server uses the transaction log when recovering from system crashes or network failures.
- You cannot directly view the transaction log because the records are in a binary format and are stored in the default persistent store for the server.

Configuring the Transaction Log

- The .tlog files are stored in the default persistent store for the server.
- Directory represents the path name to the file system directory where the file store maintains its data files.

Copyright © 2005, Oracle. All rights reserved. Oracle is a registered trademark of Oracle Corporation and/or its affiliates.



JTA Configuration Options

Field	Description
Timeout Seconds	Specifies the time in which a transaction will timeout, if uncommitted
Abandon Timeout Seconds	Specifies the maximum time that a transaction manager will persist in attempting to complete a transaction during the second phase of the transaction
Before Completion Iteration Limit	Specifies the maximum number of cycles that the transaction manager will perform the beforeCompletion() synchronization callback for this Oracle WebLogic Server domain
Max Transactions	Specifies the maximum number of simultaneous in-progress transactions allowed on a server in the domain
Max Unique Name Statistics	Specifies the maximum number of unique transaction names for which statistics will be maintained

Field	Description
Checkpoint Interval Seconds	Specifies the interval at which the transaction manager creates a new transaction log file and checks all old transaction log files to see if they are ready to be deleted

Forget Heuristics	Specifies whether the transaction manager will automatically perform an XA Resource forget() operation for heuristic transaction completions
Unregister Resource Grace Period	Specifies the grace period, in seconds, that the transaction manager waits for transactions involving the resource to complete before unregistering a resource
Security Interoperability Mode	Specifies the security mode to use for XA calls in interdomain transactions

Creating XA Resources

The screenshot shows the 'JDBC Data Source Properties' dialog box. It includes fields for Name ('JDBC Data Source-0'), JNDI Name (empty), Database Type ('PointBase'), and Database Driver ('*PointBase's Driver (Type 4 XA) Versions:4.X,5.X'). The dialog has 'Next >', 'Finish', and 'Cancel' buttons at the top.

Creating Non-XA Resources

The screenshot shows the 'JDBC Data Source Properties' dialog box. It includes fields for Name ('JDBC Data Source-0'), JNDI Name (empty), Database Type ('PointBase'), and Database Driver ('*PointBase's Driver (Type 4) Versions:4.X,5.X'). The dialog has 'Next >', 'Finish', and 'Cancel' buttons at the top.

Creating Non-XA Resources

Create a New JDBC Data Source

Transaction Options

You have selected non-XA JDBC driver to create database connection in your new data source.

Does this data source support global transactions? If yes, please choose the transaction protocol for this data source.

Supports Global Transactions

Select this option if you want to enable non-XA JDBC connections from the data source to participate in global transactions using the *Logging Last Resource* (LLR) transaction optimization. Recommended in place of Emulate Two-Phase Commit.

Logging Last Resource

Select this option if you want to enable non-XA JDBC connections from the data source to emulate participation in global transactions using JTA. Select this option only if your application can tolerate heuristic conditions.

Emulate Two-Phase Commit

Select this option if you want to enable non-XA JDBC connections from the data source to participate in global transactions using the one-phase commit transaction processing. With this option, no other resources can participate in the global transaction.

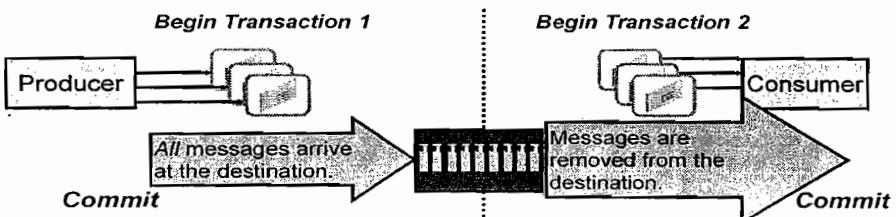
One-Phase Commit

Logging Last Resource

- You can configure a JDBC data source to enable the Logging Last Resource (LLR) transaction optimization, which:
 - Enables one non-XA resource to participate in a global transaction
 - Has improved performance and the same ACID guarantee as XA
- The LLR optimization improves performance by:
 - Removing the need for an XA JDBC driver to connect to the database. XA JDBC drivers are typically inefficient compared to non-XA JDBC drivers.
 - Reducing the number of processing steps to complete the transaction, which also reduces network traffic and I/O
 - Removing the need for XA processing at the database level (if the database is the one non-XA resource)

Transacted Messaging

- A JMS client can use 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.



Inter-Domain Transactions

- Oracle WebLogic Server supports global transactions across domains on different versions of Oracle WebLogic Server.
- To enable Oracle WebLogic Server domains for inter-domain transactions, do the following:
 - Enable trust between the different domains.
 - Ensure that each contributing XA resource has a unique name.
 - Ensure that only one of the participating resources in the distributed transaction can emulate the two-phase commit or XA protocol.

Enabling Trust Among Different Domains

The screenshot shows the 'Security' tab of the configuration interface. It displays settings for a 'Default Realm' named 'myrealm'. Key options include 'Anonymous Admin Lookup Enabled' (unchecked), 'Cross Domain Security Enabled' (unchecked), and a list of 'Excluded Domain Names' which is currently empty. On the right, a modal dialog box titled 'Advanced' is open, showing the 'Security Interoperability Mode' dropdown set to 'default'. Below it, there are fields for 'Credential' and 'Confirm Credential', both containing the value 'weblogic/weblogic'. A 'Save' button is visible at the bottom left of the main screen.

Monitoring Transactions

The screenshot shows the 'Monitoring' tab of the configuration interface. It displays a summary of all transaction information for the server. Key sections include:

- Transactions Total Count:** 0 The total number of transactions processed. This total includes all committed, rolled back, and heuristic transaction completions since the server was started. [More Info...](#)
- Transactions Committed Total Count:** 0 The total number of transactions committed since the server was started. [More Info...](#)
- Transactions Rolled Back for Timeout Total Count:** 0 The number of transactions that were rolled back due to a timeout expiration. [More Info...](#)
- Transactions Rolled Back for Resource Errors Total Count:** 0 The number of transactions that were rolled back due to a resource error. [More Info...](#)
- Transactions Rolled Back for Application Errors Total Count:** 0 The number of transactions that were rolled back due to an application error. [More Info...](#)

A note at the top states: 'This page shows the summary of all transaction information for all resource types on the server.'

Monitoring Transactions by Resource

For a particular resource, the console monitors transactional outcomes:

- The number of transactions attempted
- The number of commits or rollbacks
- The number of heuristic outcomes

Viewing Transaction Statistics for XA Resources

The screenshot shows the 'Transactions' tab for XA Resources in the Oracle WebLogic Server Administration Console. The top navigation bar includes tabs for Configuration, Protocols, Logging, Debug, Monitoring, Control, Deployments, Services, Security, and Notes. Sub-tabs under Monitoring include General, Health, Channels, Performance, Threads, Timers, Workload, Security, Default Store, and JMS. The 'XA Resources' tab is selected. Below the tabs, a message says 'Use this page to monitor transactions coordinated by this server.' A link 'Customize this table' is present. The main area displays a table titled 'XA Resources' with columns: Name, Transactions, Commits, Rollbacks, Timeout Rollbacks, Resource Rollbacks, Application Rollbacks, System Rollbacks, Heuristics, Transaction Abendo, and Total Cc. The table is currently empty, showing 'Showing 0 to 0 of 0'. Navigation links 'Previous' and 'Next' are at the bottom.

Summary

In this lesson, you should have learned how to:

- Configure transactions using the console
- Monitor transactions in Oracle WebLogic Server

Securing WLS Resources and Applications

Objectives

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

- Describe WLS security architecture
- Configure users, groups, and roles
- Configure security realms
- Secure Web applications with declarative security
- Configure policies and SSL
- Create and manage certificates
- Protect WLS against several types of attacks

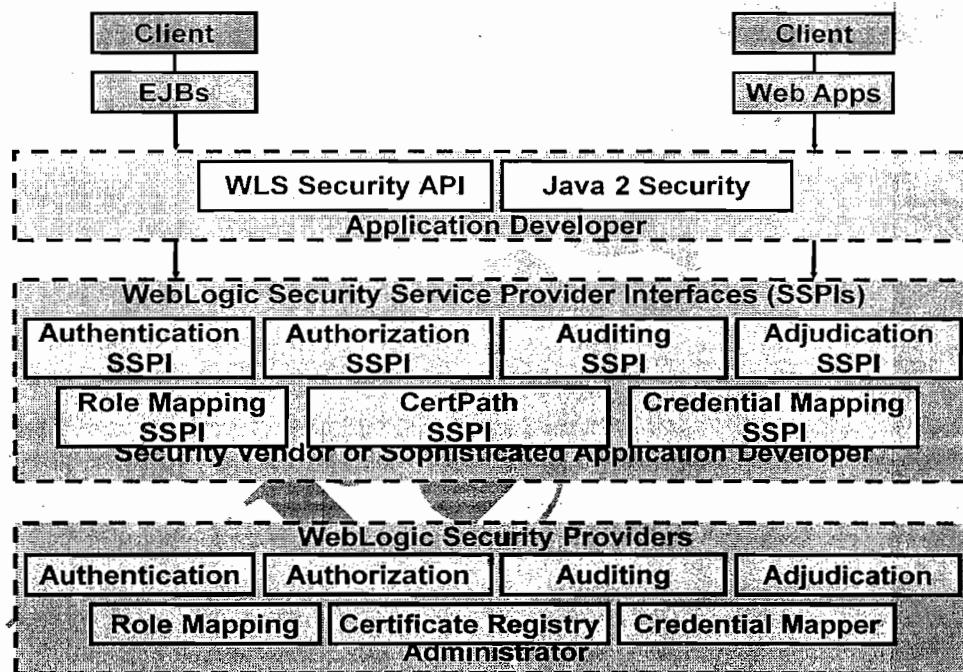
Road Map

- WLS Security Architecture Overview
 - Security Architecture
 - Security Terminology
 - Compatibility with Previous WLS Versions
- Users and Groups
- Protecting Application Resources
- Protecting Communications
- Protecting Against Attacks

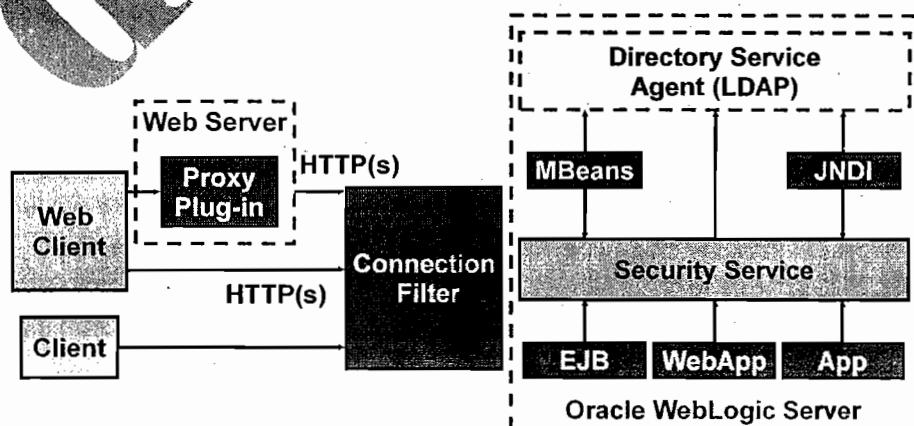
Architecture Goals

- Using Java standards (where applicable), create an architecture that unifies security enforcement and present it as a service to other components.
- Provide a security framework that allows the integration of third-party security products with minimum restrictions on functionality.
- Provide consistent and unified protection for all resources hosted on Oracle WebLogic Server:
 - EJBs
 - Web applications (Servlets, JSP)
 - Web services
 - Miscellaneous J2EE Resources (RMI objects JDBC, JNDI, and MBeans)

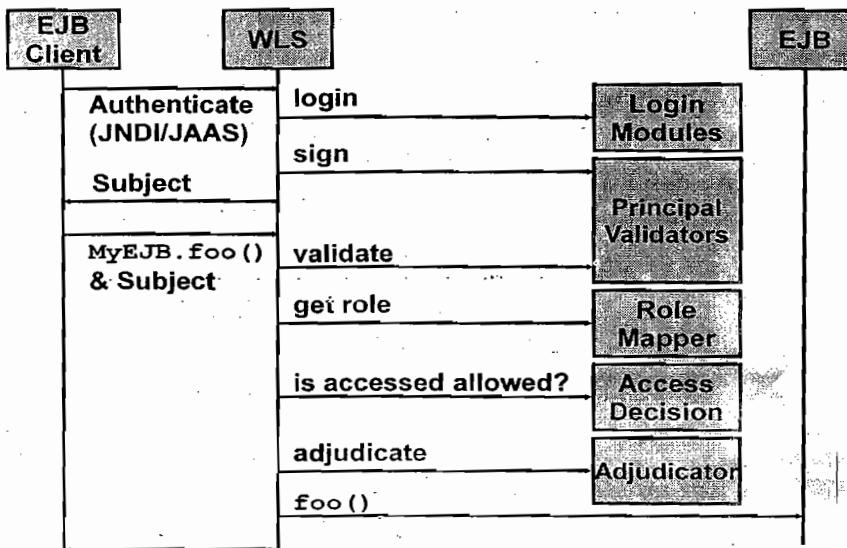
Security Architecture



Process Architecture



Security Services



Authentication Providers

- An authentication provider uses LoginModules to authenticate users within a security realm.
- An authentication provider transports identity information and makes it available to components with subjects.
- The Principal Validation provider provides additional protection by signing and verifying the authenticity of the principals.
- Identity Assertion providers, as a LoginModule, map a valid token to an Oracle WebLogic Server user.

Authorization Providers

- An Authorization provider is a process that is used to control the interactions between users and resources based on user identity.
- The Role Mapping provider supplies the security role with information to determine whether access is allowed for role-based resources, such as EJBs.
- The Access Decision component answers the question, "Is access allowed?"
- The Access Decision returns a result of PERMIT, DENY, or ABSTAIN.
- The Adjudication provider weighs the results that multiple access decisions return to determine the final decision.

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 checking 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 checking that the server's certificate and public ID are valid and are issued by a CA.

Credential Mapping

- The credential mapping process is initiated when application components must access a legacy system authentication mechanism 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.

Auditing

- Auditing provides a trail of activity.
- The Auditing provider is used to log activity before and after security operations.
- The default auditing provider records the event data that is associated with the security requests and the outcome of the requests.

```
17 PM> <Severity =FAILURE> <<<Event Type = Authentication Audit Event><rbrown>
      <AUTHENTICATE>>> Audit Record End #####
##### Audit Record Begin <Feb 7, 2004 8:15:23 PM> <Severity =FAILURE> <<<Event Type = Authentication Audit Event>
      <paulc><AUTHENTICATE>>> Audit Record End #####
##### Audit Record Begin <Feb 7, 2004 8:15:27 PM> <Severity =SUCCESS> <<<Event Type = Authentication Audit Event>
      <paulc><AUTHENTICATE>>> Audit Record End #####
```

Security Realms

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

What Is LDAP?

The Lightweight Directory Access Protocol:

- Is derived from X.500
- Provides a hierarchical lookup service
- Supports sophisticated searching
- Can be secured via SSL

Embedded LDAP Server

- In WLS, users, groups, and authorization information is stored in an embedded LDAP server.
- Several properties can be set to manage the LDAP server, including:
 - Credentials
 - Backup settings
 - Cache settings
 - Replication settings

Configuring Embedded LDAP

Settings for dizzyworld

Configuration | Monitoring | Control | Security | Web Service Security | Notes

General | Filter | Unlock User | Embedded LDAP | Roles | Policies

<p>Credential: <input type="text" value="password123"/></p> <p>Confirm Credential: <input type="text" value="password123"/></p> <p>Backup Hour: <input type="text" value="23"/></p> <p>Backup Minute: <input type="text" value="5"/></p> <p>Backup Copies: <input type="text" value="7"/></p> <p><input type="checkbox"/> Cache Enabled</p> <p>Cache Size: <input type="text" value="32"/></p> <p>Cache TTL: <input type="text" value="60"/></p> <p><input type="checkbox"/> Refresh Replica At Startup</p>	<p><input type="checkbox"/> Master First</p> <p><input type="checkbox"/> Timeout: <input type="text" value="0"/></p> <p><input type="checkbox"/> Anonymous Bind Allowed</p>
---	---

Connecting Using an External LDAP Browser

Settings for dizzyworld

Configuration | Monitoring | Control | Security | Web Service Security | Notes

General | Filter | Unlock User | Embedded LDAP | Roles | Policies

Save

This page allows you to configure the embedded LDAP server for this WebLogic Server.

Credential:

Confirm Credential:

The credentials connect to 1 info...

Change this password to allow an external browser to connect. The password can be different from the admin password.

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

Configuring New Users

*forgot to bring - my
info*

*not any
in mind.*

Settings for myrealm

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

Groups

This page displays information about each user that has been configured in this security realm.

Customize this table

Users

User	Provider	Role	Action
Sarah	DefaultAuthenticator	Employee	
Tim	DefaultAuthenticator	Employee	
Colleen	DefaultAuthenticator	Employee	
Joe	DefaultAuthenticator	Manager	

Showing 1 to 2 of 2. Prev | Next

I forgot to bring - my info

time

Create a New User

OK | **Cancel**

User Properties

The following properties will be used to identify your new User.

* Indicates required fields

Name:

What would you like to name your new User?

Description:

Please choose a provider for the user.

Provider:

The password is associated with the login name for the new User.

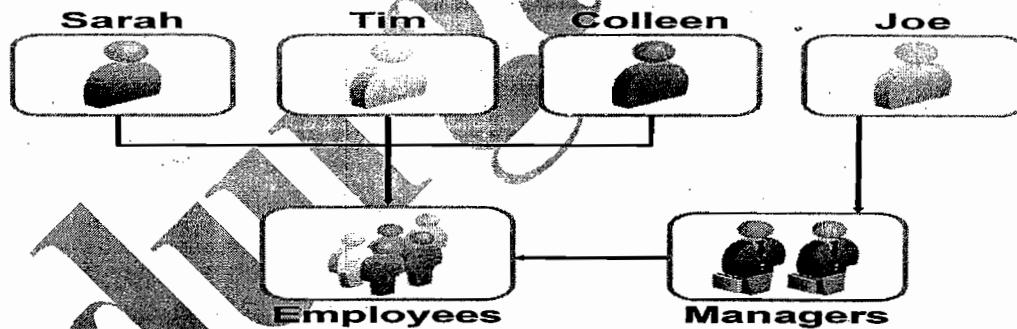
Password:

Confirm Password:

Groups

WLS provides the flexibility to organize groups in various ways:

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



Configuring New Groups

*forgot to bring - my
info*

Settings for myrealm

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

Groups

This page displays information about each group that has been configured in this security realm.

Customize this table

Groups

Group	Provider	Action
Employees	DefaultAuthenticator	
Managers	DefaultAuthenticator	

Showing 1 to 10 of 14. Prev | Next

I forgot to bring - my info

time

Create a New Group

OK | **Cancel**

Group Properties

The following properties will be used to identify your new Group.

* Indicates required fields

Name:

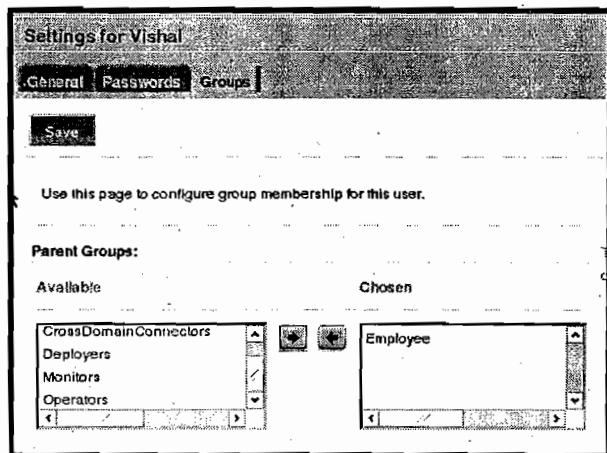
What would you like to name your new Group?

Description:

Please choose a provider for the group.

Provider:

Adding Groups to Users



Roles

- A role refers to a set of users who have the same permissions.
- 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: AppTester, Anonymous, Admin, Operator, Deployer, and Monitor.
- Roles defined in deployment descriptors can be inherited.
 - Occurs at deployment time
 - Can be disabled
- You can manage role definitions and assignments without editing deployment descriptors or redeploying the application.

Configuring New Roles

Settings for browser-starter

Overview | Deployment Plan | Configuration | Security | Targets | Control | Testing | Monitoring | Notes

Application Scope: URL Patterns

Roles Policies

This page summarizes the security roles that can be used in the policy for this Web application. These roles cannot be used in policies for individual resources or URL patterns within this application.

Note:

If you are using the DD Only security model for this deployment, then you cannot use the Administration Cons to modify its security roles.

Customize this table

Stand-Alone Web Application Scoped Roles

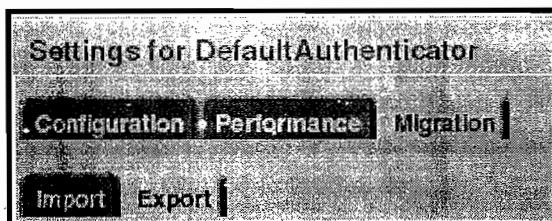
New Delete

Showing 0 to 0 of 0 Previous

Migrating Security Data

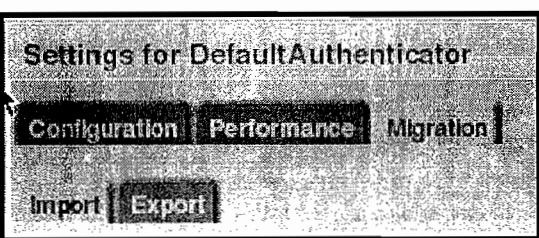
- You can export users and groups, security policies, security roles, or credential maps between security realms or domains.
- It is useful, for example, in transitioning from development to QA to production.
- You can use migration constraints (key/value pairs) to specify the export/import options.
- Currently the system supports migrating only security data between the WLS security providers.

Exporting the WLS Default Authenticator Provider



The screenshot shows the 'Settings for DefaultAuthenticator' page with the 'Export' tab selected. The main content area displays the 'Export' configuration screen. It includes fields for 'Export Format' (set to 'DefaultAtn'), 'Export File' (set to '/u01/app/oracle/product/Middleware/user_projects/domains/diz'), and an 'Overwrite' checkbox. Below these are sections for 'Supported Export Constraints' (listing 'users, groups, passwords') and 'Export Constraints (key=value)'. A 'Save' button is at the bottom.

Importing into a Different Domain



The screenshot shows the 'Settings for DefaultAuthenticator' page with the 'Import' tab selected. The main content area displays the 'Import' configuration screen. It includes fields for 'Import Format' (set to 'DefaultAtn'), 'Import File' (set to '/u01/app/oracle/product/Middleware/user_projects/domains/diz'), and an 'Overwrite' checkbox. Below these are sections for 'Supported Import Constraints' (listing 'None') and 'Import Constraints (key=value)'. A 'Save' button is at the bottom.

J2EE Declarative Security

- Is a means to describe an application's access control in a form that is external to the application
- Involves defining security roles and constraints on the Web application resources
- Uses lazy authentication to protect application resources
- Is implemented in XML-based deployment descriptors
- Applies to all types of applications

Using Deployment Descriptors

The security realm definition determines how the deployment descriptors are used.

Settings for myrealm

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

General | RDBMS Security Store | User Lockout | Performance

Name: myrealm
Security: DD Only
Model Default:
 Combined Role Mapping Enabled
 Use Authorization Providers to Protect JMX Access
Advanced
 Check Roles: Web applications and EJBs protected in DD
and Policies:
When Deploying: Initialize roles and policies from DD
Web Applications

Protecting Web Applications

To protect a Web Application with declarative security, perform the following:

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/groups in the WLS security realm.
5. Set up an authentication mechanism.

Defining Security Roles

- Define the types of users that exist in your Web application.
- Use the web.xml deployment descriptor to define security roles.

```
Defining roles:  
<security-role>  
    <role-name>Users</role-name>  
</security-role>  
  
<security-role>  
    <role-name>Managers</role-name>  
</security-role>
```

<web.xml>

Determining Protected Resources

- Web resources are defined based on URL patterns.
- URL patterns provide a flexible way to define a single resource or a group of resources.

- Example URL Patterns:

URL Pattern	Role Name
/*	Some Role Name (that is, director)
/*.jsp	" "
/EastCoast/*	" "

Mapping Roles to Resources

- Apply security constraints to specified resources in your Web application.
- Users must be authenticated when they access resources by these URL patterns.

Configuring security constraints:

```
<security-constraint>
  <web-resource-collection>
    <web-resource-name>East Coast Sales</web-resource-name>
      <url-pattern>/EastCoast/*</url-pattern>
      <http-method>POST</http-method>
      <http-method>GET</http-method>
    </web-resource-collection>
```

<web.xml>

Mapping Roles to Resources

Define the roles that may access this collection of resources.

Configuring security constraints:

```
<security-constraint>
  ...
  <auth-constraint>
    <role-name>Users</role-name>
    <role-name>Managers</role-name>
  </auth-constraint>
</security-constraint>
```

<web.xml>

Mapping Roles to Users in Realms

- Use weblogic.xml to map your Web application roles to entities in the WebLogic security realm.
- Map to Groups or individual Users.

Assigning roles:

```
<security-role-assignment>
  <role-name>Users</role-name>
  <principal-name>employees</principal-name>
</security-role-assignment>

<security-role-assignment>
  <role-name>Managers</role-name>
  <principal-name>mary</principal-name>
  <principal-name>paul</principal-name>
</security-role-assignment>
```

<weblogic.xml>

Setup Authentication

Configure how a Web application determines the security credentials of users:

- BASIC: Web browser displays a dialog box.
- FORM: Use a custom HTML form.
- CLIENT-CERT: Request a client certificate.

Configuring authentication:

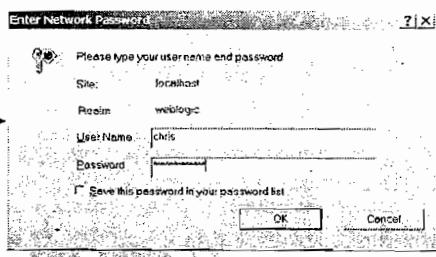
```
<login-config>
    <auth-method>BASIC, FORM, or CLIENT-CERT</auth-method>
    <form-login-config>
        <form-login-page>login.jsp</form-login-page>
        <form-error-page>badLogin.jsp</form-error-page>
    </form-login-config>
</login-config>
```

For type FORM only

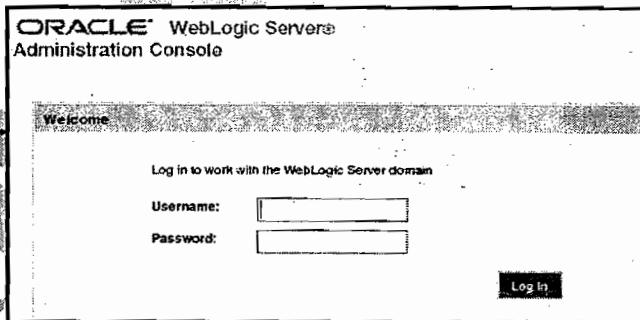
<web.xml>

Authentication Examples

BASIC Authentication



FORM-based Authentication



Policies and Roles

- Security roles:
 - Are an abstraction of users and groups
 - Can be determined dynamically for different resources
- Security policies:
 - Comprise rules and conditions that users and groups must adhere to, to be granted or denied authorization
 - Implement parameterized authorization

Defining Policies and Roles for Web Resources

Settings for browser-starter

Overview Deployment Path Configuration Security Targets Control Testing Monitoring Notes

Application Scopes URL Patterns

Roles Policies

This page summarizes the security policies that secure specific URL patterns in this stand-alone web application.

If you are using the "Only One Oracle Home" security model for this deployment, then you cannot use the Administration Console to modify its security policies.

Customize this table

Stand-Alone Web Application URL Patterns

Edit URL Pattern Rule

Save Reset Cancel

Edit Arguments

On this page you will fill in the arguments that pertain to the predicate you have chosen.

The earliest permissible time in the format of "MM/DD/YYYY", E.g., "12/31/00 AM".

Starting time:

The latest permissible time in the format of "MM/DD/YYYY", E.g., "12/31/00 AM".

Ending time:

The time offset of GMT (either an "GMT+hh:mm" or "GMT-hh:mm"). E.g., "GMT+0:00".

GMT offset:

OK Cancel Reset

Defining Policies and Roles for Other Resources

You can define roles and policies on other resources, such as JDBC and JMS.

Settings for myrealm

Configuration Users and Groups Roles and Policies Credential Mappings Providers Migration

Realm Roles Realm Policies

Roles

Edit Role

Showing 1 to 7 of 7

Name	Resource Type	Role Policy
Deployments		-
Domain		
Global Roles		
JCOM		
JDBC		
JMS		
Servers		

Policy Conditions

These conditions determine the access control to your JDBC data source.

Add Conditions Combine Uncombine Move Up Move Down Remove Negate

No Policy Specified

Settings for JDBC Data Source-O

Configuration Targets Monitoring Control Security Notes

Roles Policies Credential Mappings

Back Next Previous Cancel

Choose a Predicate

Choose the predicate you wish to use as your new condition

The predicate list is a list of available predicates which can be used to make up

Predicate List: Access occurs between specified hours

Settings for JDBC Data Source-O

Configuration Targets Monitoring Control Security Notes

Roles Policies Credential Mappings

Back Previous Finish Cancel

Edit Arguments

On this page you will fill in the arguments that pertain to the predicate you have chosen.

The earliest permissible time in the format of "MM/DD/YYYY", E.g., "12/31/00 PM".

Starting time:

The latest permissible time in the format of "MM/DD/YYYY", E.g., "12/31/00 PM".

Ending time:

The time offset of GMT (either an "GMT+hh:mm" or "GMT-hh:mm"). E.g., "GMT+0:00".

USA would be "GMT-5:00".

GMT offset:

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

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

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



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 WebLogic Server SSL Requirements

To enable Oracle WebLogic Server SSL, you must:

- Obtain an appropriate digital certificate
- Install the certificate
- Configure SSL properties
- Configure two-way authentication (if desired)
 - SSL impacts performance.

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 the display certificate and keystore contents.

Obtaining a Digital Certificate:

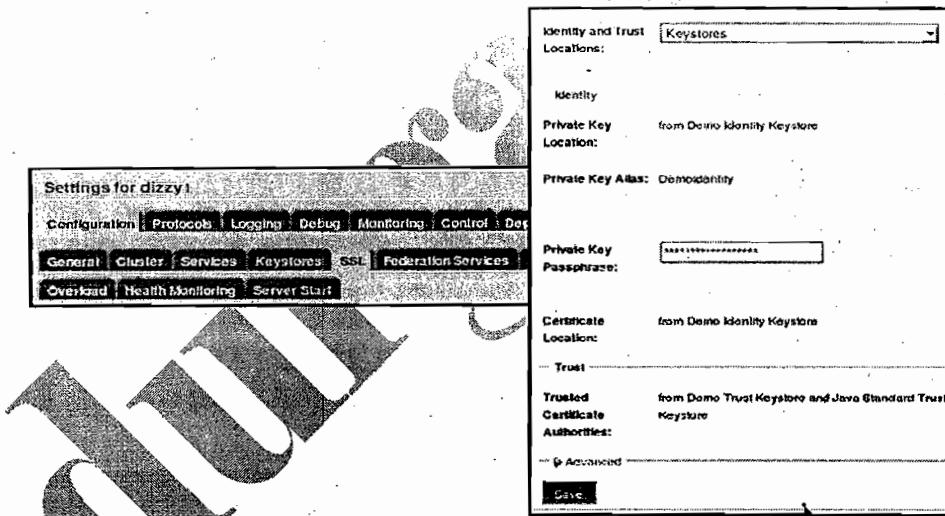
keytool Examples

```
Generate a new self-signed digital certificate
keytool -genkey -alias dwkey -keyalg RSA -keysize 512
-DK:Keystore dwkey keystore.jks
-DK:Keystorepw dwkeypass
-DK:IdentityAlias dwidentity
-DK:IdentityPass dwidentitypass

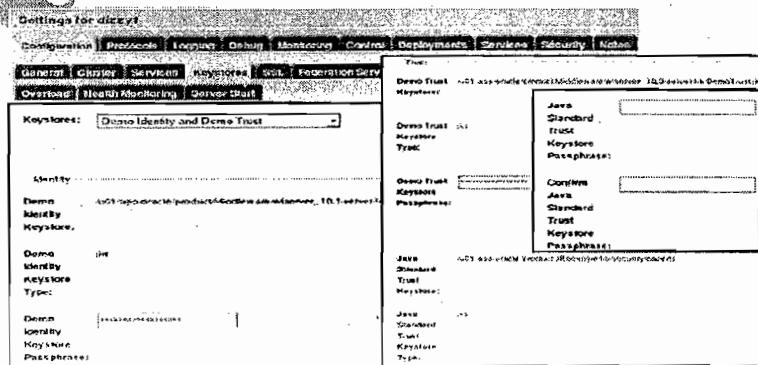
Generate a CSR
keytool -certreq -alias dwkey -file dw_cert_request.pem
-DK:Keypass dwkeypass -keystoreidw_identity.jks
-DK:Storepass dwstorepass

Import a signed certificate reply from a CA
keytool -import -alias dwkey -file dw_cert_reply.pem
-DK:Keypass dwkeypass -keystoreidw_identity.jks
-DK:Storepass dwstorepass
```

Configuring SSL for an Oracle WebLogic Server



Configuring Keystores

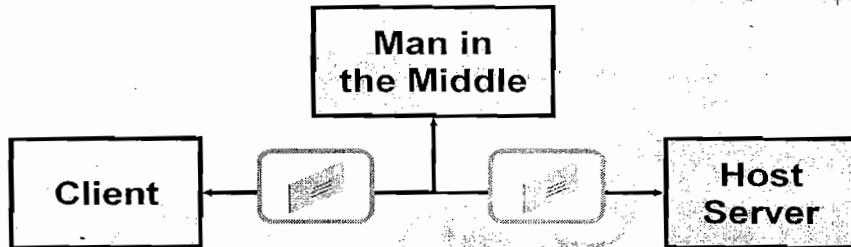


Protecting Against Attacks

- WLS can help protect applications against several attacks:
 - Man-in-the-middle attacks
 - Denial of service (DoS) attacks
 - Large buffer attacks
 - Connection starvation attacks
- The slides that follow detail the countermeasures that WLS provides to address these attacks.

Man-in-the-Middle Attacks

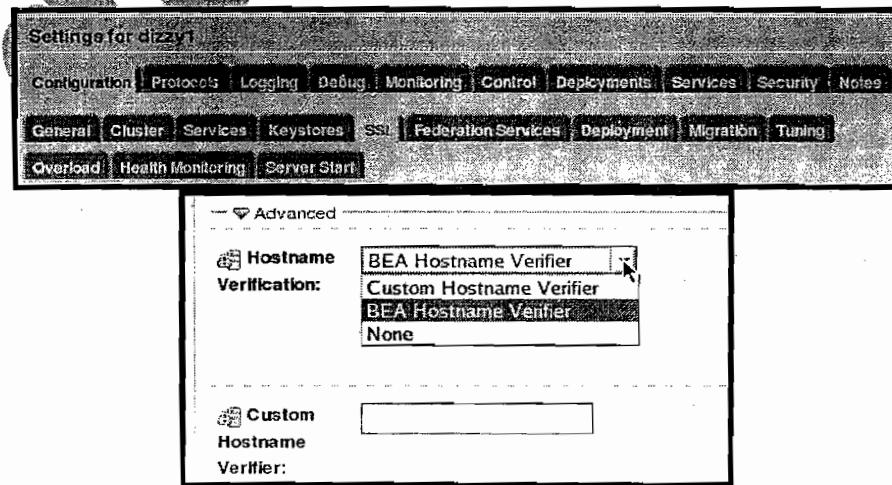
- In the “man-in-the-middle” attack, a third party poses as a destination host intercepting messages between the client and the real host.
- Instead of issuing the real destination host’s SSL certificate, the attacker issues his or her own hoping that the client would accept it as being from the real destination host.



Man-in-the-Middle Countermeasures

- The “man-in-the-middle” attacks can be resisted by using a Host Name Verifier.
- A Host Name Verifier validates that the host to which an SSL connection is made is the intended or authorized party.
- WLS provides a Host Name Verifier by default.
- A custom Host Name Verifier can be created by implementing the interface `weblogic.security.SSL.HostnameVerifier`

Configuring Hostname Verifier



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

DoS Countermeasures

Harden WLS against “denial of service” attacks by:

- Filtering incoming network connections
- Configuring consumable WLS resources with the appropriate threshold and quotas
- Limiting access to configuration information and backing up configuration files
- Preventing unauthorized access by protecting passwords against password-guessing attacks

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
 - Restrict the 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.

Connection Filter

The screenshot shows the 'Connection Filter' settings page for the 'dizzyworld' domain. The top navigation bar includes tabs for Configuration, Monitoring, Control, Security, Web Service Security, Notes, General, Filter, Unlock User, Embedded LDAP, Roles, and Policies. A 'Save' button is located at the top left. The main content area contains the following sections:

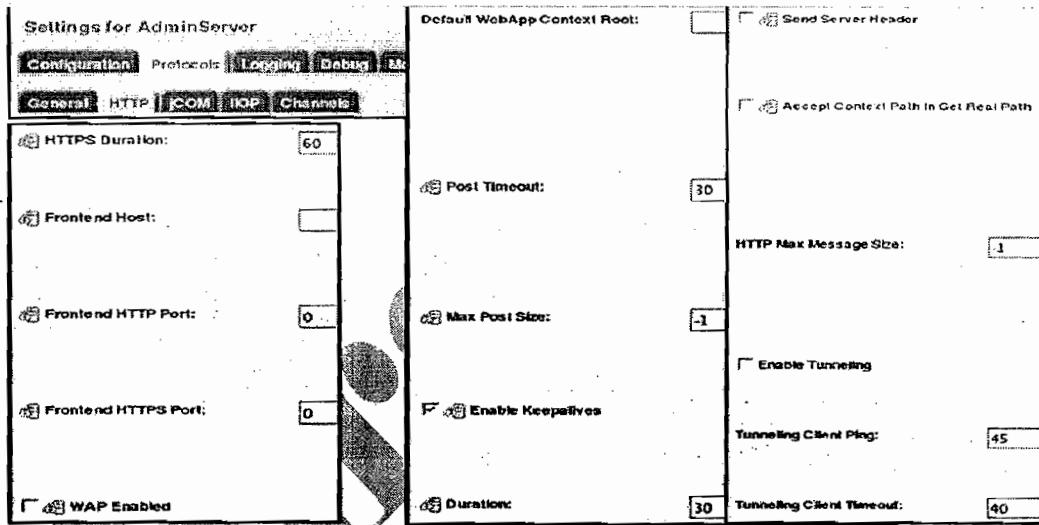
- Connection Logger Enabled:** A checkbox labeled 'Connection Logger Enabled' is checked, with a note explaining it specifies a domain shared connection logger.
- Connection Filter:** A dropdown menu labeled 'Connection Filter:' is shown, with a note explaining it is the name of the connection (WebLogic interface), connection, or connection logger.
- Connection Filter Rules:** A large text input field labeled 'Connection Filter Rules:' is present, with a note explaining it is the rules implemented by the interface, implemented specifications, and default implementation.

Consuming WLS Resources

- 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.

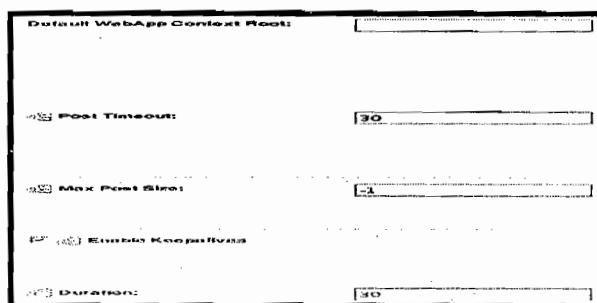
Large Buffer Attacks

- Individuals can try to 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.



Connection Starvation

- Individuals can try to 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.



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, 2007 2:42:36 PM EDT> <Notice> <Security> <BE0-090078> <User john Doe in security realm myrealm has had 5 invalid login attempts, locking account for 30 minutes.>
```

Configuring User Lockout

Settings for myrealm

Configuration | User and Groups | Roles and Policies | Credential Mgmt

General | RDBMS Security Store | User Lockout | Performance

Lockout Enabled

Lockout Threshold: 5

Lockout Duration: 30

Lockout Reset Duration: 5

Lockout Cache Size: 5

Lockout GC Threshold: 400

Save

Unlocking Users

Settings for dizzyworld

Configuration | Monitoring | Control | Security | Web Service Security | Notes

General | Filter | **Unlock User** | Embedded LDAP | Roles | Policies

If a user unsuccessfully attempts to log into a WebLogic Server server more than the configured number of times, then they are locked out of further access.

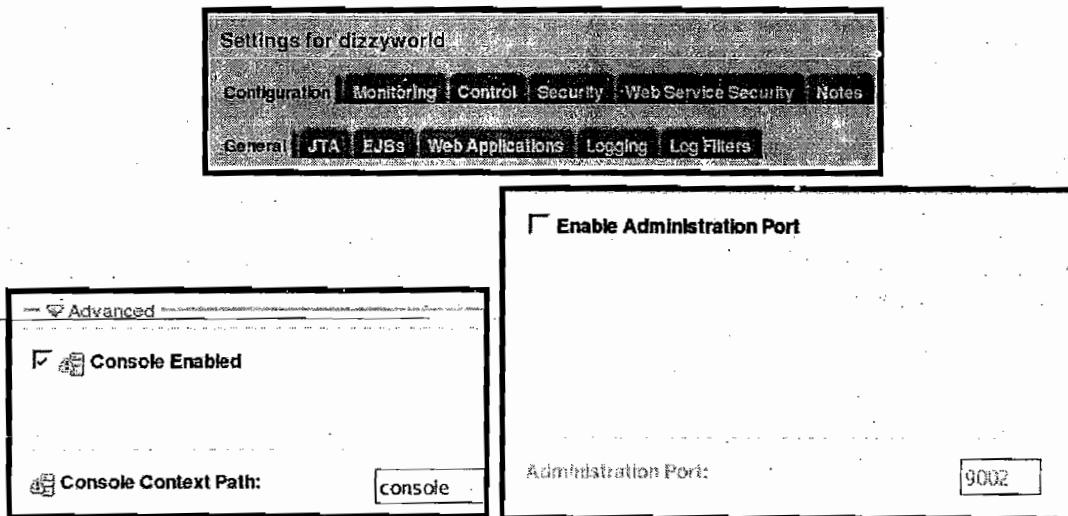
This page allows you to unlock a locked user so that they can log in again.

Unlock User: Name of a user to unlock

Save

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).



Summary

In this lesson, you should have learned how to:

- Use the WLS security architecture
- Configure users, groups, and roles
- Configure security realms
- Secure Web applications with declarative security
- Configure policies and SSL
- Create and manage certificates
- Protect WLS against several types of attacks

Advanced Deployment

Objectives

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

- Configure an application for multiple development environments
- Create a deployment plan
- Stage a deployment plan
- Use production redeployment

Road Map

- Deployment Plans
- Staged Deployment
- Production Redeployment

What Is a Deployment Plan?

- It is an optional XML document that resides outside an application archive.
- It configures an application for deployment to a specific WLS environment.
- It is created and owned by administrators or developers for a particular environment.

Deployment Plans

A JavaEE deployment plan:

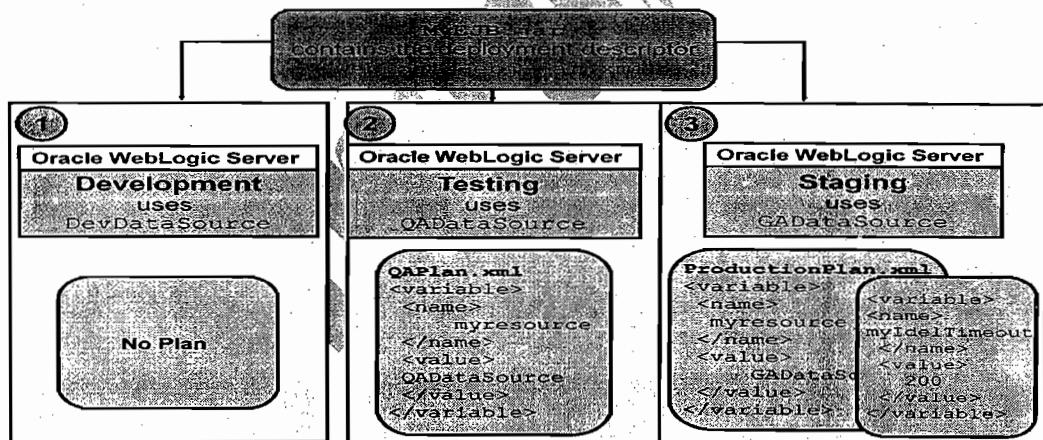
- Is an XML file that is associated with an application
- Resides outside an application archive
- Sets or overrides the values in the JavaEE deployment descriptors
- Allows a single application to be easily customized to multiple deployment environments



Deployment Plan: Advantages

- Works by setting or overriding the deployment property values that are defined in an application's WLS deployment descriptor
- Helps to easily modify an application's WLS configuration for deployment into different multiple WLS environments without modifying the deployment descriptor files that are included in the application archive
- Enables an application to be deployed to multiple domains or to multiple target servers and clusters that have a different configuration within the same domain

Configuring an Application for Multiple Deployment Environments



Sample Deployment Plan

```
<?xml version="1.0" encoding="UTF-8"?>
<deployment-plan xmlns="http://www.bea.com/ns/weblogic/100"
    xmlns:xsd="http://www.bea.com/ns/xsd/2001/WLSchema-instance"
    xsd:schemaLocation="http://www.bea.com/ns/weblogic/90/Weblogic-deployment-plan.xsd" global-variables="false">
    <application-name>myApp</application-name>
    <variable-definitions>
        <variable>
            <name>myresource</name>
            <value>CADatasource</value>
        </variable>
    </variable-definitions>
    <values>
        <value>
            <variable>myresource</variable>
            <value>myDataSource</value>
        </value>
    </values>
    <module-overrides>
        <module-name>WebArchive</module-name>
        <module-type>jar</module-type>
        <module-descriptor>external-override</module-descriptor>
        <root-element>weblogic-web-app</root-element>
        <uri>WEB-INF/web.xml</uri>
        <uri>WEB-INF/context-root</uri>
        <uri>WEB-INF/deployment-plan</uri>
        <uri>WEB-INF/weblogic.xml</uri>
        <uri>WEB-INF/weblogic-application.xml</uri>
        <uri>WEB-INF/weblogic-ejb-jar.xml</uri>
        <uri>WEB-INF/ejb-jar.xml</uri>
        <uri>WEB-INF/resource-bundles.xml</uri>
        <uri>WEB-INF/resource-bundles.xml</uri>
    </module-overrides>
    <root-element>web-app</root-element>
    <uri>WEB-INF/web.xml</uri>
    <module-descriptor>external-override</module-descriptor>
    <config-root></config-root>
</deployment-plan>
```

Creating a Deployment Plan

- Tools for creating a deployment plan:
 - weblogic.PlanGenerator
 - Administration Console
- Goals for creating a deployment plan:
 - To expose the external resource requirements of the application as variables in the deployment plan
 - To expose additional configurable properties, such as tuning parameters as variables in the deployment plan

Creating a New Deployment Plan

WLS includes tools to accelerate deployment plan creation.

The Administration Console:

- Generates a skeleton plan.xml if a plan folder is detected with a newly deployed application
- Updates plan.xml when you use the console to modify the deployment descriptor settings

The weblogic.PlanGenerator Java class can also generate a skeleton plan.xml for an existing application.

weblogic.PlanGenerator

- Is a Java-based deployment configuration tool
- Is primarily intended for developers who want to export portions of an Oracle WebLogic Server deployment configuration into an XML deployment plan
- Enables you to generate a basic Oracle WebLogic Server configuration for applications that have only J2EE deployment descriptors

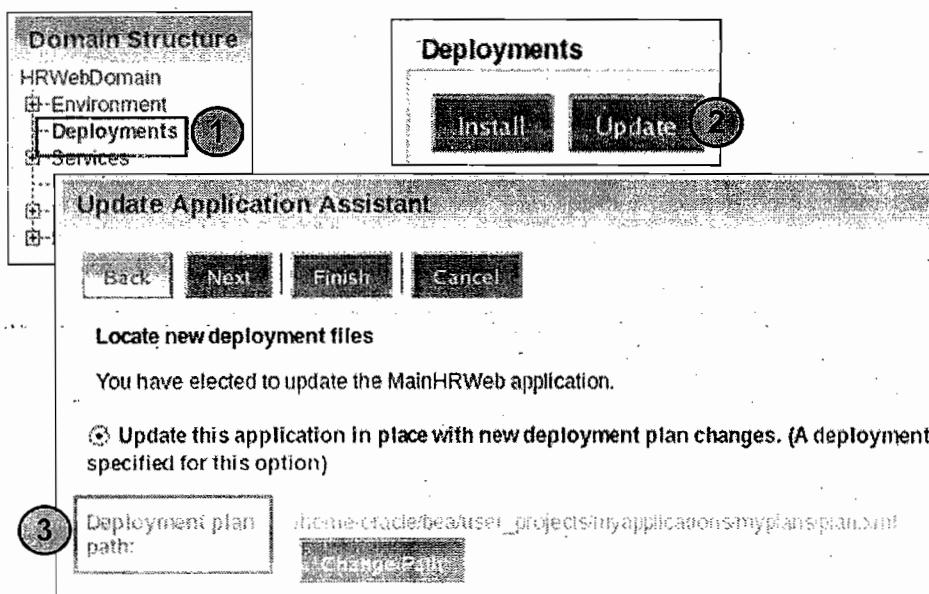
Using the Administration Console to Generate a Deployment Plan

- The Administration Console automatically generates or updates the deployment plan.
- You can generate a deployment plan using the Administration Console using the following steps:
 - Prepare the deployment files.
 - Install the application archive.
 - Save the configuration changes to a deployment plan.

Using an Existing Deployment Plan to Configure an Application

- Prepare the application.
- Place the existing deployment plan in the plan subdirectory of the application root.
- Install the application.
 - The Administration Console validates the deployment plan configuration against the target servers and clusters that are selected during the installation.
- Use the Administration Console or the weblogic.Deployer utility to identify both the application and the plan to use for deployment.

Using an Existing Deployment Plan

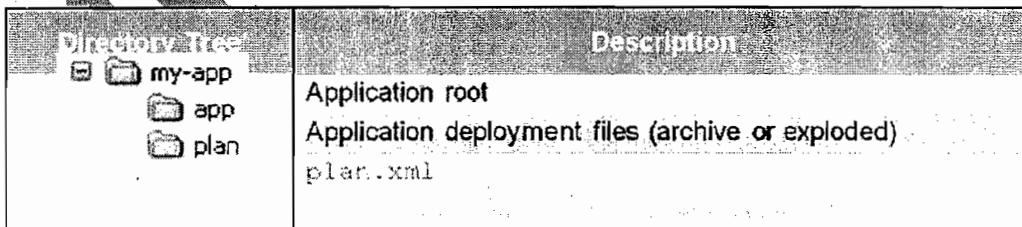


Generic File Loading Overrides

- Place application-specific files to be overridden into a new optional subdirectory (named AppFileOverrides) in the existing plan directory structure.
- The presence or absence of this new optional subdirectory controls whether file overrides are enabled for the deployment.
- If this subdirectory is present, an internal ClassFinder is added to the front of the application and module ClassLoaders for the deployment.
- The file override hierarchy rules follow the existing ClassLoader and resource loading rules and behaviors for applications.

Directory Structure for Easier Production Deployment

- The application directory structure separates the generated configuration files from the core application files.
- This allows the configuration files to be easily changed or replaced without disturbing the application itself.
- Applications can be deployed simply by specifying the installation root.



This allows the deployment configuration files to be located in a well-known location.

Sanity Checking in Production Without Disruption to Clients

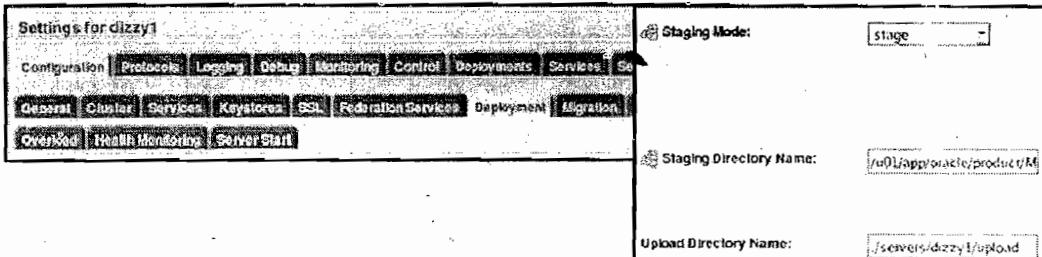
- Using the Administration mode, administrators can deploy an application into a production environment without exposing the application to external clients.
- Access to the application is restricted to a configured Administration channel.

- You can perform a final (“sanity”) check on the application directly in the production environment without disruption to the clients.

Staged Deployment

You can configure deployment per server or for each application as:

- **staged (default):** Files are copied to the preconfigured staging directory for preparation and activation.
- **nostage:** Files are deployed from a static location.
- **external_stage:** Files are copied by a user or a third-party tool before deployment.



Production Redeployment

- You can redeploy a revised version of a production application alongside the older version:
 - Without affecting the existing clients to the application
 - Without interrupting the availability of the application to the new client request
- Oracle WebLogic Server automatically manages client connections so that:
 - Existing clients continue to use the older application
 - New client requests are directed to the newer application
- The older version is undeployed after all current clients complete their work.

Application Availability

- By default, when an application is redeployed:
 - It is unavailable to clients for a brief time
 - Existing clients lose any conversational state
- Some types of applications require availability 24 hours a day, seven days a week.
- Third-party proxy solutions are possible, but they require multiple servers.

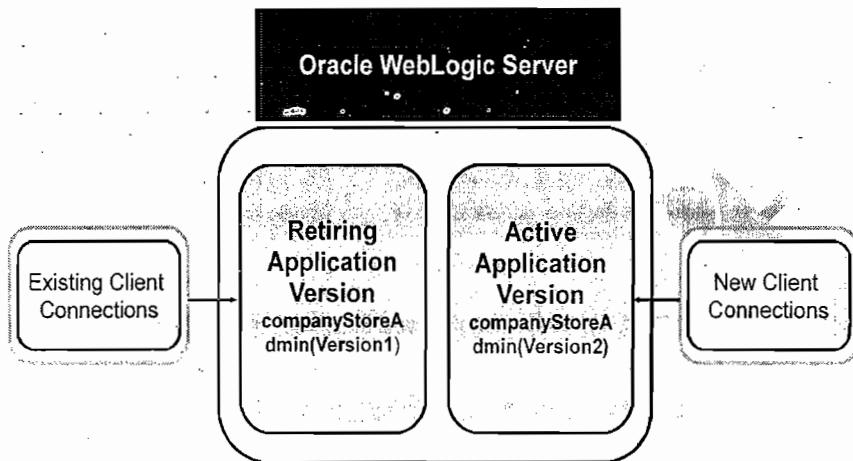


WebLogic Production Redeployment

- Production redeployment:
 - Allows two versions of a single Web application or module to run simultaneously
 - Requires you to include unique version information either:
 - Within the application’s META-INF/MANIFEST.MF file
 - As part of the deployment process
- When a new version is redeployed, WLS automatically:

- Routes existing clients to the prior (retired) version
- Routes new clients to the new version
- Undeploys the prior version when all existing clients finish their work or their conversations time out

Production Redeployment



- To support the production redeployment strategy, Oracle WebLogic Server now recognizes a unique version string entry in the Enterprise MANIFEST file.
- When a redeployment operation is requested, Oracle WebLogic Server checks the version string to determine whether to deploy a new version of the application.
- Production redeployment is performed automatically if:
 - An application supports production redeployment
 - Its deployment configuration is updated with changes to resource bindings
- This occurs even if no version string is specified in the application's manifest file.

Advantages of Production Redeployment

Saves the trouble of:

- Scheduling application downtime
- Setting up redundant servers to host new application versions
- Managing client access to multiple application versions manually
- Retiring older versions of an application manually

Requirements and Restrictions for Production Redeployment

- Production redeployment is supported for:
 - Stand-alone Web Application (WAR) modules and Enterprise Applications (EARs) whose client accesses the application via a Web application (HTTP)
 - Enterprise Applications that are accessed by inbound JMS messages from a global JMS destination, or from inbound JCA requests
 - All types of Web Services, including conversational and reliable Web Services
- Production redeployment is not supported for:
 - Stand-alone EJB or Resource Archive (RAR) modules

- Applications that use JTS drivers
- Applications that obtain JDBC data sources via the DriverManager API instead of using the JNDI lookup
- Applications that include the EJB 1.1 container-managed persistence (CMP) EJBs
- A deployed application must specify a version number.
- WLS can host a maximum of two different versions of an application at one time.
- When you are redeploying a new version of an application, the following features cannot change:
 - Deployment targets
 - Security model
 - Persistent store settings

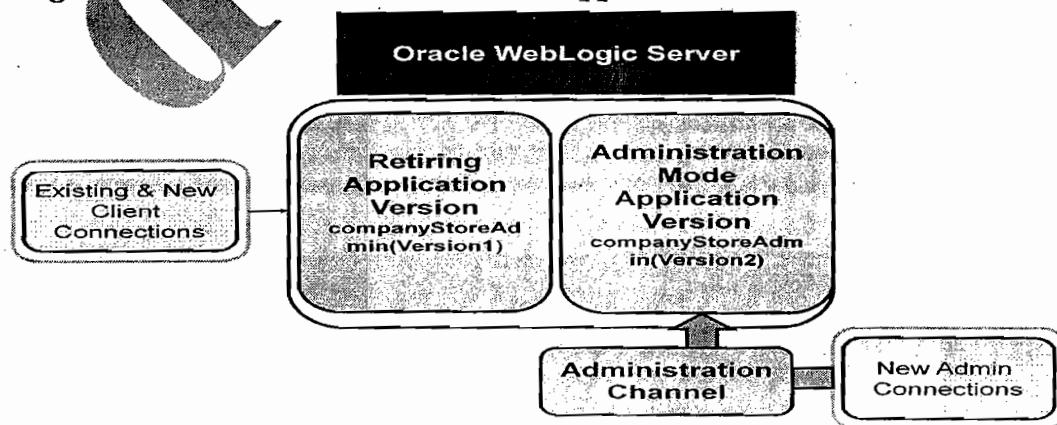
Redeploying a New Application Version

- Verify that only one version of the application is currently deployed.
- Verify the MANIFEST.MF files to ensure that both applications have different versions.
- Copy the new version into a suitable directory.
- Redeploy the new application version and specify the updated deployment files.
- Verify that both versions are deployed and that new requests are being sent to the new version.

Redeploying Versus Distributing

- Distributing is an alternative to deploying an application.
 - Distributing an application prepares it for deployment by copying its files to all target servers and validating the files.
 - You can start a distributed application in Administration mode. Access to the application is then restricted to a configured Administration channel.
- Distributing a new version of the application makes it available for testing before being released for general consumption.
- Redeploying a new version of an application places the application immediately into use and makes it available to new client requests.

Distributing a New Version of the Production Application



Distributing a New Application Version

- Use the weblogic.Deployer –distribute command.
- After the application is distributed, start the application in Administration mode.
- Test the application.
- When ready, start the application (without using –adminmode).
- Optionally set a retirement timeout for the older version of the application.

Production Redeployment

- Create MANIFEST.mf with the following contents:

Name: java/util/
Specification-Title: Java Utility Classes
Specification-Version: 1.2
Specification-Vendor: Sun Microsystems, Inc.
Implementation-Title: java.util
Implementation-Version: build57
Implementation-Vendor: Sun Microsystems, Inc.

- Then add an entry for WLS versioning:

Weblogic-Application-Version: Version1.0Beta

- Create an EAR file named someApp.ear by entering the following command:
jar cvfm someApp.ear Manifest.mf *

- This creates the EAR file with a manifest with the following contents:

Manifest-Version: 1.0
Created-By: 1.6.0 (Sun Microsystems Inc.)
Weblogic-Application-Version: Version1.0Beta
Name: java/util/
Specification-Title: Java Utility Classes
Specification-Version: 1.2
Specification-Vendor: Sun Microsystems, Inc.
Implementation-Title: java.util
Implementation-Version: build57
Implementation-Vendor: Sun Microsystems, Inc.

Quiz

Which of the following is NOT a true statement about the deployment plans in Oracle WebLogic Server?

1. Overrides values in application descriptors
2. Can be created by Oracle WebLogic Server during deployment
3. Is packaged within an application archive
4. Is an XML file
5. Can be created with weblogic.PlanGenerator

Answer: 3

When an application is in the _____ state, it is distributed to the servers but is not yet available to clients:

1. Activated

2. Staged
3. Targeted
4. Prepared
5. Loaded

Answer: 4

Name four techniques or tools that can be used to deploy new applications to Oracle WebLogic Server?

1. Administration Console
2. WLST
3. weblogic.PlanGenerator
4. weblogic.Deployer
5. JMS
6. Auto-deployment

Answer: 1, 2, 4, 6

Summary

In this lesson, you should have learned about:

- Deployment plans
- Deployment staging
- Production redeployment
- The configuration of an application for multiple development environments

Introduction to Clustering

Objectives

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

- List the uses of a cluster
- ✓ Describe the basic cluster architecture
- ✓ Describe the multitier cluster architecture
- ✓ Describe how clusters communicate

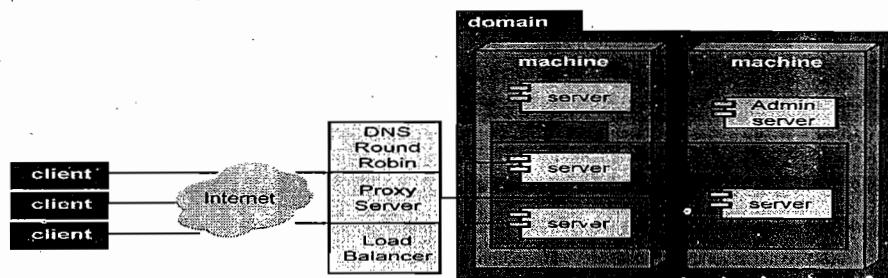
Road Map

- Cluster Architecture
 - Cluster Definition
 - Basic Cluster Architecture
 - Multitier Cluster Architecture
 - Proxy Servers
- Networks and Clusters
- Cluster Communication

Definition: Clustering

- A cluster is a group of Oracle WebLogic Server instances that work in coordination.
- Clustering provides:
 - ← High availability
 - ← Load balancing

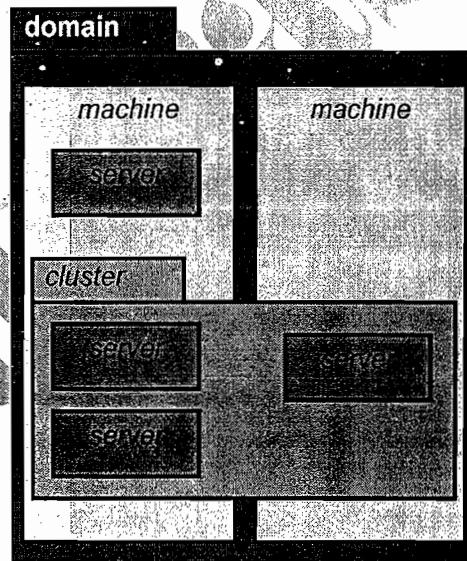
Scalability



What Is a Cluster?

A cluster:

- Is a logical group of Oracle WebLogic Servers within a domain
- Supports features to provide high availability for:
 - Whole servers
 - Web applications/services
 - EJB applications
 - JMS
- Is transparent to a client



Benefits of Clustering

- There are two main benefits of clustering together Oracle WebLogic Servers:
 - Scalability
 - High availability
- Scalability is the ability to provide more capacity for an application, in this case, by adding additional servers without having to make major architectural changes.

- High availability ensures that when a server (in a cluster) fails, there are other servers to take over the work, so that the client is not affected.

Concept	Description
Scalability	Provides more capacity for an application by adding servers, without having to make major architectural changes
Load balancing	Distributes work (client requests and so on) across the members of a cluster
Failover	After a system failure on one server, automatically continues ongoing work on another server
Migration	After a system failure on one server, continues ongoing work by moving the component to another server

Key Capabilities

The key capabilities of a WebLogic cluster are:

- Application failover
 - When an object in an application that is performing a task becomes unavailable, another object takes over and finishes the job.
- Site failover
 - When all the services and applications in a single site fail, they can switch to a separate site and continue processing.
- Server migration
 - When a server fails, pinned services can be migrated to another server in a cluster.
- Load balancing
 - Tasks and communications are evenly distributed across multiple servers.

Cluster Architecture

- Applications are generally broken into multiple tiers, each representing its 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 but load-balancing and failover is a little different.

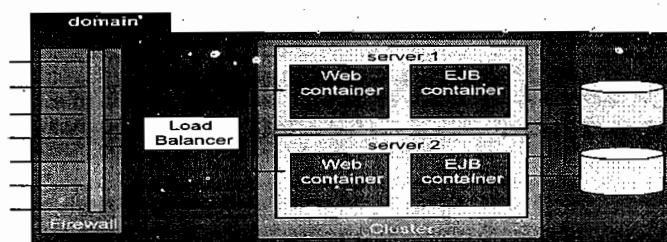
Deciding on a Cluster Architecture

- Good architecture is somewhat subjective but there are a few global considerations:
 - 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

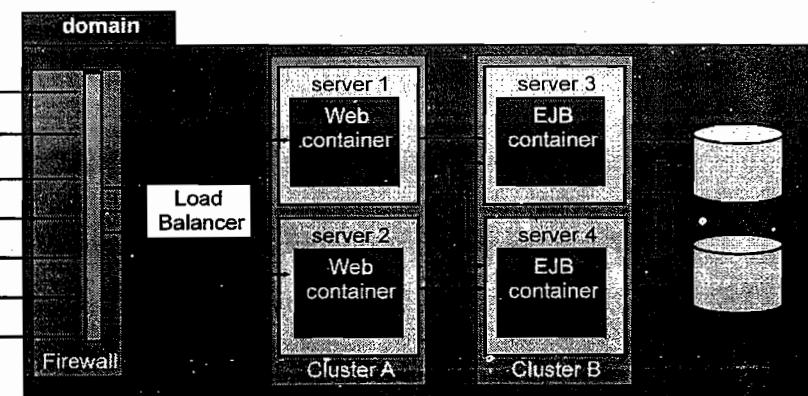
Basic Cluster Architecture

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



Multitier Cluster Architecture

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



Basic Cluster Architecture: Advantages and Disadvantages

- The basic cluster architecture has these advantages:
 - Easy administration
 - Flexible load balancing
 - Robust security
- The basic cluster architecture has these disadvantages:
 - It cannot load balance EJB method calls.
 - Load balancing across the tiers may become unbalanced.

Multitier Advantages and Disadvantages

- The multitier architecture has these advantages:
 - Improved load balancing
 - Load balancing of EJB methods
 - Higher availability
 - Improved security options

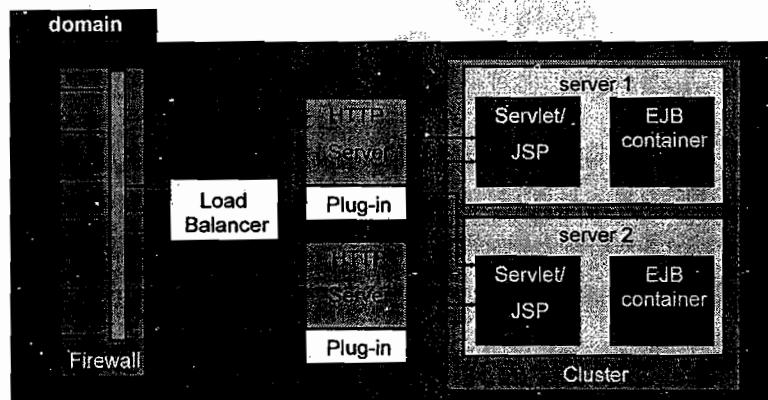
- The mult-tier architecture has these disadvantages:
 - Can create a bottleneck when the presentation tier makes frequent calls to the business logic
 - Can lead to increased licensing cost
 - Can lead to added firewall configuration complexity

Proxy Servers

- Proxy servers are used to provide load balancing and failover for a cluster. They also:
 - 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 an internal WebLogic servlet or a third-party application.
- A hardware-based proxy server is typically a physical load balancer.

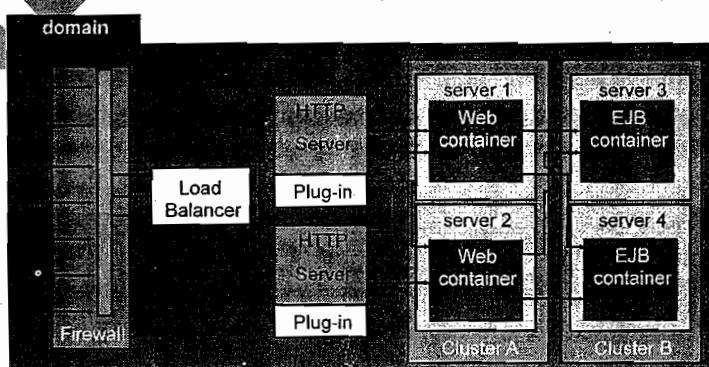
Basic Cluster Proxy Architecture

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



Multitier Cluster Proxy Architecture

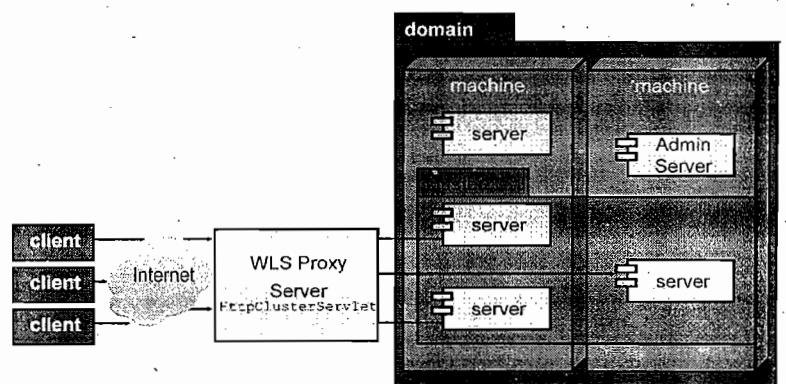
This is similar to the multitier cluster architecture, except that static content is hosted on nonclustered HTTP servers.



WLS HttpClusterServlet

HttpClusterServlet:

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



WLS Plug-Ins

WLS is compatible with major Web servers using the following plug-ins:

- Sun Java System Web Server plug-in (formerly Netscape iPlanet or Sun One Web Server)
- IIS plug-in (Microsoft IIS)
- Apache plug-in

Plug-ins:

- Delegate dynamic content requests to WLS
- Round-robin across a cluster
- Support routing based on the URL path, on the MIME type of the requested file, or both
- Route HTTP requests to back-end WLS instances based on session cookie or URL rewriting
- Avoid failed servers in the cluster

Proxy 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

Architecture Recommendations

- If possible, place static Web content on separate Web servers in the DMZ.
- Use a combined-tier architecture if your presentation and control tier makes multiple invocations of the business tier.
- Make sure that your architecture choice supports passing active and passive cookies between the cluster and the client application.

Cluster in Networks

- Oracle WebLogic Server clusters can be created in three different kinds of networks:
 - Local Area Networks (LAN)
 - Metropolitan Area Networks (MAN)
 - Wide Area Networks (WAN)
- When you configure your cluster, keep in mind the type of network that you would use.

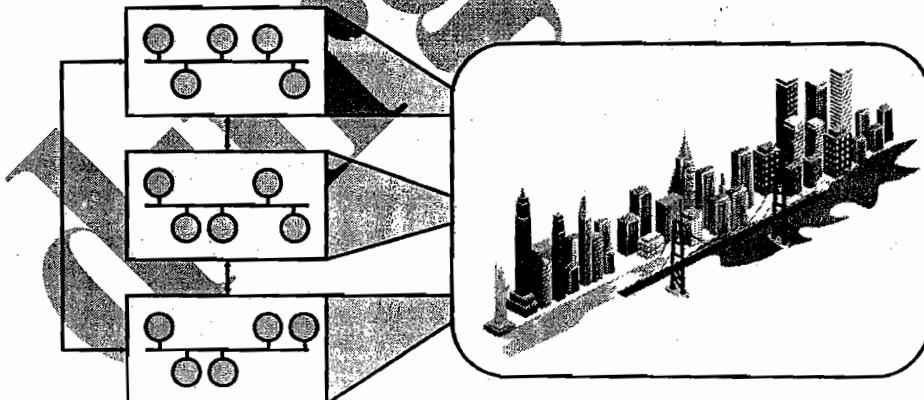
Local Area Networks

- A local area network serves a local set of computers.
 - It usually uses high quality, high-speed communication links.
 - Typical data transmission speeds are 100 megabits/second.
- Most clusters exist within a single LAN.



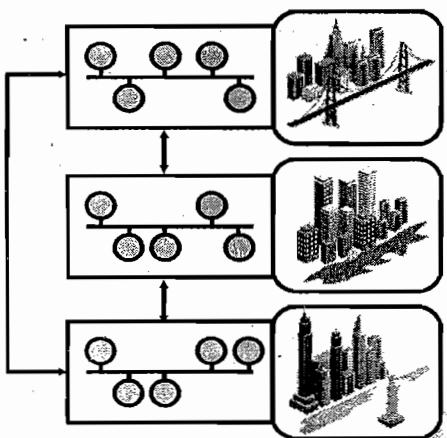
Metropolitan Area Networks

- A MAN is a network that usually spans a campus or a city.
- You can have different clusters located reasonably close to each other within a MAN.



Wide Area Networks

- A WAN usually spans a wider geographical area and can be made up of smaller MANs and LANs.
- You can have different clusters located in different regions within a WAN.
 - A cluster can be located in different LANs within a MAN or within a WAN.



Server Communication in a Cluster

- The Oracle WebLogic Server instances in a cluster communicate with one another using two different techniques:
 - Unicast/multicast (UDP)
 - Sockets (peer-to-peer TCP)
- The server instances use IP unicast or multicast to broadcast the availability of services and heartbeats that indicate continued availability.
- IP multicast broadcasts one-to-many communications among clustered instances.
- IP 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.
- IP sockets are used for peer-to-peer communications between server instances.

Cluster Communication

- Members of a cluster cooperate to achieve high availability using the following:
 - Broadcast messages such as “heartbeats”
 - Peer-to-peer IP sockets
- You can configure broadcast communication to use either:
 - IP unicast
 - A dedicated IP multicast address (224.0.0.0 through 239.255.255.255) and port
- If heartbeats are not received from a cluster member, the server is marked as “failed” and its services are not used.

Detecting a Failure

- WebLogic clusters detect the failure of a server instance in the following ways:
 - Through the use of IP sockets
 - Through the Oracle WebLogic Server heartbeat
- If a server in the cluster unexpectedly closes its socket, it is marked as “failed” and its services are not used.
- Server instances use multicast to broadcast heartbeats every 10 seconds to the other server instances in the cluster.

- If three heartbeats are missed from a peer server, the server is marked as "failed" and its services are not used.

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.

Peer-to-Peer Communications

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

- Accessing the nonclustered 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)

Multitier Communications

- Multitier clusters require more IP sockets than a combined-tier cluster:
 - One socket for replicating session states
 - One socket for each Oracle WebLogic Server in the EJB cluster, for accessing remote objects
- As an example, using a three-node cluster, the worst-case scenario would be five open sockets per server:
 - One primary and secondary replicated session
 - Each server simultaneously invokes a remote EJB method on each node in the cluster

Communication in a WAN

- In a WAN, the servers in your cluster may span multiple subnets.
- In order for multicast messages to reliably transmit across a WAN, your network must meet the following requirements:
 - Full support of IP multicast packet propagation
 - A network latency that allows multicast messages to reach their destination in 200 to 300 milliseconds
 - A multicast time-to-live value high enough to ensure that routers do not discard multicast packets

Summary

In this lesson, you should have learned about:

- The differences between a basic cluster architecture and a multitier cluster architecture
- The use of a proxy server to load balance requests to a cluster
- The way servers in a cluster communicate

Configuring a Cluster

Objectives

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

- Prepare your environment for a cluster
- Create and configure a cluster
- Create and configure a proxy server

Road Map

- Preparing for a Cluster
 - Cluster Architecture
 - Network and Security Topology
 - Machines
 - Names and Addresses
- Configuring a Cluster
- Configuring a Proxy Server

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.

Cluster Architecture

- Will you use a single-tier or a multitier architecture?
- How will you perform your load balancing?
 - Will you use basic Oracle WebLogic Server load balancing?
 - Will you use a third-party load balancer?
- Will you use demilitarized zones with firewalls?

Network and Security Topology

- Will your cluster exist in a single LAN?
- Will your cluster span a MAN or a WAN?
- 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.

Security Options for Cluster Architectures

- 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
- When you use a load balancer, in addition to the security features provided with the load balancer, you may want to place a firewall between it and the untrusted clients.
- When you use a single database that supports both internal and external data:
 - Place an additional firewall in front of the database server

Hardware

- You can set up a cluster on a single, nonmultihomed computer for demonstration or development.
 - This is not practical for production environments.
- The machine cannot have a dynamically assigned IP address.
- There is no built-in limit for the number of server instances in a cluster.
 - Large, multi-processor servers can host large clusters.
 - The recommendation is one Oracle WebLogic Server instance for every two CPUs.

Clustering on One Piece of Hardware

- To set up a cluster on one machine, the server is not required to be multihomed; the servers in a cluster can use one IP address.
- All servers in a cluster can use a dedicated multicast port number for interserver communication:
 - Required, if servers use a single IP address
 - Useful for segmenting multicast traffic among specific NICs

Names and Addresses

- Location information is needed for:
 - The Administration Server
 - Managed Servers
 - Multicast location
- For a production environment, use DNS names as opposed to IP addresses.
 - Firewalls can cause IP address translation errors.
- Each Oracle WebLogic Server resource should have a unique name.
- The multicast address should not be used for anything other than cluster communications.

Cluster Address

- The cluster address is used in entity and session beans to construct the host name portion of the request URLs.
- You can explicitly define the address of a cluster.

Do You Want to perfect every topic in JSP

Attend

JSP

(SCWCD Classes)

Mr. DURGA M.Tech

- ✓ **JSP Life Cycle**
- ✓ **JSP Elements**
- ✓ **JSP Directives**
- ✓ **Implicit Objects**
- ✓ **JSP Document**
- ✓ **JSP Standard Actions**
- ✓ **Expression Language (EL)**
- ✓ **JSTL**
- ✓ **Classic Custom Tags**
- ✓ **Simple Custom Tags**
- ✓ **Tag Files**

Other than these... Nothing else in JSP

SCWCD JSP Syllabus

Section 6: The JavaServer Pages (JSP) Technology Model

Identify, describe, or write the JSP code for the following elements: (a) template text, (b) scripting elements (comments, directives, declarations, scriptlets, and expressions), (c) standard and custom actions, and (d) expression language elements.

Write JSP code that uses the directives: (a) 'page' (with attributes 'import', 'session', 'contentType', and 'isELIgnored'), (b) 'include', and (c) 'taglib'.

Write a JSP Document (XML-based document) that uses the correct syntax.

Describe the purpose and event sequence of the JSP page life cycle: (1) JSP page translation, (2) JSP page compilation, (3) load class, (4) create instance, (5) call the `jspInit` method, (6) call the `_jspService` method, and (7) call the `jspDestroy` method.

Given a design goal, write JSP code using the appropriate implicit objects: (a) request, (b) response, (c) out, (d) session, (e) config, (f) application, (g) page, (h) pageContext, and (i) exception.

Configure the deployment descriptor to declare one or more tag libraries, deactivate the evaluation language, and deactivate the scripting language. 6.7 Given a specific design goal for including a JSP segment in another page, write the JSP code that uses the most appropriate inclusion mechanism (the include directive or the `jsp:include` standard action).

Section 7: Building JSP Pages Using the Expression Language (EL)

Given a scenario, write EL code that accesses the following implicit variables including `pageScope`, `requestScope`, `sessionScope`, and `applicationScope`, `param` and `paramValues`, `header` and `headerValues`, `cookie`, `initParam` and `pageContext`.

Given a scenario, write EL code that uses the following operators: property access (the `.` operator), collection access (the `[]` operator).

Section 8: Building JSP Pages Using Standard Actions

Given a design goal, create a code snippet using the following standard actions: `jsp:useBean` (with attributes: 'id', 'scope', 'type', and 'class'), `jsp:getProperty`, `jsp:setProperty` (with all attribute combinations), and `jsp:attribute`.

Given a design goal, create a code snippet using the following standard actions: `jsp:include`, `jsp:forward`, and `jsp:param`.

Section 9: Building JSP Pages Using Tag Libraries

For a custom tag library or a library of Tag Files, create the 'taglib' directive for a JSP page.

Given a design goal, create the custom tag structure in a JSP page to support that goal.

Given a design goal, use an appropriate JSP Standard Tag Library (JSTL v1.1) tag from the "core" tag library.

Section 10: Building a Custom Tag Library

Describe the semantics of the "Classic" custom tag event model when each event method (`doStartTag`, `doAfterBody`, and `doEndTag`) is executed, and explain what the return value for each event method means; and write a tag handler class.

Using the `PageContext` API, write tag handler code to access the JSP implicit variables and access web application attributes.

Given a scenario, write tag handler code to access the parent tag and an arbitrary tag ancestor.

Describe the semantics of the "Simple" custom tag event model when the event method (`doTag`) is executed; write a tag handler class; and explain the constraints on the JSP content within the tag.

Describe the semantics of the Tag File model; describe the web application structure for tag files; write a tag file; and explain the constraints on the JSP content in the body of the tag.

Section 11: Java EE Patterns

Given a scenario description with a list of issues, select a pattern that would solve the issues. The list of patterns you must know are: Intercepting Filter, Model-View-Controller, Front Controller, Service Locator, Business Delegate, and Transfer Object.

Match design patterns with statements describing potential benefits that accrue from the use of the pattern, for any of the following patterns: Intercepting Filter, Model-View-Controller, Front Controller, Service Locator, Business Delegate, and Transfer Object.

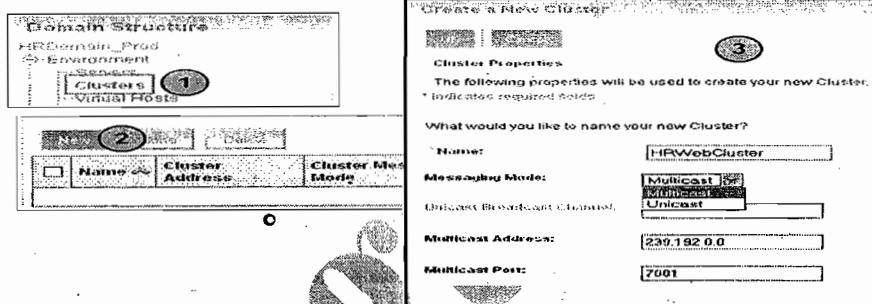
- The cluster address should be a DNS name that maps to the IP addresses or DNS names of each Oracle WebLogic Server instance in the cluster.
- You can also have Oracle WebLogic Server dynamically generate an address for each new request.
 - Minimizes configuration
 - Ensures an accurate cluster address
- The dynamic cluster address is created in the form of:
 - listenaddress1:listenport1,listenaddress2:listenport2,listenaddress3:listenport3

Configuration Options

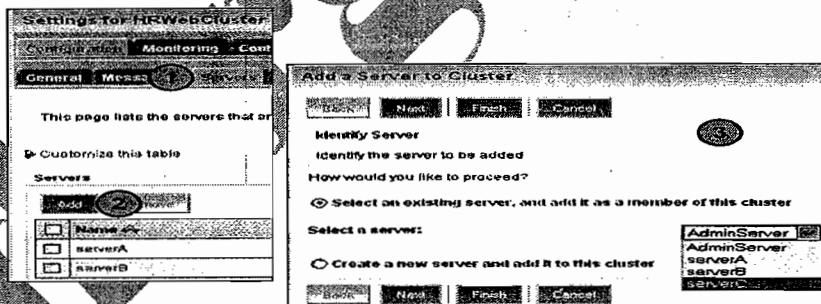
There are multiple ways to create and configure an Oracle WebLogic Server cluster:

- Administration Console
- Configuration Wizard
- WebLogic Scripting Tool (WLST)

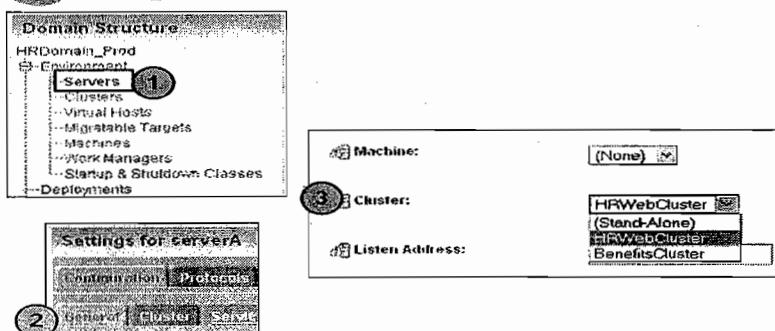
Creating a Cluster



Adding Cluster Members: Option 1



Adding Cluster Members: Option 2



Creating a Cluster by Using the Administration Console

The screenshot shows the Oracle Administration Console interface for creating a new cluster. It includes:

- Domain Structure:** Shows the hierarchy of domains, clusters, and services.
- Create a New Cluster Wizard:**
 - Step 1: General:** Shows the 'Cluster Properties' section where the cluster name is set to 'Cluster-0'.
 - Step 2: Configuration:** Shows the 'Messaging Model' set to 'Unicast' and the 'Unicast Broadcast Channel' set to 'NS:192.0.0'.
 - Step 3: Cluster Address:** Shows the 'Address' field set to '192.168.0.10' and the 'Number Of Servers In Cluster' set to '1'.
- Cluster Overview - More Columns (Excluded):** A table showing cluster details like Name, Address, Operation Status, and Type.
- Monitoring for Cluster-0:** A table showing monitoring details for the cluster.

Configuring Cluster Communication

The screenshot shows the 'Settings for HRWebCluster' page under the 'Messaging' tab. It includes:

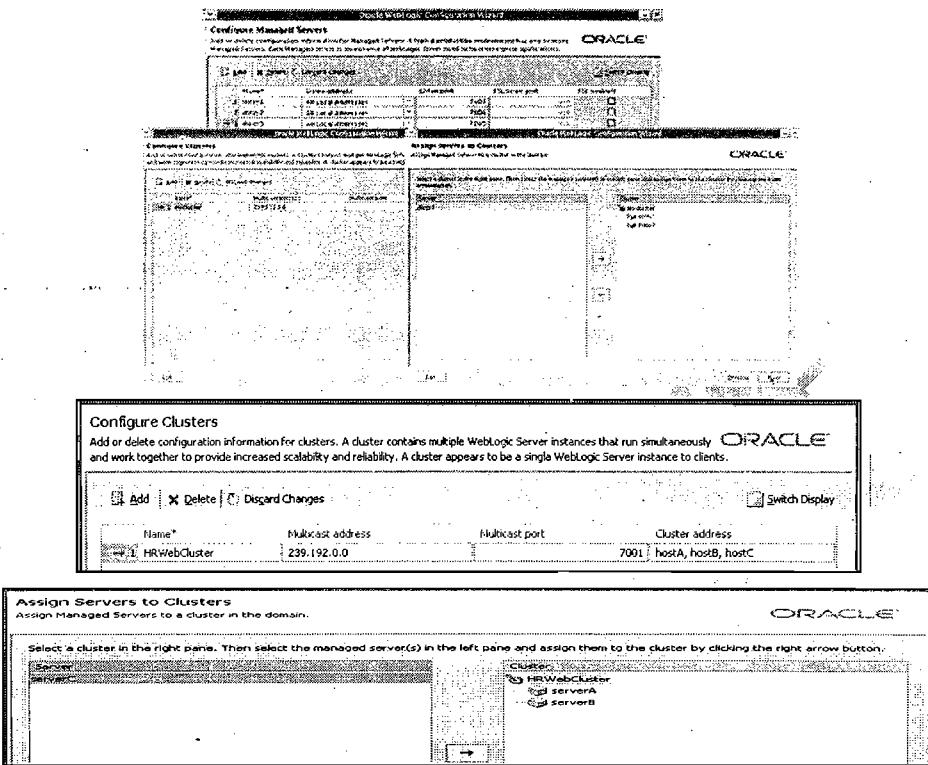
- Multicast Send Delay:** Set to 9 seconds.
- Multicast TTL:** Set to 1.
- Multicast Buffer Size:** Set to 54.
- Idle Periods Until Timeout:** Set to 3.
- Message Model:** Set to 'Unicast'.
- Unicast Broadcast Channel:** Set to 'NS:192.0.0'.
- Unicast Address:** Set to '192.168.0.10'.
- Number Of Servers In Cluster:** Set to '1'.

Adding Servers

The screenshot shows the Oracle Administration Console interface for adding servers to a cluster. It includes:

- Get Started With Clusters:** A summary page for adding servers to a cluster.
- Add Server To Cluster:**
 - Specify Server:** Fields for 'Name' (e.g., 'hrweb'), 'Port' (e.g., '8000'), and 'Protocol' (e.g., 'HTTP').
 - Select Existing Service:** A dropdown menu for selecting an existing service.
 - Configure New Port:** A checkbox for creating a new port.
- Settings for Cluster-0:** A detailed configuration page for the cluster, including tabs for General, Monitoring, Control, Deployments, Services, and Messaging.

Creating a Cluster with the Configuration Wizard



Using the Cluster MBean

- Cluster MBean can also be used to create a cluster.
- 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`

Clusters and WLST

A box containing WLST script code with annotations:

```
connect('myuser', 'mypassword', 't3://localhost:7001')
edit()
startEdit()
cd('/')
cmo.createCluster('HRWebCluster')
cd('/Clusters/HRWebCluster')
set('MulticastAddress', '239.192.0.0')
set('MulticastPort', 7050)

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()
```

Annotations in the code:

- An arrow points to the line `cmo.createCluster('HRWebCluster')` with the text "Create new cluster."
- An arrow points to the line `cmo.setCluster(cluster)` with the text "Assign cluster members."

Starting Servers in a Cluster

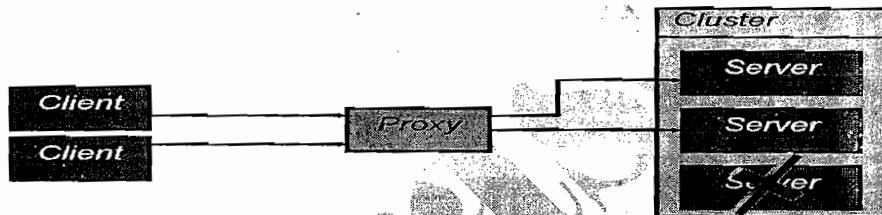
The screenshot shows a command prompt window with the following log output:

```
<Aug 5, 2005 11:30:34 AM EDT> <Notice> <Log Management> <BEA-120112> <The server initialized the domain log broadcaster successfully. Log messages will now be broadcasted to the domain log.>
<Aug 5, 2005 11:30:34 AM EDT> <Notice> <WebLogicServer> <BEA-000165> <Server state changed to STANDBY>
<Aug 5, 2005 11:30:37 AM EDT> <Notice> <WebLogicServer> <BEA-000165> <Server state changed to STARTING>
<Aug 5, 2005 11:30:42 AM EDT> <Notice> <Cluster> <BEA-000138> <Listening for announcements from cluster storecluster on 239.192.0.0:7777.>
<Aug 5, 2005 11:30:42 AM EDT> <Notice> <Cluster> <BEA-000133> <Waiting to synchronize with the existing members of storecluster.>
<Aug 5, 2005 11:31:12 AM EDT> <Notice> <WebLogicServer> <BEA-000065> <Server state changed to ADMIN>
<Aug 5, 2005 11:31:12 AM EDT> <Notice> <WebLogicServer> <BEA-000365> <Server state becomes an active member of the cluster.>
<Aug 5, 2005 11:31:12 AM EDT> <Notice> <Log Management> <BEA-120112> <The server initialized the domain log broadcaster successfully. Log messages will now be broadcasted to the domain log.>
<Aug 5, 2005 11:31:18 AM EDT> <Notice> <Cluster> <BEA-000102> <Joining cluster storecluster on 239.192.0.0:7777>
<Aug 5, 2005 11:31:18 AM EDT> <Notice> <WebLogicServer> <BEA-000360> <Server state changes to RUNNING>
<Aug 5, 2005 11:31:18 AM EDT> <Notice> <WebLogicServer> <BEA-000360> <Server state changes to ACTIVE>
```

High Availability for HTTP Clients

HTTP clients require an intermediary or *proxy* to load balance requests and avoid failed servers:

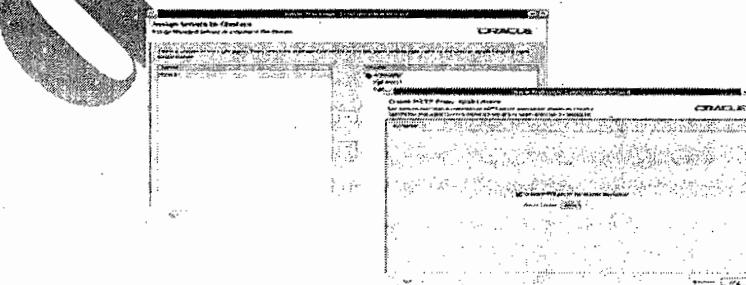
- Network hardware
- Web server
- Oracle WebLogic Server (HttpClusterServlet)



WebLogic Proxy Servers

- WebLogic HTTPClusterServlet runs within a Web application that is deployed on an Oracle WebLogic Server.
 - The servlet proxies requests to other servers in a cluster.
 - The servlet should run on a separate, nonclustered Managed Server.
- You can create a WebLogic proxy server initially by using the Configuration Wizard.
 - You can also manually set up the Web application with HTTPClusterServlet and deploy it on the Managed Server.

Creating the WebLogic Proxy Server with the Configuration Wizard



Creating the WebLogic Proxy Server Manually

- HttpClusterServlet is specified in the web.xml file of the default Web application on the proxy server.
- This file must reside in the \WEB-INF directory of the Web application directory.

- The proxy servlet needs to be defined as the default Web application for the Managed Server.
 - This is defined in the weblogic.xml deployment descriptor that is located in the \WEB-INF directory of the Web application.
 - Set the <context-root> element to a forward slash character (/) in the <weblogic-web-app> stanza

Configuring HttpClusterServlet

HttpClusterServlet is specified in the web.xml file of the default Web application on the proxy server.

HttpClusterServlet declaration:

```
<servlet>
  <servlet-name>HttpClusterServlet</servlet-name>
  <servlet-class>
    weblogic.servlet.proxy.HttpClusterServlet
  </servlet-class>

</servlet>
```

<web.xml>

Specifying Initial Parameters

```
Declaring initial parameters:
<servlet>
  <servlet-name>HttpClusterServlet</servlet-name>
  <servlet-class>
    weblogic.servlet.proxy.HttpClusterServlet
  </servlet-class>

  <init-param>
    <param-name>WebLogicCluster</param-name>
    <param-value>(serverA:7001:7002)serverB:7001:7002)serverC:7001:7002
    </param-value>
  </init-param>
  <init-param>
    <param-name>DebugEnabledConfigInfo</param-name>
    <param-value>ON</param-value>
  </init-param>
</servlet>
```

<web.xml>

Servlet Mapping

```
Configuring servlet mapping:
<servlet>
  <servlet-name>HttpClusterServlet</servlet-name>
  <servlet-class>
    weblogic.servlet.proxy.HttpClusterServlet
  </servlet-class>
  <servlet-mapping>
    <servlet-name>HttpClusterServlet</servlet-name>
    <url-pattern>/</url-pattern>
  </servlet-mapping>
</servlet>
```

<web.xml>

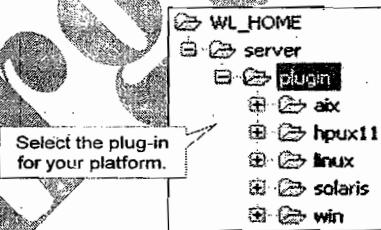
HttpClusterServlet Initialization Parameters

Parameter	Usage	Default Value
WebLogicCluster	(Required) A list of host names and port numbers of the servers to which requests are proxied	(none)
secureProxy	ON/OFF: ON enables SSL between HttpClusterServlet and the server that it proxies to.	OFF
DebugEnabledConfigInfo	ON/OFF: ON allows you to query HttpClusterServlet for debugging information.	OFF

ConnectTimeoutSecs	The maximum time in seconds that the servlet should attempt to connect to the Oracle WebLogic Server host	0 (infinite timeout)
ConnectRetrySecs	The interval in seconds that the servlet sleeps between attempts to connect to a server instance	5
pathTrim	The string to trim at the beginning of the original URL	none
trimExt	The file extension to trim at the end of the URL	none
pathPrepend	The string to prefix to the original URL after pathTrim is trimmed	none

Web Server Plug-Ins

- If you use a supported third-party Web server, you need to install and configure a WLS proxy plug-in.
- Plug-ins support the HTTP clustering features of WLS.
- Plug-ins can be installed with WLS or downloaded separately.
- Supported Web servers include:
 - Apache HTTP Server
 - Netscape Enterprise Server
 - Microsoft Internet Information Server



Installing the Apache Plug-In

- You can configure the Apache plug-in module:
 - As a dynamic shared object
 - As a statically linked module
- The available dynamic shared objects are:
 - mod_wl_20.so
 - mod_wl_22.so
 - mod_wl128_20.so
 - mod_wl128_22.so

Supports 128-bit encryption

Configuring the Apache Plug-In

```
httpd.conf excerpts:  
LoadModule weblogic_module modules/mod_wl_20.so  
...  
<IfModule mod_weblogic.c>  
    WebLogicCluster hostA:7003,hostB:7005,hostC:7007  
    MatchExpression *.jsp  
    other params  
</IfModule>  
<Location /weblogic>  
    SetHandler weblogic-handler  
    WebLogicCluster hostA:7003,hostB:7005,hostC:7007  
    PathTrim /weblogic  
    other params  
</Location>
```

Configuring the Apache Plug-In

- Install the Apache HTTP Server Plug-In as a module in your Apache HTTP Server installation.
- You can configure the module in two different ways:
 - As a dynamic shared object
 - As a statically linked module (1.3.x and higher)
- To configure the Dynamic Shared Object, select the shared object type:
 - mod_wl_20.so (Standard Apache Version 2.x)
 - mod_wl_ssl.so (Apache with SSL/EAPI)
- Edit the httpd.conf file and include the Oracle WebLogic Server modules.
 - LoadModule weblogic_module modules/mod_wl_20.so
- Add an IfModule block to define the WebLogic Cluster instance.

```
<IfModule mod_weblogic.c>  
    WebLogicCluster  
    127.0.0.1:7003,127.0.0.1:7005,127.0.0.1:7007  
</IfModule>
```

Add the proxy path.

```
<Location /weblogic>  
    SetHandler weblogic-handler  
    WebLogicCluster  
    127.0.0.1:7003,127.0.0.1:7005,127.0.0.1:7007  
    DebugConfigFileS ON  
    PathTrim /weblogic  
</Location>
```

General Plug-In Parameters

Parameter	Description
WebLogicHost	The host and port of single Oracle WebLogic Server to proxy to
WebLogicPort	
WebLogicCluster	List of Oracle WebLogic Servers to proxy to
ConnectTimeoutSecs	Maximum time that the plug-in should wait for a response from a server

ConnectRetrySecs	Time that the plug-in should wait before attempting to connect to another server
MaxSkipTime	Time that the plug-in should wait before retrying a failed server
WLCookieName	Session ID cookie name that is used by the application being proxied to
Debug	Enables additional debugging information to be written to a log file
ErrorPage	Web page to display if no servers are available

Apache Plug-In and SSL

- You can use the Apache Plug-In with SSL to guarantee the confidentiality and integrity of data.
- The Apache HTTP Server Plug-In does not use the transport protocol (HTTP or HTTPS) that is specified in the HTTP request.
- Though you can use two-way SSL authentication between the client and Apache, one-way authentication is used between Apache and WebLogic.

Configuring Apache Plug-In SSL

To enable SSL between Apache and WebLogic, perform the following:

- Configure WebLogic for SSL.
- Configure Oracle WebLogic Server's SSL listen port.
- Set WebLogicPort in httpd.conf to the WebLogic SSL listen port.
- Set the SecureProxy parameter in httpd.conf to ON.
- Install a trusted certificate authority file on Apache. You can use DemoTrust.jks that comes with WebLogic.

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

Answer: 4

Which parameters can administrators configure for any Oracle WebLogic Server proxy plug-in?

1. WebLogicCluster
2. MessagingMode
3. ConnectTimeoutSecs
4. Debug
5. Migration

Answer: 1, 3, 4

Which of the following is NOT an available option for proxying HTTP requests to an Oracle WebLogic Server cluster?

1. Web Server with Plug-in
2. Cluster-Aware Stubs
3. Hardware Load Balancer
4. WebLogic HTTPClusterServlet

Answer: 2

Summary

In this lesson, you should have learned how to:

- Prepare your environment for clusters
- Configure a cluster by using different tools
- Configure Oracle WebLogic Server and third-party proxy servers

Managing Clusters

Objectives

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

- Deploy applications to a cluster
- Manage session state in a cluster
- Troubleshoot a cluster

Road Map

- Deploying Applications to Clusters
 - Packaging Applications
 - Two-Phase Deployment and Production Redeployment
- Session Management
- Troubleshooting a Cluster

Packaging Applications

- When you deploy applications to a single Managed Server, you can deploy the applications in an exploded format.
- Oracle recommends deploying packaged applications to a cluster of Managed Servers as .war, .ear, or .jar file

Two-Phase Deployment

- Applications are deployed using two-phase deployment (TPD).
- Applications are copied to the cluster and activated in two phases:
 - 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.

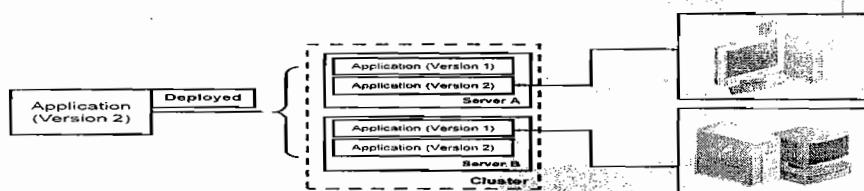
Deploying Applications to a Cluster

- All nodes must be running before an application is deployed to a cluster.

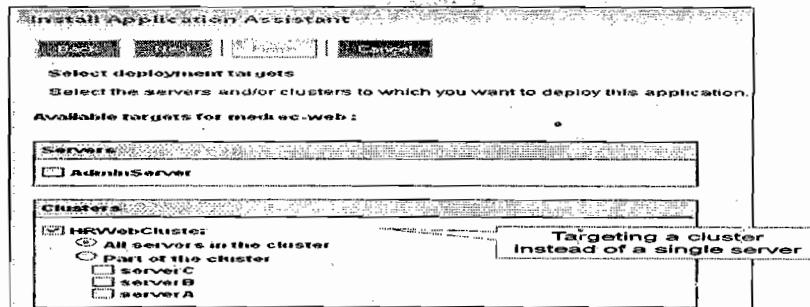
- If phase 2 of the two-phase deployment fails, the application is still deployed to other nodes in the cluster.
- WebLogic allows partial deployment of applications to a partitioned server.
- Session replication for deployed applications may fail if a node is partitioned at the time of deployment.
 - Avoid this by using the enforceClusterConstraints tag with weblogic.Deployer.
 - Or select the Enable Cluster Constraints check box in the console.
- Do not change cluster membership while deploying applications to the cluster.

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.

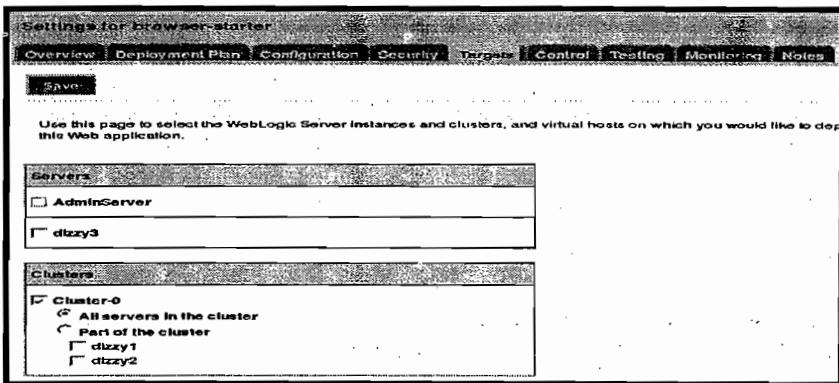


Deploying Applications to a Cluster



Deploying Applications to a Cluster

Name	State	Health	Type	Deployment Order
(1) hrwebcluster	Active	OK	Web Application	100



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.
- Similar options are available for stateful EJB applications.

HTTP Session State Replication

- Oracle WebLogic Server provides clustering support for JSPs and servlets by replicating the HTTP session state.
- To benefit from HTTP session state clustering, you must ensure that the session state is persistent, by configuring:
 - In-memory replication
 - JDBC replication
 - File system replication
- You must also access the cluster via a collection of Web servers with identically configured proxy plug-ins or load-balancing hardware.
- 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.

Machines

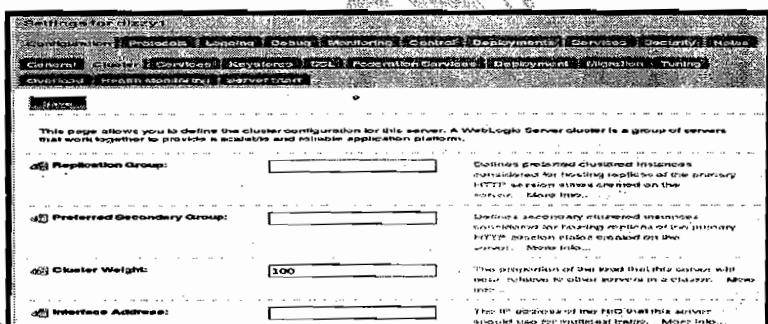
- In Oracle WebLogic Server, machine names are used to indicate that a Managed Server runs on a particular piece of hardware.
- Machine definition is one of the factors that WebLogic takes into account when it chooses another server as its backup for session information.

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
- Replication groups:
 - Represent a subset of servers within a cluster
 - Help to determine the placement of secondary sessions (avoid replicating within the same room, for example)
 - 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

Configuring Replication Groups

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



Secondary Ranking

The distribution of secondary sessions across cluster members can be controlled by using:

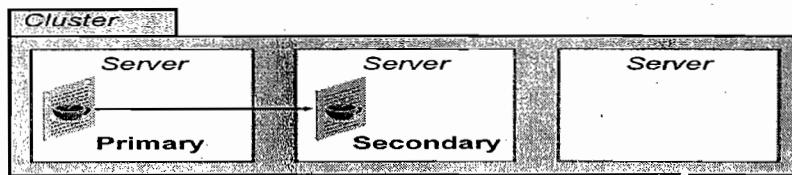
- Machine definitions
- Replication group assignments

Does the server reside on a different machine?	Is the server part of a preferred secondary replication group?	Secondary Server Rank
Yes	Yes	1
Yes	No	2
No	Yes	3
No	No	4

In-Memory Replication

- Each user's session always exists on two servers:
 - Primary
 - Secondary

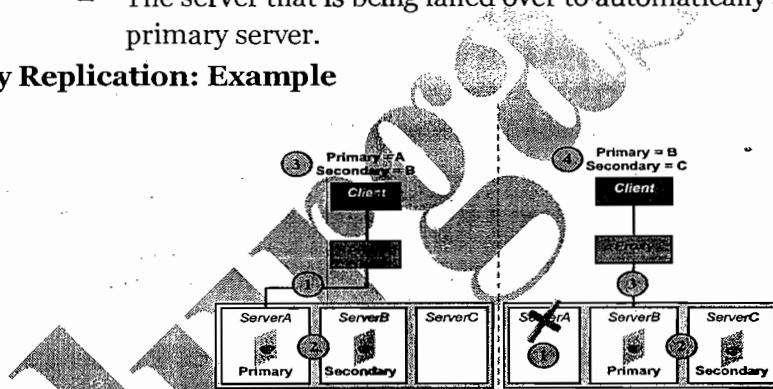
- Every update to the primary session is automatically replicated on the secondary server, either:
 - Synchronously (default)
 - Asynchronously (batch)



In-Memory Replication and Proxy Servers

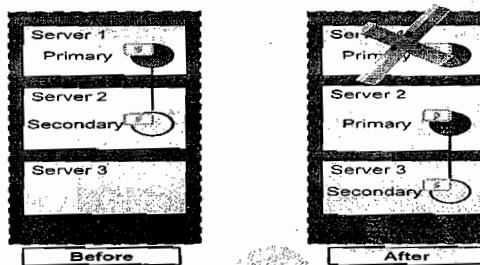
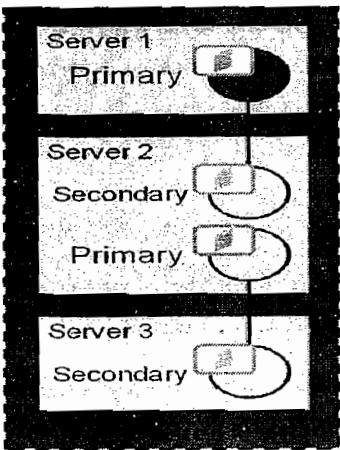
- Oracle WebLogic Server uses nonpersistent cookies to track the primary and secondary servers for each client.
- Subsequent requests from the same client must be directed to the same primary server by the proxy.
- WLS Web server plug-ins support both cookie values
 - There is one cookie with a value that contains the JVMID of the primary and secondary servers
- Network hardware that does not support the secondary cookie still functions effectively:
 - The server that is being failed over to automatically assumes the role of the primary server.

In-Memory Replication: Example

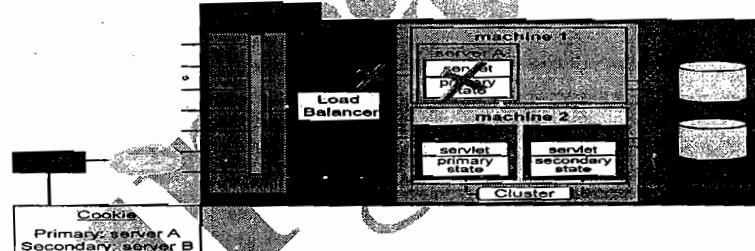


In-Memory Replication

- 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.



Failover with Load balancer



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 either:
 - The load-balancing hardware (WLS aware)
 - A collection of Web servers, or a single Web server, with WebLogic proxy plug-ins (configured identically)
 - Oracle WebLogic Server configured with HTTPClusterServlet

Configuring In-Memory Replication

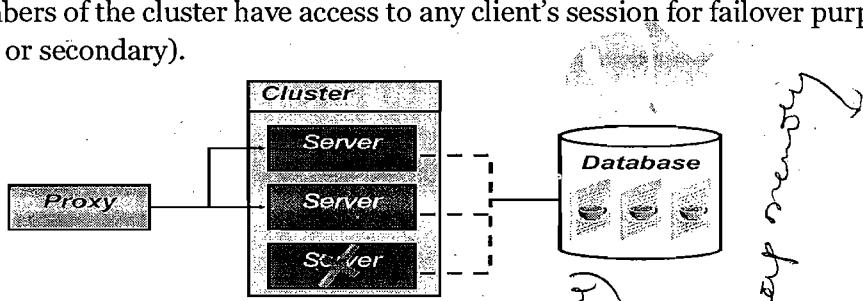
1. Configure the proxy server (if applicable).
2. Optionally define replication groups and/or machines.
3. Specify the persistence type in the weblogic.xml deployment descriptor; the options include:
 - replicated
 - replicated_if_clustered

- async-replication-across-cluster

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

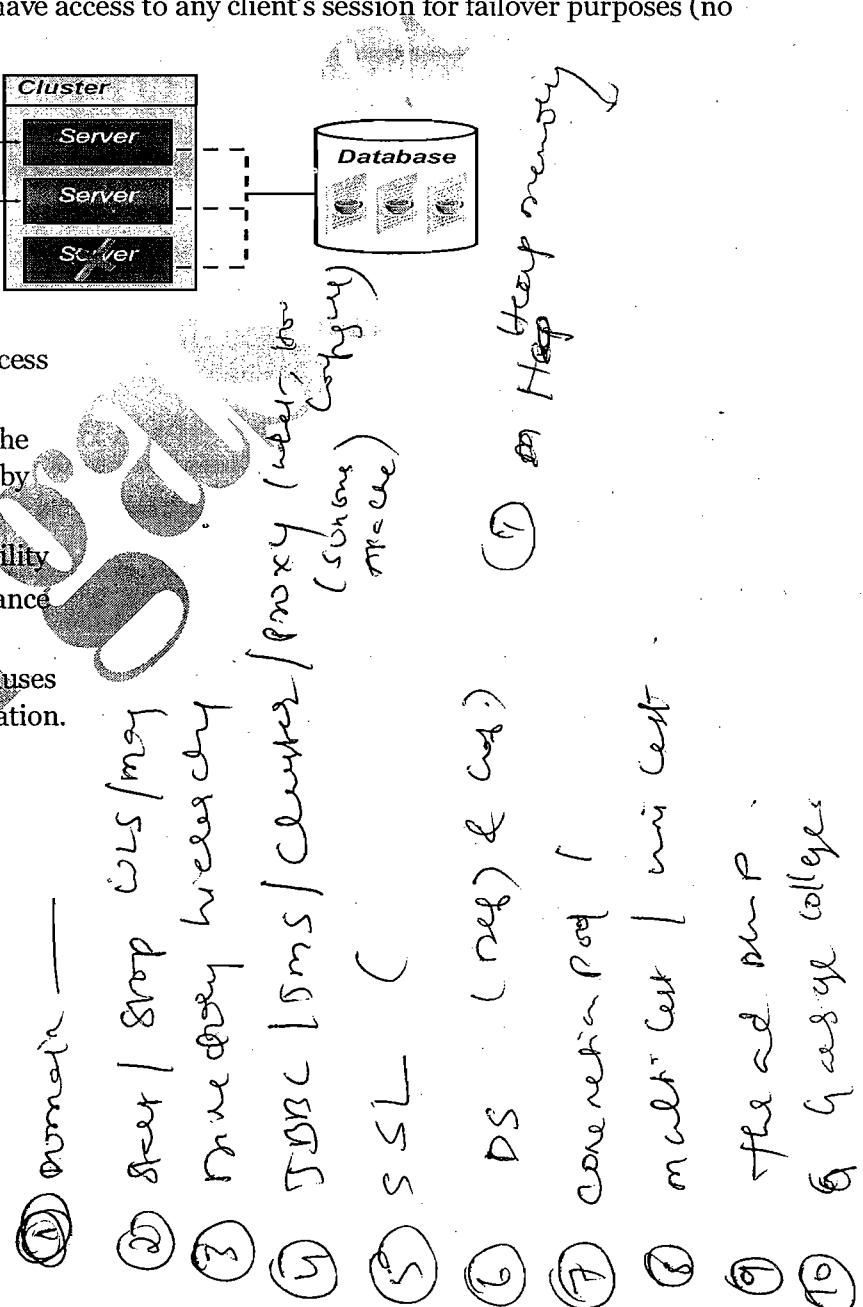
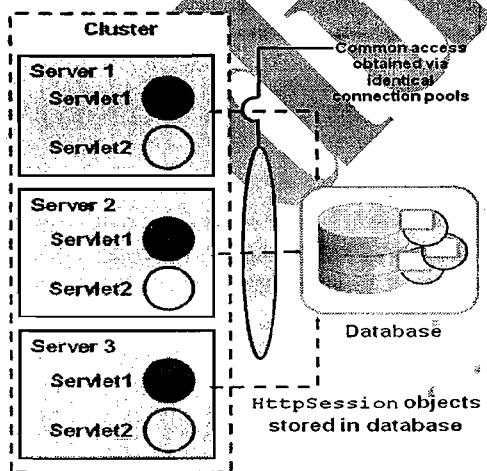
JDBC Replication

- HTTP sessions are 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).



Persistent JDBC Replication

- 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.



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>
```

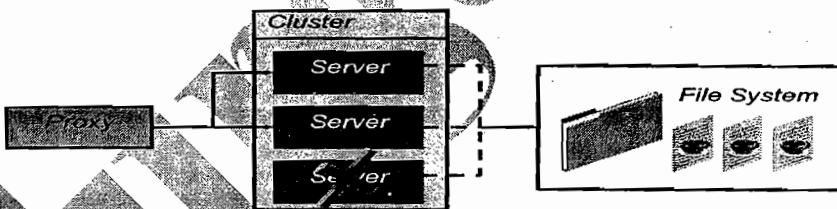
JDBC Persistent Table Configuration

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

Column Head	Column Data Type
WL_ID	char, 100 variable width char
WL_CONTENT_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

File Replication

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



File Persistence

- 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

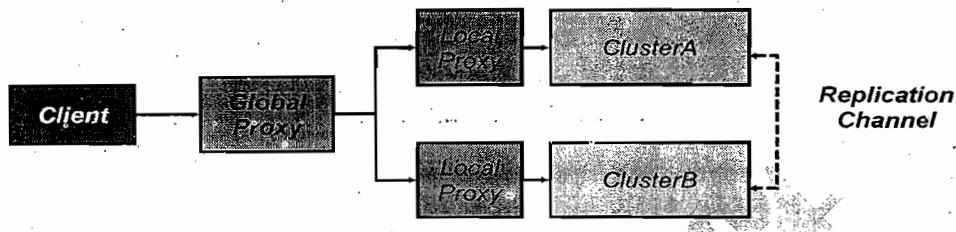
1. Create a folder shared by all servers on the cluster on a highly available file system.
2. Assign read/write privileges to the folder.
3. Configure file session persistence in the weblogic.xml deployment descriptor.

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

Cross-Cluster Replication

WebLogic provides the ability to replicate HTTP sessions across two clusters in separate domains:

- This is most applicable to clusters that are distributed geographically.
- Configure a global proxy to direct clients back to the same cluster ("cluster affinity").
- Configure a specific network channel for cross-cluster communication.

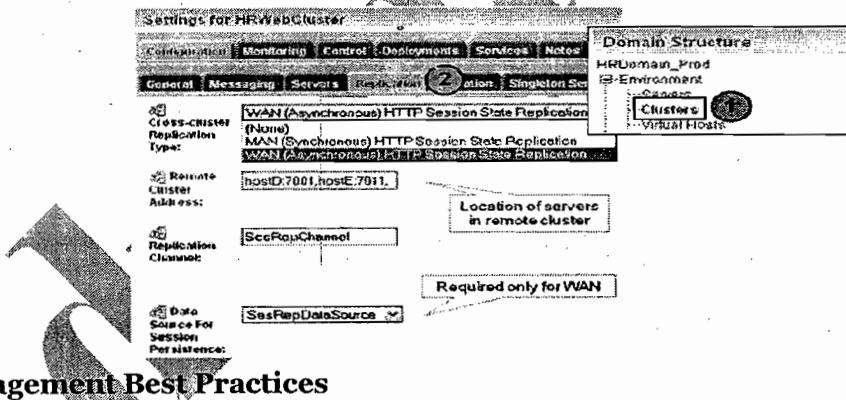


MAN and WAN Replication

Select from two cross-cluster replication strategies:

- *Metropolitan Area Network (MAN)*
 - Low latency network
 - Synchronous in-memory replication to secondary server within another cluster
- *Wide Area Network (WAN)*
 - High latency network
 - Synchronous in-memory replication to secondary server within the same cluster
 - Asynchronous JDBC replication for availability in remote clusters

Configuring Cross-Cluster Replication



State Management Best Practices

- Create WLS machines if you are replicating the state across servers on different physical machines.
- Use Replication Groups to define 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.

Server Version

- All servers in a cluster must have the same major version number.

- Servers can have different minor version numbers and service packs.
 - The Administration Server should also have the same major version number.

Multicast

- If there is an IP multicast problem, WLS starts but does not join a cluster.
 - To verify that multicast is working, run the `utils.MulticastTest` utility.

Valid IP Multicast Addresses:
224.0.0.0 to 239.255.255.255

IP Multicast Test Utility:

```
java utils.MulticastTest -N <name> -A <multicastAddress>
```

Test Multicast: Example

CLASSPATH

- The CLASSPATH value must be the same on all Managed Servers in the cluster.
 - CLASSPATH is set by the setDomainEnv, setWLSEnv, and commEnv scripts when the startManagedWebLogic file is run.

Garbage Collection

- The frequency and length of the garbage collection can affect a cluster.
 - If garbage collection takes too long, the servers will not be able to make the heartbeat signals.
 - Heap allocation can be tuned to adjust the length of the garbage collection.
 - This can be changed using the `-Xms` and `-Xmx` parameters in the server startup script.

Quiz

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

1. file
 2. replicated
 3. unicast
 4. async-replication-across-cluster
 5. jdbc
 6. async-wan

Answer: 1, 2, 4, 5

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

Answer: 2, 5

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

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

Answer: 4**Summary**

In this lesson, you should have learned how to:

- Deploy applications to a cluster
- Manage and configure session state in a cluster
- Troubleshoot common issues in a cluster

Clustering EJB Objects

Objectives

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

- Describe the clustering capabilities of EJBs
- Configure clusterable EJBs
- Describe EJB clustering best practices

Road Map

- EJB Clustering Capabilities
 - Levels of Clustering
 - Load Balancing Algorithms
- Clustering Session EJBs
- Clustering Entity EJBs

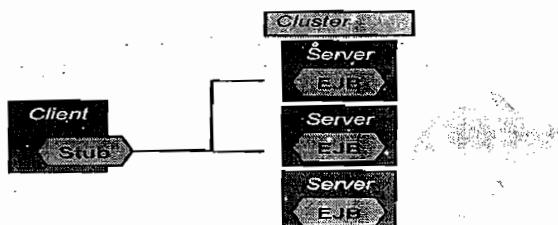
Oracle WebLogic Server EJB Clustering Capabilities

- Oracle WebLogic Server allows load balancing and failover of EJBs.
- EJB clustering is transparent to:
 - The Bean developer
 - The client application developer

- WLS EJBs can be clustered by configuring them in the weblogic-ejb-jar.xml file.

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.



Levels of Clustering for EJB

- Load balancing determines which server:
 - Processes the initial lookup
 - Is used to create or locate an EJB
 - Is used for calling the business methods
- Failover:
 - For a home skeleton, it determines how method calls are routed in a cluster.
 - For a remote skeleton, it determines whether to re-execute a business operation on a different server.

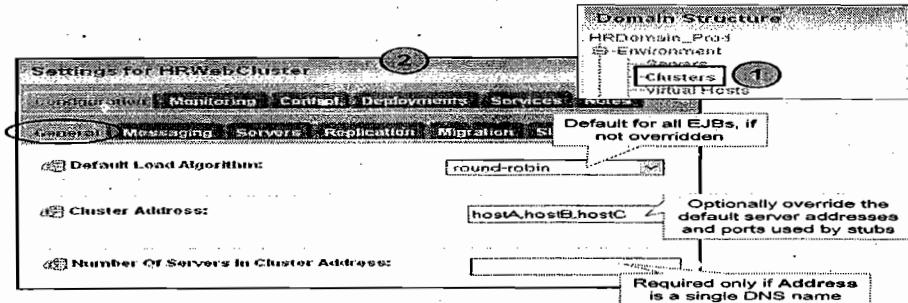
Load-Balancing Clustered EJB Objects

- Oracle WebLogic Server clusters support several algorithms for load-balancing clustered EJB objects:
 - Round-robin
 - Weight-based
 - Random
 - Parameter-based routing (programmatic)
- Server affinity minimizes the number of IP sockets that are opened between the clients and servers in a cluster.

Stateless Session Bean: Load Balancing and Failover

- Stateless session beans of the same type are identical and they hold no state:
 - Beans in different servers are still the same.
 - Separate method invocations can be sent to different servers.
- This does not apply to the stateless session bean methods that fail during execution.

Configuring EJB Clustering



Configuring EJB Clustering

Snippet from META-INF\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>
```

Configuring Clusterable Stateless Session EJBs

The WLS-specific deployment descriptor has a tag for configuring stateless session EJB clustering parameters.

Snippet from META-INF\weblogic-ejb-jar.xml:

```
<!-- Other Tags As Appropriate Here -->
<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>
```

Configuring Stateless Session Clusterable EJBs

Example of a clustered stateless session EJB:

```
<!-- LAST TAG inside <weblogic-ejb-jar> -->
<idempotent-methods>
  <method> <!-- can be replicated -->
    <ejb-name>exampleSession</ejb-name>
    <method-inter-type>remote</method-inter-type>
    <method-name>processUser</method-name>
    <method-params>
      <method-param>java.lang.String</method-param>
    </method-params>
  </method>
</idempotent-methods>
</weblogic-ejb-jar>
```

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.

Configuring Clusterable Stateful Session EJBs

- The WLS-specific deployment descriptor has a tag for configuring the stateful session EJB clustering parameters.
- A snippet from META-INF\weblogic-ejb-jar.xml:

```
<!-- Other Tags As Appropriate Here -->
<!-- Other Tags As Appropriate Here -->
<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>
```

Read-Write 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

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>
```

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.

Summary

In this lesson, you should have learned to:

- Describe how EJBs interact with clusters
- Configure clusterable session EJBs
- Configure clusterable entity EJBs
- Describe EJB clustering best practices

Clustering Services

Objectives

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

- Set up whole server migration
- Describe JNDI clustering
- Describe JDBC clustering
- Migrate a JMS server
- Migrate transactions

Road Map

- Clustering Services
 - Clusterable Services
 - Service Level Migration
 - Server Level Migration
- JNDI
- JDBC
- Transactions
- JMS

Services That Can Be Clustered

- A clustered service is an API or interface that is available on multiple servers in a cluster.
- Oracle WebLogic Server provides clustering services for:
 - Web applications
 - EJB and RMI objects
 - JNDI tree
- Oracle WebLogic Server provides limited clustering services for:
 - JDBC connections
 - JMS destinations

Service Failover

- The services that are deployed to all the servers in a cluster automatically have transparent failover.
- JMS and JTA cannot be deployed to all the servers in a cluster.
 - They are pinned services.

- There are two ways to handle JMS and JTA failover:
 - Service-level migration
 - Server-level migration

Service-Level Migration

- Service-level migration is performed manually.
- Pinned services are moved from the failed server to a new server.
 - The new server takes over and processes any data that is left over when the server failed, and then takes over any future processing.
- A migratable service is accessed by clients by using a migration-aware RMI stub.
- Oracle WebLogic Server supports the automatic migration of user-defined singleton services.

Server-Level Migration

- Server-level migration moves the whole clustered server instance in its entirety.
 - It is an alternative to service-level migration.
 - Service-level migration and server-level migration should not be used together.
- Server-level migration is:
 - Supported using the Java-based Node Manager. This is supported from 9.2 MP2 and 10.x.
 - Supported on Windows (with the wlsifconfig.cmd script)
- You can perform server-level migration both manually and automatically.
- When a migratable server fails:
 - The server is restarted on the same machine
 - If it cannot be started on the same machine, it moves to another machine

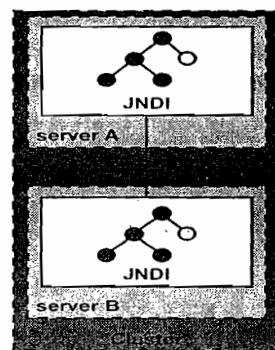
JNDI Clustering Support

- The JNDI tree contains factory objects for accessing the following items:
 - JDBC DataSources
 - EJBs, RMI objects
 - JMS connection factories
- Oracle WebLogic Server replicates clusterable objects to all the servers on a cluster, transparently.

Clusterwide JNDI Service

Each WLS server issues messages that announce the following events:

- A new object bound into the naming tree
- An object removed from the naming tree
- An object rebound (updated) into the naming tree



JNDI Naming Conflicts

- A naming conflict generally occurs when a server instance attempts to bind a service to the JNDI when that name already exists.
 - The services are nonclustered.
- In a cluster, a naming conflict can occur when the server tries to bind a clustered object with the same name as a nonclustered object.
 - The object will be bound to the JNDI locally.
 - Other servers will not bind the replica-aware stub.
- To avoid naming conflicts:
 - Deploy all clusterable objects to all the servers in the cluster.
 - Deploy to the cluster itself, not the individual servers.

JDBC Clustering

- The following JDBC objects are clusterable:
 - Data sources
 - Multi data sources
- When you target a JDBC data source, a new instance of the data source is created on the target.
 - Single server target – A data source instance is created on the server.
 - Cluster target – A data source instance is created on all member servers in the cluster.
- Clustering your JDBC objects does not enable failover of connections but it can ease the process of reconnection when a connection fails.

Clustering Versus Multi Data Sources

Feature	Cluster	Multi Data Source
Failover	None	On the same server (High-Availability Algorithm only)
Load Balance	None	On the same server (Load-Balance Algorithm only)

Targeting a DataSource to a Cluster

The screenshot shows the Oracle Database Control interface. On the left, there's a tree view labeled "Domain Structure" with nodes like "Administration", "Components", "Database", "Performance", "JDBC", "Data Sources", "Independent JDBC", "Data Sources F", "Foreign JNDI Prod", "Work Consistency", and "Clustered environments". The main area is titled "Summary of JDBC Data Sources" and contains a table with columns "Name", "Type", "Status", and "Actions". There are two rows: one for "AdminServer" (Status: OK) and one for "dizzy3" (Status: OK). At the bottom, there's a note about JDBC data sources being JNDI entries and a link to "Data Sources (More Columns Exist)".

This screenshot shows the "Settings for JDBC Data Source - d" configuration page. At the top, there are tabs for Configuration, Target, Monitoring, Control, Security, and Notes, with "Configuration" selected. Below the tabs is a "Save" button. The main content area has sections for "Servers" and "Clusters". Under "Servers", "AdminServer" and "dizzy3" are listed. Under "Clusters", "Cluster-0" is selected, with options "All servers in the cluster" and "Part of the cluster" checked. "dizzy1" and "dizzy2" are listed under "Part of the cluster". A note at the bottom says "This page allows you to select the servers or clusters on which you want to target this data source.".

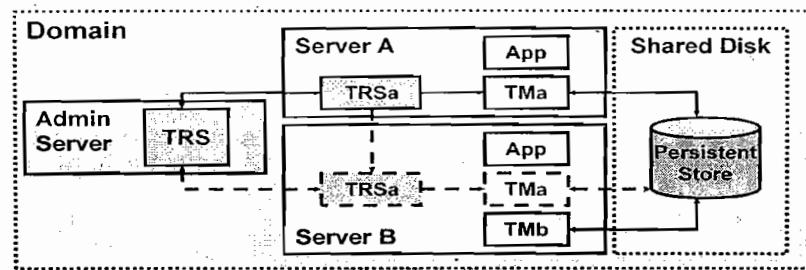
Transaction Recovery After Failure

- Transaction Manager (TM) makes every effort to resolve prepared transactions even after multiple crashes.
- TM uses the Transaction Recovery Service (TRS) which:
 - Automatically attempts to recover transactions on system startup
 - Owns the transaction log for a server
 - Parses all log files on startup for incomplete transactions and completes them

Transaction Recovery Service

When a server restarts after a crash or if the JTA is migrated, the TRS attempts to:

- Resolve prepared transactions in T-Logs
- Maintain consistency across resources
- Persist in achieving transaction resolution
- Report heuristic completions



Recovering JTA Without a Cluster

- To recover transactions from a nonclustered server, perform the following:
 - Make the persistent store for the failed server available to the new server.
 - Set the path for the default persistent store to the path to the data file.
 - Start the new server.
- The TRS searches all transaction log files for incomplete transactions.

Recovering Transactions in a Cluster

When you recover transactions in a clustered server, you have two options:

- Transaction Recovery Service migration
 - The TRS on the backup server takes ownership of the transaction log from the crashed server.
 - The TRS searches all the transaction log files from the failed server and attempts to complete in-flight transactions.
 - If the TRS on the backup server successfully completes all the incomplete transactions, the server releases ownership of the TRS.
- Whole Server migration
 - The server is migrated in its entirety, along with the services that it hosts.

Recovering Transactions for a Failed Clustered Server (Manually)

Manually migrate the TRS from the crashed server to another server in the cluster by using the Administration Console or the command-line interface:

- The TRS on the backup server takes ownership of the transaction log from the failed server.
- The TRS searches the transaction log files from the failed server for incomplete transactions and completes them.
- If the TRS on the backup server successfully completes all the incomplete transactions from the failed server, it releases ownership of the TRS (including the transaction log files) for the failed server, so that the failed server can reclaim it upon restart.

Migrating JTA

Migrating JTA can be done in two ways:

- Manual
- Automatic

Configuring JTA Service Migration

- Before you manually migrate JTA, you define which servers are available for migration.
- To set up the candidate servers, use:
 - **Environment > Servers > Configuration > Migration**

The screenshot shows the 'JTA Candidate Servers' configuration page. On the left, there's a sidebar with 'Settings for dizzy1' and tabs for Configuration, Protocols, Logging, Debug, Monitoring, Control, Deployments, Services, Security, and Notes. Below these are links for General, Cluster, Services, Keystores, SSL, Federation Services, Deployment, Monitoring, and Testing. At the bottom are StartStop, Remote Start Output, Migration, and Jobs buttons. On the right, there are two lists: 'Available' which contains 'dizzy1' and 'dizzy2', and 'Chosen' which is currently empty.

Manually Migrating the JTA to Another Server in a Cluster

When you need to migrate transactions from a failed server to a working server, use:

Environment > Servers > Control > Migration

The screenshot shows the 'JTA Transaction Recovery Service Migration' dialog box. It has sections for 'Preferred Server' (set to 'dizzy1') and 'Hosting Server' (also set to 'dizzy1'). Below these is a 'Migrate to Server:' dropdown menu with the option '(Take no action)' selected.

Configuring Automatic Migration of the JTA Transaction

To configure automatic migration of the Transaction Recovery Service for a migratable target within a cluster:

- Configure the Managed Servers and Node Manager:

- Create the Managed Servers.
 - Create and configure the machines.
 - Configure Node Manager.
- Configure the Migration Basis.
 - On the Cluster > Configuration > Migration page, configure the cluster's "Migration Basis" according to your data persistence environment configuration (use either Database leasing or Consensus leasing).
- Enable Automatic JTA Migration.
- Configure the default persistent store for the migration of the Transaction Recovery Service.
- Restart the Administration Server and Managed Servers with modified migration policies.

Automatic Failback of the Transaction Recovery Service Back to the Original Server

There are two scenarios for automatic failback of the Transaction Recovery Service to the primary server:

- Automatic failback after recovery is complete:
 - If the backup server finishes recovering the TLOG transactions before the primary server is restarted, it initiates an implicit migration of the Transaction Recovery Service back to the primary server.
 - For both manual and automatic migration, the post-deactivation script would be executed automatically.
- Automatic failback before recovery is complete:
 - If the backup server is still recovering the TLOG transactions when the primary server is started during the Transaction Recovery Service initialization of the primary server startup, it initiates an implicit migration of the Transaction Recovery Service from the backup server.

JTA Migration Limitations

Migrating the JTA has the following limitations:

- The JTA cannot be migrated to a backup server from a running server.
 - The server must be stopped before migrating the JTA.
- The backup server processes only incomplete transactions.
 - The backup server does not accept new transaction work for the failed server.
- The backup server does not process heuristic log files.

JMS Clustering Support

- Oracle WebLogic Server supports targeting JMS connection factories to a cluster.
- Oracle WebLogic Server supports the distribution of JMS destinations throughout a cluster (distributed destinations).
 - JMS queues and topics are still managed by a single server instance in the cluster.
- Oracle WebLogic Server provides failover for JMS messages through:
 - Distributed destinations

- Server-level automatic migration

JMS Connection Factory Clustering

Domain Structure

- Machines
- Work Managers
- Startup & Shutdown Classes
- Deployments
- Services
 - Messaging
 - JMS Servers
 - Store-and-forward Agents
 - JMS Modules
 - Path Services
 - Bridges
- JDBC

JMS Modules

JMS system resources are configured and stored as modules similar to standard J2EE modules. Such resources include queues, topics, connection factories, templates, destination keys, quota, distributed queues, distributed topics, foreign servers, and JMS store-and-forward (SAF) parameters. You can administratively configure and manage JMS system modules as global system resources.

This page summarizes the JMS system modules that have been created for this domain.

Customize this table

Name	Type
SystemModule-0	System
SystemModule-1	System

Show 1 to 2 of 2 Previous | Next

Settings for SystemModule

Select this page to select the server or cluster on which you would like to reconfigure changes later if you wish.

Servers

1st AdminServer
1st dclp

Clusters

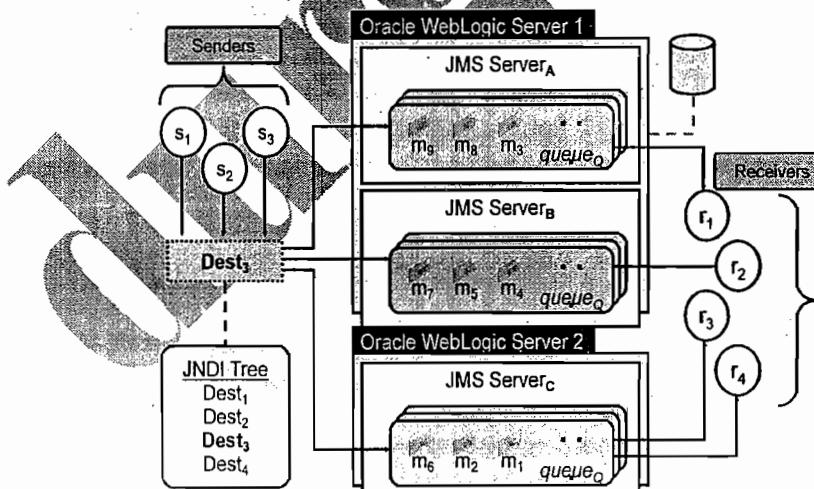
Cluster 0

- All servers in the cluster
- Part of the cluster
- dclp1
- dclp2

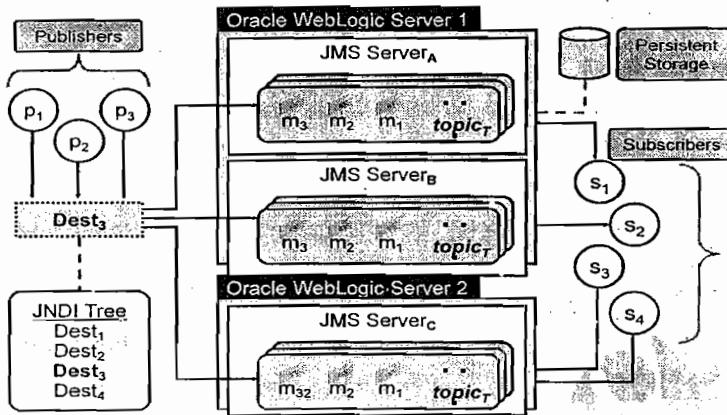
Distributed Destination

- A distributed destination has the following properties:
 - It defines multiple destinations as part of one distributed destination.
 - It is looked up as a regular destination via JNDI.
 - Its member availability is dynamically updated.
- Producers and consumers can send and receive messages through a distributed destination.
- WebLogic JMS distributes the load across the available physical destinations within the distributed destination.

Distributed Queues



Distributed Topics



Creating a Distributed Topic

The screenshot shows the Oracle WebLogic Administration Console interface for creating a new JMS System Module Resource. The 'Create a New JMS System Module Resource' dialog is open, showing options for Connection Factory, Queue, Topic, or Distributed Queue. The 'Topic' option is selected. The 'Create a New JMS System Module Resource' dialog is also shown, with 'Distributed Topic' selected. A large watermark 'DURGA' is overlaid on the bottom left of the screenshot.

Distributed Destination Threshold and Quota

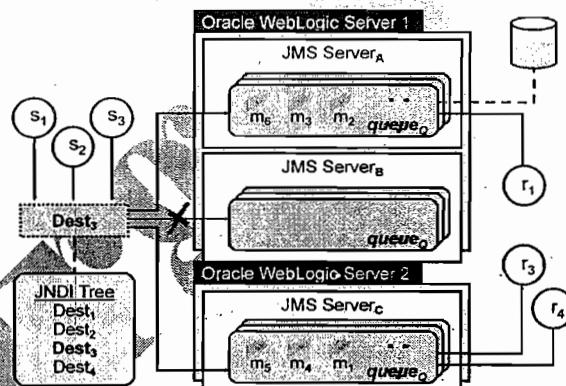
The screenshot shows the Oracle WebLogic Administration Console interface for configuring a distributed topic. The 'Settings for Distributed Topic-1' configuration page is displayed. The 'Thresholds' tab is selected, showing byte threshold high (9223372036854775807) and low (9223372036854775807), and message threshold high (9223372036854775807) and low (9223372036854775807). The 'Quotas' tab is also visible, showing a quota of 'None' and a maximum message size of 2147483647.

Server Affinity

- WebLogic tries to establish new connections to the same servers as existing connections.
- Server Affinity applies only to distributed destinations.
- For a new JMS connection, a distributed destination:
 - Attempts to create a connection on the same server as the JNDI initial context, which is the original client connection server instance
 - Attempts to create a connection to a server with which the client already has a connection, JMS or non-JMS. This may be dependent on the JTA or JMS transaction.
 - Creates the connection on any server

Zero Consumers

- Server affinity favors members on the same server as the client.
- Transaction affinity favors members that are already involved in a transaction.
- A zero consumer is a member that has no consumers; it is used as follows:
 - A producer avoids a zero consumer.
 - A consumer searches for a zero consumer.
- Persistent messages are put in a backing store.



Load Balancing

- If the destination is not distributed:
 - All production is directed at one destination
 - The consumer uses that destination
 - Load balancing does not occur
- If the destination is distributed:
 - Production is replicated across member topics and queues
 - Load balancing is either round-robin or random
 - The consumer is pinned to one member destination, which is selected at creation time

Location Transparency

- Standard destinations can be replicated in the clusterwide JNDI service.
- Select the Replicate JNDI Name in Cluster field in the Administration Console.

- Use the isJNDINameReplicated() method in JMSQueueMBean or JMSTopicMBean.
- Clients look up a destination on any server and receive a connection on the actual server.

Failover

- Failover is both automatic and manual.
- A manual failover includes migrating these elements:
 - An entire JMS server
 - A persistent store
 - A transaction log
- WLS provides a migratable JMS service that attempts to deliver outstanding JMS messages.

JMS Server Migration

- JMS is an “exactly-once” service; each JMS server exists on exactly one Oracle WebLogic Server.
- When an Oracle WebLogic Server fails, its JMS servers can be migrated. However, migration must be configured ahead of time.
- For persistent messaging, migratable targets must have access to the same JMS store as the original server.
- A JMS server can migrate to an Oracle WebLogic Server that already hosts distributed destination members.
- Migration may be a part of scheduled maintenance.

Performing Migration

- Services that are migrated to a nonrunning server are started when the server starts.
- Migration can be manually or automatically initiated.
 - Automatic migration happens only through whole server migration.
- It is possible to use third-party products to perform migration:
 - JMX
 - Veritas HA

JMS Migratable Targets

- The JMS server can be a target to a different independent Oracle WebLogic Server instance or to a migratable target in a WebLogic cluster.
- A migratable target is a special target that can migrate from one server in a cluster to another.
- A migratable target provides a way to group migratable services that should move together.
- When the migratable target is migrated, all services hosted by that target are migrated.

Migrating JMS Data

- For the JDBC store:
 - If the database is on the failed server, migrate the database to the new server and change the JDBC URL for the JDBC store’s DataSource.

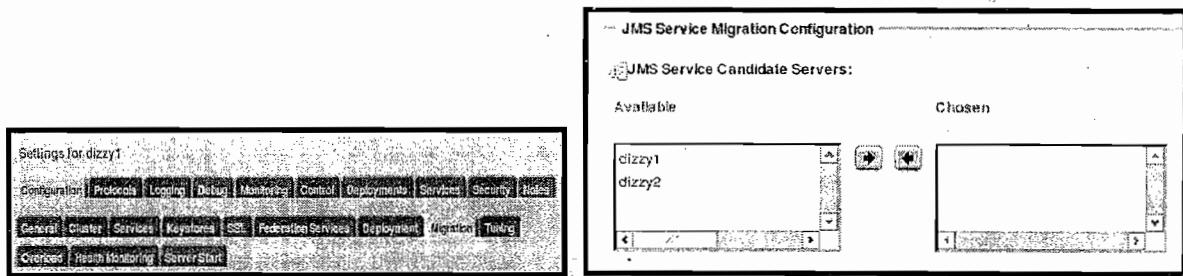
- If the database is not on the failed server, no changes are required.
- For the file store:
 - Migrate to a new server.
 - Ensure that the path name is the same on the new server as the original one.
- For transactions, also migrate the transaction logs.

Migration Configuration

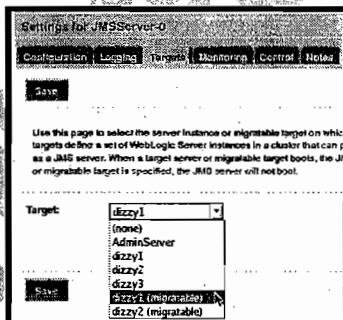
To configure a JMS migration, perform the following:

- Define migratable target servers in the cluster.
- Create JMS servers and assign to the migratable targets.
- Create physical member destinations (use the autodeploy option for distributed destinations).

Configuring a Migratable Target



Targeting JMS Server to a Migratable Target



Migrating Services

Summary

In this lesson, you should have learned how to:

- Migrate a whole server
- Cluster JNDI objects
- Cluster JDBC data sources
- Cluster transactions
- Cluster JMS connection factories

cluster

Weblogic Application Server FAQs

1. What is application server?

- An application server is a software framework dedicated to the efficient execution of procedures (scripts, routines, programs ...) for supporting the construction of applications. The term was created in the context of web applications. In these, the application server acts as a set of components accessible to the software developer through an API defined by the platform itself. These components are usually performed in the same machine where the web server is running, and their main job is to support the construction of dynamic pages.
- Other uses of the term can refer to:
 - ↳ the services that a server makes available
 - ↳ the computer hardware on which the services run
- **Java application servers**
- Following the success of the Java platform, the term application server sometimes refers to a J2EE or Java EE 5 application server. Some of the better-known Java Enterprise Edition application servers include:
 - ↳ Apache Tomcat (Apache Software Foundation)
 - ↳ Tcat Server (MuleSoft)
 - ↳ WebSphere Application Server and WebSphere Application Server Community Edition (IBM)
 - ↳ Sybase Enterprise Application Server (Sybase Inc)
 - ↳ WebLogic Server (Oracle)
 - ↳ JBoss (Red Hat)
 - ↳ Apache Geronimo (Apache Software Foundation)
 - ↳ Oracle OC4J (Oracle)
 - ↳ SAP Netweaver AS (ABAP/Java) (SAP)
 - ↳ WebObjects (Apple Inc.)
- The web modules include servlets and JavaServer Pages. Business logic resides in Enterprise JavaBeans (EJB-3 and later). The Hibernate project offers an EJB-3 container implementation for the JBoss application server. Tomcat from Apache and JOnAS from ObjectWeb exemplify typical containers which can store these modules.
- A Java Server Page (JSP) (a servlet from Java) executes in a web container — the Java equivalent of CGI scripts. JSPs provide a way to create HTML pages by embedding references to the server logic within the page. HTML coders and Java programmers can work side by side by referencing each other's code from within their own. JavaBeans are the independent class components of the Java architecture from Sun Microsystems.

2. What is web server?

- The primary function of a web server is to deliver web pages (HTML documents) and associated content (e.g. images, style sheets, JavaScript's) to clients. A client, commonly a web browser or web crawler, makes a request for a specific resource using HTTP and, if all goes well, the server responds with the content of that resource. The resource is typically a real file on the server's

secondary memory, but this is not necessarily the case and depends on how the web server is implemented.

- While the primary function is to serve content, a full implementation of HTTP also includes a way of receiving content from clients. This feature is used for submitting web forms, including uploading of files.
- Many generic web servers also support server-side scripting (e.g. Apache HTTP Server and PHP). This means that the behavior of the web server can be scripted in separate files, while the actual server software remains unchanged. Usually, this functionality is used to create HTML documents on-the-fly as opposed to return fixed documents. This is referred to as dynamic and static content respectively.

➤ **History of web servers**

- In 1989 Tim Berners-Lee proposed to his employer CERN (European Organization for Nuclear Research) a new project, which had the goal of easing the exchange of information between scientists by using a hypertext system. As a result of the implementation of this project, in 1990 Berners-Lee wrote two programs:

- ❖ a browser called Worldwide Web;
- ❖ the world's first web server, later known as CERN httpd, which ran on NeXTSTEP.
- Between 1991 and 1994 the simplicity and effectiveness of early technologies used to surf and exchange data through the World Wide Web helped to port them to many different operating systems and spread their use among lots of different social groups of people, first in scientific organizations, then in universities and finally in industry.
- In 1994 Tim Berners-Lee decided to constitute the World Wide Web Consortium to regulate the further development of the many technologies involved (HTTP, HTML, etc.) through a standardization process.

➤ **Common features**

- ❖ Virtual hosting to serve many web sites using one IP address.
- ❖ Large file support to be able to serve files whose size is greater than 2 GB on 32 bit OS.
- ❖ Bandwidth throttling to limit the speed of responses in order to not saturate the network and to be able to serve more clients.

3. What is the difference between Web server and Application Server?

Application Server

- Webserver serves pages for viewing in web browser, application server provides exposes business logic for client applications through various protocols
- Webserver exclusively handles http requests. Application server serves business logic to application programs through any number of protocols.

- Webserver delegation model is fairly simple, when the request comes into the webserver, it simply passes the request to the program best able to handle it(Server side program). It may not support transactions and database connection pooling.
- Application server is more capable of dynamic behavior than webserver. We can also configure application server to work as a webserver. Simply application server is a superset of webserver.

WEB Server

- Web Server serves static HTML pages or gifs, jpgs, etc., and can also run code written in CGI, JSP etc. A Web server handles the HTTP protocol. Eg of some web server are IIS or apache.
- An Application Server is used to run business logic or dynamically generated presentation code. It can either be .NET based or J2EE based (BEA WebLogic Server, IBM WebSphere, JBoss).
- A J2EE application server runs servlets and JSPs (infact a part of the app server called web container is responsible for running servlets and JSPs) that are used to create HTML pages dynamically. In addition, J2EE application server can run EJBs - which are used to execute business logic.
- An Application server has a 'built-in' web server; in addition to that it supports other modules or features like e-business integration, independent management and security module, portlets etc.

4. What is the Domain in Weblogic server?

- Domain is a logically related group of Oracle WebLogic Server resources that are managed as a single unit.
- Domain Provides one point of administration
- Can logically separate:
 - Development, test, and production applications
 - Organizational divisions

5. What are the Domain Restrictions?

- ❖ Each domain requires its own Administration Server.
- ❖ A cluster cannot span multiple domains.
- ❖ The Managed Servers in a domain must run the same version of Oracle WebLogic Server.
- ❖ The Administration Server in a domain must run the same or higher version as Managed Servers in the domain.

6. What is the server?

- A server is an instance of weblogic.Server executing in a Java Virtual Machine (JVM).
- A server:
 - ✓ Runs on a designated Oracle WebLogic Server machine
 - ✓ Has a dedicated amount of RAM
 - ✓ Is multithreaded
- Two types of servers:
 - ✓ Administration Server
 - ✓ Managed Server

7. What is the Administration server?

Administration server is central point of control for a domain. It stores the configuration information and logs for a domain. And it runs the Weblogic Administration console.

8. What is the Managed Server?

Managed server is a server in a domain that is not the Administration server. It contacts the administration server for configuration information. It runs business application in a production environment. It is independent of all other Managed servers in a domain (unless they are not in a cluster). You can have many managed servers in a domain. Individual managed servers are typically added for capacity and application isolation.

If not they are dependent.
How many Admin Server can a domain have.

9. How Administration server and Managed servers will interact?

- ❖ The Administration Server stores the master copy of the domain configuration, including the configuration for all Managed Servers in the domain.
- ❖ Each Managed Server stores a local copy of the domain configuration file.
- ❖ When a Managed Server starts, it connects to the Administration Server to synchronize the configuration.
- ❖ When the configuration is changed, the Administration Server sends the changed configuration to the Managed Servers.

10. What is a Machine in Oracle Weblogic Server?

A Machine in a Oracle Weblogic server is a computer that hosts the Oracle Weblogic Server instances. Runs a supported operating system platform and it is used by Node Manager to restart a failed Managed servers.

11. What is a Cluster in Oracle Weblogic server?

A cluster is a Logical group of Weblogic servers. Oracle Weblogic server provides HIGH AVAILABILITY & LOAD BALANCING with help of cluster.

12. What is a Node Manager?

(A Node Manager is a utility or process running on a physical server that enables starting, stopping, suspending or restarting the Administration and Managed servers remotely. It is not associated with a Domain. [Node manager can start any server instances that are resides on the same physical server].

13. How many ways we can install Oracle Weblogic Server?

You can install Oracle Weblogic server in three different ways.

- GUI mode (extract server103_linux32.bin/double click on server103_win32.exe)
- Console mode (c:\>server103_linux32.bin -mode=console -log=c:\consoleinstal.log)
- Silent mode (c:\> server103_linux32.bin -mode=silent -silent_xml=path_to_silent.xml -log=c:\silentinstal.log)

14. What is the default database for Weblogic server?

Point base is the default database. This database comes along with Weblogic software bundle.

15. How many ways we can configure a domain?

You can configure Oracle Weblogic server domains in two ways.

- Graphical Mode (config.cmd [Scripts are in the <WL_HOME>/common/bin directory])

- Console Mode (config.cmd –mode=console)

16. How many ways you can start Administration server?

You can start the Administration server using 5 ways.

- ✚ weblogic.Server (only in development)
- ✚ Start menu (only Windows)
- ✚ DOMAIN_DIR/bin/startWebLogic.sh (Screenshot)
- ✚ WebLogic Scripting Tool (WLST) and Node Manager
- ✚ WLST without Node Manager

(1)
(2)
(3)

17. How many ways you can configure managed servers?

You can configure the Managed server using 3 was.

- ✓ Domain Configuration Wizard
- ✓ Administration Console
- ✓ Command Line (WLST)

18. How many ways you can configure a machines?

You can configure machines by using following:

- ✚ Domain Configuration Wizard
- ✚ Administration Console
- ✚ Command Line (WLST)

19. How many ways you can start managed servers?

Start Managed Servers by using:

- ✓ weblogic.Server
- ✓ DOMAIN_DIR/bin/startManagedWebLogic.sh
- ✓ Administration Console
- ✓ WLST and Node Manager

20. How do I provide user credentials for starting a server?

When you create a domain, the Configuration Wizard prompts you to provide the username and password for an initial administrative user. If you create the domain in development mode, the wizard saves the username and encrypted password in a boot identity file. A WebLogic Server instance can refer to a boot identity file during its startup process. If a server instance does not find such a file, it prompts you to enter credentials.

If you create a domain in production mode, or if you want to change user credentials in an existing boot identity file, you can create a new boot identity file. WebLogic Server does not support copying a boot identity file from one server root directory to another. For information on creating and using boot identity files, see Boot Identity Files in Administration Console Online Help.

21. Can I start a Managed Server if the Administration Server is unavailable?

By default, if a Managed Server is unable to connect to the specified Administration Server during startup, it can retrieve its configuration by reading a configuration file and other files directly. You cannot change the server's configuration until the Administration Server is available. A Managed Server that starts in this way is running in Managed Server Independence mode. For more information, see Starting a Managed Server When the Administration Server Is Not Accessible in Configuring and Managing WebLogic Server.

22. What is the function of T3 in WebLogic Server?

(T3 provides a framework for WebLogic Server messages) that support for enhancements. These enhancements include abbreviations and features, such as object replacement, that work in the context of WebLogic Server clusters and HTTP and other product tunneling. T3 predates Java Object Serialization and RMI, while closely tracking and leveraging these specifications. T3 is a superset of Java Object. Serialization or RMI; anything you can do in Java Object Serialization and RMI can be done over T3. T3 is mandated between WebLogic Servers and between programmatic clients and a WebLogic Server cluster. HTTP and IIOP are optional protocols that can be used to communicate between other processes and WebLogic Server. It depends on what you want to do. For example: when you want to communicate between a browser and WebLogic Server -- use HTTP, or an ORB and WebLogic Server-IIOP.

23. What is the easiest way to set the classpath?

WebLogic Server installs the following script that you can use to set the classpath that a server requires:

WL_HOME\server\bin\setWLSEnv.cmd (on Windows)
WL_HOME/server/bin/setWLSEnv.sh (on UNIX)

Where WL_HOME is the directory in which you installed WebLogic Server. For more information, see "Setting the Classpath" in the WebLogic Server Command Reference.

24. How do I edit the config.xml file?

The persistent configuration for a domain of WebLogic Servers and clusters is stored in an XML configuration file (config.xml). You can modify this file in the following ways:

- Using the Administration Console.
- If you want to create scripts that automate domain management, use the weblogic.Admin utility. See "weblogic.Admin Command-Line Reference".
- If you want to create Java-based management applications, use the Java Management Extensions (JMX) Application Programming Interface (API). See the Programming WebLogic Management Services with JMX guide.
- If you want to edit the config.xml file directly (not recommended), see the BEA WebLogic Server Configuration Reference.

25. Is there a quick way to create and start a remote Managed Server?

The recommended approach is to use the Domain Configuration Wizard, as described in "Setting Up and Starting Managed Servers on a Remote Machine" in Creating WebLogic

(Diagram showing a domain with multiple managed servers (MS) connected to an administration server (AS). Labels include config.xml, SerializedSystemIni.dat, boot.properties, and ASL:11.)

Configurations Using the Configuration Wizard at
http://download.oracle.com/docs/cd/E13196_01/platform/docs81/configwiz/multi.html.

For a streamlined approach, follow the instructions at "Starting Managed Servers From a WebLogic Server Script" in the Administration Console Online Help.

26. The Tree View pane of the WebLogic Console is not visible in my browser. How do I enable it?

- ❖ Enable the Sun Java Plug-In from the control panel!

27 What is the importance of the Boot Identity file and how will you create it?

- If you create boot identity file, it will not ask the user name and password at server startup time.
- Create a file called boot.properties in the `<DOMAIN_HOME>\servers\<server_name>\security` directory that contains two lines:
✓username=username
✓password=password
- If we create a boot.properties file in prod mode also, for every Admin server.
- The first time you start the server; the server reads the Boot Identity file and overwrites it with an encrypted version of the username and password.
- Thereafter, the server remembers the credentials for subsequent startup cycles.

28. What is the MSI mode in Weblogic? How can you enable and disable this option?

- MSI is nothing but Managed Server Independence.
- By default, Managed Servers can function independently of the Administration Server.
- A Managed Server instance can start in MSI mode if the Administration Server is unavailable.
- Configure MSI mode from the Administration Console.
- To start a Managed Server in MSI mode, perform the following:
 - Ensure that the Managed Server's root directory contains the config subdirectory.
 - If the config subdirectory does not exist, copy it from the Administration Server's root directory.
 - Start the Managed Server at the command line or by using a script.
- Environment > Servers > Server_Name > Tuning > Advanced > Managed Server Independence Enabled check box

29. If the Administration server not available while starting the Managed server which is already enabled MSI, what are the files it will look for?

- If the Administration Server is unavailable at boot time, Managed Servers search for:
 - ✓config.xml
 - ✓SerializedSystemIni.dat
 - ✓boot.properties(optional)
- Each Managed Server looks in its local config directory for config.xml, a replica of the domain's config.xml.
- You cannot change the configuration of the Managed Server that is running in MSI mode until it restores communication with the Administration Server.

*Relation between
of EAS*

30. What if Administration server goes down? What is the behavior of the managed servers? What are all the things will available or not available?

- The Administration Server:
 - ✓ Can go down without affecting the operation of the Managed Servers
 - ✓ Can be restarted when the Managed Servers are still running
- When an Administration Server goes down:
 - ✓ The domain log entries are unavailable while it is down
 - ✓ Managed Servers can start in independent mode
 - ✓ The Administration Console and the management tools are unavailable

31. If an administration server running machine got crashed, how will you restart the server with same configuration on new machine?

- Oracle WebLogic Server allows the creation of a backup of the server as follows:
 - ✓ Install Oracle WebLogic Server on a backup machine.
 - ✓ Copy the application files to a backup machine.
 - ✓ Copy the configuration files to a backup machine.
 - ✓ Restart the Administration Server on a new machine.
- The new Administration Server contacts the Managed Servers and informs them that it is running on a new IP address.

32. How can you run Multiple Weblogic server instances in a same physical machine?

- You can run multiple instances of WLS using different configurations on the same physical machine at the same time by either:
 - ✓ Assigning multiple IP addresses to a machine (multihoming) and defining each server to use a unique IP address
 - ✓ Specifying the same IP address but using different listen ports
- A multihomed machine:
 - ✓ Is a machine with multiple IP addresses
 - ✓ Can run a different WLS instance that is bound to each IP address
 - ✓ Can be used to configure a cluster on a single machine

33. How will you create Domain Template? Explain briefly?

- A domain template defines the full set of resources within a domain.
- Oracle provides sample templates for creating any platform domain.
- There are three ways to create domain templates:
 - ✓ WLST offline command line tool
 - ✓ pack command
 - ✓ Domain Template Builder (config_builder.sh under WL_HOME/common/bin)
- Use the Domain Template Builder to create a domain template or an extension template.
- Using the Domain Template Builder:
 - ✓ Define a domain and replicate it across multiple projects
 - ✓ Distribute a domain packed with an application that has been developed to run in it

34. What are the default Weblogic provided Groups for security realm?

- Administrators
- Deployers
- Operators
- Monitors

- AppTesters
- CrossDomainConnectors
- AdminChannelUsers

35. What are the default Weblogic provided Roles for security realm?

- Admin
- Deployer
- Operator
- Monitor
- AppTester
- CrossDomainConnectors
- AdminChannelUsers
- Anonymous

36. What is the default Weblogic provided domain template file name and location?

Wls.jar is the default domain template and the location is WL_HOME\common\templates\domains

37. What are the elements of the Administration console?

- ✳ Change Center
- ✳ Domain Structure
- ✳ How do I...
- ✳ Tool Bar
- ✳ Breadcrumb Navigation
- ✳ System Status

38. What are the Node elements of the Administration console or Domain Structure?

- ❖ Environment (Servers, Clusters, Virtual Hosts, Migratable Targets, Machines ...)
- ❖ Deployment
- ❖ Services (Messaging, JDBC, Persistent Store, JTA, File T3, jCOM ...)
- ❖ Security Realms
- ❖ Interoperability
- ❖ Diagnostics (Log Files, Diagnostics Modules, Diagnostics Images, Archives, Context)

39. What are the Tool Bar elements in Weblogic?

- ✓ Welcome Message
- ✓ Connected to
- ✓ Home
- ✓ Log Out
- ✓ Preferences
- ✓ Record
- ✓ Help
- ✓ Search

40. How will you enable the Administration Console?

By default, the Administration Console is enabled. If you disable it, you can re-enable it using the WebLogic Scripting Tool (WLST). Start the Administration Server, then invoke WLST and use the following commands:

Using WLST to Re-enable the Console
connect("username","password")
edit()
startEdit()
cmo.setConsoleEnabled(true)
save()
activate()

The following attribute(s) have been changed on MBeans which require server re-start.MBean Changed :
com.bea:Name=mydomain,Type=Domain Attributes changed :

ConsoleEnabled
Activation complete
dDisconnect()
exit()

41. How will you Enable and disable the domain configuration lock?

The Administration Console Change Center provides a way to lock a domain configuration so you can make changes to the configuration while preventing other accounts from making changes during your edit session.

The domain configuration locking feature is always enabled in production domains. It can be enabled or disabled in development domains. It is disabled by default when you create a new development domain.

To enable or disable the domain configuration locking feature in a development domain:

- 4 In the banner toolbar region at the top of the right pane of the Console, click Preferences.
- 4 Click User Preferences.
- 4 Select or clear Automatically Acquire Lock and Activate Changes to enable or disable the feature.
- 4 Click Save.
- 4 After you finish

When you enable domain configuration locking, you must use the Change Center to lock and edit for the domain configuration.

42. What are Dynamic and Non-Dynamic Changes in the Weblogic Console? what is the difference?

Some changes you make in the Administration Console take place immediately when you activate them. Other changes require you to restart the server or module affected by the change. These latter changes are called non-dynamic changes. Non-dynamic changes are indicated in the Administration Console with this warning icon, .

Changes to dynamic configuration attributes become available once they are activated, without restarting the affected server or system restart. These changes are made available to the server and run-time hierarchies once they are activated. Changes to non-dynamic configuration attributes require that the affected servers or system resources be restarted before they become effective.

If a change is made to a non-dynamic configuration setting, no changes to dynamic configuration settings will take effect until after restart. This is to assure that a batch of updates having a combination of dynamic and non-dynamic attribute edits will not be partially activated.

Note that WebLogic Server's change management process applies to changes in domain and server configuration data, not to security or application data.

43. What is the information is going to store in the “security” folder of the Domain directory contents?

This directory holds the security-related files that are the same for every WebLogic Server instance in the domain:

- SerializedSystemIni.dat

This directory also holds security-related files that are only needed by the domain's Administration Server:

- DefaultAuthorizerInit.ldift
- DefaultAuthenticatorInit.ldift
- DefaultRoleMapperInit.ldift

44. What is the use of SerializedSystemIni.dat file in Weblogic?

It is important to protect passwords that are used to access resources in a WebLogic Server domain. In the past, usernames and passwords were stored in clear text in a WebLogic security realm. Now all the passwords in a WebLogic Server domain are hashed. The SerializedSystemIni.dat file contains the hashes for the passwords. It is associated with a specific WebLogic Server domain so it cannot be moved from domain to domain.

If the SerializedSystemIni.dat file is destroyed or corrupted, you must reconfigure the WebLogic Server domain. Therefore, you should take the following precautions:

Make a backup copy of the SerializedSystemIni.dat file and put it in a safe location. Set permissions on the SerializedSystemIni.dat file such that the system administrator of a WebLogic Server deployment has write and read privileges and no other users have any privileges.

45. Explain about Domain Directory Contents?

By default, WebLogic Server creates domain directories under the BEA_HOME/user_projects/domains directory. This section describes the contents of the domain directory and its subfolders. In this section, domain-name, deployment-name, and server-name represent names that you define when you create a domain.

Individual applications in a domain might create additional files and directories in the domain directory.

If you have not yet created a domain, you can see an example of an existing domain directory by looking in WL_HOME/examples/domains/wl_server where WL_HOME is the directory in which you installed WebLogic Server.

Domain-name

The name of this directory is the name of the domain.

autodeploy

This directory provides a quick way to deploy applications in a development server. When the WebLogic Server instance is running in development mode, it automatically deploys any applications or modules that you place in this directory.

The files you place in this directory can be Java EE applications, such as:

- ◆ An EAR file
- ◆ A WAR, EJB JAR, RAR, or CAR archived module
- ◆ An exploded archive directory for either an application or a module

Bin

This directory contains scripts that are used in the process of starting and stopping the Administration Server and the Managed Servers in the domain. These scripts are generally provided as .sh files for UNIX and .cmd files for Windows. The bin directory can optionally contain other scripts of domain-wide interest, such as scripts to start and stop database management systems, full-text search engine processes, etc.

Config

This directory contains the current configuration and deployment state of the domain. The central domain configuration file, config.xml, resides in this directory.

Config/configCache

Contains data that is used to optimize performance when validating changes in the domain's configuration documents. This data is internal to WebLogic Server and does not need to be backed up.

Config/diagnostics

This directory contains system modules for instrumentation in the WebLogic Diagnostic Framework.

Config/jdbc

This directory contains system modules for JDBC: global JDBC modules that can be configured directly from JMX (as opposed to JSR-88).

Config/jms

This directory contains system modules for JMS: global JMS modules that can be configured directly from JMX (as opposed to JSR-88).

Config/lib

This directory is not used in the current release of WebLogic Server.

Config/nodemanager

Learn From Experts

GATE

IITs

PSUs

- BSNL(TO, TTA), ► HAL
- DRDO ► NTPC
- APGENCO ► MTNL
- APTRANSCO ► BEI
- ISRO



Engineering Academy®

23/B RT, 11th Floor, Opp. Andhra Bank, Near Umesh Chandra Statue, S.R. Nagar, Hyderabad - 500 038. e-mail: info@durgasoft.com

Ph: 040-64512786, Gt: 09246212143

www.gatetrainer.com

MASTER IN JAVA TRAINING

SCJP

SCWCD

Core JAVA

Adv. JAVA

STRUTS

EJB

HIBERNATE

PROJECTS

B.Tech / MCA / M.Tech

More than 2000 Members Already Certified (SCJP & SCWCD)

More than 1000 Members got selected for MNC's



Software Solutions®

43/3RT, 11th Floor, Opp. Andhra Bank, Near Umesh Chandrapur,
Shinde S.R.T. Nagar, Hyderabad - 500 038. e-mail Info@durgasoft.com

Ph: 040-64512786, Cell: 0924621243

www.durgasoft.com

This directory holds configuration information for connection to the Node Manager.

Config/security

This directory contains system modules for the security framework. It contains one security provider configuration extension for each kind of security provider in the domain's current realm.

Config/startup

This directory contains system modules that contain startup plans. Startup plans are used to generate shell scripts that can be used as part of server startup.

ConfigArchive

This directory contains a set of JAR files that save the domain's configuration state. Just before pending changes to the configuration are activated, the domain's existing configuration state, consisting of the config.xml file and the other related configuration files, is saved in a versioned JAR file with a name like config.jar#1, config.jar#2, etc.

The maximum number of versioned JAR files to be kept is specified by the archiveConfigurationCount attribute of DomainMBean. Once this maximum number is reached, the oldest conversion archive is deleted before a new one is created.

Console-ext

This directory contains extensions to the Administration Console, which enable you to add content to the WebLogic Server Administration Console, replace content, and change the logos, styles and colors without modifying the files that are installed with WebLogic Server. For example, you can add content that provides custom monitoring and management facilities for your applications. See Extending the Administration Console.

Init-info

This directory contains files used for WebLogic domain provisioning. You should not modify any files in this directory.

Lib

Any JAR files you put in this directory are added to the system classpath of each server instance in the domain when the server's Java virtual machine starts.

Pending

This directory contains domain configuration files representing configuration changes that have been requested, but not yet activated. Once the configuration changes have been activated, the configuration files are deleted from this directory.

Security

This directory holds those security-related files that are the same for every WebLogic Server instance in the domain:

- * SerializedSystemIni.dat

This directory also holds security-related files that are only needed by the domain's Administration Server:

- * DefaultAuthorizerInit.ldift

- ◆ DefaultAuthenticatorInit.ldift
- ◆ DefaultRoleMapperInit.ldift

Servers

This directory contains one subdirectory for each WebLogic Server instance in the domain. The subdirectories contain data that is specific to each server instance.

Servers/server-name

This directory is the server directory for the WebLogic Server instance with the same name as the directory.

Servers/server-name/bin

This directory holds executable or shell files that can be or must be different for each server. The server environment script (setServerEnv.sh or setServerEnv.cmd) is an example of a file that resides here because it can differ from one WebLogic Server instance to the next, for example, depending on whether the server instance has its own startup plan.

Servers/server-name/cache

This directory holds directories and files that contain cached data. By "cached" here we mean that the data is a copy, possibly in a processed form (compiled, translated, or reformatted), of other data.

Servers/server-name/cache/EJBCompilerCache

This directory is a cache for compiled EJBs.

Servers/server-name/data

This directory holds files that maintain persistent per-server state used to run the WebLogic Server instance, other than security state, as opposed to temporary, cached or historical information. Files in this directory are important data that must be retained as the WebLogic Server instance is brought up, is brought down, crashes, restarts, or is upgraded to a new version.

Servers/server-name/data/ldap

This directory holds the embedded LDAP database. The run-time security state for the WebLogic Server instance is persisted in this directory.

Servers/server-name/data/store

This directory holds WebLogic persistent stores. For each persistent store, there is a subdirectory that holds the files that represent the persistent store. The name of the subdirectory is the name of the persistent store. By convention there is one store named default.

Servers/server-name/logs

This directory holds logs and diagnostic information. This information is historical in nature. It is not crucial to the operation of the server, and can be deleted (while the WebLogic Server instance is down, at least) without affecting proper operation. However, the information can be quite useful for debugging or auditing purposes and should not be deleted without good reason.

Servers/server-name/logs/diagnostic_images

This directory holds information created by the Server Image Capture component of the WebLogic Diagnostic Framework.

Servers/server-name/logs/jmsServers

This directory contains one subdirectory for each JMS server in the WebLogic Server instance. Each such subdirectory contains the logs for that JMS server. The name of the subdirectory is the name of the JMS server.

Servers/server-name/logs/connector

This directory is the default base directory for connector module (JCA ResourceAdapter) logs.

Servers/server-name/security

This directory holds security-related files that can be or must be different for each WebLogic Server instance. The file boot.properties is an example of a file that resides here because it can differ from one server to the next. This directory also maintains files related to SSL keys.

Servers/server-name/tmp

This directory holds temporary directories and files that are created while a server instance is running. For example, a JMS paging directory is automatically created here unless another location is specified. Files in this directory must be left alone while the server is running, but may be freely deleted when the server instance is shut down.

Tmp

This directory stores temporary files used in the change management process. You should not modify any files in this directory.

user_staged_config

By default, configuration information is automatically copied from the Administration Server to each Managed Server. If instead you prefer to stage configuration changes manually, you can use this directory as an alternative to the config directory.

46. How many ways you can change the configuration changes?

The change management features of WLS:

- Enable you to distribute configuration changes throughout a domain securely, consistently, and predictably
- Are the same, regardless of whether you are using:
 - * The WLS Administration Console
 - * The WebLogic Scripting Tool (WLST)
 - * The Java Management Extension (JMX) APIs

47. What is the user of WLST in Weblogic?

- The WLS command-line tools are useful:
 - ✓ For automating common administration activities
 - ✓ As an alternative to the Administration Console
 - ✓ When graphical tools are not supported
- WLST provides a command-line interface for:
 - ✓ Creating new WLS domains
 - ✓ Retrieving and updating WLS domain configurations
 - ✓ Deploying applications
 - ✓ Obtaining run-time server statistics

48. How many WLST modules are there? Explain?

➤ Online mode:

- ✓ Connected to a running server
- ✓ Access to all WLS configuration and run-time attributes
- ✓ Create and activate change sessions similar to the WLS console

➤ Offline mode:

- ✓ Domain not running
- ✓ Access to only persisted domain configuration (config.xml)
- ✓ Create or update domains similar to using the Configuration Wizard

49. What is the Node Manager (NM)? Explain briefly?

Node Manager (NM):

- ↳ Starts and stops Managed Servers remotely: server, domain, and cluster
- ↳ Available as either a Java-based or (for UNIX or Linux) a script-based process
- ↳ Monitors and acts on server health
- ↳ Runs on the same computers as the Managed Servers
- ↳ Can be run automatically in the background, as a Windows service or a UNIX daemon

50. What Node manager can do in Weblogic server?

You can use Node Manager to:

- ❖ Start, shut down, and restart an Administration Server
- ❖ Start, shut down, suspend, and restart Managed Servers
- ❖ Automatically restart the Administration and Managed Servers on failure
- ❖ Monitor servers and collect log data

51. How many versions of Node Managers are available?

- There are two versions of Node Manager:
 - ✓ Java-based Node Manager
 - ✓ Script-based Node Manager
- Java-based Node Manager runs within a Java Virtual Machine (JVM) process.
- Script-based Node Manager (used only for UNIX and Linux systems) does not have as much security, but provides the ability to remotely manage servers over a network using Secure Shell (SSH).

52. How Node Manager will work with the Weblogic Server? How will you configure Node Manager in WLS?

- Node Manager must run on each computer that hosts the WLS instances that you want to control with Node Manager.)
- You should configure each computer as a machine in Oracle WebLogic Server, and assign each server instance, which is to be controlled by Node Manager, to the machine that the server instance runs on.
- Node Manager should run as an operating system service, so that it automatically restarts upon system failure or reboot.

~~Ques~~ 53. What is the Node Manager Default Behavior?

- After WebLogic Server is installed, Node Manager is "ready-to-run" if Node Manager and Administration Server are on the same machine.
- By default, the following behaviors are configured:
 - ✓ The Administration Console can use Node Manager to start the Managed Servers.
 - ✓ Node Manager monitors the Managed Servers that it started.
 - ✓ The automatic restart of Managed Servers is enabled.

54. To start Node Manager at system start up time, what we have to do?

We have to configure Node Manager as a Operating System Service.

- ↳ It is recommended that you run Node Manager (NM) as:
 - A Windows service on Windows platforms and
 - A daemon on UNIX platforms
- ↳ Running NM during system startup allows it to restart automatically when the system is rebooted.
- ↳ Node Manager can be configured to start at boot time, as either of these:
 - A Windows service
 - A UNIX daemon

55. How will you configure Node Manager as Windows Service?

- Edit **installNodeMgrSvc.cmd** to specify Node Manager's listen address and listen port.
- Run **installNodeMgrSvc.cmd** to reinstall Node Manager as a service, listening on the updated address and port.
- Delete the Node Manager Service using **uninstallNodeMgrSvc.cmd**.

56. Explain about Weblogic server Log Message Format?

When a WebLogic Server instance writes a message to the server log file, the first line of each message begins with ##### followed by the message attributes. Each attribute is contained between angle brackets.

Here is an example of a message in the server log file:

#####<Jan 05, 2010 10:46:51 AM EST> <Notice> <WebLogicServer> <MyComputer> <examplesServer> <main> <<WLS Kernel>><> <null> <1080575211904> <BEA-000360> <Server started in RUNNING mode> In this example, the message attributes are: **Locale-formatted Timestamp, Severity, Subsystem, Machine Name, Server Name, Thread ID, User ID, Transaction ID, Diagnostic Context ID, Raw Time Value, Message ID, and Message Text.**

If a message is not logged within the context of a transaction, the angle brackets for Transaction ID are present even though no Transaction ID is present.

If the message includes a stack trace, the stack trace is included in the message text.

WebLogic Server uses the host computer's default character encoding for the messages it writes.

57. What is the Log Message Format of Output to Standard Out and Standard Error?

When a WebLogic Server instance writes a message to standard out, the output does not include the ##### prefix and does not include the Server Name, Machine Name, Thread ID, User ID, Transaction ID, Diagnostic Context ID, and Raw Time Value fields.

Here is an example of how the message from the previous section would be printed to standard out:

<jan 01, 2010 10:51:10 AM EST> <Notice> <WebLogicServer> <BEA-000360> <Server started in RUNNING mode> In this example, the message attributes are: **Locale-formatted Timestamp, Severity, Subsystem, Message ID, and Message Text.**

58. How many log Message Severity levels are there in Weblogic? Explain?

The severity attribute of a WebLogic Server log message indicates the potential impact of the event or condition that the message reports.

Severity	Meaning
TRACE	Used for messages from the Diagnostic Action Library. Upon enabling diagnostic instrumentation of server and application classes, TRACE messages follow the request path of a method.
INFO	Used for reporting normal operations; a low-level informational message.
NOTICE	An informational message with a higher level of importance.
WARNING	A suspicious operation or configuration has occurred but it might not affect normal operation.
ERROR	A user error has occurred. The system or application can handle the error with no interruption and limited degradation of service.
CRITICAL	A system or service error has occurred. The system can recover but there might be a momentary loss or permanent degradation of service.
ALERT	A particular service is in an unusable state while other parts of the system continue to function. Automatic recovery is not possible; the immediate attention of the administrator is needed to resolve the problem.
EMERGENCY	The server is in an unusable state. This severity indicates a severe system failure or panic.
DEBUG	A debug message was generated.

59. What is the default log Message Severity levels in Weblogic?

WebLogic Server subsystems generate many messages of lower severity and fewer messages of higher severity. For example, under normal circumstances, they generate many **INFO** messages and no **EMERGENCY** messages.

60. What is the Log Message Severity Level sequence from lowest to highest impact?

A log level object can specify any of the following values, from lowest to highest impact:
TRACE, DEBUG, INFO, NOTICE, WARNING, ERROR, CRITICAL, ALERT, EMERGENCY

61. How will you specify the logging implementation in Weblogic?

We will specify the logging implementation using "**Java Logging API**" or "**Log4j**" in Weblogic.
About Log4j

Log4j has three main components: **loggers**, **appenders**, and **layouts**. The following sections provide a brief introduction to Log4j.

Loggers

Log4j defines a Logger class. An application can create multiple loggers, each with a unique name. In a typical usage of Log4j, an application creates a Logger instance for each application class that will emit log messages. Loggers exist in a namespace hierarchy and inherit behavior from their ancestors in the hierarchy.

Appenders

Log4j defines appenders (handlers) to represent destinations for logging output. Multiple appenders can be defined. For example, an application might define an appender that sends log messages to standard out, and another appender that writes log messages to a file. Individual loggers might be configured to write to zero or more appenders. One example usage would be to send all logging messages (all levels) to a log file, but only ERROR level messages to standard out.

Layouts

Log4j defines layouts to control the format of log messages. Each layout specifies a particular message format. A specific layout is associated with each appender. This lets you specify a different log message format for standard out than for file output, for example.

Java Logging API

WebLogic logging services provide the Commons LogFactory and Log interface implementations that direct requests to the underlying logging implementation being used by WebLogic logging services.

To use Commons Logging, put the WebLogic-specific Commons classes, \$BEA_HOME/modules/com.bea.core.weblogic.common.logging_1.3.0.0.jar, together with the commons-logging.jar file in one of the following locations:

APP-INF/LIB or WEB-INF/LIB directory

DOMAIN_NAME/LIB directory

Server CLASSPATH

Note: WebLogic Server does not provide a Commons logging version in its distribution.

62. What is the user of Log Filters in Weblogic?

Log filters:

- ↳ Control the log messages that get published
- ↳ Are based on the values of message attributes
- ↳ Can be applied to different message destinations:
 - Server log file
 - Server memory buffer
 - Server standard out

- Domain log file

63. What is the user of Network channels in Weblogic?

Adds flexibility to the networking configuration:

- ↳ Multiple NICs for a single WLS server
- ↳ Specific NICs or multiple port numbers on a NIC for specific WLS servers
- ↳ Ability to use multiple IP addresses with each server
- ↳ Ability to use a single IP address with multiple ports
- ↳ Ability to configure the cluster multicast port number independently of the port numbers used by the cluster members
- ↳ Multiple SSL configurations on one server

Network channels:

- ↳ Define the set of basic attributes of a network connection to WLS
- ↳ Can assign multiple channels to a single server (segment network traffic)
- ↳ Can prioritize internal (non-URL) connections
- ↳ Can separate incoming client traffic from internal server to server traffic in a domain
- ↳ A "default" channel gets generated when a server is created.

64. How will you configure a web application in Weblogic?

Web applications are configured using the web.xml and weblogic.xml deployment descriptors, which:

- ✓ Define the run-time environment
- ✓ Map URLs to servlets and JSPs
- ✓ Define application defaults such as welcome and error pages
- ✓ Specify J2EE security constraints
- ✓ Define work managers for applications
- ✓ Set the context root for the application

65. What information will be available in "web.xml" file?

The web.xml file is a deployment descriptor that is used to configure the following:

- ✓ Servlets and JSP registration
- ✓ Servlet initialization parameters
- ✓ JSP tag libraries
- ✓ MIME type mappings
- ✓ Welcome file list
- ✓ Error pages
- ✓ Security constraints and roles
- ✓ Resources
- ✓ EJB references

66. What information will be available in "weblogic.xml" file?

Using weblogic.xml, you can configure the following:

- The application's root context path
- Application logging

- Security role mappings
- Advanced session settings
- Session clustering
- References to shared libraries
- References to server resources (data sources, EJBs, and so on)
- Work managers and threading
- Virtual directories
- JSP compiler options

67. To configure a “web service” Applications in Weblogic, what are all the files required as a deployment descriptor?

A Web service application:

- ✓ Responds to HTTP client requests using the Simple Object Access Protocol (SOAP)
- ✓ Uses the same structure as a Java EE Web application
- ✓ Supports two additional deployment descriptors:
 - ↳ webservices.xml
 - ↳ weblogic-webservices.xml

68. What is the Virtual directory Mappings? Which file you are going to provide these virtual directory mappings?

Virtual directories:

- ✿ Can be used to refer to physical directories
- ✿ Enable you to avoid the need to hard code paths to physical directories
- ✿ Allow multiple Web applications to share common physical directories for specific requests such as images
- ✿ Decrease duplication of files across applications
- ✿ Are configured in weblogic.xml

Example:

```
<virtual-directory-mapping>
  <local-path>c:/usr/gifs</local-path>
  <url-pattern>/images/*</url-pattern>
  <url-pattern>*.jpg</url-pattern>
</virtual-directory-mapping>
<virtual-directory-mapping>
  <local-path>c:/usr/common_jsp.jsp</local-path>
  <url-pattern>*.jsp</url-pattern>
</virtual-directory-mapping>
```

69. What is the deployment descriptor file for ejb applications? What are all the information is going to provide in that file?

Ejb application deployment descriptor file in Weblogic is “weblogic-ejb-jar.xml”.

Using weblogic-ejb-jar.xml, you can configure the following:

- ✿ Security role mappings
- ✿ Advanced security settings
- ✿ EJB clustering

- ↳ EJB pooling and caching
- ↳ Work managers and threading

70. What is an Enterprise Application?

- An enterprise application is a grouping of several resources into one deployable unit that is packaged in an .ear file.
- These resources include:
 - ✓ Web applications (.war)
 - ✓ EJB applications (.jar)
 - ✓ Java applications (.jar)
 - ✓ Resource adapters (.rar)

71. What is the user of Enterprise Applications?

Use enterprise applications to:

- ❖ Avoid namespace clashes
- ❖ Declare application wide security roles
- ❖ Deploy an application as one unit
- ❖ Share application wide EJB resources
- ❖ Configure local JDBC data sources
- ❖ Configure local JMS resources
- ❖ Configure local XML resources

72. What is the user of “weblogic-application.xml” deployment descriptor file?

Using weblogic-application.xml, you can configure:

- ↳ References to shared libraries
- ↳ Work managers and threading
- ↳ Default EJB and Web application parameter values

We can configure enterprise wide WLS-specific features with the weblogic-application.xml deployment descriptor:

- ↳ XML parsers
- ↳ XML entity mappings
- ↳ JDBC data sources
- ↳ JMS connection factories and destinations
- ↳ Security realms

73. What is the user of Weblogic shared java EE Libraries?

A Shared Java EE library:

- Is a reusable portion of a Web or enterprise application
- Is referenced by other deployed applications
- Avoids duplicating source files among Java EE projects
- Can contain deployment descriptors that are merged with the application's descriptors

74. Explain about deployment methods in Weblogic?

- WLS supports three deployment methods:

✓ Auto-deployment ↗

✓ Console deployment ↗ ✓

✓ Command-line deployment ↗

➤ You can deploy:

✓ Enterprise, Web, and EJB applications

✓ Web services

✓ J2EE libraries

✓ JDBC, JMS, and Diagnostic Framework modules

✓ Resource adapters

✓ Optional packages

✓ Client application archives

➤ Applications and EJBs can be deployed:

✓ In an archived file (.ear, .war, .jar)

✓ In an exploded (open) directory format

✓ Auto-deploy

75. How many ways we can deploy an application to Weblogic servers?

Several methods are available to deploy the Oracle WebLogic Server applications and shared libraries, including:

- Administration Console
- WebLogic Scripting Tool (WLST)
- weblogic.Deployer Java class
- wldeploy Ant task
- Auto-deployment folder

76. Explain about auto deployment in Weblogic?

If Production Mode is OFF:

- You can install an application simply by copying it (manually or using the console) to the "autodeploy" directory of the domain
- The Administration Server monitors this directory for new, changed, or removed applications
- This configures, targets, and deploys the application only to the Administration Server

↳ Location of Applications Directory:

↳ \$BEA_HOME/user_projects/domains/domain_name/autodeploy

77. Explain about FastSwap and On-Demand Deployment in Weblogic?

➤ WebLogic's FastSwap feature is:

- ✓ Enabled using the WebLogic deployment descriptors
- ✓ Available only if the domain is *not* running in production mode
- ✓ Applicable only to Web applications that are *not* archived

➤ When enabled:

- ✓ WebLogic automatically reloads the modified Java class files within applications
- ✓ Developers can perform iterative development without an explicit redeployment

➤ On-demand deployment:

✓ Excerpt from weblogic.xml:

<fast-swap>true</fast-swap>

78. While deploying an application to Weblogic, what is the difference between Development and Production Mode?

- An Administration Server starts using either:
 - ✓ The development mode, which turns auto-deployment on
 - ✓ The production mode, which turns auto-deployment off
- The Administration Server starts in the mode selected at domain creation time.
- The mode is set for all Oracle WebLogic Servers in a given domain.

79. Explain about console deployment method?

Deploying with the console allows full administrator control:

- Installation from a location of your choice
- Manual configuration of application name
- Targeting of application to individual servers and/or clusters
- Configuring the application without targeting it
- Activating deployment when desired

80. Explain about command line deployment?

- The weblogic.Deployer utility allows you to perform deployment operations similar to those available in the console.
- weblogic.Deployer actions can also be scripted with the Ant task wldeploy.

weblogic.Deployer Syntax:

% java weblogic.Deployer [options]
[-deploy|-undeploy|-redeploy|-start|-stop|-listapps] [file(s)]

Prepare and deploy a new application:

java weblogic.Deployer -adminurl t3://adminserver:7001
-username myuser -password mypass -name HRServices
-source /usr/HRServices.ear -targets serverA -deploy

Redeploy an application:

java weblogic.Deployer -adminurl t3://adminserver:7001
-username myuser -password mypass -name HRServices
-redeploy

Undeploy an application:

java weblogic.Deployer -adminurl t3://adminserver:7001
-username myuser -password mypass -name HRServices
-undeploy

List all applications:

java weblogic.Deployer -adminurl t3://adminserver:7001
-username myuser -password mypass -listapps

81. What is JNDI?

- The Java Naming and Directory Interface is an API for uniformly accessing the different naming and directory services.

- This is a major step forward because:
 - ✓ Different services use vastly different naming schemes
 - ✓ Java applications can now navigate seamlessly across databases, files, directories, objects, and networks

82. Why the JNDI required in Weblogic?

In Oracle WebLogic Server, JNDI serves as a repository and lookup service for J2EE objects, including:

- Enterprise JavaBeans (EJB) home stubs
- JDBC DataSources
- JMS connection factories, queues, and topics
- Remote Method Invocation (RMI) stubs



83. What is the use of Naming Services?

A naming service provides a method for mapping identifiers to entities or objects.

Term	Definition	Example
Binding	The association of an atomic name and an object	www.example.com is bound to 209.10.217.38.
Namespace	A set of unique names in a naming system	www.example.com/ products

84. Explain about Contexts and Sub-contexts in JNDI?

- Subcontexts are referenced through the dot delimiters (.).
- The subcontexts must be created before objects are placed into them.
- Typically when objects are bound to a JNDI tree, subcontexts are automatically created based on the JNDI name.

If the following context exists: com.oracle.examples

You cannot bind: com.oracle.examples.ejb.SomeObject

Without first creating: com.oracle.examples.ejb

85. How can I set deployment order for applications?

WebLogic Server 8.1 allows you to select the load order for applications. See the ApplicationMBean LoadOrder attribute in Application. WebLogic Server deploys server-level resources (first JDBC and then JMS) before deploying applications. Applications are deployed in this order: connectors, then EJBs, then web Applications. If the application is an EAR, the individual components are loaded in the order in which they are declared in the application.xml deployment descriptor.

86. Can I refresh static components of a deployed application without having to redeploy the entire application?

Yes. You can use weblogic.Deployer to specify a component and target a server, using the following syntax:

```
java weblogic.Deployer -adminurl http://admin:7001 -name appname -targets server1,server2 -deploy jsp/*.jsp
```

87. When should I use the **-nostage** option?

Set the staging mode to **-nostage** (using weblogic.Deployer or the Administration Console) if you don't want to copy deployment files but want to deploy an application from its present location. All target servers must be able to access the same set of deployment files.

88. When should I use the **external_stage** option?

Set **-external_stage** using weblogic.Deployer if you **want to stage** the application yourself, and prefer to copy it to its target by your own means.

89. What are the Deployment Tools for Developers?

WebLogic Server provides several tools for deploying applications and stand-alone modules:

- ◆ **wldeploy** is an Ant task version of the weblogic.Deployer utility. You can automate deployment tasks by placing wldeploy commands in an Ant build.xml file and running Ant to execute the commands.
- ◆ **weblogic.PlanGenerator** is a command-line tools that enables developers to export an application's configuration for deployment to multiple WebLogic Server environments.
- ◆ The deployment API allows you to perform deployment tasks programmatically using Java classes.
- ◆ The autodeploy domain directory allows you to deploy an application quickly for evaluation or testing in a development environment.

90. What is the Deployment order of Weblogic Server at Server Startup time?

By default, WebLogic Server deploys applications and resources in the following order:

- ❖ JDBC system modules
- ❖ JMS system modules
- ❖ J2EE Libraries and optional packages
- ❖ Applications and stand-alone modules
- ❖ Startup classes

Note: WebLogic Server security services are always initialized before server resources, applications, and startup classes are deployed. For this reason, you cannot configure custom security providers using startup classes, nor can custom security provider implementations rely on deployed server resources such as JDBC.

91. How will you “Kill the JVM” or running Weblogic server in different operating systems?

Each WebLogic Server instance runs in its own JVM. If you are unable to shut down a server instance using the scripts which are provided by the Weblogic (stopWeblogic.cmd/ stopManagedWeblogic.cmd), you can use an operating system command to kill the JVM.

Caution: If you kill the JVM, the server immediately stops all processing. Any session data is lost. If you kill the JVM for an Administration Server while the server is writing to the config.xml file, you can corrupt the config.xml file.

Some common ways to kill the JVM are as follows:

- ◆ If the shell (command prompt) in which you start the server is still open, you can type Ctrl-C.
- ◆ On a Windows computer, you can use the Task Manager to kill a JVM.
- ◆ On a UNIX computer, you can use the "ps" command to list all running processes. Then you can use the kill command to kill the JVM.

92. Can I generate deployment descriptor files automatically?

Yes, WebLogic Builder automatically generates deployment descriptor files for your J2EE applications. See WebLogic Builder Online Help.

93. Can I set the deployment order for application modules? For standalone modules?

The Load Order attribute controls the deployment order of standalone modules and applications relative to other modules and applications of the same type.

For example, standalone EJBs with smaller Load Order values are deployed before those with higher values.

Modules that are deployed as part of an Enterprise Application (EAR file directory) are deployed in the order in which they are specified in the application.xml deployment descriptor.

94. What is the difference between the WL_HOME/config/examples/applications folder and the WL_HOME/config/examples/stage folder?

The applications folder is intended for applications that are not yet ready for a production environment. WebLogic Server dynamically deploys the contents of the applications folder. The stage folder (or a folder that you create for the same purpose) is for storing copies of deployment files that are ready for deployment in a production environment (deployments that use the stage or external_stage deployment modes).

95. How do I turn the auto-deployment feature off?

The auto-deployment feature checks the applications folder every three seconds to determine whether there are any new applications or any changes to existing applications and then dynamically deploys these changes.

The auto-deployment feature is enabled for servers that run in development mode. To disable auto-deployment feature, use one of the following methods to place servers in production mode:

In the Administration Console, click the name of the domain in the left pane, then select the Production Mode checkbox in the right pane.

At the command line, include the following argument when starting the domain's Administration Server:

-Dweblogic.ProductionModeEnabled=true Production mode is set for all WebLogic Server instances in a given domain.

96. I downloaded the WebLogic Server installation file, but the installation program will not run. What should I do?

The installation file may have been corrupted during the download. Run a checksum on the installation file and check with technical support for the proper values.

97. Do I need to install WebLogic Server as root on Solaris?

No you don't need to be root, depending on directory permissions.

98. Can I run the Configuration Wizard outside the installer?

Yes. You can start the Configuration Wizard from the Start menu or using a script in the utils directory. See Creating Domains and Servers in Configuring and Managing WebLogic Server.

99. How do I edit the config.xml file?

The persistent configuration for a domain of WebLogic Servers and clusters is stored in an XML configuration file (config.xml). You can modify this file in the following ways:

- Use the Administration Console. See "Using the Administration Console" in the Administration Console Online Help.
- If you want to create scripts that automate domain management, use the weblogic.Admin utility. See "weblogic.Admin Command-Line Reference" in the .
- If you want to create Java-based management applications, use the Java Management Extensions (JMX) Application Programming Interface (API). See the Programming WebLogic Management Services with JMX guide.
- If you want to edit the config.xml file directly (not recommended), see the BEA WebLogic Server Configuration Reference.

100. What is the free pool?

The free pool is a data structure the EJB container uses to cache anonymous instances of a given bean type. The free pool improves performance by reusing objects and skipping container callbacks when it can.

101. Can I use the PointBase DBMS included with WebLogic Server for development or production?

PointBase Server is an all-Java DBMS product included in the WebLogic Server distribution solely in support of WebLogic Server evaluation, either in the form of custom trial applications or through packaged sample applications provided with WebLogic Server. Non-

evaluation development and/or production use of the PointBase Server requires a separate license be obtained by the end user directly from PointBase.

102. How can I enable Oracle Advanced Security encryption on the JDBC Oracle Thin driver with a WebLogic JDBC Connection Pool?

Oracle Advanced Security encryption relies on features available through connection properties in the JDBC driver from Oracle. You can specify connection properties in a WebLogic JDBC connection pool in the Properties attribute. This attribute is available on the JDBC Connection Pool → Configuration → General tab in the Administration Console. When WebLogic Server creates database connections for the connection pool, it passes the properties to the JDBC driver so that connections are created with the specified properties.

For example, to enable Oracle Advanced Security encryption, you may want to specify the following options:

Properties: user=SCOTT
oracle.net.encryption_client=ACCEPTED
oracle.net.encryption_types_client=RC4_256
protocol=thin
oracle.net.crypto_checksum_client=ACCEPTED

Note: See the Oracle documentation for details about required properties for Oracle Advanced Security encryption. Properties listed above are for illustration only.

The resulting entry in the config.xml file would look like:

```
<JDBCConnectionPool  
DriverName="oracle.jdbc.driver.OracleDriver"  
Name="oraclePool"  
Password="{3DES}1eNn7kYGZVw="  
Properties="user=SCOTT;  
oracle.net.encryption_client=ACCEPTED;  
oracle.net.encryption_types_client=RC4_256; oracle.net.crypto_checksum_client=ACCEPTED;  
protocol=thin"  
URL="jdbc:oracle:thin:@server:port:sid"/>
```

Note: Line breaks added for readability

103. When should I use a TxDataSource instead of a DataSource?

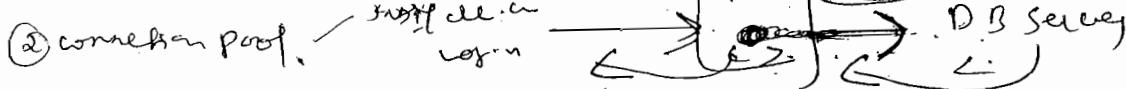
When you select Honor Global Transactions in the Administration Console, you create a JDBCDataSource in the config.xml file (the default). If you clear the Honor Global Transactions check box, you create a JDBCDataSource in the config.xml file. See "[When to Enable Global Transactions in a Data Source](#)" in the Administration Console Online Help.

104. Can I enable requests to a JDBC connection pool for a database connection to wait until a connection is available?

Yes. You can set two JDBC connection pool properties to enable connection requests to wait for a connection:

- ↳ ConnectionReserveTimeoutSeconds
- ↳ HighestNumWaiters

105. What happens when my database is restarted or becomes unreachable? Does my connection pool stick around?



Yes. The pool is independent of its ability to communicate with the DBMS. All connections in the connection pool may become defunct, but the connection pool still exists. You can configure the connection pool so that WebLogic Server tests the connections in the pool and replaces bad connections when it can.

To manually restart the connection pool using the Administration Console after a database failure, you can undeploy the connection pool by removing all of its deployment targets, and then redeploy the connection pool by adding deployment targets.

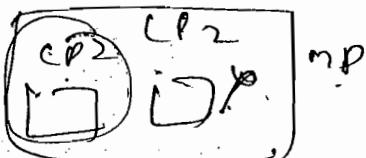
To do this from the command line using "weblogic.Admin", set the "Targets" attribute of the pool to an empty string ("") and then set it to the desired set of targets.

106. When should I use MultiPools?

You can use MultiPools in one of two ways

- 1) For high availability in the event a database connection fails, or
- 2) For load balancing between JDBC connection pools. Because you can choose only one option, you need to determine the primary purpose of your MultiPool.

Note: If you implement Multipools for a JDBC application, do not configure driver-level load balancing or failover for the connection pools used by the MultiPool—the MultiPool provides the same functionality as configuring driver-level load balancing or failover.



107. What is JDBC?

(JDBC is an API for accessing databases in a uniform way.)

JDBC provides:

- ❖ Platform-independent access to databases
- ❖ Location transparency
- ❖ Transparency to proprietary database issues
- ❖ Support for both two-tier and multitier models for database access

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 WLS or your database vendor.

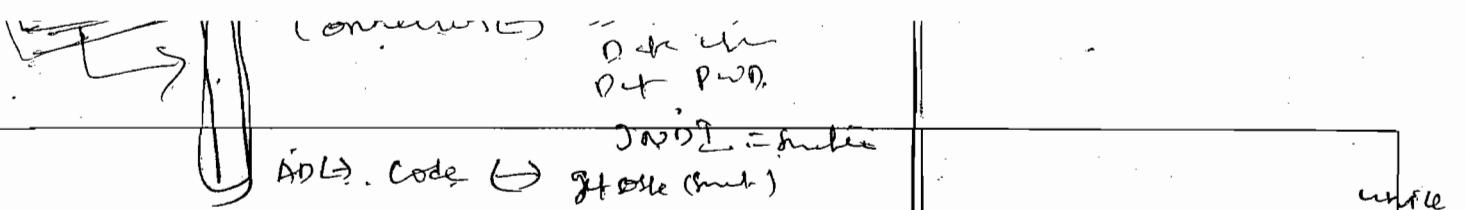
108. What is the use of Data Source in Weblogic?

Data sources:

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

109. What is the scope of the Data Source in Weblogic server?

- 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 EE enterprise applications
 - ✓ Accessible only by the containing application

unfile
Data source (impl.) is the main
identity for DBMS Connec-

110. List default Weblogic provided JDBC Drivers?

- Oracle and third-party drivers are included in 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 server's classpath.

111. What is a Data Source?

- A data source object provides a way for a JDBC client to obtain a database connection from a connection pool.
- A data source:
 - ✓ Is stored in the Oracle WebLogic Server JNDI tree
 - ✓ Can support transactions
 - ✓ Is associated with a connection pool

112. 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 load

113. What are the benefits of having Data Sources and Connection Pools in Weblogic?

- The following are some advantages of this approach:
 - ✓ Time and overhead are saved by using an existing database connection.
 - ✓ Connection information is managed in one location in the Administration Console.
 - ✓ 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

114. How Data Sources are used in Weblogic?

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

115. What is Multi Data Sources?

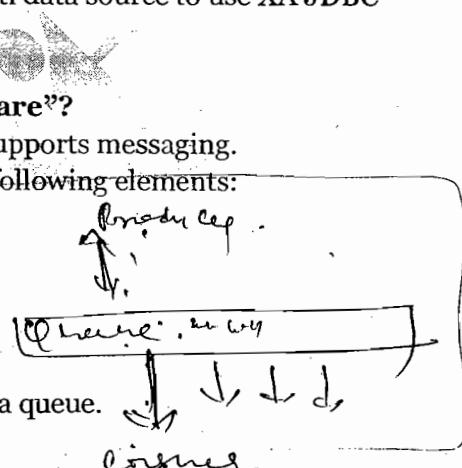
- Multi data source:
 - ✓ Is an abstraction around a group of data sources
 - ✓ Determines which data source to use to satisfy the request depending on the algorithm selected in the multi data source configuration:
 1. Load balancing or failover
 - ✓ Is bound to the JNDI tree
- XA support for multi data sources:
 - ✓ The WLS JDEC supports using multi data sources in XA transactions.
 - ✓ You can configure the data sources contained within the multi data source to use XA JDBC drivers.

116. What is the functionality of "Message-Oriented Middleware"?

- 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

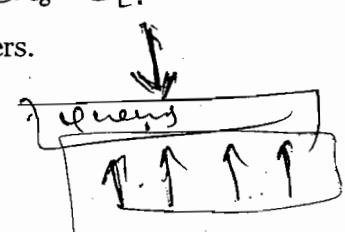
117. Explain about Point-to-Point (Queue)?

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



118. Explain about Publish-Subscribe (Topics)?

Publishing and subscribing to a topic decouples producers from consumers.



119. Explain about Oracle WebLogic Server JMS Features?

Oracle WebLogic Server JMS supports:

- PTP and pub/sub domains
- Guaranteed and transactional message delivery
- Durable subscribers
- Distributed destinations
- Recovery from failed servers

120. How will you implement "Messaging Service" in Weblogic?

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

121. What is the user of "Connection Factories" in JMS?

- 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

122. What is the use of "Destination" in JMS? How many types of "Destinations" are available in JMS?

- A destination is a lightweight object that is stored in JNDI.
- It is the target on a JMS server for sending or receiving messages.
- The JMS destination types are:
 - ✓ Queue
 - ✓ Topic

123. Explain about "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.

124. Explain about "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").

125. Explain about Threshold and Quotas in JMS?

- A threshold and a quota can be set for the server and destination objects.
- 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
- A threshold is a limit that triggers message paging, flow control, and logged warnings, using:
 - ✓ Upper and lower values for the number of bytes
 - ✓ Upper and lower values for the number of messages

126. Difference between "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 configure:
 - ✓ Where the messages are persisted
 - ✓ Persistent connection factories and destinations

127. What is "Persistent Messaging"? When to Use it?

- Persistent messaging permits messages in memory to be written out to a persistent store.
- Configure persistent messaging if:

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

128. How will you Configuring a Durable Subscription in Weblogic?

- 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:
 - ✓ A file store
 - ✓ A JDBC store (a connection pool)

129. How a Durable Subscription Works?

- If a subscriber client is active, messages are delivered normally.
- When the client becomes active again, its ID is used to retrieve and redeliver messages.

130. Does the WebLogic JMS server find out about closed or lost connections, crashes, and other problems and does it recover from them?

Yes, but how it does this depends on whether a Java client crashes or WebLogic Server crashes, as follows:

- If a Java client crashes then the JMS server will clean up all the outstanding server-side resource from the crashed client JVM, such as:
 - ✓ JMS connection(s) from the crashed client JVM
 - ✓ JMS temporary destination(s) created under the above JMS connection(s)
 - ✓ JMS session(s) created under the above JMS connection(s)
 - ✓ JMS client(s) created under the above JMS session(s) (connection consumer and regular consumer)
 - ✓ JMS browser(s) created under the above session(s)
 - ✓ JMS producer(s) created under the above session(s)
- If WebLogic Server crashes and it is the front-end to the JMS server, then:
 - ✓ A JMS client will lose all the server-side resources listed above.
 - ✓ The client's `javax.jms.ExceptionListener.onException(...)` will be called (if `javax.jms.JMSConnection.setExceptionListener` is set) with a `LostServerException`, which extends `JMSEException`.
- If WebLogic server crashes and it is a back-end to the JMS server, then:
 - ✓ A JMS client may partially lose some of the server-side resources listed above (only the resource on the crashed server, such as JMS temporary destination(s), JMS client(s) and JMS browser(s)).
 - ✓ The client's `javax.jms.ExceptionListener.onException(...)` will be called (if `weblogic.jms.extensions.WLSession.setExceptionListener` is set) with a `ConsumerClosedException`, which extends `JMSEException`.

131. What Is the Java Message Service?

An enterprise messaging system enables applications to 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, allowing you to separate the details about the destination system from the application code.

The Java Message Service (JMS) is a standard API for accessing enterprise messaging systems. Specifically, JMS:

- Enables Java applications sharing a messaging system to exchange messages
- Simplifies application development by providing a standard interface for creating, sending, and receiving messages

132. How many Messaging Modules are available in Weblogic?

JMS supports two messaging models: point-to-point (PTP) and publish/subscribe (pub/sub).

The messaging models are very similar, except for the following differences:

- PTP messaging model enables the delivery of a message to exactly one recipient.
- Pub/sub messaging model enables the delivery of a message to multiple recipients.

133. Explain about Point-to-Point Messaging?

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.

134. Explain about Publish/Subscribe Messaging?

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.

135. Explain about Message Persistence?

As per the "Message Delivery Mode" section of the JMS Specification, messages can be specified as persistent or non-persistent:

- A **persistent message** is guaranteed to be delivered once-and-only-once. The message cannot be lost due to a JMS provider failure and it must not be delivered twice. It is not considered sent until it has been safely written to a file or database. WebLogic JMS writes persistent messages to a WebLogic persistent store (disk-base file or JDBC-accessible database) that is optionally targeted by each JMS server during configuration.
- **Non-persistent messages** are not stored. They are guaranteed to be delivered at-most-once, unless there is a JMS provider failure, in which case messages may be lost, and must not be delivered twice. If a connection is closed or recovered, all non-persistent messages that have not yet been acknowledged will be redelivered. Once a non-persistent message is acknowledged, it will not be redelivered.

136. Topics vs. Queues?

Surprisingly, when you are starting to design your application, it is not always immediately obvious whether it would be better to use a Topic or Queue. In general, you should choose a Topic only if one of the following conditions applies:

- The same message must be replicated to multiple consumers.

- A message should be dropped if there are no active consumers that would select it.
- There are many subscribers, each with a unique selector.

It is interesting to note that a topic with a single durable subscriber is semantically similar to a queue. The differences are as follows:

- If you change a topic selector for a durable subscriber, all previous messages in the subscription are deleted, while if you change a queue selector for consumer, no messages in the queue are deleted.
- A queue may have multiple consumers, and will distribute its messages in a round-robin fashion, whereas a topic subscriber is limited to only one consumer.

137. Asynchronous vs. Synchronous Consumers?

In general, asynchronous (onMessage) consumers perform and scale better than synchronous consumers:

- Asynchronous consumers create less network traffic. Messages are pushed unidirectionally, and are pipelined to the message listener. Pipelining supports the aggregation of multiple messages into a single network call.
- Note:** In WebLogic Server, your synchronous consumers can also use the same efficient behavior as asynchronous consumers by enabling the Prefetch Mode for Synchronous Consumers option on JMS connection factories
- Asynchronous consumers use fewer threads. An asynchronous consumer does not use a thread while it is inactive. A synchronous consumer consumes a thread for the duration of its receive call. As a result, a thread can remain idle for long periods, especially if the call specifies a blocking timeout.
- For application code that runs on a server, it is almost always best to use asynchronous consumers, typically via MDBs. The use of asynchronous consumers prevents the application code from doing a blocking operation on the server. A blocking operation, in turn, idles a server-side thread; it can even cause deadlocks. Deadlocks occur when blocking operations consume all threads. When no threads remain to handle the operations required unblocking the blocking operation itself, that operation never stops blocking.

138. What is a Distributed Destination?

A distributed destination is a set of destinations (queues or topics) that are accessible as a single, logical destination to a client. A distributed destination has the following characteristics:

- It is referenced by its own JNDI name.
- Members of the set are usually distributed across multiple servers within a cluster, with each destination member belonging to a separate JMS server.

139. Why Use a Distributed Destination?

Applications that use distributed destinations are more highly available than applications that use simple destinations because WebLogic JMS provides load balancing and failover for member destinations of a distributed destination within a cluster. Once properly configured, your producers and consumers are able to send and receive messages through the

distributed destination. WebLogic JMS then balances the messaging load across all available members of the distributed destination. When one member becomes unavailable due a server failure, traffic is then redirected toward other available destination members in the set.

140. How many Types of Distributed Destinations are available?

- Uniform Distributed Destinations
- Weighted Distributed Destinations

Uniform Distributed Destinations

In a uniform distributed destination (UDD), each of the member destinations has a consistent configuration of all distributed destination parameters, particularly in regards to weighting, security, persistence, paging, and quotas.

Oracle recommends using UDDs because you no longer need to create or designate destination members, but instead rely on WebLogic Server to uniformly create the necessary members on the JMS servers to which a UDD is targeted. This feature of UDDs provides dynamic updating of a UDD when a new member is added or a member is removed.

Weighted Distributed Destinations

In a weighted distributed destination, the member destinations do not have a consistent configuration of all distributed destination parameters, particularly in regards to weighting, security, persistence, paging, and quotas.

Oracle recommends converting weighted distributed destinations to UDDs because of the administrative inflexibility when creating members that are intended to carry extra message load or have extra capacity (more weight). Lack of a consistent member configuration can lead to unforeseen administrative and application problems because the weighted distributed destination cannot be deployed consistently across a cluster.

141. What is JNDI?

JNDI (Java Naming and Directory Interface) is a Java EE lookup service that maps names to services and resources. JNDI provides a directory of advertised resources that exist on a particular stand-alone (unclustered) WebLogic server, or within a WebLogic server cluster. Examples of such resources include JMS connection factories, JMS destinations, JDBC (database) data sources, and application EJBs.

A client connecting to any WebLogic server in a WebLogic cluster can transparently reference any JNDI advertised service or resource hosted on any WebLogic server within the cluster. The client doesn't require explicit knowledge of which particular WebLogic server in the cluster hosts a desired resource.

142. What is a JMS connection factory?

A JMS connection factory is a named entity resource stored in JNDI. Applications, message driven beans (MDBs), and messaging bridges lookup a JMS connection factory in JNDI and use it to create JMS connections. JMS connections are used in turn to create JMS sessions, producers, and consumers that can send or receive messages.

143. What is a JMS connection-id?

JMS connection-ids are used to name JMS client connections. Durable subscribers require named connections, otherwise connections are typically unnamed. Note that within a clustered set of servers or stand-alone server, only one JMS client connection may use a particular named connection at a time. An attempt to create new connection with the same name as an existing connection will fail.

144. What is the difference between a JMS topic and a JMS queue?

JMS queues deliver a message to one consumer, while JMS topics deliver a copy of each message to each consumer.

145. What is a topic subscription?

A topic subscription can be thought of as an internal queue of messages waiting to be delivered to a particular subscriber. This internal queue accumulates copies of each message published to the topic after the subscription was created. Conversely, it does not accumulate messages that were sent before the subscription was created. Subscriptions are not sharable, only one subscriber may subscribe to a particular subscription at a time.

146. What is a non-durable topic subscriber?

A non-durable subscriber creates unnamed subscriptions that exist only for the life of the JMS client. Messages in a non-durable subscription are never persisted—even when the message's publisher specifies a persistent quality of service (QOS). Shutting down a JMS server terminates all non-durable subscriptions.

147. What is a durable subscriber?

A durable subscriber creates named subscriptions that continue to exist even after the durable subscriber exits or the server reboots. A durable subscriber connects to its subscription by specifying topic-name, connection-id, and subscriber-id. Together, the connection-id and subscriber-id uniquely name the subscriber's subscription within a cluster. A copy of each persistent message published to a topic is persisted to each of the topic's durable subscriptions. In the event of a server crash and restart, durable subscriptions and their unconsumed persistent messages are recovered.

148. What is the WebLogic Store-and-Forward Service?

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

149. When should I use the WebLogic Store-and-Forward Service?

The WebLogic Store-and-Forward (SAF) Service should be used when forwarding JMS messages between WebLogic Server 9.0 or later domains. The SAF service can deliver messages:

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

150. When can't I use WebLogic Store-and-Forward?

You can't use the WebLogic Store-and-Forward service in the following situations:

- ✗ Receiving from a remote destination—use a message driven EJB or implement a client consumer directly.
- ✗ Sending messages to a local destination—send directly to the local destination.
- ✗ Forwarding messages to prior releases of WebLogic Server. See Q. What is a messaging bridge?.
- ✗ Interoperating with third-party JMS products (for example, MQSeries). See Q. What is a messaging bridge?.
- ✗ When using temporary destinations with the JMSReplyTo field to return a response to a request.
- ✗ Environment with low tolerance for message latency. SAF increases latency and may lower throughput.

151. What is a messaging bridge?

Messaging bridges are administratively configured services that run on a WebLogic server. They automatically forward messages from a configured source JMS destination to a configured target JMS destination. These destinations can be on different servers than the bridge and can even be foreign (non-WebLogic) destinations. Each bridge destination is configured using the four common properties of a remote provider:

- ✗ The initial context factory.
- ✗ The connection URL.
- ✗ The connection factory JNDI name.
- ✗ The destination JNDI name.

Messaging bridges can be configured to use transactions to ensure exactly-once message forwarding from any XA capable (global transaction capable) JMS provider to another.

152. When should I use a messaging bridge?

Typically, messaging bridges are used to provide store-and-forward high availability design requirements. A messaging bridge is configured to consume from a sender's local destination and forward it to the sender's actual target remote destination. This provides high availability because the sender is still able to send messages to its local destination even when the target remote destination is unreachable. When a remote destination is not reachable, the local destination automatically begins to store messages until the bridge is able to forward them to the target destination when the target becomes available again.

153. When should I avoid using a messaging bridge?

Other methods are preferred in the following situations:

- ◆ Receiving from a remote destination—use a message driven EJB or implement a client consumer directly.
- ◆ Sending messages to a local destination—send directly to the local destination.
- ◆ Environment with low tolerance for message latency. Messaging Bridges increase latency and may lower throughput. Messaging bridges increase latency for messages as they introduce an extra destination in the message path and may lower throughput because they forward messages using a single thread.
- ◆ Forward messages between WebLogic 9.0 domains—Use WebLogic Store-and-Forward.

154. How many types of JMS stores are available?

- The JMS store can be configured to use either:
 - ✓ A file store
 - ✓ A JDBC store (a connection pool)

155. How will you configure a JMS JDBC Store?

- To configure JMS JDBC persistence, perform the following:
 - ✓ Create a JDBC DataSource.
 - ✓ Create a JMS 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.

156. What Is a Transaction?

- A transaction is a mechanism to handle groups of operations as though they were one.
- Either all operations in a transaction occur or none occur at all.
- The operations involved in a transaction might rely on multiple servers and databases.

157. How many Types of Transactions are there? Explain?

- A local transaction deals with a single resource manager. Local transactions use the non-Extended Architecture (non-XA) interface between Oracle WebLogic Server and the resource manager.
- A distributed transaction coordinates or spans multiple resource managers.
- Global transactions can deal with multiple resource managers. Global transactions use the Extended Architecture (XA) interface between Oracle WebLogic Server and the resource managers.
- ✓ You need to create non-XA or XA resources for local transactions. However, for global transactions, you need to create only XA resources.

158. Explain about Two-Phase Commit Protocol?

- The Two-Phase Commit (2PC) protocol uses two steps to commit changes within a distributed transaction.
 - ✓ Phase 1 asks the RMs to prepare to make the changes.

- ✓ Phase 2 asks the RMs to commit and make the changes permanent or to roll back the entire transaction.
- A global transaction ID (XID) is used to track all the changes associated with a distributed transaction.

159. Explain about Extended Architecture Protocol (XA)?

The Extended Architecture (XA) protocol:

- Is the interface that is used between WLS and the RMs
- Implements the 2PC protocol
- Allows programs to control the RMs that are involved in distributed transactions

160. Explain the relationship between Transaction and Resource Managers?

- A transaction manager coordinates multiple resource managers.
- The 2PC protocol is used to coordinate the transaction.
- The XA protocol implements 2PC.

161. What is the user of “Transaction Log”?

- Each server has a transaction log that stores information about committed transactions coordinated by the server that may not have been completed.
- ✓ Oracle WebLogic Server uses the transaction log when recovering from system crashes or network failures.
 - You cannot directly view the transaction log because the records are in a binary format and are stored in the default persistent store for the server.

162. Explain about Logging Last Resource?

- You can configure a JDBC data source to enable the Logging Last Resource (LLR) transaction optimization, which:
 - ✓ Enables one non-XA resource to participate in a global transaction
 - ✓ Has improved performance and the same ACID guarantee as XA
- The LLR optimization improves performance by:
 - ✓ Removing the need for an XA JDBC driver to connect to the database. XA JDBC drivers are typically inefficient compared to non-XA JDBC drivers.
 - ✓ Reducing the number of processing steps to complete the transaction, which also reduces network traffic and I/O
 - ✓ Removing the need for XA processing at the database level (if the database is the one non-XA resource)

163. What Is LDAP?

The Lightweight Directory Access Protocol:

- Is derived from X.500
- Provides a hierarchical lookup service
- Supports sophisticated searching
- Can be secured via SSL

164. Explain Security Realms?

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

165. Explain about Embedded LDAP Server?

- In WLS, users, groups, and authorization information is stored in an embedded LDAP server.
- Several properties can be set to manage the LDAP server, including:
 - ✓ Credentials
 - ✓ Backup settings
 - ✓ Cache settings
 - ✓ Replication settings

166. 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

167. Explain about Trust and Identity in SSL?

- 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

168. How will you access SSL enabled applications?

- 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>

169. What are the Oracle WebLogic Server SSL Requirements?

- To enable Oracle WebLogic Server SSL, you must:
 - ✓ Obtain an appropriate digital certificate
 - ✓ Install the certificate
 - ✓ Configure SSL properties

- ✓ Configure two-way authentication (if desired)
 1. SSL impacts performance.

170. What is the user of keytool Utility in WLS?

- 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 the display certificate and keystore contents.

171. How WLS Protecting Against Attacks? What kinds of Attacks are there?

- WLS can help protect applications against several attacks:
 - ✓ Man-in-the-middle attacks
 - ✓ Denial of service (DoS) attacks
 - ✓ Large buffer attacks
 - ✓ Connection starvation attacks
- The slides that follow detail the countermeasures that WLS provides to address these attacks.

172. Explain about 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.
- The “man-in-the-middle” attacks can be resisted by using a Host Name Verifier.
- A Host Name Verifier validates that the host to which an SSL connection is made is the intended or authorized party.
- WLS provides a Host Name Verifier by default.
- A custom Host Name Verifier can be created by implementing the interface `weblogic.security.SSL.HostnameVerifier`

173. Explain about Denial of Service Attacks (DOS)?

- 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

Harden WLS against “denial of service” attacks by:

- Filtering incoming network connections
- Configuring consumable WLS resources with the appropriate threshold and quotas
- Limiting access to configuration information and backing up configuration files
- Preventing unauthorized access by protecting passwords against password-guessing attacks

174. Explain about 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
 - ✓ Restrict the 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.

175. Explain about Large Buffer Attacks?

- Individuals can try to 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.

176. Explain about Connection Starvation?

- Individuals can try to 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.

177. Explain about 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.

178. How will you 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).

179. What Is a Deployment Plan?

- It is an optional XML document that resides outside an application archive.
 - It configures an application for deployment to a specific WLS environment.
 - It is created and owned by administrators or developers for a particular environment.
- A JavaEE deployment plan:
- Is an XML file that is associated with an application
 - Resides outside an application archive
 - Sets or overrides the values in the JavaEE deployment descriptors
 - Allows a single application to be easily customized to multiple deployment environments

180. What are the advantages of Deployment Plan?

- Works by setting or overriding the deployment property values that are defined in an application's WLS deployment descriptor

- Helps to easily modify an application's WLS configuration for deployment into different multiple WLS environments without modifying the deployment descriptor files that are included in the application archive
- Enables an application to be deployed to multiple domains or to multiple target servers and clusters that have a different configuration within the same domain

181. How will you create a Deployment Plan?

- Tools for creating a deployment plan:
 - ✓ weblogic.PlanGenerator
 - ✓ Administration Console
- Goals for creating a deployment plan:
 - ✓ To expose the external resource requirements of the application as variables in the deployment plan
 - ✓ To expose additional configurable properties, such as tuning parameters as variables in the deployment plan
 - WLS includes tools to accelerate deployment plan creation.
 - The Administration Console:
 - ✓ Generates a skeleton plan.xml if a plan folder is detected with a newly deployed application
 - ✓ Updates plan.xml when you use the console to modify the deployment descriptor settings
 - The weblogic.PlanGenerator Java class can also generate a skeleton plan.xml for an existing application.

182. What is the user of weblogic.PlanGenerator?

- Is a Java-based deployment configuration tool
- Is primarily intended for developers who want to export portions of an Oracle WebLogic Server deployment configuration into an XML deployment plan
- Enables you to generate a basic Oracle WebLogic Server configuration for applications that have only J2EE deployment descriptors

183. How will you generate a Deployment Plan Using the Administration Console?

- The Administration Console automatically generates or updates the deployment plan.
- You can generate a deployment plan using the Administration Console using the following steps:
 - ✓ Prepare the deployment files.
 - ✓ Install the application archive.
 - ✓ Save the configuration changes to a deployment plan.

184. How many types of Staged Deployment are available in WLS?

You can configure deployment per server or for each application as:

- **staged (default):** Files are copied to the preconfigured staging directory for preparation and activation.
- **nostage:** Files are deployed from a static location.
- **external_stage:** Files are copied by a user or a third-party tool before deployment.

185. Explain about Production Redeployment?

- You can redeploy a revised version of a production application alongside the older version:
 - ✓ Without affecting the existing clients to the application
 - ✓ Without interrupting the availability of the application to the new client request
- Oracle WebLogic Server automatically manages client connections so that:
 - ✓ Existing clients continue to use the older application
 - ✓ New client requests are directed to the newer application
 - The older version is undeployed after all current clients complete their work.

186. Explain about Application Availability, after deploying/install to WLS?

- By default, when an application is redeployed:
 - ✓ It is unavailable to clients for a brief time
 - ✓ Existing clients lose any conversational state
- Some types of applications require availability 24 hours a day, seven days a week.
- Third-party proxy solutions are possible, but they require multiple servers.

187. Explain about WebLogic Production Redeployment?

- Production redeployment:
 - ✓ Allows two versions of a single Web application or module to run simultaneously
 - ✓ Requires you to include unique version information either:
 1. Within the application's META-INF/MANIFEST.MF file
 2. As part of the deployment process
- When a new version is redeployed, WLS automatically:
 - ✓ Routes existing clients to the prior (retired) version
 - ✓ Routes new clients to the new version
 - ✓ Undeploys the prior version when all existing clients finish their work or their conversations time out

188. What are the Advantages of Production Redeployment?

Saves the trouble of:

- Scheduling application downtime
- Setting up redundant servers to host new application versions
- Managing client access to multiple application versions manually
- Retiring older versions of an application manually

189. What are the Requirements and Restrictions for Production Redeployment in WLS?

- Production redeployment is supported for:
 - ✓ Stand-alone Web Application (WAR) modules and Enterprise Applications (EARs) whose client accesses the application via a Web application (HTTP)
 - ✓ Enterprise Applications that are accessed by inbound JMS messages from a global JMS destination, or from inbound JCA requests
 - ✓ All types of Web Services, including conversational and reliable Web Services
- Production redeployment is not supported for:
 - ✓ Stand-alone EJB or Resource Archive (RAR) modules

- ✓ Applications that use JTS drivers
- ✓ Applications that obtain JDBC data sources via the DriverManager API instead of using the JNDI lookup
- ✓ Applications that include the EJB 1.1 container-managed persistence (CMP) EJBs
- A deployed application must specify a version number.
- WLS can host a maximum of two different versions of an application at one time.
- When you are redeploying a new version of an application, the following features cannot change:
 - ✓ Deployment targets
 - ✓ Security model
 - ✓ Persistent store settings

190. Redeploying Versus Distributing an applications in WLS?

- Distributing is an alternative to deploying an application.
- ✓ Distributing an application prepares it for deployment by copying its files to all target servers and validating the files.
- ✓ You can start a distributed application in Administration mode. Access to the application is then restricted to a configured Administration channel.
- Distributing a new version of the application makes it available for testing before being released for general consumption.
- Redeploying a new version of an application places the application immediately into use and makes it available to new client requests.

191. How will you Distributing a New Application Version in WLS?

- Use the weblogic.Deployer -distribute command.
- After the application is distributed, start the application in Administration mode.
- Test the application.
- When ready, start the application (without using -adminmode).
- Optionally set a retirement timeout for the older version of the application.

192. How will you Create MANIFEST.mf file for Production Redeployment?

- Create MANIFEST.mf with the following contents:

Name: java/util/
Specification-Title: Java Utility Classes
Specification-Version: 1.2
Specification-Vendor: Sun Microsystems, Inc.
Implementation-Title: java.util
Implementation-Version: build57
Implementation-Vendor: Sun Microsystems, Inc.
- Then add an entry for WLS versioning:

Weblogic-Application-Version: Version1.0Beta

193. What is a cluster Definition?

- A cluster is a group of Oracle WebLogic Server instances that work in coordination.
- Clustering provides:
 - ✓ High availability
 - ✓ Load balancing

✓ Scalability

194. What Is a Cluster?

A cluster:

- Is a logical group of Oracle WebLogic Servers within a domain
- Supports features to provide high availability for:
 - Whole servers
 - Web applications/services
 - EJB applications
 - JMS
- Is transparent to a client

195. What are the Benefits of Clustering?

- There are two main benefits of clustering together Oracle WebLogic Servers:
 - ✓ Scalability
 - ✓ High availability
- Scalability is the ability to provide more capacity for an application, in this case, by adding additional servers without having to make major architectural changes.
- High availability ensures that when a server (in a cluster) fails, there are other servers to take over the work, so that the client is not affected.

Concept	Description
Scalability	Provides more capacity for an application by adding servers, without having to make major architectural changes
Load balancing	Distributes work (client requests and so on) across the members of a cluster
Failover	After a system failure on one server, automatically continues ongoing work on another server
Migration	After a system failure on one server, continues ongoing work by moving the component to another server

196. What are the Key Capabilities of clustering?

The key capabilities of a WebLogic cluster are:

- Application failover
 - When an object in an application that is performing a task becomes unavailable, another object takes over and finishes the job.
- Site failover
 - When all the services and applications in a single site fail, they can switch to a separate site and continue processing.

- Server migration
 - When a server fails, pinned services can be migrated to another server in a cluster.
- Load balancing
 - Tasks and communications are evenly distributed across multiple servers.

197. Explain about Cluster Architecture? ✓

- Applications are generally broken into multiple tiers, each representing its 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 but load-balancing and failover is a little different.

198. How the Server will communicate in a Cluster? ✓

- The Oracle WebLogic Server instances in a cluster communicate with one another using two different techniques:
 - ✓ Unicast/multicast (UDP)
 - ✓ Sockets (peer-to-peer TCP)
- The server instances use IP unicast or multicast to broadcast the availability of services and heartbeats that indicate continued availability.
- IP multicast broadcasts one-to-many communications among clustered instances.
- IP 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.
- IP sockets are used for peer-to-peer communications between server instances.

199. Explain about Cluster Communication in WLS?

- Members of a cluster cooperate to achieve high availability using the following:
 - ✓ Broadcast messages such as "heartbeats"
 - ✓ Peer-to-peer IP sockets
- You can configure broadcast communication to use either:
 - ✓ IP unicast
 - ✓ A dedicated IP multicast address (224.0.0.0 through 239.255.255.255) and port
- If heartbeats are not received from a cluster member, the server is marked as "failed" and its services are not used.

200. How the Weblogic Server detect a server Failure?

- WebLogic clusters detect the failure of a server instance in the following ways:
 - ✓ Through the use of IP sockets
 - ✓ Through the Oracle WebLogic Server heartbeat
- If a server in the cluster unexpectedly closes its socket, it is marked as "failed" and its services are not used.

- Server instances use multicast to broadcast heartbeats every 10 seconds to the other server instances in the cluster.
 - ✓ If three heartbeats are missed from a peer server, the server is marked as "failed" and its services are not used.

201. Explain about One-to-Many Communications in WLS?

- 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.

202. Explain about Peer-to-Peer Communications in WLS?

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

- Accessing the nonclustered 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)

203. Explain about Multitier Communications in WLS?

- Multitier clusters require more IP sockets than a combined-tier cluster:
 - ✓ One socket for replicating session states
 - ✓ One socket for each Oracle WebLogic Server in the EJB cluster, for accessing remote objects
- As an example, using a three-node cluster, the worst-case scenario would be five open sockets per server:
 - ✓ One primary and secondary replicated session
 - ✓ Each server simultaneously invokes a remote EJB method on each node in the cluster.

204. How many cluster Configuration Options are available in WLS?

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

- Administration Console
- Configuration Wizard
- WebLogic Scripting Tool (WLST)
- Using the Cluster MBean

205. Explain about Packaging Applications?

- When you deploy applications to a single Managed Server, you can deploy the applications in an exploded format.
- Oracle recommends deploying packaged applications to a cluster of Managed Servers as .war, .ear, or .jar file

206. Explain about Two-Phase Deployment?

- Applications are deployed using two-phase deployment (TPD).
- Applications are copied to the cluster and activated in two phases:
 - ✓ 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.

207. How will you Deploy Applications to a Cluster?

- All nodes must be running before an application is deployed to a cluster.
- If phase 2 of the two-phase deployment fails, the application is still deployed to other nodes in the cluster.
- WebLogic allows partial deployment of applications to a partitioned server.
- Session replication for deployed applications may fail if a node is partitioned at the time of deployment.
 - ✓ Avoid this by using the enforceClusterConstraints tag with weblogic.Deployer.
 - ✓ Or select the Enable Cluster Constraints check box in the console.
- Do not change cluster membership while deploying applications to the cluster.

208. Explain about 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.
- Similar options are available for stateful EJB applications.

209. Explain about HTTP Session State Replication in WLS?

- Oracle WebLogic Server provides clustering support for JSPs and servlets by replicating the HTTP session state.
- To benefit from HTTP session state clustering, you must ensure that the session state is persistent, by configuring:
 - ✓ In-memory replication
 - ✓ JDBC replication
 - ✓ File system replication
- You must also access the cluster via a collection of Web servers with identically configured proxy plug-ins or load-balancing hardware.
- 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.

210. Explain about Replication Groups in WLS?

- 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

211. What is the user of Replication Groups in WLS?

- Replication groups:
 - ✓ Represent a subset of servers within a cluster
 - ✓ Help to determine the placement of secondary sessions (avoid replicating within the same room, for example)
 - ✓ 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

212. Explain about In-Memory Replication in WLS?

- 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)
 - ✓ Asynchronously (batch)
- 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.

213. What are all the Requirements for In-Memory Replication in WLS?

- 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 either:
 - ✓ The load-balancing hardware (WLS aware)
 - ✓ A collection of Web servers, or a single Web server, with WebLogic proxy plug-ins (configured identically)
 - ✓ Oracle WebLogic Server configured with HTTPClusterServlet

214. How will you Configuring In-Memory Replication in WLS?

- Configure the proxy server (if applicable).
- Optionally define replication groups and/or machines.
- Specify the persistence type in the weblogic.xml deployment descriptor; the options include:
 - replicated
 - replicated_if_clustered
 - async-replication-across-cluster

215. Explain about JDBC Replication in WLS?

- HTTP sessions are 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).
- 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.

216. How will you Configuring JDBC Replication in WLS?

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

Ex:

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

217. JDBC Persistent Table Configuration

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

Column Head	Column Data Type
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
WLTS_NEW	numeric, 20 digits

218. Explain about File Replication?

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

- 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

219. How will you Configuring File Replication in WLS?

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

Ex:

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

220. Explain about Cross-Cluster Replication in WLS?

WebLogic provides the ability to replicate HTTP sessions across two clusters in separate domains:

- This is most applicable to clusters that are distributed geographically.
- Configure a global proxy to direct clients back to the same cluster ("cluster affinity").
- Configure a specific network channel for cross-cluster communication.