Architecting OSB for High Availability and Whole Server Migration
Final Draft

By Jack Desai, Oracle A-Team

Published: October 2009

Contents

1 Introduction

Oracle Service Bus (OSB) is a key component of SOA Suite and Event Driven Architecture Suite in the Oracle Fusion Middleware product family. OSB uniquely delivers the integration capabilities of an Enterprise Service Bus (ESB) with operational service management in a single product with an efficient, seamless user experience. With its flexible deployment options and interoperability with SOA Governance, OSB is designed to handle the deployment, management and governance challenges of implementing Service-Oriented Architecture (SOA) from department to enterprise scale.

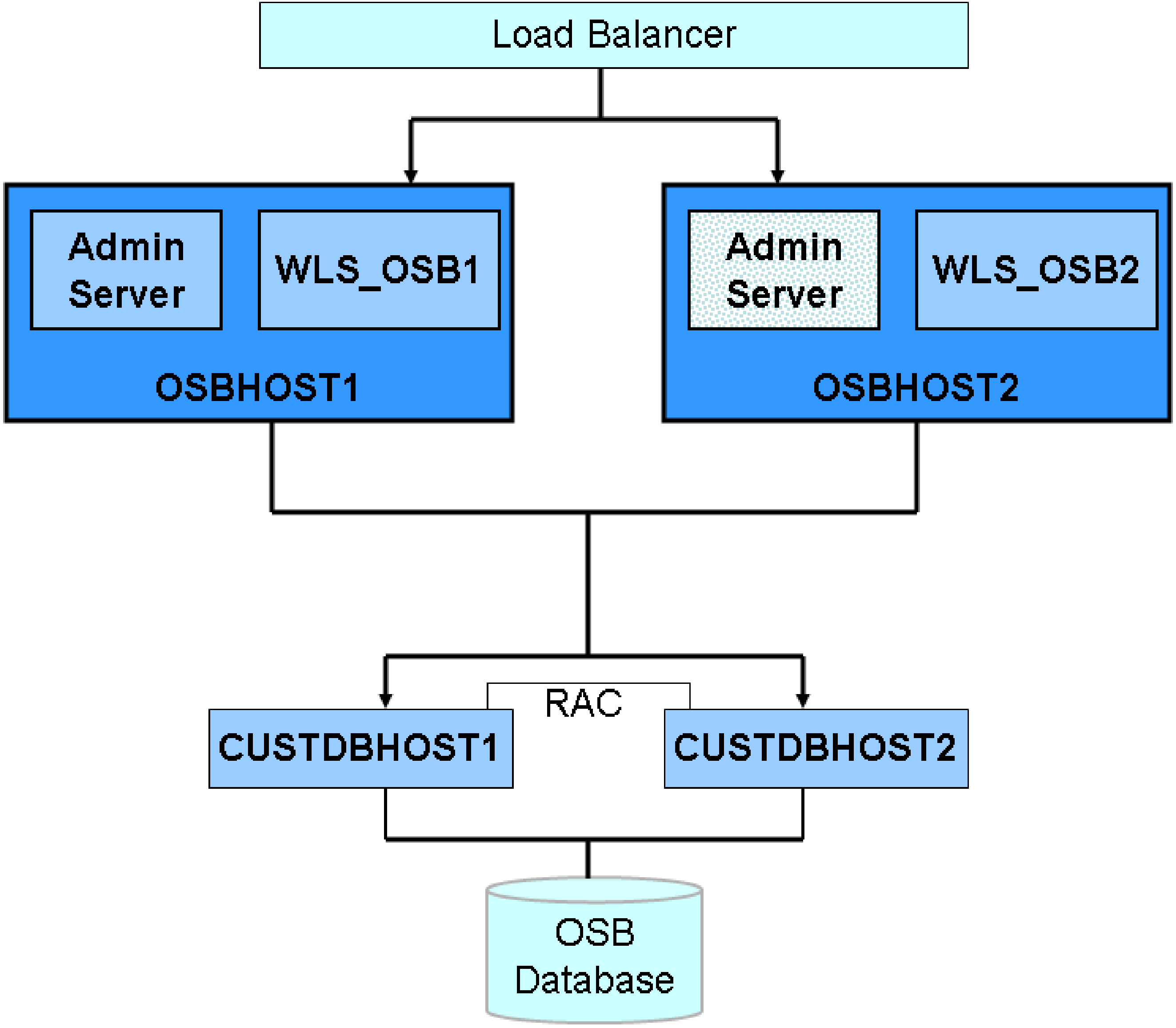
OSB is a proven, lightweight SOA integration platform, designed for connecting, mediating and managing interactions between heterogeneous services, not just web services, but also Java and .Net, messaging services and legacy endpoints. The purpose of this document is to provide the detailed steps of architecting and implementing OSB for high availability that is critical for a mission critical production deployment.

The best practices described in this document include High Availability, Scalability and Whole Server Migration across the entire OSB technology stack: Oracle RAC Database, WebLogic Admin Server and Oracle OSB Managed Servers.

1.1 OSB HA Reference Topology

The most basic deployment architecture recommendation is depicted in Figure 1. The OSB Server tier of architecture is fronted by a load balancer and consists of two clustered OSB Servers, depending on a two-node RAC cluster in the database tier. This configuration affords protection against both OSB node failure and DB node failure. Readers should be aware that a load balancer is included in the depicted architecture for completeness, but is only relevant if OSB systems deployed to the cluster are initiated by HTTP invocation.

<Figure 1: OSB HA Reference Topology Diagram>



In this architecture, there are two nodes OSBHOST1 and OSBHOST2 that run Oracle WebLogic Server configured with managed servers for running OSB components. The managed servers are configured in an active-active manner.

OSBHOST1 and OSBHOST2 also run the Oracle WebLogic Server Administration Console, but in an active-passive configuration. You can fail over the Administration Server manually (see Section 3.22, "Manually Failing Over the Administration Server to OSBHOST2"); alternatively you can configure the Oracle WebLogic Server Administration Console with CFC/CRS to fail over automatically on a separate hardware cluster (not shown in this architecture).

An RAC database runs on the nodes CUSTDBHOST1 and CUSTDBHOST2. The database contains the schemas needed by the OSB components. The OSB components running in the application tier access this database.

2 Installation and Pre-Configuration

The following sections elaborate the details of the required installation and configuration steps.

2.1 What to Install

Table 1-2 identifies the source for installation of each software component. For more information, see <add OSB doc Links>

Table 1-2 Components and Installation Sources

Component	**Distribution Medium**
Oracle Database 10g or 11g	Oracle Database CD (in 10g series, 10.2.0.4 or higher; in 11g series, 11.1.0.7 or higher)
Oracle OSB 10gR3 Server	Oracle Weblogic Server (10gR3) DVD
Oracle HTTP Server	Oracle Fusion Middleware WebTier and Utilities 11g (11.1.1.1.0) DVD

2.2 Database and Environment Pre-configuration

For the OSB enterprise topology, the database contains the repository, which is a collection of schemas used by various OSB components and Whole Server Migration

2.1 RAC Database Requirements

Oracle RAC database is implemented. The following versions of the Oracle Database are supported:

- Oracle Database 11g (11.1.0.7 or higher). It is recommended to use the Enterprise Edition version, particularly if a Disaster Recovery site is planned or expected.
- Oracle Database 10g (10.2.0.4 or higher). Oracle Database 10g Express Edition (Oracle Database XE) is not supported in Oracle Fusion Middleware 11g release 1. It is recommended to use the Enterprise Edition version, particularly if a Disaster Recovery site is planned or expected.

To check the release of your database, you can query the `PRODUCT_COMPONENT_VERSION` view as follows:

```
SQL> SELECT VERSION FROM SYS.PRODUCT_COMPONENT_VERSION WHERE PRODUCT LIKE 'Oracle%';
```

2.1.1 Initialization Parameters

Ensure that the following initialization parameter is set to the required minimum value.

Table 2-1 Required Initialization Parameters

Configuration	Parameter	Required Value	Parameter Class
OSB	`PROCESSES`	300 or greater	Static

To check the value of the initialization parameter using SQL*Plus, you can use the SHOW PARAMETER command.

As the SYS user, issue the SHOW PARAMETER command as follows:

```
SQL> SHOW PARAMETER processes
```

Set the initialization parameter using the following command:

```
SQL> ALTER SYSTEM SET processes=300 SCOPE=SPFILE;
```

Restart the database.

Note:

The method that you use to change a parameter's value depends on whether the parameter is static or dynamic, and on whether your database uses a parameter file or a server parameter file. See the Oracle Database Administrator's Guide for details on parameter files, server parameter files, and how to change parameter values.

2.1.2 Database Services

The database services must be created for the OSB Server to connect to the database. For complete instructions on creating database services, see the chapter on workload management in the Oracle Database Oracle Clusterware and Oracle Real Application Clusters Administration and Deployment Guide.

You can also use SQL*Plus to configure this using the following instructions:

1. Use the `CREATE_SERVICE` subprogram to create the `osbsvc.mycompany.com` database service. Log on to SQL*Plus as the sysdba user and run the following command:
2. `SQL> EXECUTE DBMS_SERVICE.CREATE_SERVICE`
3. `(SERVICE_NAME => 'osbsvc.mycompany.com',`
4. `NETWORK_NAME => 'osbsvc.mycompany.com',`
5. `);`
6. Add the service to the database and assign it to the instances using `srvctl`:
7. `prompt> srvctl add service -d osbdb -s osbsvc -r osbdb1,osbdb2`
8. Start the service using `srvctl`:
9. `prompt> srvctl start service -d osbdb -s osbsvc`

Note:

For more information about the SRVCTL command, see the Oracle Real Application Clusters Administration and Deployment Guide.

2.2 Network

This section covers the following topics:

- Section 2.2.3, "IPs and Virtual IPs"
- Section 2.2.4, "Firewalls and Ports"

2.2.1 IPs and Virtual IPs

Configure the Administration Server and the managed servers to listen on different virtual Ips and physical IPs. Each VIP and IP is attached to the WebLogic server that uses it. VIP1 is failed manually to restart the Administration Server in OSBHOST2. VIP2 and VIP3 fail over from OSBHOST1 to OSBHOST2 and from OSBHOST2 to OSBHOST1 respectively through Oracle WebLogic Server Migration feature. Physical IPs (non virtual) are fixed to each node. IP1 is the physical IP of OSBHOST1. IP2 is the physical IP of OSBHOST2.

Table 2-2 provides descriptions of the various virtual hosts.

Table 2-2 Virtual Hosts

Virtual IP	**VIP Maps to...**	**Description**
VIP1	OSBHOST1VHN1	OSBHOST1VHN1 is the virtual host name that is the listen address for the Administration Server and fails over with manual failover of the Administration Server. It is enabled on the node where the Admin Server process is running (OSBHOST1 by default).
VIP2	OSBHOST1VHN2	OSBHOST1VHN2 is the virtual host name that maps to the listen address for WLS_OSB1 and fails over with server migration of this managed server. It is enabled on the node where WLS_OSB1 process is running (OSBHOST1 by default).
VIP3	OSBHOST2VHN1	OSBHOST2VHN1 is the virtual host name that maps to the listen address for WLS_OSB2 and fails over with server migration of this managed server. It is enabled on the node where WLS_OSB2 process is running (OSBHOST2 by default).

2.2.2 Ports

Many Oracle Service Bus Middleware components and services use ports. As an administrator, you must know the port numbers used by these services, and to ensure that the same port number is not used by two services on a host.

Most port numbers are assigned during installation.

Table 2-3 lists the ports used in this OSB topology.

Table 2-3 Ports Used

Type	Port and Port Range	Protocol / Application	Other Considerations and Timeout Guidelines
Admin Server	7001	HTTP/t3	Set the timeout to a short period (5-10 seconds).
OSB Managed Server	8001	HTTP / WLS_OSBn	Timeout varies based on the type of process model used for OSB.
Communication between OSB Cluster members	8001	TCP/IP	By default, this communication uses the same port as the server's listen address.
Session replication within a WebLogic Server cluster	n/a	n/a	By default, this communication uses the same port as the server's listen address.
Node Manager	5556	TCP/IP	n/a
Database access	1521	SQL*Net	Timeout depends on all database content and on the type of process model used for OSB.

2.3 Shared Storage and Recommended Directory Structure

This following section details the directories and directory structure that is implemented for this reference topology. Other directory layouts are possible and supported, but the model adopted in this guide is chosen for maximum availability, providing both the best isolation of components and symmetry in the configuration and facilitating backup and disaster recovery. The rest of the document uses this directory structure and directory terminology.

2.3.1 Terminology for Directories and Directory Environment Variables

- ORACLE_BASE: This environment variable and related directory path refers to the base directory under which Oracle products are installed.
- MW_HOME: This environment variable and related directory path refers to the location where OSB Middleware resides.
- WL_HOME: This environment variable and related directory path contains installed files necessary to host a WebLogic Server.
- DOMAIN Directory: This directory path refers to the location where the Oracle WebLogic Domain information (configuration artifacts) is stored. Different WLS Servers can use different domain directories even when in the same node as described below.

2.3.2 Recommended Locations for the Different Directories

Oracle Service Bus Middleware allows creating multiple OSB servers from one single binary installation. This allows the installation of binaries in a single location on a shared storage and the reuse of this installation by the servers in different nodes. However, for maximum availability, it is recommended to use redundant binary installations. In the example, two MW HOMEs (each of which has a WL_HOME and an ORACLE_HOME for each product suite) are installed in a shared storage. Additional servers (when scaling out or up) of the same type can use either one of these two locations without requiring more installations. Ideally, users should use two different volumes (referred to as VOL1 and VOL2 below) for redundant binary location, thus isolating as much as possible the failures in each volume. For additional protection, Oracle recommends that these volumes are disk mirrored. If multiple volumes are not available, user can use mount points to simulate the same mount location in a different directory in the shared storage. Although this does not guarantee the protection that multiple volumes provide, it does allow protection from user deletions and individual file corruption.

When an ORACLE_HOME or a WL_HOME is shared by multiple servers in different nodes, it is recommended to maintain the Oracle Inventory and Middleware home list in those nodes updated for consistency in the installations and application of patches. To update the oraInventory in a node and "attach" an installation in a shared storage to it, use `ORACLE_HOME/oui/bin/attachHome.sh`. To update the Middleware home list to add or remove a WL_HOME, edit the `<user_home>/bea/beahomelist` file. This would be required for any nodes installed additionally to the two ones used in this configuration.

In this topology, it is separating the domain directory used by the administration server from the domain directory used by managed servers. This allows a symmetric configuration for the domain directories used by managed server, and isolates the failover of the administration server. The domain directory for the administration server must reside in a shared storage to allow failover to another node with the same configuration. The managed servers' domain directories can reside in a local or shared storage.

It is possible to use a shared domain directory for all managed servers in different nodes or use one domain directory per node. Sharing domain directories for managed servers facilitates the scale-out procedures. In this case, the deployment should conform to the requirements (if any) of the storage system to facilitate multiple machines mounting the same shared volume.

All procedures that apply to multiple local domains apply to a single shared domain. Hence, it uses a model where one domain directory is used per node. The directory can be local or reside in shared storage.

JMS file stores and JTA transaction logs need to be placed on a shared storage in order to ensure that they are available from multiple boxes for recovery in the case of a server failure or migration.

Based on the above assumptions, the following paragraphs describe the directories recommended. Wherever a shared storage location is directly specified, it is implied that shared storage is required for that directory. When using local disk or shared storage is optional the mount specification is qualified with "if using a shared disk." The shared storage locations are examples and can be changed as long as the provided mount points are used. However, this structure provides consistency and simplicity.

ORACLE_BASE: /u01/app/oracle

MW_HOME (application tier): ORACLE_BASE/product/osbmw

WL_HOME: MW_HOME/wlserver_10.3

OSB_HOME: MW_HOME/osb_10.3

Domain Directory for Admin Directory:
ORACLE_BASE/admin/<domain_name>/aserver/<domain_name> (The last “domain_name” gets added by config wizard)

Domain Dir for Managed Server Dir:
ORACLE_BASE/admin/<domain_name>/mserver/<domain_name>

Note: This procedure is really shared storage dependent. The above example is specific to NAS, but other storage types may provide this redundancy with different types of mappings.

Location for JMS file-based stores and Tlogs (OSB only):
ORACLE_BASE/admin/<domain_name>/<osb_cluster_name>/jms

ORACLE_BASE/admin/<domain_name>/<osb_cluster_name>/tlogs

Location for Application Directory: ORACLE_BASE/admin/<domain_name>/apps

Figure 2-2 Directory Structure

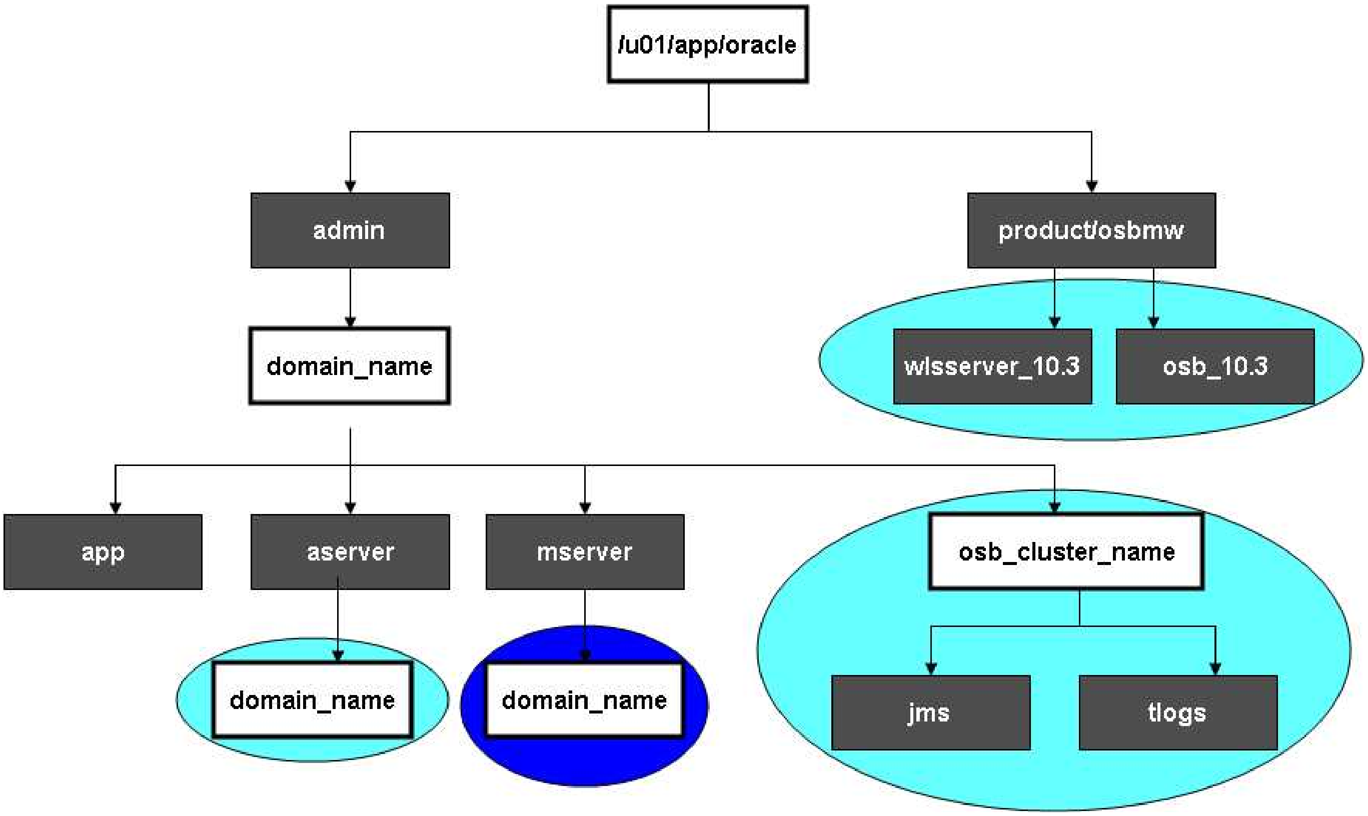


Table 2-4 Directory Structure Elements

Element	Explanation
●	The administration server domain directories, applications, deployment plans, file adapter control directory, JMS and TX logs, and the entire MW_HOME are on a shared disk.
🔵	The managed server domain directories can be on a local disk or a shared disk. Further, if you want to share the managed server domain directories on multiple nodes, then you must mount the same shared disk location across the nodes. The `instance_name` directory for the web tier can be on a local disk or a shared disk.
■	Fixed name.
□	Installation-dependent name.

Table 2-5 summarizes the directory structure for the domain.

Table 2-5 Contents of Shared Storage

Server	**Type of Data**	**Volume in Shared Storage**	**Directory**	**Files**
WLS_OSB1 / WLS_OSB2	Tx Logs	VOL1	ORACLE_BASE/admin/<domain_name>/<osb_cluster_name>/tlogs	The transaction directory is common (decided by WebLogic Server), but the files are separate.
WLS_OSB1 / WLS_OSB2	JMS Stores	VOL1	ORACLE_BASE/admin/<domain_name>/<osb_cluster_name>/jms	The transaction directory is common (decided by WebLogic Server), but the files are separate; for example: OSBJMSStore1, UMSJMSStore1, and so on.
WLS_OSB1	WLS Install	VOL1	ORACLE_BASE/product/osbmw	Individual in each volume, but both servers see same directory structure.
WLS_OSB2	WLS Install	VOL2	ORACLE_BASE/product/osbmw	Individual in each volume, but both servers see same directory structure.
WLS_OSB1	OSB Install	VOL1	ORACLE_BASE/product/osbmw/osb_10.3	Individual in each volume, but both servers see same directory structure.
WLS_OSB2	OSB Install	VOL2	ORACLE_BASE/product/osbmw/osb_10.3	Individual in each volume, but both servers see same directory structure.
WLS_OSB1	Domain Config	VOL1	ORACLE_BASE/admin/<domain_name>/mserver/<domain_name>	Individual in each volume, but both servers see same directory structure.
WLS_OSB2	Domain Config	VOL2	ORACLE_BASE/admin/<domain_name>/mserver/<domain_name>	Individual in each volume, but both servers see same directory structure.

The detailed instruction on creating the shared storage is defined in “A2 Shared Storage Configuration".

3 Creating a OSB Domain

This chapter describes how to create a domain using the Configuration Wizard, Oracle WebLogic Server Administration Console and Oracle Service Bus.

The first step is to set up a user and tablespace for the OSB Server:

1. Create a tablespace called `wls_osb`. For example, log on to SQL*Plus as the sysdba user and run the following command:

```
2.   SQL> create tablespace wls_osb_ts
3.           logging datafile 'DB_HOME/oradata/orcl/wls_osb_ts.dbf'
4.           size 32m autoextend on next 32m maxsize 2048m extent management
   local;
```

5. Create a user named `wls_osb` and assign to it the wls_osb_ts tablespace.

```
6.   SQL> create user wls_osb identified by welcome1;
7.
8.   SQL> grant create table to wls_osb;
9.
10.  SQL> grant create session to wls_osb;
11.
12.  SQL> alter user wls_osb default tablespace wls_osb_ts;
13.
14.  SQL> alter user wls_osb quota unlimited on wls_osb_ts;
```

3.1 Installing Oracle Service Bus Middleware Home

As described in Section 2.3, "Shared Storage and Recommended Directory Structure," you install Oracle Service Bus Middleware in at least two storage locations for redundancy.

3.1.1 Installing Oracle Service Bus Server

Perform these steps to install Oracle OSB Server on OSBHOST1 and OSBHOST2.

1. Start the Oracle OSB Server installer.
2. In the Welcome screen, click Next.
3. In the Choose BEA Home Directory screen, do the following:
    - Select Create a New BEA Home.
    - For BEA Home Directory, enter ORACLE_BASE/product/osbmw.

      Note:

      ORACLE_BASE is the base directory under which Oracle products are installed. The recommended value is `/u01/app/oracle`. See Section 2.3, "Shared Storage and Recommended Directory Structure" for more information.

4. Click Next.

5. In the Choose Install Type screen, select Typical, and click Next. The following products and component are chosen: Weblogic Server, Workshop, and Oracle Service Bus.
6. In the Choose Product Installation Directories, accept ORACLE_BASE/product/osbmw/wlserver_10.3, ORACLE_BASE/product/osbmw/workshop_10.3, and ORACLE_*BASE/product/osbmw/osb*_10.3; click Next.
7. In the Installation Summary screen, click Next.
8. In the Installation Complete screen, unselect Run QuickStart, and click Done.

3.2 Backing up the Installation

The Fusion Middleware Home should be backed up now (make sure that you stop the server first):

```
OSBHOST1> tar -cvpf osbmwhomeback.tar ORACLE_BASE/product/osbmw
```

This creates a backup of the installation files for both Oracle WebLogic Server and Oracle Fusion Middleware for OSB.

3.3 Enabling OSBHOST1VHN1 on OSBHOST1

Please note that this step is required for failover of the Administration Server, regardless of whether OSB is installed or not.

You will associate the Administration Server with a virtual IP (OSBHOST1VHN1). Check that OSBHOST1VHN1 is enabled on OSBHOST1.

To enable the virtual IP on Linux, run the `ifconfig` command as root:

```
/sbin/ifconfig <interface:index> <IPAddress> netmask <netmask>
/sbin/arping -q -U -c 3 -I <interface< <IPAddress>
```

For example:

```
/sbin/ifconfig ethX:Y 100.200.140.206 netmask 255.255.255.0
```

Enable your network to register the new location of the virtual IP, for example:

```
/sbin/arping -q -U -c 3 -I ethX 100.200.140.206
```

```
Note: You could also run (instead of ifconfig): ip addr add 100.200.140.206/24 dev
eth0
```

Validate that the address is available by pinging it from another node, for example:

```
/bin/ping 100.200.140.206
```

In this example 'ethX' is the ethernet interface (eth0 or eth1) and Y is the index (0, 1, 2, etc.).

3.4 Running the Configuration Wizard on OSBHOST1 to Create a Domain

Run the Configuration Wizard from the WLS home directory to create a domain containing the Administration Server and Oracle Service Bus.

1. Ensure that the database where you want to install the OSB schema is running. For RAC databases, it is recommended that all instances are running, so that the validation check later on becomes more reliable.
2. Change directory to the location of the Configuration Wizard.

```
OSBHOST1> cd ORACLE_BASE/product/osbmw/wlserver_10.3/common/bin
```

3. • Start the Oracle Fusion Middleware Configuration Wizard:

```
OSBHOST1> ./config.sh
```

4. In the Welcome screen, select Create a New WebLogic Domain, and click Next.
5. The Select Domain Source screen is displayed
6. In the Select Domain Source screen, do the following:

Select Generate a domain configured automatically to support the following products.

Select the following products:

   a. WebLogic Server (required)
   b. Workshop for WebLogic 10.3
   c. Oracle Service Bus

Click Next.

7. In the Configure Administrator Username and Password screen, enter the username and password to be used for the domain's administrator.

Click Next.

8. In the Configure Server Start Mode and JDK screen, do the following:
9. For WebLogic Domain Startup Mode, select Production Mode.
10. For JDK Selection, select JROCKIT SDK1.6.0_05.
11. Click Next.
12. In Customize Environment and Services Settings, select “Yes” and click Next.
13. In Configure RDBMS Security Store Database, Click Next to keep the default setting (bypass RDBMS option).
14. In the Configure the Administration Server screen, enter the following values:

   - Name: AdminServer
   - Listen Address: enter OSBHOST1VHN1.
   - Listen Port: 7001

- SSL listen port: N/A
- SSL enabled: unchecked

Click Next.

15. In the Configure Managed Servers screen, click Add to add the following managed servers:
16. Table 4-1 Managed Servers

Name	Listen Address	Listen Port	SSL Listen Port	SSL Enabled
WLS_OSB1	VIP1	8001	n/a	No
WLS_OSB2	VIP2	8001	n/a	No

Click Next.

17. In the Configure Clusters screen, Click Add to add the following clusters:

Table 4-2 Clusters

Name	Multicast Address	Multicast Port	Cluster Address
OSB_Cluster	<default=239.192.0.0>	<default=7001>	Leave it empty.

Click Next.

18. In the Assign Servers to Clusters screen, assign servers to clusters as follows:

OSB_Cluster:

a. WLS_OSB1
b. WLS_OSB2

Click Next.

19. In the Configure Machines screen, do the following:

Click the Unix Machine tab and then click Add to add the following machines:

Table 4-3 Machines

Name	Node Manager Listen Address	Node Manager Listen Port
OSBHOST1	OSBHOST1	5556

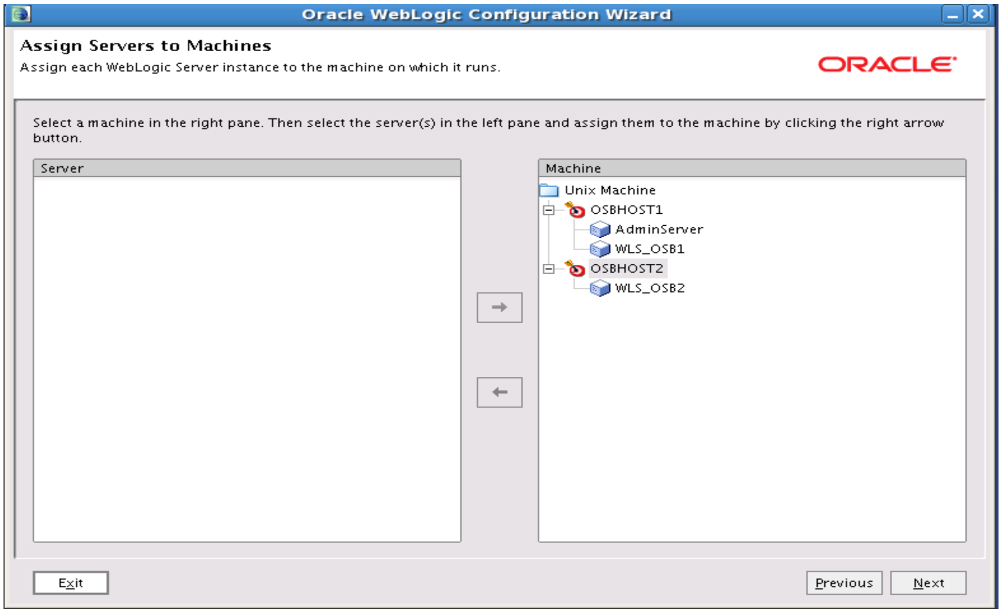
Name	Node Manager Listen Address	Node Manager Listen Port
OSBHOST2	OSBHOST2	5556

Leave all other fields to their default values.

Click Next.

20. In the Assign Servers to Machines screen, assign servers to machines as follows:

- OSBHOST1:
    - AdminServer
    - WLS_OSB1
- OSBHOST2:
    - WLS_OSB2



Oracle WebLogic Configuration Wizard

Assign Servers to Machines

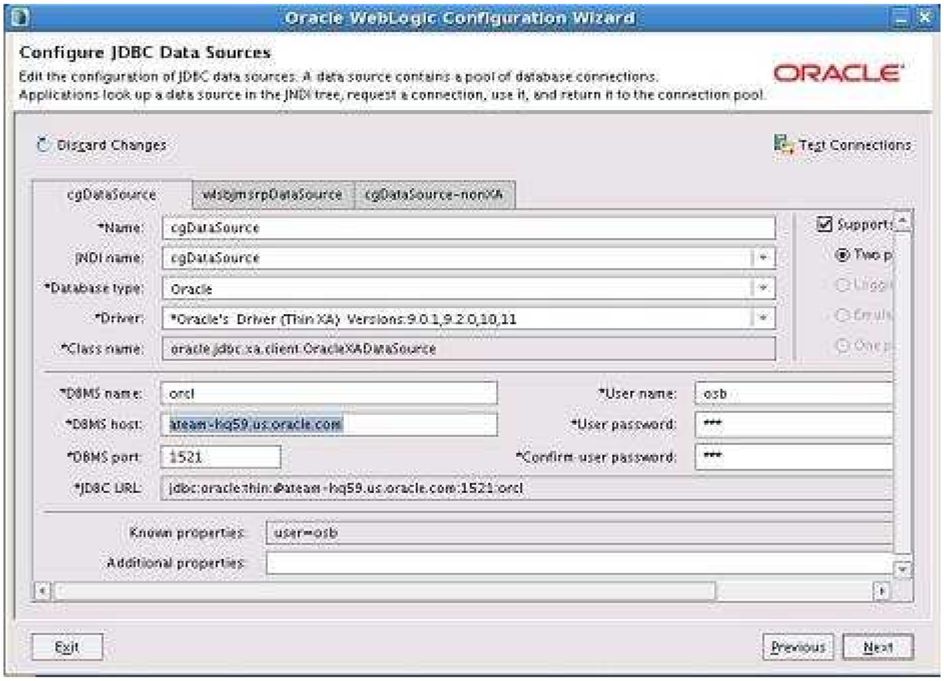
Assign each WebLogic Server instance to the machine on which it runs.

Select a machine in the right pane. Then select the server(s) in the left pane and assign them to the machine by clicking the right arrow button.

Click Next.

21. In the Configure JDBC Datasources, enter Oracle database credentials and related information. In section 4.24 Configuring JDBC Data Sources for Oracle RAC Database, we will change it to use RAC using WebLogic Multipool Data Sources. You must use the database user name “wls_osb” created in the beginning of the chapter 3.0.

For cgDataSource and wlsjmsrpDataSource



Configure JDBC Data Sources

Edit the configuration of JDBC data sources. A data source contains a pool of database connections. Applications look up a data source in the JNDI tree, request a connection, use it, and return it to the connection pool.

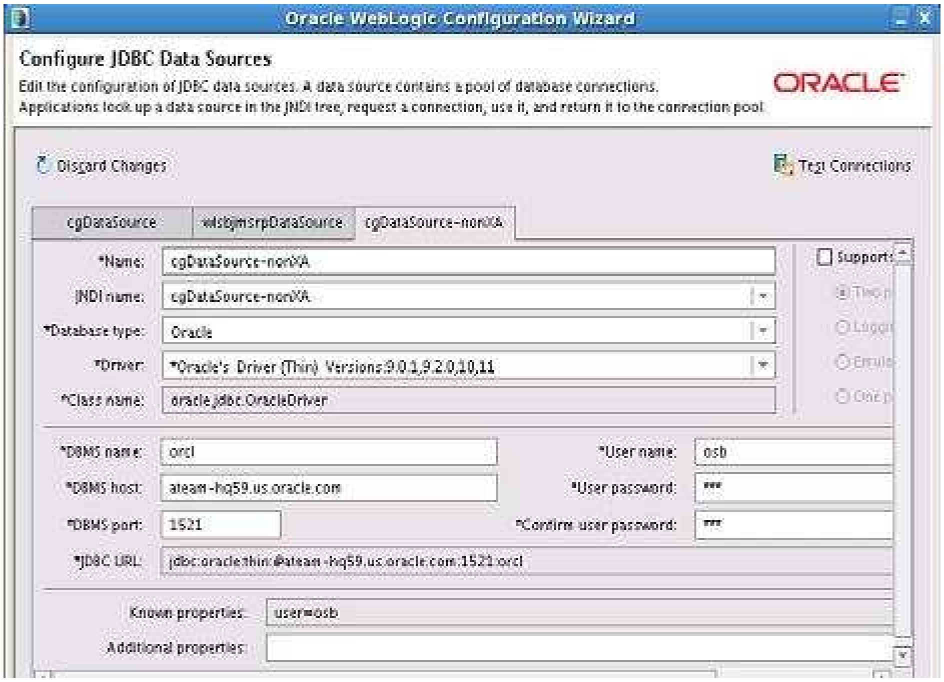
cgDataSource | wlsbjmsrpDataSource | cgDataSource-nonXA

*Name: cgDataSource
JNDI name: cgDataSource
*Database type: Oracle
*Driver: *Oracle's Driver (Thin XA) Versions:9.0.1,9.2.0,10,11
*Class name: oracle.jdbc.xa.client.OracleXADataSource

☑ Support
● Two p
○ Loggi
○ Emul
○ One p

*DBMS name: orcl
*DBMS host: ateam-hq59.us.oracle.com
*DBMS port: 1521
*JDBC URL: jdbc:oracle:thin:@ateam-hq59.us.oracle.com:1521:orcl
*User name: osb
*User password: ***
*Confirm user password: ***
Known properties: user=osb
Additional properties:

For cgDataSource-nonXA



Configure JDBC Data Sources

Edit the configuration of JDBC data sources. A data source contains a pool of database connections. Applications look up a data source in the JNDI tree, request a connection, use it, and return it to the connection pool.

cgDataSource | wlsbjmsrpDataSource | cgDataSource-nonXA

*Name: cgDataSource-nonXA
JNDI name: cgDataSource-nonXA
*Database type: Oracle
*Driver: *Oracle's Driver (Thin) Versions:9.0.1,9.2.0,10,11
*Class name: oracle.jdbc.OracleDriver

□ Support
○ Two p
○ Loggi
○ Emul
○ One p

*DBMS name: orcl
*DBMS host: ateam-hq59.us.oracle.com
*DBMS port: 1521
*JDBC URL: jdbc:oracle:thin:@ateam-hq59.us.oracle.com:1521:orcl
*User name: osb
*User password: ***
*Confirm user password: ***
Known properties: user=osb
Additional properties:

Click Next.

22. In the Run Database Scripts, click Run Scripts and when complete, click Next
23. In the Configure JMS File Stores, keep the default values and click Next. In section 4.22 Configured Shared Storage for JMS Persistence Store, we will later change it for Shared Storage.
24. In the Review WebLogic Domain screen, click Next.

25. In the Create Domain screen, enter domain name and click Done.

3.5 Creating a Separate Domain Directory for Managed Servers in the Same Node as the Administration Server

Use the `pack` and `unpack` commands to separate the domain directory used by the Administration Server from the domain directory used by the managed server in OSBHOST1as recommended in Chapter 2, "Database and Environment Preconfiguration."

1. Run the pack command on OSBHOST1 to create a template pack as follows:

```
2.   OSBHOST1> cd ORACLE_BASE/product/osbmw/osb/common/bin
3.
4.   OSBHOST1> ./pack.sh -managed=true -
   domain=ORACLE_BASE/admin/<domain_name>/aserver/<domain_name> -
   template=osbdomaintemplate.jar -template_name=osb_domain_template
```

5. Run the unpack command on OSBHOST1 to unpack the template in the managed server domain directory as follows:

```
6.   OSBHOST1> cd ORACLE_BASE/product/osbmw/osb/common/bin
7.
8.   OSBHOST1> ./unpack.sh -
   domain=ORACLE_BASE/admin/<domain_name>/mserver/<domain_name> -
   template=osbdomaintemplate.jar
```

3.6 Creating boot.properties for the Administration Server on OSBHOST1

Create a `boot.properties` file for the Administration Server on OSBHOST1. The `boot.properties` file enables the Administration Server to start without prompting you for the administrator username and password.

For the Administration Server:

1. Create the following directory structure:

```
2.   mkdir ORACLE_BASE/admin/<domain_name>/aserver/<domain_name>/servers
3.
4.   mkdir
   ORACLE_BASE/admin/<domain_name>/aserver/<domain_name>/servers/AdminServer
5.
6.   mkdir
   ORACLE_BASE/admin/<domain_name>/aserver/<domain_name>/servers/AdminServer/secur
   ity
```

7. In a text editor, create a file called `boot.properties` in the last directory created in the previous step, and enter the following lines in the file:

```
8.   username=<adminuser>
9.   password=<password>
```

Note:

When you start the Administration Server, the username and password entries in the file get encrypted. You start the Administration Server in Section 3.7, "Starting the Administration Server on OSBHOST1."

For security reasons, you want to minimize the time the entries in the file are left unencrypted: after you edit the file, you should start the server as soon as possible so that the entries get encrypted.

3.7 Starting the Administration Server on OSBHOST1

Start the Administration Server:

```
OSBHOST1> cd ORACLE_BASE/admin/<domain_name>/aserver/<domain_name>/bin
OSBHOST1> ./startWebLogic.sh
```

3.8 Validating the Administration Server

Perform these steps to ensure that the Administration Server is properly configured:

1. In a browser, go to `http://OSBHOST1VHN1:7001/console`.
2. Log in as the administrator.
3. Verify that the WLS_OSB1 and WLS_OSB2 managed servers are listed.
4. Verify that the OSB_Cluster cluster is listed.

3.9 Disabling Host Name Verification for the WLS_OSB1 Managed Server

This step is required if you have not set up the appropriate certificates to authenticate the different nodes with the administration server (see Chapter 4, "Setting Up Node Manager"). If you have not configured the server certificates, you will receive errors when managing the different WebLogic Servers. To avoid these errors, disable host name verification while setting up and validating the topology, and enable it again once the configuration is complete as described in Chapter 4, "Setting Up Node Manager."

To disable host name verification, complete these steps:

1. Log in to Oracle WebLogic Server Administration Console.
2. Click Lock and Edit.
3. Expand the Environment node in the Domain Structure window.
4. Click Servers. The Summary of Servers page appears.
5. Select WLS_OSB1 (represented as a hyperlink) from the Names column of the table. The Settings page appears.
6. Select the SSL tab.
7. Expand the Advanced section of the page.
8. Set Hostname Verification to None.
9. Click Save.
10. Repeat these steps for the WLS_OSB2 managed server.
11. Save and activate the changes.

3.9.1 Modify other NodeManager Properties

Please note that this steps are required if SSL is enabled, but keystores not configured as defined in “section 4.2 Enabling Host Name Verification Certificates” and security is not configured to use Oracle database

Open ORACLE_BASE/wlserver_10.3/common/bin/commEnv.sh

a. Add export JAVA_OPTIONS="$JAVA_OPTIONS -Djava.security.egd=file:/dev/./urandom"
b. Add export JAVA_OPTIONS="$JAVA_OPTIONS -Dweblogic.nodemanager.sslHostNameVerificationEnabled=false"
c. Save file.

3.10 Starting/Restarting the Node Manager on OSBHOST1/OSBHOST2

To start/restart the Node Manager on OSBHOST1/OSBHOST2, complete these steps:

1. Stop Node Manager (if running)

```
Example: OSBHOST1> kill -9 <pid> (pid => process id of node manager)
```

2. Start Node Manager:

```
Example:
OSBHOST1> cd /ORACLE_BASE/product/osbmw/wlserver_10.3/server/bin
./startNodeManager.sh
```

3.11 Propagating the Domain Changes to the Managed Server Domain Directory

To propagate the start scripts and classpath configuration from the Administration Server's domain directory to the managed server domain directory:

1. Create a copy of the managed server domain directory and the managed server applications directory.
2. Move these directories using the following commands (both on a single line):
3. `OSBHOST1> mv ORACLE_BASE/admin/<domain_name>/mserver/<domain_name> ORACLE_BASE/admin/<domain_name>/mserver/<domain_name>backup`
4. Run the `pack` command on OSBHOST1 to create a template pack using the following commands:
5. `OSBHOST1> cd ORACLE_BASE/product/osbmw/wlserver_10.3/common/bin`
6.
7. `OSBHOST1> ./pack.sh -managed=true -domain=ORACLE_BASE/admin/<domain_name>/aserver/<domain_name> -template=osbdomaintemplateExtOSB.jar -template_name=osb_domain_templateExtOSB`
8. Run the `unpack` command on OSBHOST1 to `unpack` the propagated template to the domain directory of the managed server using the following command:
9. `OSBHOST1> ./unpack.sh -domain=ORACLE_BASE/admin/<domain_name>/mserver/<domain_name>/ -`

```
template=osbdomaintemplateExtOSB.jar -
app_dir=ORACLE_BASE/admin/<domain_name>/mserver/apps
```

3.12 Setting OSB_Cluster’s cluster address

1. Login to AdminServer and navigate to Environment/cluster
2. Click on OSB_Cluster.
3. In “Cluster Address:” add the following: <OSBHOST1>:7001,<OSBHOST2>:7001 (there is no space after “’”)
4. Restart Admin Server in Section 3.7, "Starting the Administration Server on OSBHOST1."

3.13 Starting the WLS_OSB1 Managed Server on OSBHOST1

To start the WLS_OSB1 managed server on OSBHOST1, complete these steps:

1. Access the Administration Console at `http://OSBHOST1VHN1:7001/console`.
2. Click Servers.
3. Open the Control tab.
4. Select WLS_OSB1.
5. Click Start.

Note:

OSBHOST1VHN1 is the virtual host name that maps to the virtual IP where the Administration Server is listening (in OSBHOST1).

3.14 Validating the WLS_OSB1 Managed Servers

To validate the WLS_OSB1 managed servers, complete these steps:

Verify that the server status is reported as "Running" in the Admin Console. If the server is shown as "Starting" or "Resuming," wait for the server status to change to "Started." If another status is reported (such as "Admin" or "Failed"), check the server output log files for errors.

3.15 Propagating the Domain Configuration to OSBHOST2 Using the unpack Utility

To propagate the domain configuration, complete these steps:

1. Run the following command on OSBHOST1 to copy the template file created in the previous step to OSBHOST2.
2. `OSBHOST1> cd /ORACLE_BASE/product/osbmw/wlserver_10.3/common/bin`
3.
4. `OSBHOST1> scp osbdomaintemplateExtOSB.jar oracle@node2:ORACLE_BASE/product/osbmw/wlserver_10.3/common/bin`
5. Run the `unpack` command on OSBHOST2 to unpack the propagated template.

   Note:

"It is required to remove the existing domain directory in OSBHOST2 or the operation will fail. The unpack command will create a new domain directory for you. This means that if you have made any configuration changes specific to the managed servers (such as WLS_WSM2) running on the node (OSBHOST2) where you are running the unpack command, you will have to redo these configuration changes.

```
OSBHOST2> cd ORACLE_BASE/product/osbmw/wlserver_10.3/common/bin

OSBHOST2> ./unpack.sh -
domain=ORACLE_BASE/admin/<domain_name>/mserver/<domain_name>/ -
template=osbdomaintemplateExtOSB.jar -
app_dir=ORACLE_BASE/admin/<domain_dir>/mserver/apps
```

To Enroll OSBHOST2 Node Manager to Admin Server, complete these steps

```
OSBHOST2>cd ORACLE_BASE/admin/<domain_name>/mserver/<domain_name>/bin
OSBHOST2>Source setDomainEnv.cmd
OSBHOST2>java weblogic.WLST
OSBHOST2>connect('weblogic','welcome1','t3://osbhost1:7001')
OSBHOST2>nmEnroll('/ORACLE_BASE/admin/<domain_name>/mserver/<domain_name>','/ORACLE_BASE/wlserver_10.3/common/nodemanager')
```

3.16 Disabling Host Name Verification for the WLS_OSB2 Managed Server

Perform the steps in Section 3.9, "Disabling Host Name Verification" on OSBHOST2.

3.17 Restarting Node Manager on OSBHOST2

Perform the steps in Section 3.10, "Starting/Restarting the Node Manager on OSBHOST1/OSBHOST2" on OSBHOST2.

3.18 Starting and Validating the WLS_OSB2 Managed Server

Perform these steps to start the WLS_OSB2 managed server and check that it is configured correctly:

1. Start the WLS_OSB2 managed server using the Administration Console.
2. Verify that the server status is reported as "Running" in the Admin Console. If the server is shown as "Starting" or "Resuming," wait for the server status to change to "Started." If another status is reported (such as "Admin" or "Failed"), check the server output log files for errors.

3.19 Configuring a Shared JMS Persistence Store

Configure the location for all of the persistence stores as a directory that is visible from both nodes. For more information see Section 2.3, "Shared Storage and Recommended Directory Structure," You must then change all of the persistent stores to use this shared base directory as follows:

1. Log into the Oracle WebLogic Server Administration Console.
2. In the Domain Structure window, expand the Services node and then click the Persistence Stores node. The Summary of Persistence Stores page appears.
3. Select the persistence store (represented as a hyperlink) from the Name column of the table. The Settings page for the persistence store appear.
4. In the Configuration tab, enter the location on a persistent storage solution (such as NAS or SAN) that is available to other servers in the cluster in the Directory field. Specifying this location enables pending JMS messages to be sent. The location should follow the following directory structure:
5. `ORACLE_BASE/admin/<domain_name>/<osb_cluster_name>/jms`

   Note:

   Both WLS_OSB1 and WLS_OSB2 must be able to access this directory. This directory must also exist before you restart the server.

6. Click Save and activate changes.
7. Restart the servers to make the change in the persistent stores effective.

3.20 Configuring a Default Persistence Store for Transaction Recovery

Each server has a transaction log that stores information about committed transactions that are coordinated by the server that may not have been completed. The WebLogic Server uses this transaction log for recovery from system crashes or network failures. To leverage the migration capability of the Transaction Recovery Service for the servers within a cluster, store the transaction log in a location accessible to a server and its backup servers.

Note:

Preferably, this location should be a dual-ported SCSI disk or on a Storage Area Network (SAN).

To set the location for the default persistence store, complete these steps:

1. Log into the Oracle WebLogic Server Administration Console.
2. In the Domain Structure window, expand the Environment node and then click the Servers node. The Summary of Servers page appears.
3. Click the name of the server (represented as a hyperlink) in Name column of the table. The settings page for the selected server appears and defaults to the Configuration tab.
4. Click the Services tab.
5. In the Default Store section of the page, enter the path to the folder where the default persistent stores will store its data files. The directory structure of the path is as follows:
6. `ORACLE_BASE/admin/<domain_name>/<osb_cluster_name>/tlogs`
7. Click Save.

Note:

To enable migration of the Transaction Recovery Service, specify a location on a persistent storage solution that is available to other servers in the cluster. Both WLS_OSB1 and WLS_OSB2 must be able to access this directory. This directory must also exist before you restart the server.

3.21 Configuring JDBC Data Sources for Oracle RAC Database

Configure WebLogic Multipool Data Sources to connect to Oracle RAC databases. Ensure that all the RAC instances are running and accessible. You create a data source to each of the RAC database instances during the process of setting up the multi-data source, both for all the three OSB data sources: cgDataSource, wlsjmsrpDataSource and cgDataSource-nonXA.

When you create a data source for cgDatasource and wlsjmsrpDatasource:

- The names of the multi-data sources are in the format of <MultiDS>-rac0, <MultiDS>-rac1, and so on
- Use Oracle's Driver (Thin XA) Version 9.0.1, 9.2.0, 10, 11
- Use Supports Global Transactions, Two-Phase Commit, and specify a service name for your database
- Target these data sources to the OSB cluster

When you create a data source for cgDatasource-nonXA:

- The names of the multi-data sources are in the format of <MultiDS>-rac0, <MultiDS>-rac1, and so on
- Use Oracle's Driver (Thin) Version 9.0.1, 9.2.0, 10, 11
- Uncheck Supports Global Transactions, and specify a service name for your database
- Target these data sources to the OSB cluster

Note: The multi-data source name should be same as standalone jdbc data source. In order to create the multi-data source with the same name, shut down OSB servers and delete individual jdbc data sources respectively.

Creating a Multi-Data Source

To create a multi-data source, complete these steps for each Data Sources:

1. From Domain Structure window in the Oracle WebLogic Server Administration Console, expand the Services node, then expand the JDBC node.
2. Click Multi Data Sources. The Summary of JDBC Multi Data Source page appears.
3. Click Lock and Edit.
4. Click New. The Create a New JDBC Multi Data Source page appears.
5. Click New.
6. Enter <data_source> as the Name (Ex: wlsjmsrpDataSource)
7. Enter <data_source> as the JNDI name (Ex: wlsjmsrpDataSource)
8. Select Failover as algorithm (default).
9. Click Next.

10. Select OSB_Cluster as the target.
11. Click Next.
12. Select non-XA driver (the default) for cgDatasource-nonXA or XA driver for cgDataSource and wlsjmsrpDataSource
13. Click Next.
14. Click Create New Data Source.
15. Enter <data_source>-rac0 as name. Enter respective <data_source>-rac0 as JNDI name. Enter oracle as the database type. For the driver type, enter Oracle Driver (Thin) for RAC server-Instance connection Version 10,11 for cgDataSource-nonXA and Driver (Thin XA) for RAC server for cgDataSource and wlsjmsrpDataSource
16. Click Next.
17. Deselect Supports Global Transactions for cgDatasource-nonXA or Select for cgDataSource and wlsjmsrpDataSource
18. Click Next.
19. Enter the service name, database name, host port, and password for your OSB schema.
20. Click Next.
21. Click Test Configuration and verify the connection works.
22. Target the data source to OSB_Cluster.
23. Select the data source and add it to the right screen.
24. Click Create a New Data Source for the first instance of your RAC database, target it to OSB_Cluster, and repeat the steps for the second instance of your RAC database.
25. Add the second data source to your multi-data source.
26. Click Activate Changes.
27. Restart Amin Server and OSB Servers/Cluster.

3.22 Manually Failing Over the Administration Server to OSBHOST2

In case a node fails, you can fail over the Administration Server to another node. This section describes how to fail over the Administration Server from OSBHOST1 to OSBHOST2.

Assumptions:

- The Administration Server is configured to listen on OSBHOST1VHN1, and not on ANY address.
- These procedures assume that the two nodes use two individual domain directories, and that the directories reside in local storage or in shared storage in different volumes.
- The Administration Server is failed over from OSBHOST1 to OSBHOST2, and the two nodes have these IPs:
  - OSBHOST1: 100.200.140.165
  - OSBHOST2: 100.200.140.205
  - VIPHOST1: 100.200.140.206. This is the VIP where the Administration Server is running, assigned to ethX:Y, available in OSBHOST1 and OSBHOST2.
- The domain directory where the administration server is running in OSBHOST1 is on a shared storage and is mounted also from OSBHOST2.

The following procedure shows how to fail over the Administration Server to a different node (OSBHOST2), but the Administration Server will still use the same WebLogic Server machine (which is a logical machine, not a physical machine).

1. Stop the Administration Server.
2. Migrate IP to the second node.
   1. Run the following command as root on OSBHOST1 (where X:Y is the current interface used by OSBHOST1VHN1):
   2. `OSBHOST1> /sbin/ifconfig ethX:Y down`
   3. Run the following command on OSBHOST2:
   4. `OSBHOST2> /sbin/ifconfig <interface:index> <IP_Address> netmask <netmask>`

      For example:

      ```
      /sbin/ifconfig eth0:1 10.0.0.1 netmask 255.255.255.0
      ```

      Note:

      Ensure that the netmask and interface to be used to match the available network configuration in OSBHOST2.

3. Update routing tables through `arping`, for example:
4. `OSBHOST2> /sbin/arping -b -A -c 3 -I eth0 10.0.0.1`
5. Start the Administration Server on OSBHOST2.
6. `OSBHOST2> cd ORACLE_BASE/admin/<domain_name>/aserver/<domain_name>/bin`
7.
8. `OSBHOST2> ./startWebLogic.sh`
9. Test that you can access the Administration Server on OSBHOST2 as follows:
   1. Ensure that you can access the Oracle WebLogic Server Administration Console at `http://OSBHOST1VHN1:7001/console`.
   2. Check that you can access and verify the status of components in the Oracle Enterprise Manager at `http://OSBHOST1VHN1:7001/sbconsole`.

3.23 Failing the Administration Server Back to OSBHOST1

This step checks that you can fail back the Administration Server, that is, stop it on OSBHOST2 and run it on OSBHOST1. To do this, migrate OSBHOST1VHN1 back to OSBHOST1 node as follows:

1. Run the following command on OSBHOST2.
2. `OSBHOST2> /sbin/ifconfig ethZ:N down`
3. Run the following command on OSBHOST1:
4. `OSBHOST1> /sbin/ifconfig ethX:Y 100.200.140.206 netmask 255.255.255.0`

   Note:

   Ensure that the netmask and interface to be used match the available network configuration in OSBHOST1

5. Update routing tables through arping. Run the following command from OSBHOST1.
6. `OSBHOST1> /sbin/arping -b -A -c 3 -I ethZ 100.200.140.206`
7. Start the Administration Server again on OSBHOST1.
8. `OSBHOST1> cd ORACLE_BASE/admin/<domain_name>/aserver/<domain_name>/bin`
9. `OSBHOST1> ./startWebLogic.sh`
10.
11. Test that you can access the Oracle WebLogic Server Administration Console at `http://OSBHOST1VHN1:7001/console`.
12. Check that you can access and verify the status of components in the Oracle Enterprise Manager at `http://OSBHOST1VHN1:7001/em`.

3.24 Backing Up the Installation

Perform a backup to save your domain configuration (make sure that you stop the server first). The configuration files all exist under the `ORACLE_BASE/admin/<domain_name>` directory.

```
OSBHOST1> tar -cvpf mydomainback.tar ORACLE_BASE/admin/<domain_name>
```

4 Setting Up Node Manager

This chapter describes how to configure Node Manager to enable host name verification for the communications between Node Manager and the Administration Server. This requires the use of certificates for the different addresses communicating with the Administration Server. In this chapter, the steps for configuring OSBHOST1 and OSBHOST2 certificates for host name verification are provided.

4.1 About the Node Manager

The Node Manager enables you to start and stop the Administration Server and the managed servers.

4.2 Enabling Host Name Verification Certificates for Node Manager in OSBHOST1

Perform these steps to set up host name verification certificates for communication between the Node Manager and the Administration Server.

- Step 1: Generating Self-Signed Certificates Using the utils.CertGen Utility
- Step 2: Creating an Identity Keystore Using the utils.ImportPrivateKey Utility
- Step 3: Creating a Trust Keystore Using the keytool Utility
- Step 4: Configuring Node Manager to Use the Custom Keystores

4.2.1 Generating Self-Signed Certificates Using the utils.CertGen Utility

Follow these steps to create self-signed certificates on OSBHOST1.mycompany.com. These certificates should be created using the network name/alias. For information on on using trust CA certificates instead, see "Configuring Identity and Trust" in Oracle Fusion Middleware Securing Oracle WebLogic Server.

1. Set up your environment by running the `ORACLE_BASE/product/osbmw/wlserver_10.3/server/bin/setWLSEnv.sh` script:

   In the Bourne shell, run the following command:

   ```
   OSBHOST1> . setWLSEnv.sh
   ```

   Verify that the CLASSPATH environment variable is set:

   ```
   OSBHOST1> echo $CLASSPATH
   ```

2. Create a user-defined directory for the certificates. For example, create a directory called certs under the `ORACLE_BASE/product/osbmw/` directory. Note that certificates can be shared across WLS domains.
3. `OSBHOST1> cd ORACLE_BASE/product/osbmw`
4. `OSBHOST1> mkdir certs`
5. Change directory to the user-defined directory.

6. `OSBHOST1> cd certs`
7. Run the `utils.CertGen` tool from the user-defined directory to create the certificates for both OSBHOST1 and OSBHOST1VHN1.

   Syntax:

   ```
   java utils.CertGen <key_passphrase> <cert_file_name> <key_file_name> [export |
   domestic] [hostname]
   ```

   Examples:

   ```
   OSBHOST1> java utils.CertGen welcome1 OSBHOST1_cert OSBHOST1_key
                domestic OSBHOST1.mycompany.com

   OSBHOST1> java utils.CertGen welcome1 VIPHOST1_cert VIPHOST1_key
                domestic OSBHOST1VHN1.mycompany.com
   ```

4.2.2 Creating an Identity Keystore Using the utils.ImportPrivateKey Utility

Follow these steps to create an Identity Keystore on OSBHOST1.mycompany.com.

1. Create a new identity keystore called appIdentityKeyStore using the utils.ImportPrivateKey utility.

   Create this keystore under the same directory as the certificates (that is, `ORACLE_BASE/product/osbmw/certs`).

   Note:

   The Identity Store is created (if none exists) when you import a certificate and the corresponding key into the Identity Store using the `utils.ImportPrivateKey` utility.

   Import the certificate and private key for both OSBHOST1 and VIPHOST1 into the Identity Store. Make sure that you use a different alias for each of the certificate/key pair imported.

   Syntax:

   ```
   java utils.ImportPrivateKey <keystore_file> <keystore_password>
   <certificate_alias_to_use> <private_key_passphrase> <certificate_file>
   <private_key_file> [<keystore_type>]
   ```

   Examples:

   ```
   OSBHOST1> java utils.ImportPrivateKey appIdentityKeyStore.jks welcome1
               appIdentity1 welcome1
               ORACLE_BASE/product/osbmw/certs/OSBHOST1_cert.pem
               ORACLE_BASE/product/osbmw/certs/OSBHOST1_key.pem

   OSBHOST1> java utils.ImportPrivateKey appIdentityKeyStore.jks welcome1
               appIdentity2 welcome1
               ORACLE_BASE/product/osbmw/certs/VIPHOST1_cert.pem
   ```

```
                    ORACLE_BASE/product/osbmw/certs/VIPHOST1_key.pem
```

4.2.3 Creating a Trust Keystore Using the keytool Utility

Follow these steps to create the Trust Keystore on OSBHOST1.mycompany.com.

1. Copy the standard java keystore to create the new trust keystore since it already contains most of the root CA certificates needed. Oracle does not recommend modifying the standard Java trust key store directly. Copy the standard Java keystore CA certificates located under the WL_HOME/server/lib directory to the same directory as the certificates. For example:

```
cp WL_HOME/server/lib/cacerts
ORACLE_BASE/admin/<domain_name>/aserver/<domain_name>/certs/appTrustKeyStore.jks
```

2. The default password for the standard Java keystore is `changeit`. Oracle recommends always changing the default password. Use the keytool utility to do this. The syntax is:

```
keytool -storepasswd -new <NewPassword> -keystore <TrustKeyStore> -storepass
<Original Password>
```

For example:

```
keytool -storepasswd -new welcome1 -keystore appTrustKeyStore.jks -storepass
changeit
```

3. The CA certificate `CertGenCA.der` is used to sign all certificates generated by the utils.CertGen tool and is located at WL_HOME/server/lib directory. This CA certificate must be imported into the `appTrustKeyStore` using the keytool utility. The syntax is:

```
keytool -import -v -noprompt -trustcacerts -alias <AliasName>
 -file <CAFileLocation> -keystore <KeyStoreLocation> -storepass <KeyStore
Password>
```

For example:

```
keytool -import -v -noprompt -trustcacerts -alias clientCACert -file
      $WL_HOME/server/lib/CertGenCA.der -keystore appTrustKeyStore.jks -storepass
      welcome1
```

4.2.4 Configuring Node Manager to Use the Custom Keystores

To configure the Node Manager to use the custom keystores, add the following lines to the end of the `nodemanager.properties` file located in the `ORACLE_BASE/product/osbmw/wlserver_10.3/common/nodemanager` directory:

```
KeyStores=CustomIdentityAndCustomTrust
CustomIdentityKeyStoreFileName=<Identity KeyStore>
CustomIdentityKeyStorePassPhrase=<Identity KeyStore Passwd>
```

```
CustomIdentityAlias=<Identity Key Store Alias>
CustomIdentityPrivateKeyPassPhrase=<Private Key used when creating Certificate>
Make sure to use the correct value for CustomIdentityAlias on each node. For example
on OSBHOST1, use appIdentity1, and on VIPHOST1, use appIdentity2.
Example for Node 1:
KeyStores=CustomIdentityAndCustomTrust
CustomIdentityKeyStoreFileName=ORACLE_BASE/product/osbmw/certs/
appIdentityKeyStore.jks
CustomIdentityKeyStorePassPhrase=welcome1
CustomIdentityAlias=appIdentity1
CustomIdentityPrivateKeyPassPhrase=welcome1
```

The passphrase entries in the `nodemanager.properties` file get encrypted when you start Node Manager as described in Section 4.3, "Starting the Node Manager on OSBHOST1." For security reasons, you want to minimize the time the entries in the `nodemanager.properties` file are left unencrypted. After you edit the file, you should start Node Manager as soon as possible so that the entries get encrypted.

When using a common/shared storage installation for OSBMW_HOME, Node Manager is started from different nodes using the same base configuration (nodemanager.properties). In that case, it is required to add the certificate for all the nodes that share the binaries to the appIdentityKeyStore.jks identity store. To do this, create the certificate for the new node and import it to appIdentityKeyStore.jks as described above. Once the certificates are available in the store, each node manager needs to point to a different identity alias to send the correct certificate to the administration server. To do this, set different environment variables before starting Node Manager in the different nodes:

```
OSBHOST1>cd ORACLE_BASE/product/osbmw/wlserver_10.3/server/bin
OSBHOST1>export JAVA_OPTIONS=-DCustomIdentityAlias=appIdentityOSBHOST1

OSBHOSTn>cd ORACLE_BASE/product/osbmw/wlserver_10.3/server/bin
OSBHOSTn>export JAVA_OPTIONS=-DCustomIdentityAlias=appIdentityOSBHOSTn
```

4.3 Starting the Node Manager on OSBHOST1

Run these commands to start Node Manager on OSBHOST1:

```
OSBHOST1> cd ORACLE_BASE/product/osbmw/wlserver_10.3/server/bin
OSBHOST1> ./startNodeManager.sh
```

4.4 Enabling Host Name Verification Certificates for the Node Manager in OSBHOST2

Perform these steps to set up SSL for communication between the Node Manager and the Administration Server:

- Step 1: Generating Self-Signed Certificates Using the utils.CertGen Utility
- Step 2: Creating an Identity Keystore Using the "utils.ImportPrivateKey" Utility
- Step 3: Creating a Trust Keystore Using the keytool Utility
- Step 4: Configuring Node Manager to Use the Custom Keystores

4.4.1 Generating Self-Signed Certificates Using the utils.CertGen Utility

Follow these steps to create self-signed certificates on OSBHOST2.mycompany.com. These certificates should be created using the network name/alias.

1. Set up your environment by running the `ORACLE_BASE/product/osbmw/wlserver_10.3/server/bin/setWLSEnv.sh` script:

   In the Bourne shell, run the following command:

   ```
   OSBHOST2> . setWLSEnv.sh
   ```

   Verify that the CLASSPATH environment variable is set:

   ```
   OSBHOST2> echo $CLASSPATH
   ```

2. Create a user-defined directory for the certificates. For example, create a directory called certs under the `ORACLE_BASE/product/osbmw/` directory. Note that certificates can be shared across WLS domains.
3. `OSBHOST2> cd ORACLE_BASE/product/osbmw`
4. `OSBHOST2> mkdir certs`
5. Change directory to the user-defined directory.
6. `OSBHOST2> cd certs`
7. Run the utils.CertGen tool from the user-defined directory to create the certificates for both OSBHOST2 and VIPHOST1.

   Syntax:

   ```
   java utils.CertGen <key_passphrase> <cert_file_name> <key_file_name> [export |
   domestic] [hostname]
   ```

   Examples:

   ```
   OSBHOST2> java utils.CertGen welcome1 OSBHOST2_cert OSBHOST2_key
                domestic OSBHOST2.mycompany.com

   OSBHOST2> java utils.CertGen welcome1 VIPHOST1_cert VIPHOST1_key
                domestic OSBHOST1VHN1.mycompany.com
   ```

4.4.2 Creating an Identity Keystore Using the "utils.ImportPrivateKey" Utility

Follow these steps to create an Identity Keystore on OSBHOST2.mycompany.com.

1. Create a new identity keystore called "appIdentityKeyStore" using the "utils.ImportPrivateKey" utility.

   Create this keystore under the same directory as the certificates (that is, ORACLE_BASE/product/osbmw/certs).

Note that the Identity Store is created (if none exists) when you import a certificate and the corresponding key into the Identity Store using the "utils.ImportPrivateKey" utility.

Import the certificate and private key for both OSBHOST2 and VIPHOST1 into the Identity Store. Make sure that you use a different alias for each of the certificate/key pair imported.

Syntax:

```
java utils.ImportPrivateKey <keystore_file> <keystore_password>
<certificate_alias_to_use> <private_key_passphrase> <certificate_file>
<private_key_file> [<keystore_type>]
```

Examples:

```
OSBHOST2> java utils.ImportPrivateKey appIdentityKeyStore.jks welcome1
            appIdentity1 welcome1
            ORACLE_BASE/product/osbmw/certs/OSBHOST2_cert.pem
            ORACLE_BASE/product/osbmw/certs/OSBHOST2_key.pem

OSBHOST2> java utils.ImportPrivateKey appIdentityKeyStore.jks welcome1
            appIdentity2 welcome1
            ORACLE_BASE/product/osbmw/certs/VIPHOST1_cert.pem
            ORACLE_BASE/product/osbmw/certs/VIPHOST1_key.pem
```

4.4.3 Creating a Trust Keystore Using the keytool Utility

Follow these steps to create the Trust Keystore on OSBHOST2.mycompany.com.

1. Copy the standard java keystore to create the new trust keystore since it already contains most of the root CA certificates needed. Oracle does not recommend modifying the standard Java trust key store directly. Copy the standard Java keystore CA certificates located under the WL_HOME/server/lib directory to the same directory as the certificates. For example:

```
cp WL_HOME/server/lib/cacerts
ORACLE_BASE/admin/<domain_name>/aserver/<domain_name>/certs/appTrustKeyStore.jks
```

2. The default password for the standard Java keystore is `changeit`. Oracle recommends always changing the default password. Use the keytool utility to do this. The syntax is:

```
keytool -storepasswd -new <NewPassword> -keystore <TrustKeyStore> -storepass
<Original Password>
```

For example:

```
keytool -storepasswd -new welcome1 -keystore appTrustKeyStore.jks -storepass
changeit
```

3. The CA certificate `CertGenCA.der` is used to sign all certificates generated by the utils.CertGen tool and is located at WL_HOME/server/lib directory. This CA certificate must be imported into the `appTrustKeyStore` using the keytool utility. The syntax is:

```
keytool -import -v -noprompt -trustcacerts -alias <AliasName>
 -file <CAFileLocation> -keystore <KeyStoreLocation> -storepass <KeyStore
Password>
```

For example:

```
keytool -import -v -noprompt -trustcacerts -alias clientCACert -file
      $WL_HOME/server/lib/CertGenCA.der -keystore appTrustKeyStore.jks -storepass
      welcome1
```

4.4.4 Configuring Node Manager to Use the Custom Keystores

Follow these steps to configure the Node Manager to use the custom keystores.

1. Add the following lines to the end of the `nodemanager.properties` file located in the `ORACLE_BASE/product/osbmw/wlserver_10.3/common/nodemanager` directory.

```
KeyStores=CustomIdentityAndCustomTrust

CustomIdentityKeyStoreFileName=<Identity KeyStore>

CustomIdentityKeyStorePassPhrase=<Identity KeyStore Passwd>

CustomIdentityAlias=<Identity Key Store Alias>

CustomIdentityPrivateKeyPassPhrase=<Private Key used when creating Certificate>
```

   Make sure to use the correct value for `CustomIdentityAlias` on each node. For example on OSBHOST2, use "appIdentity2", and on VIPHOST1, use "appIdentity2".

   Example for Node 1:

```
KeyStores=CustomIdentityAndCustomTrust

CustomIdentityKeyStoreFileName=ORACLE_BASE/product/osbmw/certs/appIdentityKeySt
ore.jks

CustomIdentityKeyStorePassPhrase=welcome1

CustomIdentityAlias=appIdentity1

CustomIdentityPrivateKeyPassPhrase=welcome1
```

   Note:

The passphrase entries in the `nodemanager.properties` file get encrypted when you start Node Manager, as described in Section 4.5, "Starting the Node Manager on OSBHOST2."

For security reasons, you want to minimize the time the entries in the `nodemanager.properties` file are left unencrypted. After you edit the file, you should start Node Manager as soon as possible so that the entries get encrypted.

4.5 Starting the Node Manager on OSBHOST2

Run these commands to start Node Manager on OSBHOST2:

```
OSBHOST2> cd ORACLE_BASE/product/osbmw/wlserver_10.3/server/bin
OSBHOST2> ./startNodeManager.sh
```

5 Server Migration

In this enterprise topology, you must configure server migration for the WLS_OSB1 and WLS_OSB2 managed servers. The WLS_OSB1 managed server is configured to restart on OSBHOST2 should a failure occur. The WLS_OSB2 managed server is configured to restart on OSBHOST1 should a failure occur. For this configuration, the WLS_OSB1 and WLS_OSB2 servers listen on specific floating IPs that are failed over by WLS Server Migration. Configuring server migration for the WLS_OSBn managed servers consists of the following steps:

- Step 1: Setting Up a User and Tablespace for the Server Migration Leasing Table
- Step 2: Creating a Multi-Data Source Using the Oracle WebLogic Server Administration Console
- Step 3: Enabling Host Name Verification Certificates between OSBHOST1 and OSBHOST2 and the Administration Server
- Step 4: Editing the Node Manager's Properties File
- Step 5: Setting Environment and Superuser Privileges for the wlsifconfig.sh Script
- Step 6: Configuring Server Migration Targets
- Step 7: Testing the Server Migration

5.1 Setting Up a User and Tablespace for the Server Migration Leasing Table

The first step is to set up a user and tablespace for the server migration leasing table:

15. Create a tablespace called `leasing`. For example, log on to SQL*Plus as the sysdba user and run the following command:

```
16.  SQL> create tablespace leasing
17.          logging datafile 'DB_HOME/oradata/orcl/leasing.dbf'
18.          size 32m autoextend on next 32m maxsize 2048m extent management
   local;
```

19. Create a user named `leasing` and assign to it the leasing tablespace.

```
20.  SQL> create user leasing identified by welcome1;
21.
22.  SQL> grant create table to leasing;
23.
24.  SQL> grant create session to leasing;
25.
26.  SQL> alter user leasing default tablespace leasing;
27.
28.  SQL> alter user leasing quota unlimited on LEASING;
```

29. Create the leasing table using the `leasing.ddl` script.
    1. Copy the `leasing.ddl` file located in either the `ORACLE_BASE/product/osbmw/wlserver_10.3/server/db/oracle/817` or the `ORACLE_BASE/product/osbmw/wlserver_10.3/server/db/oracle/920` directory to your database node.
    2. Connect to the database as the `leasing` user.
    3. Run the `leasing.ddl` script in SQL*Plus.

```
4.    SQL> @copy_location/leasing.ddl;
```

5.2 Creating a Multi-Data Source Using the Oracle WebLogic Server Administration Console

The second step is to create a multi-data source for the leasing table from the Oracle WebLogic Server Administration Console:

You create a data source to each of the RAC database instances during the process of setting up the multi-data source, both for these data sources and the global leasing multi-data source. When you create a data source:

- Make sure that this is a non-xa data source
- The names of the multi-data sources are in the format of <MultiDS>-rac0, <MultiDS>-rac1, and so on
- Use Oracle's Driver (Thin) Version 9.0.1, 9.2.0, 10, 11
- Use Supports Global Transactions, One-Phase Commit, and specify a service name for your database
- Target these data sources to the OSB cluster

Creating a Multi-Data Source

To create a multi-data source, complete these steps:

28. From Domain Structure window in the Oracle WebLogic Server Administration Console, expand the Services node, then expand the JDBC node.
29. Click Multi Data Sources. The Summary of JDBC Multi Data Source page appears.
30. Click Lock and Edit.
31. Click New. The Create a New JDBC Multi Data Source page appears.
32. Click New.
33. Enter leasing as the Name
34. Enter jdbc/leasing as the JNDI name.
35. Select Failover as algorithm (default).
36. Click Next.
37. Select OSB_Cluster as the target.
38. Click Next.
39. Select non-XA driver (the default).
40. Click Next.
41. Click Create New Data Source.
42. Enter leasing-rac0 as name. Enter jdbc/leasing-rac0 as JNDI name. Enter oracle as the database type. For the driver type, enter Oracle Driver (Thin) for RAC server-Instance connection Version 10,11.

    Note:

    When creating the multi-datasources for the leasing table, enter names in the format of <MultiDS>-rac0, <MultiDS>-rac1, and so on.

43. Click Next.
44. Deselect Supports Global Transactions.
45. Click Next.
46. Enter the service name, database name, host port, and password for your leasing schema.
47. Click Next.
48. Click Test Configuration and verify the connection works.
49. Target the data source to OSB_Cluster.
50. Select the data source and add it to the right screen.
51. Click Create a New Data Source for the first instance of your RAC database, target it to OSB_Cluster, and repeat the steps for the second instance of your RAC database.
52. Add the second data source to your multi-data source.
53. Click Activate Changes.

5.3 Enabling Host Name Verification Certificates between OSBHOST1 and OSBHOST2 and the Administration Server

The third step is to create the appropriate certificates for host name verification between the Node Manager and the Administration Server. This procedure is described in Section 4.2, "Enabling Host Name Verification Certificates for Node Manager in OSBHOST1" and Section 4.4, "Enabling Host Name Verification Certificates for the Node Manager in OSBHOST2."

5.4 Editing the Node Manager's Properties File

The fourth step is to edit the Node Manager's properties file. This file is called `nodemanager.properties` and is located in the `ORACLE_BASE/product/osbmw/wlserver_10.3/common/nodemanager` directory. For server migration to work properly, you need to add the properties listed below:

```
Interface=eth0

NetMask=255.255.255.0

UseMACBroadcast=true
```

- `Interface`

  This property specifies the interface name for the floating IP (for example, eth0).

- `NetMask`

  This property specifies the net mask for the interface for the floating IP. The net mask should the same as the net mask on the interface; 255.255.255.0 is used as an example in this document.

- `UseMACBroadcast`

  This property specifies whether or not to use a node's MAC address when sending ARP packets, that is, whether or not to use the -b flag in the `arping` command.

Verify in Node Manager's output (shell where Node Manager is started) that these properties are being used, or problems may arise during migration. You should see something like this in the Node Manager's output:

```
...
StateCheckInterval=500
Interface=eth0
NetMask=255.255.255.0
...
```

Note: The steps below are not required if the server properties (start properties) have been properly set and the Node Manager can start the servers remotely.

1. Set the following property in the `nodemanager.properties` file.
   - `StartScriptEnabled`

     Set this property to `true`. This is required for the shiphome to enable the Node Manager to start the managed servers.

2. Start the Node Manager on Node 1 and Node 2 by running the `startNodeManager.sh` script located in the `ORACLE_BASE/product/osbmw/wlserver_10.3//server/bin` directory.

Note: When running Node Manager from a shared storage installation, multiple nodes are started using the same `nodemanager.properties` file. However, each node may require different `NetMask` or `Interface` properties. In this case, specify individual parameters on a per-node basis using environment variables. For example, to use a different interface (eth3) in OSBHOSTn, use the `Interface` environment variable as follows: `OSBHOSTn> export JAVA_OPTIONS=-DInterface=eth3` and start Node Manager after the varibale has been set in the shell.

5.5 Setting Environment and Superuser Privileges for the wlsifconfig.sh Script

The fifth step is to set environment and superuser privileges for the `wlsifconfig.sh` script:

1. Ensure that your PATH environment variable includes these files:

   Table 8-1 Files Required for the PATH Environment Variable

File	**Located in this directory**
`wlsifconfig.sh`	`ORACLE_BASE/admin/<domain_name>/mserver/<domain_name>/bin/server_migration`
`wlscontrol.sh`	`ORACLE_BASE/product/osbmw/wlserver_10.3/common/bin`
`nodemanager.domains`	`ORACLE_BASE/product/osbmw/wlserver_10.3/common`

2. Grant sudo configuration for the `wlsifconfig.sh script`.
    - Configure sudo to work without a password prompt.
    - For security reasons, sudo should be restricted to the subset of commands required to run the `wlsifconfig.sh` script. For example, to set the environment and superuser privileges for the `wlsifconfig.sh` script, complete these steps:
        1. Grant sudo privilege to the WebLogic user ('oracle') with no password restriction, and grant execute privilege on the /sbin/ifconfig and /sbin/arping binaries.
        2. Make sure the script is executable by the WebLogic user ('oracle'). The following is an example of an entry inside /etc/sudoers granting sudo execution privilege for `oracle` and also over `ifconfig` and `arping`:
        3. `oracle ALL=NOPASSWD: /sbin/ifconfig,/sbin/arping`

Note:

Ask the system adminsitrator for the sudo and system rights as appropriate to this step.

5.6 Configuring Server Migration Targets

The sixth step is to configure server migration targets. Configuring Cluster Migration sets the `DataSourceForAutomaticMigration` property to true. Follow the steps below to configure cluster migration in a migration in a cluster:

1. Log into the Oracle WebLogic Server Administration Console (`http://<host>:<adminPort>/console`. Typically, `adminPort` is 7001 by default).
2. In the Domain Structure window, expand Environment and select Clusters. The Summary of Clusters page appears.
3. Click the cluster for which you want to configure migration (OSB_Cluster) in the Name column of the table.
4. Click the Migration tab.
5. In the Available field, select the machine to which to allow migration and click the right arrow. In this case, select OSBHOST1 and OSBHOST2.
6. Select the data source to be used for automatic migration. In this case select the leasing data source.
7. Click Save.
8. Set the Candidate Machines for Server Migration. You must perform this task for all of the managed servers as follows:
    1. In Domain Structure window of the Oracle WebLogic Server Administration Console, expand Environment and select Servers.
    2. Select the server for which you want to configure migration.
    3. Click the Migration tab.
    4. In the Available field, located in the Migration Configuration section, select the machines to which to allow migration and click the right arrow. For WLS_OSB1, select OSBHOST2. For WLS_OSB2, select OSBHOST1.
    5. Select Automatic Server Migration Enabled. This enables the Node Manager to start a failed server on the target node automatically.
    6. Click Save.
    7. Restart the Administration Server.

Tip:

Click Customize this table in the Summary of Servers page, move Current Machine from the Available Window to the Chosen window to view the machine on which the server is running. This will be different from the configuration if the server gets migrated automatically.

5.7 Testing the Server Migration

The seventh and final step is to test the server migration. To verify that Server Migration is working properly, follow these steps:

From Node 1:

1. Stop the WLS_OSB1 managed server.

   To do this, run this command:

   ```
   OSBHOST1> kill -9 <pid>
   ```

   pid specifies the process ID of the managed server. You can identify the pid in the node by running this command:

   ```
   OSBHOST1> ps -ef | grep WLS_OSB1
   ```

2. Watch the Node Manager console: you should see a message indicating that WLS_OSB1's floating IP has been disabled.
3. Wait for the Node Manager to try a second restart of WLS_OSB1. Node Manager waits for a fence period of 30 seconds before trying this restart.
4. Once Node Manager restarts the server, stop it again. Now Node Manager should log a message indicating that the server will not be restarted again locally.

From Node2:

1. Watch the local Node Manager console. After 30 seconds since the last try to restart WLS_OSB1on Node 1, Node Manager on Node 2 should prompt that the floating IP for WLS_OSB1 is being brought up and that the server is being restarted in this node.
2. Access the osb-infra console in the same IP.

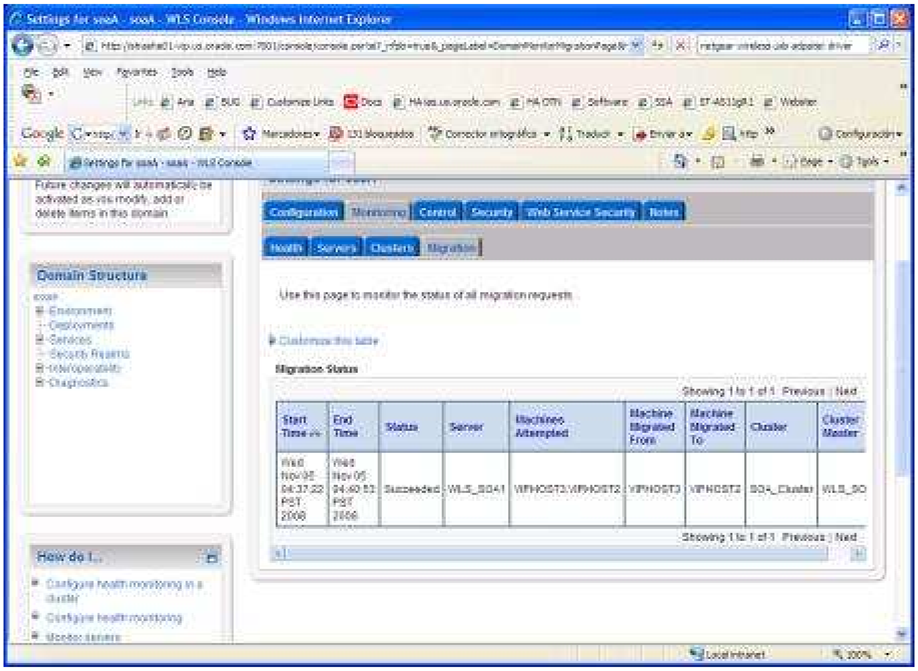
Verification From the Administration Console

Migration can also be verified in the Administration Console:

1. Log into the Administration Console.
2. Click on Domain on the left console.
3. Click the Monitoring tab and then the Migration subtab.

The Migration Status table provides information on the status of the migration.

Figure 5-1 Migration Status Screen in the Administration Console



Future changes will automatically be activated as you modify, add or delete items in this domain.

Use this page to monitor the status of all migration requests.

Customize this table

Migration Status

Showing 1 to 1 of 1 Previous | Next

Start Time	End Time	Status	Server	Machines Attempted	Machine Migrated From	Machine Migrated To	Cluster	Cluster Master
Wed Nov 05 04:37:22 PST 2008	Wed Nov 05 04:40:53 PST 2008	Succeeded	WLS_SOA1	VIPHOST3,VIPHOST2	VIPHOST3	VIPHOST2	SOA_Cluster	WLS_SO

Showing 1 to 1 of 1 Previous | Next

Description of "Figure 5-1 Migration Status Screen in the Administration Console"

Appendix A

A.1 Terminology

This section identifies terms used to describe components in prior releases, and the terms to which they correlate in 11g Release 1 (11.1.1).

- **Oracle home:** An Oracle home contains installed files necessary to host a specific product. For example, the OSB Oracle home contains a directory that contains binary and library files for Oracle Service Bus. An Oracle home resides within the directory structure of the Middleware home. Each Oracle home can be associated with multiple Oracle instances or Oracle WebLogic Server domains.
- **WebLogic Server home:** A WebLogic Server home contains installed files necessary to host a WebLogic Server. The WebLogic Server home directory is a peer of Oracle home directories and resides within the directory structure of the Middleware home.
- **Middleware home:** A Middleware home consists of the Oracle WebLogic Server home, and, optionally, one or more Oracle homes. A Middleware home can reside on a local file system or on a remote shared disk that is accessible through NFS.
- **Oracle instance:** An Oracle instance contains one or more active middleware system components, for example Oracle Web Cache, Oracle HTTP Server, or Oracle Internet Directory. You determine which components are part of an instance, either at install time or by creating and configuring an instance at a later time. An Oracle instance contains files that can be updated, such as configuration files, log files, temporary files.
- **failover:** When a member of a high availability system fails unexpectedly (unplanned downtime), in order to continue offering services to its consumers, the system undergoes a failover operation. If the system is an active-passive system, the passive member is activated during the failover operation and consumers are directed to it instead of the failed member. The failover process can be performed manually, or it can be automated by setting up hardware cluster services to detect failures and move cluster resources from the failed node to the standby node. If the system is an active-active system, the failover is performed by the load balancer entity serving requests to the active members. If an active member fails, the load balancer detects the failure and automatically redirects requests for the failed member to the surviving active members.
- **failback:** After a system undergoes a successful failover operation, the original failed member can be repaired over time and be re-introduced into the system as a standby member. If desired, a failback process can be initiated to activate this member and deactivate the other. This process reverts the system back to its pre-failure configuration.
- **shared storage:** Shared storage is the storage subsystem that is accessible by all the machines in the domain. Among other things, the following is located on the shared disk:
    - Middleware Home software
    - AdminServer Domain Home
    - JMS
    - Tlogs (where applicable)

  Managed Server homes can also be optionally located in the shared disk. The shared storage can be a Network Attached Storage (NAS), a Storage Area Network (SAN) or any other storage system that multiple nodes can access simultaneously and can read-write.

- **primary node:** The node that is actively running an Oracle Fusion Middleware instance at any given time and has been configured to have a backup/secondary node. If the primary node fails, Oracle Fusion Middleware instance is failed over to the secondary node. This failover can be manual or automated using the Clusterware for Admin Server. For a server migration based scenario, WebLogic Whole Server Migration is used for automated failover.
- **secondary node:** The node that is the backup node for an Oracle Fusion Middleware instance. This is where the active instance fails over when the primary node is no longer available. See the definition for primary node in this section.
- **Network host name:** Network host name is a name assigned to an IP address either through the `/etc/hosts` file or through DNS resolution. This name is visible in the network that the machine to which it refers to is connected. Often, the network host name and physical host name are identical. However, each machine has only one physical host name but may have multiple network host names. Thus, a machine's network host name may not always be its physical host name.
- **physical host name:** This guide differentiates between the terms physical host name and network host name. This guide uses physical host name to refer to the "internal name" of the current machine. On UNIX, this is the name returned by the `hostname` command.

  Physical host name is used by Oracle Fusion Middleware to reference the local host. During installation, the installer automatically retrieves the physical host name from the current machine and stores it in the Oracle Fusion Middleware configuration metadata on disk.

- **physical IP:** Physical IP refers to the IP of a machine on the network. In almost all cases, it is normally associated with the physical host name of the machine (see the definition of the physical host name). In contrast to a virtual IP, it is always associated with the same machine when on a network.
- **switchover:** During normal operation, active members of a system may require maintenance or upgrading. A switchover process can be initiated to allow a substitute member to take over the workload performed by the member that requires maintenance or upgrading, which undergoes planned downtime. The switchover operation ensures continued service to consumers of the system.
- **switchback:** When a switchover operation is performed, a member of the system is deactivated for maintenance or upgrading. When the maintenance or upgrading is completed, the system can undergo a switchback operation to activate the upgraded member and bring the system back to the pre-switchover configuration.
- **virtual host name:** Virtual host name is a network addressable host name that maps to one or more physical machines via a load balancer or a hardware cluster. For load balancers, the name "virtual server name" is used interchangeably with virtual host name in this book. A load balancer can hold a virtual host name on behalf of a set of servers, and clients communicate indirectly with the machines using the virtual host name. A virtual host name in a hardware cluster is a network host name assigned to a cluster virtual IP. Because the cluster virtual IP is not permanently attached to any particular node of a cluster, the virtual host name is not permanently attached to any particular node either.

  Note: Whenever the term "virtual host name" is used in this document, it is assumed to be associated with a virtual IP address. In cases where just the IP address is needed or used, it will be explicitly stated.

- **virtual IP:** Also, cluster virtual IP and load balancer virtual IP. Generally, a virtual IP can be assigned to a hardware cluster or load balancer. To present a single system view of a cluster to network clients, a virtual IP serves as an entry point IP address to the group of servers which are members of the cluster. A virtual IP can be assigned to a server load balancer or a hardware cluster.

  A hardware cluster uses a cluster virtual IP to present to the outside world the entry point into the cluster (it can also be set up on a standalone machine). The hardware cluster's software manages the movement of this IP address between the two physical nodes of the cluster while clients connect to this IP address without the need to know which physical node this IP address is currently active on. In a typical two-node hardware cluster configuration, each machine has its own physical IP address and physical host name, while there could be several cluster IP addresses. These cluster IP addresses float or migrate between the two nodes. The node with current ownership of a cluster IP address is active for that address.

  A load balancer also uses a virtual IP as the entry point to a set of servers. These servers tend to be active at the same time. This virtual IP address is not assigned to any individual server but to the load balancer which acts as a proxy between servers and their clients.

A.2 Shared Storage Configuration

The following steps show to create and mount shared storage locations so that OSBHOST1 and OSBHOST2 can see the same location for binary installation in two separate volumes.

"nasfiler" is the shared storage filer.

From OSBHOST1:

```
OSBHOST1> mount
nasfiler:/vol/vol1/u01/app/oracle/product/osbmw/u01/app/oracle/product/osbmw -t nfs
```

From OSBHOST2:

```
OSBHOST2> mount
nasfiler:/vol/vol2/u01/app/oracle/product/osbmw/u01/app/oracle/product/osbmw -t nfs
```

If only one volume is available, users can provide redundancy for the binaries by using two different directories in the shared storage and mounting them to the same dir in the OSB Servers:

From OSBHOST1:

```
OSBHOST1> mount nasfiler:/vol/vol1/u01/app/oracle/product/osbmw1
u01/app/oracle/product/osbmw -t nfs
```

From OSBHOST2:

```
OSBHOST2> mount
nasfiler:/vol/vol2/u01/app/oracle/product/osbmw2u01/app/oracle/product/osbmw -t nfs
```

The following commands show how to share the OSB TX logs location across different nodes:

```
OSBHOST1> mount
nasfiler:/vol/vol1/u01/app/oracle/stores/osbdomain/osb_cluster/tlogs/u01/app/oracle/s
tores/osbdomain/osb_cluster/tlogs -t nfs

OSBHOST2> nasfiler:/vol/vol1/u01/app/oracle/stores/osbdomain/osb_cluster/tlogs
/u01/app/oracle/stores/osbdomain/osb_cluster/tlogs -t nfs
```

Note:

The shared storage can be a NAS or SAN device. The following illustrates an example of creating storage for a NAS device from OSBHOST1. The options may differ.

```
OSBHOST1> mount nasfiler:/vol/vol1/osbmw11shared ORACLE_BASE/wls -t nfs -o
rw,bg,hard,nointr,tcp,vers=3,timeo=300,rsize=32768,wsize=32768
```

A.3 Load Balancer Requirements

This enterprise topology uses an external load balancer. This external load balancer should have the following features:

- Ability to load-balance traffic to a pool of real servers through a virtual host name: Clients access services using the virtual host name (instead of using actual host names). The load balancer can then load balance requests to the servers in the pool.
- Port translation configuration should be possible so that incoming requests on the virtual host name and port are directed to a different port on the backend servers.
- Monitoring of ports on the servers in the pool to determine availability of a service.
- Virtual servers and port configuration: Ability to configure virtual server names and ports on your external load balancer, and the virtual server names and ports must meet the following requirements:
    - The load balancer should allow configuration of multiple virtual servers. For each virtual server, the load balancer should allow configuration of traffic management on more than one port. For example, for Oracle HTTP Server in the web tier, the load balancer needs to be configured with a virtual server and ports for HTTP and HTTPS traffic.
    - The virtual server names must be associated with IP addresses and be part of your DNS. Clients must be able to access the external load balancer through the virtual server names.
- Ability to detect node failures and immediately stop routing traffic to the failed node.
- Fault-tolerant mode: It is highly recommended that you configure the load balancer to be in fault-tolerant mode.
- It is highly recommended that you configure the load balancer virtual server to return immediately to the calling client when the backend services to which it forwards traffic are unavailable. This is preferred over the client disconnecting on its own after a timeout based on the TCP/IP settings on the client machine.
- Sticky routing capability: Ability to maintain sticky connections to components. Examples of this include cookie-based persistence, IP-based persistence, and so on.

The load balancer should be able to terminate SSL requests at the load balancer and forward traffic to the backend real servers using the equivalent non-SSL protocol (for example, HTTPS to HTTP). Typically, this feature is called SSL acceleration and it is required.

Note: The Oracle Technology Network (`http://www.oracle.com/technology/index.html`) provides a list of validated load balancers and their configuration at `http://www.oracle.com/technology/products/ias/hi_av/Tested_LBR_FW_SSLAccel.html`.

A.4 Unicast Requirement

Oracle recommends that the nodes in the recommended topology communicate using unicast. Unlike multicast communication, unicast does not require cross-network configuration and it reduces potential network errors that can occur from multicast address conflicts as well.

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

- All members of a WebLogic cluster must use the same message type. Mixing between multicast and unicast messaging is not allowed.
- Individual cluster members cannot override the cluster messaging type.
- The entire cluster must be shut down and restarted to change the message modes (from unicast to multicast or from multicast to unicast).
- JMS topics configured for multicasting can access WebLogic clusters configured for unicast because a JMS topic publishes messages on its own multicast address that is independent of the cluster address. However, the following considerations apply:
    - The router hardware configurations that allow unicast clusters may not allow JMS multicast subscribers to work.
    - JMS multicast subscribers need to be in a network hardware configuration that allows multicast accessibility. (That is, JMS subscribers must be in a muticast-enabled network to access multicast topics.)

A.5 OSB References

1. Understanding Oracle Service Bus High Availability

2. Understanding Oracle Service Bus Clusters

3. WebLogic Whole Server Migration