Oracle® Fusion Middleware

Enterprise Deployment Guide for Oracle WebCenter Content

Release 12.2.1.3
E95487-04
June 2021

ORACLE

Oracle Fusion Middleware Enterprise Deployment Guide for Oracle WebCenter Content, Release 12.2.1.3

E95487-04

Primary Author: Oracle Corporation

Contents

12 Extending the Domain to Include Oracle WebCenter Content

13 Extending the Domain with Oracle SOA Suite

14 Extending the Domain to Include Inbound Refinery

15 Extending the Domain to Include Capture

16 Extending the Domain to Include Imaging

17 Extending the Domain to Include WebCenter Content User Interface

Part IV Common Configuration and Management Procedures for an Enterprise Deployment

18 Common Configuration and Management Tasks for an Enterprise Deployment

19 Using Whole Server Migration and Service Migration in an Enterprise Deployment

20 Configuring Single Sign-On for an Enterprise Deployment

A Using Multi Data Sources with Oracle RAC

Preface

This guide explains how to install, configure, and manage a highly available Oracle Fusion Middleware enterprise deployment..

Audience

In general, this document is intended for administrators of Oracle Fusion Middleware, who are assigned the task of installing and configuring Oracle Fusion Middleware software for production deployments.

Specific tasks can also be assigned to specialized administrators, such as database administrators (DBAs) and network administrators, where applicable.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc`.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` or visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` if you are hearing impaired.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Note:

This guide focuses on the implementation of the enterprise deployment reference topology on Oracle Linux systems.

The topology can be implemented on any certified, supported operating system, but the examples in this guide typically show the commands and configuration steps as they should be performed using the bash shell on Oracle Linux.

Part I
Understanding an Enterprise Deployment

It is important to understand the concept and general characteristics of a typical enterprise deployment, before you configure the Oracle WebCenter Content enterprise deployment topology.

This part of the Enterprise Deployment Guide contains the following topics.

- Enterprise Deployment Overview
  The Enterprise Deployment Guide provides detailed, validated instructions that help you plan, prepare, install, and configure a multi-host, secure, highly available, production topology for selected Oracle Fusion Middleware products.
- About a Typical Enterprise Deployment
  It is essential to understand the components of a typical enterprise deployment topology.
- Understanding the WebCenter Content Enterprise Deployment Topology
  The Oracle WebCenter Content enterprise deployment topology represents a specific reference implementation of the concepts described in Understanding a Typical Enterprise Deployment.

1

Enterprise Deployment Overview

The Enterprise Deployment Guide provides detailed, validated instructions that help you plan, prepare, install, and configure a multi-host, secure, highly available, production topology for selected Oracle Fusion Middleware products.

This chapter introduces the concept of an Oracle Fusion Middleware enterprise deployment. It also provides information on when to use the Enterprise Deployment guide.

- About the Enterprise Deployment Guide
  An Enterprise Deployment Guide provides a comprehensive, scalable example for installing, configuring, and maintaining a secure, highly available, production-quality deployment of selected Oracle Fusion Middleware products. The resulting environment is known as an **enterprise deployment topology**.
- When to Use the Enterprise Deployment Guide
  This guide describes one of the three primary installation and configuration options for Oracle Fusion Middleware. Use this guide to help you plan, prepare, install, and configure a multi-host, secure, highly available, production topology for selected Oracle Fusion Middleware products.

About the Enterprise Deployment Guide

An Enterprise Deployment Guide provides a comprehensive, scalable example for installing, configuring, and maintaining a secure, highly available, production-quality deployment of selected Oracle Fusion Middleware products. The resulting environment is known as an **enterprise deployment topology**.

For example, the enterprise deployment topology introduces key concepts and best practices that you can use to implement a similar Oracle Fusion Middleware environment for your organization.

Each Enterprise Deployment Guide provides detailed, validated instructions for implementing the reference topology. Along the way, the guide also offers links to supporting documentation that explains concepts, reference material, and additional options for an Oracle Fusion Middleware enterprise deployment.

Note that the enterprise deployment topologies described in the enterprise deployment guides cannot meet the exact requirements of all Oracle customers. In some cases, you can consider alternatives to specific procedures in this guide, depending on whether the variations to the topology are documented and supported by Oracle.

Oracle recommends customers use the Enterprise Deployment Guides as a first option for deployment. If variations are required, then those variations should be verified by reviewing the related Oracle documentation or by working with Oracle Support.

When to Use the Enterprise Deployment Guide

This guide describes one of the three primary installation and configuration options for Oracle Fusion Middleware. Use this guide to help you plan, prepare, install, and configure a multi-host, secure, highly available, production topology for selected Oracle Fusion Middleware products.

Alternatively, you can use the other primary installation and configuration options:

- Review *Planning an Installation of Oracle Fusion Middleware*, which provides additional information to help you prepare for any Oracle Fusion Middleware installation.

2

About a Typical Enterprise Deployment

It is essential to understand the components of a typical enterprise deployment topology.

This chapter provides information on the Enterprise Deployment Topology diagram.

- Diagram of a Typical Enterprise Deployment
  This diagram shows all the components of a typical enterprise deployment, including the Web tier, Application tier, and Data tier. All enterprise deployments are based on these basic principles.
- About the Typical Enterprise Deployment Topology Diagram
  A typical enterprise deployment topology consists of a Hardware Load Balancer (LBR), web tier, an application tier, and data tier. This section provides detailed information on these components.

Diagram of a Typical Enterprise Deployment

This diagram shows all the components of a typical enterprise deployment, including the Web tier, Application tier, and Data tier. All enterprise deployments are based on these basic principles.

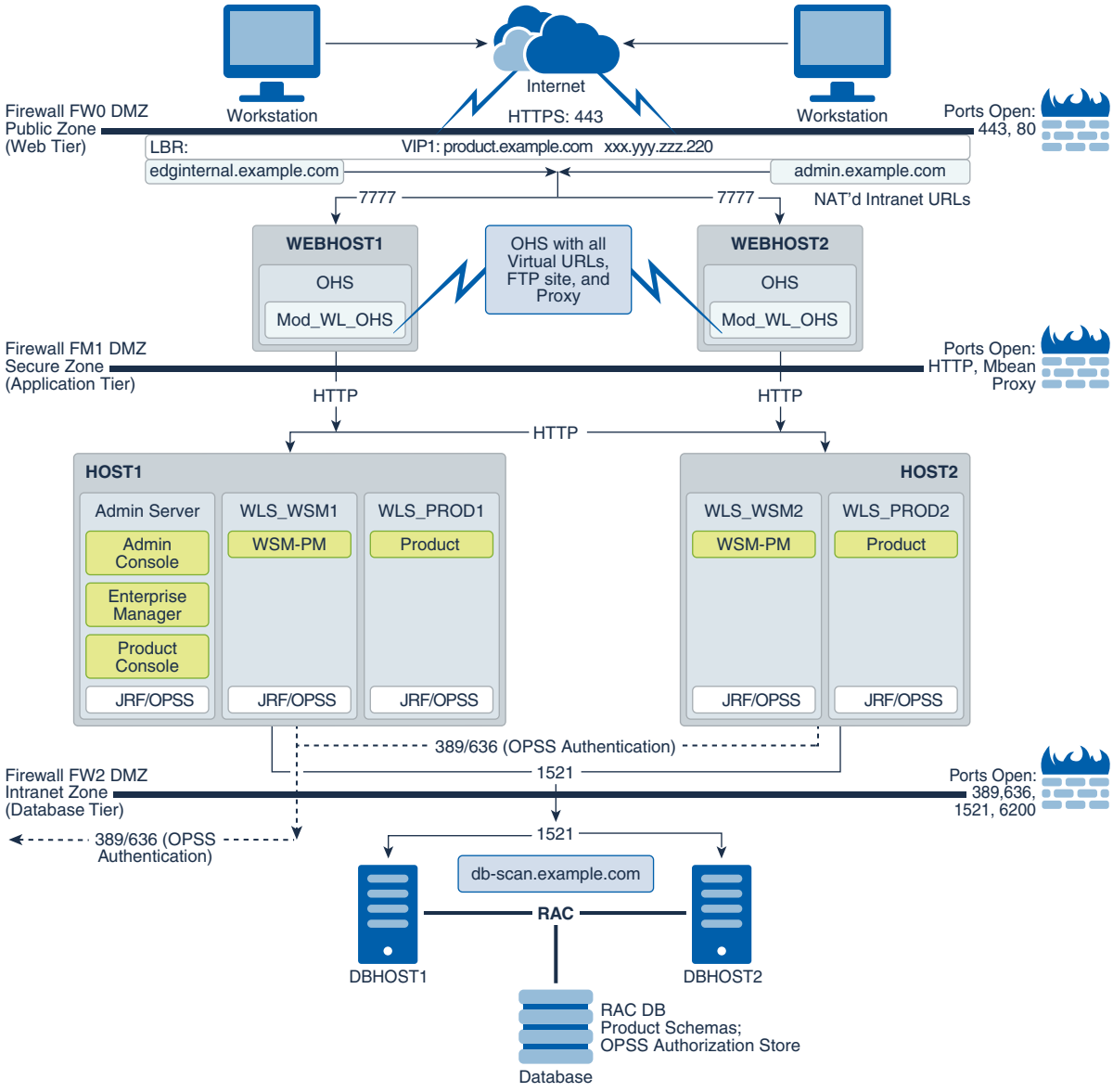
All Oracle Fusion Middleware enterprise deployments are designed to demonstrate the best practices for installing and configuring an Oracle Fusion Middleware production environment.

A best practices approach starts with the basic concept of a multi-tiered deployment and standard communications between the different software tiers.

Figure 2-1 shows a typical enterprise deployment, including the Web tier, Application tier, and Data tier. All enterprise deployments are based on these basic principles.

For a description of each tier and the standard protocols used for communications within a typical Oracle Fusion Middleware enterprise deployment, see About the typical Enterprise Deployment Topology Diagram.

Figure 2-1 Typical Enterprise Deployment Topology Diagram



About the Typical Enterprise Deployment Topology Diagram

A typical enterprise deployment topology consists of a Hardware Load Balancer (LBR), web tier, an application tier, and data tier. This section provides detailed information on these components.

- Understanding the Firewalls and Zones of a Typical Enterprise Deployment
- Understanding the Elements of a Typical Enterprise Deployment Topology
- Receiving Requests Through Hardware Load Balancer
- Understanding the Web Tier
- Understanding the Application Tier

- About the Data Tier

Understanding the Firewalls and Zones of a Typical Enterprise Deployment

The topology is divided into several security zones, which are separated by firewalls:

- The web tier (or DMZ), which is used for the hardware load balancer and Web servers (in this case, Oracle HTTP Server instances) that receive the initial requests from users. This zone is accessible only through a single virtual server name that is defined on the load balancer.
- The application tier, which is where the business and application logic resides.
- The data tier, which is not accessible from the Internet and reserved in this topology for the highly available database instances.

The firewalls are configured to allow data to be transferred only through specific communication ports. Those ports (or in some cases, the protocols that need open ports in the firewall) are shown on each firewall line in the diagram.

For example:

- On the firewall protecting the web tier, only the HTTP ports are open: 443 for HTTPS and 80 for HTTP.
- On the firewall protecting an application tier, HTTP ports, and MBean proxy port are open.

  Applications that require external HTTP access can use the Oracle HTTP Server instances as a proxy. Note that this port for outbound communications only and the proxy capabilities on the Oracle HTTP Server must be enabled.
- On the firewall protecting the data tier, the database listener port (typically, 1521) must be open.

  The LDAP ports (typically, 389 and 636) are also required to be open for communication between the authorization provider and the LDAP-based identity store.

  The ONS port (typically, 6200) is also required so that the application tier can receive notifications about workload and events in the Oracle RAC Database. These events are used by the Oracle WebLogic Server connection pools to adjust quickly (creating or destroying connections), depending on the availability and workload on the Oracle RAC database instances.

For a complete list of the ports that you must open for a specific Oracle Fusion Middleware enterprise deployment topology, see the chapter that describes the topology that you want to implement, or refer to the *Enterprise Deployment Workbook* for the topology that you want to implement. See Using the Enterprise Deployment Workbook.

Understanding the Elements of a Typical Enterprise Deployment Topology

The enterprise deployment topology consists of the following high-level elements:

- A hardware load balancer that routes requests from the Internet to the web servers in the web tier. It also routes requests from internal clients or other components that perform internal invocations within the corporate network.
- A web tier, consisting of a hardware load balancer and two or more physical computers that host the web server instances (for high availability).

  The web server instances are configured to authenticate users (through an external identity store and a single sign-on server) and then route the HTTP requests to the Oracle Fusion Middleware products and components that are running in the Application tier.

  The web server instances also host static web content that does not require the application logic to be delivered. Placing such content in the web tier reduces the overhead on the application servers and eliminates unnecessary network activity.
- An application tier, consisting of two or more physical computers that are hosting a cluster of Oracle WebLogic Server Managed Servers, and the Administration Server for the domain. The Managed Servers are configured to run the various Oracle Fusion Middleware products, such as Oracle SOA Suite, Oracle Service Bus, Oracle WebCenter Content, and Oracle WebCenter Portal, depending on your choice of products in the enterprise deployment.
- A data tier, consisting of two or more physical hosts that are hosting an Oracle RAC Database.

Receiving Requests Through Hardware Load Balancer

The following topics describe the hardware load balancer and its role in an enterprise deployment.

- Purpose of the Hardware Load Balancer (LBR)
- Summary of the Typical Load Balancer Virtual Server Names
- HTTPS Versus HTTP Requests to the External Virtual Server Name

Purpose of the Hardware Load Balancer (LBR)

There are two types of load balancers, Local Load Balancers and Global Load Balancers. Load balancers can either be hardware devices such as Big IP, Cisco, Brocade, and so on—or they can be software applications such as Oracle Traffic Director. Most load balancer appliances can be configured for both local and global load balancers.

Load balancers should always be deployed in pairs to ensure that no single load balancer is a single point of failure. Most load balancers do this in an active-passive way. You should consult your load balancer documentation on how best to achieve this.

> **Note:**
>
> Oracle does not certify against specific load balancers. The configuration information of load balancers given in the Enterprise Deployment guide are for guidance only and you should consult with your load balancer vendor about the best practices that are associated with the configuration of the device that you are using.

A local load balancer is used to distribute traffic within a site. It can distribute both HTTP and TCP traffic and the requirements of your deployment dictates which options you should use. Local load balancers often provide acceleration for SSL encryption and decryption as well as the ability to terminate or `off-load` SSL requests. SSL termination at the load balancer provides a significant performance gain to applications, ensuring that traffic to and from a site remains encrypted without the overhead of on the fly software encryption inside the deployment itself. Enterprise Deployment guide environments always utilize a local load balancer.

A global load balancer is used when you have multiple sites that need to function as the same logical environment. Its purpose is to distribute requests between the sites based on a pre-determined set of rules. Global load balancers are typically used in Disaster Recovery (DR) deployments or Active/Active Multi-Data Center (MDC) deployments.

The following topics describe the types of requests that are handled by the hardware load balancer in an Enterprise Deployment:

- HTTP Requests From the Internet to the Web Server Instances in the Web Tier
- Specific Internal-Only Communications Between the Components of the Application Tier
- Load Balancer Considerations for Disaster Recovery and Multi-Data Center Topologies

HTTP Requests From the Internet to the Web Server Instances in the Web Tier

The hardware load balancer balances the load on the web tier by receiving requests to a single virtual host name and then routing each request to one of the web server instances, based on a load balancing algorithm. In this way, the load balancer ensures that no one web server is overloaded with HTTP requests.

For more information about the purpose of specific virtual host names on the hardware load balancer, see Summary of the Typical Load Balancer Virtual Server Names.

Note that in the reference topology, only HTTP requests are routed from the hardware load balancer to the web tier. Secure Socket Layer (SSL) requests are terminated at the load balancer and only HTTP requests are forwarded to the Oracle HTTP Server instances. This guide does not provide instructions for SSL configuration between the load balancer and the Oracle HTTP Server instances or between the web tier and the application tier.

The load balancer provides high availability by ensuring that if one web server goes down, requests are routed to the remaining web servers that are up and running.

Further, in a typical highly available configuration, the hardware load balancers are configured such that a hot standby device is ready to resume service in case a failure occurs in the main load balancing appliance. This is important because for many types of services and systems, the hardware load balancer becomes the unique point of access to make invocations and, as a result, becomes a single point of failure (SPOF) for the whole system if it is not protected.

Specific Internal-Only Communications Between the Components of the Application Tier

In addition, the hardware load balancer routes specific communications between the Oracle Fusion Middleware components and applications on the application tier. The

internal-only requests are also routed through the load balancer by using a unique virtual host name.

Load Balancer Considerations for Disaster Recovery and Multi-Data Center Topologies

In addition to the load-balancing features for local site traffic as described in the previous topics, many LBR also include features for configuring global load-balancing across multiple sites in DR or active/active MDC topologies.

A global load balancer configuration uses conditional DNS to direct traffic to local load balancers at different sites. A global load balancer for Oracle Fusion Middleware is typically configured for DR or MDC topologies:

- Active/Passive DR: Always send requests to site 1 unless site 1 in unavailable in which case send traffic to site 2.
- Active/Active MDC: Always send requests to both site 1 and site 2, often based on the geographic location of the source request in relation to the physical geographical location of the sites. Active/Active deployments are available only to those applications which support it.

For example:

```
Application entry point:  app.example.com

Site 1 - Local Load Balancer Virtual Host:  site1app.example.com
Site 2 - Local Load Balancer Virtual Host:  site2app.example.com
```

When a request for `app.example.com` is received, the global load balancer would:

- If the topology is active/passive DR:

  Change the IP address of `app.example.com` in DNS to resolve as the IP address of the local load balancer Virtual Host for the active site. For example: `site1app.example.com` (assuming that is the active site).

- If the topology is active/active MDC:

  Change the IP address of `app.example.com` in DNS to resolve as either the IP address of `site1app.example.com` or `site2app.example.com` depending on which site is nearest to the client making the request.

For information on Disaster Recovery, see *Oracle Fusion Middleware Disaster Recovery Guide*.

For more information on Multi-Data Center topologies for various Fusion Middleware products, see the MAA Best Practices for Fusion Middleware page on the Oracle Technology Network website.

Summary of the Typical Load Balancer Virtual Server Names

In order to balance the load on servers and to provide high availability, the hardware load balancer is configured to recognize a set of virtual server names. By using the naming convention in Figure 2-1, the following virtual server names are recognized by the hardware load balancer in this topology:

- `product.example.com`: This virtual server name is used for all incoming traffic.

Users enter this URL to access the Oracle Fusion Middleware product that you have deployed and the custom applications that are available on this server. The load balancer then routes these requests (by using a load balancing algorithm) to one of the servers in the web tier. In this way, the single virtual server name can be used to route traffic to multiple servers for load balancing and high availability of the web servers instances.

- `productinternal.example.com`: This virtual server name is for internal communications only.

  The load balancer uses its **Network Address Translation (NAT)** capabilities to route any internal communication from the application tier components that are directed to this URL. This URL is not exposed to external customers or users on the Internet. Each product has specific uses for the internal URL, so in the deployment instructions, the virtual server name is prefixed with the product name.

- `admin.example.com`: This virtual server name is for administrators who need to access the Oracle Enterprise Manager Fusion Middleware Control and Oracle WebLogic Server Administration Console interfaces.

  This URL is known only to internal administrators. It also uses the NAT capabilities of the load balancer to route administrators to the active Administration Server in the domain.

For a complete set of virtual server names that you must define for your topology, see the chapter that describes the product-specific topology.

HTTPS Versus HTTP Requests to the External Virtual Server Name

Note that when you configure the hardware load balancer, a best practice is to assign the main external URL (for example, `http://myapplication.example.com`) to port 80 and port 443.

Any request on port 80 (non-SSL protocol) should be redirected to port 443 (SSL protocol). Exceptions to this rule include requests from public WSDLs. See Configuring Virtual Hosts on the Hardware Load Balancer.

Understanding the Web Tier

The Web tier of the reference topology consists of the Web servers that receive requests from the load balancer. In the typical enterprise deployment, at least two Oracle HTTP Server instances are configured in the Web tier. The following topics provide more detail.

- Benefits of Using Oracle HTTP Server Instances to Route Requests
- Configuration of Oracle HTTP Server in the Web Tier
- About Mod_WL_OHS

Benefits of Using Oracle HTTP Server Instances to Route Requests

A Web tier with Oracle HTTP Server is not a requirement for many of the Oracle Fusion Middleware products. You can route traffic directly from the hardware load balancer to the WLS servers in the Application Tier. However, a Web tier does provide several advantages, which is why it is recommended as part of the reference topology:

- The Web tier provides DMZ public zone, which is a common requirement in security audits. If a load balancer routes directly to the WebLogic Server, requests move from the load balancer to the application tier in one single HTTP jump, which can cause security concerns.
- The Web tier allows the WebLogic Server cluster membership to be reconfigured (new servers added, others removed) without having to change the Web server configuration (as long as at least some of the servers in the configured list remain alive).
- Oracle HTTP Server delivers static content more efficiently and faster than WebLogic Server; it also provides FTP services, which are required for some enterprise deployments, as well as the ability to create virtual hosts and proxies via the Oracle HTTP Server configuration files.
- Oracle HTTP Server provides HTTP redirection over and above what WebLogic Server provides. You can use Oracle HTTP Server as a front end against many different WebLogic Server clusters, and in some cases, control the routing via content based routing.
- Oracle HTTP Server provides the ability to integrate single sign-on capabilities into your enterprise deployment. For example, you can later implement single sign-on for the enterprise deployment, using Oracle Access Manager, which is part of the Oracle Identity and Access Management family of products.
- Oracle HTTP Server provides support for WebSocket connections deployed within WebLogic Server.

For more information about Oracle HTTP Server, see Introduction to Oracle HTTP Server in *Oracle Fusion Middleware Administering Oracle HTTP Server*.

Configuration of Oracle HTTP Server in the Web Tier

Starting with Oracle Fusion Middleware 12*c*, the Oracle HTTP Server software can be configured in one of two ways: as part of an existing Oracle WebLogic Server domain or in its own standalone domain. Each configuration offers specific benefits.

When you configure Oracle HTTP Server instances as part of an existing WebLogic Server domain, you can manage the Oracle HTTP Server instances, including the wiring of communications between the web servers and the Oracle WebLogic Server Managed Servers by using Oracle Enterprise Manager Fusion Middleware Control. When you configure Oracle HTTP Server in a standalone configuration, you can configure and manage the Oracle HTTP Server instances independently of the application tier domains.

For this enterprise deployment guide, the Oracle HTTP Server instances are configured as separate standalone domains, one on each Web tier host. You can choose to configure the Oracle HTTP Server instances as part of the application tier domain, but this enterprise deployment guide does not provide specific steps to configure the Oracle HTTP Server instances in that manner.

See About Oracle HTTP Server in *Oracle Fusion Middleware Installing and Configuring Oracle HTTP Server*.

About Mod_WL_OHS

As shown in the diagram, the Oracle HTTP Server instances use the WebLogic Proxy Plug-In (`mod_wl_ohs`) for proxying HTTP requests from Oracle HTTP Server to the Oracle WebLogic Server Managed Servers in the Application tier.

See What are Oracle WebLogic Server Proxy Plug-Ins? in *Oracle Fusion Middleware Using Oracle WebLogic Server Proxy Plug-Ins 12.2.1.1*.

Understanding the Application Tier

The application tier consists of two physical host computers, where Oracle WebLogic Server and the Oracle Fusion Middleware products are installed and configured. The application tier computers reside in the secured zone between firewall 1 and firewall 2.

The following topics provide more information:

- Configuration of the Administration Server and Managed Servers Domain Directories
- About the Node Manager Configuration in a Typical Enterprise Deployment
- About Using Unicast for Communications within the Application Tier
- Understanding OPSS and Requests to the Authentication and Authorization Stores

Configuration of the Administration Server and Managed Servers Domain Directories

Unlike the Managed Servers in the domain, the Administration Server uses an active-passive high availability configuration. This is because only one Administration Server can be running within an Oracle WebLogic Server domain.

In the topology diagrams, the Administration Server on HOST1 is in the active state and the Administration Server on HOST2 is in the passive (inactive) state.

To support the manual fail over of the Administration Server in the event of a system failure, the typical enterprise deployment topology includes:

- A Virtual IP Address (VIP) for the routing of Administration Server requests.
- The configuration of the Administration Server domain directory on a shared storage device.

In the event of a system failure (for example a failure of HOST1), you can manually reassign the Administration Server VIP address to another host in the domain, mount the Administration Server domain directory on the new host, and then start the Administration Server on the new host.

However, unlike the Administration Server, there is no benefit to storing the Managed Servers on shared storage. In fact, there is a potential performance impact when Managed Server configuration data is not stored on the local disk of the host computer.

As a result, in the typical enterprise deployment, after you configure the Administration Server domain on shared storage, a copy of the domain configuration is placed on the local storage device of each host computer, and the Managed Servers are started

from this copy of the domain configuration. You create this copy by using the Oracle WebLogic Server pack and unpack utilities.

The resulting configuration consists of separate domain directories on each host: one for the Administration Server (on shared storage) and one for the Managed Servers (on local storage). Depending upon the action required, you must perform configuration tasks from one domain directory or the other.

For more information about structure of the Administration Server domain directory and the Managed Server domain directory, as well as the variables used to reference these directories, see Understanding the Recommended Directory Structure for an Enterprise Deployment.

There is an additional benefit to the multiple domain directory model. It allows you to isolate the Administration Server from the Managed Servers. By default, the primary enterprise deployment topology assumes the Administration Server domain directory is on one of the application tier hosts, but if necessary, you could isolate the Administration Server further by running it from its own host, for example in cases where the Administration Server is consuming high CPU or RAM. Some administrators prefer to configure the Administration Server on a separate, dedicated host, and the multiple domain directory model makes that possible.

About the Node Manager Configuration in a Typical Enterprise Deployment

Starting with Oracle Fusion Middleware 12*c*, you can use either a per domain Node Manager or a per host Node Manager. The following sections of this topic provide more information on the impact of the Node Manager configuration on a typical enterprise deployment.

> **Note:**
>
> For general information about these two types of Node Managers, see Overview in *Oracle Fusion Middleware Administering Node Manager for Oracle WebLogic Server*.

About Using a Per Domain Node Manager Configuration

In a per domain Node Manager configuration—as opposed to a per host Node Manager configuration—you actually start two Node Manager instances on the Administration Server host: one from the Administration Server domain directory and one from the Managed Servers domain directory. In addition, a separate Node Manager instance runs on each of the other hosts in the topology.

The Node Manager that controls the Administration Server uses the listen address of the virtual host name created for the Administration Server. The Node Manager that controls the Managed Servers uses the listen address of the physical host. When the Administration Server fails over to another host, an additional instance of Node Manager is started to control the Administration Server on the failover host.

The key advantages of the per domain configuration are an easier and simpler initial setup of the Node Manager and the ability to set Node Manager properties that are unique to the Administration Server. This last feature was important in previous releases because some features, such as Crash Recovery, applied only to

the Administration Server and not to the Managed servers. In the current release, the Oracle SOA Suite products can be configured for Automated Service Migration, rather than Whole Server Migration. This means the Managed Servers, as well as the Administration Server, can take advantage of Crash Recovery, so there is no need to apply different properties to the Administration Server and Managed Server domain directories.

Another advantage is that the per domain Node Manager provides a default SSL configuration for Node Manager-to-Server communication, based on the Demo Identity store created for each domain.

About Using a Per Host Node Manager Configuration

In a per host Node Manager configuration, you start a single Node Manager instance to control the Administration Server and all Managed Servers on a host, even those that reside in different domains. This reduces the footprint and resource utilization on the Administration Server host, especially in those cases where multiple domains coexist on the same computer.

A per host Node Manager configuration allows all Node Managers to use a listen address of ANY, so they listen on all addresses available on the host. This means that when the Administration Server fails over to a new host, no additional configuration is necessary. The per host configuration allows for simpler maintenance, because you can update and maintain a single Node Manager properties file on each host, rather than multiple node manager property files.

The per host Node Manager configuration requires additional configuration steps. If you want SSL for Node Manager-to-Server communication, then you must configure an additional Identity and Trust store, and it also requires using Subject Alternate Names (SAN), because the Node Manager listens on multiple addresses. Note that SSL communications are typically not required for the application tier, because it is protected by two firewalls.

About Using Unicast for Communications within the Application Tier

Oracle recommends the unicast communication protocol for communication between the Managed Servers and hosts within the Oracle WebLogic Server clusters in an enterprise deployment. 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.

When you consider using the multicast or unicast protocol for your own deployment, consider the type of network, the number of members in the cluster, and the reliability requirements for cluster membership. Also consider the following features of each protocol.

Features of unicast in an enterprise deployment:

- Uses a group leader that every server sends messages directly to. This leader is responsible for retransmitting the message to every other group member and other group leaders, if applicable.
- Works out of the box in most network topologies
- Requires no additional configuration, regardless of the network topology.
- Uses a single missed heartbeat to remove a server from the cluster membership list.

Features of multicast in an enterprise deployment:

- Multicast uses a more scalable peer-to-peer model, where a server sends each message directly to the network once and the network makes sure that each cluster member receives the message directly from the network.
- Works out of the box in most modern environments, where the cluster members are in a single subnet.
- Requires additional configuration in the routers and WebLogic Server (that is, Multicast TTL) if the cluster members span more than one subnet.
- Uses three consecutive missed heartbeats to remove a server from the cluster membership list.

Depending on the number of servers in your cluster and on whether the cluster membership is critical for the underlying application (for example, in session-replication intensive applications or clusters with intensive RMI invocations across the cluster), each model may act better.

Consider whether your topology is going to be part of an active-active disaster recovery system or if the cluster is going to traverse multiple subnets. In general, unicast acts better in those cases.

For more information about multicast and unicast communication types, see the following resources:

- Configuring Multicast Messaging for WebLogic Server Clusters in *Oracle Fusion Middleware High Availability Guide*
- One-to-Many Communication Using Unicast in *Oracle Fusion Middleware Administering Clusters for Oracle WebLogic Server*

Understanding OPSS and Requests to the Authentication and Authorization Stores

Many of the Oracle Fusion Middleware products and components require an Oracle Platform Security Services (OPSS) security store for authentication providers (an identity store), policies, credentials, keystores, and for audit data. As a result, communications must be enabled so the application tier can send requests to and from the security providers.

For authentication, this communication is to an LDAP directory, such as Oracle Internet Directory (OID) or Oracle Unified Directory (OUD), which typically communicates over port 389 or 636. When you configure an Oracle Fusion Middleware domain, the domain is configured by default to use the WebLogic Server Authentication provider. However, for an enterprise deployment, you must use a dedicated, centralized LDAP-compliant authentication provider.

For authorization (and the policy store), the location of the security store varies, depending upon the tier:

- For the application tier, the authorization store is database-based, so frequent connections from the Oracle WebLogic Server Managed Servers to the database are required for the purpose of retrieving the required OPSS data.
- For the web tier, the authorization store is file-based, so connections to the database are not required.

For more information about OPSS security stores, see the following sections of *Oracle Fusion Middleware Securing Applications with Oracle Platform Security Services*:

- Authentication Basics
- The Security Model

About the Data Tier

In the data tier, an Oracle RAC database runs on the two hosts (DBHOST1 and DBHOST2). The database contains the schemas required by the Oracle WebCenter Content components and the Oracle Platform Security Services (OPSS) policy store.

You can define multiple services for the different products and components in an enterprise deployment to isolate and prioritize throughput and performance accordingly. In this guide, one database service is used as an example. Furthermore, you can use other high availability database solutions to protect the database:

- Oracle Data Guard: See Introduction to Oracle Data Guard in *Oracle Data Guard Concepts and Administration*.
- Oracle RAC One Node: See Overview of Oracle RAC One Node in *Oracle Real Application Clusters Administration and Deployment Guide*.

These solutions above provide protection for the database beyond the information provided in this guide, which focuses on using an Oracle RAC Database, given the scalability and availability requirements that typically apply to an enterprise deployment.

For more information about using Oracle Databases in a high availability environment, see Database Considerations in *Oracle Fusion Middleware High Availability Guide*.

3

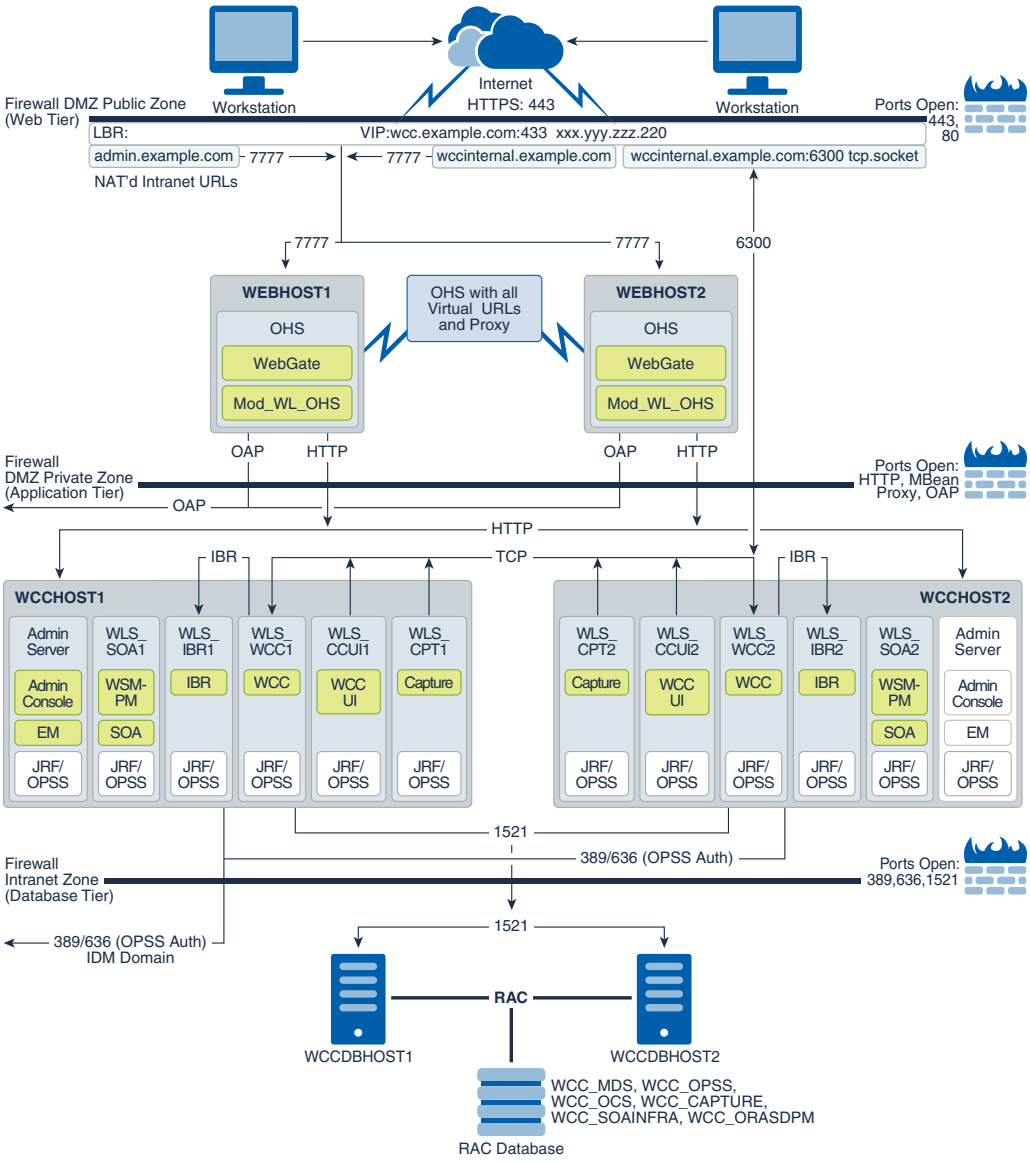
Understanding the WebCenter Content Enterprise Deployment Topology

The Oracle WebCenter Content enterprise deployment topology represents a specific reference implementation of the concepts described in Understanding a Typical Enterprise Deployment.

- Diagram of the WebCenter Content Enterprise Topology
  The below diagram shows the primary Oracle WebCenter Content enterprise deployment topologies.
- Understanding the WebCenter Content Enterprise Topology Diagram
  Although most of the elements of Oracle WebCenter Content topology represent standard features of any enterprise topology, there are some elements that are unique to the WebCenter Content topology.
- Flow Chart and Roadmap for Implementing the WebCenter Content Enterprise Topology
  It is essential to understand the steps that you need to perform to install and configure the WebCenter Content enterprise topology.

Diagram of the WebCenter Content Enterprise Topology

The below diagram shows the primary Oracle WebCenter Content enterprise deployment topologies.



Understanding the WebCenter Content Enterprise Topology Diagram

Although most of the elements of Oracle WebCenter Content topology represent standard features of any enterprise topology, there are some elements that are unique to the WebCenter Content topology.

Most of the elements of Oracle WebCenter Content topologies represent standard features of any enterprise topology that follows the Oracle-recommended best practices. These elements are described detail in Understanding a Typical Enterprise Deployment.

Before you review the information in the following topics, it is assumed you have reviewed the information in Understanding a Typical Enterprise Deployment and that you are familiar with the general concepts of an enterprise deployment topology.

- Summary of Oracle WebCenter Content Load Balancer Virtual Server Names
- Summary of the Managed Servers and Clusters on the WebCenter Content Application Tier

Summary of Oracle WebCenter Content Load Balancer Virtual Server Names

In order to balance the load on servers and to provide high availability, the hardware load balancer is configured to recognize a set of virtual server names.

For information about the purpose of each of these server names, see Summary of the Typical Load Balancer Virtual Server Names.

The following virtual server names are recognized by the hardware load balancer in Oracle WebCenter Content topologies:

- `wcc.example.com` - This virtual server name is used for all incoming traffic. It acts as the access point for all HTTP traffic to the runtime Oracle WebCenter Content components. The load balancer routes all requests to this virtual server name over SSL. As a result, clients access this service using the following secure address:

  ```
  wcc.example.com:443
  ```

- `wccinternal.example.com` - This virtual server name is for internal communications between the application tier components only and is not exposed to the Internet.

  The traffic from clients to this URL is not SSL-enabled. Clients access this service using the following address and the requests are forwarded to port 7777 on WEBHOST1 and WEBHOST2:

  ```
  wccinternal.example.com:80
  ```

  Note that this URL can also be set as the URL to be used for internal service invocations while modeling composites or at runtime with the appropriate Enterprise Manager MBeans.

  This virtual server name also acts as the access point for all internal Remote Intradoc Client (RIDC) TCP traffic to the runtime Oracle WebCenter Content components. Applications like Imaging and Capture access this service using the address `ucminternal.example.com:6300` for RIDC connections, and the requests are forwarded to port 4444 on WCCHOST1 and WCCHOST2.

- `admin.example.com` - This virtual server name is for administrators who need to access the Oracle Enterprise Manager Fusion Middleware Control and Oracle WebLogic Server Administration Console interfaces.

Information later in this guide provide instructions so you can:

- Configure the hardware load balancer to recognize and route requests to the virtual host names
- Configure the Oracle HTTP Server instances on the Web Tier to recognize and properly route requests to these virtual host names to the correct host computers.

Summary of the Managed Servers and Clusters on the WebCenter Content Application Tier

The Application tier hosts the Administration Server and Managed Servers in the Oracle WebLogic Server domain.

Depending upon the topology you select, the Oracle WebLogic Server domain for the Oracle WebCenter Content domain consists of the clusters shown in Table 3-1. These clusters function as active-active high availability configurations.

Table 3-1 Summary of the Clusters in the Oracle WebCenter Content Enterprise Deployment Topology

Cluster	Managed Servers
Oracle WebCenter Content Cluster	WLS_WCC1, WLS_WCC2
Oracle SOA Suite Cluster	WLS_SOA1, WLS_SOA2
Oracle Inbound Refinery Cluster	WLS_IBR1, WLS_IBR2
Oracle WebCenter Enterprise Capture Cluster	WLS_CPT1, WLS_CPT2
WebCenter Content user interface Cluster	WLS_WCCUI1, WLS_WCCUI2

Flow Chart and Roadmap for Implementing the WebCenter Content Enterprise Topology

It is essential to understand the steps that you need to perform to install and configure the WebCenter Content enterprise topology.

- Flow Chart of the Steps to Install and Configure the WebCenter Content Enterprise Topology
- Roadmap Table for Planning and Preparing for an Enterprise Deployment
- Roadmap Table for Configuring the Oracle WebCenter Content Topology

Flow Chart of the Steps to Install and Configure the WebCenter Content Enterprise Topology

Figure 3-1 shows a flow chart of the steps required to install and configure the primary enterprise deployment topologies described in this chapter. The sections following the flow chart explain each step in the flow chart.

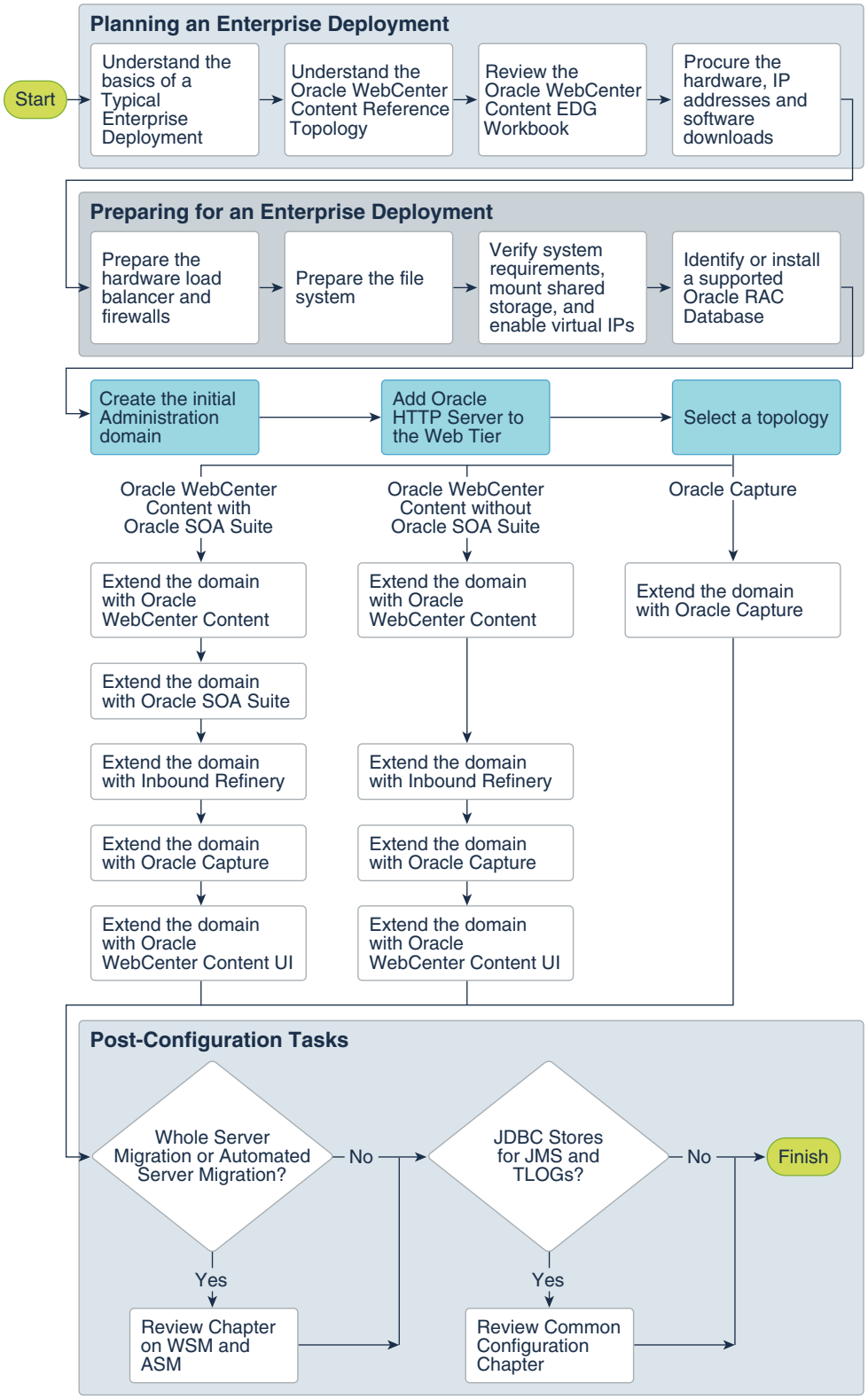
This guide is designed so you can start with a working WebCenter Content domain and then later extend the domain to add additional capabilities.

This modular approach to building the topology allows you to make strategic decisions, based on your hardware and software resources, as well as the Oracle WebCenter Content features that are most important to your organization.

It also allows you to validate and troubleshoot each individual product or component as they are configured.

This does not imply that configuring multiple products in one Configuration Wizard session is not supported; it is possible to group various extensions like the ones presented in this guide in one Configuration Wizard execution. However, the instructions in this guide focus primarily on the modular approach to building an enterprise deployment.

Figure 3-1 Flow Chart of the Enterprise Topology Configuration Steps



Roadmap Table for Planning and Preparing for an Enterprise Deployment

The following table describes each of the planning and preparing steps shown in the enterprise topology flow chart.

Flow Chart Step	More Information
Understand the basics of a Typical Enterprise Deployment	Understanding a Typical Enterprise Deployment
Understand the specific reference topology for the products that you plan to deploy	Review the product-specific topologies and the description of the topologies, including the virtual servers required and the summary of clusters and Managed Servers recommended for the product-specific deployment.
Review the Oracle WebCenter Content EDG Workbook	Using the Enterprise Deployment Workbook
Procure the hardware, IP addresses, and software downloads	Procuring Resources for an Enterprise Deployment
Prepare the hardware load balancer and firewalls	Preparing the Load Balancer and Firewalls for an Enterprise Deployment
Prepare the file system	Preparing the File System for an Enterprise Deployment
Verify system requirements, mount shared storage, and enable virtual IPs	Preparing the Host Computers for an Enterprise Deployment
Identify or install a supported Oracle RAC Database	Preparing the Database for an Enterprise Deployment

Roadmap Table for Configuring the Oracle WebCenter Content Topology

The following table describes each of the configuration steps required when configuring the topology shown in Diagram of the WebCenter Content Enterprise Deployment Topology.

These steps correspond to the steps shown in the flow chart.

Flow Chart Step	More Information
Create the initial Infrastructure domain	Creating the Initial Infrastructure Domain for an Enterprise Deployment
Extend the domain to Include the Web Tier	Configuring the Web Tier for an Enterprise Deployment
Extend the domain with Oracle WebCenter Content	Extending the Domain to Include Oracle WebCenter Content
Extend the domain with Oracle SOA Suite	Extending the Domain with Oracle SOA Suite
Extend the domain with Inbound Refinery	Extending the Domain to Include Inbound Refinery
Extend the domain with Oracle WebCenter Enterprise Capture	Extending the Domain to Include Capture
Extend the domain with WebCenter Content user interface	Extending the Domain to Include WebCenter Content User Interface
Integrate the Enterprise Deployment with Oracle Identity Management	Configuring Single Sign-On for an Enterprise Deployment

Part II
Preparing for an Enterprise Deployment

It is important to understand the tasks that need to be performed to prepare for an enterprise deployment.

This part of the enterprise deployment guide contains the following topics.

- Using the Enterprise Deployment Workbook
  The Enterprise Deployment workbook enables you to plan an enterprise deployment for your organization.
- Procuring Resources for an Enterprise Deployment
  It is essential to procure the required hardware, software, and network settings before you configure the Oracle WebCenter Content reference topology.
- Preparing the Load Balancer and Firewalls for an Enterprise Deployment
  It is important to understand how to configure the hardware load balancer and ports that must be opened on the firewalls for an enterprise deployment.
- Preparing the File System for an Enterprise Deployment
  Preparing the file system for an enterprise deployment involves understanding the requirements for local and shared storage, as well as the terminology that is used to reference important directories and file locations during the installation and configuration of the enterprise topology.
- Preparing the Host Computers for an Enterprise Deployment
  It is important to perform a set of tasks on each computer or server before you configure the enterprise deployment topology. This involves verifying the minimum hardware and operating system requirements for each host, configuring operating system users and groups, enabling Unicode support, mounting the required shared storage systems to the host and enabling the required virtual IP addresses on each host.
- Preparing the Database for an Enterprise Deployment
  Preparing the database for an enterprise deployment involves ensuring that the database meets specific requirements, creating database services, using SecureFiles for large objects in the database, and creating database backup strategies.

4

Using the Enterprise Deployment Workbook

The Enterprise Deployment workbook enables you to plan an enterprise deployment for your organization.

This chapter provides an introduction to the Enterprise Deployment workbook, use cases, and information on who should use the Enterprise Deployment workbook.

- Introduction to the Enterprise Deployment Workbook
  The Enterprise Deployment workbook is a spreadsheet that is used by architects, system engineers, database administrators, and others to plan and record all the details for an environment installation (such as server names, URLs, port numbers, installation paths, and other resources).
- Typical Use Case for Using the Workbook
  It is important to understand the roles and tasks involved in a typical use case of the Enterprise Deployment workbook.
- Using the Oracle WebCenter Content Enterprise Deployment Workbook
  Locating and understanding the Oracle WebCenter Content Enterprise Deployment workbook enables you to use it efficiently.
- Who Should Use the Enterprise Deployment Workbook?
  The details of the Enterprise Deployment workbook are filled in by the individual or a team that is responsible for planning, procuring, or setting up each category of resources.

Introduction to the Enterprise Deployment Workbook

The Enterprise Deployment workbook is a spreadsheet that is used by architects, system engineers, database administrators, and others to plan and record all the details for an environment installation (such as server names, URLs, port numbers, installation paths, and other resources).

The Enterprise Deployment workbook serves as a single document that you can use to track input variables for the entire process, allowing for:

- Separation of tasks between architects, system engineers, database administrators, and other key organizational roles.
- Comprehensive planning before the implementation.
- Validation of planned decisions before the actual implementation.
- Consistency during implementation.
- A record of the environment for future use.

Typical Use Case for Using the Workbook

It is important to understand the roles and tasks involved in a typical use case of the Enterprise Deployment workbook.

A typical use case for the Enterprise Deployment workbook involves the following roles and tasks, in preparation for an Oracle Fusion Middleware Enterprise Deployment:

- Architects read through the first five chapters of this guide, and fill in the corresponding sections of the workbook.
- The workbook is validated by other architects and system engineers.
- The architect uses the validated workbook to initiate network and system change requests with the system engineering departments.
- The Administrators and System Integrators who install and configure the software refer to the workbook and the subsequent chapters of this guide to perform the installation and configuration tasks.

Using the Oracle WebCenter Content Enterprise Deployment Workbook

Locating and understanding the Oracle WebCenter Content Enterprise Deployment workbook enables you to use it efficiently.

The following sections provide an introduction to the location and contents of the Oracle WebCenter Content Enterprise Deployment workbook:

- Locating the Oracle WebCenter Content Enterprise Deployment Workbook
- Understanding the Contents of the Oracle WebCenter Content Enterprise Deployment Workbook

Locating the Oracle WebCenter Content Enterprise Deployment Workbook

The Oracle WebCenter Content Enterprise Deployment workbook is available as a Microsoft Excel spreadsheet in the Oracle Fusion Middleware documentation library. It is available as a link on the Install, Patch, and Upgrade page of the library.

Understanding the Contents of the Oracle WebCenter Content Enterprise Deployment Workbook

The following sections describe the contents of the Oracle WebCenter Content Enterprise Deployment workbook. The workbook is divided into tabs, each containing a set of related variables and values that you need to install and configure the Enterprise Deployment topologies.

- Using the Start Tab
- Using the Hardware - Host Computers Tab
- Using the Network - Virtual Hosts & Ports Tab

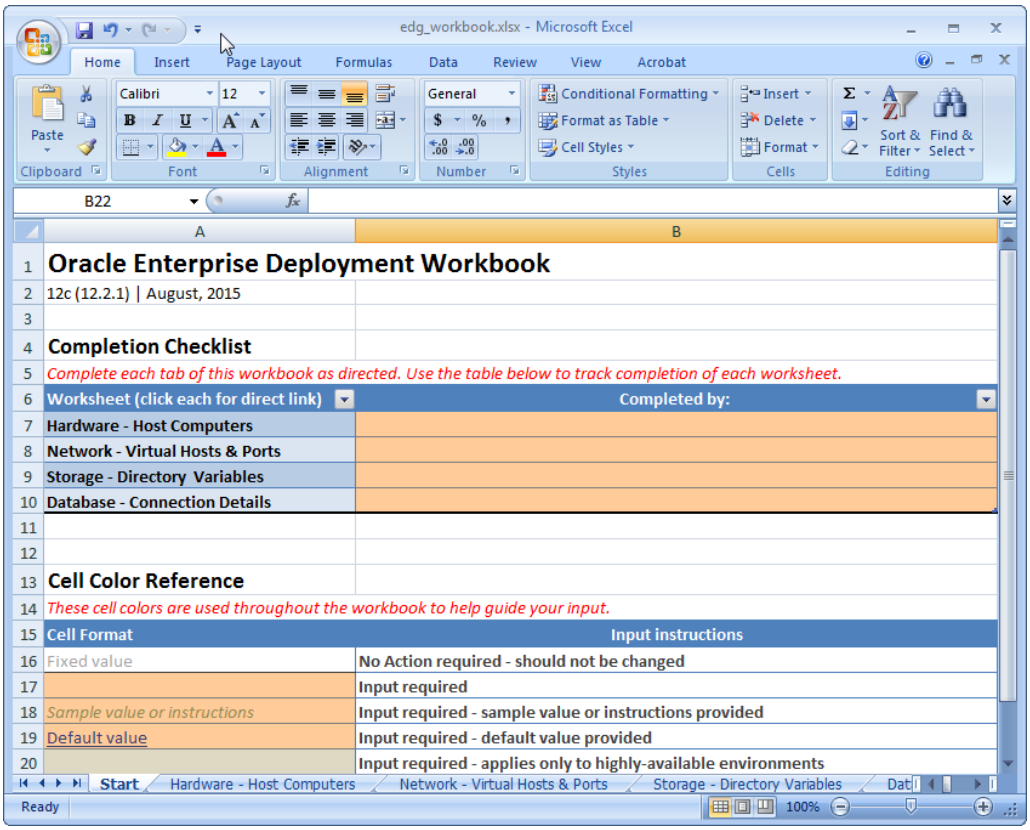
Using the Start Tab

The Start tab of the Enterprise Deployment workbook serves as a table of contents for the rest of the workbook. You can also use it to identify the people who will be completing the spreadsheet.

The Start tab also provides a key to identify the colors used to identify workbook fields that need values, as well as those that are provided for informational purposes.

The following image shows the Start tab of the Enterprise Deployment workbook.

Figure 4-1 Start Tab of the Enterprise Deployment workbook



Using the Hardware - Host Computers Tab

The Hardware - Host Computers tab lists the host computers that are required to install and configure the Oracle WebCenter Content Enterprise Deployment topology.

The reference topologies typically require a minimum of six host computers: two for the web tier, two for the application tier, and two for the Oracle RAC database on the data tier. If you decide to expand the environment to include more systems, add a row for each additional host computer.

The **Abstract Host Name** is the name used throughout this guide to reference the host. For each row, procure a host computer, and enter the **Actual Host Name**. You can then use the actual host name when any of the abstract names is referenced in this guide.

For example, if a procedure in this guide references WCCHOST1, you can then replace the WCCHOST1 variable with the actual name provided on the Hardware - Host Computers tab of the workbook.

> **Note:**
>
> If two domains share the same node, for example, if you set up the Oracle SOA suite, and then create MFT with its own domain, you have two domains on the same node. In this case, you use WCCHOST1 and MFTHOST1 at the same time, one for each domain.

For easy reference, Oracle also recommends that you include the IP address, Operating System (including the version), number of CPUs, and the amount of RAM for each host. This information can be useful during the installation, configuration, and maintenance of the enterprise deployment. See Preparing the Host Computers for an Enterprise Deployment.

Using the Network - Virtual Hosts & Ports Tab

The Network - Virtual Hosts & Ports tab lists the virtual hosts that must be defined by your network administrator before you can install and configure the enterprise deployment topology.

The port numbers are important for several reasons. You must have quick reference to the port numbers so that you can access the management consoles; the firewalls must also be configured to allow network traffic through specific ports.

Each virtual host, virtual IP address, and each network port serves a distinct purpose in the deployment. See Preparing the Load Balancer and Firewalls for an Enterprise Deployment.

In the Network - Virtual Hosts table, review the items in the **Abstract Virtual Host or Virtual IP Name** column. These are the virtual host and virtual IP names that are used in the procedures in this guide. For each abstract name, enter the actual virtual host name that is defined by your network administrator. Whenever this guide references one of the abstract virtual host or virtual IP names, replace that value with the actual corresponding value in this table.

Similarly, in many cases, this guide assumes that you are using default port numbers for the components or products you install and configure. However, in reality, you are likely to use different port numbers. Use the Network - Port Numbers table to map the default port values to the actual values that are used in your specific installation.

Using the Storage - Directory Variables Tab

As part of preparing for an enterprise deployment, it is assumed you are using a standard directory structure, which is recommended for Oracle enterprise deployments.

In addition, procedures in this book reference specific directory locations. Within the procedures, each directory is assigned a consistent variable, which you should replace with the actual location of the directory in your installation.

For each of the directory locations listed on this tab, provide the actual directory path in your installation.

In addition, for the application tier, it is recommended that many of these standard directories be created on a shared storage device. For those directories, the table also provides fields so you can enter the name of the shared storage location and the mount point that is used when you mounted the shared location. See Preparing the File System for an Enterprise Deployment.

Using the Database - Connection Details Tab

When you install and configure the enterprise deployment topology, you often have to make connections to a highly available Oracle Real Application Clusters (RAC) database. In this guide, the procedures reference a set of variables that identify the information you need to provide to connect to the database from tools, such as the Configuration Wizard and the Repository Creation Utility.

To be sure that you have these values handy, use this tab to enter the actual values for these variables in your database installation. See Preparing the Database for an Enterprise Deployment.

Who Should Use the Enterprise Deployment Workbook?

The details of the Enterprise Deployment workbook are filled in by the individual or a team that is responsible for planning, procuring, or setting up each category of resources.

The information in the Enterprise Deployment workbook is divided into categories. Depending on the structure of your organization and roles that are defined for your team, you can assign specific individuals in your organization to fill in the details of the workbook. Similarly, the information in each category can be assigned to the individual or team that is responsible for planning, procuring, or setting up each category of resources.

For example, the workbook can be filled in, reviewed, and used by people in your organization that fill the following roles:

- Information Technology (IT) Director
- Architect
- System Administrator
- Network Engineer
- Database Administrator

5

Procuring Resources for an Enterprise Deployment

It is essential to procure the required hardware, software, and network settings before you configure the Oracle WebCenter Content reference topology.

This chapter provides information on how to reserve the required IP addresses and identify and obtain software downloads for an enterprise deployment.

- Hardware and Software Requirements for the Enterprise Deployment Topology
  It is important to understand the hardware load balancer requirements, host computer hardware requirements, and operating system requirements for the enterprise deployment topology.
- Reserving the Required IP Addresses for an Enterprise Deployment
  You have to obtain and reserve a set of IP addresses before you install and configure the enterprise topology. The set of IP addresses that need to be reserved are listed in this section.
- Identifying and Obtaining Software Distributions for an Enterprise Deployment
  Before you begin to install and configure the enterprise topology, you must obtain the software distributions that you need to implement the topology.

Hardware and Software Requirements for the Enterprise Deployment Topology

It is important to understand the hardware load balancer requirements, host computer hardware requirements, and operating system requirements for the enterprise deployment topology.

This section includes the following sections.

- Hardware Load Balancer Requirements
  The section lists the wanted features of the external load balancer.
- Host Computer Hardware Requirements
  This section provides information to help you procure host computers that are configured to support the enterprise deployment topologies.
- Operating System Requirements for an Enterprise Deployment Topology
  This section provides details about the operating system requirements.

Hardware Load Balancer Requirements

The section lists the wanted features of the external load balancer.

The enterprise topology uses an external load balancer. The features of the external load balancer are:

- Ability to load-balance traffic to a pool of real servers through a virtual host name: Clients access services by 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.
- Ability to configure names and ports on your external load balancer. 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.
- 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.
- 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 by using the equivalent non-SSL protocol (for example, HTTPS to HTTP).
- SSL acceleration (this feature is recommended, but not required for the enterprise topology).
- The ability to route TCP/IP requests; this is a requirement for Oracle SOA Suite for healthcare integration, which uses the Minimum Lower Layer Protocol (MLLP) over TCP.

Host Computer Hardware Requirements

This section provides information to help you procure host computers that are configured to support the enterprise deployment topologies.

It includes the following topics.

- General Considerations for Enterprise Deployment Host Computers
  This section specifies the general considerations that are required for the enterprise deployment host computers.
- Reviewing the Oracle Fusion Middleware System Requirements
  This section provides reference to the system requirements information to help you ensure that the environment meets the necessary minimum requirements.

- Typical Memory, File Descriptors, and Processes Required for an Enterprise Deployment
  This section specifies the typical memory, number of file descriptors, and operating system processes and tasks details required for an enterprise deployment.
- Typical Disk Space Requirements for an Enterprise Deployment
  This section specifies the disk space that is typically required for this enterprise deployment.

General Considerations for Enterprise Deployment Host Computers

This section specifies the general considerations that are required for the enterprise deployment host computers.

Before you start the process of configuring an Oracle Fusion Middleware enterprise deployment, you must perform the appropriate capacity planning to determine the number of nodes, CPUs, and memory requirements for each node depending on the specific system's load as well as the throughput and response requirements. These requirements vary for each application or custom Oracle WebCenter Content system being used.

The information in this chapter provides general guidelines and information that helps you determine the host computer requirements. It does not replace the need to perform capacity planning for your specific production environment.

> **Note:**
>
> As you obtain and reserve the host computers in this section, note the host names and system characteristics in the Enterprise Deployment workbook. You will use these addresses later when you enable the IP addresses on each host computer. See Using the Enterprise Deployment Workbook.

Reviewing the Oracle Fusion Middleware System Requirements

This section provides reference to the system requirements information to help you ensure that the environment meets the necessary minimum requirements.

Review the *Oracle Fusion Middleware System Requirements and Specifications* to ensure that your environment meets the minimum installation requirements for the products that you are installing.

The Requirements and Specifications document contains information about general Oracle Fusion Middleware hardware and software requirements, minimum disk space and memory requirements, database schema requirements, and the required operating system libraries and packages.

It also provides some general guidelines for estimating the memory requirements for your Oracle Fusion Middleware deployment.

Typical Memory, File Descriptors, and Processes Required for an Enterprise Deployment

This section specifies the typical memory, number of file descriptors, and operating system processes and tasks details required for an enterprise deployment.

The following table summarizes the memory, file descriptors, and processes required for the Administration Server and each of the Managed Servers computers in a typical Oracle WebCenter Content enterprise deployment. These values are provided as an example only, but they can be used to estimate the minimum amount of memory required for an initial enterprise deployment.

The example in this topic reflects the minimum requirements for configuring the Managed Servers and other services required on WCCHOST1, as depicted in the reference topologies.

When you are procuring machines, use the information in the **Approximate Top Memory** column as a guide when determining the minimum physical memory each host computer should have available.

After you procure the host computer hardware and verify the operating system requirements, review the software configuration to be sure the operating system settings are configured to accommodate the number of open files listed in the **File Descriptors** column and the number processes listed in the **Operating System Processes and Tasks** column. See Setting the Open File Limit and Number of Processes Settings on UNIX Systems.

Managed Server, Utility, or Service	Approximate Top Memory	Number of File Descriptors	Operating System Processes and Tasks
Administration Server	3.5 GB	3500	165
WLS_WCC	4.0 GB	3100	240
WLS_WCCUI	4.0 GB	3100	100
WLS_CPT	3.0 GB	1300	100
WLS_IBR	3.0 GB	1300	100
WLS_SOA	4.0 GB	3100	240
WLST (connection to the Node Manager)	1.5 GB	910	20
Configuration Wizard	1.5 GB	700	20
Node Manager	1.0 GB	720	15
TOTAL	11.5 GB*	TBD	TBD

* Approximate total, with consideration for Operating System and other additional memory requirements.

Typical Disk Space Requirements for an Enterprise Deployment

This section specifies the disk space that is typically required for this enterprise deployment.

For the latest disk space requirements for the Oracle Fusion Middleware 12*c* (12.2.1.3.0) products, including the Oracle WebCenter Content products, review the *Oracle Fusion Middleware System Requirements and Specifications*.

In addition, the following table summarizes the disk space that is typically required for an Oracle WebCenter Content enterprise deployment.

Use the this information and the information in Preparing the File System for an Enterprise Deployment to determine the disk space requirements required for your deployment.

Server	Disk
Database	nXm n = number of disks, at least 4 (striped as one disk) m = size of the disk (minimum of 30 GB)
WEBHOST*n*	10 GB
WCCHOST*n*	20 GB*

* For a shared storage Oracle home configuration, two installations suffice by making a total of 20 GB.

Operating System Requirements for an Enterprise Deployment Topology

This section provides details about the operating system requirements.

The Oracle Fusion Middleware software products and components that are described in this guide are certified on various operating systems and platforms, which are listed in *Oracle Fusion Middleware System Requirements and Specifications*.

> **Note:**
>
> This guide focuses on the implementation of the enterprise deployment reference topology on Oracle Linux systems.
>
> The topology can be implemented on any certified, supported operating system, but the examples in this guide typically show the commands and configuration steps as they should be performed by using the bash shell on Oracle Linux.

Reserving the Required IP Addresses for an Enterprise Deployment

You have to obtain and reserve a set of IP addresses before you install and configure the enterprise topology. The set of IP addresses that need to be reserved are listed in this section.

Before you begin installing and configuring the enterprise topology, you must obtain and reserve a set of IP addresses:

- Physical IP (IP) addresses for each of the host computers that you have procured for the topology
- A virtual IP (VIP) address for the Administration Server
- Additional VIP addresses for each Managed Server that is configured for Whole Server Migration

  For Fusion Middleware 12*c* products that support Automatic Service Migration, VIPs for the Managed Servers are typically not necessary.
- A unique virtual host name to be mapped to each VIP.

You can then work with your network administrator to be sure that these required VIPs are defined in your DNS server. Alternatively, for non-production environments, you can use the /etc/hosts file to define these virtual hosts.

For more information, see the following topics.

- What is a Virtual IP (VIP) Address?
  This section defines the virtual IP address and specifies its purpose.
- Why Use Virtual Host Names and Virtual IP Addresses?
  For an enterprise deployment, in particular, it is important that a set of VIPs--and the virtual host names to which they are mapped--are reserved and enabled on the corporate network.
- Physical and Virtual IP Addresses Required by the Enterprise Topology
  This section describes the physical IP (IP) and virtual IP (VIP) addresses that are required for the Administration Server and each of the Managed Servers in a typical Oracle WebCenter Content enterprise deployment topology.

What is a Virtual IP (VIP) Address?

This section defines the virtual IP address and specifies its purpose.

A virtual IP address is an unused IP Address that belongs to the same subnet as the host's primary IP address. It is assigned to a host manually. If a host computer fails, the virtual address can be assigned to a new host in the topology. For the purposes of this guide, *virtual* IP addresses are referenced, which can be reassigned from one host to another, and *physical* IP addresses are referenced, which are assigned permanently to hardware host computer.

Why Use Virtual Host Names and Virtual IP Addresses?

For an enterprise deployment, in particular, it is important that a set of VIPs--and the virtual host names to which they are mapped--are reserved and enabled on the corporate network.

Alternatively, host names can be resolved through appropriate `/etc/hosts` file propagated through the different nodes.

In the event of the failure of the host computer where the IP address is assigned, the IP address can be assigned to another host in the same subnet, so that the new host can take responsibility for running the Managed Servers that are assigned to it.

The reassignment of virtual IP address for the Administration Server must be performed manually, but the reassignment of virtual IP addresses for Managed Servers can be performed automatically by using the Whole Server Migration feature of Oracle WebLogic Server.

Whether you should use Whole Server Migration or not depends upon the products that you are deploying and whether they support Automatic Service Migration.

Physical and Virtual IP Addresses Required by the Enterprise Topology

This section describes the physical IP (IP) and virtual IP (VIP) addresses that are required for the Administration Server and each of the Managed Servers in a typical Oracle WebCenter Content enterprise deployment topology.

Before you begin to install and configure the enterprise deployment, reserve a set of host names and IP addresses that correspond to the VIPs in Table 5-1.

You can assign any unique host name to the VIPs, but in this guide, each VIP is referenced by using the suggested host names in the table.

> **Note:**
>
> As you obtain and reserve the IP addresses and their corresponding virtual host names in this section, note the values of the IP addresses and host names in the Enterprise Deployment workbook. You will use these addresses later when you enable the IP addresses on each host computer. See Using the Enterprise Deployment Workbook .

Table 5-1 Summary of the Virtual IP Addresses Required for the Enterprise Deployment

Virtual IP	VIP Maps to...	Description
VIP1	ADMINVHN	ADMINVHN is the virtual host name used as 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 Administration Server process is running.

Identifying and Obtaining Software Distributions for an Enterprise Deployment

Before you begin to install and configure the enterprise topology, you must obtain the software distributions that you need to implement the topology.

The following table lists the distributions used in this guide.

For general information about how to obtain Oracle Fusion Middleware software, see Obtaining Product Distributions in *Oracle Fusion Middleware Planning an Installation of Oracle Fusion Middleware*.

For more specific information about locating and downloading specific Oracle Fusion Middleware products, see the *Oracle Fusion Middleware Download, Installation, and Configuration Readme Files* on OTN.

> **Note:**
>
> The information in this guide is meant to complement the information contained in the Oracle Fusion Middleware certification matrixes. If there is a conflict of information between this guide and the certification matrixes, then the information in the certification matrixes must be considered the correct version, as they are frequently updated.

Distribution	Description
Oracle Fusion Middleware 12*c* (12.2.1.3.0) Infrastructure	Download this distribution to install the Oracle Fusion Middleware Infrastructure, which includes Oracle WebLogic Server and Java Required Files software required for Oracle Fusion Middleware products. This distribution also installs the Repository Creation Utility (RCU), which in previous Oracle Fusion Middleware releases was packaged in its own distribution.
Oracle HTTP Server 12*c* (12.2.1.3.0)	Download this distribution to install the Oracle HTTP Server software on the Web Tier.
Oracle Fusion Middleware 12*c* (12.2.1.3.0) WebCenter Content	Download this distribution to install the Oracle WebCenter Content.

Distribution	Description
Oracle Fusion Middleware 12*c* (12.2.1.3.0) SOA Suite	Download this distribution if you plan to install and configure Oracle SOA Suite as part of the Oracle WebCenter Content enterprise topology.

6

Preparing the Load Balancer and Firewalls for an Enterprise Deployment

It is important to understand how to configure the hardware load balancer and ports that must be opened on the firewalls for an enterprise deployment.

- Configuring Virtual Hosts on the Hardware Load Balancer
  The hardware load balancer configuration facilitates to recognize and route requests to several virtual servers and associated ports for different types of network traffic and monitoring.
- Configuring the Firewalls and Ports for an Enterprise Deployment
  As an administrator, it is important that you become familiar with the port numbers used by various Oracle Fusion Middleware products and services. This ensures that the same port number is not used by two services on the same host, and that the proper ports are open on the firewalls in the enterprise topology.

Configuring Virtual Hosts on the Hardware Load Balancer

The hardware load balancer configuration facilitates to recognize and route requests to several virtual servers and associated ports for different types of network traffic and monitoring.

The following topics explain how to configure the hardware load balancer, provide a summary of the virtual servers that are required, and provide additional instructions for these virtual servers:

- Overview of the Hardware Load Balancer Configuration
- Typical Procedure for Configuring the Hardware Load Balancer
- Summary of the Virtual Servers Required for an Enterprise Deployment
- Additional Instructions for admin.example.com
- Additional Instructions for wcc.example.com
- Additional Instructions for wccinternal.example.com

Overview of the Hardware Load Balancer Configuration

As shown in the topology diagrams, you must configure the hardware load balancer to recognize and route requests to several virtual servers and associated ports for different types of network traffic and monitoring.

In the context of a load-balancing device, a virtual server is a construct that allows multiple physical servers to appear as one for load-balancing purposes. It is typically represented by an IP address and a service, and it is used to distribute incoming client requests to the servers in the server pool.

The virtual servers should be configured to direct traffic to the appropriate host computers and ports for the various services that are available in the enterprise deployment.

In addition, you should configure the load balancer to monitor the host computers and ports for availability so that the traffic to a particular server is stopped as soon as possible when a service is down. This ensures that incoming traffic on a given virtual host is not directed to an unavailable service in the other tiers.

Note that after you configure the load balancer, you can later configure the web server instances in the web tier to recognize a set of virtual hosts that use the same names as the virtual servers that you defined for the load balancer. For each request coming from the hardware load balancer, the web server can then route the request appropriately, based on the server name included in the header of the request. See Configuring Oracle HTTP Server for Administration and Oracle Web Services Manager.

Typical Procedure for Configuring the Hardware Load Balancer

The following procedure outlines the typical steps for configuring a hardware load balancer for an enterprise deployment.

Note that the actual procedures for configuring a specific load balancer will differ, depending on the specific type of load balancer. There may also be some differences depending on the type of protocol that is being load balanced. For example, TCP virtual servers and HTTP virtual servers use different types of monitors for their pools. Refer to the vendor-supplied documentation for actual steps.

1. Create a pool of servers. This pool contains a list of servers and the ports that are included in the load-balancing definition.

   For example, for load balancing between the web hosts, create a pool of servers that would direct requests to hosts WEBHOST1 and WEBHOST2 on port 7777.

2. Create rules to determine whether a given host and service is available and assign it to the pool of servers that are described in Step 1.

3. Create the required virtual servers on the load balancer for the addresses and ports that receive requests for the applications.

   For a complete list of the virtual servers required for the enterprise deployment, see Summary of the Virtual Servers Required for an Enterprise Deployment.

   When you define each virtual server on the load balancer, consider the following:

   a. If your load balancer supports it, specify whether the virtual server is available internally, externally, or both. Ensure that internal addresses are only resolvable from inside the network.

   b. Configure SSL Termination, if applicable, for the virtual server.

   c. Assign the pool of servers created in Step 1 to the virtual server.

Summary of the Virtual Servers Required for an Enterprise Deployment

This topic provides details of the virtual servers that are required for an enterprise deployment.

The following table provides a list of the virtual servers that you must define on the hardware load balancer for the Oracle WebCenter Content enterprise topology:

Virtual Host	Server Pool	Protocol	SSL Termination?	External?
admin.example.com:80	WEBHOST1.example.com:7777 WEBHOST2.example.com:7777	HTTP	No	No
wcc.example.com:443	WEBHOST1.example.com:7777 WEBHOST2.example.com:7777	HTTPS	Yes	Yes
wccinternal.example.com:80	WEBHOST1.example.com:7777 WEBHOST2.example.com:7777	HTTP	No	No
wccinternal.example.com:6300	WCCHOST1.example.com:4444 WCCHOST2.example.com:4444	TCP	No	No

Additional Instructions for admin.example.com

This section provides additional instructions that are required for the virtual server-admin.example.com.

When you configure this virtual server on the hardware load balancer:

- Enable address and port translation.
- Enable reset of connections when services or hosts are down.

Additional Instructions for wcc.example.com

The address `wcc.example.com` is a virtual server name that acts as the access point for all HTTP traffic to the runtime Oracle WebCenter Content components. Traffic to SSL is configured. Clients access this service using the address `wcc.example.com:443` for HTTP. When you configure this virtual server on the hardware load balancer:

- Use port 80 and port 443. Any request that goes to port 80 (non-SSL protocol) should be redirected to port 443 (SSL protocol).
- Specify ANY as the protocol (non-HTTP protocols are required for B2B).
- Enable address and port translation.
- Enable reset of connections when services and/or nodes are down.
- Create rules to filter out access to `/console` and `/em` on this virtual server.

  These context strings direct requests to Oracle WebLogic Server Administration Console and to Oracle Enterprise Manager Fusion Middleware Control and should be used only when accessing the system from `admin.example.com`.

Additional Instructions for wccinternal.example.com

The address `wccinternal.example.com` is a virtual server name used for internal invocations of WebCenter Content and SOA services. This URL is not exposed to the Internet and is accessible only from the intranet. The incoming traffic from clients is not SSL enabled.

This address is used for both HTTP and Remote Intradoc Client (RIDC) traffic using different ports:

- HTTP traffic is routed via ports 7777 and 80.
- RIDC traffic is routed via ports 6300 and 4400 over TCP.

HTTP traffic goes to goes to Web tier hosts pool (WEBHOST1 and WEBHOST2) pool.

The end-user RIDC traffic (such as from the WebCenter Content ADF UI) goes directly to the WebCenter Content hosts pool (WCCHOST1 and WCCHOST2). The RIDC traffic uses port 6300 on the load balancer (6300 is mainly for masking), but the traffic ultimately routes to the RIDC port (4444) on the WebCenter hosts.

For Oracle SOA Suite systems, users can set this URL while modeling composites or at runtime with the appropriate MBean through Fusion Middleware Control, as the URL to be used for invoking internal services. You can use this URL for any service callback to WebCenter Content components from Oracle SOA Suite. Clients access this service using the address `wccinternal.example.com:80`, and the requests are forwarded to port 7777 on WEBHOST1 and WEBHOST2.

When you configure this virtual server on the hardware load balancer:

- Enable address and port translation.
- Enable reset of connections when services or nodes are down.
- As with `wcc.example.com`, create rules to filter out access to `/console` and `/em` on this virtual server.

Configuring the Firewalls and Ports for an Enterprise Deployment

As an administrator, it is important that you become familiar with the port numbers used by various Oracle Fusion Middleware products and services. This ensures that the same port number is not used by two services on the same host, and that the proper ports are open on the firewalls in the enterprise topology.

The following tables lists the ports that you must open on the firewalls in the topology:

Firewall notation:

- FW0 refers to the outermost firewall.
- FW1 refers to the firewall between the web tier and the application tier.
- FW2 refers to the firewall between the application tier and the data tier.

Table 6-1 Firewall Ports Common to All Fusion Middleware Enterprise Deployments

Type	Firewall	Port and Port Range	Protocol / Application	Inbound / Outbound	Other Considerations and Timeout Guidelines
Browser request	FW0	80	HTTP / Load Balancer	Inbound	Timeout depends on the size and type of HTML content.

Table 6-1 (Cont.) Firewall Ports Common to All Fusion Middleware Enterprise Deployments

Type	Firewall	Port and Port Range	Protocol / Application	Inbound / Outbound	Other Considerations and Timeout Guidelines
Browser request	FW0	443	HTTPS / Load Balancer	Inbound	Timeout depends on the size and type of HTML content.
Browser request	FW1	80	HTTP / Load Balancer	Outbound (for intranet clients)	Timeout depends on the size and type of HTML content.
Browser request	FW1	443	HTTPS / Load Balancer	Outbound (for intranet clients)	Timeout depends on the size and type of HTML content.
Callbacks and Outbound invocations	FW1	80	HTTP / Load Balancer	Outbound	Timeout depends on the size and type of HTML content.
Callbacks and Outbound invocations	FW1	443	HTTPS / Load Balancer	Outbound	Timeout depends on the size and type of HTML content.
Load balancer to Oracle HTTP Server	n/a	7777	HTTP	n/a	n/a
OHS registration with Administration Server	FW1	7001	HTTP/t3	Inbound	Set the timeout to a short period (5-10 seconds).
OHS management by Administration Server	FW1	OHS Admin Port (7779)	TCP and HTTP, respectively	Outbound	Set the timeout to a short period (5-10 seconds).
Session replication within a WebLogic Server cluster	n/a	n/a	n/a	n/a	By default, this communication uses the same port as the server's listen address.

Table 6-1 (Cont.) Firewall Ports Common to All Fusion Middleware Enterprise Deployments

Type	Firewall	Port and Port Range	Protocol / Application	Inbound / Outbound	Other Considerations and Timeout Guidelines
Administration Console access	FW1	7001	HTTP / Administration Server and Enterprise Manager t3	Both	You should tune this timeout based on the type of access to the admin console (whether it is planned to use the Oracle WebLogic Server Administration Console from application tier clients or clients external to the application tier).
Database access	FW2	1521	SQL*Net	Both	Timeout depends on database content and on the type of process model used for SOA.
Coherence for deployment	n/a	8088 Range: 8000 - 8090		n/a	n/a
Oracle Unified Directory access	FW2	389 636 (SSL)	LDAP or LDAP/ssl	Inbound	You should tune the directory server's parameters based on load balancer, and not the other way around.
Oracle Notification Server (ONS)	FW2	6200	ONS	Both	Required for Gridlink. An ONS server runs on each database server.

Table 6-2 Firewall Ports for Product-specific Components in Oracle Fusion Middleware Enterprise Deployments

Type	Firewall	Port and Port Range	Protocol / Application	Inbound / Outbound	Other Considerations and Timeout Guidelines
Oracle SOA Suite and WSM Server access	FW1	8001 Range: 8000 - 8080	HTTP / WLS_SOA*n*	Inbound	Timeout varies based on the type of process model used for SOA.
Oracle WebCenter Content Access	FW1	16200	HTTP / WLS_WCC*n*	Inbound	Browser-based access. Configurable session timeouts.
RIDC API requests	FW1	6300	TCP/WLS_WCC*n*	Inbound	n/a
Oracle WebCenter Enterprise Capture access	FW1	16400	HTTP / WLS_CPT*n*	Inbound	Browser-based access. Configurable session timeouts.
Communication between SOA_Cluster members	n/a	8001	TCP/IP Unicast	n/a	By default, this communication uses the same port as the server's listen address.
Communication between WCC_Cluster members	n/a	16200	TCP/IP Unicast	n/a	By default, this communication uses the same port as the server's listen address.
Communication between CPT_Cluster members	n/a	16400	TCP/IP Unicast	n/a	By default, this communication uses the same port as the server's listen address.

7

Preparing the File System for an Enterprise Deployment

Preparing the file system for an enterprise deployment involves understanding the requirements for local and shared storage, as well as the terminology that is used to reference important directories and file locations during the installation and configuration of the enterprise topology.

This chapter describes how to prepare the file system for an Oracle Fusion Middleware enterprise deployment.

- Overview of Preparing the File System for an Enterprise Deployment
  It is important to set up your storage in a way that makes the enterprise deployment easy to understand, configure, and manage.
- Shared Storage Recommendations When Installing and Configuring an Enterprise Deployment
  Oracle recommends that you implement certain guidelines regarding shared storage when you install and configure an enterprise deployment.
- Understanding the Recommended Directory Structure for an Enterprise Deployment
  The diagrams in this section show the recommended directory structure for a typical Oracle Fusion Middleware enterprise deployment.
- File System and Directory Variables Used in This Guide
  Understanding the file system directories and the directory variables used to reference these directories is essential for installing and configuring the enterprise deployment topology.
- About Creating and Mounting the Directories for an Enterprise Deployment
  Oracle recommends that you implement certain best practices when you create or mount the top-level directories in an enterprise deployment.
- Summary of the Shared Storage Volumes in an Enterprise Deployment
  It is important to understand the shared volumes and their purpose in a typical Oracle Fusion Middleware enterprise deployment.

Overview of Preparing the File System for an Enterprise Deployment

It is important to set up your storage in a way that makes the enterprise deployment easy to understand, configure, and manage.

This chapter provides an overview of the process of preparing the file system for an enterprise deployment. Oracle recommends setting up your storage according to information in this chapter. The terminology defined in this chapter is used in the diagrams and procedures throughout the guide.

Use this chapter as a reference to understand the directory variables that are used in the installation and configuration procedures.

Other directory layouts are possible and supported, but the model adopted in this guide was designed 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.

Shared Storage Recommendations When Installing and Configuring an Enterprise Deployment

Oracle recommends that you implement certain guidelines regarding shared storage when you install and configure an enterprise deployment.

Before you implement the detailed recommendations in this chapter, be sure to review the recommendations and general information about using shared storage in the *Oracle Fusion Middleware High Availability Guide*.

The recommendations in this chapter are based on the concepts and guidelines described in the *Oracle Fusion Middleware High Availability Guide*.

Table 7-1 lists the key sections that you should review and how those concepts apply to an enterprise deployment.

Table 7-1 Shared Storage Resources in the High Availability Guide

Section in *High Availability Guide*	Importance to an Enterprise Deployment
Shared Storage Prerequisites	Describes guidelines for disk format and the requirements for hardware devices that are optimized for shared storage.
Using Shared Storage for Binary (Oracle Home) Directories	Describes your options for storing the Oracle home on a shared storage device that is available to multiple hosts. For an enterprise deployment, Oracle recommends that you use redundant Oracle homes on separate storage volumes. If a separate volume is not available, a separate partition on the shared disk should be used to provide redundant Oracle homes to application tier hosts.
Using Shared Storage for Domain Configuration Files	Describes the concept of creating separate domain homes for the Administration Server and the Managed Servers in the domain. For an enterprise deployment, the Administration Server domain home location is referenced by the *ASERVER_HOME* variable.
Shared Storage Requirements for JMS Stores and JTA Logs	Provides instructions for setting the location of the transaction logs and JMS stores for an enterprise deployment.
Introduction to Zero Downtime Patching	Describes the Zero Downtime feature and the procedure to configure and monitor workflows.

> **Note:**
>
> Zero Downtime Patching (ZDT Patching) provides an automated mechanism to orchestrate the rollout of patches while avoiding downtime or loss of sessions. ZDT reduces risks and downtime of mission-critical applications that require availability and predictability while applying patches.
>
> By using the workflows that you define, you can patch or update any number of nodes in a domain with little or no manual intervention. Changes are rolled out to one node at a time. This preemptively allows for session data to be migrated to compatible servers in the cluster and allows service migration of singleton services, such as JTA and JMS.
>
> When you patch the Oracle home, the current Oracle home must be installed locally on each node that is included in the workflow. Although it is not required, Oracle also recommends that the Oracle home be in the same location on each node.

Understanding the Recommended Directory Structure for an Enterprise Deployment

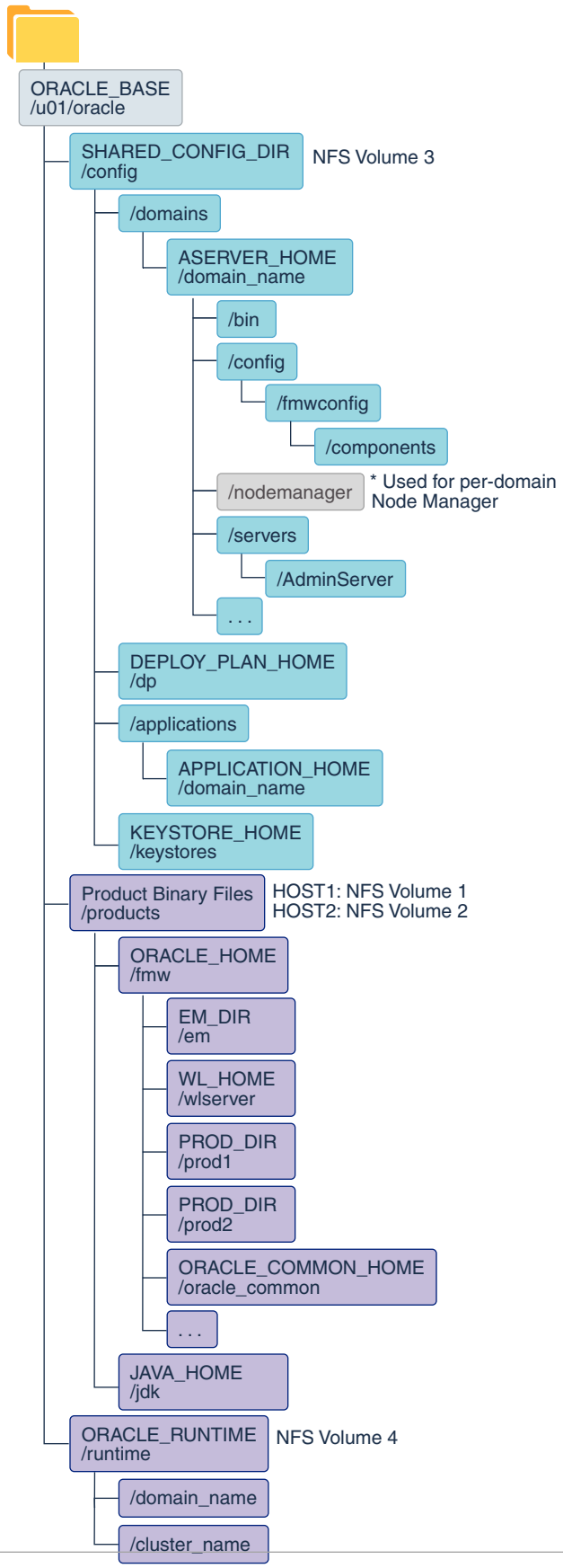
The diagrams in this section show the recommended directory structure for a typical Oracle Fusion Middleware enterprise deployment.

The directories shown in the diagrams contain binary files that are installed on disk by the Oracle Fusion Middleware installers, domain-specific files generated via the domain configuration process, as well as domain configuration files that are propagated to the various host computers via the Oracle WebLogic Server `pack` and `unpack` commands:

- Figure 7-1 shows the resulting directory structure on the shared storage device after you have installed and configured a typical Oracle Fusion Middleware enterprise deployment. The shared storage directories are accessible by the application tier host computers.
- Figure 7-2 shows the resulting directory structure on the local storage device for a typical application tier host after you have installed and configured an Oracle Fusion Middleware enterprise deployment. The Managed Servers in particular are stored on the local storage device for the application tier host computers.
- Figure 7-3 shows the resulting directory structure on the local storage device for a typical Web tier host after you have installed and configured an Oracle Fusion Middleware enterprise deployment. Note that the software binaries (in the Oracle home) are installed on the local storage device for each Web tier host.

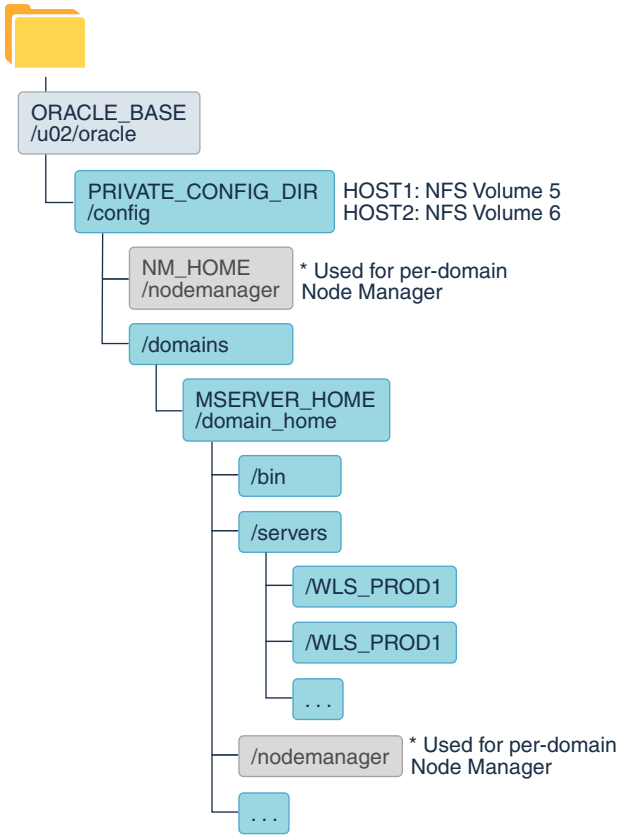
Where applicable, the diagrams also include the standard variables used to reference the directory locations in the installation and configuration procedures in this guide.

Figure 7-1 Recommended Shared Storage Directory Structure for an Enterprise Deployment



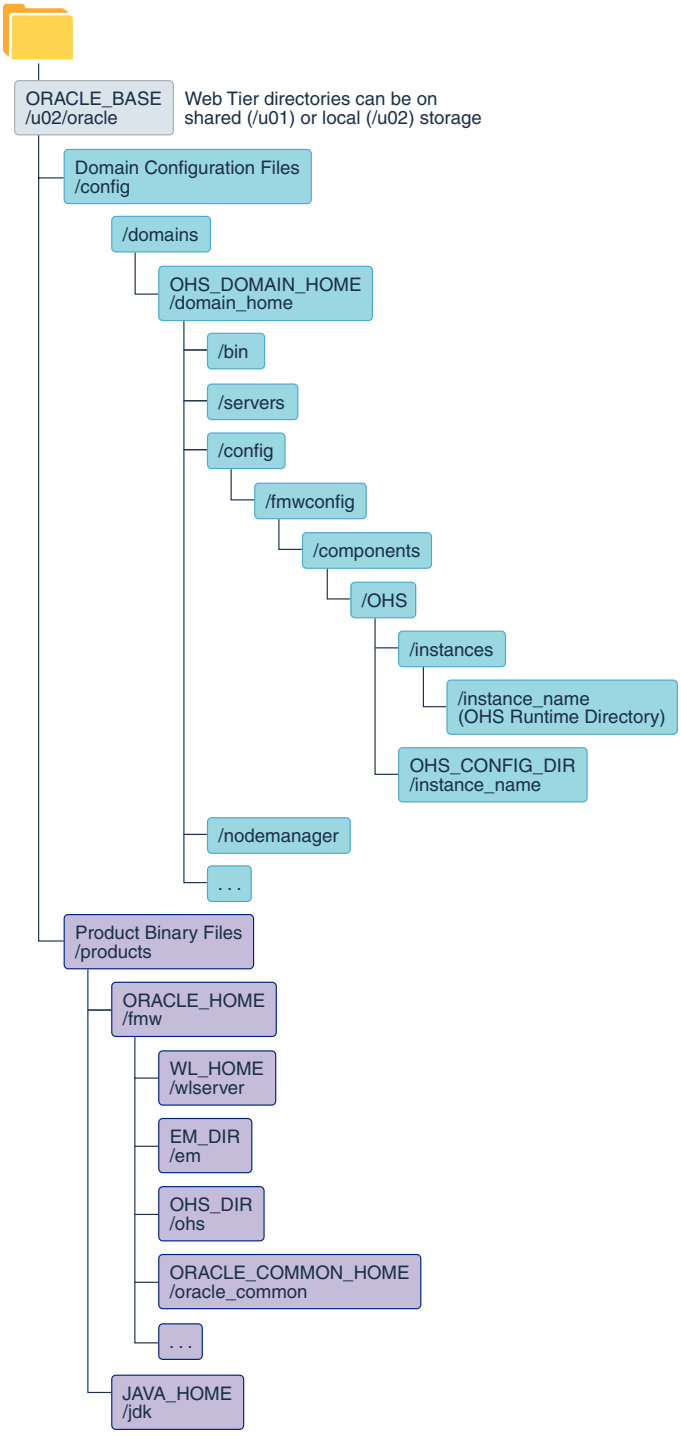
* See About the Node Manager Configuration in a Typical Enterprise Deployment.

Figure 7-2 Recommended Local Storage Directory Structure for an Application Tier Host Computer in an Enterprise Deployment



* See About the Node Manager Configuration in a Typical Enterprise Deployment.

Figure 7-3 Recommended Local Storage Directory Structure for a Web Tier Host Computer in an Enterprise Deployment



File System and Directory Variables Used in This Guide

Understanding the file system directories and the directory variables used to reference these directories is essential for installing and configuring the enterprise deployment topology.

Table 7-2 lists the file system directories and the directory variables that are used to reference the directories on the application tier. Table 7-3 lists the file system directories and variables that are used to reference the directories on the web tier.

For additional information about mounting these directories when you use shared storage, see About Creating and Mounting the Directories for an Enterprise Deployment.

Throughout this guide, the instructions for installing and configuring the topology refer to the directory locations that use the variables shown here.

You can also define operating system variables for each of the directories listed in this section. If you define system variables for the particular UNIX shell that you are using, you can then use the variables as they are used in this document, without having to map the variables to the actual values for your environment.

> **Note:**
>
> As you configure your storage devices to accommodate the recommended directory structure, note the actual directory paths in the Enterprise Deployment workbook. You will use these addresses later when you enable the IP addresses on each host computer.
>
> See Using the Enterprise Deployment Workbook.

Table 7-2 Sample Values for Key Directory Variables on the Application Tier

Directory Variable	Description	Relative Path	Sample Value on the Application Tier
ORACLE_BASE	The base directory, under which Oracle products are installed.	N/A	`/u01/oracle`
ORACLE_HOME	The read-only location for the product binaries. For the application tier host computers, it is stored on shared disk. The Oracle home is created when you install the Oracle Fusion Middleware Infrastructure software. You can then install additional Oracle Fusion Middleware products into the same Oracle home.	*ORACLE_BASE*`/products/fmw`	`/u01/oracle/products/fmw`

Table 7-2 (Cont.) Sample Values for Key Directory Variables on the Application Tier

Directory Variable	Description	Relative Path	Sample Value on the Application Tier
ORACLE_COMMON_HOME	The directory within the Oracle Fusion Middleware Oracle home where common utilities, libraries, and other common Oracle Fusion Middleware products are stored.	*ORACLE_HOME*/oracle_common	/u01/oracle/products/fmw/oracle_common
WL_HOME	The directory within the Oracle home where the Oracle WebLogic Server software binaries are stored.	*ORACLE_HOME*/wlserver	/u01/oracle/products/fmw/wlserver
PROD_DIR	Individual product directories for each Oracle Fusion Middleware product that you install.	*ORACLE_HOME*/*prod_dir*	/u01/oracle/products/fmw/*prod_dir* The product can be soa, wcc, idm, bi, or another value, depending on your enterprise deployment.
EM_DIR	The product directory used to store the Oracle Enterprise Manager Fusion Middleware Control software binaries.	*ORACLE_HOME*/em	/u01/oracle/products/fmw/em
JAVA_HOME	The location where you install the supported Java Development Kit (JDK).	*ORACLE_BASE*/products/jdk	/u01/oracle/products/jdk
SHARED_CONFIG_DIR	The shared parent directory for shared environment configuration files, including domain configuration, keystores, runtime artifacts, and application deployments	*ORACLE_BASE*/config	/u01/oracle/config
PRIVATE_CONFIG_DIR	The local or nfs-mounted private configuration directory unique to a given host containing the machine-specific domain directory (MSERVER_HOME). Directory variable: PRIVATE_CONFIG_DIR	/u02/oracle/config	/u02/oracle/config
ASERVER_HOME	The Administration Server domain home, which is installed on a shared disk.	*SHARED_CONFIG_DIR*/domains/*domain_name*	/u01/oracle/config/domains/*domain_name* In this example, replace *domain_name* with the name of the WebLogic Server domain.

Table 7-2 (Cont.) Sample Values for Key Directory Variables on the Application Tier

Directory Variable	Description	Relative Path	Sample Value on the Application Tier
MSERVER_HOME	The Managed Server domain home, which is created by using the `unpack` command on the local disk of each application tier host.	*PRIVATE_CONFIG_DIR*/domains/*domain_name*	/u02/oracle/config/domains/*domain_name* In this example, replace *domain_name* with the name of the WebLogic Server domain.
APPLICATION_HOME	The Application home directory, which is installed on shared disk, so the directory is accessible by all the application tier host computers.	*SHARED_CONFIG_DIR*/applications/*domain_name*	/u01/oracle/config/applications/*domain_name* In this example, replace *domain_name* with the name of the WebLogic Server domain.
ORACLE_RUNTIME	This directory contains the Oracle runtime artifacts, such as the JMS logs and TLogs. Typically, you mount this directory as a separate shared file system, which is accessible by all hosts in the domain. When you run the Configuration Wizard or perform post-configuration tasks, and you identify the location of JMS stores or tlogs persistent stores, then you can use this directory, qualified with the name of the domain, the name of the cluster, and the purpose of the directory. For example: *ORACLE_RUNTIME*/*cluster_name*/jms	*ORACLE_BASE*/runtime	/u01/oracle/runtime/
NM_HOME	The directory used by the Per Machine Node Manager start script and configuration files. **Note:** This directory is necessary only if you are using a Per Machine Node Manager configuration. See About the Node Manager Configuration in a Typical Enterprise Deployment.	*PRIVATE_CONFIG_DIR*/node_manager	/u02/oracle/config/node_manager

Table 7-2 (Cont.) Sample Values for Key Directory Variables on the Application Tier

Directory Variable	Description	Relative Path	Sample Value on the Application Tier
DEPLOY_PLAN_HOME	The deployment plan directory, which is used as the default location for application deployment plans. **Note:** This directory is required only when you are deploying custom applications to the application tier.	*SHARED_CONFIG_DIR*/dp	/u01/oracle/config/dp
KEYSTORE_HOME	The shared location for custom certificates and keystores.	*SHARED_CONFIG_DIR*/keystores	/u01/oracle/config/keystores

Table 7-3 Sample Values for Key Directory Variables on the Web Tier

Directory Variable	Description	Sample Value on the Web Tier
WEB_ORACLE_HOME	The read-only location for the Oracle HTTP Server product binaries. For the web tier host computers, this directory is stored on the local disk. The Oracle home is created when you install the Oracle HTTP Server software .	/u02/oracle/products/fmw
ORACLE_COMMON_HOME	The directory within the Oracle HTTP Server Oracle home where common utilities, libraries, and other common Oracle Fusion Middleware products are stored.	/u02/oracle/products/fmw/oracle_common
WL_HOME	The directory within the Oracle home where the Oracle WebLogic Server software binaries are stored.	/u02/oracle/products/fmw/wlserver
PROD_DIR	Individual product directories for each Oracle Fusion Middleware product that you install.	/u02/oracle/products/fmw/ohs
JAVA_HOME	The location where you install the supported Java Development Kit (JDK).	/u02/oracle/products/jdk
WEB_DOMAIN_HOME	The Domain home for the standalone Oracle HTTP Server domain, which is created when you install Oracle HTTP Server on the local disk of each web tier host.	/u02/oracle/config/domains/*domain_name* In this example, replace *domain_name* with the name of the WebLogic Server domain.

Table 7-3 (Cont.) Sample Values for Key Directory Variables on the Web Tier

Directory Variable	Description	Sample Value on the Web Tier
WEB_CONFIG_DIR	This is the location where you edit the Oracle HTTP Server configuration files (for example, `httpd.conf` and `moduleconf/*.conf`) on each web host. Note this directory is also referred to as the OHS Staging Directory. Changes made here are later propagated to the OHS Runtime Directory. See Staging and Run-time Configuration Directories in the *Oracle Fusion Middleware Administering Oracle HTTP Server*.	`/u02/oracle/config/domains/`*`/domain_name/`*`config/fmwconfig/components/OHS/`*`instance/`*`/instance_name`
WEB_KEYSTORE_HOME	If you use Oracle Traffic Director as your web server, this is the location for custom certificates and keystores.	`/u02/oracle/config/keystores`

About Creating and Mounting the Directories for an Enterprise Deployment

Oracle recommends that you implement certain best practices when you create or mount the top-level directories in an enterprise deployment.

- For the application tier, install the Oracle home, which contains the software binaries, on a second shared storage volume or second partition that is mounted to WCCHOST2. Be sure the directory path to the binaries on WCCHOST2 is identical to the directory path on WCCHOST1.

  For example:

  ```
  /u01/oracle/products/fmw/
  ```

  See Shared Storage Recommendations When Installing and Configuring an Enterprise Deployment.

- This enterprise deployment guide assumes that the Oracle Web tier software is installed on a local disk.

  The Web tier installation is typically performed on local storage to the WEBHOST nodes. When you use shared storage, you can install the Oracle Web tier binaries (and create the Oracle HTTP Server instances) on a shared disk. However, if you do so, then the shared disk *must* be separate from the shared disk used for the application tier, and you must consider the appropriate security restrictions for access to the storage device across tiers.

  As with the application tier servers (WCCHOST1 and WCCHOST2), use the same directory path on both computers.

  For example:

  ```
  /u02/oracle/products/fmw/
  ```

Summary of the Shared Storage Volumes in an Enterprise Deployment

It is important to understand the shared volumes and their purpose in a typical Oracle Fusion Middleware enterprise deployment.

You can use shared storage to host the Web tier binaries and config to make backups easier so that files are stored on a more fault-tolerant hardware, but each node needs to use a private directory that is not shared with the other nodes.

The following table summarizes the shared volumes and their purpose in a typical Oracle Fusion Middleware enterprise deployment.

See Shared Storage Recommendations When Installing and Configuring an Enterprise Deployment.

Table 7-4 Shared Storage Volumes in an Enterprise Deployment

Volume in Shared Storage	Mounted to Host	Mount Directories	Description and Purpose
NFS Volume 1	WCCHOST1	`/u01/oracle/products/`	Storage for the product binaries to be used by WCCHOST1; this is where the Oracle home directory and product directories are installed. Used initially by WCCHOST1, but can be shared with other hosts when scaling-out the topology.
NFS Volume 2	WCCHOST2	`/u01/oracle/products/`	Storage for the product binaries to be used by WCCHOST2; this is where the Oracle home directory and product directories are installed. Used initially by WCCHOST2, but can be shared with other hosts when scaling-out the topology.
NFS Volume 3	WCCHOST1 WCCHOST2	`/u01/oracle/config/`	Administration Server domain configuration, mounted to all hosts; used initially by WCCHOST1, but can be failed over to any host.

Table 7-4 (Cont.) Shared Storage Volumes in an Enterprise Deployment

Volume in Shared Storage	Mounted to Host	Mount Directories	Description and Purpose
NFS Volume 4	WCCHOST1 WCCHOST2	`/u01/oracle/runtime/`	The runtime artifacts directory, mounted to all hosts, contains runtime artifacts such as JMS logs, blogs, and any cluster-dependent shared files needed.
NFS Volume 5	WCCHOST1	`/u02/oracle/config/`	Local storage for the Managed Server domain directory to be used by WCCHOST1, if the private Managed Server domain directory resides on shared storage.
NFS Volume 6	WCCHOST2	`/u02/oracle/config/`	Local storage for the Managed Server domain directory to be used by WCCHOST2, if the private Managed Server domain directory resides on shared storage.
NFS Volume 7	WEBHOST1	`/u02/oracle/`	Local storage for the Oracle HTTP Server or the Oracle Traffic Director software binaries (Oracle home) and domain configuration files that are used by WEBHOST1, if the web tier private binary and config directories reside on shared storage.
NFS Volume 8	WEBHOST2	`/u02/oracle/`	Local storage for the Oracle HTTP Server or the Oracle Traffic Director software binaries (Oracle home) and domain configuration files that are used by WEBHOST2, if the Web Tier private binary and config directories reside on shared storage.

8

Preparing the Host Computers for an Enterprise Deployment

It is important to perform a set of tasks on each computer or server before you configure the enterprise deployment topology. This involves verifying the minimum hardware and operating system requirements for each host, configuring operating system users and groups, enabling Unicode support, mounting the required shared storage systems to the host and enabling the required virtual IP addresses on each host.

This chapter describes the tasks that you must perform from each computer or server that is hosting the enterprise deployment.

- Verifying the Minimum Hardware Requirements for Each Host
  After you procure the required hardware for the enterprise deployment, it is important to ensure that each host computer meets the minimum system requirements.
- Verifying Linux Operating System Requirements
  You can review the typical Linux operating system settings for an enterprise deployment in this section.
- Setting the DNS Settings
- Configuring Operating System Users and Groups
  The users and groups to be defined on each of the computers that host the enterprise deployment are listed in this section.
- Enabling Unicode Support
  It is recommended to enable Unicode support in your operating system so as to allow processing of characters in Unicode.
- Mounting the Required Shared File Systems on Each Host
  It is important to understand how to mount the shared storage to all the servers that require access.
- Enabling the Required Virtual IP Addresses on Each Host
  You must enable the required virtual IP addresses on each host in order to prepare the host for the enterprise deployment.

Verifying the Minimum Hardware Requirements for Each Host

After you procure the required hardware for the enterprise deployment, it is important to ensure that each host computer meets the minimum system requirements.

After you have procured the required hardware for the enterprise deployment, log in to each host computer and verify the system requirements listed in Hardware and Software Requirements for the Enterprise Deployment Topology.

If you deploy to a virtual server environment, such as Oracle Exalogic, ensure that each of the virtual servers meets the minimum requirements.

Ensure that you have sufficient local disk storage and shared storage configured as described in Preparing the File System for an Enterprise Deployment.

Allow sufficient swap and temporary space; specifically:

- **Swap Space**–The system must have at least 500 MB.
- **Temporary Space**–There must be a minimum of 500 MB of free space in the `/tmp` directory.

Verifying Linux Operating System Requirements

You can review the typical Linux operating system settings for an enterprise deployment in this section.

To ensure the host computers meet the minimum operating system requirements, ensure that you have installed a certified operating system and that you have applied all the necessary patches for the operating system.

In addition, review the following sections for typical Linux operating system settings for an enterprise deployment.

- Setting Linux Kernel Parameters
- Setting the Open File Limit and Number of Processes Settings on UNIX Systems
- Verifying IP Addresses and Host Names in DNS or hosts File

Setting Linux Kernel Parameters

The kernel-parameter and shell-limit values shown in Table 8-1are recommended values only. Oracle recommends that you tune these values to optimize the performance of the system. See your operating system documentation for more information about tuning kernel parameters.

Kernel parameters must be set to a minimum of those in Table 8-1 on all nodes in the topology.

The values in the following table are the current Linux recommendations. For the latest recommendations for Linux and other operating systems, see *Oracle Fusion Middleware System Requirements and Specifications*.

If you deploy a database onto the host, you might need to modify additional kernel parameters. Refer to 12*c* (12.2.1.3.0) Configuring Kernel Parameters in *Oracle Grid Infrastructure Installation Guide for Linux*.

Table 8-1 UNIX Kernel Parameters

Parameter	Value
kernel.sem	256 32000 100 142
kernel.shmmax	4294967295

To set these parameters:

1. Sign in as `root` and add or amend the entries in the `/etc/sysctl.conf` file.
2. Save the file.
3. Activate the changes by entering the following command:

   ```
   /sbin/sysctl -p
   ```

Setting the Open File Limit and Number of Processes Settings on UNIX Systems

On UNIX operating systems, the `Open File Limit` is an important system setting, which can affect the overall performance of the software running on the host computer.

For guidance on setting the `Open File Limit` for an Oracle Fusion Middleware enterprise deployment, see Host Computer Hardware Requirements.

> **Note:**
>
> The following examples are for Linux operating systems. Consult your operating system documentation to determine the commands to be used on your system.

For more information, see the following sections.

- Viewing the Number of Currently Open Files
- Setting the Operating System Open File and Processes Limits

Viewing the Number of Currently Open Files

You can see how many files are open with the following command:

```
/usr/sbin/lsof | wc -l
```

To check your open file limits, use the following commands.

C shell:

```
limit descriptors
```

Bash:

```
ulimit -n
```

Setting the Operating System Open File and Processes Limits

To change the Open File Limit values:

1. Sign in as `root` user and edit the following file:

   ```
   /etc/security/limits.conf
   ```

2. Add the following lines to the`limits.conf` file. (The values shown here are for example only):

```
* soft  nofile  4096
* hard  nofile  65536
* soft  nproc   2047
* hard  nproc   16384
```

The `nofiles` values represent the open file limit; the `nproc` values represent the number of processes limit.

3. Save the changes, and close the`limits.conf` file.

> **Note:**
>
> If you are running Oracle Enterprise Linux 6 or Red Hat Linux 6, locate the following operating system configuration file:`/etc/security/limits.d/90-nproc.conf`
>
> Ensure that the same values are added to the `90-nproc.conf` file. Otherwise, the values in the `90-nproc.conf` file overrides the values in the `limits.conf` file.

4. Re-login into the host computer.

Verifying IP Addresses and Host Names in DNS or hosts File

Before you begin the installation of the Oracle software, ensure that the IP address, fully qualified host name, and the short name of the host are all registered with your DNS server. Alternatively, you can use the local `hosts` file and add an entry similar to the following:

`IP_Address Fully_Qualified_Name Short_Name`

For example:

```
10.229.188.205  host1.example.com  host1
```

Setting the DNS Settings

Configure the host to access your corporate DNS hosts. To do this, update DNS settings by updating the file `/etc/resolv.conf`.

Configuring Operating System Users and Groups

The users and groups to be defined on each of the computers that host the enterprise deployment are listed in this section.

Groups

You must create the following groups on each node.

- `oinstall`
- `dba`

Users

You must create the following user on each node.

- `nobody`–An unprivileged user.
- `oracle`–The owner of the Oracle software. You may use a different name. The primary group for this account must be `oinstall`. The account must also be in the `dba` group.

> **Note:**
>
> - The group `oinstall` must have write privileges to all the file systems on shared and local storage that are used by the Oracle software.
> - Each group must have the same Group ID on every node.
> - Each user must have the same User ID on every node.

Enabling Unicode Support

It is recommended to enable Unicode support in your operating system so as to allow processing of characters in Unicode.

Your operating system configuration can influence the behavior of characters supported by Oracle Fusion Middleware products.

On UNIX operating systems, Oracle highly recommends that you enable Unicode support by setting the `LANG` and `LC_ALL` environment variables to a locale with the UTF-8 character set. This enables the operating system to process any character in Unicode. Oracle WebCenter Content technologies, for example, are based on Unicode.

If the operating system is configured to use a non-UTF-8 encoding, Oracle WebCenter Content components may function in an unexpected way. For example, a non-ASCII file name might make the file inaccessible and cause an error. Oracle does not support problems caused by operating system constraints.

Mounting the Required Shared File Systems on Each Host

It is important to understand how to mount the shared storage to all the servers that require access.

The shared storage configured, as described in Shared Storage Recommendations When Installing and Configuring an Enterprise Deployment, must be available on the hosts that use it.

In an enterprise deployment, it is assumed that you have a hardware storage filer, which is available and connected to each of the host computers that you have procured for the deployment.

You must mount the shared storage to all servers that require access.

Each host must have appropriate privileges set within the Network Attached Storage (NAS) or Storage Area Network (SAN) so that it can write to the shared storage.

Follow the best practices of your organization for mounting shared storage. This section provides an example of how to do this on Linux by using NFS storage.

You must create and mount shared storage locations so that WCCHOST1 and WCCHOST2 can see the same location if it is a binary installation in two separate volumes.

See Shared Storage Recommendations When Installing and Configuring an Enterprise Deployment.

You use the following command to mount shared storage from a NAS storage device to a Linux host. If you are using a different type of storage device or operating system, refer to your manufacturer documentation for information about how to do this.

> **Note:**
>
> The user account used to create a shared storage file system owns and has read, write, and execute privileges for those files. Other users in the operating system group can read and execute the files, but they do not have write privileges.
>
> See Selecting an Installation User in the *Oracle Fusion Middleware Installation Planning Guide*.

In the following example, `nasfiler` represents the shared storage filer. Also note that these are examples only. Typically, the mounting of these shared storage locations should be done by using the `/etc/fstabs` file on UNIX systems, so that the mounting of these devices survives a reboot. Refer to your operating system documentation for more information.

To mount the shared storage on Linux:

1. Create the mount directories on WCCHOST1, as described in Summary of the Shared Storage Volumes in an Enterprise Deployment, and then mount the shared storage. For example:

   ```
   mount -t nfs nasfiler:VOL1/oracle/products/ /u01/oracle/products/
   ```

2. Repeat the procedure on WCCHOST2 using VOL2.

Validating the Shared Storage Configuration

Ensure that you can read and write files to the newly mounted directories by creating a test file in the shared storage location that you just configured.

For example:

```
$ cd newly mounted directory
$ touch testfile
```

Verify that the owner and permissions are correct:

```
$ ls -l testfile
```

Then remove the file:

```
$ rm testfile
```

> **Note:**
>
> The shared storage can be a NAS or SAN device. The following example illustrates creating storage for a NAS device from WCCHOST1. The options may differ depending on the specific storage device.
>
> ```
> mount -t nfs -o
> rw,bg,hard,nointr,tcp,vers=3,timeo=300,rsize=32768,wsize=32768
> nasfiler:VOL1/Oracle/u01/oracle
> ```
>
> Contact your storage vendor and machine administrator to learn about the appropriate options for your environment.

Enabling the Required Virtual IP Addresses on Each Host

You must enable the required virtual IP addresses on each host in order to prepare the host for the enterprise deployment.

To prepare each host for the enterprise deployment, you must enable the virtual IP (VIP) addresses that are described in Reserving the Required IP Addresses for an Enterprise Deployment.

It is assumed that you have already reserved the VIP addresses and host names and that they have been enabled by your network administrator. You can then enable the VIPs on the appropriate host.

Note that the virtual IP addresses used for the enterprise topology are not persisted because they are managed by Whole Server Migration (for selected Managed Servers and clusters) or by manual failover (for the Administration Server).

Starting with Oracle Enterprise Linux 6, the "`ifconfig`" command is deprecated and is replaced with the "`ip`" command.

To enable the VIP addresses on each host, run the following commands as `root`:

1. Determine the CIDR notation of the netmask. Each Netmask has a CIDR notation. For example, 255.255.240.0 has a CIDR of 20.

   If the netmask you are adding is the same as the interface, the fastest way to determine this is to examine the existing IP address that are assigned to the network card. You can do this by using the following command:

   ```
   ip addr show dev eth0
   ```

   Sample output:

   ```
   2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
   pfifo_fast qlen 1000
   link/ether 00:21:f6:03:85:9f brd ff:ff:ff:ff:ff:ff
   int 192.168.20.1/20 brd 10.248.11.255 scope global eth0
   ```

   In this example, the CIDR value is the value after the forward slash (/), which is, 20. If you are unsure of the CIDR value, contact your network administrator.

2. Configure the additional IP address on the appropriate network interface card with an appropriately suffixed label using the following command:

   ```
   ip addr add VIP/CIDR dev nic# label nic#:n
   ```

   > **Note:**
   >
   > For each VIP/VHN that you need to add, increment the :n suffix starting with :1

   Example: For VIP IP of 192.168.20.3, netmask: 255.255.240.0 (CIDR: 20), and the eth0 NIC:

   ```
   ip addr add 192.168.20.3/20 dev eth0 label eth0:1
   ```

3. For each of the virtual IP addresses that you define, update the ARP caches by using the following command:

   ```
   arping -b -A -c 3 -I eth0 192.168.20.3
   ```

9

Preparing the Database for an Enterprise Deployment

Preparing the database for an enterprise deployment involves ensuring that the database meets specific requirements, creating database services, using SecureFiles for large objects in the database, and creating database backup strategies.

This chapter provides information about the database requirements, creating database services, and about the database backup strategies.

- Overview of Preparing the Database for an Enterprise Deployment
  It is important to understand how to configure a supported database as part of an Oracle Fusion Middleware enterprise deployment.
- About Database Requirements
  Before you configure the enterprise deployment topology, you have to verify that the database meets the requirements described in the following sections.
- Creating Database Services
  When multiple Oracle Fusion Middleware products are sharing the same database, each product should be configured to connect to a separate, dedicated database service. This service should be different from the default database service. Having a different service name from the default, allows you to create role based database services for Disaster Recovery and Multi-Datacenter topologies.
- Using SecureFiles for Large Objects (LOBs) in an Oracle Database
  SecureFiles is a new LOB storage architecture introduced in Oracle Database 11g Release 1. It is recommended to use SecureFiles for the Oracle Fusion Middleware schemas, in particular for the Oracle SOA Suite schemas.
- About Database Backup Strategies
  Performing a database backup at key points in the installation and configuration of an enterprise deployment enables you to recover quickly from any issue that might occur in the later configuration steps.

Overview of Preparing the Database for an Enterprise Deployment

It is important to understand how to configure a supported database as part of an Oracle Fusion Middleware enterprise deployment.

Most Oracle Fusion Middleware products require a specific set of schemas that must be installed in a supported database. The schemas are installed by using the Oracle Fusion Middleware Repository Creation Utility (RCU).

In an enterprise deployment, Oracle recommends a highly available Real Application Clusters (Oracle RAC) database for the Oracle Fusion Middleware product schemas.

About Database Requirements

Before you configure the enterprise deployment topology, you have to verify that the database meets the requirements described in the following sections.

- Supported Database Versions
- Additional Database Software Requirements
- Installing and Validating Oracle Text

Supported Database Versions

Use the following information to verify what databases are supported by each Oracle Fusion Middleware release and which version of the Oracle database you are currently running:

- For a list of all certified databases, refer to *Oracle Fusion Middleware Supported System Configurations*.
- To check the release of your database, query the `PRODUCT_COMPONENT_VERSION` view:

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

Oracle Fusion Middleware requires that the database supports the AL32UTF8 character set. Check the database documentation for information on choosing a character set for the database.

For enterprise deployments, Oracle recommends that you use GridLink data sources to connect to Oracle RAC databases.

> **Note:**
>
> For more information about using GridLink data sources and SCAN, see Using Active GridLink Data Sources in *Oracle Fusion Middleware Administering JDBC Data Sources for Oracle WebLogic Server*.
>
> Use of Active GridLink has specific licensing requirements, including a valid WebLogic Suite license. See Oracle Oracle WebLogic Server data sheet.

Additional Database Software Requirements

In the enterprise topology, there are two database host computers in the data tier that host the two instances of the RAC database. These hosts are referred to as DBHOST1 and DBHOST2.

Before you install or configure the enterprise topology, you must ensure that the following software is installed and available on DBHOST1 and DBHOST2:

- **Oracle Clusterware**

  See Installing Oracle Grid Infrastructure for a Cluster in *Oracle Grid Infrastructure Installation Guide for Linux*.

- **Oracle Real Application Clusters**

  See Installing Oracle RAC and Oracle RAC One Node in *Oracle Real Application Clusters Installation Guide for Linux and UNIX*.

- **Time synchronization between Oracle RAC database instances**

  The clocks of the database instances must be in sync if they are used by servers in a Fusion Middleware cluster configured with server migration.

- **Automatic Storage Management** (optional)

  See Introducing Oracle Automatic Storage Management in *Oracle Automatic Storage Management Administrator's Guide*.

Installing and Validating Oracle Text

Before you install or configure the WebCenter Content enterprise topology, you must be sure that Oracle Text is installed and available on DBHOST1 and DBHOST2.

For more information on installing Oracle Text, see the *Oracle Database Installation Guide for Linux*.

To make sure that the database used for WebCenter Content installation has Oracle Text enabled, run the following command:

```
SQL> select comp_name, status, substr(version,1,10) as version from dba_registry
where comp_id = 'CONTEXT';

COMP_NAME STATUS VERSION
----------- ------- -------
Oracle Text VALID 12.1.0.1.0
```

Creating Database Services

When multiple Oracle Fusion Middleware products are sharing the same database, each product should be configured to connect to a separate, dedicated database service. This service should be different from the default database service. Having a different service name from the default, allows you to create role based database services for Disaster Recovery and Multi-Datacenter topologies.

> **Note:**
>
> The instructions in this section are for the Oracle Database 12*c* (12.1) release. If you are using another supported database, refer to the appropriate documentation library for more up-to-date and release-specific information.

For more information about connecting to Oracle databases using services, see Overview of Using Dynamic Database Services to Connect to Oracle Databases in *Real Application Clusters Administration and Deployment Guide*.

In addition, the database service should be different from the default database service. For complete instructions on creating and managing database services for an Oracle Database 12*c* database, see Overview of Automatic Workload Management

with Dynamic Database Services in *Real Application Clusters Administration and Deployment Guide*.

Runtime connection load balancing requires configuring Oracle RAC Load Balancing Advisory with service-level goals for each service for which load balancing is enabled.

You can configure the Oracle RAC Load Balancing Advisory for `SERVICE_TIME` or `THROUGHPUT`. Set the connection load-balancing goal to **SHORT**.

You create and modify Oracle Database services by using the `srvctl` utility.

To create and modify a database service:

1. Add the service to the database and assign it to the instances by using `srvctl`:

   ```
   srvctl add service -db wccdb -service wccedg.example.com -preferred
   wccdb1,wccdb2
   ```

   > **Note:**
   >
   > For the Service Name of the Oracle RAC database, use lowercase letters, followed by the domain name. For example: `wccedg.example.com`

2. Start the service:

   ```
   srvctl start service –db wccdb –service wccedg.example.com
   ```

   > **Note:**
   >
   > For complete instructions on creating and managing database services with SRVCTL, see Creating Services with SRVCTL in the *Real Application Clusters Administration and Deployment Guide*.

3. Modify the service so that it uses the Load Balancing Advisory and the appropriate service-level goals for runtime connection load balancing.

   Use the following resources in the Oracle Database 12*c Real Application Clusters Administration and Deployment Guide* to set the SERVICE_TIME and THROUGHPUT service-level goals:

   - Overview of the Load Balancing Advisory
   - Configuring Your Environment to Use the Load Balancing Advisory

   For example:

   Check the default configuration of the service by using this command:

   ```
   srvctl config service -db wccdb -service wccedg.example.com
   ```

   Several parameters are shown. Check the following parameters:

   - Connection Load Balancing Goal: Long
   - Runtime Load Balancing Goal: NONE

You can modify these parameters by using the following command:

```
srvctl modify service -db wccdb -service wccedg.example.com -
rlbgoal SERVICE_TIME -clbgoal SHORT
```

4. Restart the service:

```
srvctl stop service -db wccdb -service wccedg.example.com
srvctl start service -db wccdb -service wccedg.example.com
```

5. Verify the change in the configuration:

```
srvctl config service -db wccdb -service wccedg.example.com
Runtime Load Balancing Goal: SERVICE_TIME
  Service name: wccedg.example.com
  Service is enabled
  Server pool: wccdb_wccedg.example.com
  ...
  Connection Load Balancing Goal: SHORT
  Runtime Load Balancing Goal: SERVICE_TIME
  ...
```

Using SecureFiles for Large Objects (LOBs) in an Oracle Database

SecureFiles is a new LOB storage architecture introduced in Oracle Database 11g Release 1. It is recommended to use SecureFiles for the Oracle Fusion Middleware schemas, in particular for the Oracle SOA Suite schemas.

Beginning with Oracle Database 11g Release 1, Oracle introduced SecureFiles, a new LOB storage architecture. Oracle recommends that you use SecureFiles for the Oracle Fusion Middleware schemas, in particular for the Oracle SOA Suite schemas. See Using Oracle SecureFiles LOBs in the *Oracle Database SecureFiles and Large Objects Developer's Guide*.

In Oracle 12*c* Databases, the default setting for using SecureFiles is `PREFERRED` . This means that the database attempts to create a SecureFiles LOB unless a BasicFiles LOB is explicitly specified for the LOB or the parent LOB (if the LOB is in a partition or sub-partition). The Oracle Fusion Middleware schemas do not explicitly specify BasicFiles, which means that Oracle Fusion Middleware LOBs defaults to SecureFiles when installed in an Oracle 12*c* database.

For Oracle 11*g* databases, the `db_securefile` system parameter controls the SecureFiles usage policy. This parameter can be modified dynamically. The following options can be used for using SecureFiles:

- `PERMITTED`: Allows SecureFiles to be created (This is the default setting for `db_securefile`. The default storage method uses BasicFiles).
- `FORCE`: Creates all (new) LOBs as SecureFiles.
- `ALWAYS`: Tries to create LOBs as SecureFiles, but falls back to BasicFiles if not possible (if ASSM is disabled).

Other values for the `db_securefile` parameter are:

- `IGNORE`: Ignore attempts to create SecureFiles.
- `NEVER`: Disallow new SecureFiles creations.

For Oracle 11*g* Databases, Oracle recommends that you set the `db_securefile` parameter to `FORCE` before you create the Oracle Fusion Middleware schemas with the Repository Creation Utility (RCU).

Note that the SecureFiles segments require tablespaces managed with automatic segment space management (ASSM). This means that LOB creation on SecureFiles will fail if ASSM is not enabled. However, the Oracle Fusion Middleware tablespaces are created by default with ASSM enabled. As a result, with the default configuration, nothing needs to be changed to enable SecureFiles for the Oracle Fusion Middleware schemas.

About Database Backup Strategies

Performing a database backup at key points in the installation and configuration of an enterprise deployment enables you to recover quickly from any issue that might occur in the later configuration steps.

At key points in the installation and configuration of an enterprise deployment, this guide recommends that you back up your current environment. For example, after you install the product software and create the schemas for a particular Oracle Fusion Middleware product, you should perform a database backup. Performing a backup allows you to perform a quick recovery from any issue that might occur in the later configuration steps.

You can choose to use your own backup strategy for the database, or you can simply make a backup by using operating system tools or RMAN for this purpose.

Oracle recommends that you use Oracle Recovery Manager for the database, particularly if the database was created using Oracle Automatic Storage Management. If possible, you can also perform a cold backup by using operating system tools such as tar.

Part III
Configuring the Enterprise Deployment

The tasks that need to be performed to configure the enterprise deployment topology are detailed in this section.

Part III contains the following chapters:

- Creating the Initial Infrastructure Domain for an Enterprise Deployment
- Configuring the Web Tier for an Enterprise Deployment
- Extending the Domain to Include Oracle WebCenter Content
  You need to perform certain tasks in order to extend the enterprise deployment domain with the Oracle WebCenter Content software. This includes installing the WebCenter Content, extending the domain for WebCenter Content and completing post-configuration and verification tasks.
- Extending the Domain with Oracle SOA Suite
- Extending the Domain to Include Inbound Refinery
- Extending the Domain to Include Capture
  You need to perform certain tasks in order to extend the enterprise deployment domain with Oracle WebCenter Enterprise Capture software.
- Extending the Domain to Include Imaging
  You need to perform certain tasks in order to extend the enterprise deployment domain with Oracle WebCenter Enterprise Imaging software.
- Extending the Domain to Include WebCenter Content User Interface
  You need to perform certain tasks in order to extend the enterprise deployment domain to include Oracle WebCenter Content User Interface software.

10

Creating the Initial Infrastructure Domain for an Enterprise Deployment

It is important to understand how to install and configure an initial domain, which can be used as the starting point for an enterprise deployment. You can extend this initial domain with the various products and components that constitute the enterprise topology you are deploying.
This chapter provides information on variables used when creating the infrastructure domain, creating the database schemas and configuring the infrastructure domain.

- Variables Used When Creating the Infrastructure Domain
  While creating the infrastructure domain, you will be referencing the directory variables listed in this section.
- Understanding the Initial Infrastructure Domain
  Before creating the initial Oracle Fusion Middleware Infrastructure domain, ensure that you review the following key concepts.
- Installing the Oracle Fusion Middleware Infrastructure on WCCHOST1
  Use the following sections to install the Oracle Fusion Middleware Infrastructure software in preparation for configuring a new domain for an enterprise deployment.
- Creating the Database Schemas
  Oracle Fusion Middleware components require the existence of schemas in a database before you configure a Fusion Middleware Infrastructure domain. Install the schemas listed in this topic in a certified database for use with this release of Oracle Fusion Middleware.
- Configuring the Infrastructure Domain
  You can create and configure a WebLogic domain for the enterprise deployment topology using the configuration wizard.
- Starting the Administration Server
  After the domain is created, you can then perform the following tasks on WCCHOST1.
- Creating a New LDAP Authenticator and Provisioning Enterprise Deployment Users and Group
  When you configure an Oracle Fusion Middleware domain, the domain is configured by default to use the WebLogic Server authentication provider (`DefaultAuthenticator`). However, for an enterprise deployment, Oracle recommends that you use a dedicated, centralized LDAP-compliant authentication provider.

Variables Used When Creating the Infrastructure Domain

While creating the infrastructure domain, you will be referencing the directory variables listed in this section.

The directory variables are defined in File System and Directory Variables Used in This Guide.

- *ORACLE_HOME*
- *APPLICATION_HOME*
- *JAVA_HOME*

In addition, you'll be referencing the following virtual IP (VIP) addresses and host names defined in Reserving the Required IP Addresses for an Enterprise Deployment:

- ADMINVHN
- WCCHOST1
- WCCHOST2
- DBHOST1
- DBHOST2
- SCAN Address for the Oracle RAC Database (`DB-SCAN.example.com`)

Understanding the Initial Infrastructure Domain

Before creating the initial Oracle Fusion Middleware Infrastructure domain, ensure that you review the following key concepts.

- About the Infrastructure Distribution
- Characteristics of the Initial Infrastructure Domain

About the Infrastructure Distribution

You create the initial Infrastructure domain for an enterprise deployment by using the Oracle Fusion Middleware Infrastructure distribution. This distribution contains both the Oracle WebLogic Server software and the Oracle JRF software.

The Oracle JRF software consists of Oracle Web Services Manager, Oracle Application Development Framework (Oracle ADF), Oracle Enterprise Manager Fusion Middleware Control, the Repository Creation Utility (RCU), and other libraries and technologies that are required to support the Oracle Fusion Middleware products.

Later in this guide, you can then extend the domain to support the Oracle Fusion Middleware products that are required for your enterprise deployment.

See Understanding Oracle Fusion Middleware Infrastructure in *Understanding Oracle Fusion Middleware*.

Characteristics of the Initial Infrastructure Domain

The following table lists some of the key characteristics of the initial Infrastructure domain. By reviewing and understanding these characteristics, you can better understand the purpose and context of the procedures used to configure the domain.

Many of these characteristics are described in more detail in Understanding a Typical Enterprise Deployment.

Characteristic of the Domain	More Information
Contains only an Administration Server. Managed Servers are added to the domain later, when you extend the initial domain to include Oracle Fusion Middleware products.	About a Typical Enterprise Deployment
Uses a separate virtual IP (VIP) address for the Administration Server.	Configuration of the Administration Server and Managed Servers Domain Directories
Uses a per domain Node Manager configuration.	About the Node Manager Configuration in a Typical Enterprise Deployment
Requires a separately installed LDAP-based authentication provider.	Understanding OPSS and Requests to the Authentication and Authorization Stores

Installing the Oracle Fusion Middleware Infrastructure on WCCHOST1

Use the following sections to install the Oracle Fusion Middleware Infrastructure software in preparation for configuring a new domain for an enterprise deployment.

- Installing a Supported JDK
- Starting the Infrastructure Installer on WCCHOST1
- Navigating the Infrastructure Installation Screens
- Installing Oracle Fusion Middleware Infrastructure on the Other Host Computers
- Checking the Directory Structure
  After you install the Oracle Fusion Middleware Infrastructure and create the Oracle home, you should see the directory and sub-directories listed in this topic. The contents of your installation vary based on the options that you selected during the installation.
- Disabling the Derby Database

Installing a Supported JDK

Oracle Fusion Middleware requires that a certified Java Development Kit (JDK) is installed on your system.

- Locating and Downloading the JDK Software
- Installing the JDK Software

Locating and Downloading the JDK Software

To find a certified JDK, see the certification document for your release on the Oracle Fusion Middleware Supported System Configurations page.

After you identify the Oracle JDK for the current Oracle Fusion Middleware release, you can download an Oracle JDK from the following location on Oracle Technology Network:

```
http://www.oracle.com/technetwork/java/index.html
```

Be sure to navigate to the download for the Java SE JDK.

Installing the JDK Software

Install the JDK onto the `VOL1` and `VOL2` shared storage volumes mounted to `/u01/oracle/products` on the application tier hosts. Name the folder for the JDK without version numbers to avoid re-configuration challenges during JDK upgrades. Example: `/u01/oracle/products/jdk`.

> **Note:**
>
> Multiple installations may be needed as recommended mount points use multiple product shared volumes.

For more information about the recommended location for the JDK software, see the Understanding the Recommended Directory Structure for an Enterprise Deployment.

The following example describes how to install a recent version of JDK 1.8.0_131.

1. Change directory to the location where you downloaded the JDK archive file.

   ```
   cd download_dir
   ```

2. Unpack the archive into the JDK home directory, and then run the following commands:

   ```
   tar -xzvf jdk-8u131-linux-x64.tar.gz
   ```

   Note that the JDK version listed here was accurate at the time this document was published. For the latest supported JDK, see the *Oracle Fusion Middleware System Requirements and Specifications* for the current Oracle Fusion Middleware release.

3. Move the JDK directory to the recommended location in the directory structure.

   For example:

   ```
   mv ./jdk1.8.0_131 /u01/oracle/products/jdk
   ```

   See File System and Directory Variables Used in This Guide.

4. Define the *JAVA_HOME* and *PATH* environment variables for running Java on the host computer.

   For example:

   ```
   export JAVA_HOME=/u01/oracle/products/jdk
   export PATH=$JAVA_HOME/bin:$PATH
   ```

5. Run the following command to verify that the appropriate `java` executable is in the path and your environment variables are set correctly:

   ```
   java -version
   ```

   The Java version in the output should be displayed as `1.8.0_131`.

6. Repeat steps 1 through 5 for each unique *products* shared volume on an appropriate host. For example: WCCHOST1 and WCCHOST2.

Starting the Infrastructure Installer on WCCHOST1

To start the installation program, perform the following steps.

1. Log in to WCCHOST1.
2. Go to the directory where you downloaded the installation program.
3. Launch the installation program by invoking the `java` executable from the JDK directory on your system, as shown in the following example:

   ```
   $JAVA_HOME/bin/java -d64 -jar distribution_file_name.jar
   ```

   In this example:

   - Replace *JAVA_HOME* with the environment variable or actual JDK location on your system.
   - Replace *distribution_file_name* with the actual name of the distribution JAR file.

     If you download the distribution from the Oracle Technology Network (OTN), then the JAR file is typically packaged inside a downloadable compressed file.

     To install the software required for the initial Infrastructure domain, the distribution you want to install is **fmw_12.2.1.3.0_infrastructure.jar**.

     For more information about the actual file names of each distribution, see Identifying and Obtaining Software Downloads for an Enterprise Deployment.

When the installation program appears, you are ready to begin the installation. See Navigating the Installation Screens for a description of each installation program screen.

Navigating the Infrastructure Installation Screens

The installation program displays a series of screens, in the order listed in the following table.

If you need additional help with any of the installation screens, click the screen name or click the **Help** button on the screen.

Table 10-1 Navigating the Infrastructure Installation Screens

Screen	Description
Installation Inventory Setup	On UNIX operating systems, this screen appears if you are installing any Oracle product on this host for the first time. Specify the location where you want to create your central inventory. Ensure that the operating system group name selected on this screen has write permissions to the central inventory location. See Understanding the Oracle Central Inventory in *Oracle Fusion Middleware Installing Software with the Oracle Universal Installer*. **Note:** Oracle recommends that you configure the central inventory directory on the products shared volume. Example: `/u01/oracle/products/oraInventory` You may also need to execute the `createCentralinventory.sh` script as root from the `oraInventory` folder after the installer completes.
Welcome	This screen introduces you to the product installer.
Auto Updates	Use this screen to search My Oracle Support automatically for available patches or automatically search a local directory for patches that you have already downloaded for your organization.
Installation Location	Use this screen to specify the location of your Oracle home directory. For the purposes of an enterprise deployment, enter the value of the *ORACLE_HOME* variable listed in Table 7-2.
Installation Type	Use this screen to select the type of installation and as a consequence, the products and feature sets that you want to install. For this topology, select **Fusion Middleware Infrastructure**. **Note:** The topology in this document does not include server examples. Oracle strongly recommends that you do not install the examples into a production environment.
Prerequisite Checks	This screen verifies that your system meets the minimum requirements. If there are any warning or error messages, refer to the Oracle Fusion Middleware System Requirements and Specifications document on the Oracle Technology Network (OTN).
Security Updates	If you already have an Oracle Support account, use this screen to indicate how you would like to receive security updates. If you do not have one and are sure that you want to skip this step, clear the check box and verify your selection in the follow-up dialog box.

Table 10-1 (Cont.) Navigating the Infrastructure Installation Screens

Screen	Description
Installation Summary	Use this screen to verify the installation options that you have selected. If you want to save these options to a response file, click **Save Response File** and provide the location and name of the response file. Response files can be used later in a silent installation situation. For more information about silent or command-line installation, see Using the Oracle Universal Installer in Silent Mode in *Oracle Fusion Middleware Installing Software with the Oracle Universal Installer*.
Installation Progress	This screen allows you to see the progress of the installation.
Installation Complete	This screen appears when the installation is complete. Review the information on this screen, then click **Finish** to dismiss the installer.

Installing Oracle Fusion Middleware Infrastructure on the Other Host Computers

If you have configured a separate shared storage volume or partition for secondary hosts, then you must install the Infrastructure on one of those hosts.

See Shared Storage Recommendations When Installing and Configuring an Enterprise Deployment.

To install the software on the other host computers in the topology, log in to each host, and use the instructions in Starting the Infrastructure Installer on WCCHOST1 and Navigating the Infrastructure Installation Screens to create the Oracle home on the appropriate storage device.

> **Note:**
>
> In previous releases, the recommended enterprise topology included a colocated set of Oracle HTTP Server instances. In those releases, there was a requirement to install the Infrastructure on the web tier hosts (WEBHOST1 and WEBHOST2). However, for this release, the Enterprise Deployment topology assumes that the web servers are installed and configured in standalone mode, so they are not considered part of the application tier domain. See Configuring the Web Tier for an Enterprise Deployment

Checking the Directory Structure

After you install the Oracle Fusion Middleware Infrastructure and create the Oracle home, you should see the directory and sub-directories listed in this topic. The contents of your installation vary based on the options that you selected during the installation.

To check the directory structure:

1. Change to the *ORACLE_HOME* directory where you installed the Infrastructure.

2. Enter the following command:

   ```
   ls --format=single-column
   ```

   The directory structure on your system must match the structure shown in the following example:

   ```
   cfgtoollogs
   coherence
   em
   inventory
   OPatch
   oracle_common
   oraInst.loc
   oui
   wlserver
   ```

   See What are the Key Oracle Fusion Middleware Directories? in *Understanding Oracle Fusion Middleware*.

Disabling the Derby Database

Disable the embedded Derby database, which is a file-based database, packaged with Oracle WebLogic Server. The Derby database is used primarily for development environments. As a result, you must disable it when you are configuring a production-ready enterprise deployment environment; otherwise, the Derby database process starts automatically when you start the Managed Servers.

To disable the Derby database:

1. Navigate to the following directory in the Oracle home:

   ```
   cd WL_HOME/common/derby/lib
   ```

2. Rename the Derby library jar file:

   ```
   mv derby.jar disable_derby.jar
   ```

3. If each host uses a separate file system, repeat steps 1 and 2 on each host.

Creating the Database Schemas

Oracle Fusion Middleware components require the existence of schemas in a database before you configure a Fusion Middleware Infrastructure domain. Install the schemas listed in this topic in a certified database for use with this release of Oracle Fusion Middleware.

- Metadata Services (MDS)
- Audit Services (IAU)
- Audit Services Append (IAU_APPEND)
- Audit Services Viewer (IAU_VIEWER)
- Oracle Platform Security Services (OPSS)
- User Messaging Service (UMS)
- WebLogic Services (WLS)

- Common Infrastructure Services (STB)

Use the Repository Creation Utility (RCU) to create the schemas. This utility is installed in the Oracle home for each Oracle Fusion Middleware product. For more information about RCU and how the schemas are created and stored in the database, see Preparing for Schema Creation in *Oracle Fusion Middleware Creating Schemas with the Repository Creation Utility*.

Complete the following steps to install the required schemas:

- Installing and Configuring a Certified Database
- Starting the Repository Creation Utility (RCU)
- Navigating the RCU Screens to Create the Schemas
- Verifying Schema Access

Installing and Configuring a Certified Database

Make sure that you have installed and configured a certified database, and that the database is up and running.

See the Preparing the Database for an Enterprise Deployment.

Starting the Repository Creation Utility (RCU)

To start the Repository Creation Utility (RCU):

1. Set the *JAVA_HOME* environment variable so it references the location where you installed a supported JDK.

   See File System and Directory Variables Used in This Guide.

2. Navigate to the following directory on WCCHOST1:

   ```
   ORACLE_HOME/oracle_common/bin
   ```

3. Start RCU:

   ```
   ./rcu
   ```

> **Note:**
>
> If your database has Transparent Data Encryption (TDE) enabled, and you want to encrypt your tablespaces created by the RCU, provide the `-encryptTablespace true` option when you start the RCU.
>
> This will default the appropriate RCU GUI Encrypt Tablespace checkbox selection on the Map Tablespaces screen without further effort during the RCU execution. See Encrypting Tablespaces in *Oracle Fusion Middleware Creating Schemas with the Repository Creation Utility*.

Navigating the RCU Screens to Create the Schemas

Follow the instructions in this section to create the schemas for the Fusion Middleware Infrastructure domain:

- Task 1, Introducing RCU
- Task 2, Selecting a Method of Schema Creation
- Task 3, Providing Database Connection Details
- Task 4, Specifying a Custom Prefix and Selecting Schemas
- Task 5, Specifying Schema Passwords
- Task 6, Verifying the Tablespaces for the Required Schemas
- Task 7, Creating Schemas
- Task 8, Reviewing Completion Summary and Completing RCU Execution

Task 1 Introducing RCU
Review the Welcome screen and verify the version number for RCU. Click **Next** to begin.

Task 2 Selecting a Method of Schema Creation
If you have the necessary permission and privileges to perform DBA activities on your database, select **System Load and Product Load** on the Create Repository screen. The procedure in this document assumes that you have the necessary privileges.
If you do not have the necessary permission or privileges to perform DBA activities in the database, you must select **Prepare Scripts for System Load** on this screen. This option generates a SQL script, which can be provided to your database administrator. See Understanding System Load and Product Load in *Oracle Fusion Middleware Creating Schemas with the Repository Creation Utility*.
Click **Next**.

> **Tip:**
>
> For more information about the options on this screen, see Create repository in *Creating Schemas with the Repository Creation Utility*.

Task 3 Providing Database Connection Details
Provide the database connection details for RCU to connect to your database.

1. In the **Host Name** field, enter the SCAN address of the Oracle RAC Database.
2. Enter the **Port** number of the RAC database scan listener, for example `1521`.
3. Enter the RAC **Service Name** of the database.
4. Enter the **User Name** of a user that has permissions to create schemas and schema objects, for example SYS.
5. Enter the **Password** of the user name that you provided in step 4.
6. If you have selected the SYS user, ensure that you set the role to SYSDBA.
7. Click **Next** to proceed, and then click **OK** on the dialog window confirming that connection to the database was successful.

> **Tip:**
>
> For more information about the options on this screen, see Database Connection Details in *Oracle Fusion Middleware Creating Schemas with the Repository Creation Utility*.

Task 4 Specifying a Custom Prefix and Selecting Schemas

1. Specify the custom prefix that you want to use to identify the Oracle Fusion Middleware schemas.

   The custom prefix is used to logically group these schemas together for use in this domain. For the purposes of this guide, use the prefix `FMW1221_`

   > **Tip:**
   >
   > Make a note of the custom prefix that you choose to enter here; you l need this later, during the domain creation process.
   >
   > For more information about custom prefixes, see Understanding Custom Prefixes in *Creating Schemas with the Repository Creation Utility*.

2. Select **AS Common Schemas**.

   When you select **AS Common Schemas**, all the schemas in this section are automatically selected.

   If the schemas in this section are not automatically selected, then select the required schemas.

There are two mandatory schemas that are selected by default. You cannot deselect them: **Common Infrastructure Services** (the STB schema) and **WebLogic Services** (the WLS schema). The **Common Infrastructure Services** schema enables you to retrieve information from RCU during domain configuration. See Understanding the Service Table Schema in *Oracle Fusion Middleware Creating Schemas with the Repository Creation Utility*.

> **Tip:**
>
> For more information about how to organize your schemas in a multi-domain environment, see Planning Your Schema Creation in *Oracle Fusion Middleware Creating Schemas with the Repository Creation Utility*.

Click **Next** to proceed, and then click **OK** on the dialog window confirming that prerequisite checking for schema creation was successful.

Task 5 Specifying Schema Passwords

Specify how you want to set the schema passwords on your database, then specify and confirm your passwords. Ensure that the complexity of the passwords meet the database security requirements before you continue. RCU proceeds at this point even if you do not meet the password polices. Hence, perform this check outside RCU itself.
Click **Next**.

> **Tip:**
>
> You must make a note of the passwords you set on this screen; you need them later on during the domain creation process.

Task 6 Verifying the Tablespaces for the Required Schemas
You can accept the default settings on the remaining screens, or you can customize how RCU creates and uses the required tablespaces for the Oracle Fusion Middleware schemas.

> **Note:**
>
> You can configure a Fusion Middleware component to use JDBC stores for JMS servers and Transaction Logs, by using the Configuration Wizard. These JDBC stores are placed in the Weblogic Services component tablespace. If your environment expects to have a high level of transactions and JMS activity, you can increase the default size of the <*PREFIX*>_WLS tablespace to better suit the environment load.

Click **Next** to continue, and then click **OK** on the dialog window to confirm the tablespace creation.
For more information about RCU and its features and concepts, see About the Repository Creation Utility in *Oracle Fusion Middleware Creating Schemas with the Repository Creation Utility*.

Task 7 Creating Schemas
Review the summary of the schemas to be loaded and click **Create** to complete schema creation.

> **Note:**
>
> If failures occurred, review the listed log files to identify the root cause, resolve the defects, and then use RCU to drop and recreate the schemas before you continue.

Task 8 Reviewing Completion Summary and Completing RCU Execution
When you reach the Completion Summary screen, verify that all schema creations have been completed successfully, and then click **Close** to dismiss RCU.

Verifying Schema Access

Verify schema access by connecting to the database as the new schema users are created by the RCU. Use SQL*Plus or another utility to connect, and provide the appropriate schema names and passwords entered in the RCU.

For example:

```
./sqlplus

SQL*Plus: Release 12.1.0.2.0 Production on Wed Mar 15 03:17:54 2017
```

```
Copyright (c) 1982, 2014, Oracle.  All rights reserved.

Enter user-name: FMW1221_WLS
Enter password: WLS_schema_password
Last Successful login time: Tue Feb 28 2017 09:37:25 -07:00

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production
With the Real Application Clusters, Automatic Storage Management, OLAP, Advanced
Analytics and Real Application Testing options

SQL>
```

Configuring the Infrastructure Domain

You can create and configure a WebLogic domain for the enterprise deployment topology using the configuration wizard.

For more information on other methods available for domain creation, see "Additional Tools for Creating, Extending, and Managing WebLogic Domains" in *Creating WebLogic Domains Using the Configuration Wizard*.

- Starting the Configuration Wizard
- Navigating the Configuration Wizard Screens to Configure the Infrastructure Domain

Starting the Configuration Wizard

To begin domain configuration, run the following command in the Oracle Fusion Middleware Oracle home on WCCHOST1.

```
ORACLE_HOME/oracle_common/common/bin/config.sh
```

Navigating the Configuration Wizard Screens to Configure the Infrastructure Domain

Follow the instructions in this section to create and configure the domain for the topology.

Task 1 Selecting the Domain Type and Domain Home Location
On the Configuration Type screen, select **Create a New Domain**.
In the **Domain Location** field, specify the value of the *ASERVER_HOME* variable, as defined in File System and Directory Variables Used in This Guide.

> **Tip:**
>
> For more information about the other options on this screen of the Configuration Wizard, see "Configuration Type" in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 2 Selecting the Configuration Templates
On the Templates screen, make sure **Create Domain Using Product Templates** is selected, then select the following templates:

- **Oracle Enterprise Manager - 12.2.1.2.0 [em]**

  Selecting this template automatically selects the following dependencies:

  - **Oracle JRF - 12.2.1.2.0 [oracle_common]**
  - **WebLogic Coherence Cluster Extension - 12.2.1.2.0 [wlserver]**

> **Tip:**
>
> More information about the options on this screen can be found in Templates in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 3 Selecting the Application Home Location
On the Application Location screen, specify the value of the *APPLICATION_HOME* variable, as defined in File System and Directory Variables Used in This Guide.

> **Tip:**
>
> More information about the options on this screen can be found in Application Location in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 4 Configuring the Administrator Account
On the Administrator Account screen, specify the user name and password for the default WebLogic Administrator account for the domain.
Make a note of the user name and password specified on this screen; you will need these credentials later to boot and connect to the domain's Administration Server.

Task 5 Specifying the Domain Mode and JDK
On the Domain Mode and JDK screen:

- Select **Production** in the **Domain Mode** field.
- Select the **Oracle Hotspot** JDK in the **JDK** field.

Selecting **Production Mode** on this screen gives your environment a higher degree of security, requiring a user name and password to deploy applications and to start the Administration Server.

> **Tip:**
>
> More information about the options on this screen, including the differences between development mode and production mode, can be found in Domain Mode and JDK in *Creating WebLogic Domains Using the Configuration Wizard*.
> In production mode, a boot identity file can be created to bypass the need to provide a user name and password when starting the Administration Server. For more information, see Creating the boot.properties File.

Task 6 Specifying the Database Configuration Type
Select **RCU Data** to activate the fields on this screen.
The **RCU Data** option instructs the Configuration Wizard to connect to the database and Service Table (STB) schema to automatically retrieve schema information for the schemas needed to configure the domain.

> **Note:**
>
> If you choose to select **Manual Configuration** on this screen, you will have to manually fill in the parameters for your schema on the JDBC Component Schema screen.

After selecting **RCU Data**, fill in the fields as shown in the following table. Refer to #unique_146/unique_146_Connect_42_BABHHCBF for a partial screen shot of a sample Database Configuration Type screen.

Field	Description
DBMS/Service	Enter the service name for the Oracle RAC database where you will install the product schemas. For example: `orcl.example.com` Be sure this is the common service name that is used to identify all the instances in the Oracle RAC database; do not use the host-specific service name.
Host Name	Enter the Single Client Access Name (SCAN) Address for the Oracle RAC database, which you entered in the *Enterprise Deployment Workbook*.
Port	Enter the port number on which the database listens. For example, 1521.
Schema Owner Schema Password	Enter the user name and password for connecting to the database's Service Table schema. This is the schema user name and password that was specified for the Service Table component on the Schema Passwords screen in RCU (see Creating the Database Schemas). The default user name is *prefix*_STB, where *prefix* is the custom prefix that you defined in RCU.

Click **Get RCU Configuration** when you are finished specifying the database connection information. The following output in the Connection Result Log indicates that the operation succeeded:

```
Connecting to the database server...OK
Retrieving schema data from database server...OK
Binding local schema components with retrieved data...OK

Successfully Done.
```

Click **Next** if the connection to the database is successful.

> **Tip:**
>
> More information about the **RCU Data** option can be found in "Understanding the Service Table Schema" in *Creating Schemas with the Repository Creation Utility*.
> More information about the other options on this screen can be found in Datasource Defaults in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 7 Specifying JDBC Component Schema Information
Verify that the values on the JDBC Component Schema screen are correct for all schemas.
The schema table should be populated because you selected **Get RCU Data** on the previous screen. As a result, the Configuration Wizard locates the database connection values for all the schemas required for this domain.
At this point, the values are configured to connect to a single-instance database. However, for an enterprise deployment, you should use a highly available Real Application Clusters (RAC) database, as described in Preparing the Database for an Enterprise Deployment.
In addition, Oracle recommends that you use an Active GridLink datasource for each of the component schemas. For more information about the advantages of using GridLink data sources to connect to a RAC database, see "Database Considerations" in the *High Availability Guide*.
To convert the data sources to GridLink:

1. Select all the schemas by selecting the check box at in the first header row of the schema table.
2. Click **Convert to GridLink** and click **Next**.

Task 8 Providing the GridLink Oracle RAC Database Connection Details
On the GridLink Oracle RAC Component Schema screen, provide the information required to connect to the RAC database and component schemas, as shown in Table 10-2 and in #unique_146/unique_146_Connect_42_BABJBCFF.

Element	Description and Recommended Value
SCAN, Host Name, and Port	Select the **SCAN** check box. In the **Host Name** field, enter the Single Client Access Name (SCAN) Address for the Oracle RAC database. In the **Port** field, enter the SCAN listening port for the database (for example, 1521).

Element	Description and Recommended Value
ONS Host and Port	In the **ONS Port** field, enter the SCAN address for the Oracle RAC database. In the Port field, enter the ONS Remote port (typically, `6200`).
Enable Fan	Select the **Enable Fan** check box to receive and process FAN events.

For more information about specifying the information on this screen, as well as information about how to identify the correct SCAN address, see "Configuring Active GridLink Data Sources with Oracle RAC" in the *High Availability Guide*.
You can also click **Help** to display a brief description of each field on the screen.

Task 9 Testing the JDBC Connections
Use the JDBC Component Schema Test screen to test the data source connections you have just configured.
A green check mark in the **Status** column indicates a successful test. If you encounter any issues, see the error message in the Connection Result Log section of the screen, fix the problem, then try to test the connection again.

> **Tip:**
>
> More information about the other options on this screen can be found in Test Component Schema in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 10 Selecting Advanced Configuration
To complete domain configuration for the topology, select the following options on the Advanced Configuration screen:

- **Administration Server**

  This is required to properly configure the listen address of the Administration Server.

- **Node Manager**

  This is required to configure Node Manager.

- **Topology**

  This is required to configure the Managed Servers and cluster, and also for configuring the machine and targeting the Managed Servers to the machine.

> **Note:**
>
> When using the Advanced Configuration screen in the Configuration Wizard:
>
> - If any of the above options are not available on the screen, then return to the Templates screen, and be sure you selected the required templates for this topology.
> - Do not select the **Domain Frontend Host Capture** advanced configuration option. You will later configure the frontend host property for specific clusters, rather than for the domain.

Task 11 Configuring the Administration Server Listen Address
On the Administration Server screen:

1. In the **Server Name** field, retain the default value - `AdminServer`.
2. In the **Listen Address** field, enter the virtual host name that corresponds to the VIP of the ADMINVHN that you procured in Procuring Resources for an Enterprise Deployment and enabled in Preparing the Host Computers for an Enterprise Deployment.

   For more information on the reasons for using the ADMINVHN virtual host, see Reserving the Required IP Addresses for an Enterprise Deployment.
3. Leave the other fields at their default values.

   In particular, be sure that no server groups are assigned to the Administration Server.

Task 12 Configuring Node Manager
Select **Per Domain Default Location** as the Node Manager type, then specify the Node Manager credentials you will use to connect to the Node Manager.

> **Tip:**
>
> For more information about the options on this screen, see Node Manager in *Creating WebLogic Domains Using the Configuration Wizard*.
> For more information about per domain and per host Node Manager implementations, see About the Node Manager Configuration in a Typical Enterprise Deployment.
> For additional information, see Configuring Node Manager on Multiple Machines in *Administering Node Manager for Oracle WebLogic Server*.

Task 13 Configuring Managed Servers
There are no Managed Servers in the initial Infrastructure domain. Click **Next** to proceed to the next screen.

Task 14 Configuring a Cluster
There are no clusters in the initial Infrastructure domain. Click **Next** to proceed to the next screen.

Task 15 Configure Server Templates
There are no server templates in the initial Infrastructure domain. Click **Next** to proceed to the next screen.

Task 16 Configure Coherence Clusters
There are no clusters in the initial Infrastructure domain. Click **Next** to proceed to the next screen.

Task 17 Creating Machines
Use the Machines screen to create a new machine in the domain. A machine is required in order for the Node Manager to be able to start and stop the servers.

1. Select the **Unix Machine** tab.
2. Click the **Add** button to create a new Unix machine.

   Use the values in Table 10-3 to define the Name and Node Manager Listen Address of the new machine.
3. Verify the port in the **Node Manager Listen Port** field.

   The port number 5556, shown in this example, may be referenced by other examples in the documentation. Replace this port number with your own port number as needed.

Name	Node Manager Listen Address	Node Manager Listen Port
ADMINHOST	Enter the value of the ADMINVHN variable.	5556

> **Tip:**
>
> More information about the options on this screen can be found in Machines in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 18 Assigning Server To Machine
Complete the following steps:

1. Select admin server from the Server pane.
2. Select `ADMINHOST` from the Machines pane.
3. Click **Next** to proceed to the next screen.

Task 19 Reviewing Virtual Targets
Click **Next** to proceed to the next screen.

Task 20 Reviewing Partitions
Click **Next** to proceed to the next screen.

Task 21 Reviewing Your Configuration Specifications and Configuring the Domain
The Configuration Summary screen contains the detailed configuration information for the domain you are about to create. Review the details of each item on the screen and verify that the information is correct.
You can go back to any previous screen if you need to make any changes, either by using the **Back** button or by selecting the screen in the navigation pane.
Domain creation will not begin until you click **Create**.

> **Tip:**
>
> More information about the options on this screen can be found in Configuration Summary in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 22 Monitoring Configuration Progress
After all the processes are complete, click **Next** to proceed to the next screen.

Task 23 Writing Down Your Domain Home and Administration Server URL
The Configuration Success screen will show the following items about the domain you just configured:

- Domain Location
- Administration Server URL

You must make a note of both items as you will need them later; the domain location is needed to access the scripts used to start the Node Manager and Administration Server, and the URL is needed to access the Administration Server.

Click **Finish** to dismiss the configuration wizard.

Starting the Administration Server

After the domain is created, you can then perform the following tasks on WCCHOST1.

- Starting the Node Manager in the Administration Server Domain Home on WCCHOST1
  Use these steps to start the per-domain Node Manager for the *ASERVER_HOME* domain directory.
- Creating the boot.properties File
  You must create a `boot.properties` if you want to start the Administrator Server without being prompted for the Administrator Server credentials. This step is required in an enterprise deployment. When you start the Administration Server, the credentials that you enter in this file are encrypted.
- Starting the Administration Server Using the Node Manager
  After you have configured the domain and configured the Node Manager, you can start the Administration Server by using the Node Manager. In an enterprise deployment, the Node Manager is used to start and stop the Administration Server and all the Managed Servers in the domain.
- Validating the Administration Server
  Before you proceed with the configuration steps, validate that the Administration Server has started successfully by making sure that you have access to the Oracle WebLogic Server Administration Console and Oracle Enterprise Manager Fusion Middleware Control; both of these are installed and configured on the Administration Servers.

Starting the Node Manager in the Administration Server Domain Home on WCCHOST1

Use these steps to start the per-domain Node Manager for the *ASERVER_HOME* domain directory.

1. Verify that the listen address in the `nodemanager.properties` file is set correctly.
   a. Open the nodemanager.properties file for editing:

      ASERVER_HOME`/nodemanager/nodemanager.properties`

   b. Make sure the `ListenAddress` property is set to the value of the ADMINVHN virtual IP address.
   c. Make sure that QuitEnabled is set to 'true'. If this line is not present in the nodemanager.properties file, add the following line:

      ```
      QuitEnabled=true
      ```

2. Change to the following directory:

   ASERVER_HOME`/bin`

3. Start the Node Manager by entering the following command:

   ```
   nohup ./startNodeManager.sh > ASERVER_HOME/nodemanager/nodemanager.out 2>&1 &
   ```

   For more information about additional Node Manager configuration options, see *Administering Node Manager for Oracle WebLogic Server*.

Creating the boot.properties File

You must create a `boot.properties` if you want to start the Administrator Server without being prompted for the Administrator Server credentials. This step is required in an enterprise deployment. When you start the Administration Server, the credentials that you enter in this file are encrypted.

To create a `boot.properties` file for the Administration Server:

1. Create the following directory structure:

   ```
   mkdir -p ASERVER_HOME/servers/AdminServer/security
   ```

2. In a text editor, create a file called `boot.properties` in the `security` directory that you created in the previous step, and enter the Administration Server credentials that you defined when you ran the Configuration Wizard to create the domain:

   ```
   username=adminuser
   password=password
   ```

> **Note:**
>
> When you start the Administration Server, the `username` and `password` entries in the file are encrypted.
>
> For security reasons, minimize the amount of 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 are encrypted.

3. Save the file and close the editor.

Starting the Administration Server Using the Node Manager

After you have configured the domain and configured the Node Manager, you can start the Administration Server by using the Node Manager. In an enterprise deployment, the Node Manager is used to start and stop the Administration Server and all the Managed Servers in the domain.

To start the Administration Server by using the Node Manager:

1. Start the WebLogic Scripting Tool (WLST):

```
cd ORACLE_COMMON_HOME/common/bin
./wlst.sh
```

2. Connect to Node Manager by using the Node Manager credentials:

```
wls:/
offline>nmConnect('nodemanager_username','nodemanager_password',
            'ADMINVHN','5556','domain_name',
            'ASERVER_HOME')
```

> **Note:**
>
> This user name and password are used only to authenticate connections between Node Manager and clients. They are independent of the server administrator ID and password and are stored in the `nm_password.properties` file located in the following directory:
>
> *ASERVER_HOME*`/config/nodemanager`

3. Start the Administration Server:

```
nmStart('AdminServer')
```

> **Note:**
>
> When you start the Administration Server, it attempts to connect to Oracle Web Services Manager for WebServices policies. It is expected that the WSM-PM Managed Servers are not yet started, and so, the following message appears in the Administration Server log:
>
> ```
> <Warning><oracle.wsm.resources.policymanager>
> <WSM-02141><Unable to connect to the policy access service
> due to Oracle WSM policy manager host server being down.>
> ```

4. Exit WLST:

   ```
   exit()
   ```

Validating the Administration Server

Before you proceed with the configuration steps, validate that the Administration Server has started successfully by making sure that you have access to the Oracle WebLogic Server Administration Console and Oracle Enterprise Manager Fusion Middleware Control; both of these are installed and configured on the Administration Servers.

To navigate to Fusion Middleware Control, enter the following URL, and log in with the Oracle WebLogic Server administrator credentials:

```
ADMINVHN:7001/em
```

To navigate to the Oracle WebLogic Server Administration Console, enter the following URL, and log in with the same administration credentials:

```
ADMINVHN:7001/console
```

Creating a New LDAP Authenticator and Provisioning Enterprise Deployment Users and Group

When you configure an Oracle Fusion Middleware domain, the domain is configured by default to use the WebLogic Server authentication provider (`DefaultAuthenticator`). However, for an enterprise deployment, Oracle recommends that you use a dedicated, centralized LDAP-compliant authentication provider.

The following topics describe how to use the Oracle WebLogic Server Administration Console to create a new authentication provider for the enterprise deployment domain. This procedure assumes that you have already installed and configured a supported LDAP directory, such as Oracle Unified Directory or Oracle Internet Directory.

- About the Supported Authentication Providers
- About the Enterprise Deployment Users and Groups
- Prerequisites for Creating a New Authentication Provider and Provisioning Users and Groups
- Provisioning a Domain Connector User in the LDAP Directory

About the Supported Authentication Providers

Oracle Fusion Middleware supports a variety of LDAP authentication providers. See Identity Store Types and WebLogic Authenticators in *Securing Applications with Oracle Platform Security Services*.

The instructions in this guide assume that you are using one of the following providers:

- Oracle Unified Directory
- Oracle Internet Directory
- Microsoft Active Directory

> **Note:**
>
> By default, the instructions here describe how to configure the identity service instance to support querying against a single LDAP identity store with an unencrypted connection.
>
> If the connection to your identity provider has to be secured through SSL, then additional keystone configuration is required for role management in the Enterprise Manager Fusion Middleware Control to function correctly. For additional configuration information, see Doc ID 1670789.1 at support.oracle.com.
>
> Also, you can configure the service to support a virtualized identity store, which queries multiple LDAP identity stores, by using LibOVD.
>
> For more information about configuring a Multi-LDAP lookup, refer to Configuring the Identity Store Service in *Securing Applications with Oracle Platform Security Services*.

About the Enterprise Deployment Users and Groups

The following topics provide important information on the purpose and characteristics of the enterprise deployment administration users and groups.

About Using Unique Administration Users for Each Domain

When you use a central LDAP user store, you can provision users and groups for use with multiple Oracle WebLogic Server domains. As a result, there is a possibility that one WebLogic administration user can have access to all the domains within an enterprise.

It is a best practice to create and assign a unique distinguished name (DN) within the directory tree for the users and groups that you provision for the administration of your Oracle Fusion Middleware domains.

For example, if you plan to install and configure an Oracle WebCenter Content enterprise deployment domain, then create a user called **`weblogic_wcc`** and an administration group called **`WCCAdministrators`**.

About the Domain Connector User

Oracle recommends that you create a separate domain connector user (for example, `wccLDAP`) in your LDAP directory. This user allows the domain to connect to the LDAP directory for the purposes of user authentication. It is recommended that this user be a non-administrative user.

In a typical Oracle Identity and Access Management deployment, you create this user in the `systemids` container. This container is used for system users that are not normally visible to users. Placing the user into the `systemids` container ensures that customers who have Oracle Identity Governance do not reconcile this user.

A few products, such as IPM require the domain connector user to have the permission to modify data of the LDAP directory. If such products are included, then the domain connector user should be the administrative user.

About Adding Users to the Central LDAP Directory

After you configure a central LDAP directory to be the authenticator for the enterprise domain, then you should add all new users to the new authenticator and not to the default WebLogic Server authenticator.

To add new users to the central LDAP directory, you cannot use the WebLogic Administration Console. Instead, you must use the appropriate LDAP modification tools, such as ldapbrowser or JXplorer.

When you are using multiple authenticators (a requirement for an enterprise deployment), login and authentication will work, but role retrieval will not. The role is retrieved from the first authenticator only. If you want to retrieve roles using any other authenticator, then you must enable virtualization for the domain.

To enable virtualization:

1. Browse to the Fusion Middleware Control, and log in with the administrative credentials.

   `http://adminvhn:7001/em`

2. Navigate to **WebLogic Domain** > **Security** > **Security Provider Configuration**.
3. Expand **Security Store Provider**.
4. Expand **Identity Store Provider**.

5. Click **Configure**.
6. Add a custom property.
7. Select property `virtualize` with value `true` and click **OK**.
8. Select property `user.create.bases`, example value used in this guide is `cn=users,dc=example,dc=com`.
9. Select property `group.create.bases`, example value used in this guide is `cn=groups,dc=example,dc=com`.
10. Click **OK** again to persist the change.
11. Restart the Administration Server and all managed servers.

For more information about the virtualize property, see OPSS System and Configuration Properties in *Oracle Fusion Middleware Securing Applications with Oracle Platform Security Services*.

About Product-Specific Roles and Groups for Oracle WebCenter Content

Each Oracle Fusion Middleware product implements its own predefined roles and groups for administration and monitoring.

As a result, as you extend the domain to add additional products, you can add these product-specific roles to the `WCCAdministrators` group. After they are added to the `WCCAdministrators` group, each product administrator user can administer the domain with the same set of privileges for performing administration tasks.

For instructions on adding additional roles to the `WCCAdministrators` group, see Common Configuration and Management Tasks for an Enterprise Deployment.

Example Users and Groups Used in This Guide

In this guide, the examples assume that you provision the following administration user and group with the following DNs:

- Admin User DN:

  `cn=weblogic_wcc,cn=users,dc=example,dc=com`

- Admin Group DN:

  `cn=WCCAdministrators,cn=groups,dc=example,dc=com`

- Product-specific LDAP Connector User:

  `cn=wccLDAP,cn=systemids,dc=example,dc=com`

  This is the user that you use to connect WebLogic Managed Servers to the LDAP authentication provider. This user must have permissions to read and write to the Directory Trees:

  ```
  cn=users,dc=example,dc=com
  cn=groups,dc=example,dc=com
  ```

> **Note:**
>
> This user needs to be granted membership in the following groups to provide read and write access:
>
> ```
> cn=orclFAUserReadPrivilegeGroup,cn=groups,dc=example,dc=com
> cn=orclFAUserWritePrivilegeGroup,cn=groups,dc=example,dc=com
> cn=orclFAGroupReadPrivilegeGroup,cn=groups,dc=example,dc=com
> cn=orclFAGroupWritePrivilegeGroup,cn=groups,dc=example,dc=com
> ```

Prerequisites for Creating a New Authentication Provider and Provisioning Users and Groups

Before you create a new LDAP authentication provider, back up the relevant configuration files:

```
ASERVER_HOME/config/config.xml
ASERVER_HOME/config/fmwconfig/jps-config.xml
ASERVER_HOME/config/fmwconfig/system-jazn-data.xml
```

In addition, back up the `boot.properties` file for the Administration Server in the following directory:

```
ASERVER_HOME/servers/AdminServer/security
```

- Backing up the Configuration
- Enabling Authentication Provider Virtualization

Backing up the Configuration

Before you create a new LDAP authentication provider, back up the relevant configuration files:

```
ASERVER_HOME/config/config.xml
ASERVER_HOME/config/fmwconfig/jps-config.xml
ASERVER_HOME/config/fmwconfig/system-jazn-data.xml
```

In addition, back up the `boot.properties` file for the Administration Server in the following directory:

```
ASERVER_HOME/servers/AdminServer/security
```

Enabling Authentication Provider Virtualization

When you are using multiple authenticators (a requirement for an enterprise deployment), login and authentication will work, but role retrieval will not. The role is retrieved from the first authenticator only. If you want to retrieve roles using any other authenticator, then you must enable virtualization for the domain.

To enable virtualization:

1. Sign-in to the Fusion Middleware Control by using the administrator's account. For example: `weblogic`.

   `http://adminvhn:7001/em`

2. Click **WebLogic Domain** > **Security** > **Security Provider Configuration**.
3. Expand **Security Store Provider**.
4. Expand **Identity Store Provider**.
5. Click **Configure**.
6. Add a custom property.
7. Select property **virtualize** with value **true** and click **OK**.
8. Click **OK** again to persist the change.
9. Restart the Administration Server and all managed servers.

For more information about the virtualize property, see OPSS System and Configuration Properties in *Oracle Fusion Middleware Securing Applications with Oracle Platform Security Services*.

Provisioning a Domain Connector User in the LDAP Directory

This example shows how to create a user called `wccLDAP` in the central LDAP directory.

To provision the user in the LDAP provider:

1. Create an LDIF file named `domain_user.ldif` with the following contents and then save the file:

```
dn: cn=wccLDAP,cn=systemids,dc=example,dc=com
changetype: add
orclsamaccountname: wccLDAP
userpassword: password
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: inetorgperson
objectclass: orcluser
objectclass: orcluserV2
mail: wccLDAP@example.com
givenname: wccLDAP
sn: wccLDAP
cn: wccLDAP
uid: wccLDAP
```

Note:

If you use Oracle Unified Directory, then add the following four group memberships to the end of the LDIF file to grant the appropriate read/write privileges:

```
dn:
cn=orclFAUserReadPrivilegeGroup,cn=groups,dc=example,dc=com
changetype: modify
add: uniquemember
uniquemember: cn=wccLDAP,cn=systemids,dc=example,dc=com

dn:
cn=orclFAGroupReadPrivilegeGroup,cn=groups,dc=example,dc=com
changetype: modify
add: uniquemember
uniquemember: cn=wccLDAP,cn=systemids,dc=example,dc=com

dn:
cn=orclFAUserWritePrivilegeGroup,cn=groups,dc=example,dc=com
changetype: modify
add: uniquemember
uniquemember: cn=wccLDAP,cn=systemids,dc=example,dc=com

dn:
cn=orclFAGroupWritePrivilegeGroup,cn=groups,dc=example,dc=co
m
changetype: modify
add: uniquemember
uniquemember: cn=wccLDAP,cn=systemids,dc=example,dc=com
```

2. Provision the user in the LDAP directory.

   For example, for an Oracle Unified Directory LDAP provider:

```
OUD_INSTANCE_HOME/bin/ldapmodify -a \
                                 -h idstore.example.com
                                 -D "cn=oudadmin" \
                                 -w password \
                                 -p 1389 \
                                 -f domain_user.ldif
```

   For Oracle Internet Directory:

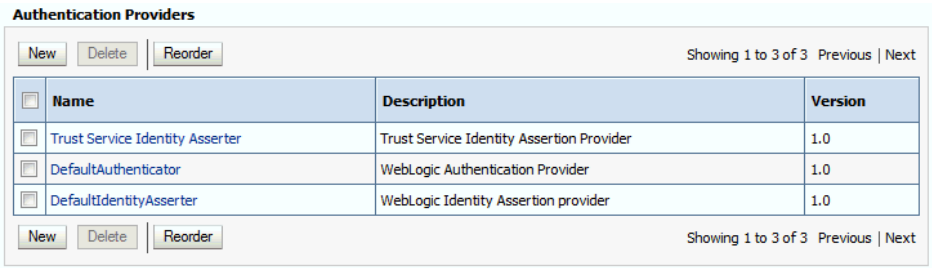
```
OID_ORACLE_HOME/bin/ldapadd -h idstore.example.com \
                            -p 3060 \
                            -D cn="orcladmin" \
                            -w password \
                            -c \
                            -v \
                            -f domain_user.ldif
```

Creating the New Authentication Provider

To configure a new LDAP-based authentication provider:

1. Log in to the WebLogic Server Administration Console.
2. Click **Security Realms** in the left navigational bar.
3. Click the **myrealm** default realm entry.
4. Click the **Providers** tab.

   Note that there is a `DefaultAuthenticator` provider configured for the realm. This is the default WebLogic Server authentication provider.



5. Click **Lock & Edit** in the Change Center.
6. Click the **New** button below the **Authentication Providers** table.
7. Enter a name for the provider.

   Use one of the following names, based on the LDAP directory service that you plan to use as your credential store:

   - `OUDAuthenticator` for Oracle Unified Directory
   - `OIDAuthenticator` for Oracle Internet Directory
   - `OVDAuthenticator` for Oracle Virtual Directory
8. Select the authenticator type from the **Type** drop-down list.

   Select one of the following types, based on the LDAP directory service that you plan to use as your credential store:

   - `OracleUnifiedDirectoryAuthenticator` for Oracle Unified Directory
   - `OracleInternetDirectoryAuthenticator` for Oracle Internet Directory
   - `OracleVirtualDirectoryAuthenticator` for Oracle Virtual Directory
9. Click **OK** to return to the Providers screen.
10. On the Providers screen, click the newly created authenticator in the table.
11. Select **SUFFICIENT** from the **Control Flag** drop-down menu.

Setting the control flag to **SUFFICIENT** indicates that if the authenticator can successfully authenticate a user, then the authenticator should accept that authentication and should not continue to invoke any additional authenticators.

If the authentication fails, it falls through to the next authenticator in the chain. Make sure all subsequent authenticators also have their control flags set to **SUFFICIENT**; in particular, check the **`DefaultAuthenticator`** option and make sure that its control flag is set to **SUFFICIENT**.

12. Click **Save** to save the control flag settings.
13. Click the **Provider Specific** tab and enter the details specific to your LDAP server, as shown in the following table.

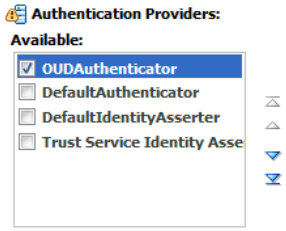
    Note that only the required fields are discussed in this procedure. For information about all the fields on this page, consider the following resources:

    - To display a description of each field, click **Help** on the **Provider Specific** tab.
    - For more information on setting the **User Base DN**, **User From Name Filter**, and **User Attribute** fields, see Configuring Users and Groups in the Oracle Internet Directory and Oracle Virtual Directory Authentication Providers in *Oracle Fusion Middleware Administering Security for Oracle WebLogic Server*.

Parameter	Sample Value	Value Description
Host	For example: `idstore.example.com`	The LDAP server's server ID.
Port	For example: `1389`	The LDAP server's port number.
Principal	For example: `cn=wccLDAP, cn=systemids,dc=example,dc=com`	The LDAP user DN used to connect to the LDAP server.
Credential	Enter LDAP password.	The password used to connect to the LDAP server.
SSL Enabled	Unchecked (clear)	Specifies whether SSL protocol is used when connecting to the LDAP server.
User Base DN	For example: `cn=users,dc=example,dc=com`	Specify the DN under which your users start.
All Users Filter	`(&(uid=*)(objectclass=person))`	Instead of a default search criteria for **All Users Filter**, search all users based on the `uid` value. If the **User Name Attribute** for the user object class in the LDAP directory structure is a type other than `uid`, then change that type in the **User From Name Filter** field. For example, if the **User Name Attribute** type is `cn`, then this field should be set to: `(&(cn=*)(objectclass=person))`

Parameter	Sample Value	Value Description
User From Name Filter	For example: `(&(uid=%u)(objectclass=person))`	If the **User Name Attribute** for the user object class in the LDAP directory structure is a type other than `uid`, then change that type in the settings for the **User From Name** Filter. For example, if the **User Name Attribute** type is `cn`, then this field should be set to: `(&(cn=%u)(objectclass=person))`.
User Name Attribute	For example: `uid`	The attribute of an LDAP user object that specifies the name of the user.
Group Base DN	For example: `cn=groups,dc=example,dc=com`	Specify the DN that points to your Groups node.
Use Retrieved User Name as Principal	Checked	Must be turned on.
GUID Attribute	`entryuuid`	This value is prepopulated with `entryuuid` when `OracleUnifiedDirectoryAuthenticator` is used for OUD. Check this value if you use Oracle Unified Directory as your authentication provider.

14. Click **Save** to save the changes.
15. Click **Security Realms** in the right navigation pane, and then click the default realm name (**myrealm**), and then **Providers** to return to the Providers page.
16. Click **Reorder**, and then use the resulting page to make the Provider you just created first in the list of authentication providers.



17. Click **OK**.
18. On the Providers Page, click **DefaultAuthenticator**.
19. From the Control Flag drop-down, select **SUFFICIENT**.
20. Click **Save** to update the DefaultAuthenticator settings.
21. In the Change Center, click **Activate Changes**.
22. Restart the Administration Server.

    To stop and start the Administration Server using the Node Manager:

    a. Start WLST:

    ```
    cd ORACLE_COMMON_HOME/common/bin
    ./wlst.sh
    ```

b. Connect to Node Manager using the Node Manager credentials you defined in when you created the domain in the Configuration Wizard:

```
wls:/offline>nmConnect('nodemanager_username','nodemanager_password',
            'ADMINVHN','5556','domain_name',
            'ASERVER_HOME')
```

c. Stop the Administration Server:

```
nmKill('AdminServer')
```

d. Start the Administration Server:

```
nmStart('AdminServer')
```

e. Exit WLST:

```
exit()
```

23. After the restart, review the contents of the following log file:

```
ASERVER_HOME/servers/AdminServer/logs/AdminServer.log
```

Verify that no LDAP connection errors occurred. For example, look for errors such as the following:

```
The LDAP authentication provider named "OUDAuthenticator" failed to make
connection to ldap server at ...
```

If you see such errors in the log file, then check the authorization provider connection details to verify that they are correct and try saving and restarting the Administration Server again.

24. After you restart and verify that no LDAP connection errors are in the log file, try browsing the users and groups that exist in the LDAP provider:

In the Administration Console, navigate to the **Security Realms > myrealm > Users and Groups** page. You should be able to see all users and groups that exist in the LDAP provider structure.

Provisioning an Enterprise Deployment Administration User and Group

This example shows how to create a user called **`weblogic_wcc`** and a group called **`WCCAdministrators`**.

To provision the administration user and group in LDAP provider:

1. Create an `LDIF` file named `admin_user.ldif` with the following contents and then save the file:

```
dn: cn=weblogic_wcc,cn=users,dc=example,dc=com
changetype: add
orclsamaccountname: weblogic_wcc
userpassword: password
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: inetorgperson
objectclass: orcluser
objectclass: orcluserV2
mail: weblogic_wcc@example.com
givenname: weblogic_wcc
```

```
sn: weblogic_wcc
cn: weblogic_wcc
uid: weblogic_wcc
```

2. Provision the user in the LDAP directory.

   For example, for an Oracle Unified Directory LDAP provider:

```
OUD_INSTANCE_HOME/bin/ldapmodify -a \
                                 -h idstore.example.com
                                 -D "cn=oudadmin" \
                                 -w password \
                                 -p 1389 \
                                 -f admin_user.ldif
```

   For Oracle Internet Directory:

```
OID_ORACLE_HOME/bin/ldapadd -h idstore.example.com \
                            -p 3060 \
                            -D cn="orcladmin" \
                            -w password \
                            -c \
                            -v \
                            -f admin_user.ldif
```

3. Create an LDIF file named admin_group.ldif with the following contents and then save the file:

```
dn: cn=WCCAdministrators,cn=Groups,dc=example,dc=com
displayname: WCCAdministrators
objectclass: top
objectclass: groupOfUniqueNames
objectclass: orclGroup
uniquemember: cn=weblogic_wcc,cn=users,dc=example,dc=com
cn: WCCAdministrators
uniquemember: cn=wccLDAP, cn=systemids, dc=example, dc=com
cn: WCCAdministrators
description: Administrators Group for the Oracle WebCenter Content
Domain
```

4. Provision the group in the LDAP Directory.

   For Oracle Unified Directory:

```
OUD_INSTANCE_HOME/bin/ldapmodify -a \
                                 -D "cn=oudadmin" \
                                 -h oudhost.example.com \
                                 -w password \
                                 -p 1380 \
                                 -f admin_group.ldif
```

   For Oracle Internet Directory:

```
OID_ORACLE_HOME/bin/ldapadd -h oidhost.example.com \
                           -p 3060 \
                           -D cn="orcladmin" \
                           -w password \
                           -c \
                           -v \
                           -f admin_group.ldif
```

5. Verify that the changes were made successfully:
   a. Log in to the Oracle WebLogic Server Administration Console.
   b. In the left pane of the console, click **Security Realms**.
   c. Click the default security realm (**myrealm**).
   d. Click the **Users and Groups** tab.
   e. Verify that the administrator user and group that you provisioned are listed on the page.

Adding the Administration Role to the New Administration Group

After you add the users and groups to Oracle Internet Directory, the group must be assigned the Administration role within the WebLogic domain security realm. This enables all users that belong to the group to be administrators for the domain.

To assign the Administration role to the new enterprise deployment administration group:

1. Log in to the WebLogic Administration Server Console by using the administration credentials that you provided in the Configuration Wizard.

   Do not use the credentials for the administration user that you created and provided for the new authentication provider.
2. In the left pane of the Administration Console, click **Security Realms**.
3. Click the default security realm (**myrealm**).
4. Click the **Roles and Policies** tab.
5. Expand the **Global Roles** entry in the table and click **Roles**.

Figure 10-1 Global Roles Under Security Realms

6. Click the **Admin** role.

Figure 10-2 Adding Conditions for the Admin Role

7. Click **Add conditions**.
8. Select **Group** from the **Predicate List** drop-down menu, and then click **Next**.
9. Enter `WCCAdministrators` in the **Group Argument Name** field, and then click **Add**.

`WCCAdministrators` is added to the list box of arguments.

10. Click **Finish** to return to the Edit Global Role page.

    The `WCCAdministrators` group is now listed.

11. Click **Save** to finish adding the **Admin** Role to the `WCCAdministrators` group.

12. Validate that the changes were made by logging in to the WebLogic Administration Server Console by using the new `weblogic_wcc` user credentials.

    If you can log in to the Oracle WebLogic Server Administration Console and Fusion Middleware Control with the credentials of the new administration user that you just provisioned in the new authentication provider, then you have configured the provider successfully.

Updating the boot.properties File and Restarting the System

After you create the new administration user and group, you must update the Administration Server `boot.properties` file with the administration user credentials that you created in the LDAP directory:

1. On WCCHOST1, go the following directory:

   ASERVER_HOME`/servers/AdminServer/security`

2. Rename the existing `boot.properties` file:

   ```
   mv boot.properties boot.properties.backup
   ```

3. Use a text editor to create a file called `boot.properties` under the security directory.

4. Enter the following lines in the file:

   ```
   username=weblogic_wcc
   password=password
   ```

5. Save the file.

6. Restart the Administration Server.

11

Configuring the Web Tier for an Enterprise Deployment

It is important to understand how to install and configure a standalone Oracle HTTP Server domain that contains two Oracle HTTP Server instances: one on WEBHOST1 and one on WEBHOST2.
This chapter provides information on variables used when configuring the web tier and installing and configuring a web tier domain.

- Variables Used When Configuring the Web Tier
  While configuring the web tier, you will be referencing the directory variables listed in this section.
- About the Web Tier Domains
  In an enterprise deployment, each Oracle HTTP Server instance is configured on a separate host and in its own standalone domain. This allows for a simple configuration that requires a minimum amount of configuration and a minimum amount of resources to run and maintain.
- Installing a Supported JDK
- Installing Oracle HTTP Server on WEBHOST1
  It is important to understand the procedure for installing the Oracle HTTP Server software on the web tier.
- Creating a Web Tier Domain on WEBHOST1
  It is essential to understand how to create a new Oracle HTTP Server standalone domain on the first Web tier host.
- Installing and Configuring a Web Tier Domain on WEBHOST2
  After you install Oracle HTTP Server and configure a Web Tier domain on WEBHOST1, then you must also perform the same tasks on WEBHOST2.
- Starting the Node Manager and Oracle HTTP Server Instances on WEBHOST1 and WEBHOST2
  It is important to understand how to start the Oracle HTTP Server instances on WEBHOST1 and WEBHOST2.
- Configuring Oracle HTTP Server to Route Requests to the Application Tier
  It is important to understand how to update the Oracle HTTP Server configuration files so that the web server instances route requests to the servers in the domain.

Variables Used When Configuring the Web Tier

While configuring the web tier, you will be referencing the directory variables listed in this section.

The values for several directory variables are defined in File System and Directory Variables Used in This Guide.

- *OHS_ORACLE_HOME*
- *OHS_DOMAIN_HOME*

In addition, you'll be referencing the following virtual IP (VIP) address and host names:

- ADMINVHN
- WEBHOST1
- WEBHOST2

About the Web Tier Domains

In an enterprise deployment, each Oracle HTTP Server instance is configured on a separate host and in its own standalone domain. This allows for a simple configuration that requires a minimum amount of configuration and a minimum amount of resources to run and maintain.

For more information about the role and configuration of the Oracle HTTP Server instances in the web tier, see Understanding the Web Tier.

Installing a Supported JDK

Oracle Fusion Middleware requires that a certified Java Development Kit (JDK) is installed on your system.

- Locating and Downloading the JDK Software
- Installing the JDK Software
  Oracle Fusion Middleware requires you to install a certified Java Development Kit (JDK) on your system.

Locating and Downloading the JDK Software

To find a certified JDK, see the certification document for your release on the Oracle Fusion Middleware Supported System Configurations page.

After you identify the Oracle JDK for the current Oracle Fusion Middleware release, you can download an Oracle JDK from the following location on Oracle Technology Network:

```
http://www.oracle.com/technetwork/java/index.html
```

Be sure to navigate to the download for the Java SE JDK.

Installing the JDK Software

Oracle Fusion Middleware requires you to install a certified Java Development Kit (JDK) on your system.

You must install the JDK in the following locations:

On the local storage device for each of the Web tier host computers. The Web tier host computers, which reside in the DMZ, do not necessarily have access to the shared storage on the application tier.

See the Understanding the Recommended Directory Structure for an Enterprise Deployment.

To install JDK 1.8.0_131:

1. Change directory to the location where you downloaded the JDK archive file.

   ```
   cd download_dir
   ```

2. Unpack the archive into the JDK home directory, and then run the following commands:

   ```
   tar -xzvf jdk-8u131-linux-x64.tar.gz
   ```

   Note that the JDK version listed here was accurate at the time this document was published. For the latest supported JDK, see the *Oracle Fusion Middleware System Requirements and Specifications* for the current Oracle Fusion Middleware release.

3. Move the JDK directory to the recommended location in the directory structure.

   For example:

   ```
   mv ./jdk1.8.0_131 /u02/oracle/products/jdk
   ```

   See File System and Directory Variables Used in This Guide.

4. Define the *JAVA_HOME* and *PATH* environment variables for running Java on the host computer.

   For example:

   ```
   export JAVA_HOME=/u02/oracle/products/jdk
   export PATH=$JAVA_HOME/bin:$PATH
   ```

5. Run the following command to verify that the appropriate `java` executable is in the path and your environment variables are set correctly:

   ```
   java -verison
   ```

   The Java version in the output should be displayed as `1.8.0_131`.

Installing Oracle HTTP Server on WEBHOST1

It is important to understand the procedure for installing the Oracle HTTP Server software on the web tier.

- Starting the Installer on WEBHOST1
- Navigating the Oracle HTTP Server Installation Screens
- Verifying the Oracle HTTP Server Installation

Starting the Installer on WEBHOST1

To start the installation program, perform the following steps.

1. Log in to WEBHOST1.
2. Go to the directory in which you downloaded the installation program.
3. Enter the following command to launch the installation program:

   ```
   ./fmw_12.2.1.3.0_ohs_linux64.bin
   ```

When the installation program appears, you are ready to begin the installation.

Navigating the Oracle HTTP Server Installation Screens

The following table lists the screens in the order that the installation program displays them.

If you need additional help with any of the installation screens, click the screen name.

Table 11-1 Oracle HTTP Server Installation Screens

Screen	Description
Installation Inventory Setup	On UNIX operating systems, this screen appears if you install any Oracle product on this host for the first time. Specify the location where you want to create your central inventory. Ensure that the operating system group name selected on this screen has write permissions to the central inventory location. See Understanding the Oracle Central Inventory in *Oracle Fusion Middleware Installing Software with the Oracle Universal Installer*. **Note:** Oracle recommends that you configure the central inventory directory within the products directory. Example: `/u02/oracle/products/oraInventory` You may also need to execute the `createCentralInventory.sh` script as root from the `oraInventory` folder after the installer completes.
Welcome	This screen introduces you to the product installer.
Auto Updates	Use this screen to automatically search My Oracle Support for available patches or automatically search the local directory for patches that you have already downloaded for your organization.
Installation Location	Use this screen to specify the location of your Oracle home directory. For the purposes of an enterprise deployment, enter the value of the *WEB_ORACLE_HOME* variable listed in Table 7-3.

Table 11-1 (Cont.) Oracle HTTP Server Installation Screens

Screen	Description
Installation Type	Select **Standalone HTTP Server (Managed independently of WebLogic server)**. This installation type allows you to configure the Oracle HTTP Server instances independently from any other existing Oracle WebLogic Server domains.
JDK Selection	For the value of JDK Home, enter the value of *JAVA_HOME* that you set when installing the JDK software.
Prerequisite Checks	This screen verifies that your system meets the minimum necessary requirements. If there are any warning or error messages, verify that your host computers and the required software meet the system requirements and certification information described in Host Computer Hardware Requirements and Operating System Requirements for the Enterprise Deployment Topology.
Installation Summary	Use this screen to verify the installation options that you selected. If you want to save these options to a response file, click **Save Response File** and provide the location and name of the response file. Response files can be used later in a silent installation situation. See Using the Oracle Universal Installer in Silent Mode in *Oracle Fusion Middleware Installing Software with the Oracle Universal Installer*.
Installation Progress	This screen allows you to see the progress of the installation.
Installation Complete	This screen appears when the installation is complete. Review the information on this screen, then click **Finish** to close the installer.

Verifying the Oracle HTTP Server Installation

Verify that the Oracle HTTP Server installation completed successfully by validating the `WEB_ORACLE_HOME` folder contents.

Run the following command to compare the installed folder structure with the following list:

```
ls --format=single-column WEB_ORACLE_HOME
```

The following files and directories are listed in theOracle HTTP Server Oracle Home:

```
bin
cdata
cfgtoollogs
crs
css
cv
has
```

```
install
inventory
jlib
ldap
lib
network
nls
ohs
OPatch
oracle_common
oracore
oraInst.loc
oui
perl
plsql
plugins
precomp
QOpatch
racg
rdbms
slax
sqlplus
srvm
webgate
wlserver
xdk
```

Creating a Web Tier Domain on WEBHOST1

It is essential to understand how to create a new Oracle HTTP Server standalone domain on the first Web tier host.

Starting the Configuration Wizard on WEBHOST1

To start the Configuration Wizard, navigate to the following directory and start the WebLogic Server Configuration Wizard, as follows:

```
cd WEB_ORACLE_HOME/oracle_common/common/bin
./config.sh
```

Navigating the Configuration Wizard Screens for an Oracle HTTP Server Domain

Oracle recommends that you create a standalone domain for the Oracle HTTP Server instances on each web tier host.

The following topics describe how to create a new standalone Oracle HTTP Server domain:

- Task 1, Selecting the Domain Type and Domain Home Location
- Task 2, Selecting the Configuration Templates
- Task 3, Selecting the JDK for the Web Tier Domain.
- Task 4, Configuring System Components
- Task 5, Configuring OHS Server
- Task 7, Reviewing Your Configuration Specifications and Configuring the Domain
- Task 8, Writing Down Your Domain Home

Task 1 Selecting the Domain Type and Domain Home Location
On the Configuration Type screen, select **Create a new domain**.
In the **Domain Location** field, enter the value assigned to the *WEB_DOMAIN_HOME* variable.
Note the following:

- The Configuration Wizard creates the new directory that you specify here.
- Create the directory on local storage, so the web servers do not have any dependencies on storage devices outside the DMZ.

Task 2 Selecting the Configuration Templates
On the Templates screen, select **Oracle HTTP Server (Standalone) - 12.2.1.3.0 [ohs]**.

> **Tip:**
>
> More information about the options on this screen can be found in Templates in *Oracle Fusion Middleware Creating WebLogic Domains Using the Configuration Wizard*.

Task 3 Selecting the JDK for the Web Tier Domain.
Select the Oracle HotSpot JDK installed in the `/u02/oracle/products/jdk` directory prior to the Oracle HTTP Server installation.

Task 4 Configuring System Components
On the System Components screen, configure one Oracle HTTP Server instance. The screen should, by default, have a single instance defined. This is the only instance that you need to create.

1. The default instance name in the **System Component** field is `ohs1`. Use this default name when you configure `WEBHOST1`.
2. Make sure that `OHS` is selected in the **Component Type** field.
3. If an application is not responding, use the **Restart Interval Seconds** field to specify the number of seconds to wait before you attempt a restart if an application is not responding.
4. Use the **Restart Delay Seconds** field to specify the number of seconds to wait between restart attempts.

Task 5 Configuring OHS Server
Use the OHS Server screen to configure the OHS servers in your domain:

1. Select **ohs1** from the **System Component** drop-down menu.
2. In the **Listen Address** field, enter *WEBHOST1*.

   All the remaining fields are prepopulated, but you can change the values as required for your organization. See OHS Server in *Oracle Fusion Middleware Creating WebLogic Domains Using the Configuration Wizard*.
3. In the **Server Name** field, verify the value of the listen address and listen port.

   It should appear as follows:

   ```
   http://WEBHOST1:7777
   ```

Task 6 Configuring Node Manager
Select **Per Domain Default Location** as the Node Manager type, and specify the user name and password for the Node Manager.

> **Note:**
>
> For more information about the options on this screen, see Node Manager in *Oracle Fusion Middleware Creating WebLogic Domains Using the Configuration Wizard*.
> For information about Node Manager configuration, see Configuring Node Manager on Multiple Machines in *Oracle Fusion Middleware Administering Node Manager for Oracle WebLogic Server*.

Task 7 Reviewing Your Configuration Specifications and Configuring the Domain
The Configuration Summary screen contains detailed configuration information for the domain that you are about to create. Review the details of each item on the screen and verify that the information is correct.
If you need to make any changes, you can go back to any previous screen either by using the **Back** button or by selecting the screen in the navigation pane.
Domain creation does not begin until you click **Create**.
In the Configuration Progress screen, click **Next** when it finishes.

> **Tip:**
>
> More information about the options on this screen can be found in Configuration Summary in *Oracle Fusion Middleware Creating WebLogic Domains Using the Configuration Wizard*.

Task 8 Writing Down Your Domain Home
The Configuration Success screen shows the domain home location.
Make a note of the information provided here, as you need it to start the servers and access the Administration Server.
Click **Finish** to close the Configuration Wizard.

Installing and Configuring a Web Tier Domain on WEBHOST2

After you install Oracle HTTP Server and configure a Web Tier domain on WEBHOST1, then you must also perform the same tasks on WEBHOST2.

1. Log in to WEBHOST2 and install Oracle HTTP Server, using the instructions in Installing Oracle HTTP Server on WEBHOST1.
2. Configure a new standalone domain on WEBHOST2, using the instructions in Creating a Web Tier Domain on WEBHOST1.

   Use the name `ohs2` for the instance on WEBHOST2, and be sure to replace all occurrences of WEBHOST1 with WEBHOST2 and all occurrences of `ohs1` with `ohs2` in each of the examples.

Starting the Node Manager and Oracle HTTP Server Instances on WEBHOST1 and WEBHOST2

It is important to understand how to start the Oracle HTTP Server instances on WEBHOST1 and WEBHOST2.

- Starting the Node Manager on WEBHOST1 and WEBHOST2
- Starting the Oracle HTTP Server Instances

Starting the Node Manager on WEBHOST1 and WEBHOST2

Before you can start the Oracle HTTP Server instances, you must start the Node Manager on WEBHOST1 and WEBHOST2:

1. Log in to WEBHOST1 and navigate to the following directory:

   WEB_DOMAIN_HOME`/bin`

2. Start the Node Manager as shown in the following sections by using `nohup` and `nodemanager.out` as an example output file:

   ```
   nohup WEB_DOMAIN_HOME/bin/startNodeManager.sh > WEB_DOMAIN_HOME/nodemanager/
   nodemanager.out 2>&1 &
   ```

3. Log in to WEBHOST2 and perform steps 1 and 2.

See Advanced Node Manager Configuration in *Oracle Fusion Middleware Administering Node Manager for Oracle WebLogic Server*.

Starting the Oracle HTTP Server Instances

To start the Oracle HTTP Server instances:

1. Navigate to the following directory on WEBHOST1:

   WEB_DOMAIN_HOME`/bin`

For more information about the location of the WEB_DOMAIN_HOME directory, see File System and Directory Variables Used in This Guide.

2. Enter the following command:

```
./startComponent.sh ohs1
```

> **Note:**
>
> Every time you start the Oracle HTTP server, you will be asked for the Node Manager password. If you do not wish this behaviour, then use the following command the first time you start the Oracle HTTP server:
>
> ```
> ./startComponent.sh ohs1 storeUserConfig
> ```
>
> This time when you enter the Node Manager password, it will be encrypted and stored. Future start and stop of the Oracle HTTP server will not require you to enter the Node Manager password.

3. When prompted, enter the Node Manager password.
4. Repeat steps 1 through 3 to start the `ohs2` instance on WEBHOST2. See Starting Oracle HTTP Server Instances in *Administering Oracle HTTP Server*.

Configuring Oracle HTTP Server to Route Requests to the Application Tier

It is important to understand how to update the Oracle HTTP Server configuration files so that the web server instances route requests to the servers in the domain.

- About the Oracle HTTP Server Configuration for an Enterprise Deployment
- Modifying the httpd.conf File to Include Virtual Host Configuration Files
- Creating the Virtual Host Configuration Files
- Validating the Virtual Server Configuration on the Load Balancer
- Configuring Routing to the Administration Server
- Turning On the WebLogic Server Plug-In Enabled Flag
- Setting the Front-End URL for the Administration Console and Setting Redirection Preferences
- Validating Access to the Management Consoles and Administration Server

About the Oracle HTTP Server Configuration for an Enterprise Deployment

The following topics provide overview information about the changes that are required to the Oracle HTTP Server configuration files in an enterprise deployment.

- Purpose of the Oracle HTTP Server Virtual Hosts
- Recommended Structure of the Oracle HTTP Server Configuration Files

Purpose of the Oracle HTTP Server Virtual Hosts

The reference topologies in this guide require that you define a set of virtual servers on the hardware load balancer. You can then configure Oracle HTTP Server to recognize requests to specific virtual hosts (that map to the load balancer virtual servers) by adding `<VirtualHost>` directives to the Oracle HTTP Server instance configuration files.

For each Oracle HTTP Server virtual host, you define a set of specific URLs (or context strings) that route requests from the load balancer through the Oracle HTTP Server instances to the appropriate Administration Server or Managed Server in the Oracle WebLogic Server domain.

Recommended Structure of the Oracle HTTP Server Configuration Files

Rather than adding multiple virtual host definitions to the `httpd.conf` file, Oracle recommends that you create separate, smaller, and more specific configuration files for each of the virtual servers required for the products that you are deploying. This avoids populating an already large `httpd.conf` file with additional content, and it can make troubleshooting configuration problems easier.

For example, in a typical Oracle Fusion Middleware Infrastructure domain, you can add a specific configuration file called `admin_vh.conf` that contains the virtual host definition for the Administration Server virtual host (ADMINVHN).

Modifying the httpd.conf File to Include Virtual Host Configuration Files

Perform the following tasks to prepare the `httpd.conf` file for the additional virtual hosts required for an enterprise topology:

1. Log in to WEBHOST1.
2. Locate the `httpd.conf` file for the first Oracle HTTP Server instance (`ohs1`) in the domain directory:

   ```
   cd WEB_DOMAIN_HOME/config/fmwconfig/components/OHS/ohs1/
   ```

3. Verify if the `httpd.conf` file has the appropriate configuration as follows:
   a. Run the following command to verify the `ServerName` parameter, be sure that it is set correctly, substituting the correct value for the current WEBHOST*n*:

      ```
      grep "ServerName http" httpd.conf
      ServerName http://WEBHOST1:7777
      ```

   b. Run the following command to verify there is an include statement that includes all `*.conf` files from the moduleconf subdirectory:

      ```
      grep moduleconf httpd.conf
      IncludeOptional "moduleconf/*.conf"
      ```

c. If either validation fails to return results, or returns results that are commented out, open the `httpd.conf` file in a text editor and make the required changes in the appropriate locations.

```
#
# ServerName gives the name and port that the server uses to
identify itself.
# This can often be determined automatically, but we recommend
you specify
# it explicitly to prevent problems during startup.
#
# If your host doesn't have a registered DNS name, enter its IP
address here.
#
ServerName http://WEBHOST1:7777
#  and at the end of the file:
# Include the admin virtual host (Proxy Virtual Host) related
configuration
include "admin.conf"
IncludeOptional "moduleconf/*.conf"
```

d. Save the `httpd.conf` file.

4. Log in to `WEBHOST2` and perform steps 2 and 3 for the `httpd.conf` file, replacing any occurrences of `WEBHOST1` or `ohs1` with `WEBHOST2` or `ohs2` in the instructions as necessary.

Creating the Virtual Host Configuration Files

To create the virtual host configuration files:

> **Note:**
>
> Before you create the virtual host configuration files, be sure that you have configured the virtual servers on the load balancer, as described in Purpose of the Oracle HTTP Server Virtual Hosts.

1. Log in to WEBHOST1 and change directory to the configuration directory for the first Oracle HTTP Server instance (`ohs1`):

```
cd WEB_DOMAIN_HOME/config/fmwconfig/components/OHS/ohs1/moduleconf
```

2. Create the `admin_vh.conf` file and add the following directive:

```
<VirtualHost WEBHOST1:7777>
    ServerName admin.example.com:80
    ServerAdmin you@your.address
    RewriteEngine On
    RewriteOptions inherit
</VirtualHost>
```

3. Create the `wccinternal_vh.conf` file and add the following directive:

```
<VirtualHost WEBHOST1:7777>
    ServerName wccinternal.example.com:80
    ServerAdmin you@your.address
```

```
    RewriteEngine On
    RewriteOptions inherit
</VirtualHost>
```

4. Restart the `ohs1` instance:
   a. Change directory to the following location:

      ```
      cd WEB_DOMAIN_HOME/bin
      ```

   b. Enter the following commands to stop and start the instance; provide the node manager password when prompted:

      ```
      ./stopComponent.sh ohs1
      ./startComponent.sh ohs1
      ```

5. Copy the `admin_vh.conf` file and the `wccinternal_vh.conf` file to the configuration directory for the second Oracle HTTP Server instance (`ohs2`) on WEBHOST2:

   ```
   WEB_DOMAIN_HOME/config/fmwconfig/components/OHS/ohs2/moduleconf
   ```

6. Edit the `admin_vh.conf` and `wccinternal_vh.conf` files and change any references from WEBHOST1 to WEBHOST2 in the `<VirtualHost>` directives.
7. Restart the `ohs2` instance:
   a. Change directory to the following location:

      ```
      cd WEB_DOMAIN_HOME/bin
      ```

   b. Enter the following commands to stop and start the instance:

      ```
      ./stopComponent.sh ohs2
      ./startComponent.sh ohs2
      ```

Validating the Virtual Server Configuration on the Load Balancer

From the load balancer, access the following URLs to ensure that your load balancer and Oracle HTTP Server are configured properly. These URLs should show the initial Oracle HTTP Server 12*c* web page.

- `http://admin.example.com/index.html`
- `http://wccinternal.example.com/index.html`

Configuring Routing to the Administration Server

To enable Oracle HTTP Server to route to the Administration Server, you must add a set of `<Location>` directives to the Oracle HTTP Server `admin_vh.conf` configuration file:

1. Log in to WEBHOST1, and change directory to the following location:

   ```
   cd OHS_DOMAIN_HOME/config/fmwconfig/components/OHS/ohs1/moduleconf/
   ```

2. Add the following directives to the `admin_vh.conf` file within the `<VirtualHost>` tags:

   ```
   # Admin Server and EM
   <Location /console>
       SetHandler weblogic-handler
       WebLogicHost ADMINVHN
       WeblogicPort 7001
   </Location>
   ```

```
<Location /consolehelp>
    SetHandler weblogic-handler
    WebLogicHost ADMINVHN
    WeblogicPort 7001
</Location>

<Location /em>
    SetHandler weblogic-handler
    WebLogicHost ADMINVHN
    WeblogicPort 7001
</Location>
```

The `admin_vh.conf` file should appear as it does in Example 11-1.

3. Restart the `ohs1` instance:
   a. Change directory to the following location:

      ```
      cd OHS_DOMAIN_HOME/bin
      ```

   b. Enter the following commands to stop and start the instance:

      ```
      ./stopComponent.sh ohs1
      ./startComponent.sh ohs1
      ```

4. Copy the `admin_vh.conf` file to the configuration directory for the second Oracle HTTP Server instance (`ohs2`) on WEBHOST2:

   OHS_DOMAIN_HOME`/config/fmwconfig/components/OHS/`**ohs2**`/moduleconf/`

5. Edit the `admin_vh.conf` file and change any references of WEBHOST1 to WEBHOST2 in the `<VirtualHost>` directives.

6. Restart the `ohs2` instance:
   a. Change directory to the following location:

      ```
      cd OHS_DOMAIN_HOME/bin
      ```

   b. Enter the following commands to stop and start the instance:

      ```
      ./stopComponent.sh ohs2
      ./startComponent.sh ohs2
      ```

Example 11-1 Sample Contents of the admin_vh.conf File

```
<VirtualHost WEBHOST1:7777>
    ServerName admin.example.com:7777
    ServerAdmin you@your.address
    RewriteEngine On
    RewriteOptions inherit

# Admin Server and EM
<Location /console>
    SetHandler weblogic-handler
    WebLogicHost ADMINVHN
    WeblogicPort 7001
</Location>

<Location /consolehelp>
    SetHandler weblogic-handler
    WebLogicHost ADMINVHN
    WeblogicPort 7001
</Location>
```

```
<Location /em>
    SetHandler weblogic-handler
    WebLogicHost ADMINVHN
    WeblogicPort 7001
</Location>
</VirtualHost>
```

Turning On the WebLogic Server Plug-In Enabled Flag

For security purposes, and since the load balancer terminates SSL requests (Oracle HTTP Server routes the requests as non-SSL to WebLogic Server), once you configure SSL for the load balancer, turn on the WebLogic Server plug-in enabled flag for the domain. To turn on the WebLogic Server plug-in enabled flag:

1. Log in to the Administration Console.
2. Click the domain name in the navigation tree on the left.
3. Open the **Web Applications** tab.
4. Click **Lock & Edit**.
5. Select the **WebLogic Plugin Enabled** checkbox.
6. Click **Save** and then click **Activate Changes**.
7. Restart the Administration Server (even if the Administration Console does not specifically prompt for that):
    a. Start the Oracle WebLogic Scripting Tool (WLST):

    ```
    cd ORACLE_COMMON_HOME/common/bin

    ./wlst.sh
    ```

    b. In the WLST shell, connect to Node Manager with the `nmConnect` command and the Node Manager credentials (make sure Node Manager is up and running), and start the Node Manager using nmStart:

    ```
    wls:/offline>nmConnect("node_admin_user","node_admin_password",
    "ADMINVHN", "domain_name","ASERVER_HOME")

    wls:/nm/domain_name>nmStart("AdminServer")
    ```

    The values of *node_admin_user* and *node_admin_password* are the Node Manager credentials.

    c. Stop the Administration Server, using either the Administration Console or, as follows, the WLST `nmKill` command:

    ```
    wls:/nm/domain_name>nmKill("AdminServer")
    ```

    d. Start the Administration Server using `nmStart`:

    ```
    wls:/offline>nmConnect("node_admin_user","node_admin_password",
    "WCCHOST1","5556", "domain_name","ASERVER_HOME")
    ```

```
wls:/nm/domain_namenmStart("AdminServer")
```

Setting the Front-End URL for the Administration Console and Setting Redirection Preferences

When you access the WebLogic Server Administration Console using a load balancer, you must change the Administration Server's front-end URL so that the user's web browser is redirected to the appropriate load-balancer address.

The WebLogic Server Administration Console application tracks changes made to ports, channels and security using the console. When changes made through the console are activated, the console validates its current listen address, port and protocol. If the listen address, port and protocol are still valid, the console redirects the HTTP request, replacing the host and port information with the Administration Server's listen address and port.

To change the Administration Server's front-end URL:

1. Log in to the WebLogic Server Administration Console.
2. Click **Lock & Edit**.
3. Expand the **Environment** node in the **Domain Structure** tree on the left.
4. Click **Servers**.
5. On the Summary of Servers page, click **Admin Server** in the **Names** column of the table.
6. On the settings page for Administration Server, click the **Protocols** tab.
7. Click the **HTTP** tab.
8. Set the **Frontend Host** field to `admin.example.com` and the **Frontend HTTP Port** field to `80` (modify accordingly if HTTPS is used for the admin URL).
9. Save and activate the changes.
10. Restart the Administration Server for changes to the front-end host directive to take effect.

Validating Access to the Management Consoles and Administration Server

To verify the changes that you have made in this chapter:

1. Use the following URL to the hardware load balancer to display the Oracle WebLogic Server Administration Console, and log in by using the Oracle WebLogic Server administrator credentials:

   ```
   http://admin.example.com/console
   ```

   This validates that the `admin.example.com` virtual host on the load balancer is able to route requests to the Oracle HTTP Server instances on the web tier, which in turn can route requests for the Oracle WebLogic Server Administration Console to the Administration Server in the application tier.

2. Similarly, you should be able to access the Fusion Middleware Control by using a similar URL:

```
http://admin.example.com/em
```

12

Extending the Domain to Include Oracle WebCenter Content

You need to perform certain tasks in order to extend the enterprise deployment domain with the Oracle WebCenter Content software. This includes installing the WebCenter Content, extending the domain for WebCenter Content and completing post-configuration and verification tasks.

This chapter provides information on installing the WebCenter Content, extending the domain for WebCenter Content and completing post-configuration and verification tasks.

- Installing WebCenter Content for an Enterprise Deployment
  The procedure for installing WebCenter Content in an enterprise deployment domain is explained in this section.
- Creating the Oracle WebCenter Content Database Schemas
  Before you can configure an Oracle WebCenter Content domain, you must install the required schemas in a certified database for use with this release of Oracle Fusion Middleware.
- Extending the Domain for WebCenter Content
  You need to perform the following tasks in order to extend the existing enterprise deployment domain with the Oracle WebCenter Content software.
- Completing Postconfiguration and Verification Tasks for WebCenter Content
  Several configuration and validation steps must be performed to bring the content servers online. Complete the following sections in the order listed.
- Configuring Content Server for the WebCenter Content User Interface
  If you are planning to use the WebCenter Content user interface (in addition to the native user interface for Content Server), you need to enable the Content Server parameters.
- Configuring Oracle HTTP Server for the WebCenter Content Cluster
  The instructions for configuring Oracle HTTP Server for the WebCenter Content Cluster are available in this section.
- Enabling JDBC Persistent Stores
  Oracle recommends that you use JDBC stores, which leverage the consistency, data protection, and high availability features of an oracle database and makes resources available for all the servers in the cluster.
- Enabling Automatic Service Migration
  To ensure high availability for the product installed in this chapter, you must configure service migration appropriately.

Installing WebCenter Content for an Enterprise Deployment

The procedure for installing WebCenter Content in an enterprise deployment domain is explained in this section.

This section contains the following procedures.

- Starting the Installation Program
- Navigating the Installation Screens
- Installing Oracle WebCenter Content on the Other Host Computers
- Verifying the Installation

Starting the Installation Program

To start the installation program:

1. Log in to WCCHOST1.
2. Go to the directory where you downloaded the installation program.
3. Launch the installation program by invoking the `java` executable from the JDK directory on your system, as shown in the example below.

   ```
   JAVA_HOME/bin/java -d64 -jar fmw_12.2.1.3.0_wccontent.jar
   ```

   Be sure to replace the JDK location in these examples with the actual JDK location on your system.

When the installation program appears, you are ready to begin the installation.

Navigating the Installation Screens

The installation program displays a series of screens, in the order listed in the following table.

If you need additional help with any of the installation screens, click the screen name.

Screen	Description
Installation Inventory Screen	If you did not create a central inventory when you installed the Oracle Fusion Middleware Infrastructure software, then this dialog box appears. Edit the **Inventory Directory** field so it points to the location of your local inventory, and then click **OK**.
Welcome	This screen introduces you to the product installer.
Auto Updates	Use this screen to automatically search My Oracle Support for available patches or automatically search a local directory for patches that you've already downloaded for your organization.
Installation Location	Use this screen to specify the location of your Oracle home directory. For more information about Oracle Fusion Middleware directory structure, see "Selecting Directories for Installation and Configuration" in *Planning an Installation of Oracle Fusion Middleware*.

Screen	Description
Prerequisite Checks	This screen verifies that your system meets the minimum necessary requirements. If there are any warning or error messages, you can refer to one of the documents in the Roadmap for Verifying Your System Environment section in *Planning Your Oracle Fusion Middleware Infrastructure Installation.*
Installation Summary	Use this screen to verify the installation options you selected. Click **Install** to begin the installation.
Installation Progress	This screen allows you to see the progress of the installation. Click **Next** when the progress bar reaches 100% complete.
Installation Complete	Review the information on this screen, then click **Finish** to dismiss the installer.

Installing Oracle WebCenter Content on the Other Host Computers

If you have followed the EDG shared storage recommendations, there is a separate shared storage volume for product installations on WCCHOST2, and you must also install the software on WCCHOST2. See Shared Storage Recommendations When Installing and Configuring an Enterprise Deployment.

Verifying the Installation

After you complete the installation, you can verify it by successfully completing the following tasks.

- Reviewing the Installation Log Files
- Checking the Directory Structure
- Viewing the Contents of Your Oracle Home

Reviewing the Installation Log Files

Review the contents of the installation log files to make sure that no problems were encountered. For a description of the log files and where to find them, see Understanding Installation Log Files in *Oracle Fusion Middleware Installing Software with the Oracle Universal Installer*.

Checking the Directory Structure

The contents of your installation vary based on the options you selected during the installation.

The addition of Oracle WebCenter Content will add the following directory and sub-directories:

```
/u01/oracle/products/fmw/wccontent
axf
common
ipm
plugins
ucm
```

```
wccadf
wccadfrui

/u01/oracle/products/fmw/wccapture
capture
common
plugins
```

For more information about the directory structure you should see after installation, see "What are the Key Oracle Fusion Middleware Directories?" in *Understanding Oracle Fusion Middleware*.

Viewing the Contents of Your Oracle Home

You can also view the contents of your Oracle home by using the `viewInventory` script. See Viewing the contents of an Oracle home in *Oracle Fusion Middleware Installing Software with the Oracle Universal Installer*.

Creating the Oracle WebCenter Content Database Schemas

Before you can configure an Oracle WebCenter Content domain, you must install the required schemas in a certified database for use with this release of Oracle Fusion Middleware.

Follow the instructions in this section to install the schemas.

- Starting the Repository Creation Utility (RCU)
- Navigating the RCU Screens to Create the Schemas

Starting the Repository Creation Utility (RCU)

To start the Repository Creation Utility (RCU):

1. Navigate to the *ORACLE_HOME*`/oracle_common/bin` directory on your system.
2. Make sure that the *JAVA_HOME* environment variable is set to the location of a certified JDK on your system. The location should be up to but not including the `bin` directory. For example, if your JDK is located in `/u01/oracle/products/jdk`:

   On UNIX operating systems:

   ```
   export JAVA_HOME=/u01/oracle/products/jdk
   ```
3. Start RCU:

   On UNIX operating systems:

   ```
   ./rcu
   ```

> **Note:**
>
> If your database has Transparent Data Encryption (TDE) enabled, and you want to encrypt your tablespaces that are created by the RCU, provide the `-encryptTablespace true` option when you start RCU.
>
> This defaults the appropriate RCU GUI Encrypt Tablespace checkbox selection on the Map Tablespaces screen without further effort during the RCU execution. See Encrypting Tablespaces in *Oracle Fusion Middleware Creating Schemas with the Repository Creation Utility*.

Navigating the RCU Screens to Create the Schemas

After you start the RCU, you can then use the wizard screens to select and install the required schemas for your Oracle Fusion Middleware product. Schema creation involves the following tasks.

Task 1 Introducing RCU
Click **Next**.

Task 2 Selecting a Method of Schema Creation
If you have the necessary permission and privileges to perform DBA activities on your database, select **System Load and Product Load**. This procedure assumes that you have the necessary privileges.
If you do not have the necessary permission or privileges to perform DBA activities in the database, you must select **Prepare Scripts for System Load** on this screen. This option will generate a SQL script, which can be provided to your database administrator. See "Understanding System Load and Product Load" in *Creating Schemas with the Repository Creation Utility*.

Task 3 Providing Database Connection Details
Provide the database connection details for RCU to connect to your database.

1. In the **Host Name** field, enter the SCAN address of the Oracle RAC Database.
2. Select **Connection Parameters**.
3. Enter the **DBMS/Service** details.
4. Enter the **Schema Owner** and **Schema Password** details.
5. Click **Next** to proceed, then click **OK** on the dialog window confirming that connection to the database was successful.

Task 4 Specifying a Custom Prefix and Selecting Schemas
Select **existing prefix**, and select the prefix you created while configuring the initial domain.
From the list of schemas, select the **WebCenter Content** schema. This will automatically select the following schemas as dependencies:

- **Oracle WebCenter Content Imaging**
- **Oracle WebCenter Content Server - Complete**
- **Oracle WebCenter Enterprise Capture**

- **Oracle WebCenter Content Server - Search Only**

The custom prefix is used to logically group these schemas together for use in this domain only; you must create a unique set of schemas for each domain as schema sharing across domains is not supported.

> **Tip:**
>
> For more information about custom prefixes, see "Understanding Custom Prefixes" in *Creating Schemas with the Repository Creation Utility*.
> For more information about how to organize your schemas in a multi-domain environment, see "Planning Your Schema Creation" in *Creating Schemas with the Repository Creation Utility*.

> **Tip:**
>
> You must make a note of the custom prefix you choose to enter here; you will need this later on during the domain creation process.

Click **Next** to proceed, then click **OK** in the dialog window confirming that prerequisite checking for schema creation was successful.

Task 5 Specifying Schema Passwords
Specify how you want to set the schema passwords on your database, then specify and confirm your passwords.

> **Tip:**
>
> You must make a note of the passwords you set on this screen; you will need them later on during the domain creation process.

Task 6 Verifying the Tablespaces for the Required Schemas
On the Map Tablespaces screen, review the information, and then click **Next** to accept the default values.
Click **OK** in the confirmation dialog box.

Task 7 Completing Schema Creation
Navigate through the remainder of the RCU screens to complete schema creation. When you reach the Completion Summary screen, click **Close** to dismiss RCU.

Task 8 Verifying the Schema Creation
To verify that the schemas were created successfully, and to verify the database connection details, use SQL*Plus or another utility to connect to the database, using the OCS schema name and the password you provided.
For example:

```
./sqlplus

SQL*Plus: Release 12.1.0.1.0 Production on Tue Mar 7 00:28:06 2017
```

```
Copyright (c) 1982, 2013, Oracle. All rights reserved.

Enter user-name: FMW12213
Enter password:

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 - 64bit Production
With the Partitioning, Real Application Clusters, Automatic Storage Management,
OLAP,
Advanced Analytics and Real Application Testing options

SQL> exit
Disconnected from Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, Real Application Clusters, Automatic Storage Management,
OLAP,
Advanced Analytics and Real Application Testing options
```

Extending the Domain for WebCenter Content

You need to perform the following tasks in order to extend the existing enterprise deployment domain with the Oracle WebCenter Content software.

Extending the domain involves the following tasks.

- Starting the Configuration Wizard
  Start the Configuration Wizard as the first step to extend the existing enterprise deployment domain.
- Navigating the Configuration Wizard Screens to Extend the Domain with WebCenter Content

Starting the Configuration Wizard

Start the Configuration Wizard as the first step to extend the existing enterprise deployment domain.

> **Note:**
>
> If you added any customizations directly to the start scripts in the domain, those are overwritten by the configuration wizard. To customize server startup parameters that apply to all servers in a domain, you can create a file called `setUserOverridesLate.sh` and configure it, for example, add custom libraries to the WebLogic Server classpath, specify Additional JAVA command line options for running the servers, or specify additional environment variables. Any customizations you add to this file are preserved during domain upgrade operations, and are carried over to remote servers when using the **`pack`** and **`unpack`** commands.

To start the Configuration Wizard:

1. From the WebLogic Server Console, stop any managed servers that are modified by this domain extension. Managed Servers that are not effected can remain on-line.

2. For any managed servers to be modified, verify that the managed server shutdown has completed.
3. Stop the Administration Server once all managed servers are in a steady state.
4. Navigate to the following directory and start the WebLogic Server Configuration Wizard.

```
cd ORACLE_HOME/oracle_common/common/bin
./config.sh
```

Navigating the Configuration Wizard Screens to Extend the Domain with WebCenter Content

Follow the instructions in the following sections to extend the domain for the topology with static or dynamic clusters.

- Extending the Domain with Static Clusters
- Extending the Domain with Dynamic Clusters

Extending the Domain with Static Clusters

Follow the instructions in this section to extend the domain for the topology with static clusters.

> **Note:**
>
> You can use the procedure described in this section to extend an existing domain with static clusters. If your needs do not match the instructions given in the procedure, be sure to make your selections accordingly, or refer to the supporting documentation for additional details.

Domain extension and configuration includes the following tasks:

- Task 1, Selecting the Domain Type and Domain Home Location
- Task 2, Selecting the Configuration Template
- Task 3, Configuring High Availability Options
- Task 4, Specifying the Database Configuration Type
- Task 5, Specifying JDBC Component Schema Information
- Task 6, Providing the GridLink Oracle RAC Database Connection Details
- Task 7, Testing the JDBC Connections
- Task 8, Selecting Advanced Configuration
- Task 9, Configuring Managed Servers
- Task 10, Configuring a Cluster
- Task 11, Assigning Server Templates
- Task 12, Configuring Dynamic Servers
- Task 13, Assigning Managed Servers to the Cluster

- Task 14, Configuring Coherence Clusters
- Task 15, Creating Machines for WebCenter Content Servers
- Task 16, Assigning Servers to Machines
- Task 17, Configuring Virtual Targets
- Task 18, Configuring Partitions
- Task 19, Configuring the File Store
- Task 20, Reviewing Your Configuration Specifications and Configuring the Domain
- Task 21, Writing Down Your Domain Home and Administration Server URL
- Task 22, Start the Administration Server

Task 1 Selecting the Domain Type and Domain Home Location
On the Configuration Type screen, select **Update an existing domain**.
In the **Domain Location** field, select the value of the ASERVER_HOME variable, which represents the complete path to the Administration domain home you created as part of the initial domain.
For more information about the directory location variables, see File System and Directory Variables Used in This Guide.

> **Tip:**
>
> More information about the other options on this screen can be found in Configuration Type in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 2 Selecting the Configuration Template
On the Templates screen, make sure **Update Domain Using Product Templates** is selected, then select the following templates:

- **Oracle Universal Content Management - Content Server - 12.2.1.3.0 [wccontent]**

  In addition, the following additional templates should already be selected, because they were used to create the initial Infrastructure domain:

  - **Oracle Enterprise Manager - 12.2.1.3.0 [em]**
  - **Oracle JRF - 12.2.1.3.0 [oracle_common]**
  - **WebLogic Coherence Cluster Extension - 12.2.1.3.0 [wlserver]**

> **Tip:**
>
> More information about the options on this screen can be found in Templates in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 3 Configuring High Availability Options
This screen appears for the first time when you create a cluster that uses Automatic Service Migration or JDBC stores or both. After you select HA Options for a cluster, all subsequent clusters that are added to the domain by using the Configuration Wizard,

automatically apply HA options (that is, the Configuration Wizard creates the JDBC stores and configures ASM for them).
On the High Availability Options screen:

- Select **Enable Automatic Service Migration** with **Database Basis**.
- Set **JTA Transaction Log Persistence** to **JDBC TLog Store**.
- Set **JMS Server Persistence** to **JMS JDBC Store**.

> **Note:**
>
> Oracle recommends that you use JDBC stores, which leverage the consistency, data protection, and high availability features of an oracle database and makes resources available for all the servers in the cluster. So, the Configuration Wizard steps assume that the JDBC persistent stores are used along with Automatic Service Migration.
> When you choose JDBC persistent stores, additional unused File Stores are automatically created but are not targeted to your clusters. Ignore these File Stores.
> If, for any reason, you want to use Files Stores, you can retain the default values for TLOGs and JMS persistent store options in this screen and configure them in a shared location later. See Task 8, Selecting Advanced Configuration. Shared location is required to resume JMS and HA in a failover scenario.
> You can also configure TLOGs and JMS persistent stores manually in a post step. For information about the differences between JDBC and Files Stores, and for specific instructions to configure them manually, see Using Persistent Stores for TLOGs and JMS in an Enterprise Deployment.

Click **Next**.

Task 4 Specifying the Database Configuration Type
On the Database Configuration Type screen, select **RCU Data**.
All fields are pre-populated, because you already configured the domain to reference the Fusion Middleware schemas that are required for the Infrastructure domain.
Verify and ensure that credentials in all the fields are the same that you have provided while configuring the Oracle Fusion Middleware Infrastructure.
Click **Get RCU Configuration** after you finish verifying the database connection information. The following output in the Connection Result Log indicates that the operating succeeded:

```
Connecting to the database server...OK
Retrieving schema data from database server...OK
Binding local schema components with retrieved data...OK

Successfully Done.
```

> **Tip:**
>
> For more information about the **RCU Data** option, see "Understanding the Service Table Schema" in *Creating Schemas with the Repository Creation Utility*.
> For more information about the other options on this screen, see "Datasource Defaults" in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 5 Specifying JDBC Component Schema Information
On the JDBC Component Schema screen, select all the UCM schemas (for WebCenter Content) in the table.
When you select the schemas, the fields on the page are activated and the database connection fields are populated automatically.
Click **Convert to GridLink** and click **Next**.

Task 6 Providing the GridLink Oracle RAC Database Connection Details
On the GridLink Oracle RAC Component Schema screen, provide the information that is required to connect to the RAC database and component schemas, as shown in the following table.

Element	Description and Recommended Value
SCAN, Host Name, and Port	Select the **SCAN** check box. In the **Host Name** field, enter the Single Client Access Name (SCAN) Address for the Oracle RAC database. In the **Port** field, enter the SCAN listening port for the database (for example, 1521)
ONS Host and Port	In the **ONS Host** field, enter the SCAN address for the Oracle RAC database. In the **Port** field, enter the ONS Remote port (typically, 6200).
Enable Fan	Verify that the **Enable Fan** check box is selected, so the database can receive and process FAN events.

Task 7 Testing the JDBC Connections
Use the JDBC Component Schema Test screen to test the data source connections you have just configured.
A green check mark in the **Status** column indicates a successful test. If you encounter any issues, see the error message in the Connection Result Log section of the screen, fix the problem, then try to test the connection again.

> **Tip:**
>
> For more information about the other options on this screen, see "Test Component Schema" in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 8 Selecting Advanced Configuration
To complete domain configuration for the topology, select the following options on the Advanced Configuration screen:

- **Topology**
- **File Store**

Task 9 Configuring Managed Servers
On the Managed Servers screen, a new Managed Server for Oracle WebCenter Content appears in the list of servers.
Perform the following tasks to modify the default Oracle WebCenter Content Managed Server and create a second Managed Server:

1. Rename the default Managed Server to `WLS_WCC1`.
2. Click **Add** to create a new Managed Server and name it `WLS_WCC2`.

> **Tip:**
>
> The server names recommended here will be used throughout this document; if you choose different names, be sure to replace them as needed.

3. Use the information in the following table to fill in the rest of the columns for each Oracle WebCenter Content Managed Server.

Server Name	Listen Address	Listen Port	Enable SSL	SSL Listen Port	Server Groups
WLS_WCC1	WCCHOST1	16200	Unchecked	Disabled	UCM-MGD-SVR
WLS_WCC2	WCCHOST2	16200	Unchecked	Disabled	UCM-MGD-SVR

> **Tip:**
>
> More information about the options on the Managed Server screen can be found in Managed Servers in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 10 Configuring a Cluster
In this task, you create a cluster of Managed Servers to which you can target the Oracle WebCenter Content software.
For more information about the `wcc.example.com` virtual server address, see Configuring Virtual Hosts on the Hardware Load Balancer.
Use the Clusters screen to create a new cluster:

1. Click the **Add** button.
2. Specify `WCC_Cluster` in the **Cluster Name** field.
3. From the **Dynamic Server Groups** drop-down list, select `Unspecified`.

> **Note:**
>
> By default, server instances in a cluster communicate with one another using unicast. If you want to change your cluster communications to use multicast, refer to Considerations for Choosing Unicast or Multicast in *Administering Clusters for Oracle WebLogic Server*.

> **Tip:**
>
> More information about the options on this screen can be found in Clusters in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 11 Assigning Server Templates
Click **Next** to proceed to the next screen.

Task 12 Configuring Dynamic Servers
Verify that all dynamic server options are disabled for clusters that are to remain as static clusters.

1. Confirm that the **Dynamic Cluster**, **Calculated Listen Port**, and **Calculated Machine Names** checkboxes on this screen are unchecked.
2. Confirm the **Server Template** selection is **Unspecified**.
3. Click **Next**.

Task 13 Assigning Managed Servers to the Cluster
Use the Assign Servers to Clusters screen to assign `WLS_WCC1` and `WLS_WCC2` to the new cluster, `WCC_Cluster`:

1. In the Clusters pane, select the cluster to which you want to assign the servers; in this case, `WCC_Cluster`.
2. In the Servers pane, assign WLS_WCC1 to `WCC_Cluster` by doing one of the following:
   - Click once on the `WLS_WCC1` Managed Server to select it, then click on the right arrow to move it beneath the selected cluster in the Clusters pane.
   - Double-click on `WLS_WCC1` to move it beneath the selected cluster in the clusters pane.
3. Repeat to assign `WLS_WCC2` to `WCC_Cluster`.
4. Click **Next** to proceed to the next screen.

> **Tip:**
>
> More information about the options on this screen can be found in Assign Servers to Clusters in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 14 Configuring Coherence Clusters
Use the Coherence Clusters screen to configure the Coherence cluster that is automatically added to the domain. Update the port number value to **9991**.

> **Note:**
>
> For Coherence licensing information, refer to *Oracle Coherence* in *Oracle Fusion Middleware Licensing Information*.

Task 15 Creating Machines for WebCenter Content Servers
If required, use the Machines screen to add two new Unix Machines:

1. On the Unix Machines tab, click the **Add** button.
2. Enter WCCHOST1 in the **Name** field.
3. Enter the host name of WCCHOST1 for the Node Manage Listener address. Leave the Node Manager port to the default value of 5556.
4. Repeat the above steps for WCCHOST2.

Under the **Unix Machine** tab, verify the names of the machines you created when creating the initial Infrastructure domain, as shown in the following table.
Click **Next** to proceed.

Name	Node Manager Listen Address	Node Manager Listen Port
WCCHOST1	The value of the WCCHOST1 host name variable. For example, `WCCHOST1.example.com`.	5556
WCCHOST2	The value of the WCCHOST2 host name variable. For example, `WCCHOST2.example.com`.	5556
ADMINHOST	Enter the value of the ADMINVHN variable.	5556

Task 16 Assigning Servers to Machines
Use the Assign Servers to Machines screen to assign the Oracle WebCenter Content Managed Servers you just created to the corresponding machines in the domain. Use a similar process as when assigning managed servers to the cluster. See Task 13, Assigning Managed Servers to the Cluster.
Assign ADMINVHN to ADMINHOST.
Assign WLS_WCC1 to WCCHOST1, and assign WLS_WCC2 to WCCHOST2.

> **Tip:**
>
> More information about the options on this screen can be found in Assign Servers to Machines in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 17 Configuring Virtual Targets
Click **Next** to proceed to the next screen.

Task 18 Configuring Partitions
Click **Next** to proceed to the next screen.

Task 19 Configuring the File Store
In the File Stores screen, assign the following directory for each of the WebCenter Content Persistence stores including Content Server JMS file stores:

ORACLE_RUNTIME/*domain_name*/*WCC_Cluster*/jms

> **Note:**
>
> Create the jms folder before starting the managed servers.

In this example, replace *ORACLE_RUNTIME* with the value of the variable for you environment. Replace *WCC_Cluster* with the name you assigned to the WebCenter Content cluster.
Select **Direct-Write** from the drop-down list for **Synchronous Write Policy** (for both the stores).

Task 20 Reviewing Your Configuration Specifications and Configuring the Domain
The Configuration Summary screen contains the detailed configuration information for the domain you are about to create. Review the details of each item on the screen and verify that the information is correct.
You can go back to any previous screen if you need to make any changes, either by using the **Back** button or by selecting the screen in the navigation pane.
Click **Update** to execute the domain extension.

> **Tip:**
>
> More information about the options on this screen can be found in Configuration Summary in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 21 Writing Down Your Domain Home and Administration Server URL
The Configuration Success screen will show the following items about the domain you just configured:

- Domain Location
- Administration Server URL

You must make a note of both items as you will need them later; the domain location is needed to access the scripts used to start the Node Manager and Administration Server, and the URL is needed to access the Administration Server.
Click **Finish** to dismiss the configuration wizard.

Task 22 Start the Administration Server
Start the Administration Server, login, and then verify the clusters and servers views to ensure that the changes made to the domain have been applied.

After you have completed extending the domain with static clusters, go to Completing Postconfiguration and Verification Tasks for WebCenter Content.

Extending the Domain with Dynamic Clusters

Follow the instructions in this section to extend the domain for the topology with dynamic clusters.

> **Note:**
>
> This procedure assumes that you are extending an existing domain. If your needs do not match the instructions given in the procedure, ensure that you make your selections accordingly, or refer to the supporting documentation for additional details.

Domain extension and configuration includes the following tasks:

- Task 1, Selecting the Domain Type and Domain Home Location
- Task 2, Selecting the Configuration Template
- Task 3, Configuring High Availability Options
- Task 4, Specifying the Database Configuration Type
- Task 5, Specifying JDBC Component Schema Information
- Task 6, Providing the GridLink Oracle RAC Database Connection Details
- Task 7, Testing the JDBC Connections
- Task 8, Selecting Advanced Configuration
- Task 9, Configuring Managed Servers
- Task 10, Configuring a Cluster
- Task 11, Assigning Server Templates
- Task 12, Configuring Dynamic Servers
- Task 13, Configuring Coherence Clusters
- Task 14, Creating Machines for WebCenter Content Servers
- Task 15, Assigning Servers to Machines
- Task 16, Configuring Virtual Targets
- Task 17, Configuring Partitions
- Task 18, Reviewing Your Configuration Specifications and Configuring the Domain
- Task 19, Writing Down Your Domain Home and Administration Server URL
- Task 20, Start the Administration Server

Task 1 Selecting the Domain Type and Domain Home Location
On the Configuration Type screen, select **Update an existing domain**.
In the **Domain Location** field, select the value of the ASERVER_HOME variable, which represents the complete path to the Administration domain home you created as part of the initial domain.

For more information about the directory location variables, see File System and Directory Variables Used in This Guide.

> **Tip:**
>
> More information about the other options on this screen can be found in Configuration Type in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 2 Selecting the Configuration Template
On the Templates screen, make sure **Update Domain Using Product Templates** is selected, then select the following templates:

- **Oracle Universal Content Management - Content Server - 12.2.1.3.0 [wccontent]**

  In addition, the following additional templates should already be selected, because they were used to create the initial Infrastructure domain:

  – **Oracle Enterprise Manager - 12.2.1.3.0 [em]**
  – **Oracle JRF - 12.2.1.3.0 [oracle_common]**
  – **WebLogic Coherence Cluster Extension - 12.2.1.3.0 [wlserver]**

> **Tip:**
>
> More information about the options on this screen can be found in Templates in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 3 Configuring High Availability Options
This screen appears for the first time when you create a cluster that uses Automatic Service Migration or JDBC stores or both. After you select HA Options for a cluster, all subsequent clusters that are added to the domain by using the Configuration Wizard, automatically apply these HA options.
On the High Availability Options screen, complete the following steps:

1. Verify that **Enable Automatic Service Migration** is not selected.
2. Verify that Default Persistent Store is selected as the **JTA Transaction Log Persistence** option.
3. Select JDBC Store as the **JMS Service Persistence** option.

You can configure only JMS Server persistence for Dynamic Clusters by using the Configuration Wizard. You cannot configure Service Migration and JTA Transaction Logs Persistence for Dynamic Clusters by using the Configuration Wizard; you have to configure them manually. Instructions are covered in later chapters of this guide.

> **Note:**
>
> Oracle recommends that you use JDBC stores, which leverage the consistency, data protection, and high availability features of an oracle database and makes resources available for all the servers in the cluster. So, the Configuration Wizard steps assume that the JDBC persistent stores are used along with Automatic Service Migration.
> When you choose JDBC persistent stores, additional unused File Stores are automatically created but are not targeted to your clusters. Ignore these File Stores.
> If, for any reason, you want to use Files Stores, you can retain the default values for TLOGs and JMS persistent store options in this screen and configure them in a shared location later. See Task 8, Selecting Advanced Configuration. Shared location is required to resume JMS and HA in a failover scenario.
> You can also configure TLOGs and JMS persistent stores manually in a post step. For information about the differences between JDBC and Files Stores, and for specific instructions to configure them manually, see Using Persistent Stores for TLOGs and JMS in an Enterprise Deployment.

Click **Next**.

Task 4 Specifying the Database Configuration Type

1. On the Database Configuration Type screen, select **RCU Data**.

   All fields are pre-populated, because you already configured the domain to reference the Fusion Middleware schemas that are required for the Infrastructure domain.

2. Verify and ensure that credentials in all the fields are the same that you have provided while configuring the Oracle Fusion Middleware Infrastructure.

3. Select **Connection Parameter**.

4. Click **Get RCU Configuration** after you finish verifying the database connection information. The following output in the Connection Result Log indicates that the operating succeeded:

   ```
   Connecting to the database server...OK
   Retrieving schema data from database server...OK
   Binding local schema components with retrieved data...OK

   Successfully Done
   ```

> **Tip:**
>
> For more information about the **RCU Data** option, see "Understanding the Service Table Schema" in *Creating Schemas with the Repository Creation Utility*.
> For more information about the other options on this screen, see "Datasource Defaults" in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 5 Specifying JDBC Component Schema Information
On the JDBC Component Schema screen, select all the UCM schemas (for WebCenter Content) in the table.
When you select the schemas, the fields on the page are activated and the database connection fields are populated automatically.
Click **Convert to GridLink** and click **Next**.

Task 6 Providing the GridLink Oracle RAC Database Connection Details
On the GridLink Oracle RAC Component Schema screen, provide the information that is required to connect to the RAC database and component schemas, as shown in the following table.

Element	Description and Recommended Value
SCAN, Host Name, and Port	Select the **SCAN** check box. In the **Host Name** field, enter the Single Client Access Name (SCAN) Address for the Oracle RAC database. In the **Port** field, enter the SCAN listening port for the database (for example, 1521)
ONS Host and Port	In the **ONS Host** field, enter the SCAN address for the Oracle RAC database. In the **Port** field, enter the ONS Remote port (typically, `6200`).
Enable Fan	Verify that the **Enable Fan** check box is selected, so the database can receive and process FAN events.

Task 7 Testing the JDBC Connections
Use the JDBC Component Schema Test screen to test the data source connections you have just configured.
A green check mark in the **Status** column indicates a successful test. If you encounter any issues, see the error message in the Connection Result Log section of the screen, fix the problem, then try to test the connection again.

> **Tip:**
>
> For more information about the other options on this screen, see "Test Component Schema" in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 8 Selecting Advanced Configuration
To complete domain configuration for the topology, select the following options on the Advanced Configuration screen:

- **Topology**

> **Note:**
>
> JMS JDBC stores are recommended and selected in Task 3, Configuring High Availability Options so there is no need to configure File Stores.
> If you choose JMS File Stores in Task 3, Configuring High Availability Options, you have to select the File Stores option to configure them in a shared location in *ORACLE_RUNTIME*/domain_name/SOA_Cluster/jms. Shared location is required to resume JMS and HA in a failover scenario.

Task 9 Configuring Managed Servers
On the Managed Servers screen, a new Managed Server for Oracle WebCenter Content appears in the list of servers.
Static Managed Server definitions are not needed for dynamic cluster configurations.
To remove the default Managed Server, complete the following steps:

1. Delete the default Managed Server.
2. Click **Next** to proceed to the next screen.

Task 10 Configuring a Cluster
In this task, you create a cluster of Managed Servers to which you can target the Oracle WebCenter Content software.
For more information about the wcc.example.com virtual server address, see Configuring Virtual Hosts on the Hardware Load Balancer.
Use the Clusters screen to create a new cluster:

1. Click the **Add** button.
2. Specify WCC_Cluster in the **Cluster Name** field.
3. Leave the **Cluster Address** field blank.
4. From the **Dynamic Server Groups** drop-down list, select UCM-DYN-CLUSTER.

> **Note:**
>
> By default, server instances in a cluster communicate with one another using unicast. If you want to change your cluster communications to use multicast, refer to "Considerations for Choosing Unicast or Multicast" in *Administering Clusters for Oracle WebLogic Server*.

> **Tip:**
>
> More information about the options on this screen can be found in Clusters in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 11 Assigning Server Templates
On the Server Templates screen, verify and update the following UCM server template configuration, add a template if necessary.
For the UCM server template, complete the following steps:

1. Verify `UCM-server-template` is listed in the **Name** field.
2. Specify `16200` in the **Listen Port** field.
3. Leave the **Enable SSL** option unchecked.
4. Click **Next** to proceed to the next screen.

Task 12 Configuring Dynamic Servers
Use the Dynamic Clusters screen to configure the required clusters:

1. Locate the row with the Cluster Name value of `WCC_Cluster`.
2. Verify `WLS_WCC` is listed in the **Server Name Prefix** field.
3. From the **Server Template** drop-down list, select `UCM-server-template`.
4. Specify `2` in the **Maximum Dynamic Server Count** field.
5. Specify `WCCHOST*` in the **Machine Name Match Expression** field and select **Calculated Machine Names**.

> **Note:**
>
> The dynamic cluster `Calculated Machine Names` and `Machine Name Match Expression` attributes control how server instances in a dynamic cluster are assigned to a machine.
>
> - If the `Calculated Machine Names` attribute is set to `False`, the dynamic servers will not be assigned to a machine.
> - If the `Calculated Machine Names` attribute is set to `True`, the `Machine Name Match Expression` attribute is used to select the set of machines that is used for the dynamic servers.
> - If the `Machine Name Match Expression` attribute is not set, all of the machines in the domain get selected. Assignments are made by using a round robin algorithm.
>
> To make things easier regardless of your actual physical hostname, Oracle recommends that you use `WCCHOST`*n* as your WebLogic machine names, where *n* is a sequential number. This is explained in Task 14, Creating Machines for WebCenter Content Servers of configuring the infrastructure domain. This convention makes it easy for dynamic clusters to determine where to start each cluster member. If you want to follow this convention, in the `Machine Match Expression` field, enter `WCCHOST*`.
>
> If you do not adopt this convention, the cluster members will be started on each machine that you define in Task 14, Creating Machines for WebCenter Content Servers, including that of `ADMINHOST`. This situation is undesirable as you would end you with two cluster members that run on the same physical server but are attached to two different domain homes.

6. Select **Calculated Listen Ports** and **Dynamic Cluster** fields.
7. Click **Next** to proceed to the next screen.

Task 13 Configuring Coherence Clusters
Use the Coherence Clusters screen to configure the Coherence cluster that is automatically added to the domain. Update the port number value to **9991**.

> **Note:**
>
> For Coherence licensing information, refer to *Oracle Coherence* in *Oracle Fusion Middleware Licensing Information*.

Task 14 Creating Machines for WebCenter Content Servers
Under the **Unix Machine** tab, verify the names of the machines you created when creating the initial Infrastructure domain, as shown in the following table.

Name	Node Manager Listen Address	Node Manager Listen Port
ADMINHOST	Enter the value of the ADMINVHN variable.	5556
WCCHOST1	The value of the WCCHOST1 host name variable. For example, `WCCHOST1.example.com`.	5556
WCCHOST2	The value of the WCCHOST2 host name variable. For example, `WCCHOST2.example.com`.	5556

If required, use the Machines screen to add two new Unix Machines:

1. On the Unix Machines tab, click the **Add** button.
2. Enter WCCHOST1 in the **Name** field.
3. Enter the host name of WCCHOST1 for the Node Manage Listener address. Leave the Node Manager port to the default value of 5556.
4. Repeat the above steps for WCCHOST2.
5. Click **Next** to proceed.

Task 15 Assigning Servers to Machines
Click **Next** to proceed to the next screen.

Task 16 Configuring Virtual Targets
Click **Next** to proceed to the next screen.

Task 17 Configuring Partitions
Click **Next** to proceed to the next screen.

Task 18 Reviewing Your Configuration Specifications and Configuring the Domain
The Configuration Summary screen contains the detailed configuration information for the domain you are about to create. Review the details of each item on the screen and verify that the information is correct.
You can go back to any previous screen if you need to make any changes, either by using the **Back** button or by selecting the screen in the navigation pane.
Click **Update** to execute the domain extension.

> **Tip:**
>
> More information about the options on this screen can be found in Configuration Summary in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 19 Writing Down Your Domain Home and Administration Server URL
The Configuration Success screen will show the following items about the domain you just configured:

- Domain Location
- Administration Server URL

You must make a note of both items as you will need them later; the domain location is needed to access the scripts used to start the Node Manager and Administration Server, and the URL is needed to access the Administration Server.
Click **Finish** to dismiss the configuration wizard.

Task 20 Start the Administration Server
Start the Administration Server, login, and then verify the clusters and servers views to ensure that the changes made to the domain have been applied.

Completing Postconfiguration and Verification Tasks for WebCenter Content

Several configuration and validation steps must be performed to bring the content servers online. Complete the following sections in the order listed.

Propagating the Extended Domain to the Domain Directories and Machines

After you have extended the domain with the Oracle Content, and you have started the Administration Server on WCCHOST1, you can then propagate the domain changes to the domain directories and machines.

- Packing Up the Extended Domain on WCCHOST1
- Unpacking the Domain in Managed Server Domain Home on WCCHOST1
- Unpacking the Domain on WCCHOST2

Packing Up the Extended Domain on WCCHOST1

Use the following steps to create a template JAR file that contains the domain configuration information, which now includes configuration information about Oracle WebCenter Content:

1. Log in to WCCHOST1 and run the `pack` command to create a template JAR file as follows:

   ```
   cd ORACLE_COMMON_HOME/common/bin

   ./pack.sh -managed=true
             -domain=ASERVER_HOME
             -template=/full_path/edgdomaintemplateExtWCC.jar
             -template_name=edgdomaintemplateExtWCC
   ```

   In this example:

   - Replace *`ASERVER_HOME`* with the actual path to the domain directory you created on the shared storage device.
   - Replace *`full_path`* with the complete path to the directory where you want the template jar file saved.
   - `edgdomaintemplateExtWCC.jar` is a sample name for the JAR file you are creating, which will contain the domain configuration files, including the configuration files for Oracle WebCenter Content.
   - `edgdomaintemplateExtWCC` is the name assigned to the domain template file.

2. Make a note of the location of the `edgdomaintemplateExtWCC.jar` file you just created with the `pack` command.

   By default, the pack template file is created in the current directory where you ran the `pack` command. In this example, it would be created in the *`ORACLE_COMMON_HOME`*`/common/bin` directory, but you can specify a full path for the template JAR file as part of the `-template` argument to the `pack` command.

   > **Tip:**
   >
   > For more information about the `pack` and `unpack` commands, see "Overview of the Pack and Unpack Commands" in *Creating Templates and Domains Using the Pack and Unpack Commands*.

Unpacking the Domain in Managed Server Domain Home on WCCHOST1

To copy the updated domain configuration information from the Administration Server domain directory to the Managed Servers domain directory:

1. Log in to WCCHOST1 if you haven't already.
2. If you haven't already, create the recommended directory structure for the Managed Server domain on WCCHOST1.

   Use the examples in File System and Directory Variables Used in This Guide as a guide.
3. Run the `unpack` command to unpack the template in the domain directory on the local storage, as follows:

   ```
   cd ORACLE_COMMON_HOME/common/bin

   ./unpack.sh -domain=MSERVER_HOME  \
       -template=template=/full_path/edgdomaintemplateExtWCC.jar \
       -overwrite_domain=true \
       -app_dir=APPLICATION_HOME \
   ```

   In this example:

   - Replace *`MSERVER_HOME`* with the complete path to the domain home to be created on the local storage disk. This is the location where the copy of the domain will be unpacked.
   - Replace *`full_path`* with the complete path to the directory where you want the template jar file saved.
   - `edgdomaintemplateExtWCC.jar` is the directory path and name of the template you created when you ran the `pack` command to pack up the domain on the shared storage device.
   - The `-overwrite_domain=true` argument is necessary when you are unpacking a Managed Server template into an existing domain and existing applications.

     For any file that is overwritten, a backup copy of the original is created. If any modifications had been applied to the start scripts and EAR files in the Managed Server domain directory, they must be restored after this unpack operation.
   - Replace *`APPLICATION_HOME`* with the complete path to the `applications` directory for the domain on shared storage.

   > **Tip:**
   >
   > For more information about the `pack` and `unpack` commands, see "Overview of the Pack and Unpack Commands" in *Creating Templates and Domains Using the Pack and Unpack Commands*.
4. Change directory to the newly created *MSERVER_HOME* directory and verify that the domain configuration files were copied to the correct location on the Unpacking

the Domain in on Managed Server Domain Home on WCCHOST1 shared storage device.

Unpacking the Domain on WCCHOST2

After you create a domain template jar, you can propagate the domain configuration to other hosts using the unpack command. To unpack the domain template jar file:

1. Log in to WCCHOST2.
2. If you haven't already, create the recommended directory structure for the Managed Server domain on WCCHOST2.

   Use the examples in File System and Directory Variables Used in This Guide as a guide.
3. Make sure the domain template jar file is accessible to WCCHOST2.

   For example, if you are using a separate shared storage volume or partition for WCCHOST2, then copy the template to the volume or partition mounted to WCCHOST2.
4. Run the `unpack` command to unpack the template in the domain directory on the local storage, as follows:

   ```
   cd ORACLE_COMMON_HOME/common/bin

   ./unpack.sh -domain=MSERVER_HOME \
       -template=edgdomaintemplateExtWCC.jar \
       -app_dir=APPLICATION_HOME \
       -overwrite_domain=true
   ```

   In this example:

   - Replace *`MSERVER_HOME`* with the complete path to the domain home to be created on the local storage disk. This is the location where the copy of the domain will be unpacked.
   - `edgdomaintemplateExtWCC.jar` is the directory path and name of the template you created when you ran the `pack` command to pack up the domain.

     Note that if you are using a separate shared storage volume or partition for WCCHOST2 (and redundant Oracle homes), then you must first copy the template to the volume or partition mounted to WCCHOST2.
   - Replace *`APPLICATION_HOME`* with the complete path to the `applications` directory for the domain on shared storage.

   > **Tip:**
   >
   > For more information about the `pack` and `unpack` commands, see "Overview of the Pack and Unpack Commands" in *Creating Templates and Domains Using the Pack and Unpack Commands*.
5. Change directory to the newly created *`MSERVER_HOME`* directory and verify that the domain configuration files were copied to the correct location on WCCHOST2.

Starting the Node Manager in the Managed Server Domain Directory on WCCHOST1

After you create the Managed Server domain directory, there are two domain home directories and two corresponding Node Manager instances on WCCHOST1. You use one Node Manager to control the Administration Server, running from Administration Server domain home, and you use the other Node Manager to control the Managed Servers, running from the Managed Server domain home.

You must start the two Node Managers independently.

> **Note:**
>
> The Node Manager for the Managed Server's *MSERVER_HOME* will be reset every time the domain configuration is unpacked. The `ListenAddress` will be changed to the *ADMINVHN* instead of the correct hostname. This needs to be changed to the correct value before starting the Node Manager service after an unpack is performed.

Follow these steps to update and start the Node Manager from the Managed Server home:

1. Verify that the listen address in the nodemanager.properties file is set correctly, by completing the following steps:
   a. Change to the following directory:

      MSERVER_HOME`/nodemanager/`

   b. Open the nodemanager.properties file for editing.
   c. Update the `ListenAddress` property to the correct hostname as follows:

      ```
      WCCHOST1: ListenAddress=WCCHOST1
      ```

   d. Update the `ListenPort` property with the correct Listen Port details.
   e. Make sure that QuitEnabled is set to ‘true’. If this line is not present in the nodemanager.properties file, add the following line:

      ```
      QuitEnabled=true
      ```

2. Change to the following directory:

   MSERVER_HOME`/bin`

3. Use the following command to start the Node Manager:

   ```
   nohup ./startNodeManager.sh > MSERVER_HOME/nodemanager/nodemanager.out 2>&1 &
   ```

For information about additional Node Manager configuration options, see *Oracle Fusion Middleware Administering Node Manager for Oracle WebLogic Server*.

Starting the WLS_WCC1 Managed Server

To start the WLS_WCC1 Managed Server:

1. Log in to the Oracle WebLogic Server Administration Console at `http://admin.example.com/console`.
2. Start the WLS_WCC1 Managed Server using the WebLogic Server Administration Console, as follows:
   a. Expand the **Environment** node in the **Domain Structure** tree on the left.
   b. Click **Servers**.
   c. On the Summary of Servers page, open the **Control** tab.
   d. Select **WLS_WCC1**, and then click **Start**.
3. Verify that the server status is reported as `Running` in the Administration 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.

Configuring the Content Server on WLS_WCC1 Managed Server

To configure Content Server:

1. Create the runtime cluster subdirectories required for the Oracle WebCenter Content cluster configuration.

   The Oracle WebCenter Content configuration files are on a shared disk so that all members of the cluster can access them. The shared disk location of the Oracle WebCenter Content enterprise deployment is located at *`ORACLE_RUNTIME`*`/WCDomain/WCC_Cluster`.

   Run the following commands to create the required subdirectories:

   ```
   mkdir -p ORACLE_RUNTIME/domain_name/WCC_Cluster/cs/vault
   mkdir -p ORACLE_RUNTIME/domain_name/WCC_Cluster/cs/weblayout
   mkdir -p ORACLE_RUNTIME/domain_name/WCC_Cluster/cs/data/users/
   profiles
   ```

2. Log in to WLS_WCC1 at `http://WCCHOST1:16200/cs` using the `weblogic` user name and password to display a configuration page.

   The Oracle WebCenter Content configuration files are on a shared disk so that all members of the cluster can access them. The shared disk location of the Oracle WebCenter Content enterprise deployment is at *`ORACLE_RUNTIME`*`/wcpedg_domain/WCC_Cluster`.

3. Change the following values on the server configuration page:

   Make sure that the **Is new Content Server Instance?** check box is selected.

   - **Content Server Instance Folder**: Set this to *`ORACLE_RUNTIME/domain_name/`*`WCC_Cluster/cs/`
     For example:`/u01/oracle/runtime/wcpedg_domain/WCC_Cluster/cs/`
   - **Native File Repository Location**: Set this to *`ORACLE_RUNTIME/domain_name/`*`WCC_Cluster/cs/vault/`

For example:`/u01/oracle/runtime/wcpedg_domain/WCC_Cluster/cs/vault/`

- **WebLayout Folder**: Set this to *`ORACLE_RUNTIME`*`/domain_name/WCC_Cluster/cs/weblayout/`
  For example:`/u01/oracle/runtime/wcpedg_domain/WCC_Cluster/cs/weblayout/`
- **User Profile Folder**: Set this to *`ORACLE_RUNTIME`*`/domain_name/WCC_Cluster/cs/data/users/profiles/`
  For example:`/u01/oracle/runtime/wcpedg_domain/WCC_Cluster/cs/data/users/profiles/`
- **Server Socket Port**: 4444
- **Incoming Socket Connection Address Security Filter**: A pipe-delimited list of the local host and the server IP addresses:

  ```
  127.0.0.1|0:0:0:0:0:0:0:1|WCCHOST1-IP|WCCHOST2-IP|WCPHOST1-IP|WCPHOST2-IP|WEBHOST1-IP|WEBHOST2-IP|wcp.example.com-IP|wcpinternal.example.com-IP|load-balancer-host-IP
  ```

  > **Note:**
  >
  > Must use IP addresses for all entries, including the load-balancer IP addresses for the internal virtual server and the primary interface depending on network address translation configuration settings at the load-balancer.

- **WebServer HTTP/HTTPS Address:** `wcc.example.com:443`
- **Web Address is HTTPS**: Select this checkbox.
- **Server Instance Name**: `WCC_Cluster`
- **Server Instance Label**: `WCC_Cluster`
- **Server Instance Description:** WebCenter Content cluster
- **Auto_Number Prefix**: `WCC_Cluster-`

4. Click **Submit** when finished.
5. Restart the Managed Server by using the WebLogic Server Administration Console.

Updating the cwallet File in the Administration Server

Content Server updates the `cwallet.sso` file located in the *`MSERVER_HOME`*`/config/fmwconfig` directory when it starts. This change needs to be propagated back to the Administration Server.

To do this, on WCCHOST1, copy the `cwallet.sso` file to *`ASERVER_HOME`*`/config/fmwconfig/` using the following command (note the back-slash for multi-line format):

```
cp MSERVER_HOME/config/fmwconfig/cwallet.sso \
ASERVER_HOME/config/fmwconfig/cwallet.sso
```

> **Note:**
>
> If any operation is performed in a WLS_WCC*n* server that modifies the `cwallet.sso` file in the *MSERVER_HOME*`/config/fmwconfig/` directory, the file will have to be immediately copied to the Administration Server domain configuration directory on WCCHOST1 at *ASERVER_HOME*`/config/fmwconfig`.

Starting the Node Manager in the Managed Server Domain Directory WCCHOST2

After you have propagated the domain configuration to WCCHOST2, you can update and start the Node Manager for the *MSERVER_HOME* domain directory.

You must start the two Node Managers independently.

> **Note:**
>
> The Node Manager for the Managed Server's *MSERVER_HOME* will be reset every time the domain configuration is unpacked. The `ListenAddress` will be changed to the *ADMINVHN* instead of the correct hostname. This needs to be changed to the correct value before starting the Node Manager service after an unpack is performed.

Follow these steps to update and start the Node Manager from the Managed Server home:

1. Verify that the listen address in the `nodemanager.properties` file is set correctly, by completing the following steps:
   a. Change directory to the *MSERVER_HOME*/nodemanager directory:

      ```
      cd MSERVER_HOME/nodemanager
      ```

   b. Open the nodemanager.properties file for editing.
   c. Validate the `ListenAddress` property to the correct hostname as follows:

      ```
      WCCHOST2: ListenAddress=WCCHOST2
      ```

   d. Update the `ListenPort` property with the correct Listen Port details.
   e. Make sure that QuitEnabled is set to ‘true’. If this line is not present in the nodemanager.properties file, add the following line:

      ```
      QuitEnabled=true
      ```

2. Change directory to the *MSERVER_HOME* binary directory:

   ```
   cd MSERVER_HOME/bin
   ```

3. Use the following command to start the Node Manager:

   ```
   nohup ./startNodeManager.sh > $MSERVER_HOME/nodemanager/nodemanager.out 2>&1
   &
   ```

For information about additional Node Manager configuration options, see *Oracle Fusion Middleware Administering Node Manager for Oracle WebLogic Server*.

Starting the WLS_WCC2 Managed Server

To start the WLS_WCC2 Managed Server:

1. Start the WLS_WCC2 Managed Server using the WebLogic Server Administration Console, as follows:
   a. Expand the **Environment** node in the **Domain Structure** tree on the left.
   b. Click **Servers**.
   c. On the Summary of Servers page, open the **Control** tab.
   d. Select **WLS_WCC2**, and then click **Start**.
2. Verify that the server status is reported as `Running` in the Administration 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.

Configuring the Content Server on WLS_WCC2 Managed Server

To configure Content Server:

1. Log in to WLS_WCC2 at `http://WCCHOST2:16200/cs` using the `weblogic` administration user name and password to display a configuration page.

   The Oracle WebCenter Content configuration files are on a shared disk so that all members of the cluster can access them. The shared disk location of the Oracle WebCenter Content enterprise deployment is at *`ORACLE_RUNTIME`*`/WCDomain/WCC_Cluster`.
2. Change the following values on the server configuration page:
   - **Content Server Instance Folder**: Set this to *`ORACLE_RUNTIME`*`/WCDomain/WCC_Cluster/cs`.
   - **Native File Repository Location**: Set this to *`ORACLE_RUNTIME`*`/WCDomain/WCC_Cluster/cs/vault`.
   - **WebLayout Folder**: Set this to *`ORACLE_RUNTIME`*`/WCDomain/WCC_Cluster/cs/weblayout`.
   - **User Profile Folder**: Set this to *`ORACLE_RUNTIME`*`/WCDomain/WCC_Cluster/cs/data/users/profiles`.
   - **Content Server URL Prefix**: `/cs/ (default value)`

   Make sure that the **Is new Content Server Instance?** check box is not selected.
3. Click **Submit** when finished.
4. Restart the Managed Server by using the WebLogic Server Administration Console.

Validating GridLink Data Sources

After the servers are started, verify that the GridLink data sources are correctly configured and that the ONS setup is correct. Perform these procedures for every GridLink data source created.

- Verifying the Configuration of a GridLink Data Source for WebCenter Content
- Verifying the Configuration of ONS for a GridLink Data Source

Verifying the Configuration of a GridLink Data Source for WebCenter Content

To verify the configuration of a GridLink data source for WebCenter Content:

1. Log in to the WebLogic Server Administration Console.
2. In the **Domain Structure** tree, expand **Services**, then click **Data Sources**.
3. Click the name of a GridLink data source that was created.
4. Click the **Monitoring** tab.
5. Click the **Testing** tab, select one of the servers, and click **Test Data Source**.

   The test should be successful if the configuration is correct.
6. Repeat the test for every WebLogic Server instance that uses the GridLink data source.

Verifying the Configuration of ONS for a GridLink Data Source

To verify the configuration of ONS for a GridLink data source for WebCenter Content:

1. In the **Domain Structure** tree on the Administration Console, expand **Services**, then click **Data Sources**.
2. Click the name of a GridLink data source.
3. Click the **Monitoring** tab.
4. Click the name of the server (WLS_WCC1).
5. Click the **ONS** tab.
6. In the **ONS** tab, select the **Testing** tab.
7. Select a server, and click **Test ONS**.

   The test should be successful if the configuration is correct. If the ONS test fails, verify that the ONS service is running in the Oracle RAC database nodes:

```
[orcl@WCCDBHOST1 ~]$ srvctl status scan_listener
SCAN Listener LISTENER_SCAN1 is enabled
SCAN listener LISTENER_SCAN1 is running on node WCCDBHOST1
SCAN Listener LISTENER_SCAN2 is enabled
SCAN listener LISTENER_SCAN2 is running on node WCCDBHOST2
SCAN Listener LISTENER_SCAN3 is enabled
SCAN listener LISTENER_SCAN3 is running on node WCCDBHOST2


[orcl@WCCDBHOST1 ~]$ srvctl config nodeapps -s
```

```
ONS exists: Local port 6100, remote port 6200, EM port 2016


[orcl@WCCDBHOST1 ~]$ srvctl status nodeapps | grep ONS
ONS is enabled
ONS daemon is running on node: WCCDBHOST1
ONS daemon is running on node: WCCDBHOST2
```

8. Repeat the ONS test for every WebLogic Server instance that uses the GridLink data source.

Configuring Additional Parameters

Using a text editor, add the following options to each cluster node's *MSERVER_HOME*/ucm/cs/bin/WLS_WCC*n*_intradoc.cfg file, where the directories specified are on a direct-bus-attached-controlled local disk and not a remote file system, such as a UNIX/Linux mounted NFS or clustered file system (like OCFS2, GFS2, or GPFS):

```
TraceDirectory=MSERVER_HOME/servers/WLS_WCCN/logs
EventDirectory=MSERVER_HOME/servers/WLS_WCCN/logs/event/
ArchiverDoLocks=true
DisableSharedCacheChecking=true
```

The trailing *N* should match your nodes server names, like WLS_WCC1 is for WCCHOST1 and WLS_WCC2 is for WCCHOST2, and so on.

These changes will take effect after a restart of all WebCenter Content Managed Servers, at the end of the procedure described in Configuring Service Retries for Oracle WebCenter Content.

> **Note:**
>
> The directories can reside in any local disk path that you have determined to have enough space to hold the WebCenter Content logs and any trace that you may configure. The preceding paths are a suggestion.

Configuring Service Retries for Oracle WebCenter Content

The following parameter should be set in the Content Server config.cfg file to enable login retries during an Oracle RAC failover:

```
ServiceAllowRetry=true
```

If this value is not set, users will need to manually retry any operation that was in progress when the failover began.

To configure service retries for Oracle WebCenter Content:

1. Go to Content Server at http://WCCHOST1:16200/cs, and log in using the non-LDAP WebLogic Server administration user name (for example, *weblogic*) and password.

2. From the **Administration** tray or menu, choose **Admin Server**, then **General Configuration**.
3. On the General Configuration page, add the following parameter in the **Additional Configuration Variables** box:

   ```
   ServiceAllowRetry=true
   ```

4. Click **Save**.

> **Note:**
>
> The new parameter is included in the `config.cfg` file, which is at the following location:
>
> *ORACLE_RUNTIME/domain_name/cluster_name*`/cs/config/config.cfg`
>
> (You can also edit this file directly in a text editor. Remember to restart all WebCenter Content Managed Servers.

Granting user administrative access to Oracle WebCenter Content

To grant users administrative access to Oracle WebCenter Content, configure the Administrators group in the LDAP directory and then add the `weblogic_wcc` user as a member of the group.

If adding the `weblogic_wcc` user to Administrators group is not allowed by your LDAP directory administrator, see Granting the WebCenter Content Administrative Roles via Credential Map.

Granting the WebCenter Content Administrative Roles through Credential Map

You must configure the Credential map to grant the Content Server administrative roles to the **`WCCAdministrators`** LDAP group.

The **`WCCAdministrators`** LDAP group is created in the Provisioning an Enterprise Deployment Administration User and Group section completed earlier. This configuration of credential map ensures consistent use of the LDAP administrative user for all configuration, administration, and maintenance tasks.

To configure a credential map and provide the necessary role grants to the LDAP-based **`WCCAdministrators`** group:

1. Log in to content server using the *weblogic* account.
2. Expand the **Administration** menu, select **Credential Maps**.
3. In the Map Identifier Field, enter a name for the new credential map: **LDAPAdmins**.

4. Add the following lines to map the LDAP group to the multiple administrative roles:

```
# Assign full set of administration roles to the LDAP
WCCAdministrators group
     WCCAdministrators, admin
     WCCAdministrators, sysmanager
     WCCAdministrators, refineryadmin
     WCCAdministrators, rmaadmin
     WCCAdministrators, pcmadmin
     WCCAdministrators, ermadmin
# Allow existing roles to propagate without mappings
     |#all|             , %%
#
# Comment the following if you are not implementing Accounts in
Content Server
     WCCAdministrators, @#all(RWDA)
     WCCAdministrators, @#none(RWDA)
```

> **Note:**
>
> If you are not implementing **Accounts**, comment out the last two lines of the previous example.

5. Click **Update**.
6. Navigate to **Administration** > **Providers**.
7. Click the **info** link for the existing JPS provider.
8. Make sure that the **Credential Map** parameter does not already have a map identifier listed.
9. Click the **Edit** button.
10. Enter the name of the Map Identifier from step 3 above as the Credential Map value.

> **Note:**
>
> Double-check the value entered for any typos, extra characters, and so on. If this is set incorrectly, you will not be able to log-in to your content server instances.

11. Click **Update**.
12. Repeat a modified process for the content server on WCCHOST2.
    a. Confirm that the `LDAPAdmins` credential map is already available for selection on the Credential Maps view.
    b. Repeat the edit of the `JpsUserProvider` noting that even though the correct `LDAPAdmins` credential Map value appears in the form automatically, it must still be submitted on each server to take effect.
13. Restart the managed servers in the **WCC_Cluster**.

14. Log in to each content server using the `weblogic_wcc` LDAP user and verify that the administrative menu options appear in the user interface.

> **Note:**
>
> If the provider configuration was entered incorrectly and you can no longer log-in, the `jpsuserprovider` data file needs to be corrected manually. In this case, shutdown all content server instances and edit the value of the `ProviderCredentialsMap` parameter in *`ORACLE_RUNTIME/DOMAIN_NAME/WCC_Cluster/cs/data/providers/jpsuserprovider/provider.hda`*, and restart/test one server instance at a time.

Configuring Content Server for the WebCenter Content User Interface

If you are planning to use the WebCenter Content user interface (in addition to the native user interface for Content Server), you need to enable the Content Server parameters.

From the **Administration** tray or menu, select **Admin Server** > **Component Manager** and then enable the following parameters:

- AutoSuggestConfig
- DynamicConverter
- FrameworkFolders

In addition, you need to set up the Remote Intradoc Client (RIDC) for communication between the WebCenter Content user interface and Content Server.

You can also set the following Content Server parameters for folders and searching, for the WebCenter Content user interface. To set the Content Server parameters for folders and searching for the WebCenter Content user interface:

1. From the **Administration** menu, select **Admin Server**.
2. Select **General Configuration**.
3. On the General Configuration page, add the following parameters in the **Additional Configuration Variables** tab:
   - `FoldersIndexParentFolderValues=true`
   - `FldEnforceFolderFileNameUniqueness=true`
   - `FldEnforceCaseInsensitiveNameUniqueness=true`
   - `SearchIndexerEngineName=OracleTextSearch` or `SearchIndexerEngineName=DATABASE.METADATA`

     Select either one of the values - `OracleTextSearch` or `DATABASE.METADATA`.

Configuring Oracle HTTP Server for the WebCenter Content Cluster

The instructions for configuring Oracle HTTP Server for the WebCenter Content Cluster are available in this section.

- Configuring Oracle HTTP Server for the WLS_WCC Managed Servers
- Enabling Access for Content Server Admin Applets
- Validating Access Through the Load Balancer

Configuring Oracle HTTP Server for the WLS_WCC Managed Servers

To configure the Oracle HTTP Server instances in the web tier so they route requests correctly to the Oracle WebCenter Content cluster, use the following procedure to create an additional Oracle HTTP Server configuration file that creates and defines the parameters of the `wcc.example.com` virtual server. To configure Oracle HTTP Server for the WLS_WCC Managed Servers:

1. Log in to WEBHOST1 and change directory to the configuration directory for the first Oracle HTTP Server instance (`ohs1`).

   ```
   cd WEB_DOMAIN_HOME/config/fmwconfig/components/OHS/ohs1/moduleconf/
   ```

   > **Note:**
   >
   > There are separate directories for configuration and runtime instance files. The runtime files under the `.../OHS/instances/ohsn/*` folder should not be edited directly. Edit only the `.../OHS/ohsn/*` configuration files.

2. Create the `wcc_vh.conf` file and add the following directive:

   ```
   <VirtualHost WEBHOST1:7777>
       ServerName https://wcc.example.com:443
       ServerAdmin you@your.address
       RewriteEngine On
       RewriteOptions inherit
   </VirtualHost>
   ```

3. In the `wcc_vh.conf` file, add the following lines between the `<VirtualHost>` and `</VirtualHost>` tags:

> **Note:**
>
> - For static clusters: The port numbers for WCCHOST1 and WCCHOST2 are `16200`.
> - For dynamic clusters: If the **Calculated Listen Ports** option is enabled, then the port number for WCCHOST1 is `16201` and WCCHOST2 is `16202`.
> - In the following example, it is assumed that you are using dynamic clusters and the **Calculated Listen Ports** option is enabled.

```
#UCM
<Location /cs>
   WebLogicCluster WCCHOST1:16201,WCCHOST2:16202
   WLSRequest ON
   WLCookieName JSESSIONID
   WLProxySSL ON
   WLProxySSLPassThrough ON
</Location>

<Location /adfAuthentication>
   WebLogicCluster WCCHOST1:16201,WCCHOST2:16202
   WLSRequest ON
   WLCookieName JSESSIONID
   WLProxySSL ON
   WLProxySSLPassThrough ON
</Location>

<Location /_ocsh>
   WebLogicCluster WCCHOST1:16201,WCCHOST2:16202
   WLSRequest ON
   WLCookieName JSESSIONID
   WLProxySSL ON
   WLProxySSLPassThrough ON
</Location>
```

4. Copy the `wcc_vh.conf` file to the configuration directory for the second Oracle HTTP Server instance (`ohs2`):

   `WEB_DOMAIN_HOME``/config/fmwconfig/components/`**`ohs2`**`/moduleconf/`

5. Edit the `wcc_vh.conf` and change any references of WEBHOST1 to WEBHOST2 in the `<VirtualHost>` directives.
6. Restart the Oracle HTTP server instances on WEBHOST1 and WEBHOST2.

Enabling Access for Content Server Admin Applets

Because the SSL termination is configured on Oracle Web Tier for the enterprise deployment reference topology, the WebLogic Plug-In needs to be enabled for each WebCenter Content Managed Server for successful access to the Content Server Admin Applets. To enable access for Content Server Admin Applets:

1. Log in to the WebLogic Server Administration Console at `http://host.domain:port/console`; for example:

   `http://ADMINVHN:7001/console`
2. *(For Dynamic clusters only)* On the **Environment** tab, click **Servers**.
3. Click **WLS_WCC1**.
4. Click the **General** tab on the **Configuration** tab, and go to the Advanced section.
5. Click **Lock & Edit**.
6. Select **Yes** from the **WebLogic Plug-In Enabled** drop-down menu.
7. Click the **Save** button to save the change.
8. Click **Activate Changes**.
9. Restart the WLS_WCC1 Managed Server from the Administration Console.
10. Repeat the above instructions for the WLS_WCC2 Managed Server.

Validating Access Through the Load Balancer

You should verify URLs to ensure that appropriate routing and failover is working from Oracle HTTP Server to WCC_Cluster.

- Verifying the URLs
- Verifying the Cluster Nodes

Verifying the URLs

To verify the URLs:

1. While WLS_WCC2 is running, stop WLS_WCC1 using the WebLogic Server Administration Console.
2. Access `https://wcc.example.com/cs` to verify that it is functioning properly.
3. Start WLS_WCC1 from the WebLogic Server Administration Console.
4. Stop WLS_WCC2 from the WebLogic Server Administration Console.
5. Access `https://wcc.example.com/cs` to verify that it is functioning properly.

You can verify the cluster node to which you were directed after the traffic balancing provided through your load balancer and then again through the web tier.

Verifying the Cluster Nodes

To verify the cluster node:

1. Log in to the following WebCenter Content page, using your administrator user and password credentials:

   `https://wcc.example.com/cs/idcplg?IdcService=CONFIG_INFO`
2. Browse to the Administration/Configuration for WCC_Cluster page.
3. In the Options and Others section of the page, click **Java Properties** on the right.
4. Obtain the value for **weblogic.Name**.

   This value denotes the cluster node you are accessing at the moment.

Enabling JDBC Persistent Stores

Oracle recommends that you use JDBC stores, which leverage the consistency, data protection, and high availability features of an oracle database and makes resources available for all the servers in the cluster.

Follow these guidelines to ensure that you use JDBC stores, when you use static or dynamic clusters:

- **For static clusters**

  If you have made the following selections in the High Availability Options screen, as recommended in this guide for static clusters, then JDBC persistent stores are already configured for both JMS and TLOGS:

  – Set **JTA Transaction Log Persistence** to **JDBC TLog Store**.

  – Set **JMS Server Persistence** to **JMS JDBC Store**.

- **For dynamic clusters**

  You can configure only JMS Server persistence for dynamic clusters by using the Configuration Wizard. JTA Transaction Logs Persistence must be configure manually, if required. If you have made the following selections in the High Availability Options screen, as recommended in this guide for dynamic clusters, then JDBC persistent stores are already configured for JMS.

  – Set **JMS Server Persistence** to **JMS JDBC Store**.

  – Verify that **JTA Transaction Log Persistence** is set to **Default Persistent Store**.

  Additional steps are needed to configure JTA Transaction Log with JDBC store. See Roadmap for Configuring a JDBC Persistent Store for TLOGs.

In case you did not select JDBC for JMS and TLOGS persistent in the High Availability Options screen, you can still configure JDBC stores manually in a post step. For specific instructions to configure them manually, see Using JDBC Persistent Stores for TLOGs and JMS in an Enterprise Deployment.

> **Note:**
>
> The High Availability Options screen appears during the Configuration Wizard session for the first time when you create a cluster that uses Automatic Service Migration or JDBC stores or both. All subsequent clusters that are added to the domain by using the Configuration Wizard, automatically apply the selected HA options.

Enabling Automatic Service Migration

To ensure high availability for the product installed in this chapter, you must configure service migration appropriately.

Follow these guidelines to ensure that you provide the required high availability for Weblogic services when you use static or dynamic clusters:

- **For static clusters**

  Automatic Service Migration is already configured if you select **Enable Automatic Service Migration** with **Database Basis** in the High Availability Options screen.

  The Database Leasing is already configured and the migratable targets are created with the appropriate policies for the cluster. If you have implemented these settings, validate the configuration, as described in Validating Automatic Service Migration in Static Clusters.

  In case you do not select this option during the Configuration Wizard session, you can configure automatic migration manually in a post step. For instructions to complete the steps for static clusters, see Configuring Automatic Service Migration in an Enterprise Deployment.

- **For dynamic clusters**

  You cannot configure Service Migration for dynamic clusters by using the Configuration Wizard, it needs to be configured manually. The following steps are needed:

  – Configure the database leasing for the cluster.

  – Set the appropriate migration policies for JTA Service and JMS Persistent Stores.

  For instructions to complete the steps for dynamic clusters, see Configuring Automatic Service Migration in an Enterprise Deployment.

> **Note:**
>
> The High Availability Options screen appears during the Configuration Wizard session for the first time when you create a cluster that uses Automatic Service Migration or JDBC stores or both. All subsequent clusters that are added to the domain by using the Configuration Wizard, automatically apply the selected HA options.

13

Extending the Domain with Oracle SOA Suite

You need to perform certain tasks in order to extend the enterprise deployment domain with the Oracle SOA Suite software.

- Variables Used When Configuring Oracle SOA Suite
  While extending the domain with Oracle SOA Suite, you will be referencing the directory variables listed in this section.
- Support for Dynamic Clusters in Oracle SOA Suite
  SOA supports two different topologies: static clusters-based topology and dynamic clusters-based topology. When choosing the dynamic cluster topology, there are some differences with respect to the conventional static clusters configuration.
- Synchronizing the System Clocks
  Before you extend the domain to include Oracle SOA Suite, verify that the system clocks on each host computer are synchronized.
- Installing the Software for an Enterprise Deployment
  The procedure to install the software for an enterprise deployment is explained in this section.
- Creating the Oracle SOA Suite Database Schemas
  Before you can configure an Oracle SOA Suite domain, you must install the required schemas in a certified database for use with this release of Oracle Fusion Middleware.
- Extending the Enterprise Deployment Domain with Oracle SOA Suite
  Perform the following tasks to extend the existing enterprise deployment domain with the Oracle SOA Suite software.
- Propagating the Extended Domain to the Domain Directories and Machines
  After you have extended the domain with the Oracle WebCenter Content instances, and you have restarted the Administration Server on WCCHOST1, you must then propagate the domain changes to the domain directories and machines.
- Updating the NodeManager Configuration After Unpacking the Domain
  When extending a domain, the `nodemanager.properties` file in *MSERVER_HOME* may be overwritten with some values from the `nodemanager.properties` file for *ASERVER_HOME*. Specifically, the `ListenAddress` and/or `CustomIdentityAlias` values can be reset.
- Modifying the Upload and Stage Directories to an Absolute Path in an Enterprise Deployment
  After you configure the domain and unpack it to the Managed Server domain directories on all the hosts, verify and update the upload and stage directories for Managed Servers in the new clusters. Also, update the upload directory for the AdminServer to have the same absolute path instead of relative, otherwise deployment issues can occur. If you implement dynamic clusters, the configuration of the server template assigned to each newly added cluster should be verified and updated, otherwise, verify and update every statically-defined Managed Server for the newly added clusters.

- Configuring Listen Addresses When Using Dynamic Clusters
- Starting and Validating the WLS_SOA1 Managed Server
  Now that you have extended the domain, started the Administration Server, and propagated the domain to the other hosts, you can start the newly configured Oracle SOA Suite Managed Servers.
- Starting and Validating the WLS_SOA2 Managed Server
  After you validate the successful configuration and startup of the WLS_SOA1 Managed Server, you can start and validate the WLS_SOA2 Managed Server.
- Configuring Oracle HTTP Server for the Extended Domain
  The following sections describe how to configure the Oracle HTTP Server instances so they route requests for both public and internal URLs to the proper clusters in the enterprise topology.
- Post-Configuration Steps for Oracle SOA Suite
  After you install and configure Oracle SOA Suite, consider the following post-configuration tasks.
- Enabling JDBC Persistent Stores
  Oracle recommends that you use JDBC stores, which leverage the consistency, data protection, and high availability features of an oracle database and makes resources available for all the servers in the cluster.
- Enabling Automatic Service Migration
  To ensure high availability for the product installed in this chapter, you must configure service migration appropriately.

Variables Used When Configuring Oracle SOA Suite

While extending the domain with Oracle SOA Suite, you will be referencing the directory variables listed in this section.

The values for several directory variables are defined in File System and Directory Variables Used in This Guide.

- ORACLE_HOME
- ASERVER_HOME
- MSERVER_HOME
- APPLICATION_HOME
- DEPLOY_PLAN_HOME
- OHS_DOMAIN_HOME
- JAVA_HOME
- ORACLE_RUNTIME

In addition, you'll be referencing the following virtual IP (VIP) address defined in Reserving the Required IP Addresses for an Enterprise Deployment:

- ADMINVHN

Actions in this chapter will be performed on the following host computers:

- WCCHOST1
- WCCHOST2

- WEBHOST1
- WEBHOST2

Support for Dynamic Clusters in Oracle SOA Suite

SOA supports two different topologies: static clusters-based topology and dynamic clusters-based topology. When choosing the dynamic cluster topology, there are some differences with respect to the conventional static clusters configuration.

Static clusters, also called configured clusters, are conventional clusters where you manually configure and add each server instance. A dynamic cluster includes a new "server-template" object that is used to define a centralized configuration for all generated (dynamic) server instances. When you create a dynamic cluster, the dynamic servers are preconfigured and automatically generated for you. This feature enables you to scale up the number of server instances in the dynamic cluster when you need additional server capacity. You can simply start the dynamic servers without having to first manually configure and add them to the cluster.

The steps in this section include instructions to configure the domain for both static or dynamic topologies. The differences between the two types of configurations are listed below:

- The Configuration Wizard process may differ for each case. For example, you should define server templates for dynamic clusters instead of servers.
- For dynamic clusters, you should perform the server-specific configurations such as setting the listen address, configuring the upload and staging directories, or configuring the keystores in the server template instead of in the server.
- Service migration is configured in a different way for dynamic clusters. Dynamic clusters do not use migratable targets, instead the JMS resources are targeted to the cluster. Specific procedure for configuring service migration for dynamic clusters is included in this guide.

Mixed clusters (clusters that contains both dynamic and configured server instances) are not supported in the Oracle WebCenter Content enterprise deployment.

Synchronizing the System Clocks

Before you extend the domain to include Oracle SOA Suite, verify that the system clocks on each host computer are synchronized.

To verify the time synchronization, query the NTP service by running the `ntpstat` command on each host.

Sample output:

```
$ ntpstat
synchronised to NTP server (10.132.0.121) at stratum 3
 time correct to within 42 ms
 polling server every 16 s
```

Installing the Software for an Enterprise Deployment

The procedure to install the software for an enterprise deployment is explained in this section.

Starting the Oracle SOA Suite Installer on WCCHOST1

To start the installation program:

1. Log in to WCCHOST1.
2. Go to the directory where you downloaded the installation program.
3. Launch the installation program by invoking the `java` executable from the JDK directory on your system, as shown in the following example:

   ```
   JAVA_HOME/bin/java -d64 -jar Installer File Name
   ```

   Be sure to replace the JDK location in these examples with the actual JDK location on your system.

   Replace *`Installer File Name`* with the name of the actual installer file for your product listed in Identifying and Obtaining Software Distributions for an Enterprise Deployment.

When the installation program appears, you are ready to begin the installation.

Navigating the Installation Screens

The installation program displays a series of screens, in the order listed in the following table.

If you need additional help with any of the installation screens, click the screen name.

Screen	Description
Welcome	This screen introduces you to the product installer.
Auto Updates	Use this screen to automatically search My Oracle Support for available patches or automatically search a local directory for patches that you have already downloaded for your organization.
Installation Location	Use this screen to specify the location of your Oracle home directory. For more information about Oracle Fusion Middleware directory structure, see Selecting Directories for Installation and Configuration in *Planning an Installation of Oracle Fusion Middleware*.
Installation Type	Use this screen to select the type of installation and consequently, the products and feature sets you want to install. • Select **SOA Suite**

Screen	Description
Prerequisite Checks	This screen verifies that your system meets the minimum necessary requirements. If there are any warning or error messages, you can refer to one of the documents in the Roadmap for Verifying Your System Environment section in *Oracle Fusion Middleware Installing and Configuring the Oracle Fusion Middleware Infrastructure*.
Installation Summary	Use this screen to verify the installation options that you selected. Click **Install** to begin the installation.
Installation Progress	This screen allows you to see the progress of the installation. Click **Next** when the progress bar reaches 100% complete.
Installation Complete	Review the information on this screen, then click **Finish** to dismiss the installer.

Installing Oracle SOA Suite on the Other Host Computers

If you have configured a separate shared storage volume or partition for the products mount point and *ORACLE_HOME* on WCCHOST2, then you must also perform the product installation on WCCHOST2.

See Shared Storage Recommendations When Installing and Configuring an Enterprise Deployment.

To install the software on the other host computers in the topology, log in to each host, and use the instructions in Starting the Infrastructure Installer on WCCHOST1 and Navigating the Infrastructure Installation Screens to create the Oracle home on the appropriate storage device.

Verifying the Installation

After you complete the installation, you can verify it by successfully completing the following tasks.

- Reviewing the Installation Log Files
- Checking the Directory Structure
- Viewing the Contents of Your Oracle Home

Reviewing the Installation Log Files

Review the contents of the installation log files to make sure that no problems were encountered. For a description of the log files and where to find them, see Understanding Installation Log Files in *Oracle Fusion Middleware Installing Software with the Oracle Universal Installer*.

Checking the Directory Structure

The contents of your installation vary based on the options that you select during the installation.

The addition of Oracle SOA Suite adds the following directory and sub-directories. Use the `ls --format=single-column` command to verify the directory structure.

```
ls --format=single-column /u01/oracle/products/fmw/soa

bam
bin
bpm
common
integration
jlib
modules
plugins
readme.txt
reports
soa
```

For more information about the directory structure you should see after installation, see What are the Key Oracle Fusion Middleware Directories? in *Understanding Oracle Fusion Middleware*.

Viewing the Contents of Your Oracle Home

You can also view the contents of your Oracle home by using the `viewInventory` script. See Viewing the contents of an Oracle home in *Installing Software with the Oracle Universal Installer*.

Creating the Oracle SOA Suite Database Schemas

Before you can configure an Oracle SOA Suite domain, you must install the required schemas in a certified database for use with this release of Oracle Fusion Middleware.

- Starting the Repository Creation Utility (RCU)
- Navigating the RCU Screens to Create the Schemas
- Verifying Schema Access
- Configuring SOA Schemas for Transactional Recovery

Starting the Repository Creation Utility (RCU)

To start the Repository Creation Utility (RCU):

1. Navigate to the *`ORACLE_HOME`*`/oracle_common/bin` directory on your system.
2. Make sure that the *`JAVA_HOME`* environment variable is set to the location of a certified JDK on your system. The location should be up to but not including the `bin` directory. For example, if your JDK is located in `/u01/oracle/products/jdk`:

   On UNIX operating systems:

   ```
   export JAVA_HOME=/u01/oracle/products/jdk
   ```

3. Start RCU:

   On UNIX operating systems:

   ```
   ./rcu
   ```

> **Note:**
>
> If your database has Transparent Data Encryption (TDE) enabled, and you want to encrypt your tablespaces that are created by the RCU, provide the `-encryptTablespace true` option when you start RCU.
>
> This defaults the appropriate RCU GUI Encrypt Tablespace checkbox selection on the Map Tablespaces screen without further effort during the RCU execution. See Encrypting Tablespaces in *Oracle Fusion Middleware Creating Schemas with the Repository Creation Utility*.

Navigating the RCU Screens to Create the Schemas

Schema creation involves the following tasks:

- Task 1, Introducing RCU
- Task 2, Selecting a Method of Schema Creation
- Task 3, Providing Database Connection Details
- Task 4, Specifying a Custom Prefix and Selecting Schemas
- Task 5, Specifying Schema Passwords
- Task 6, Specifying Custom Variables
- Task 7, Verifying the Tablespaces for the Required Schemas
- Task 8, Creating Schemas
- Task 9, Reviewing Completion Summary and Completing RCU Execution

Task 1 Introducing RCU
Click **Next**.

Task 2 Selecting a Method of Schema Creation
If you have the necessary permission and privileges to perform DBA activities on your database, select **System Load and Product Load**. This procedure assumes that you have the necessary privileges.
If you do not have the necessary permission or privileges to perform DBA activities in the database, you must select **Prepare Scripts for System Load** on this screen. This option generates a SQL script, which can be provided to your database administrator to create the required schema. See Understanding System Load and Product Load in *Oracle Fusion Middleware Creating Schemas with the Repository Creation Utility*.
Click **Next**.

Task 3 Providing Database Connection Details
Provide the database connection details for RCU to connect to your database.

1. In the **Host Name** field, enter the SCAN address of the Oracle RAC Database.
2. Enter the **Port** number of the RAC database scan listener, for example 1521.
3. Enter the RAC **Service Name** of the database.
4. Enter the **User Name** of a user that has permissions to create schemas and schema objects, for example SYS.

5. Enter the **Password** of the user name that you provided in step 4.
6. If you have selected the SYS user, ensure that you set the role to SYSDBA.
7. Click **Next** to proceed, then click **OK** on the dialog window confirming that connection to the database was successful.

Task 4 Specifying a Custom Prefix and Selecting Schemas
Choose **Select existing prefix**, and then select the prefix you used when you created the initial domain.
From the list of schemas, select the **SOA Suite** schema. This automatically selects **SOA Infrastructure**. In addition, the following dependent schemas have already been installed with the Infrastructure and are grayed out:

- **Common infrastructure Services**
- **Oracle Platform Security Services**
- **User Messaging Service**
- **Audit Services**
- **Audit Services Append**
- **Audit Services Viewer**
- **Metadata Services**
- **Weblogic Services**
- **Oracle WebCenter Content Server - Complete**

The custom prefix is used to logically group these schemas together for use in this domain only; you must create a unique set of schemas for each domain as schema sharing across domains is not supported.

> **Tip:**
>
> For more information about custom prefixes, see Understanding Custom Prefixes in *Creating Schemas with the Repository Creation Utility*.
> For more information about how to organize your schemas in a multi-domain environment, see Planning Your Schema Creation in *Creating Schemas with the Repository Creation Utility*.

Click **Next** to proceed, then click **OK** on the dialog window to confirm that prerequisite checking for schema creation was successful.

Task 5 Specifying Schema Passwords
Specify how you want to set the schema passwords on your database, then specify and confirm your passwords. Ensure that the complexity of the passwords meet the database security requirements before you continue. RCU proceeds at this point even if you do not meet the password polices. Hence, perform this check outside RCU itself.

> **Tip:**
>
> You must make a note of the passwords that you set on this screen; you need them later on during the domain creation process.

Click **Next**.

Task 6 Specifying Custom Variables
Specify the custom variables for the SOA Infrastructure schema.
For the enterprise deployment topology, enter `MEDIUM` for the **Database Profile** custom variable; enter `NO` for the **Healthcare Integration** variable. See About the Custom Variables Required for the SOA Suite Schemas in *Installing and Configuring Oracle SOA Suite and Business Process Management*.
Click **Next**.

Task 7 Verifying the Tablespaces for the Required Schemas
On the Map Tablespaces screen, review the information, and then click **Next** to accept the default values.
Click **OK** in the confirmation dialog box.
Click **Next**.

Task 8 Creating Schemas
Review the summary of the schemas to be loaded, and click **Create** to complete schema creation.

> **Note:**
>
> If failures occurred, review the listed log files to identify the root cause, resolve the defects, and then use RCU to drop and recreate the schemas before you continue.

Task 9 Reviewing Completion Summary and Completing RCU Execution
When you reach the Completion Summary screen, verify that all schema creations have been completed successfully, and then click **Close** to dismiss RCU.

Verifying Schema Access

Verify schema access by connecting to the database as the new schema users created by the RCU. Use SQL*Plus or another utility to connect, and provide the appropriate schema names and passwords entered in the RCU.

For example:

```
./sqlplus

SQL*Plus: Release 12.1.0.2.0 Production on Wed Aug 31 05:41:31 2016

Copyright (c) 1982, 2014, Oracle.  All rights reserved.

Enter user-name: FMW1221_SOAINFRA
Enter password: soainfra_password

Connected to:
```

```
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production
With the Partitioning, Real Application Clusters, Automatic Storage Management,
OLAP, Advanced Analytics and Real Application Testing options

SQL>
```

Configuring SOA Schemas for Transactional Recovery

After you have installed the Oracle SOA Suite schemas successfully, use the procedure in this section to configure the schemas for transactional recovery.

This procedure sets the appropriate database privileges so that the Oracle WebLogic Server transaction manager can query the schemas for transaction state information and issue the appropriate commands, such as commit and rollback, during recovery of in-flight transactions after a WebLogic Server is unexpectedly unavailable.

These privileges should be granted to the owner of the SOAINFRA schema, which you defined when you created the schemas with the RCU.

To configure the SOA schemas for transactional recovery privileges:

1. Log on to SQL*Plus as a user with sysdba privileges. For example:

   ```
   sqlplus "/ as sysdba"
   ```

2. Enter the following commands:

   ```
   SQL> Grant select on sys.dba_pending_transactions to
   soa_schema_prefix_soainfra;

   Grant succeeded.

   SQL> Grant force any transaction to soa_schema_prefix_soainfra;

   Grant succeeded.

   SQL>
   ```

Extending the Enterprise Deployment Domain with Oracle SOA Suite

Perform the following tasks to extend the existing enterprise deployment domain with the Oracle SOA Suite software.

> **Note:**
>
> For an improved footprint and to optimize startup, only core adapters are targeted to the SOA cluster (MFT Cluster if you are configuring MFT) after the Configuration Wizard session. You must target the second-tier adapters manually, if required. See Targeting Adapters Manually.

Extending the domain involves the following tasks:

- Starting the Configuration Wizard
  Start the Configuration Wizard as the first step to extend the existing enterprise deployment domain.
- Navigating the Configuration Wizard Screens to Extend the Domain with Oracle SOA Suite
  Follow the instructions in these sections to extend the domain for Oracle SOA Suite, with static or dynamic clusters.
- Targeting Adapters Manually
  Only core adapters are targeted to the SOA cluster after you run the Configuration Wizard. You must target second-tier adapters manually, on a need basis.

Starting the Configuration Wizard

Start the Configuration Wizard as the first step to extend the existing enterprise deployment domain.

> **Note:**
>
> If you added any customizations directly to the start scripts in the domain, those are overwritten by the configuration wizard. To customize server startup parameters that apply to all servers in a domain, you can create a file called `setUserOverridesLate.sh` and configure it, for example, add custom libraries to the WebLogic Server classpath, specify Additional JAVA command line options for running the servers, or specify additional environment variables. Any customizations you add to this file are preserved during domain upgrade operations, and are carried over to remote servers when using the **`pack`** and **`unpack`** commands.

To start the Configuration Wizard:

1. From the WebLogic Server Console, stop any managed servers that are modified by this domain extension. Managed Servers that are not effected can remain on-line.
2. For any managed servers to be modified, verify that the managed server shutdown has completed.
3. Stop the Administration Server once all managed servers are in a steady state.
4. Navigate to the following directory and start the WebLogic Server Configuration Wizard.

   ```
   cd ORACLE_HOME/oracle_common/common/bin
   ./config.sh
   ```

Navigating the Configuration Wizard Screens to Extend the Domain with Oracle SOA Suite

Follow the instructions in these sections to extend the domain for Oracle SOA Suite, with static or dynamic clusters.

- Extending the Domain with Static Clusters

- Extending the Domain with Dynamic Clusters

Extending the Domain with Static Clusters

Follow the instructions in this section to extend the domain for Oracle SOA Suite, with static clusters.

> **Note:**
>
> This procedure assumes that you are extending an existing domain. If your needs do not match the instructions given in the procedure, ensure that you make your selections accordingly, or refer to the supporting documentation for additional details.

Domain creation and configuration includes the following tasks:

- Task 1, Selecting the Domain Type and Domain Home Location
- Task 2, Selecting the Configuration Template
- Task 3, Configuring High Availability Options
- Task 4, Specifying the Database Configuration Type
- Task 5, Specifying JDBC Component Schema Information
- Task 6, Providing the GridLink Oracle RAC Database Connection Details
- Task 7, Testing the JDBC Connections
- Task 8, Keystore
- Task 9, Selecting Advanced Configuration
- Task 10, Configuring Managed Servers
- Task 11, Configuring a Cluster
- Task 12, Assigning Server Templates
- Task 13, Configuring Dynamic Servers
- Task 14, Assigning Managed Servers to the Cluster
- Task 15, Configuring Coherence Clusters
- Task 16, Verifying the Existing Machines
- Task 17, Assigning Servers to Machines
- Task 18, Configuring Virtual Targets
- Task 19, Configuring Partitions
- Task 20, Reviewing Your Configuration Specifications and Configuring the Domain
- Task 21, Writing Down Your Domain Home and Administration Server URL
- Task 22, Start the Administration Server

Task 1 Selecting the Domain Type and Domain Home Location
On the Configuration Type screen, select **Update an existing domain**.

In the **Domain Location** field, select the value of the *ASERVER_HOME* variable, which represents the complete path to the Administration Server domain home that you created in Creating the Initial Infrastructure Domain for an Enterprise Deployment .
For more information about the directory location variables, see File System and Directory Variables Used in This Guide.
For more information about the other options on this screen, see Configuration Type in *Oracle Fusion Middleware Creating WebLogic Domains Using the Configuration Wizard*.

Task 2 Selecting the Configuration Template
On the Templates screen, make sure **Update Domain Using Product Templates** is selected, then select the following templates:

- **Oracle SOA Suite - 12.2.1.3.0 [soa]**

  The following additional templates should already be selected, because they were used to create the initial domain:

  - Oracle Enterprise Manager - 12.2.1.3.0[em]
  - Oracle WSM Policy Manager - 12.2.1.3.0[oracle_common]
  - Oracle JRF - 12.2.1.3.0[oracle_common]
  - WebLogic Coherence Cluster Extension - 12.2.1.3.0[wlserver]

  And the following template should also be selected, because you already configured WebCenter Content:

  - Oracle Universal Content Management - Content Server - 12.2.1.3.0[wccontent]

For more information about the options on this screen, see Templates in *Oracle Fusion Middleware Creating WebLogic Domains Using the Configuration Wizard*.

Task 3 Configuring High Availability Options
This screen appears for the first time when you create a cluster that uses Automatic Service Migration or JDBC stores or both. After you select HA Options for a cluster, all subsequent clusters that are added to the domain by using the Configuration Wizard, automatically apply HA options (that is, the Configuration Wizard creates the JDBC stores and configures ASM for them).
On the High Availability Options screen:

- Select **Enable Automatic Service Migration** with **Database Basis**.
- Set **JTA Transaction Log Persistence** to **JDBC TLog Store**.
- Set **JMS Server Persistence** to **JMS JDBC Store**.

> **Note:**
>
> Oracle recommends that you use JDBC stores, which leverage the consistency, data protection, and high availability features of an oracle database and makes resources available for all the servers in the cluster. So, the Configuration Wizard steps assume that the JDBC persistent stores are used along with Automatic Service Migration.
> When you choose JDBC persistent stores, additional unused File Stores are automatically created but are not targeted to your clusters. Ignore these File Stores.
> If, for any reason, you want to use Files Stores, you can retain the default values for TLOGs and JMS persistent store options in this screen and configure them in a shared location later. See Task 9, Selecting Advanced Configuration. Shared location is required to resume JMS and JTA in a failover scenario.
> You can also configure TLOGs and JMS persistent stores manually in a post step. For information about the differences between JDBC and Files Stores, and for specific instructions to configure them manually, see Using Persistent Stores for TLOGs and JMS in an Enterprise Deployment.

Click **Next**.

Task 4 Specifying the Database Configuration Type
On the Database Configuration Type screen, select **RCU Data**.
All fields are prepopulated, because you already configured the domain to reference the Fusion Middleware schemas that are required for the Infrastructure domain. In the RCU Data screen:

- Verify that **Vendor** is `Oracle` and **Driver** is `*Oracle's Driver (Thin) for Service Connections; Versions: Any`.
- Verify that **Connection Parameters** is selected.
- Verify and ensure that credentials in all the fields are the same as those provided during the configuration of Oracle Fusion Middleware Infrastructure.

Click **Get RCU Configuration** after you finish verifying the database connection information. The following output in the Connection Result Log indicates that the operation is successful.

```
Connecting to the database server...OK
Retrieving schema data from database server...OK
Binding local schema components with retrieved data...OK

Successfully Done.
```

> **Tip:**
>
> For more information about the **RCU Data** option, see Understanding the Service Table Schema in *Oracle Fusion Middleware Creating Schemas with the Repository Creation Utility*.
> For more information about the other options on this screen, see Datasource Defaults in *Oracle Fusion Middleware Creating WebLogic Domains Using the Configuration Wizard*.

Task 5 Specifying JDBC Component Schema Information
On the JDBC Component Schema screen, select all the SOA schemas in the table. When you select the schemas, the fields on the page are activated and the database connection fields are populated automatically.
Click **Convert to GridLink** , and then click **Next**.

Task 6 Providing the GridLink Oracle RAC Database Connection Details
On the GridLink Oracle RAC Component Schema screen, provide the information that is required to connect to the RAC database and component schemas, as shown in the following table.

Element	Description and Recommended Value
SCAN, Host Name, and Port	Select the **SCAN** check box. In the **Host Name** field, enter the Single Client Access Name (SCAN) Address for the Oracle RAC database. In the **Port** field, enter the SCAN listening port for the database (for example, 1521).
ONS Host and Port	In the **ONS Host** field, enter the SCAN address for the Oracle RAC database. In the **Port** field, enter the ONS Remote port (typically, 6200). These values are required when connecting to Oracle 11*g* databases but optional when connecting to Oracle database 12*c* and higher. If you are using an Oracle 12*c* database, the ONS list is automatically provided from the database to the driver.
Enable Fan	Verify that the **Enable Fan** check box is selected, so that the database can receive and process FAN events.

Task 7 Testing the JDBC Connections
Use the JDBC Component Schema Test screen to test the data source connections that you have just configured.
A green check mark in the **Status** column indicates a successful test. If you encounter any issues, see the error message in the Connection Result Log section of the screen, fix the problem, then try to test the connection again.
For more information about the other options on this screen, see Test Component Schema in *Oracle Fusion Middleware Creating WebLogic Domains Using the Configuration Wizard*.

Task 8 Keystore
Use this screen to specify details about the keystore to be used in the domain.

For a typical enterprise deployment, you can leave the default values.
See Keystore in *Oracle Fusion Middleware Creating WebLogic Domains Using the Configuration Wizard*.

Task 9 Selecting Advanced Configuration
To complete domain configuration for the topology, select **Topology** on the Advanced Configuration screen.

> **Note:**
>
> JDBC stores are recommended and selected in Task 3, Configuring High Availability Options so there is no need to configure File Stores.
> If you choose File Stores in Task 3, Configuring High Availability Options, you have to select the File Stores option here to configure them in a shared location in `ORACLE_RUNTIME/domain_name/SOA_Cluster/jms`. Shared location is required to resume JMS and JTA in a failover scenario.

Task 10 Configuring Managed Servers
On the Managed Servers screen, a new Managed Server for Oracle SOA Suite appears in the list of servers. This server was created automatically by the Oracle SOA Suite configuration template that you selected in Task 2, Selecting the Configuration Template.
Perform the following tasks to modify the default Oracle SOA Suite Managed Server and create a second Oracle SOA Suite Managed Server:

1. Rename the default Oracle SOA Suite Managed Server to `WLS_SOA1`.
2. Click **Add** to create a new Oracle SOA Suite Managed Server, and name it `WLS_SOA2`.

> **Tip:**
>
> The server names recommended here are used throughout this document; if you choose different names, be sure to replace them as needed.

3. Use the information in the following table to fill in the rest of the columns for each Oracle SOA Suite Managed Server.

For more information about the options on the Managed Server screen, see Managed Servers in *Oracle Fusion Middleware Creating WebLogic Domains Using the Configuration Wizard*.

Server Name	Listen Address	Listen Port	Enable SSL	SSL Listen Port	Server Groups
WLS_SOA1	WCCHOST1	8001	No	Disabled	SOA-MGD-SVRS-ONLY, WSMPM-MAN-SVR, WSM-CACHE-SVR, JRF-MAN-SVR
WLS_SOA2	WCCHOST2	8001	No	Disabled	SOA-MGD-SVRS-ONLY, WSMPM-MAN-SVR, WSM-CACHE-SVR, JRF-MAN-SVR

Task 11 Configuring a Cluster
In this task, you create a cluster of Managed Servers to which you can target the Oracle SOA Suite software.
Use the Clusters screen to create a new cluster:

1. Click the **Add** button.
2. Specify `SOA_Cluster` in the **Cluster Name** field.
3. From the **Dynamic Server Groups** drop-down list, select `Unspecified`.

> **Note:**
>
> By default, server instances in a cluster communicate with one another by using unicast. If you want to change your cluster communications to use multicast, refer to Considerations for Choosing Unicast or Multicast in *Oracle Fusion Middleware Administering Clusters for Oracle WebLogic Server*.

For more information about the options on this screen, see Clusters in *Oracle Fusion Middleware Creating WebLogic Domains Using the Configuration Wizard*.

Task 12 Assigning Server Templates
Click **Next** to continue.

Task 13 Configuring Dynamic Servers
Verify that all dynamic server options are disabled for clusters that are to remain as static clusters. To configure dynamic servers:

1. Confirm that the **Dynamic Cluster**, **Calculated Listen Port**, and **Calculated Machine Names** checkboxes on this screen are unchecked.
2. Confirm the **Server Template** selection is **Unspecified**.
3. Click **Next**.

Task 14 Assigning Managed Servers to the Cluster
Use the Assign Servers to Clusters screen to assign `WLS_SOA1` and `WLS_SOA2` to the new cluster `SOA_Cluster`:

1. In the Clusters pane, select the cluster to which you want to assign the servers; in this case, `SOA_Cluster`.
2. In the Servers pane, assign `WLS_SOA1` to `SOA_Cluster` by doing one of the following:
   - Click `WLS_SOA1` Managed Server once to select it, and then click on the right arrow to move it beneath the selected cluster in the Clusters pane.
   - Double-click `WLS_SOA1` to move it beneath the selected cluster in the clusters pane.
3. Repeat to assign `WLS_SOA2` to `SOA_Cluster`.

For more information about the options on this screen, see Assign Servers to Clusters in *Oracle Fusion Middleware Creating WebLogic Domains Using the Configuration Wizard*.

Task 15 Configuring Coherence Clusters
Use the Coherence Clusters screen to configure the Coherence cluster that is automatically added to the domain. Leave the port number value at `9991`, as it was defined during the initial Infrastructure domain creation.
For Coherence licensing information, see Oracle Coherence Products in *Oracle Fusion Middleware Licensing Information User Manual*.

Task 16 Verifying the Existing Machines
Under the **Unix Machine** tab, verify the names of the machines you created when creating the initial Infrastructure domain.
Click **Next** to proceed.

Task 17 Assigning Servers to Machines
Use the Assign Servers to Machines screen to assign the Oracle SOA Suite Managed Servers you just created to the corresponding machines in the domain.
Assign `WLS_SOA1` to `WCCHOST1`, and assign `WLS_SOA2` to WCCHOST2.
For more information about the options on this screen, see Assign Servers to Machines in *Oracle Fusion Middleware Creating WebLogic Domains Using the Configuration Wizard*.

Task 18 Configuring Virtual Targets
Click **Next**.

Task 19 Configuring Partitions
Click **Next**.

Task 20 Reviewing Your Configuration Specifications and Configuring the Domain
The Configuration Summary screen contains detailed configuration information for the domain that you are about to extend. Review the details of each item on the screen and verify that the information is correct.
If you need to make any changes, you can go back to any previous screen if you need to make any changes, either by using the **Back** button or by selecting the screen in the navigation pane.
Click **Update** to execute the domain extension.
In the Configuration Progress screen, click **Next** when it finishes.
For more information about the options on this screen, see Configuration Summary in *Oracle Fusion Middleware Creating WebLogic Domains Using the Configuration Wizard*.

Task 21 Writing Down Your Domain Home and Administration Server URL
The Configuration Success screen shows the following items about the domain that you just configured, including:

- Domain Location
- Administration Server URL

Make a note of both these items, because you need them later; you need the domain location to access the scripts used to start the Administration Server, and you need the Administration Server URL to access the WebLogic Server Administration Console and Oracle Enterprise Manager Fusion Middleware Control.
Click **Finish** to dismiss the Configuration Wizard.
If the Admin Server was running during the domain extension process, restart the server before you continue.

Task 22 Start the Administration Server
Start the Administration Server to ensure that the changes that you have made to the domain have been applied.

After you complete extending the domain with static clusters, go to Targeting Adapters Manually.

Extending the Domain with Dynamic Clusters

Follow the instructions in this section to extend the domain for Oracle SOA Suite, with dynamic clusters.

> **Note:**
>
> This procedure assumes that you are extending an existing domain. If your needs do not match the instructions given in the procedure, ensure that you make your selections accordingly, or refer to the supporting documentation for additional details.

Domain creation and configuration includes the following tasks.

- Task 1, Selecting the Domain Type and Domain Home Location
- Task 2, Selecting the Configuration Template
- Task 3, Specifying the Database Configuration Type
- Task 4, Specifying JDBC Component Schema Information
- Task 5, Providing the GridLink Oracle RAC Database Connection Details
- Task 6, Testing the JDBC Connections
- Task 7, Keystore
- Task 8, Selecting Advanced Configuration
- Task 9, Configuring Managed Servers
- Task 10, Configuring a Cluster
- Task 11, Assigning Server Templates
- Task 12, Configuring Dynamic Servers
- Task 13, Assign Servers to Clusters
- Task 14, Configuring Coherence Clusters
- Task 15, Verifying the Existing Machines
- Task 16, Assigning Servers to Machines
- Task 17, Virtual Targets
- Task 18, Partitions
- Task 19, Reviewing Your Configuration Specifications and Configuring the Domain
- Task 20, Writing Down Your Domain Home and Administration Server URL
- Task 21, Start the Administration Server

Task 1 Selecting the Domain Type and Domain Home Location
On the Configuration Type screen, select **Update an existing domain**.
In the **Domain Location** field, select the value of the *ASERVER_HOME* variable, which represents the complete path to the Administration Server domain home you created when you created the initial domain.
Do not enter the value of the *MSERVER_HOME* variable, which represents the location of the Managed Servers domain directory.
For more information about the directory location variables, see File System and Directory Variables Used in This Guide.

> **Tip:**
>
> More information about the other options on this screen can be found in Configuration Type in *Oracle Fusion Middleware Creating WebLogic Domains Using the Configuration Wizard*.

Task 2 Selecting the Configuration Template
On the Templates screen, make sure **Update Domain Using Product Templates** is selected, then select the following templates:

- **Oracle SOA Suite - 12.2.1.3.0 [soa]**

  The following additional templates should already be selected, because they were used to create the initial domain:

  - Oracle Enterprise Manager - 12.2.1.3.0 [em]
  - Oracle WSM Policy Manager - 12.2.1.3.0 [oracle_common]
  - Oracle JRF - 12.2.1.3.0 [oracle_common]
  - WebLogic Coherence Cluster Extension - 12.2.1.3.0 [wlserver]

  And the following template should also be selected, because you already configured WebCenter Content:

  - Oracle Universal Content Management - Content Server - 12.2.1.3.0 [wccontent]

> **Tip:**
>
> More information about the options on this screen can be found in Templates in *Oracle Fusion Middleware Creating WebLogic Domains Using the Configuration Wizard*.

Task 3 Specifying the Database Configuration Type
On the Database Configuration Type screen, select **RCU Data**.
All fields are prepopulated, because you already configured the domain to reference the Fusion Middleware schemas that are required for the Infrastructure domain. On the RCU Data screen:

- Verify that **Vendor** is `Oracle` and **Driver** is `*Oracle's Driver (Thin) for Service Connections; Versions: Any`.
- Verify that **Connection Parameters** is selected.

- Verify and ensure that credentials in all the fields are the same as those provided during the configuration of Oracle Fusion Middleware Infrastructure.

Click **Get RCU Configuration** after you finish verifying the database connection information. The following output in the Connection Result Log indicates that the operation is successful:

```
Connecting to the database server...OK
Retrieving schema data from database server...OK
Binding local schema components with retrieved data...OK

Successfully Done.
```

Click **Next**.

> **Tip:**
>
> For more information about the **RCU Data** option, see Understanding the Service Table Schema in *Oracle Fusion Middleware Creating Schemas with the Repository Creation Utility*.
> For more information about the other options on this screen, see Datasource Defaults in *Oracle Fusion Middleware Creating WebLogic Domains Using the Configuration Wizard*.

Task 4 Specifying JDBC Component Schema Information
On the JDBC Component Schema screen, select all the SOA schemas in the table. When you select the schemas, the fields on the page are activated and the database connection fields are populated automatically.
Click **Convert to GridLink** and click **Next**.

Task 5 Providing the GridLink Oracle RAC Database Connection Details
On the GridLink Oracle RAC Component Schema screen, provide the information that is required to connect to the RAC database and component schemas, as shown in the following table.

Element	Description and Recommended Value
SCAN, Host Name, and Port	Select the **SCAN** check box. In the **Host Name** field, enter the Single Client Access Name (SCAN) Address for the Oracle RAC database. In the **Port** field, enter the SCAN listening port for the database (for example, 1521).
ONS Host and Port	In the **ONS Host** field, enter the SCAN address for the Oracle RAC database. In the **Port** field, enter the ONS Remote port (typically, `6200`). These values are required when connecting to Oracle 11*g* databases but optional when connecting to Oracle database 12*c* and higher. If you are using an Oracle 12*c* database, the ONS list is automatically provided from the database to the driver.

Element	Description and Recommended Value
Enable Fan	Verify that the **Enable Fan** check box is selected, so the database can receive and process FAN events.

Task 6 Testing the JDBC Connections
Use the JDBC Component Schema Test screen to test the data source connections you have just configured.
A green check mark in the **Status** column indicates a successful test. If you encounter any issues, see the error message in the Connection Result Log section of the screen, fix the problem, then try to test the connection again.

> **Tip:**
>
> For more information about the other options on this screen, see Test Component Schema in *Oracle Fusion Middleware Creating WebLogic Domains Using the Configuration Wizard*.

Task 7 Keystore
Use this screen to specify details about the keystore to be used in the domain.
For a typical enterprise deployment, you can leave the default values.
See Keystore in *Oracle Fusion Middleware Creating WebLogic Domains Using the Configuration Wizard*.

Task 8 Selecting Advanced Configuration
To complete domain configuration for the topology, select **Topology** and **Deployments and Services** on the Advanced Configuration screen.

Task 9 Configuring Managed Servers
On the Managed Servers screen, a new Managed Server for Oracle SOA Suite appears in the list of servers. This server was created automatically by the Oracle SOA Suite configuration template that you selected in Task 2, Selecting the Configuration Template.
SOA Static Managed Server definitions are not needed for dynamic cluster configuration. To remove the default Managed Server, complete the following steps:

1. Click the `soa_server1` managed server and click **Delete**.
2. Click **Next**.

Task 10 Configuring a Cluster
In this task, you create a cluster of Managed Servers to which you can target the Oracle SOA Suite software.
Use the Clusters screen to create a new cluster:

1. Click the **Add** button.
2. Specify `SOA_Cluster` in the **Cluster Name** field.
3. From the **Dynamic Server Groups** drop-down list, select `SOA-DYN-CLUSTER-ONLY`.

> **Note:**
>
> By default, server instances in a cluster communicate with one another using unicast. If you want to change your cluster communications to use multicast, refer to Considerations for Choosing Unicast or Multicast in *Oracle Fusion Middleware Administering Clusters for Oracle WebLogic Server*.

> **Tip:**
>
> More information about the options on this screen can be found in Clusters in *Oracle Fusion Middleware Creating WebLogic Domains Using the Configuration Wizard*.

Task 11 Assigning Server Templates
To configure a template, complete the following steps:

1. Verify that `soa-server-template` is selected in the **Name** field.
2. Specify `8000` in the **Listen Port** field.
3. Leave the **Enable SSL** option unchecked.
4. Click **Next**.

Task 12 Configuring Dynamic Servers
Use the Dynamic Clusters screen to configure the required clusters:

1. Verify `SOA_Cluster` is listed in the **Cluster Name** field.
2. Specify `WLS_SOA` in the **Server Name Prefix** field.
3. From the **Server Template** drop-down list, select `soa-server-template`.
4. Specify `2` in the **Dynamic Cluster Size** field.
5. Specify `WCCHOST*` in the **Machine Name Match Expression** field and select **Calculated Machine Names**.

> **Note:**
>
> The dynamic cluster **Calculated Machine Names** and **Machine Name Match Expression** attributes control how server instances in a dynamic cluster are assigned to a machine. If the **Calculated Machine Names** attribute is set to *False*, the dynamic servers are not assigned to a machine. If the **Calculated Machine Names** attribute is set to *True*, the **Machine Name Match Expression** attribute is used to select the set of machines that is used for the dynamic servers. If the **Machine Name Match Expression** attribute is not set, all the machines in the domain are selected. Assignments are made by using a round robin algorithm.

6. Select **Calculated Listen Ports** and **Dynamic Cluster fields**.

> **Note:**
>
> Dynamic clusters with the Calculated Listen Port option selected have incremental port numbers for each dynamic managed server that is created automatically: dynamic server 1 will use Listen Port+1, dynamic server 2 will use Listen Port+2.
>
> Since the Listen Port configured is 8000 and calculated ports is checked, SOA dynamic servers use the following port numbers:
>
> - WLS_SOA1 server listens in 8001 port
> - WLS_SOA2 server listens in 8002 port

7. Click **Next**.

> **Note:**
>
> The Configuration Wizard does not allow you to specify a specific listen address for dynamic servers. For information about setting a specific listen address for WebLogic servers that are members of a dynamic cluster, see Configuring Listen Addresses in Dynamic Cluster Server Templates.

Task 13 Assign Servers to Clusters
This screen appears during dynamic cluster configurations when static configured clusters and managed servers exist, even if the current extension does not include any new static servers.
For the SOA extension with dynamic clusters, no changes are needed.
Click **Next**.

Task 14 Configuring Coherence Clusters
Use the Coherence Clusters screen to configure the Coherence cluster that is automatically added to the domain. Leave the port number value at `9991`, as it was defined during the initial Infrastructure domain creation.

> **Note:**
>
> For Coherence licensing information, see Oracle Coherence Products in *Oracle Fusion Middleware Licensing Information User Manual*.

Task 15 Verifying the Existing Machines
Under the **Unix Machine** tab, verify the names of the machines that you created when creating the initial Infrastructure domain.
Click **Next**.

Task 16 Assigning Servers to Machines
Click **Next**.

Task 17 Virtual Targets
Click **Next**.

Task 18 Partitions
Click **Next**.

Task 19 Reviewing Your Configuration Specifications and Configuring the Domain
The Configuration Summary screen contains detailed configuration information for the domain that you are about to extend. Review the details of each item on the screen and verify that the information is correct.
If you need to make any changes, you can go back to any previous screen either by using the **Back** button or by selecting the screen in the navigation pane.
Click **Update** to execute the domain extension.
In the Configuration Progress screen, click **Next** when it finishes.

> **Tip:**
>
> More information about the options on this screen can be found in Configuration Summary in *Oracle Fusion Middleware Creating WebLogic Domains Using the Configuration Wizard*.

Task 20 Writing Down Your Domain Home and Administration Server URL
The Configuration Success screen shows the following items about the domain that you just configured, including:

- Domain Location
- Administration Server URL

Make a note of both these items, because you need them later; you need the domain location to access the scripts used to start the Administration Server, and you need the Administration Server URL to access the WebLogic Server Administration Console and Oracle Enterprise Manager Fusion Middleware Control.
Click **Finish** to dismiss the configuration wizard.
If the Admin Server was running during the domain extension process, restart the server before you continue.

Task 21 Start the Administration Server
Start the Administration Server to ensure that the changes that you have made to the domain have been applied.

Targeting Adapters Manually

Only core adapters are targeted to the SOA cluster after you run the Configuration Wizard. You must target second-tier adapters manually, on a need basis.

The following second-tier adapters have to be targeted manually:

> **Note:**
>
> Some of these adapters may not be available with the default installation. See Oracle Technology Network for Adapter availability.

- MSMQAdapter
- SocketAdapter

- OracleBamAdapter
- CoherenceAdapter
- SAPAdapter
- SiebelAdapter
- ERPAdapter
- Oracle SalesCloudAdapter
- RightNowAdapter
- EloquaAdapter
- NetSuiteAdapter
- LdapAdapter
- JDEWorldAdapter

To target a second-tier adapter manually:

1. Navigate to and log into the Oracle WebLogic Server Administration Console. For example: `http://ADMINVHN:7001/console`.

   > **Note:**
   >
   > If you have already configured web tier, use `http://admin.example.com/console`.

2. In the left pane of the console, click **Deployments**.
3. Locate and click the name of the adapter in the Summary of the Deployments table.
4. Click **Lock & Edit**.
5. In the **Targets** tab, select **SOA_Cluster**.

   > **Note:**
   >
   > If you are deploying MFT, select MFT_Cluster as the target.

6. Click **Save**.
7. Activate the changes.
8. In the left pane of the console, click **Deployments** and verify that the adapter is in the Active state.

Propagating the Extended Domain to the Domain Directories and Machines

After you have extended the domain with the Oracle WebCenter Content instances, and you have restarted the Administration Server on WCCHOST1, you must then propagate the domain changes to the domain directories and machines.

Table 13-1 summarizes the steps required to propagate the changes to all the domain directories and machines.

Note that there is no need to propagate the updated domain to the WEBHOST1 and WEBHOST2 machines because there are no changes to the Oracle HTTP Server instances on those host computers.

Table 13-1 Summary of Tasks Required to Propagate the Domain Changes to Domain Directories and Machines

Task	Description	More Information
Pack up the Extended Domain on WCCHOST1	Use the `pack` command to create a new template JAR file that contains the new Oracle SOA Suite Managed Servers configuration. When you pack up the domain, create a template JAR file called `wccdomaintemplateExtSOA.jar`.	Packing Up the Extended Domain on WCCHOST1
Unpack the Domain in the Managed Servers directory on WCCHOST1	Unpack the template JAR file in the Managed Servers directory on WCCHOST1 local storage.	Unpacking the Domain in the Managed Servers Domain Directory on WCCHOST1
Unpack the Domain on WCCHOST2	Unpack the template JAR file in the Managed Servers directory on the WCCHOST2local storage.	Unpacking the Domain on WCCHOST2

- Packing Up the Extended Domain on WCCHOST1
- Unpacking the Domain in the Managed Servers Domain Directory on WCCHOST1
- Unpacking the Domain on WCCHOST2

Packing Up the Extended Domain on WCCHOST1

Use the following steps to create a template JAR file that contains the domain configuration information:

1. Log in to WCCHOST1 and run the `pack` command to create a template JAR file as follows:

```
cd ORACLE_COMMON_HOME/common/bin

./pack.sh -managed=true \
          -domain=ASERVER_HOME \
          -template=full_path/wccdomaintemplateExtSOA.jar \
          -template_name=wcc_domain_template_extension_soa \
```

```
-log=/tmp/pack_soa.log \
-log_priority=debug
```

In this example:

- Replace *ASERVER_HOME* with the actual path to the domain directory that you created on the shared storage device.
- Replace *full_path* with the complete path to the directory where you want the template jar file saved.
- `wccdomaintemplateExtSOA.jar` is a sample name for the JAR file that you are creating, which contains the domain configuration files, including the configuration files for the Oracle HTTP Server instances.
- `wcc_domain_template_extension_soa` is the name assigned to the domain template file.

2. Make a note of the location of the template JAR file that you just created with the `pack` command.

> **Tip:**
>
> For more information about the `pack` and `unpack` commands, see Overview of the Pack and Unpack Commands in *Oracle Fusion Middleware Creating Templates and Domains Using the Pack and Unpack Commands*.

Unpacking the Domain in the Managed Servers Domain Directory on WCCHOST1

To copy the updated domain configuration information from the Administration Server domain directory to the Managed Servers domain directory:

1. Log in to WCCHOST1 if you haven't already.
2. If you haven't already, create the recommended directory structure for the Managed Server domain on the WCCHOST1 local storage device.

   Use the examples in File System and Directory Variables Used in This Guide as a guide.
3. Run the `unpack` command to unpack the template in the domain directory onto the local storage, as follows:

```
cd ORACLE_COMMON_HOME/common/bin

./unpack.sh -domain=MSERVER_HOME \
    -overwrite_domain=true \
    -template=/full_path/wccdomaintemplateExtSOA.jar \
    -log_priority=DEBUG \
    -log=/tmp/unpack.log \
    -app_dir=APPLICATION_HOME
```

> **Note:**
>
> The `-overwrite_domain` option in the `unpack` command allows you to unpack a managed server template into an existing domain and existing applications directories. For any file that is overwritten, a backup copy of the original is created. If any modifications had been applied to the start scripts and ear files in the managed server domain directory, they must be restored after this unpack operation.
>
> Additionally, to customize server startup parameters that apply to all servers in a domain, you can create a file called `setUserOverridesLate.sh` and configure it to, for example, add custom libraries to the WebLogic Server classpath, specify additional JAVA command-line options for running the servers, or specify additional environment variables. Any customizations that you add to this file are preserved during domain upgrade operations, and are carried over to remote servers when you use the `pack` and `unpack` commands.

In this example:

- Replace *`MSERVER_HOME`* with the complete path to the domain home to be created on the local storage disk. This is the location where the copy of the domain is unpacked.
- Replace `/full_path/wccdomaintemplateExtSOA.jar` with the complete path and file name of the domain template jar file that you created when you ran the `pack` command to pack up the domain on the shared storage device.
- Replace *`APPLICATION_HOME`* with the complete path to the applications directory for the domain on shared storage. See File System and Directory Variables Used in This Guide

> **Tip:**
>
> For more information about the `pack` and `unpack` commands, see Overview of the Pack and Unpack Commands in *Oracle Fusion Middleware Creating Templates and Domains Using the Pack and Unpack Commands*.

4. Change directory to the newly created *`MSERVER_HOME`* directory and verify that the domain configuration files were copied to the correct location on the WCCHOST1 local storage device.

Unpacking the Domain on WCCHOST2

This procedure assumes you have copied the file that you created earlier in a location that is accessible from both WCCHOST1 and WCCHOST2; such as the *ASERVER_HOME* directory, which is located on the shared storage filer:

1. Log in to WCCHOST2
2. If you haven't already, create the recommended directory structure for the Managed Server domain on the WCCHOST2 storage device.

Use the examples in File System and Directory Variables Used in This Guide as a guide.

3. Make sure the `wccdomaintemplateExtSOA.jar` accessible to WCCHOST2.

   For example, if you are using a separate shared storage volume or partition for WCCHOST2, then copy the template to the volume or partition mounted to WCCHOST2.

4. Run the `unpack` command to unpack the template in the domain directory onto the local storage, as follows:

```
cd ORACLE_COMMON_HOME/common/bin

./unpack.sh -domain=MSERVER_HOME \
            -overwrite_domain=true \
            -template=/full_path/wccdomaintemplateExtSOA.jar \
            -log_priority=DEBUG \
            -log=/tmp/unpack.log \
            -app_dir=APPLICATION_HOME
```

> **Note:**
>
> The `-overwrite_domain` option in the unpack command allows unpacking a managed server template into an existing domain and existing applications directories. For any file that is overwritten, a backup copy of the original is created. If any modifications had been applied to the start scripts and ear files in the managed server domain directory, they must be restored after this unpack operation.
>
> Additionally, to customize server startup parameters that apply to all servers in a domain, you can create a file called `setUserOverridesLate.sh` and configure it to, for example, add custom libraries to the WebLogic Server classpath, specify additional JAVA command line options for running the servers, or specify additional environment variables. Any customizations you add to this file are preserved during domain upgrade operations, and are carried over to remote servers when using the pack and unpack commands.

In this example:

- Replace *MSERVER_HOME* with the complete path to the domain home to be created on the local storage disk. This is the location where the copy of the domain will be unpacked.
- Replace `/full_path/wccdomaintemplateExtSOA.jar` with the complete path and file name of the domain template jar file that you created when you ran the pack command to pack up the domain on the shared storage device.
- Replace *APPLICATION_HOME* with the complete path to the Application directory for the domain on shared storage. See File System and Directory Variables Used in This Guide.

> **Tip:**
>
> For more information about the pack and unpack commands, see Overview of the Pack and Unpack Commands in *Oracle Fusion Middleware Creating Templates and Domains Using the Pack and Unpack Commands*.

5. Change directory to the newly created MSERVER_HOME directory and verify that the domain configuration files were copied to the correct location on the WCCHOST2 local storage device.

Updating the NodeManager Configuration After Unpacking the Domain

When extending a domain, the `nodemanager.properties` file in *MSERVER_HOME* may be overwritten with some values from the `nodemanager.properties` file for *ASERVER_HOME*. Specifically, the `ListenAddress` and/or `CustomIdentityAlias` values can be reset.

> **Notes::**
>
> - The `ListenAddress` may typically get reset on the *MSERVER_HOME* nodemanager residing on the same host as the *ASERVER_HOME* nodemanager. In this topology, WCCHOST1.
> - For domain extensions prior to Enabling SSL Communication Between the SOA Servers and the Hardware Load Balancer, steps 2 through 4 regarding the `CustomIdentityAlias` may not be applicable.

For the *`MSERVER_HOME`*`/nodemanager/nodemanager.properties` file on each host:

1. Verify the correct `ListenAddress` parameter value and reset it, if required.

   ```
   grep ListenAddress MSERVER_HOME/nodemanager/nodemanager.properties
   ```

2. Confirm the list of configured Identity Aliases from the domain configuration file as a reference for the next command.

   ```
   grep server-private-key-alias ASERVER_HOME/config/config.xml | sort
   | uniq
   ```

> **Note:**
>
> When using Dynamic Clusters, this listing will present only the `ADMINVHN` and wildcard certificate identity aliases.
>
> Use the appropriate host-specific certificate identity aliases when updating the `nodemanger.properties CustomIdentityAlias` property in the next instruction.

3. Verify the current `nodemanager.properties CustomIdentityAlias` parameter value matches the alias for the host.

   ```
   grep CustomIdentityAlias MSERVER_HOME/nodemanager/
   nodemanager.properties
   ```

4. Reset the `CustomIdentityAlias` parameter value to the correct alias string appropriate for the current host, if required.
5. Restart the `nodemanager` process:

   ```
   kill `ps -eaf | grep weblogic.NodeManager | grep MSERVER_HOME |
   grep -v grep | awk '{print $2}' `
   nohup MSERVER_HOME/bin/startNodeManager.sh > MSERVER_HOME/
   nodemanager/nodemanager.out 2>&1 &
   ```

6. Stop and restart the AdminServer once all node managers are appropriately configured and online.

> **Note:**
>
> For more information about the `CustomIdentityAlias` parameter, see Configuring Node Manager to Use the Custom Keystores.

Modifying the Upload and Stage Directories to an Absolute Path in an Enterprise Deployment

After you configure the domain and unpack it to the Managed Server domain directories on all the hosts, verify and update the upload and stage directories for Managed Servers in the new clusters. Also, update the upload directory for the AdminServer to have the same absolute path instead of relative, otherwise deployment issues can occur. If you implement dynamic clusters, the configuration of the server template assigned to each newly added cluster should be verified and updated, otherwise, verify and update every statically-defined Managed Server for the newly added clusters.

> **Note:**
>
> This option is applicable only for static clusters.

This step is necessary to avoid potential issues when you perform remote deployments and for deployments that require the stage mode.

To update the directory paths for the Deployment Stage and Upload locations, complete the following steps:

1. Log in to the Oracle WebLogic Server Administration Console.
2. In the left navigation tree, expand **Domain**, and then **Environment**.
3. Click **Lock & Edit**.
4. Navigate to and edit the appropriate objects for your cluster type.
   a. For Static Clusters, navigate to **Servers** and click the name of the Managed Server you want to edit.
   b. For Dynamic Clusters, navigate to **Clusters** > **Server Templates**, and click on the name of the server template to be edited.
5. For each new Managed Server or Server Template to be edited:
   a. Click the **Configuration** tab, and then click the **Deployment** tab.
   b. Verify that the **Staging Directory Name** is set to the following:

      `MSERVER_HOME/servers/server_or_template_name/stage`

      Replace *MSERVER_HOME* with the full path for the *MSERVER_HOME* directory.

      If you use static clusters, update with the correct name of the Managed Server that you are editing.

      If you use dynamic clusters, leave the template name intact.
      For example: `/u02/oracle/config/domains/wccedg_domain/servers/XYZ-server-template/stage`
   c. Update the **Upload Directory Name** to the following value:

      `ASERVER_HOME/servers/AdminServer/upload`

      Replace *ASERVER_HOME* with the directory path for the *ASERVER_HOME* directory.
   d. Click **Save**.
   e. Return to the Summary of Servers or Summary of Server Templates screen as applicable.
6. Repeat the previous steps for each of the new managed servers or dynamic cluster server templates.
7. Navigate to and update the Upload Directory Name value for the AdminServer:
   a. Navigate to **Servers**, and select the AdminServer.
   b. Click the **Configuration** tab, and then click the **Deployment** Tab.
   c. Verify that the **Staging Directory Name** is set to the following absolute path:

      `ASERVER_HOME/servers/AdminServer/stage`
   d. Update the **Upload Directory Name** to the following absolute path:

      `ASERVER_HOME/servers/AdminServer/upload`

      Replace *ASERVER_HOME* with the directory path for the *ASERVER_HOME* directory.

e. Click **Save**.

8. When you have modified all the appropriate objects, click **Activate Changes**.
9. Restart all Managed Servers for the changes to take effect.

> **Note:**
>
> If you continue directly with further domain configurations, a restart to enable the stage and upload directory changes is not strictly necessary at this time.

Configuring Listen Addresses When Using Dynamic Clusters

The default configuration for dynamic managed servers in dynamic clusters is to listen on all available network interfaces. In most cases, the default configuration may be undesirable. To limit the listen address to a specific address when you use dynamic clusters, see Configuring Listen Addresses in Dynamic Cluster Server Templates. Reverify the test URLs that are provided in the previous sections after you change the listen address and restart the clustered managed servers.

Starting and Validating the WLS_SOA1 Managed Server

Now that you have extended the domain, started the Administration Server, and propagated the domain to the other hosts, you can start the newly configured Oracle SOA Suite Managed Servers.

This process involves three tasks as described in the following sections.

- Starting the WLS_SOA1 Managed Server
- Adding the SOAAdmin Role to the Administrators Group
- Validating the Managed Server by Logging in to the SOA Infrastructure

Starting the WLS_SOA1 Managed Server

To start the WLS_SOA1 Managed Server:

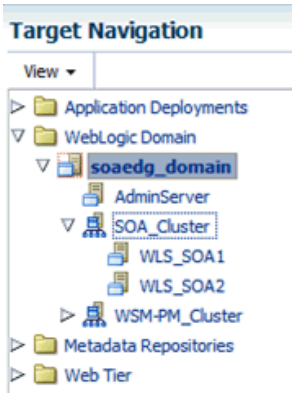
1. Enter the following URL into a browser to display the Fusion Middleware Control login screen:

   `http://ADMINVHN:7001/em`

   > **Note:**
   >
   > If you have already configured web tier, use `http://admin.example.com/console`.

2. Sign in to the Fusion Middleware Control by using the administrator's account. For example: `weblogic_wcc`.

3. In the **Target Navigation** pane, expand the domain to view the Managed Servers in the domain.



4. Select only the **WLS_SOA1** Managed Server and click **Start Up** on the Oracle WebLogic Server toolbar.

> **Note:**
>
> SOA Servers depend on the policy access service to be functional. This implies that the WSM-PM Managed Servers in the domain need to be up and running and reachable before the SOA servers are started.

5. When the startup operation is complete, navigate to the Domain home page and verify that the WLS_SOA1 Managed Server is up and running.

Adding the SOAAdmin Role to the Administrators Group

Before you validate the Oracle SOA Suite configuration on the WLS_SOA1 Managed Server, add the `SOAAdmin` administration role to the enterprise deployment administration group (`WCCAdministrators`).

To perform this task, refer to Configuring Roles for Administration of an Enterprise Deployment.

Validating the Managed Server by Logging in to the SOA Infrastructure

After you add the `SOAAdmin` role to the SOA Administrators group, you can then validate the configuration of the Oracle SOA Suite software on the WLS_SOA1 Managed Server as follows:

1. Use your web browser to navigate to the following URL:

   `http://WCCHOST1:8001/soa-infra/`

2. Log in by using the enterprise deployment administrator user credentials (`weblogic_wcc`).

   You should see a web page with the following title:

   `Welcome to the Oracle SOA Platform on WebLogic`

Starting and Validating the WLS_SOA2 Managed Server

After you validate the successful configuration and startup of the WLS_SOA1 Managed Server, you can start and validate the WLS_SOA2 Managed Server.

To start and validate the WLS_SOA2 Managed Server, use the procedure in Starting and Validating the WLS_SOA1 Managed Serverfor WLS_SOA2 Managed Server.

For validation of the URL, enter the following URL in your web browser and log in by using the enterprise deployment administrator user (`weblogic_soa`):

For Static cluster:

`http://WCCHOST2:8001/soa-infra/`

For Dynamic cluster:

`http://WCCHOST2:8002/soa-infra/`

Configuring Oracle HTTP Server for the Extended Domain

The following sections describe how to configure the Oracle HTTP Server instances so they route requests for both public and internal URLs to the proper clusters in the enterprise topology.

- Configuring Oracle HTTP Server for SOA in an Oracle WebCenter Content Enterprise Deployment
- Validating the Oracle SOA Suite URLs Through the Load Balancer

Configuring Oracle HTTP Server for SOA in an Oracle WebCenter Content Enterprise Deployment

Configure the virtual host configuration files so that requests are routed properly to the Oracle SOA Suite clusters:

1. Log in to WEBHOST1 and change directory to the configuration directory for the first Oracle HTTP Server instance (`ohs1`):

   `cd WEB_DOMAIN_HOME/config/fmwconfig/components/OHS/ohs1/moduleconf/`

2. Edit the `wccinternal_vh.conf` file and add the following directives inside the `<VirtualHost>` tags:

Note:

- The URL entry for `/workflow` is optional. It is for workflow tasks associated with Oracle ADF task forms. The `/workflow` URL itself can be a different value, depending on the form.
- Configure the port numbers appropriately as assigned for your static or dynamic cluster. Dynamic clusters with the Calculate Listen Port option selected will have incremental port numbers for each dynamic managed server created.
- The WebLogicCluster directive needs only a sufficient number of redundant `server:port` combinations to guarantee initial contact in case of a partial outage. The actual total list of cluster members is retrieve automatically upon first contact with any given node.

```
# soa-infra
<Location /soa-infra>
    WLSRequest ON
    WebLogicCluster WCCHOST1:8001,WCCHOST2:8001
    WLProxySSL OFF
    WLProxySSLPassThrough OFF
</Location>

# SOA inspection.wsil
<Location /inspection.wsil>
    WLSRequest ON
    WebLogicCluster WCCHOST1:8001,WCCHOST2:8001
    WLProxySSL OFF
    WLProxySSLPassThrough OFF
</Location>

# Worklist
<Location /integration>
    WLSRequest ON
    WebLogicCluster WCCHOST1:8001,WCCHOST2:8001
    WLProxySSL OFF
    WLProxySSLPassThrough OFF
</Location>

# UMS prefs
<Location /sdpmessaging/userprefs-ui>
    WLSRequest ON
    WebLogicCluster WCCHOST1:8001,WCCHOST2:8001
    WLProxySSL OFF
    WLProxySSLPassThrough OFF
</Location>

# Default to-do taskflow
<Location /DefaultToDoTaskFlow>
    WLSRequest ON
    WebLogicCluster WCCHOST1:8001,WCCHOST2:8001
    WLProxySSL OFF
    WLProxySSLPassThrough OFF
</Location>
```

```
    # Workflow
    <Location /workflow>
        WLSRequest ON
        WebLogicCluster WCCHOST1:8001,WCCHOST2:8001
        WLProxySSL OFF
        WLProxySSLPassThrough OFF
    </Location>

    #Required if attachments are added for workflow tasks
     <Location /ADFAttachmentHelper>
        WLSRequest ON
        WebLogicCluster WCCHOST1:8001,WCCHOST2:8001
        WLProxySSL OFF
        WLProxySSLPassThrough OFF
    </Location>

    # SOA composer application
     <Location /soa/composer>
        WLSRequest ON
        WebLogicCluster WCCHOST1:8001,WCCHOST2:8001
        WLProxySSL OFF
        WLProxySSLPassThrough OFF
    </Location>

     <Location /frevvo>
        WLSRequest ON
        WebLogicCluster WCCHOST1:8001,WCCHOST2:8001
        WLProxySSL OFF
        WLProxySSLPassThrough OFF
    </Location>
    </VirtualHost>
```

3. Copy the `wccinternal_vh.conf` to the configuration directory for the second Oracle HTTP Server instance (`ohs2`):

   WEB_DOMAIN_HOME`/config/fmwconfig/components/`**`ohs2`**`/moduleconf/`

4. Edit the `wccinternal_vh.conf` to change any references to WEBHOST1 to WEBHOST2 in the `<VirtualHost>` directives.

5. Restart both Oracle HTTP servers.

Example 13-1 Sample Content for the `wccinternal_vh.conf` File

```
<VirtualHost WEBHOST1:7777>
    ServerName http://wccinternal.example.com:80
    ServerAdmin you@your.address
    RewriteEngine On
    RewriteOptions inherit

# WSM-PM
<Location /wsm-pm>
     WebLogicCluster WCPHOST1:7010,WCPHOST2:7010
     WLSRequest ON
     WLProxySSL OFF
     WLProxySSLPassThrough OFF
</Location>

#soa-infra
<Location /soa-infra>
    WLSRequest ON
    WebLogicCluster WCCHOST1:8001,WCCHOST2:8001
    WLProxySSL OFF
```

```
    WLProxySSLPassThrough OFF
</Location>

# SOA inspection.wsil
<Location /inspection.wsil>
    WLSRequest ON
    WebLogicCluster WCCHOST1:8001,WCCHOST2:8001
    WLProxySSL OFF
    WLProxySSLPassThrough OFF
</Location>

# Worklist
<Location /integration>
    WLSRequest ON
    WebLogicCluster WCCHOST1:8001,WCCHOST2:8001
    WLProxySSL OFF
    WLProxySSLPassThrough OFF
</Location>

# UMS prefs
<Location /sdpmessaging/userprefs-ui>
    WLSRequest ON
    WebLogicCluster WCCHOST1:8001,WCCHOST2:8001
    WLProxySSL OFF
    WLProxySSLPassThrough OFF
</Location>

# Default to-do taskflow
<Location /DefaultToDoTaskFlow>
    WLSRequest ON
    WebLogicCluster WCCHOST1:8001,WCCHOST2:8001
    WLProxySSL OFF
    WLProxySSLPassThrough OFF
</Location>

# Workflow
<Location /workflow>
    WLSRequest ON
    WebLogicCluster WCCHOST1:8001,WCCHOST2:8001
    WLProxySSL OFF
    WLProxySSLPassThrough OFF
</Location>

#Required if attachments are added for workflow tasks
 <Location /ADFAttachmentHelper>
    WLSRequest ON
    WebLogicCluster WCCHOST1:8001,WCCHOST2:8001
    WLProxySSL OFF
    WLProxySSLPassThrough OFF
</Location>

# SOA composer application
 <Location /soa/composer>
    WLSRequest ON
    WebLogicCluster WCCHOST1:8001,WCCHOST2:8001
    WLProxySSL OFF
    WLProxySSLPassThrough OFF
</Location>

 <Location /frevvo>
    WLSRequest ON
```

```
    WebLogicCluster WCCHOST1:8001,WCCHOST2:8001
    WLProxySSL OFF
    WLProxySSLPassThrough OFF
</Location>
</VirtualHost>
```

Validating the Oracle SOA Suite URLs Through the Load Balancer

To validate the configuration of the Oracle HTTP Server virtual hosts and to verify that the hardware load balancer can route requests through the Oracle HTTP Server instances to the application tier:

1. Verify that the server status is reported as **Running** in the Administration 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.

2. Verify that you can access these URLs:
   - `http://wccinternal.example.com/soa-infra`
   - `http://wccinternal.example.com/integration/worklistapp`
   - `http://wccinternal.example.com/sdpmessaging/userprefs-ui`
   - `http://wccinternal.example.com/soa/composer`
   - `http://wccinternal.example.com/wsm-pm`

Post-Configuration Steps for Oracle SOA Suite

After you install and configure Oracle SOA Suite, consider the following post-configuration tasks.

- Configuring Oracle Adapters for Oracle SOA Suite
- Enabling SSL Communication Between the SOA Servers and the Hardware Load Balancer
- Considerations for Sync-Async Interactions in a SOA Cluster
- Updating FusionAppsFrontendHostUrl

Configuring Oracle Adapters for Oracle SOA Suite

If the Oracle SOA Suite applications that you are developing take advantage of any of the Oracle adapters for Oracle SOA Suite, then you should make sure that the adapters are configured to work efficiently and securely in the enterprise topology.

See the following topics for more information.

- Enabling High Availability for Oracle File and FTP Adapters
- Enabling High Availability for Oracle JMS Adapters
- Enabling High Availability for the Oracle Database Adapter

Enabling High Availability for Oracle File and FTP Adapters

If the Oracle SOA Suite applications that you are developing or deploying require the Oracle File and FTP Adapters, you must configure the adapters for high availability in the enterprise deployment topology.

Use the following sections to complete this task.

- Understanding the Oracle File and FTP Adapter Configuration
- Configuring the Oracle File Adapter in the Administration Console
- Editing the JCA File Within the Composite Application
- Configuring the Oracle FTP Adapter

Understanding the Oracle File and FTP Adapter Configuration

The Oracle File and FTP adapters enable a BPEL process or an Oracle Mediator to read and write files on private file systems and on remote file systems through the File Transfer Protocol (FTP).

When configured properly, these adapters support high availability for an active-active topology with Oracle BPEL Process Manager and Oracle Mediator service engines for both inbound and outbound operations.

For general information about this task, see Configuring Oracle File and FTP Adapters in *Oracle Fusion Middleware Understanding Technology Adapters*. The instructions provided here are specific to the Oracle SOA Suite enterprise deployment.

> **Note:**
>
> The File Adapter picks up a file from the inbound directory, processes it, and then outputs a file to the output directory. Because the File Adapter is non-transactional, files can be processed twice. As a result, it is possible to get duplicate files when there is failover in the RAC backend or in the SOA managed servers.

Configuring the Oracle File Adapter in the Administration Console

To make the Oracle File Adapter highly available, first modify the Oracle File Adapter deployment descriptor for the connection-instance that corresponds to `eis/HAFileAdapter`.

You can perform this task from the Oracle WebLogic Server console:

1. Navigate to and log into the Oracle WebLogic Server Administration Console.

   For example:

   `http://ADMINVHN:7001/console`

> **Note:**
>
> If you have already configured web tier, use `http://admin.example.com/console`.

2. In the left pane of the console, click **Deployments**.
3. Locate the **FileAdapter** resource adapter in the Summary of Deployments table.
4. Click **FileAdapter** to display the Settings for FileAdapter page.
5. Click **Configuration**.
6. Click **Outbound Connection Pools**.
7. Expand **javax.resource.cci.ConnectionFactory** to see the configured connection factories.
8. Click **eis/HAFileAdapter**.

   The Outbound Connection Properties for the connection factory appears.
9. Click **Lock & Edit**.

   The property value column becomes editable (you can click on any of the rows in the Property Value column and modify the value).
10. Enter the values as shown in Table 13-2.

> **Note:**
>
> Update controlDir and check other values against the default values as mentioned in Table 13-2.

Table 13-2 Values to Provide for the javax.resource.cci.Connectionfactory

Parameter	Description
controlDir	Enter the directory where you want the control files to be stored. You must set it to a shared location if multiple WebLogic Server instances run in a cluster. Structure the directory for shared storage as follows: *ORACLE_RUNTIME*/*domain_name*/*cluster_name*/fadapter
inboundDataSource	Set the value to `jdbc/SOADataSource`.
outboundDataSource	Set the value to `jdbc/SOADataSource`.
outboundDataSourceLocal	Set the value to `jdbc/SOALocalTxDataSource`. This is the data source where the schemas that corresponds to high availability are precreated.

Table 13-2 (Cont.) Values to Provide for the javax.resource.cci.Connectionfactory

Parameter	Description
outboundLockTypeForWrite	Set the value to `oracle` if you are using Oracle Database. By default the Oracle File and FTP Adapters use an in-memory mutex to lock outbound write operations. You must choose from the following values for synchronizing write operations: • `memory`: The Oracle File and FTP Adapters use an in-memory mutex to synchronize access to the file system. • `oracle`: The adapter uses Oracle Database sequence. • `db`: The adapter uses a pre-created database table (`FILEADAPTER_MUTEX`) as the locking mechanism. You must use this option only if you are using a schema other than the Oracle Database schema. • `user-defined`: The adapter uses a user-defined mutex. To configure the user-defined mutex, you must implement the mutex interface: `oracle.tip.adapter.file.Mutex` and then configure a new binding-property with the name `oracle.tip.adapter.file.mutex` and value as the fully qualified class name for the mutex for the outbound reference.
workingDirectory	Retain the default value.

11. Click **Save** after you update the properties. The Save Deployment Plan page appears.
12. Create DEPLOY_PLAN_HOME directory.

    ```
    mkdir -p DEPLOY_PLAN_HOME/wccedg_domain
    ```

    In this example, replace *DEPLOY_PLAN_HOME* with the actual path to the deployment plan directory that is defined in File System and Directory Variables Used in This Guide.
13. Enter a shared storage location for the deployment plan **path** value. The directory structure is as follows:

    ```
    DEPLOY_PLAN_HOME/wccedg_domain/FileAdapterPlan.xml
    ```
14. Click **OK** to save the storage location.
15. Click **Save** to save and then click **Activate Changes** to apply your changes to the File Adapter.
16. Update the deployment in the console:
    a. Click **Deployments**.
    b. Click **Lock & Edit.**
    c. Select the checkbox for the **File Adapter deployment**.
    d. Click **Update**.
    e. Select the option: **Update this application in place with new deployment plan changes (A deployment plan must be specified for this option.)**
    f. Click the **Change Path** button and select the **FileAdapterPlan.xml** file from the path to the shared storage location.
    g. Click **Finish**.
    h. Activate the changes.
17. Verify that the FileAdapter deployment is activated and running:

a. In the Administration Console, click **Deployments** in the left pane.

b. Locate the FileAdapter deployment in the Deployments table.

c. If it is not in the active state, click the Control tab under **Summary of Deployments**, and then select **FileAdapter** under **Deployments**. Select **Start**, and then **Servicing All Requests**.

d. Click **Yes**.

Editing the JCA File Within the Composite Application

After you have configured the FileAdapter deployment in the Administration Console, you can edit the .jca file that is included in the composite applications to be deployed so that they can use the connection factory that was configured in the previous steps, as shown in Example 13-2.

> **Note:**
>
> The location attribute is set to `eis/HAFileAdapter` for the connection factory.

Example 13-2 Example of the File Adapter .JCA File Modifications for an Enterprise Deployment

```
<adapter-config name="FlatStructureOut"
                adapter="File Adapter"
                xmlns="http://platform.integration.oracle/blocks/adapter/fw/metadata">
     <connection-factory location="eis/HAFileAdapter" adapterRef=""/>
     <endpoint-interaction portType="Write_ptt"
                          operation="Write">
          <interaction-spec className="oracle.tip.adapter.file.outbound.FileInteractionSpec">
                <property../>
                <property../>
          </interaction-spec>
     </endpoint-interaction>
</adapter-config>
```

Configuring the Oracle FTP Adapter

If your application requires an FTP Adapter, then repeat the procedures Configuring the Oracle File Adapter in the Administration Console and Editing the JCA File Within the Composite Application, with the following differences:

- Locate the **FtpAdapter** deployment in the list of deployments in the Administration Console.
- Click **FtpAdapter** to display the Settings for the FtpAdapter page.
- Click **Configuration**.
- Click **Outbound Connection Pools**.
- Expand **javax.resource.cci.ConnectionFactory** to see the configured connection factories.
- Click **eis/Ftp/HAFtpAdapter**.

  The Outbound Connection Properties for the connection factory appears.

- Click **Lock & Edit**.
- Modify the adapter properties for high availability. See Table 13-2.
- Update the ControlDir property so it points to the following location:

  ORACLE_RUNTIME/*domain_name*/*cluster_name*/ftpadapter
- Enter a shared storage location for the deployment plan. The directory structure is as follows:

  DEPLOY_PLAN_HOME/wccedg_domain/FtpAdapterPlan.xml

Enabling High Availability for Oracle JMS Adapters

When the Oracle JMS adapter communicates with multiple servers in a cluster, the adapter's connection factory property `FactoryProperties` must list available servers. If it does not list servers, the connection establishes to only one random server. If that particular server goes down, no further messages are processed.

To verify the adapter's JCA connection factory:

1. Log into your Oracle WebLogic Server Administration Console using the following URL:

   http://*ADMINVHN*:7001/console

   > **Note:**
   >
   > If you have already configured Web tier, use `http://admin.example.com/console`.

2. Click **Deployments** in the left pane for Domain Structure.
3. Click **JmsAdapter** under **Summary of Deployments** on the right pane.
4. Click the **Configuration** tab.
5. Click the **Outbound Connection Pools** tab and expand `oracle.tip.adapter.jms.IJmsConnectionFactory` to see the configured connection factories.
6. Click **Lock & Edit**.
7. Click the specific instance you are using (for example, `eis/wls/Queue`). The Outbound Connection Properties for the connection factory opens.
8. In the **FactoryProperties** field (click on the corresponding cell under Property value), enter the following, all on one line, separated by semicolons:

   ```
   java.naming.factory.initial=weblogic.jndi.WLInitialContextFactory;
   java.naming.provider.url=cluster:t3://SOA_Cluster;
   java.naming.security.principal=weblogic;
   java.naming.security.credentials=mypassword
   ```

9. Click **Save** after you update the properties. The Save Deployment Plan page appears.
10. (First time only) Enter a shared storage location for the deployment plan. The directory structure is as follows:

DEPLOY_PLAN_HOME/soaedg_domain/JMSAdapterPlan.xml

11. Click **OK** to commit the updated storage path.
12. Click **Save**.
13. Repeat steps 7 through 9 for all required connection factories.
14. Click **Activate Changes**.
15. Update the deployment in the console:
    a. Click **Deployments**.
    b. Click **Lock & Edit.**
    c. Select the checkbox for the **JMS Adapater**.
    d. Click **Update**.
    e. Select **Update this application in place with new deployment plan changes (A deployment plan must be specified for this option.)** and select the deployment plan saved in a shared storage location; all servers in the cluster must be able to access the plan.
    f. Click **Finish**.
    g. Activate the changes.

Enabling High Availability for the Oracle Database Adapter

To ensure High Availability while leveraging the Oracle Database Adapter, the Logical Delete Polling Strategy is used normally as it performs better than a physical delete. However, when you have a clustered environment where multiple nodes are polling for the same data, a single record might get processed more than once. To avoid this problem, Oracle Database Adapter uses a distributed polling technique that uses an Oracle Database feature called skip locking.

If you were using the Logical Delete Polling Strategy approach previously, you can remove (in `db.jca`) or clear (Logical Delete Page of wizard) the `MarkReservedValue`, and you automatically get skip locking.

The benefits of using skip locking over a reserved value include:

- Skip locking scales better in a cluster and under load.
- All work is in one transaction (as opposed to update/reserve, then commit, then select in a new transaction), so the risk of facing a non-recoverable situation in a high availability environment is minimized.
- No unique MarkReservedValue must be specified. Previously, for this to work you would have to configure a complex variable, such as `R${weblogic.Name-2}-${IP-2}-${instance}`.

If you are using Logical Delete polling, and you set `MarkReservedValue`, skip locking is not used.

For more information, see "Scalability" and "Polling Strategies" in the Oracle Fusion Middleware User's Guide for Technology Adapters.

Enabling SSL Communication Between the SOA Servers and the Hardware Load Balancer

After you extend the domain with Oracle SOA Suite, you should also ensure that the Administration Server and Managed Servers can access the front-end SSL URL of the hardware load balancer.

This allows SOA Composite applications and web services to invoke callbacks and other communications with the front-end secure URL. See Enabling SSL Communication Between the Middle Tier and the Hardware Load Balancer.

Considerations for Sync-Async Interactions in a SOA Cluster

In a SOA cluster, the following scenarios are not supported:

- Synchronous BPEL process with mid-process receive.
- Synchronous BPEL process calling asynchronous services.
- Callback from synchronous processes.

Updating FusionAppsFrontendHostUrl

You must configure Oracle Workflow with the appropriate URL so that the default-to-do tasks and custom tasks' details use the front-end load balancer to create task-display URLs.
To configure the appropriate URLs:

1. Log in to Oracle Enterprise Manager Fusion Middleware Control with the username and password that you specified in the `boot.properties` file. See Creating the boot.properties File.
2. In the left navigation tree, expand **WebLogic Domain**, and then click **System MBean Browser**.
3. Navigate to **Application Defined Mbean**> **oracle.as.soainfra.config**.
   a. If you are configuring a static cluster, navigate to **Server: WLS_SOA1**>**WorkflowConfig**.
   b. If you are configuring a dynamic cluster, navigate to **Domain: wccedg_domain** >**WorkflowConfig**.
4. Click **human-workflow**.

> **Note:**
>
> In a clustered environment, there are multiple human-workflow Mbeans, one for every server in the cluster. Modify any one of them to update the property centrally in MDS for the entire cluster.

5. On the right panel, look for the **FusionAppsFrontendHostUrl** attribute.
6. For the **FusionAppsFrontendHostUrl** attribute, specify the value `*=https://wcc.example.com:443`.

7. Click **Apply**.

Enabling JDBC Persistent Stores

Oracle recommends that you use JDBC stores, which leverage the consistency, data protection, and high availability features of an oracle database and makes resources available for all the servers in the cluster.

Follow these guidelines to ensure that you use JDBC stores, when you use static or dynamic clusters:

- **For static clusters**

  If you have made the following selections in the High Availability Options screen, as recommended in this guide for static clusters, then JDBC persistent stores are already configured for both JMS and TLOGS:

  – Set **JTA Transaction Log Persistence** to **JDBC TLog Store**.

  – Set **JMS Server Persistence** to **JMS JDBC Store**.

- **For dynamic clusters**

  You can configure only JMS Server persistence for dynamic clusters by using the Configuration Wizard. JTA Transaction Logs Persistence must be configure manually, if required. If you have made the following selections in the High Availability Options screen, as recommended in this guide for dynamic clusters, then JDBC persistent stores are already configured for JMS.

  – Set **JMS Server Persistence** to **JMS JDBC Store**.

  – Verify that **JTA Transaction Log Persistence** is set to **Default Persistent Store**.

  Additional steps are needed to configure JTA Transaction Log with JDBC store. See Roadmap for Configuring a JDBC Persistent Store for TLOGs.

In case you did not select JDBC for JMS and TLOGS persistent in the High Availability Options screen, you can still configure JDBC stores manually in a post step. For specific instructions to configure them manually, see Using JDBC Persistent Stores for TLOGs and JMS in an Enterprise Deployment.

> **Note:**
>
> The High Availability Options screen appears during the Configuration Wizard session for the first time when you create a cluster that uses Automatic Service Migration or JDBC stores or both. All subsequent clusters that are added to the domain by using the Configuration Wizard, automatically apply the selected HA options.

Enabling Automatic Service Migration

To ensure high availability for the product installed in this chapter, you must configure service migration appropriately.

Follow these guidelines to ensure that you provide the required high availability for Weblogic services when you use static or dynamic clusters:

- **For static clusters**

  Automatic Service Migration is already configured if you select **Enable Automatic Service Migration** with **Database Basis** in the High Availability Options screen.

  The Database Leasing is already configured and the migratable targets are created with the appropriate policies for the cluster. If you have implemented these settings, validate the configuration, as described in Validating Automatic Service Migration in Static Clusters.

  In case you do not select this option during the Configuration Wizard session, you can configure automatic migration manually in a post step. For instructions to complete the steps for static clusters, see Configuring Automatic Service Migration in an Enterprise Deployment.

- **For dynamic clusters**

  You cannot configure Service Migration for dynamic clusters by using the Configuration Wizard, it needs to be configured manually. The following steps are needed:

  – Configure the database leasing for the cluster.

  – Set the appropriate migration policies for JTA Service and JMS Persistent Stores.

  For instructions to complete the steps for dynamic clusters, see Configuring Automatic Service Migration in an Enterprise Deployment.

> **Note:**
>
> The High Availability Options screen appears during the Configuration Wizard session for the first time when you create a cluster that uses Automatic Service Migration or JDBC stores or both. All subsequent clusters that are added to the domain by using the Configuration Wizard, automatically apply the selected HA options.

14

Extending the Domain to Include Inbound Refinery

You need to perform certain tasks in order to extend the enterprise deployment domain to include Inbound Refinery software.

- Overview of Extending the Domain to Include Inbound Refinery
  Inbound Refinery is required for document conversion by Oracle WebCenter Content Server.
- Extending the Domain for Inbound Refinery
  The instructions for extending the existing enterprise deployment domain with the Inbound Refinery software are detailed in this section.
- Completing Postconfiguration and Verification Tasks for Inbound Refinery
  After extending the domain with the Inbound Refinery software, consider the following post-configuration and verification tasks.
- Configuring the Inbound Refinery Managed Servers
  To initialize the configuration of an Inbound Refinery Managed Server, you need to access it only once through HTTP. You can do this directly at the Managed Server's listen address. An Inbound Refinery instance should not be placed behind an HTTP server.
- Validating the Configuration of the Inbound Refinery Managed Servers
  To ensure that the Inbound Refinery Managed Servers you have created are properly configured, validate the configuration by logging in to Content Server and verifying that a file with an extension recognized as valid for conversion is correctly converted.

Overview of Extending the Domain to Include Inbound Refinery

Inbound Refinery is required for document conversion by Oracle WebCenter Content Server.

The actual number of Inbound Refinery Managed Servers varies depending on requirements. For availability reasons, Oracle recommends configuring at least two Inbound Refinery Managed Servers, each installed and configured on a separate machine. In the reference Oracle WebCenter Content enterprise deployment topology, Inbound Refinery will be configured on the same machine as Content Server.

Even though multiple Managed Servers are created in the process of extending the domain with Inbound Refinery in this enterprise deployment topology, each Inbound Refinery instance is completely independent. Inbound Refinery does not run in a cluster.

Extending the Domain for Inbound Refinery

The instructions for extending the existing enterprise deployment domain with the Inbound Refinery software are detailed in this section.

- Starting the Configuration Wizard
  Start the Configuration Wizard as the first step to extend the existing enterprise deployment domain.
- Navigating the Configuration Wizard Screens to Extend the Domain

Starting the Configuration Wizard

Start the Configuration Wizard as the first step to extend the existing enterprise deployment domain.

> **Note:**
>
> If you added any customizations directly to the start scripts in the domain, those are overwritten by the configuration wizard. To customize server startup parameters that apply to all servers in a domain, you can create a file called `setUserOverridesLate.sh` and configure it, for example, add custom libraries to the WebLogic Server classpath, specify Additional JAVA command line options for running the servers, or specify additional environment variables. Any customizations you add to this file are preserved during domain upgrade operations, and are carried over to remote servers when using the **`pack`** and **`unpack`** commands.

To start the Configuration Wizard:

1. From the WebLogic Server Console, stop any managed servers that are modified by this domain extension. Managed Servers that are not effected can remain on-line.
2. For any managed servers to be modified, verify that the managed server shutdown has completed.
3. Stop the Administration Server once all managed servers are in a steady state.
4. Navigate to the following directory and start the WebLogic Server Configuration Wizard.

   ```
   cd ORACLE_HOME/oracle_common/common/bin
   ./config.sh
   ```

Navigating the Configuration Wizard Screens to Extend the Domain

Follow the instructions in this section to update and configure the domain for the topology.

> **Note:**
>
> You can use the same procedure described in this section to extend an existing domain. If your needs do not match the instructions given in the procedure, be sure to make your selections accordingly, or refer to the supporting documentation for additional details.

Domain creation and configuration includes the following tasks:

Task 1 Selecting the Domain Type and Domain Home Location
On the Configuration Type screen, select **Update an existing domain**.
In the **Domain Location** field, select the value of the *ASERVER_HOME* variable, which represents the complete path to the initial Administration Server domain home you created.
For more information about the directory location variables, see File System and Directory Variables Used in This Guide.

> **Tip:**
>
> More information about the other options on this screen can be found in Configuration Type in *Creating WebLogic Domains Using the Configuration Wizard*.

Click **Next** to proceed.

Task 2 Selecting the Configuration Template
On the Templates screen, make sure **Update Domain Using Product Templates** is selected, then select the following templates:

- **Oracle Universal Content Management - Inbound Refinery - 12.2.1.3.0 [wccontent]**

  The following additional templates should already be selected, because they were used to create the initial domain:

  - **Oracle Enterprise Manager - 12.2.1.3.0 [em]**
  - **Oracle SOA Suite - 12.2.1.3.0 [soa]**
  - **Oracle WebCenter Content - 12.2.1.3.0 [wcc]**
  - **Oracle WSM Policy Manager - 12.2.1.3.0 [oracle_common]**
  - **Oracle JRF - 12.2.1.3.0 [oracle_common]**
  - **WebLogic Coherence Cluster Extension - 12.2.1.3.0 [wlserver]**

> **Tip:**
>
> More information about the options on this screen can be found in Templates in *Creating WebLogic Domains Using the Configuration Wizard*.

Click **Next** to proceed.

Task 3 Providing the GridLink Oracle RAC Database Connection Details
On the GridLink Oracle RAC Component Schema screen, click **Next**.

Task 4 Testing the JDBC Connections
Click **Next** to continue.

Task 5 Selecting Advanced Configuration
To complete domain configuration for the topology, select the following option on the Advanced Configuration screen:
Topology
Click **Next** to proceed.

Task 6 Configuring Managed Servers
On the Managed Servers screen, a new Managed Server appears in the list of servers.
Perform the following tasks to modify the default Managed Server and create a second Managed Server:

1. Rename the default Managed Server to `WLS_IBR1`.
2. Click **Add** to create a new Managed Server and name it `WLS_IBR2`.

> **Tip:**
>
> The server names recommended here will be used throughout this document. If you choose different names be sure to replace them as needed.

3. Use the information in the following table to fill in the rest of the columns for each Managed Server.

Server Name	Listen Address	Listen Port	Enable SSL	SSL Listen Port	Server Group
WLS_IBR1	WCCHOST1	16250	No	Disabled	IBR-MGD-SVR
WLS_IBR2	WCCHOST2	16250	No	Disabled	IBR-MGD-SVR

> **Tip:**
>
> More information about the options on the Managed Server screen can be found in Managed Servers in *Creating WebLogic Domains Using the Configuration Wizard*.

Click **Next** to proceed.

Task 7 Configuring a Cluster
In this task, you create a cluster of Managed Servers to which you can target the Oracle Inbound Refinery software.
Use the Clusters screen to create a new cluster:

1. Click the **Add** button.
2. Specify `IBR_Servers` in the **Cluster Name** field.

3. From the **Dynamic Server Groups** drop-down list, select `Unspecified`.
4. Click **Next** to proceed to the next screen.

> **Note:**
>
> By default, server instances in a cluster communicate with one another using unicast. If you want to change your cluster communications to use multicast, refer to "Considerations for Choosing Unicast or Multicast" in *Administering Clusters for Oracle WebLogic Server*.

> **Tip:**
>
> More information about the options on this screen can be found in Clusters in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 8 Assigning Server Templates
Click **Next** to proceed to the next screen.

Task 9 Configuring Dynamic Servers
Verify that all dynamic server options are disabled for clusters that are to remain as static clusters.

1. Confirm that the **Dynamic Cluster**, **Calculated Listen Port**, and **Calculated Machine Names** checkboxes on this screen are unchecked.
2. Confirm the **Server Template** selection is **Unspecified**.
3. Click **Next** to proceed.

Task 10 Assigning Managed Servers to the Cluster
Use the Assign Servers to Clusters screen to assign `WLS_IBR1` and `WLS_IBR2` to the new cluster `IBR_Servers`:

1. In the Clusters pane, select the cluster to which you want to assign the servers; in this case, `IBR_Servers`.
2. In the Servers pane, assign WLS_IBR1 to `IBR_Servers` by doing one of the following:
   - Click once on `WLS_IBR1` Managed Server to select it, then click on the right arrow to move it beneath the selected cluster in the Clusters pane.
   - Double-click `WLS_IBR1` to move it beneath the selected cluster in the clusters pane.
3. Repeat to assign `WLS_IBR2` to `IBR_Servers`.

> **Tip:**
>
> More information about the options on this screen can be found in Assign Servers to Clusters in *Creating WebLogic Domains Using the Configuration Wizard*.

4. Click **Next** to proceed.

Task 11 Configuring Coherence Clusters
Use the Coherence Clusters screen to configure the Coherence cluster that is automatically added to the domain. Leave the port number value at `9991`, as it was defined during the initial Infrastructure domain creation.

> **Note:**
>
> For Coherence licensing information, refer to *Oracle Coherence* in *Oracle Fusion Middleware Licensing Information*.

Click **Next** to proceed.

Task 12 Verifying the Existing Machines
Under the **Unix Machine** tab, verify the names of the machines you created when creating the initial Infrastructure domain.
Click **Next** to proceed.

Task 13 Assigning Servers to Machines
Use the Assign Servers to Machines screen to assign the Oracle Inbound Refinery Managed Servers you just created to the corresponding machines in the domain. Assign `WLS_IBR1` to WCCHOST1, and assign `WLS_IBR2` to WCCHOST2.

> **Tip:**
>
> More information about the options on this screen can be found in Assign Servers to Machines in *Creating WebLogic Domains Using the Configuration Wizard*.

Click **Next** to proceed.

Task 14 Reviewing Virtual Targets
Click **Next** to proceed to the next screen.

Task 15 Reviewing Partitions
Click **Next** to proceed to the next screen.

Task 16 Reviewing Your Configuration Specifications and Configuring the Domain
The Configuration Summary screen contains the detailed configuration information for the domain. Review the details of each item on the screen and verify that the information is correct.

You can go back to any previous screen if you need to make any changes, either by using the **Back** button or by selecting the screen in the navigation pane.
Click **Update** to execute the domain extension.

> **Tip:**
>
> More information about the options on this screen can be found in Configuration Summary in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 17 Writing Down Your Domain Home and Administration Server URL
The Configuration Success screen will show the following items about the domain you just configured:

- Domain Location
- Administration Server URL

You must make a note of both items as you will need them later; the domain location is needed to access the scripts used to start the Node Manager and Administration Server, and the URL is needed to access the Administration Server.
Click **Finish** to dismiss the Configuration Wizard.

Task 18 Start the Administration Server
Start the Administration Server to ensure the changes you have made to the domain have been applied.

Completing Postconfiguration and Verification Tasks for Inbound Refinery

After extending the domain with the Inbound Refinery software, consider the following post-configuration and verification tasks.

- Propagate the Domain Configuration Updates for Inbound Refinery
- Starting the Node Manager in the Managed Server Domain Directory
- Starting the Inbound Refinery Managed Servers

Propagate the Domain Configuration Updates for Inbound Refinery

Propagate the start scripts and classpath configuration from the Administration Server's domain directory to the Managed Server domain directory. To propagate the domain configuration to the Inbound Refinery Managed Servers:

1. Create a copy of the Managed Server domain directory and the Managed Server `applications` directory.
2. Run the following `pack` command on WCCHOST1 to create a template pack:

```
cd ORACLE_COMMON_HOME/common/bin

./pack.sh -managed=true -domain=ASERVER_HOME -
template=edgdomaintemplateExtIBR.jar -template_name=edgdomain_templateIBR
```

3. Run the following `unpack` command on WCCHOST1 to propagate the template created in the preceding step to the `WLS_IBR1` domain directory:

   ```
   cd ORACLE_COMMON_HOME/common/bin

   ./unpack.sh -domain=MSERVER_HOME -template=edgdomaintemplateExtIBR.jar -
   app_dir=APPLICATION_HOME -overwrite_domain=true
   ```

4. Run the following command on WCCHOST1 to copy the template pack created in step 1 to WCCHOST2:

   ```
   scp edgdomaintemplateIBR.jar oracle@WCCHOST2:ORACLE_COMMON_HOME/common/bin
   ```

5. Run the `unpack` command on WCCHOST2 to unpack the propagated template to the `WLS_IBR1` domain directory.

   ```
   cd ORACLE_COMMON_HOME/common/bin

   ./unpack.sh -domain=MSERVER_HOME -template=edgdomaintemplateExtIBR.jar -
   app_dir=APPLICATION_HOME -overwrite_domain=true
   ```

Starting the Node Manager in the Managed Server Domain Directory

Follow these steps to start the Node Manager from the Managed Server home:

1. On WCCHOST1, navigate to the following directory:

   MSERVER_HOME`/bin`

2. Use the following command to start the Node Manager:

   ```
   nohup ./startNodeManager.sh > ./nodemanager.out 2>&1 &
   ```

3. Change directory to the following directory on WCCHOST2 and repeat the preceding step.

   MSERVER_HOME`/bin`

For information about additional Node Manager configuration options, see *Administering Node Manager for Oracle WebLogic Server*.

Starting the Inbound Refinery Managed Servers

To start the `WLS_IBR1` Managed Server on WCCHOST1:

1. Log in to the Oracle WebLogic Server Administration Console as the `weblogic_wcc` user at:

   ```
   http://admin.example.com/console
   ```

2. Start the `WLS_IBR1` Managed Server through the Administration Console, as follows:

   a. Expand the **Environment** node in the **Domain Structure** tree on the left.

   b. Click **Servers**.

   c. On the Summary of Servers page, click the **Control** tab.

   d. Select **WLS_IBR1** from the **Servers** column of the table.

   e. Click **Start**.

3. Verify that the server status is reported as `Running` in the Administration 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.

4. Repeat the preceding steps to start the `WLS_IBR2` Managed Server on WCCHOST2.

Configuring the Inbound Refinery Managed Servers

To initialize the configuration of an Inbound Refinery Managed Server, you need to access it only once through HTTP. You can do this directly at the Managed Server's listen address. An Inbound Refinery instance should not be placed behind an HTTP server.

All subsequent access to the Inbound Refinery instance is through the socket listener. This listener is protected through the incoming socket connection address security filter configured in the next section.

Oracle recommends configuring each Content Server instance with all Inbound Refinery instances. The process for configuring Content Server is to add each Inbound Refinery instance as a provider. You also need to perform some post-installation steps with Inbound Refinery.

The following sections describe the procedures for post-installation configuration of each Inbound Refinery instance.

- Configuring Inbound Refinery Settings
  After starting the Inbound Refinery Managed Servers, configure the settings for each server on its post-installation configuration screen.
- Setting Up Content Server to Send Jobs to Inbound Refinery for Conversion
  Before Oracle WebCenter Content Server can send jobs to Inbound Refinery for conversion, you need to perform the setup tasks described in the following sections for each Inbound Refinery Managed Server.

Configuring Inbound Refinery Settings

After starting the Inbound Refinery Managed Servers, configure the settings for each server on its post-installation configuration screen.

To configure the settings for each Inbound Refinery instance, complete the following steps:

1. Create unique IBR directories on the *`ORACLE_RUNTIME`* shared filesystem for each IBR server as required for the Oracle WebCenter Content Inbound Refinery configuration. The Oracle WebCenter Content Inbound Refinery configuration requires a unique and separate directory for each IBR instance's runtime files. The EDG architecture recommends using the *`ORACLE_RUNTIME`* shared filesystem consistently for all runtime file-based data storage. The recommended base path for the Oracle WebCenter Content Inbound Refinery runtime file storage is *`ORACLE_RUNTIME`*`/domain_name/IBR_Servers/`

> **Note:**
>
> The IBR servers do not share file-based data between instances. Unlike the Content Server instances, there is no product-specific requirement to implement a shared filesystem for the IBR data. Use of the shared filesystem for IBR data is for architectural consistency and DR replication efficiency.

Run the following commands to create the required unique subdirectories for each IBR managed server:

```
mkdir -p ORACLE_RUNTIME/domain_name/IBR_Servers/ibr1/vault
mkdir -p ORACLE_RUNTIME/domain_name/IBR_Servers/ibr1/weblayout
mkdir -p ORACLE_RUNTIME/domain_name/IBR_Servers/ibr1/data/users/
profiles
mkdir -p ORACLE_RUNTIME/domain_name/IBR_Servers/ibr2/vault
mkdir -p ORACLE_RUNTIME/domain_name/IBR_Servers/ibr2/weblayout
mkdir -p ORACLE_RUNTIME/domain_name/IBR_Servers/ibr2/data/users/
profiles
```

2. Access the Inbound Refinery post-installation configuration screen at the following URL for each WCCHOST:

   `http://WCCHOSTN:16250/ibr/`

3. On the Configuration screen, you will see **Inbound Refinery Instance Identifier:** *name*. Set the remaining configuration settings for this instance as follows.

> **Note:**
>
> Each Inbound Refinery instance and associated runtime file repository directory are unique and independent of the other instances. Use the specific directory paths just created in this section for the corresponding configuration settings of each instance. Inbound Refinery Instance Folder: Set this to *ORACLE_RUNTIME*/domain_name/IBR_Servers/ibrN

- **Inbound Refinery Instance Folder**: Set this to *ORACLE_RUNTIME*/*domain_name*/IBR_Servers/ibr*N*

  For example: `/u01/oracle/runtime/wccedg_domain/IBR_Servers/ibr1`

- **Native File Repository Location**: Set this to *ORACLE_RUNTIME*/*domain_name*/IBR_Servers/ibr*N*/vault

  For example: `/u01/oracle/runtime/wccedg_domain/IBR_Servers/ibr1/vault`

- **WebLayout Folder**: Set this to *ORACLE_RUNTIME*/*domain_name*/IBR_Servers/ibr*N*/weblayout

  For example: `/u01/oracle/runtime/wccedg_domain/IBR_Servers/ibr1/weblayout`

- **User Profile Folders**: Set this to *ORACLE_RUNTIME*/*domain_name*/IBR_Servers/ibr*N*/data/users/profiles

For example: `/u01/oracle/runtime/wccedg_domain/IBR_Servers/ibr1/data/users/profiles`

- **Incoming Socket Connection Address Security Filter**: A pipe-delimited list of localhost and the server IP addresses:

  `127.0.0.1|0:0:0:0:0:0:0:1|WCCHOST1-IP|WCCHOST2-IP|WEBHOST1-IP|WEBHOST2-IP`

  This setting enables access from Content Server. The values for *WCCHOST1-IP* and *WCCHOST2-IP* should be the IP addresses of the machines with the Content Server instance or instances that will send jobs to Inbound Refinery, not necessarily the IP address of Inbound Refinery. (In the reference topology used in this enterprise deployment guide, however, these IP addresses are the same.)

  The **Incoming Socket Connection Address Security Filter:** field accepts wildcards in the value; for example, 192.0.2.*.

  You can change this value later by setting `SocketHostAddressSecurityFilter` in the`/u02/oracle/runtime/wccedg_domain/IBR_Servers/ibrN/config/config.cfg` file and then restarting the Inbound Refinery Managed Server.

  Where *N* is 1 for `http://WCCHOST1:16250/ibr/` and *N* is 2 for `http://WCCHOST2:16250/ibr/`

- **Server Socket Port:** Enter an unused port number, such as 5555. This value is the number of the port for calling top-level services.

  Take note of the port number because you need it later for configuring Oracle WebCenter Content.

  Changing this field value changes the `IntradocServerPort` entry in `/u01/oracle/runtime/wccedg_domain/IBR_Servers/ibrN/config/config.cfg`

  Where *N* is 1 for `http://WCCHOST1:16250/ibr/` and *N* is 2 for `http://WCCHOST2:16250/ibr/`

- **Server Instance Name:** Specify a name for the Inbound Refinery server instance.

  You can accept the default value or change it to a name that is more useful to you. Take note of the server name because you will need it later for configuring Oracle WebCenter Content.

You can leave all other fields on the configuration page as they are.

Click **Submit**, and you should get the following message:

```
Post-install configuration complete. Please restart this node.
```

4. Restart the Inbound Refinery Managed Server, using the WebLogic Server Administration Console.
5. Repeat the preceding steps for each Inbound Refinery instance, using different names for the content folders.

Setting Up Content Server to Send Jobs to Inbound Refinery for Conversion

Before Oracle WebCenter Content Server can send jobs to Inbound Refinery for conversion, you need to perform the setup tasks described in the following sections for each Inbound Refinery Managed Server.

- Creating an Outgoing Provider
  Before Content Server can send files to Inbound Refinery for conversion, you must set up an outgoing provider from Content Server to each Inbound Refinery with the **Handles Inbound Refinery Conversion Jobs** option checked.
- Enabling Components for Inbound Refinery on Content Server
  Some conversion types require *helper* components to be enabled on Content Server. The InboundRefinerySupport component must always be enabled on any Content Server instance that uses Inbound Refinery for document conversion. It is enabled by default on a new Content Server installation.
- Selecting File Formats To Be Converted
  To tell Content Server which files to send to Inbound Refinery to be converted, you need to select file formats.

Creating an Outgoing Provider

Before Content Server can send files to Inbound Refinery for conversion, you must set up an outgoing provider from Content Server to each Inbound Refinery with the **Handles Inbound Refinery Conversion Jobs** option checked.

To create an outgoing provider for each Inbound Refinery instance:

1. Log in to Content Server at the following URL:

   `http://WCCHOST1:16200/cs/`

2. Open the **Administration** tray or menu, then choose **Providers**.
3. In the **Create a New Provider** table of the Providers page, click **Add** in the **outgoing** row.
4. Enter the following values for the fields:
   - **Provider Name:** Any short name with no spaces. It is a good idea to use the same value as the **Instance Name** value
   - **Provider Description:** Any text string.
   - **Server Host Name:** The name of the host machine where the Inbound Refinery instance is running: `WCCHOST1`.
   - **HTTP Server Address:** The address of the Inbound Refinery instance: `WCCHOST1:16250`.
   - **Server Port:** The value of the **Server Socket Port** field for the Inbound Refinery instance as specified in Configuring Inbound Refinery Settings; for example, 5555. This is the `IntradocServerPort` value in the Inbound Refinery`config.cfg` file.

- **Instance Name:** The server instance name for Inbound Refinery as specified in Configuring Inbound Refinery Settings. This is the `IDC_Name` value in the Inbound Refinery `config.cfg` file.
- **Relative Web Root:** The web root of the Inbound Refinery instance: `/ibr/`

5. Under Conversion Options, check **Handles Inbound Refinery Conversion Jobs**.

   Do not check **Inbound Refinery Read Only Mode**.

6. Click **Add**.
7. Restart the Inbound Refinery Managed Server and Oracle WebCenter Content Server (WebCenter Content Managed Server), using the WebLogic Server Administration Console.
8. Go back to the Providers page, and check that the **Connection State** value is `good` for the provider.

   If the value is not `good`, double-check that you entered all the preceding entries correctly, and check that the Content Server and Inbound Refinery instances can ping each other.

9. Complete steps 1 through 8 for the second IBR server.

For more information about setting up providers, see "Configuring Content Server and Refinery Communication" in *Oracle Fusion Middleware Managing Oracle WebCenter Content*.

Enabling Components for Inbound Refinery on Content Server

Some conversion types require *helper* components to be enabled on Content Server. The InboundRefinerySupport component must always be enabled on any Content Server instance that uses Inbound Refinery for document conversion. It is enabled by default on a new Content Server installation.

To enable Inbound Refinery components on Content Server:

1. Log in to Content Server at the following URL:

   ```
   https://wcc.example.com/cs
   ```

2. From the **Administration** tray or menu, choose **Admin Server**, then **Component Manager**.
3. On the Component Manager page, select **Inbound Refinery**, then select components that you want to enable under Inbound Refinery, such as **XMLConverterSupport**, and then click **Update**.
4. Restart both Content Servers by restarting the WebCenter Content Managed Servers, using the WebLogic Server Administration Console.

Selecting File Formats To Be Converted

To tell Content Server which files to send to Inbound Refinery to be converted, you need to select file formats.

To select file formats to be converted:

1. Log in to Content Server at the following URL:

   ```
   https://wcc.example.com/cs/
   ```

2. Open the **Administration** tray or menu, then choose **Refinery Administration**, and then **File Formats Wizard** to open the File Formats Wizard page.

   This page specifies what file formats will be sent to Inbound Refinery for conversion when they are checked into Content Server.

3. Select the formats you want converted, such as **doc**, **dot**, **docx**, and **dotx** for Microsoft Word documents.

4. Click **Update**.

You can also select file formats with the Configuration Manager, with more fine-grained control, including file formats that the wizard does not list. See Managing File Types in *Oracle Fusion Middleware Managing Oracle WebCenter Content*.

Validating the Configuration of the Inbound Refinery Managed Servers

To ensure that the Inbound Refinery Managed Servers you have created are properly configured, validate the configuration by logging in to Content Server and verifying that a file with an extension recognized as valid for conversion is correctly converted.

For example, if you selected `docx` as a format to be converted, you can convert a Microsoft Word document with a `.docx` extension to PDF format.

For information about the check-in and check-out procedures, see "Uploading Documents" and "Checking Out and Downloading Files" in *Oracle Fusion Middleware Using Oracle WebCenter Conten*t.

For information about the conversion process, see "Configuring Content Servers to Send Jobs to Refineries" in *Oracle Fusion Middleware Managing Oracle WebCenter Content*.

15

Extending the Domain to Include Capture

You need to perform certain tasks in order to extend the enterprise deployment domain with Oracle WebCenter Enterprise Capture software.

- Overview of Extending the Domain to Include Capture
  Oracle WebCenter Enterprise Capture provides organizations with a single system to capture both paper and electronic documents.
- Extending the Domain for Capture
  The instructions for extending the existing enterprise deployment domain with the Capture software are detailed in this section.
- Propagating the Domain Configuration to WLS_CPT1 and WLS_CPT2
  You need to perform the following steps in order to propagate the domain configuration to the Capture Managed Servers.
- Configuring Oracle HTTP Server for the Capture Cluster
  To enable Oracle HTTP Server to route to CPT_Cluster, which contains the WLS_CPT1 and WLS_CPT2 Managed Servers, you must set the `WebLogicCluster` parameter to the list of nodes in the cluster.

Overview of Extending the Domain to Include Capture

Oracle WebCenter Enterprise Capture provides organizations with a single system to capture both paper and electronic documents.

Capture supports both centralized and distributed image capture from a user-friendly web interface capable of using high-volume, production-level scanners. Support for the industry-standard TWAIN scanning interface enables Capture to use a wide variety of industry-leading document imaging scanners to digitize paper content. Existing electronic document files can be easily captured by users or automatically captured through an importing process that can monitor an email server or network folder. Once captured, documents are organized and indexed by applying metadata through manual or automated processes that use bar code recognition technology. After documents are completed, they are committed into a content management system. Capture is fully integrated with Oracle WebCenter Content to provide organizations with one system to capture, store, manage and retrieve their mission critical business content.

Extending the Domain for Capture

The instructions for extending the existing enterprise deployment domain with the Capture software are detailed in this section.

Extending the domain involves the following tasks.

- Starting the Configuration Wizard
  Start the Configuration Wizard as the first step to extend the existing enterprise deployment domain.

- Navigating the Configuration Wizard Screens to Extend the Domain

Starting the Configuration Wizard

Start the Configuration Wizard as the first step to extend the existing enterprise deployment domain.

> **Note:**
>
> If you added any customizations directly to the start scripts in the domain, those are overwritten by the configuration wizard. To customize server startup parameters that apply to all servers in a domain, you can create a file called `setUserOverridesLate.sh` and configure it, for example, add custom libraries to the WebLogic Server classpath, specify Additional JAVA command line options for running the servers, or specify additional environment variables. Any customizations you add to this file are preserved during domain upgrade operations, and are carried over to remote servers when using the **`pack`** and **`unpack`** commands.

To start the Configuration Wizard:

1. From the WebLogic Server Console, stop any managed servers that are modified by this domain extension. Managed Servers that are not effected can remain on-line.
2. For any managed servers to be modified, verify that the managed server shutdown has completed.
3. Stop the Administration Server once all managed servers are in a steady state.
4. Navigate to the following directory and start the WebLogic Server Configuration Wizard.

   ```
   cd ORACLE_HOME/oracle_common/common/bin
   ./config.sh
   ```

Navigating the Configuration Wizard Screens to Extend the Domain

Follow the instructions in the following sections to create and configure the domain for the topology with static and dynamic clusters.

- Extending the Domain with Static Clusters
- Extending the Domain with Dynamic Clusters

Extending the Domain with Static Clusters

Follow the instructions in this section to create and configure the domain for the topology with static clusters.

> **Note:**
>
> You can use the same procedure described in this section to extend an existing domain with static clusters. If your needs do not match the instructions given in the procedure, be sure to make your selections accordingly, or refer to the supporting documentation for additional details.

Domain creation and configuration includes the following tasks:

- Task 1, Selecting the Domain Type and Domain Home Location
- Task 2, Selecting the Configuration Template
- Task 3, Specifying the Database Configuration Type
- Task 4, Specifying JDBC Component Schema Information
- Task 5, Providing the GridLink Oracle RAC Database Connection Details
- Task 6, Testing the JDBC Connections
- Task 7, Selecting Advanced Configuration
- Task 8, Configuring Managed Servers
- Task 9, Configuring a Cluster
- Task 10, Assigning Server Templates
- Task 11, Configuring Dynamic Servers
- Task 12, Assigning Managed Servers to the Cluster
- Task 13, Configuring Coherence Clusters
- Task 14, Verifying the Existing Machines
- Task 15, Assigning Servers to Machines
- Task 16, Virtual Targets
- Task 17, Partitions
- Task 18, Configuring the JMS File Store
- Task 19, Reviewing Your Configuration Specifications and Configuring the Domain
- Task 20, Writing Down Your Domain Home and Administration Server URL
- Task 21, Start the Administration Server

Task 1 Selecting the Domain Type and Domain Home Location
On the Configuration Type screen, select **Update an existing domain**.
In the **Domain Location** field, select the value of the *ASERVER_HOME* variable, which represents the complete path to the Administration Server domain home you created in Creating the Initial Infrastructure Domain for an Enterprise Deployment and while extending the domain with WebCenter Content, SOA and Inbound Refinery.
For more information about the directory location variables, see File System and Directory Variables Used in This Guide.

> **Tip:**
>
> More information about the other options on this screen can be found in Configuration Type in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 2 Selecting the Configuration Template
On the Templates screen, make sure **Update Domain Using Product Templates** is selected, then select the following templates:

- **Oracle WebCenter Enterprise Capture - 12.2.1.2.0 [wccapture]**

  In addition, the following additional templates should already be selected, because they were used to create the initial domain. These templates are not required to run Capture. They are already selected as part of the enterprise deployment configuration.

  - **Oracle Universal Content Management - Content Server - 12.2.1.2.0 [wccontent]**
  - **Oracle SOA Suite - 12.2.1.2.0 [soa]**
  - **Oracle Universal Content Management - Inbound Refinery - 12.2.1.2.0 [wccontent]**
  - **Oracle Enterprise Manager - 12.2.1.2.0 [em]**
  - **Oracle WSM Policy Manager - 12.2.1.2.0 [oracle_common]**
  - **Oracle JRF - 12.2.1.2.0 [oracle_common]**
  - **WebLogic Coherence Cluster Extension - 12.2.1.2.0 [wlserver]**

> **Tip:**
>
> More information about the options on this screen can be found in Templates in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 3 Specifying the Database Configuration Type
On the Database Configuration Type screen, select **RCU Data**.
All fields are pre-populated, because you already configured the domain to reference the Fusion Middleware schemas that are required for the Infrastructure domain. Verify and ensure that credentials in all the fields are the same that you have provided while configuring Oracle Fusion Middleware Infrastructure.
Click **Get RCU Configuration** after you finish verifying the database connection information. The following output in the Connection Result Log indicates that the operating succeeded:

```
Connecting to the database server...OK
Retrieving schema data from database server...OK
Binding local schema components with retrieved data...OK

Successfully Done.
```

> **Tip:**
>
> For more information about the **RCU Data** option, see Understanding the Service Table Schema in *Creating Schemas with the Repository Creation Utility*.
> For more information about the other options on this screen, see Datasource Defaults in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 4 Specifying JDBC Component Schema Information
Click **Convert to GridLink** and click **Next**.

Task 5 Providing the GridLink Oracle RAC Database Connection Details
On the GridLink Oracle RAC Component Schema screen, perform the following tasks.

1. Select the **SCAN** check box.
2. In the **Host Name** field, enter the Single Client Access Name (SCAN) Address for the Oracle RAC database.
3. In the **SCAN Port** field, enter the SCAN listening port for the database (for example, 1521).
4. In the **ONS Host** field, enter the SCAN address for the Oracle RAC database.
5. In the **ONS Port** field, enter the ONS Remote port (for example, 6200).

Click **Next**.

Task 6 Testing the JDBC Connections
Click **Next** to continue.

Task 7 Selecting Advanced Configuration
To complete domain configuration for the topology, select the following options on the Advanced Configuration screen:

- **Topology**
- **File Store**

Task 8 Configuring Managed Servers
On the Managed Servers screen, a new Managed Server appears in the list of servers.
Perform the following tasks to modify the default Managed Server and create a second Managed Server:

1. Rename the default Managed Server to `WLS_CPT1`.
2. Click **Add** to create a new Managed Server and name it `WLS_CPT2`.

> **Tip:**
>
> The server names recommended here will be used throughout this document; if you choose different names, be sure to replace them as needed.

3. Use the information in the following table to fill in the rest of the columns for each Capture Managed Server.

> **Tip:**
>
> More information about the options on the Managed Server screen can be found in Managed Servers in *Creating WebLogic Domains Using the Configuration Wizard*.

Server Name	Listen Address	Listen Port	Enable SSL	SSL Listen Port	Server Groups
WLS_CPT1	WCCHOST1	16400	No	Disabled	JRF-MAN-SVR CAPTURE-MGD-SVR
WLS_CPT2	WCCHOST2	16400	No	Disabled	JRF-MAN-SVR CAPTURE-MGD-SVR

Task 9 Configuring a Cluster
In this task, you create a cluster of Managed Servers to which you can target the Capture software.
You will also set the **Frontend Host** property for the cluster, which ensures that, when necessary, WebLogic Server will redirect Web services callbacks and other redirects to `wcc.example.com` on the load balancer rather than the address in the HOST header of each request.
For more information about the `wcc.example.com` virtual server address, see Configuring Virtual Hosts on the Hardware Load Balancer.
Use the Clusters screen to create a new cluster:

1. Click the **Add** button.
2. Specify `CPT_Cluster` in the **Cluster Name** field.
3. From the **Dynamic Server Groups** drop-down list, select `Unspecified`.

> **Note:**
>
> By default, server instances in a cluster communicate with one another using unicast. If you want to change your cluster communications to use multicast, refer to "Considerations for Choosing Unicast or Multicast" in *Administering Clusters for Oracle WebLogic Server*.

> **Tip:**
>
> More information about the options on this screen can be found in Clusters in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 10 Assigning Server Templates
Click **Next** to proceed to the next screen.

Task 11 Configuring Dynamic Servers
Verify that all dynamic server options are disabled for clusters that are to remain as static clusters.

1. Confirm that the **Dynamic Cluster**, **Calculated Listen Port**, and **Calculated Machine Names** checkboxes on this screen are unchecked.
2. Confirm the **Server Template** selection is **Unspecified**.
3. Click **Next**.

Task 12 Assigning Managed Servers to the Cluster
Use the Assign Servers to Clusters screen to assign `WLS_CPT1` and `WLS_CPT2` to the new cluster, `CPT_Cluster`:

1. In the Clusters pane, select the cluster to which you want to assign the servers; in this case, `CPT_Cluster`.
2. In the Servers pane, assign `WLS_CPT1` to `CPT_Cluster` by doing one of the following:
   - Click the `WLS_CPT1` Managed Server once to select it, then click on the right arrow to move it beneath the selected cluster in the Clusters pane.
   - Double-click `WLS_CPT1` to move it beneath the selected cluster in the clusters pane.
3. Repeat to assign `WLS_CPT2` to `CPT_Cluster`.

> **Tip:**
>
> More information about the options on this screen can be found in Assign Servers to Clusters in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 13 Configuring Coherence Clusters
Use the Coherence Clusters screen to configure the Coherence cluster that is automatically added to the domain. Leave the port number value at `0`, as it was defined during the initial Infrastructure domain creation. Oracle Enterprise Capture does not use Coherence Clusters. This step is a part of the domain extension process.

> **Note:**
>
> For Coherence licensing information, refer to *Oracle Coherence* in *Oracle Fusion Middleware Licensing Information*.

Task 14 Verifying the Existing Machines
Under the **Unix Machine** tab, verify the names of the machines you created when creating the initial Infrastructure domain.
Click **Next** to proceed.

Task 15 Assigning Servers to Machines
Use the Assign Servers to Machines screen to assign the Capture Managed Servers you just created to the corresponding machines in the domain.

Assign `WLS_CPT1` to `WCCHOST1`, and assign `WLS_CPT2` to `WCCHOST2`.

> **Tip:**
>
> More information about the options on this screen can be found in Assign Servers to Machines in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 16 Virtual Targets
Click **Next** to proceed to the next screen.

Task 17 Partitions
Click **Next** to proceed to the next screen.

Task 18 Configuring the JMS File Store
In the JMS File Stores screen, assign the following directory for each of the WCC Persistence stores including UMS and BPM file stores:

```
ORACLE_RUNTIME/CPT_Cluster/jms
```

In this example, replace *ASERVER_HOME* with the value of the variable for you environment. Replace *CPT_Cluster* with the name you assigned to the Capture cluster.

Task 19 Reviewing Your Configuration Specifications and Configuring the Domain
The Configuration Summary screen contains the detailed configuration information for the domain you are about to create. Review the details of each item on the screen and verify that the information is correct.
You can go back to any previous screen if you need to make any changes, either by using the **Back** button or by selecting the screen in the navigation pane.
Domain creation will not begin until you click **Update**.

> **Tip:**
>
> More information about the options on this screen can be found in Configuration Summary in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 20 Writing Down Your Domain Home and Administration Server URL
The Configuration Success screen will show the following items about the domain you just configured:

- Domain Location
- Administration Server URL

You must make a note of both items as you will need them later; the domain location is needed to access the scripts used to start the Node Manager and Administration Server, and the URL is needed to access the Administration Server.
Click **Finish** to dismiss the Configuration Wizard.

Task 21 Start the Administration Server
Start the Administration Server to ensure the changes you have made to the domain have been applied.

After you have completed extending the domain with static clusters, go to Propagating the Domain Configuration to WLS_CPT1 and WLS_CPT2.

Extending the Domain with Dynamic Clusters

Follow the instructions in this section to create and configure the domain for the topology with dynamic clusters.

> **Note:**
>
> You can use the same procedure described in this section to extend an existing domain with Dynamic Clusters. If your needs do not match the instructions given in the procedure, be sure to make your selections accordingly, or refer to the supporting documentation for additional details.

Domain creation and configuration includes the following tasks:

- Task 1, Selecting the Domain Type and Domain Home Location
- Task 2, Selecting the Configuration Template
- Task 3, Specifying the Database Configuration Type
- Task 4, Specifying JDBC Component Schema Information
- Task 5, Providing the GridLink Oracle RAC Database Connection Details
- Task 6, Testing the JDBC Connections
- Task 7, Selecting Advanced Configuration
- Task 8, Configuring Managed Servers
- Task 9, Configuring a Cluster
- Task 10, Assigning Server Templates
- Task 11, Configuring Dynamic Servers
- Task 12, Assigning Managed Servers to the Cluster
- Task 13, Configuring Coherence Clusters
- Task 14, Verifying the Existing Machines
- Task 15, Assigning Servers to Machines
- Task 16, Virtual Targets
- Task 17, Partitions
- Task 18, Configuring the File Stores
- Task 19, Reviewing Your Configuration Specifications and Configuring the Domain
- Task 20, Writing Down Your Domain Home and Administration Server URL
- Task 21, Start the Administration Server

Task 1 Selecting the Domain Type and Domain Home Location
On the Configuration Type screen, select **Update an existing domain**.
In the **Domain Location** field, select the value of the *ASERVER_HOME* variable, which represents the complete path to the Administration Server domain home you created in Creating the Initial Infrastructure Domain for an Enterprise Deployment and while extending the domain with WebCenter Content, SOA and Inbound Refinery.
For more information about the directory location variables, see File System and Directory Variables Used in This Guide.

> **Tip:**
>
> More information about the other options on this screen can be found in Configuration Type in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 2 Selecting the Configuration Template
On the Templates screen, make sure **Update Domain Using Product Templates** is selected, then select the following templates:

- **Oracle WebCenter Enterprise Capture - 12.2.1.3.0 [wccapture]**

  In addition, the following additional templates should already be selected, because they were used to create the initial domain. These templates are not required to run Capture. They are already selected as part of the enterprise deployment configuration.

  - **Oracle Universal Content Mnagement - Content Server - 12.2.1.3.0 [wccontent]**
  - **Oracle SOA Suite - 12.2.1.3.0 [soa]**
  - **Oracle Universal Content Management - Inbound Refinery - 12.2.1.3.0 [wccontent]**
  - **Oracle Enterprise Manager - 12.2.1.3.0 [em]**
  - **Oracle WSM Policy Manager - 12.2.1.3.0 [oracle_common]**
  - **Oracle JRF - 12.2.1.3.0 [oracle_common]**
  - **WebLogic Coherence Cluster Extension - 12.2.1.3.0 [wlserver]**

> **Tip:**
>
> More information about the options on this screen can be found in Templates in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 3 Specifying the Database Configuration Type
On the Database Configuration Type screen, select **RCU Data**.
All fields are pre-populated, because you already configured the domain to reference the Fusion Middleware schemas that are required for the Infrastructure domain.
Verify and ensure that credentials in all the fields are the same that you have provided while configuring Oracle Fusion Middleware Infrastructure.
Click **Get RCU Configuration** after you finish verifying the database connection information. The following output in the Connection Result Log indicates that the operating succeeded:

```
Connecting to the database server...OK
Retrieving schema data from database server...OK
Binding local schema components with retrieved data...OK

Successfully Done.
```

> **Tip:**
>
> For more information about the **RCU Data** option, see "Understanding the Service Table Schema" in *Creating Schemas with the Repository Creation Utility*.
> For more information about the other options on this screen, see "Datasource Defaults" in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 4 Specifying JDBC Component Schema Information
Click **Convert to GridLink** and click **Next**.

Task 5 Providing the GridLink Oracle RAC Database Connection Details
On the GridLink Oracle RAC Component Schema screen, perform the following tasks.

1. Select the **SCAN** check box.
2. In the **Host Name** field, enter the Single Client Access Name (SCAN) Address for the Oracle RAC database.
3. In the **SCAN Port** field, enter the SCAN listening port for the database (for example, 1521).
4. In the **ONS Host** field, enter the SCAN address for the Oracle RAC database.
5. In the **ONS Port** field, enter the ONS Remote port (for example, 6200).

Click **Next**.

Task 6 Testing the JDBC Connections
Click **Next** to continue.

Task 7 Selecting Advanced Configuration
To complete domain configuration for the topology, select the following options on the Advanced Configuration screen:

- **Topology**
- **File Store**

Task 8 Configuring Managed Servers
On the Managed Servers screen, a new Managed Server appears in the list of servers.
Perform the following tasks to modify the default Managed Server and create a second Managed Server:

1. Delete the default Managed Server.
2. Click **Next** to proceed to the next screen.

Task 9 Configuring a Cluster
In this task, you create a cluster of Managed Servers to which you can target the Capture software.

You will also set the **Frontend Host** property for the cluster, which ensures that, when necessary, WebLogic Server will redirect Web services callbacks and other redirects to `wcc.example.com` on the load balancer rather than the address in the HOST header of each request.
For more information about the `wcc.example.com` virtual server address, see Configuring Virtual Hosts on the Hardware Load Balancer.
Use the Clusters screen to create a new cluster:

1. Click the **Add** button.
2. Specify `CPT_Cluster` in the **Cluster Name** field.
3. From the **Dynamic Server Groups** drop-down list, select `ODDC-DYN-CLUSTER`.

> **Note:**
>
> By default, server instances in a cluster communicate with one another using unicast. If you want to change your cluster communications to use multicast, refer to "Considerations for Choosing Unicast or Multicast" in *Administering Clusters for Oracle WebLogic Server*.

> **Tip:**
>
> More information about the options on this screen can be found in Clusters in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 10 Assigning Server Templates
Use the Server Templates screen to add a template:

1. Specify `CPT-server-template` in the **Name** field.
2. Specify `16400` in the **Listen Port** field.
3. Leave the **Enable SSL** option unchecked.
4. From the **Cluster** drop-down list, select `CPT_Cluster`.
5. Click **Next** to proceed to the next screen.

Task 11 Configuring Dynamic Servers
Use the Dynamic Clusters screen to configure the required clusters:

1. Specify `CPT_Cluster` in the **Cluster Name** field.
2. Specify `WLS_CPT` in the **Server Name Prefix** field.
3. From the **Server Template** drop-down list, select `CPT-server-template`.
4. Specify `2` in the **Maximum Dynamic Server Count** field.
5. Specify `WCCHOST*` in the **Machine Name Match Expression** field.
6. Select the **Calculated Machine Names** and **Dynamic Cluster** fields.
7. Select the **Calculated Listen Ports** field.
8. Click **Next** to proceed to the next screen.

Task 12 Assigning Managed Servers to the Cluster
Click **Next** to proceed to the next screen.

Task 13 Configuring Coherence Clusters
Use the Coherence Clusters screen to configure the Coherence cluster that is automatically added to the domain. Leave the port number value at `0`, as it was defined during the initial Infrastructure domain creation. Oracle Enterprise Capture does not use Coherence Clusters. This step is a part of the domain extension process.

> **Note:**
>
> For Coherence licensing information, refer to *Oracle Coherence* in *Oracle Fusion Middleware Licensing Information*.

Task 14 Verifying the Existing Machines
Under the **Unix Machine** tab, verify the names of the machines you created when creating the initial Infrastructure domain.
Click **Next** to proceed to the next screen.

Task 15 Assigning Servers to Machines
Click **Next** to proceed to the next screen.

Task 16 Virtual Targets
Click **Next** to proceed to the next screen.

Task 17 Partitions
Click **Next** to proceed to the next screen.

Task 18 Configuring the File Stores
In the JMS File Stores screen, assign the following directory for each of the WCC Persistence stores including UMS and BPM file stores:

```
ORACLE_RUNTIME/CPT_Cluster/jms
```

In this example, replace *ASERVER_HOME* with the value of the variable for you environment. Replace *CPT_Cluster* with the name you assigned to the Capture cluster.

Task 19 Reviewing Your Configuration Specifications and Configuring the Domain
The Configuration Summary screen contains the detailed configuration information for the domain you are about to create. Review the details of each item on the screen and verify that the information is correct.
You can go back to any previous screen if you need to make any changes, either by using the **Back** button or by selecting the screen in the navigation pane.
Domain creation will not begin until you click **Update**.

> **Tip:**
>
> More information about the options on this screen can be found in Configuration Summary in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 20 Writing Down Your Domain Home and Administration Server URL
The Configuration Success screen will show the following items about the domain you just configured:

- Domain Location
- Administration Server URL

You must make a note of both items as you will need them later; the domain location is needed to access the scripts used to start the Node Manager and Administration Server, and the URL is needed to access the Administration Server.
Click **Finish** to dismiss the Configuration Wizard.

Task 21 Start the Administration Server
Start the Administration Server to ensure the changes you have made to the domain have been applied.

Propagating the Domain Configuration to WLS_CPT1 and WLS_CPT2

You need to perform the following steps in order to propagate the domain configuration to the Capture Managed Servers.

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. Run the following `pack` command on WCCHOST1 to create a template pack:

```
cd ORACLE_COMMON_HOME/common/bin

./pack.sh -managed=true -domain=ASERVER_HOME -
template=edgdomaintemplateExtCPT.jar -template_name=edgdomain_templateCPT
```

3. Run the following `unpack` command on WCCHOST1 to propagate the template created in the preceding step to the WLS_CPT1 domain directory:

```
cd ORACLE_COMMON_HOME/common/bin

./unpack.sh -domain=MSERVER_HOME -template=edgdomaintemplateExtCPT.jar -
app_dir=APPLICATION_HOME -overwrite_domain=true
```

4. Run the following command on WCCHOST1 to copy the template pack created in step 1 to WCCHOST2:

```
scp edgdomaintemplateCPT.jar oracle@WCCHOST2:ORACLE_COMMON_HOME/common/bin
```

5. Run the `unpack` command on WCCHOST2 to unpack the propagated template to the WLS_CPT2 domain directory.

```
cd ORACLE_COMMON_HOME/common/bin

./unpack.sh -domain=MSERVER_HOME -template=edgdomaintemplateExtCPT.jar -
app_dir=APPLICATION_HOME -overwrite_domain=true
```

6. Restart the Administration Server to make these changes take effect, stopping it with the `nmKill` command, or with the Administration Console, and then starting it with the `nmStart` command. Before the restart, stop all Managed Servers in the domain through the Administration Console, and then start them after the restart. Log in to the Administration Console using the credentials for the `weblogic` user.

Configuring Oracle HTTP Server for the Capture Cluster

To enable Oracle HTTP Server to route to CPT_Cluster, which contains the WLS_CPT1 and WLS_CPT2 Managed Servers, you must set the `WebLogicCluster` parameter to the list of nodes in the cluster.

This section includes the following topics.

Configuring Oracle HTTP Server for the WLS_CPT Managed Servers

To configure Oracle HTTP Server for the WLS_CPT Managed Servers:

1. For each of the web servers on WEBHOST1 and WEBHOST2, add the following lines to the `ORACLE_INSTANCE/config/OHS/ohs1/moduleconf/wcc_vh.conf` and `ORACLE_INSTANCE/config/OHS/ohs2/moduleconf/wcc_vh.conf` files:

```
#DC-Console
<Location /dc-console>
   WebLogicCluster WCCHOST1:16400,WCCHOST2:16400
   WLSRequest ON
   WLCookieName JSESSIONID
   WLProxySSL ON
   WLProxySSLPassThrough ON
</Location>

#DC-Client
<Location /dc-client>
   WebLogicCluster WCCHOST1:16400,WCCHOST2VHN316400
   WLSRequest ON
   WLCookieName JSESSIONID
   WLProxySSL ON
   WLProxySSLPassThrough ON
</Location>
```

2. Restart Oracle HTTP Server on both WEBHOST1 and WEBHOST2:

```
./stopComponent.sh ohsn
./startComponent.sh ohsn
```

For WEBHOST1, use `ohs1` (where n=1) and for WEBHOST2, use `ohs2` (where n=2).

Setting the Front-End HTTP Host and Port for the Capture Cluster

To set the front-end HTTP host and port for the Capture cluster:

1. Log in to the WebLogic Server Administration Console.
2. Go to the Change Center section and click **Lock & Edit**.
3. Expand the **Environment** node in the **Domain Structure** tree on the left.
4. Click **Clusters**.
5. On the Summary of Clusters page, select **CPT_Cluster**.
6. Open the **HTTP** tab.
7. Set the following values:
   - **Frontend Host:** `wcc.example.com`
   - **Frontend HTTPS Port:** `443`
   - **Frontend HTTP Port:** `80`
8. Click **Save**.
9. Click **Activate Changes** in the Change Center section of the Administration Console.
10. Restart the servers to make the front-end host directive in the cluster take effect.

Validating Access Through the Load Balancer

Verify URLs to ensure that appropriate routing and failover is working from the HTTP Server to CPT_Cluster. To verify the URLs:

1. While WLS_CPT2 is running, stop WLS_CPT1 from the WebLogic Server Administration Console.
2. Access `https://wcc.example.com/dc-console` to verify that it is functioning properly. (You will not be able to retrieve reports or data because the Capture server is down.)
3. Start WLS_CPT1 from the WebLogic Server Administration Console.
4. Stop WLS_CPT2 from the WebLogic Server Administration Console.
5. Access `https://wcc.example.com/dc-console` to verify that it is functioning properly.
6. Start WLS_CPT2 from the WebLogic Server Administration Console.

16

Extending the Domain to Include Imaging

You need to perform certain tasks in order to extend the enterprise deployment domain with Oracle WebCenter Enterprise Imaging software.

- Overview of Extending the Domain to Include Imaging
  Oracle WebCenter Enterprise Imaging provides organizations with a single system to capture both paper and electronic documents.
- Extending the Domain for Imaging
  The instructions for extending the existing enterprise deployment domain with the Imaging software are detailed in this section.
- Navigating the Configuration Wizard Screens to Extend the Domain
  Follow the instructions in the following sections to create and configure the domain for the topology with static and dynamic clusters.
- Propagating the Domain Configuration to WLS_IPM1 and WLS_IPM2
  You need to perform the following steps in order to propagate the domain configuration to the Imaging Managed Servers.
- Configuring Oracle HTTP Server for the Imaging Cluster
  To enable Oracle HTTP Server to route to `IPM_Cluster`, which contains the `WLS_IPM1` and `WLS_IPM2` Managed Servers, you must set the `WebLogicCluster` parameter to the list of nodes in the cluster.

Overview of Extending the Domain to Include Imaging

Oracle WebCenter Enterprise Imaging provides organizations with a single system to capture both paper and electronic documents.

Oracle WebCenter Content: Imaging provides organizations with a scalable solution upon which to develop process-oriented imaging applications and image-enablement solutions for enterprise applications. It enables image capture through Oracle WebCenter Capture, annotation and markup of images, routing and approval automation, and support for high-volume applications for billions of items. With Imaging, organizations can quickly integrate their content and processes directly with Oracle enterprise applications, such as Oracle E-Business Suite, PeopleSoft Enterprise, and JD Edwards EnterpriseOne. Users bene fit by having a single source for all transaction-based content, eliminating the need for double entry.

Extending the Domain for Imaging

The instructions for extending the existing enterprise deployment domain with the Imaging software are detailed in this section.

Extending the domain involves the following tasks.

Navigating the Configuration Wizard Screens to Extend the Domain

Follow the instructions in the following sections to create and configure the domain for the topology with static and dynamic clusters.

- Extending the Domain with Static Clusters
- Extending the Domain with Dynamic Clusters

Extending the Domain with Static Clusters

Follow the instructions in this section to create and configure the domain for the topology with static clusters.

> **Note:**
>
> You can use the same procedure described in this section to extend an existing domain with static clusters. If your needs do not match the instructions given in the procedure, be sure to make your selections accordingly, or refer to the supporting documentation for additional details.

Domain creation and configuration includes the following tasks:

- Task 1, Selecting the Domain Type and Domain Home Location
- Task 2, Selecting the Configuration Template
- Task 3, Specifying the Database Configuration Type
- Task 4, Specifying JDBC Component Schema Information
- Task 5, Providing the GridLink Oracle RAC Database Connection Details
- Task 6, Testing the JDBC Connections
- Task 7, Selecting Advanced Configuration
- Task 8, Configuring Managed Servers
- Task 9, Configuring a Cluster
- Task 10, Assigning Server Templates
- Task 11, Configuring Dynamic Servers
- Task 12, Assigning Managed Servers to the Cluster
- Task 13, Configuring Coherence Clusters
- Task 14, Verifying the Existing Machines
- Task 15, Assigning Servers to Machines
- Task 16, Virtual Targets
- Task 17, Partitions

- Task 18, Configuring the JMS File Store
- Task 19, Reviewing Your Configuration Specifications and Configuring the Domain
- Task 20, Writing Down Your Domain Home and Administration Server URL
- Task 21, Start the Administration Server

Task 1 Selecting the Domain Type and Domain Home Location
On the Configuration Type screen, select **Update an existing domain**.
In the **Domain Location** field, select the value of the *ASERVER_HOME* variable, which represents the complete path to the Administration Server domain home you created in Creating the Initial Infrastructure Domain for an Enterprise Deployment and while extending the domain with WebCenter Content, SOA, and Inbound Refinery.
For more information about the directory location variables, see File System and Directory Variables Used in This Guide.

> **Tip:**
>
> For more information about the other options on this screen can be found in Configuration Type in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 2 Selecting the Configuration Template
On the Templates screen, make sure **Update Domain Using Product Templates** is selected, then select the following templates:

- **Oracle WebCenter Content: Imaging - 12.2.1.3.0 [wccontent]**
- In addition, the following additional templates should already be selected, because they were used to create the initial domain. These templates are not required to run Imaging. They are already selected as part of the enterprise deployment configuration.
  - **Oracle Universal Content Management - Content Server - 12.2.1.3.0 [wccontent]**
  - **Oracle SOA Suite - 12.2.1.3.0 [soa]**
  - **Oracle Universal Content Management - Inbound Refinery - 12.2.1.3.0 [wccontent]**
  - **Oracle Enterprise Manager - 12.2.1.3.0 [em]**
  - **Oracle WSM Policy Manager - 12.2.1.3.0 [oracle_common]**
  - **Oracle JRF - 12.2.1.3.0 [oracle_common]**
  - **WebLogic Coherence Cluster Extension - 12.2.1.3.0 [wlserver]**

> **Tip:**
>
> For more information about the options on this screen can be found in Templates in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 3 Providing the GridLink DS Oracle RAC Database Connection Details
On the GridLink Oracle RAC Component Schema screen, perform the following tasks.

1. Select the **SCAN** check box.
2. In the **Host Name** field, enter the Single Client Access Name (SCAN) Address for the Oracle RAC database.
3. In the **SCAN Port** field, enter the SCAN listening port for the database (for example, 1521).
4. In the **ONS Host** field, enter the SCAN address for the Oracle RAC database.
5. In the **ONS Port** field, enter the ONS Remote port (for example, 6200).
6. Click **Next**.

Task 4 Testing the JDBC DS Connections
Click **Next** to continue.

Task 5 Specifying the Database Configuration Type
On the Database Configuration Type screen, select **RCU Data**.
All fields are pre-populated, because you already configured the domain to reference the Fusion Middleware schemas that are required for the imaging domain.
Verify and ensure that credentials in all the fields are the same that you have provided while configuring Oracle Fusion Middleware Infrastructure.
Click **Get RCU Configuration** after you finish verifying the database connection information. The following output in the Connection Result Log indicates that the operating succeeded:

```
Connecting to the database server...OK
Retrieving schema data from database server...OK
Binding local schema components with retrieved data...OK

Successfully Done.
```

> **Tip:**
>
> For more information about the **RCU Data** option, see Understanding the Service Table Schema in *Creating Schemas with the Repository Creation Utility*.
> For more information about the other options on this screen, see Datasource Defaults in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 6 Specifying JDBC Component Schema Information
Click **Convert to GridLink** and click **Next**.

Task 7 Providing the GridLink Oracle RAC Database Connection Details
On the GridLink Oracle RAC Component Schema screen, perform the following tasks.

1. Select the **SCAN** check box.
2. In the **Host Name** field, enter the Single Client Access Name (SCAN) Address for the Oracle RAC database.
3. In the **SCAN Port** field, enter the SCAN listening port for the database (for example, 1521).
4. In the **ONS Host** field, enter the SCAN address for the Oracle RAC database.

5. In the **ONS Port** field, enter the ONS Remote port (for example, 6200).
6. Click **Next**.

Task 8 Testing the JDBC Connections
Click **Next** to continue.

Task 9 Selecting Advanced Configuration
To complete domain configuration for the topology, select the following options on the Advanced Configuration screen:

- **Topology**
- **File Store**

Task 10 Configuring Managed Servers
On the Managed Servers screen, a new Managed Server appears in the list of servers.
Perform the following tasks to modify the default Managed Server and create a second Managed Server:

1. Rename the default Managed Server to `WLS_IPM1`.
2. Click **Add** to create a new Managed Server and name it `WLS_IPM2`.

> **Tip:**
>
> The server names recommended here will be used throughout this document. If you choose different names, be sure to replace them as needed.

3. Use the information in the following table to fill in the rest of the columns for each Imaging Managed Server.

> **Tip:**
>
> For more information about the options on the Managed Server screen can be found in Managed Servers in *Creating WebLogic Domains Using the Configuration Wizard*.

Server Name	Listen Address	Listen Port	Enable SSL	SSL Listen Port	Server Groups
WLS_IPM1	WCCHOST1	16000	No	Disabled	IPM-MGD-SVR
WLS_IPM2	WCCHOST2	16000	No	Disabled	IPM-MGD-SVR

Task 11 Configuring a Cluster
In this task, you create a cluster of Managed Servers to which you can target the Imaging software.
You will also set the **Frontend Host** property for the cluster, which ensures that, when necessary, WebLogic Server will redirect Web services callbacks and other redirects to `wcc.example.com` on the load balancer rather than the address in the HOST header of each request.
For more information about the `wcc.example.com` virtual server address, see Configuring Virtual Hosts on the Hardware Load Balancer.

Use the Clusters screen to create a new cluster:

1. Click the **Add** button.
2. Specify `IPM_Cluster` in the **Cluster Name** field.
3. From the **Dynamic Server Groups** drop-down list, select `Unspecified`.

> **Note:**
>
> By default, server instances in a cluster communicate with one another using unicast. If you want to change your cluster communications to use multicast, refer to Considerations for Choosing Unicast or Multicast in *Administering Clusters for Oracle WebLogic Server*.

> **Tip:**
>
> For more information about the options on this screen can be found in Clusters in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 12 Assigning Server Templates
Click **Next** to proceed to the next screen.

Task 13 Configuring Dynamic Servers
Verify that all dynamic server options are disabled for clusters that are to remain as static clusters.

1. Confirm that the **Dynamic Cluster**, **Calculated Listen Port**, and **Calculated Machine Names** checkboxes on this screen are unchecked.
2. Confirm the **Server Template** selection is **Unspecified**.
3. Click **Next**.

Task 14 Assigning Managed Servers to the Cluster
Use the Assign Servers to Clusters screen to assign `WLS_IPM1` and `WLS_IPM2` to the new cluster, `IPM_Cluster`:

1. In the Clusters pane, select the cluster to which you want to assign the servers; in this case, `IPM_Cluster`.
2. In the Servers pane, assign `WLS_IPM1` to `IPM_Cluster` by doing one of the following:
   - Click the `WLS_IPM1` Managed Server once to select it, then click on the right arrow to move it beneath the selected cluster in the Clusters pane.
   - Double-click `WLS_IPM1` to move it beneath the selected cluster in the clusters pane.
3. Repeat to assign `WLS_IPM2` to `IPM_Cluster`.

> **Tip:**
>
> For more information about the options on this screen can be found in Assign Servers to Clusters in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 15 Configuring Coherence Clusters
Use the Coherence Clusters screen to configure the Coherence cluster that is automatically added to the domain. Leave the port number value at `0`, as it was defined during the initial Infrastructure domain creation. Oracle Enterprise Imaging does not use Coherence Clusters. This step is a part of the domain extension process.

> **Note:**
>
> For Coherence licensing information, refer to *Oracle Coherence* in *Oracle Fusion Middleware Licensing Information*.

Task 16 Verifying the Existing Machines
Under the **Unix Machine** tab, verify the names of the machines you created when creating the initial Infrastructure domain.
Click **Next** to proceed.

Task 17 Assigning Servers to Machines
Use the Assign Servers to Machines screen to assign the Imaging Managed Servers you just created to the corresponding machines in the domain.
Assign `WLS_IPM1` to `WCCHOST1`, and assign `WLS_IPM2` to `WCCHOST2`.

> **Tip:**
>
> For more information about the options on this screen can be found in Assign Servers to Machines in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 18 Virtual Targets
Click **Next** to proceed to the next screen.

Task 19 Partitions
Click **Next** to proceed to the next screen.

Task 20 Configuring the JMS File Store
In the JMS File Stores screen, assign the following directory for each of the WCC Persistence stores including UMS and BPM file stores:

```
ORACLE_RUNTIME/IPM_Cluster/jms
```

In this example, replace *ASERVER_HOME* with the value of the variable for you environment. Replace *IPM_Cluster* with the name you assigned to the Imaging cluster.

Task 21 Reviewing Your Configuration Specifications and Configuring the Domain
The Configuration Summary screen contains the detailed configuration information for the domain you are about to create. Review the details of each item on the screen and verify that the information is correct.
You can go back to any previous screen if you need to make any changes, either by using the **Back** button or by selecting the screen in the navigation pane.
Domain creation will not begin until you click **Update**.

> **Tip:**
>
> More information about the options on this screen can be found in Configuration Summary in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 22 Writing Down Your Domain Home and Administration Server URL
The Configuration Success screen will show the following items about the domain you just configured:

- Domain Location
- Administration Server URL

You must make a note of both items as you will need them later; the domain location is needed to access the scripts used to start the Node Manager and Administration Server, and the URL is needed to access the Administration Server.
Click **Finish** to dismiss the Configuration Wizard.

Task 23 Start the Administration Server
Start the Administration Server to ensure the changes you have made to the domain have been applied.

After you have completed extending the domain with static clusters, go to Propagating the Domain Configuration to WLS_CPT1 and WLS_CPT2.

Extending the Domain with Dynamic Clusters

Follow the instructions in this section to create and configure the domain for the topology with dynamic clusters.

> **Note:**
>
> You can use the same procedure described in this section to extend an existing domain with Dynamic Clusters. If your needs do not match the instructions given in the procedure, be sure to make your selections accordingly, or refer to the supporting documentation for additional details.

Domain creation and configuration includes the following tasks:

- Task 1, Selecting the Domain Type and Domain Home Location
- Task 2, Selecting the Configuration Template
- Task 3, Specifying the Database Configuration Type

- Task 4, Specifying JDBC Component Schema Information
- Task 5, Providing the GridLink Oracle RAC Database Connection Details
- Task 6, Testing the JDBC Connections
- Task 7, Selecting Advanced Configuration
- Task 8, Configuring Managed Servers
- Task 9, Configuring a Cluster
- Task 10, Assigning Server Templates
- Task 11, Configuring Dynamic Servers
- Task 12, Assigning Managed Servers to the Cluster
- Task 13, Configuring Coherence Clusters
- Task 14, Verifying the Existing Machines
- Task 15, Assigning Servers to Machines
- Task 16, Virtual Targets
- Task 17, Partitions
- Task 18, Configuring the File Stores
- Task 19, Reviewing Your Configuration Specifications and Configuring the Domain
- Task 20, Writing Down Your Domain Home and Administration Server URL
- Task 21, Start the Administration Server

Task 1 Selecting the Domain Type and Domain Home Location
On the Configuration Type screen, select **Update an existing domain**.
In the **Domain Location** field, select the value of the *ASERVER_HOME* variable, which represents the complete path to the Administration Server domain home you created in Creating the Initial Infrastructure Domain for an Enterprise Deployment and while extending the domain with WebCenter Content, SOA and Inbound Refinery.
For more information about the directory location variables, see File System and Directory Variables Used in This Guide.

> **Tip:**
>
> For more information about the other options on this screen can be found in Configuration Type in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 2 Selecting the Configuration Template
On the Templates screen, make sure **Update Domain Using Product Templates** is selected, then select the following templates:

- **Oracle WebCenter Content: Imaging - 12.2.1.3.0 [wccontent]**
- In addition, the following additional templates should already be selected, because they were used to create the initial domain. These templates are not required to run Imaging. They are already selected as part of the enterprise deployment configuration.

- **Oracle Universal Content Management - Content Server - 12.2.1.3.0 [wccontent]**
- **Oracle SOA Suite - 12.2.1.3.0 [soa]**
- **Oracle Universal Content Management - Inbound Refinery - 12.2.1.3.0 [wccontent]**
- **Oracle Enterprise Manager - 12.2.1.3.0 [em]**
- **Oracle WSM Policy Manager - 12.2.1.3.0 [oracle_common]**
- **Oracle JRF - 12.2.1.3.0 [oracle_common]**
- **WebLogic Coherence Cluster Extension - 12.2.1.3.0 [wlserver]**

> **Tip:**
>
> For more information about the options on this screen can be found in Templates in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 3 Providing the GridLink DS Oracle RAC Database Connection Details
On the GridLink Oracle RAC Component Schema screen, perform the following tasks.

1. Select the **SCAN** check box.
2. In the **Host Name** field, enter the Single Client Access Name (SCAN) Address for the Oracle RAC database.
3. In the **SCAN Port** field, enter the SCAN listening port for the database (for example, 1521).
4. In the **ONS Host** field, enter the SCAN address for the Oracle RAC database.
5. In the **ONS Port** field, enter the ONS Remote port (for example, 6200).
6. Click **Next**.

Task 4 Testing the JDBC DS Connections
Click **Next** to continue.

Task 5 Specifying the Database Configuration Type
On the Database Configuration Type screen, select **RCU Data**.
All fields are pre-populated, because you already configured the domain to reference the Fusion Middleware schemas that are required for the imaging domain.
Verify and ensure that credentials in all the fields are the same that you have provided while configuring Oracle Fusion Middleware Infrastructure.
Click **Get RCU Configuration** after you finish verifying the database connection information. The following output in the Connection Result Log indicates that the operating succeeded:

```
Connecting to the database server...OK
Retrieving schema data from database server...OK
Binding local schema components with retrieved data...OK

Successfully Done.
```

> **Tip:**
>
> For more information about the **RCU Data** option, see Understanding the Service Table Schema in *Creating Schemas with the Repository Creation Utility*.
> For more information about the other options on this screen, see Datasource Defaults in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 6 Specifying JDBC Component Schema Information
Click **Convert to GridLink** and click **Next**.

Task 7 Providing the GridLink Oracle RAC Database Connection Details
On the GridLink Oracle RAC Component Schema screen, perform the following tasks.

1. Select the **SCAN** check box.
2. In the **Host Name** field, enter the Single Client Access Name (SCAN) Address for the Oracle RAC database.
3. In the **SCAN Port** field, enter the SCAN listening port for the database (for example, 1521).
4. In the **ONS Host** field, enter the SCAN address for the Oracle RAC database.
5. In the **ONS Port** field, enter the ONS Remote port (for example, 6200).
6. Click **Next**.

Task 8 Testing the JDBC Connections
Click **Next** to continue.

Task 9 Selecting Advanced Configuration
To complete domain configuration for the topology, select the following options on the Advanced Configuration screen:

- **Topology**
- **File Store**

Task 10 Configuring Managed Servers
On the Managed Servers screen, a new Managed Server appears in the list of servers.
Perform the following tasks to modify the default Managed Server and create a second Managed Server:

1. Delete the default Managed Server.
2. Click **Next** to proceed to the next screen.

Task 11 Configuring a Cluster
In this task, you create a cluster of Managed Servers to which you can target the Imaging software.
You will also set the **Frontend Host** property for the cluster, which ensures that, when necessary, WebLogic Server will redirect Web services callbacks and other redirects to `wcc.example.com` on the load balancer rather than the address in the HOST header of each request.

For more information about the `wcc.example.com` virtual server address, see Configuring Virtual Hosts on the Hardware Load Balancer.
Use the Clusters screen to create a new cluster:

1. Click the **Add** button.
2. Specify `IPM_Cluster` in the **Cluster Name** field.
3. From the **Dynamic Server Groups** drop-down list, select `IPM-DYN-CLUSTER`.

> **Note:**
>
> By default, server instances in a cluster communicate with one another using unicast. If you want to change your cluster communications to use multicast, refer to Considerations for Choosing Unicast or Multicast in *Administering Clusters for Oracle WebLogic Server*.

> **Tip:**
>
> For more information about the options on this screen can be found in Clusters in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 12 Assigning Server Templates
Use the Server Templates screen to add a template:

1. Specify `IPM-server-template` in the **Name** field.
2. Specify `7100` in the **Listen Port** field.
3. Leave the **Enable SSL** option unchecked.
4. From the **Cluster** drop-down list, select `IPM_Cluster`.
5. Click **Next** to proceed to the next screen.

Task 13 Configuring Dynamic Servers
Use the Dynamic Clusters screen to configure the required clusters:

1. Specify `IPM_Cluster` in the **Cluster Name** field.
2. Specify `WLS_IPM` in the **Server Name Prefix** field.
3. From the **Server Template** drop-down list, select `IPM-server-template`.
4. Specify `2` in the **Dynamic Cluster Size** field.
5. Specify `WCCHOST*` in the **Machine Name Match Expression** field.
6. Select the **Calculated Machine Names** and **Dynamic Cluster** fields.
7. Select the **Calculated Listen Ports** field.
8. Click **Next** to proceed to the next screen.

Task 14 Assigning Managed Servers to the Cluster
Click **Next** to proceed to the next screen.

Task 15 Configuring Coherence Clusters
Use the Coherence Clusters screen to configure the Coherence cluster that is automatically added to the domain. Leave the port number value at `0`, as it was defined during the initial Infrastructure domain creation. Oracle Enterprise Imaging does not use Coherence Clusters. This step is a part of the domain extension process.

> **Note:**
>
> For Coherence licensing information, refer to *Oracle Coherence* in *Oracle Fusion Middleware Licensing Information*.

Task 16 Verifying the Existing Machines
Under the **Unix Machine** tab, verify the names of the machines you created when creating the initial Infrastructure domain.
Click **Next** to proceed to the next screen.

Task 17 Assigning Servers to Machines
Click **Next** to proceed to the next screen.

Task 18 Virtual Targets
Click **Next** to proceed to the next screen.

Task 19 Partitions
Click **Next** to proceed to the next screen.

Task 20 Configuring the File Stores
In the JMS File Stores screen, assign the following directory for each of the WCC Persistence stores including UMS and BPM file stores:

```
ORACLE_RUNTIME/IPM_Cluster/jms
```

In this example, replace *ASERVER_HOME* with the value of the variable for you environment. Replace *IPM_Cluster* with the name you assigned to the Imaging cluster.

Task 21 Reviewing Your Configuration Specifications and Configuring the Domain
The Configuration Summary screen contains the detailed configuration information for the domain you are about to create. Review the details of each item on the screen and verify that the information is correct.
You can go back to any previous screen if you need to make any changes, either by using the **Back** button or by selecting the screen in the navigation pane.
Domain creation will not begin until you click **Update**.

> **Tip:**
>
> More information about the options on this screen can be found in Configuration Summary in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 22 Writing Down Your Domain Home and Administration Server URL
The Configuration Success screen will show the following items about the domain you just configured:

- Domain Location
- Administration Server URL

You must make a note of both items as you will need them later; the domain location is needed to access the scripts used to start the Node Manager and Administration Server, and the URL is needed to access the Administration Server.
Click **Finish** to dismiss the Configuration Wizard.

Task 23 Start the Administration Server
Start the Administration Server to ensure the changes you have made to the domain have been applied.

Propagating the Domain Configuration to WLS_IPM1 and WLS_IPM2

You need to perform the following steps in order to propagate the domain configuration to the Imaging Managed Servers.

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. Run the following `pack` command on WCCHOST1 to create a template pack:

```
cd ORACLE_COMMON_HOME/common/bin

./pack.sh -managed=true -domain=ASERVER_HOME -
template=edgdomaintemplateExtIPM.jar -template_name=edgdomain_templateIPM
```

3. Run the following `unpack` command on WCCHOST1 to propagate the template created in the preceding step to the WLS_IPM1 domain directory:

```
cd ORACLE_COMMON_HOME/common/bin

./unpack.sh -domain=MSERVER_HOME -template=edgdomaintemplateExtIPM.jar -
app_dir=APPLICATION_HOME -overwrite_domain=true
```

4. Run the following command on WCCHOST1 to copy the template pack created in step 1 to WCCHOST2:

```
scp edgdomaintemplateIPM.jar oracle@WCCHOST2:ORACLE_COMMON_HOME/common/bin
```

5. Run the `unpack` command on WCCHOST2 to unpack the propagated template to the WLS_IPM2 domain directory.

```
cd ORACLE_COMMON_HOME/common/bin

./unpack.sh -domain=MSERVER_HOME -template=edgdomaintemplateExtIPM.jar -
app_dir=APPLICATION_HOME -overwrite_domain=true
```

6. Restart the Administration Server to make these changes take effect, stopping it with the `nmKill` command, or with the Administration Console, and then starting it with the `nmStart` command. Before the restart, stop all Managed Servers in the

domain through the Administration Console, and then start them after the restart. Log in to the Administration Console using the credentials for the `weblogic` user.

Configuring Oracle HTTP Server for the Imaging Cluster

To enable Oracle HTTP Server to route to `IPM_Cluster`, which contains the `WLS_IPM1` and `WLS_IPM2` Managed Servers, you must set the `WebLogicCluster` parameter to the list of nodes in the cluster.

This section includes the following topics.

- Configuring Oracle HTTP Server for the WLS_IPM Managed Servers
- Setting the Front-End HTTP Host and Port for the Imaging Cluster
- Validating Access Through the Load Balancer

Configuring Oracle HTTP Server for the WLS_IPM Managed Servers

To configure Oracle HTTP Server for the WLS_IPM Managed Servers:

1. For each of the web servers on WEBHOST1 and WEBHOST2, add the following lines to the `ORACLE_INSTANCE/config/OHS/ohs1/moduleconf/wcc_vh.conf` and `ORACLE_INSTANCE/config/OHS/ohs2/moduleconf/wcc_vh.conf` files:

```
#Imaging
<Location /imaging>
   WebLogicCluster WCCHOST1:16000,WCCHOST2:16000
   WLSRequest ON
   WLCookieName JSESSIONID
   WLProxySSL ON
   WLProxySSLPassThrough ON
</Location>
```

> **Note:**
>
> If AXF webservices are used, then you have to add the following lines to the `wcc_vh.conf` file:

```
#AXF Webservices
<Location /axf-ws>
   WebLogicCluster WCCHOST1:16000,WCCHOST2:16000
   WLSRequest ON
   WLCookieName JSESSIONID
   WLProxySSL ON
   WLProxySSLPassThrough ON
</Location>
```

2. Restart Oracle HTTP Server on both WEBHOST1 and WEBHOST2:

   ```
   ./stopComponent.sh ohsn
   ./startComponent.sh ohsn
   ```

   For WEBHOST1, use `ohs1` (where n=1) and for WEBHOST2, use `ohs2` (where n=2).

Setting the Front-End HTTP Host and Port for the Imaging Cluster

To set the front-end HTTP host and port for the Imaging cluster:

1. Log in to the WebLogic Server Administration Console.
2. Go to the Change Center section and click **Lock & Edit**.
3. Expand the **Environment** node in the **Domain Structure** tree on the left.
4. Click **Clusters**.
5. On the Summary of Clusters page, select **IPM_Cluster**.
6. Open the **HTTP** tab.
7. Set the following values:
   - **Frontend Host:** `wcc.example.com`
   - **Frontend HTTPS Port:** `443`
   - **Frontend HTTP Port:** `80`
8. Click **Save**.
9. Click **Activate Changes** in the Change Center section of the Administration Console.
10. Restart the servers to make the front-end host directive in the cluster take effect.

Validating Access Through the Load Balancer

Verify URLs to ensure that appropriate routing and failover is working from the HTTP Server to IPM_Cluster. To verify the URLs:

1. While WLS_IPM2 is running, stop WLS_IPM1 from the WebLogic Server Administration Console.
2. Access `https://wcc.example.com/imaging` to verify that it is functioning properly. (You will not be able to retrieve reports or data because the Imaging server is down.)
3. Start WLS_IPM1 from the WebLogic Server Administration Console.
4. Stop WLS_IPM2 from the WebLogic Server Administration Console.
5. Access `https://wcc.example.com/imaging` to verify that it is functioning properly.
6. Start WLS_IPM2 from the WebLogic Server Administration Console.

17

Extending the Domain to Include WebCenter Content User Interface

You need to perform certain tasks in order to extend the enterprise deployment domain to include Oracle WebCenter Content User Interface software.

This chapter provides information on modifying system-level setting through MBeans and configuring http server with the WebCenter Content user interface cluster.

- Extending the Domain for WebCenter Content User Interface
  The instructions for extending the existing enterprise deployment domain with the Oracle WebCenter Content user interface software are detailed in this section.
- Propagating the Domain Configuration to WLS_WCCUI1 and WLS_WCCUI2
  You need to perform the following steps in order to propagate the domain configuration to the WebCenter Content user interface Managed Servers.
- Modifying System-Level Settings Through MBeans
  To ensure high availability, modify system-level configuration settings for WebCenter Content user interface through the System Configuration page in Fusion Middleware Control. The settings on this page configure the WebCenter Content user interface MBeans for the domain.
- Configuring Oracle HTTP Server with the WebCenter Content User Interface Cluster
  Configure Oracle HTTP Server with the WebCenter Content User Interface Cluster, set the front-end HTTP Host and Port for the cluster, and validate access through the load balancer.
- Completing the Workflow Configuration
  To complete the workflow configuration for the WebCenter Content user interface, you need to restart the Managed Servers and verify the configuration. The `UseDatabaseWfInQueue` configuration variable enables the WebCenter Content user interface to filter workflows assigned to a user. The `EmailNotificationType` configuration variable specifies where the links in notification emails point for workflows and subscriptions in different Content Server user interfaces, and its default value is `NativeWebUI`.

Extending the Domain for WebCenter Content User Interface

The instructions for extending the existing enterprise deployment domain with the Oracle WebCenter Content user interface software are detailed in this section.

Extending the domain involves the following tasks.

- Starting the Configuration Wizard
  Start the Configuration Wizard as the first step to extend the existing enterprise deployment domain.

- Navigating the Configuration Wizard Screens to Extend the Domain

Starting the Configuration Wizard

Start the Configuration Wizard as the first step to extend the existing enterprise deployment domain.

> **Note:**
>
> If you added any customizations directly to the start scripts in the domain, those are overwritten by the configuration wizard. To customize server startup parameters that apply to all servers in a domain, you can create a file called `setUserOverridesLate.sh` and configure it, for example, add custom libraries to the WebLogic Server classpath, specify Additional JAVA command line options for running the servers, or specify additional environment variables. Any customizations you add to this file are preserved during domain upgrade operations, and are carried over to remote servers when using the **`pack`** and **`unpack`** commands.

To start the Configuration Wizard:

1. From the WebLogic Server Console, stop any managed servers that are modified by this domain extension. Managed Servers that are not effected can remain on-line.
2. For any managed servers to be modified, verify that the managed server shutdown has completed.
3. Stop the Administration Server once all managed servers are in a steady state.
4. Navigate to the following directory and start the WebLogic Server Configuration Wizard.

   ```
   cd ORACLE_HOME/oracle_common/common/bin
   ./config.sh
   ```

Navigating the Configuration Wizard Screens to Extend the Domain

Follow the instructions in the following sections to create and configure the domain for the topology with static or dynamic clusters.

- Extending the Domain with Static Clusters
- Extending the Domain with Dynamic Clusters

Extending the Domain with Static Clusters

Follow the instructions in this section to create and configure the domain for the topology.

> **Note:**
>
> You can use the same procedure described in this section to extend an existing domain. If your needs do not match the instructions given in the procedure, be sure to make your selections accordingly, or refer to the supporting documentation for additional details.

Domain creation and configuration includes the following tasks:

- Task 1, Selecting the Domain Type and Domain Home Location
- Task 2, Selecting the Configuration Template
- Task 3, Specifying JDBC Data Sources Information
- Task 4, Providing the GridLink Oracle RAC Data Sources Details
- Task 5, Testing the JDBC Data Sources
- Task 6, Providing the GridLink Oracle RAC Component Schema Details
- Task 7, Testing the JDBC Component Schema
- Task 8, Specifying Credentials for wccadfConnectUser
- Task 9, Selecting Advanced Configuration
- Task 10, Configuring Managed Servers
- Task 11, Configuring a Cluster
- Task 12, Assigning Server Templates
- Task 13, Configuring Dynamic Servers
- Task 14, Assigning Managed Servers to the Cluster
- Task 15, Configuring Coherence Clusters
- Task 16, Verifying the Existing Machines
- Task 17, Assigning Servers to Machines
- Task 18, Configuring Virtual Targets
- Task 19, Configuring Partitions
- Task 20, Reviewing Configuration Summary
- Task 21, Reviewing Configuration Progress
- Task 22, Reviewing Your Configuration Specifications and Configuring the Domain
- Task 23, Writing Down Your Domain Home and Administration Server URL
- Task 24, Start the Administration Server

Task 1 Selecting the Domain Type and Domain Home Location
On the Configuration Type screen, select **Update an existing domain**.
In the **Domain Location** field, select the value of the *ASERVER_HOME* variable, which represents the complete path to the Administration Server domain home you created in Creating the Initial Infrastructure Domain for an Enterprise Deployment.
For more information about the directory location variables, see File System and Directory Variables Used in This Guide.

> **Tip:**
>
> More information about the other options on this screen can be found in Configuration Type in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 2 Selecting the Configuration Template
On the Templates screen, make sure **Update Domain Using Product Templates** is selected, then select the following templates:

- **Oracle WebCenter Content - Web UI - 12.2.1.3.0 [wccontent]**

  In addition, the following additional templates should already be selected, because they were used to create the initial domain:

  - **Oracle Universal Content Mnagement - Content Server - 12.2.1.3.0 [wccontent]**
  - **Oracle SOA Suite - 12.2.1.3.0 [soa]**
  - **Oracle Universal Content Management - Inbound Refinery - 12.2.1.3.0 [wccontent]**
  - **Oracle WebCenter Enterprise Capture - 12.2.1.3.0 [wccapture]**
  - **Oracle Enterprise Manager - 12.2.1.3.0 [em]**
  - **Oracle WSM Policy Manager - 12.2.1.3.0 [oracle_common]**
  - **Oracle JRF - 12.2.1.3.0 [oracle_common]**
  - **WebLogic Coherence Cluster Extension - 12.2.1.3.0 [wlserver]**

> **Tip:**
>
> More information about the options on this screen can be found in Templates in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 3 Specifying JDBC Data Sources Information
Click **Convert to GridLink**, update the required database details, and then click **Next**.

Task 4 Providing the GridLink Oracle RAC Data Sources Details
On the GridLink Oracle RAC Data Sources screen, perform the following tasks.
In the SCAN, Host Name, and Port section:

1. Select the SCAN check box.
2. In the **Host Name** field, enter the Single Client Access Name (SCAN) Address for the Oracle RAC database.
3. In the **Port** field, enter the SCAN listening port for the database (for example, `1521`).

In the ONS Host and Port section:

1. In the **ONS Host** field, enter the SCAN address for the Oracle RAC database.
2. In the **Port** field, enter the ONS Remote port (typically, `6200`).

Click **Next**.

Task 5 Testing the JDBC Data Sources
Click **Next** to continue.

Task 6 Providing the GridLink Oracle RAC Component Schema Details
Click **Next** to continue.

Task 7 Testing the JDBC Component Schema
Click **Next** to continue.

Task 8 Specifying Credentials for wccadfConnectUser
On the Credentials screen, enter the WebLogic user name(not `weblogic_wcc`) and password.
Click **Next** to continue.

Task 9 Selecting Advanced Configuration
To complete domain configuration for the topology, select the following options on the Advanced Configuration screen:
Topology

Task 10 Configuring Managed Servers
On the Managed Servers screen, a new Managed Server appears in the list of servers.
Perform the following tasks to modify the default Managed Server and create a second Managed Server:

1. Rename the default Managed Server to `WLS_WCCUI1`.
2. Click **Add** to create a new Managed Server and name it `WLS_WCCUI2`.

> **Tip:**
>
> The server names recommended here will be used throughout this document; if you choose different names, be sure to replace them as needed.

3. Use the following table to fill in the rest of the columns for each Oracle WebCenter Content user interface Managed Server.

> **Tip:**
>
> More information about the options on the Managed Server screen can be found in Managed Servers in *Creating WebLogic Domains Using the Configuration Wizard*.

Server Name	Listen Address	Listen Port	Enable SSL	SSL Listen Port	Server Groups
WLS_WCCUI 1	WCCHOST1	16225	No	Disabled	UCM-ADF-MGD-SVR

Server Name	Listen Address	Listen Port	Enable SSL	SSL Listen Port	Server Groups
WLS_WCCUI2	WCCHOST2	16225	No	Disabled	UCM-ADF-MGD-SVR

Task 11 Configuring a Cluster
In this task, you create a cluster of Managed Servers to which you can target the Oracle WebCenter Content software.
You will also set the **Frontend Host** property for the cluster, which ensures that, when necessary, WebLogic Server will redirect Web services callbacks and other redirects to `wcc.example.com` on the load balancer rather than the address in the HOST header of each request.
For more information about the `wcc.example.com` virtual server address, see Configuring Virtual Hosts on the Hardware Load Balancer.
Use the Clusters screen to create a new cluster:

1. Click the **Add** button.
2. Specify `WCCUI_Cluster` in the **Cluster Name** field.
3. From the **Dynamic Server Groups** drop-down list, select `Unspecified`.

> **Note:**
>
> By default, server instances in a cluster communicate with one another using unicast. If you want to change your cluster communications to use multicast, refer to "Considerations for Choosing Unicast or Multicast" in *Administering Clusters for Oracle WebLogic Server*.

> **Tip:**
>
> More information about the options on this screen can be found in Clusters in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 12 Assigning Server Templates
Click **Next** to proceed to the next screen.

Task 13 Configuring Dynamic Servers
Verify that all dynamic server options are disabled for clusters that are to remain as static clusters.

1. Confirm that the **Dynamic Cluster**, **Calculated Listen Port**, and **Calculated Machine Names** checkboxes on this screen are unchecked.
2. Confirm the **Server Template** selection is **Unspecified**.
3. Click **Next**.

Task 14 Assigning Managed Servers to the Cluster
Use the Assign Servers to Clusters screen to assign `WLS_WCCUI1` and `WLS_WCCUI2` to the new cluster, `WCCUI_Cluster`:

1. In the Clusters pane, select the cluster to which you want to assign the servers; in this case, `WCCUI_Cluster`.
2. In the Servers pane, assign WLS_WCCUI1 to `WCCUI_Cluster` by doing one of the following:
   - Click the `WLS_WCCUI1` Managed Server once to select it, then click on the right arrow to move it beneath the selected cluster in the Clusters pane.
   - Double-click `WLS_WCCUI1` to move it beneath the selected cluster in the clusters pane.
3. Repeat to assign `WLS_WCCUI2` to `WCCUI_Cluster`.

> **Tip:**
>
> More information about the options on this screen can be found in Assign Servers to Clusters in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 15 Configuring Coherence Clusters
Use the Coherence Clusters screen to configure the Coherence cluster that is automatically added to the domain. Leave the port number value at `0`, as it was defined during the initial Infrastructure domain creation.

> **Note:**
>
> For Coherence licensing information, refer to *Oracle Coherence* in *Oracle Fusion Middleware Licensing Information*.

Task 16 Verifying the Existing Machines
Under the **Unix Machine** tab, verify the names of the machines you created when creating the initial Infrastructure domain.
Click **Next** to proceed.

Task 17 Assigning Servers to Machines
Use the Assign Servers to Machines screen to assign the Oracle WebCenter Content user interface Managed Servers you just created to the corresponding machines in the domain.
Assign `WLS_WCCUI1` to `WCCHOST1`, and assign `WLS_WCCUI2` to `WCCHOST2`.

> **Tip:**
>
> More information about the options on this screen can be found in Assign Servers to Machines in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 18 Configuring Virtual Targets
Click **Next** to proceed to the next screen.

Task 19 Configuring Partitions
Click **Next** to proceed to the next screen.

Task 20 Reviewing Configuration Summary
Click **Update** to proceed to the next screen.

Task 21 Reviewing Configuration Progress
After all the processes are complete, click **Next** to proceed to the next screen.

Task 22 Reviewing Your Configuration Specifications and Configuring the Domain
The Configuration Summary screen contains the detailed configuration information for the domain you are about to create. Review the details of each item on the screen and verify that the information is correct.
You can go back to any previous screen if you need to make any changes, either by using the **Back** button or by selecting the screen in the navigation pane.
Domain creation will not begin until you click **Update**.

> **Tip:**
>
> More information about the options on this screen can be found in Configuration Summary in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 23 Writing Down Your Domain Home and Administration Server URL
The Configuration Success screen will show the following items about the domain you just configured:

- Domain Location
- Administration Server URL

You must make a note of both items as you will need them later; the domain location is needed to access the scripts used to start the Node Manager and Administration Server, and the URL is needed to access the Administration Server.
Click **Finish** to dismiss the configuration wizard.

Task 24 Start the Administration Server
Start the Administration Server to ensure the changes you have made to the domain have been applied.

After you have completed extending the domain with static clusters, go to Propagating the Domain Configuration to WLS_WCCUI1 and WLS_WCCUI2.

Extending the Domain with Dynamic Clusters

Follow the instructions in this section to create and configure the domain for the topology with Dynamic Clusters.

> **Note:**
>
> You can use the same procedure described in this section to extend an existing domain with Dynamic Clusters. If your needs do not match the instructions given in the procedure, be sure to make your selections accordingly, or refer to the supporting documentation for additional details.

Domain creation and configuration includes the following tasks:

- Task 1, Selecting the Domain Type and Domain Home Location
- Task 2, Selecting the Configuration Template
- Task 3, Specifying JDBC Data Sources Information
- Task 4, Providing the GridLink Oracle RAC Data Sources Details
- Task 5, Testing the JDBC Data Sources
- Task 6, Providing the GridLink Oracle RAC Component Schema Details
- Task 7, Testing the JDBC Component Schema
- Task 8, Specifying Credentials for wccadfConnectUser
- Task 9, Selecting Advanced Configuration
- Task 10, Configuring Managed Servers
- Task 11, Configuring a Cluster
- Task 12, Assigning Server Templates
- Task 13, Configuring Dynamic Servers
- Task 14, Assigning Managed Servers to the Cluster
- Task 15, Configuring Coherence Clusters
- Task 16, Verifying the Existing Machines
- Task 17, Assigning Servers to Machines
- Task 18, Configuring Virtual Targets
- Task 19, Configuring Partitions
- Task 20, Reviewing Configuration Summary
- Task 21, Reviewing Configuration Progress
- Task 22, Reviewing Your Configuration Specifications and Configuring the Domain
- Task 23, Writing Down Your Domain Home and Administration Server URL
- Task 24, Start the Administration Server

Task 1 Selecting the Domain Type and Domain Home Location
On the Configuration Type screen, select **Update an existing domain**.
In the **Domain Location** field, select the value of the *ASERVER_HOME* variable, which represents the complete path to the Administration Server domain home you created in Creating the Initial Infrastructure Domain for an Enterprise Deployment.
For more information about the directory location variables, see File System and Directory Variables Used in This Guide.

> **Tip:**
>
> More information about the other options on this screen can be found in Configuration Type in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 2 Selecting the Configuration Template
On the Templates screen, make sure **Update Domain Using Product Templates** is selected, then select the following templates:

- **Oracle WebCenter Content - Web UI - 12.2.1.3.0 [wccontent]**

  In addition, the following additional templates should already be selected, because they were used to create the initial domain:

  - **Oracle Universal Content Mnagement - Content Server - 12.2.1.3.0 [wccontent]**
  - **Oracle SOA Suite - 12.2.1.3.0 [soa]**
  - **Oracle Universal Content Management - Inbound Refinery - 12.2.1.3.0 [wccontent]**
  - **Oracle WebCenter Enterprise Capture - 12.2.1.3.0 [wccapture]**
  - **Oracle Enterprise Manager - 12.2.1.3.0 [em]**
  - **Oracle WSM Policy Manager - 12.2.1.3.0 [oracle_common]**
  - **Oracle JRF - 12.2.1.3.0 [oracle_common]**
  - **WebLogic Coherence Cluster Extension - 12.2.1.3.0 [wlserver]**

> **Tip:**
>
> More information about the options on this screen can be found in Templates in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 3 Specifying JDBC Data Sources Information
Click **Convert to GridLink**, update the required database details, and then click **Next**.

Task 4 Providing the GridLink Oracle RAC Data Sources Details
On the GridLink Oracle RAC Data Sources screen, perform the following tasks.
In the SCAN, Host Name, and Port section:

1. Select the **SCAN** check box.
2. In the **Host Name** field, enter the Single Client Access Name (SCAN) Address for the Oracle RAC database.
3. In the **Port** field, enter the SCAN listening port for the database (for example, `1521`).

In the ONS Host and Port section:

1. In the **ONS Host** field, enter the SCAN address for the Oracle RAC database.
2. In the **Port** field, enter the ONS Remote port (typically, `6200`).

Click **Next**.

Task 5 Testing the JDBC Data Sources
Click **Next** to continue.

Task 6 Providing the GridLink Oracle RAC Component Schema Details
Click **Next** to continue.

Task 7 Testing the JDBC Component Schema
Click **Next** to continue.

Task 8 Specifying Credentials for wccadfConnectUser
On the Credentials screen, enter the WebLogic user name (not `weblogic_wcc`) and password.
Click **Next** to continue.

Task 9 Selecting Advanced Configuration
To complete domain configuration for the topology, select the following options on the Advanced Configuration screen:
Topology

Task 10 Configuring Managed Servers
On the Managed Servers screen, a new Managed Server appears in the list of servers.
Perform the following tasks to modify the default Managed Server and create a second Managed Server:

1. Delete the default Managed Server.
2. Click **Next** to proceed to the next screen.

Task 11 Configuring a Cluster
In this task, you create a cluster of Managed Servers to which you can target the Oracle WebCenter Content software.
You will also set the **Frontend Host** property for the cluster, which ensures that, when necessary, WebLogic Server will redirect Web services callbacks and other redirects to `wcc.example.com` on the load balancer rather than the address in the HOST header of each request.
For more information about the `wcc.example.com` virtual server address, see Configuring Virtual Hosts on the Hardware Load Balancer.
Use the Clusters screen to create a new cluster:

1. Click the **Add** button.
2. Specify `WCCUI_Cluster` in the **Cluster Name** field.
3. From the Dynamic Server Groups drop-down list, select `WCCADF-DYN-CLUSTER`.

> **Note:**
>
> By default, server instances in a cluster communicate with one another using unicast. If you want to change your cluster communications to use multicast, refer to "Considerations for Choosing Unicast or Multicast" in *Administering Clusters for Oracle WebLogic Server*.

> **Tip:**
>
> More information about the options on this screen can be found in Clusters in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 12 Assigning Server Templates
Use the Server Templates screen to add a template:

1. Specify `WCCADF-server-template` in the **Name** field.
2. Specify `16225` in the **Listen Port** field.
3. Leave the **Enable SSL** option unchecked.
4. From the **Cluster** drop-down list, select `Unspecified`.
5. Click **Next** to proceed to the next screen.

Task 13 Configuring Dynamic Servers
Use the Dynamic Clusters screen to configure the required clusters:

1. Specify `WCCUI_Cluster` in the **Cluster Name** field.
2. Specify `WLS_WCCUI` in the **Server Name Prefix** field.
3. From the **Server Template** drop-down list, select `WCCADF-server-template`.
4. Specify `2` in the **Maximum Dynamic Server Count** field.
5. Specify `WCCHOST*` in the **Machine Name Match Expression** field.
6. Select the **Calculated Machine Names** and **Dynamic Cluster** fields.
7. Select the **Calculated Listen Ports** field.
8. Click **Next** to proceed to the next screen.

Task 14 Assigning Managed Servers to the Cluster
Click **Next** to proceed to the next screen.

Task 15 Configuring Coherence Clusters
Use the Coherence Clusters screen to configure the Coherence cluster that is automatically added to the domain. Leave the port number value at `0`, as it was defined during the initial Infrastructure domain creation.

> **Note:**
>
> For Coherence licensing information, refer to *Oracle Coherence* in *Oracle Fusion Middleware Licensing Information*.

Task 16 Verifying the Existing Machines
Under the **Unix Machine** tab, verify the names of the machines you created when creating the initial Infrastructure domain.
Click **Next** to proceed to the next screen.

Task 17 Assigning Servers to Machines
Click **Next** to proceed to the next screen.

Task 18 Configuring Virtual Targets
Click **Next** to proceed to the next screen.

Task 19 Configuring Partitions
Click **Next** to proceed to the next screen.

Task 20 Reviewing Configuration Summary
Click **Update** to proceed to the next screen.

Task 21 Reviewing Configuration Progress
After all the processes are complete, click **Next** to proceed to the next screen.

Task 22 Reviewing Your Configuration Specifications and Configuring the Domain
The Configuration Summary screen contains the detailed configuration information for the domain you are about to create. Review the details of each item on the screen and verify that the information is correct.
You can go back to any previous screen if you need to make any changes, either by using the **Back** button or by selecting the screen in the navigation pane.
Domain creation will not begin until you click **Update**.

> **Tip:**
>
> More information about the options on this screen can be found in Configuration Summary in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 23 Writing Down Your Domain Home and Administration Server URL
The Configuration Success screen will show the following items about the domain you just configured:

- Domain Location
- Administration Server URL

You must make a note of both items as you will need them later; the domain location is needed to access the scripts used to start the Node Manager and Administration Server, and the URL is needed to access the Administration Server.
Click **Finish** to dismiss the configuration wizard.

Task 24 Start the Administration Server
Start the Administration Server to ensure the changes you have made to the domain have been applied.

Propagating the Domain Configuration to WLS_WCCUI1 and WLS_WCCUI2

You need to perform the following steps in order to propagate the domain configuration to the WebCenter Content user interface Managed Servers.

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. Run the following `pack` command on WCCHOST1 to create a template pack:

```
cd ORACLE_COMMON_HOME/common/bin

./pack.sh -managed=true -domain=ASERVER_HOME -
template=edgdomaintemplateWCCUI.jar -template_name=edgdomain_templateWCCUI
```

3. Run the following `unpack` command on WCCHOST1 to propagate the template created in the preceding step to the WLS_WCCUI1 domain directory:

```
cd ORACLE_COMMON_HOME/common/bin

./unpack.sh -domain=MSERVER_HOME -template=edgdomaintemplateWCCUI.jar -
app_dir=APPLICATION_HOME -overwrite_domain=true
```

4. Run the following command on WCCHOST1 to copy the template pack created in step 1 to WCCHOST2:

```
scp edgdomaintemplateWCCUI.jar oracle@WCCHOST2:ORACLE_COMMON_HOME/common/bin
```

5. Run the `unpack` command on WCCHOST2 to unpack the propagated template to the WLS_WCCUI2 domain directory.

```
cd ORACLE_COMMON_HOME/common/bin

./unpack.sh -domain=MSERVER_HOME -template=edgdomaintemplateWCCUI.jar -
app_dir=APPLICATION_HOME -overwrite_domain=true
```

6. Restart the Administration Server to make these changes take effect, stopping it with the `nmKill` command, or with the Administration Console, and then starting it with the `nmStart` command. Before the restart, stop all Managed Servers in the domain through the Administration Console, and then start them after the restart. Log in to the Administration Console using the credentials for the `weblogic` user.

Modifying System-Level Settings Through MBeans

To ensure high availability, modify system-level configuration settings for WebCenter Content user interface through the System Configuration page in Fusion Middleware Control. The settings on this page configure the WebCenter Content user interface MBeans for the domain.

For information about how to modify system-level settings in Fusion Middleware control, see Modifying System Configuration Settings in *Oracle Fusion Middleware Administering Oracle WebCenter Capture*.

Modify the required parameters:

1. To set the `AdfScopeHaSupport` parameter, complete the following steps:
   a. From the **WebLogic Domin** menu, select **System MBean Browser**.
   b. From the left navigation, go to **Application Defined MBeans** > **oracle.adf.share.config** > **Domain:***WCC server name* > **Application: Oracle WebCenter Content - Web UI** > **ADFConfig** > **ADFConfig** > **ADFConfig** > **ADFcConfiguration**.
   c. Set the `AdfScopeHaSupport` parameter to `true`.
   d. Click **Apply**.

2. To set the `ClusterCompatible` and `TemporaryDirectory` parameters, complete the following steps:

a. From the **WebLogic Domin** menu, select **System MBean Browser**.

b. From the left navigation, go to **Application Defined MBeans** > **oracle.adf.share.config** > **Domain:***WCC server name* > **Application: Oracle WebCenter Content - Web UI** > **ADFConfig** > **ADFConfig** > **ADFConfig** > **WccAdfConfiguration**.

c. Set the `ClusterCompatible` parameter to `true`.

d. Set the `TemporaryDirectory` parameter to `/u01/oracle/config/domains/WCCDomain/WCCUI_Cluster/tempdir`.

e. Click **Apply**.

3. To set the `PropConnectionUrl` parameter, complete the following steps:

a. From the **WebLogic Domin** menu, select **System MBean Browser**.

b. From the left navigation, go to **Application Defined MBeans** > **oracle.adf.share.connections** > **Domain:***WCC server name* > **Application: Oracle WebCenter Content - Web UI** > **ADFConnections** > **ADFConnections** > **WccConnection** > **WccAdfServerConnection**.

c. Set the `PropConnectionUrl` parameter to `idc://wccinternal.example.com:6300`.

d. Click **Apply**.

4. To set the `ApplicationUrl` parameter, complete the following steps:

a. From the **WebLogic Domin** menu, select **System MBean Browser**.

b. From the left navigation, go to **Application Defined MBeans** > **oracle.adf.share.connections** > **Domain:***WCC server name* > **Application: Oracle WebCenter Content - Web UI** > **ADFConfig** > **ADFConfig** > **ADFConfig** > **WccAdfConfiguration**.

c. Set the `ApplicationUrl` parameter to `https://wcc.example.com:443`.

d. Click **Apply**.

Configuring Oracle HTTP Server with the WebCenter Content User Interface Cluster

Configure Oracle HTTP Server with the WebCenter Content User Interface Cluster, set the front-end HTTP Host and Port for the cluster, and validate access through the load balancer.

This section contains the following tasks.

- Configuring Oracle HTTP Server for the WLS_WCCUI Managed Servers
- Setting the Front-End HTTP Host and Port for the WebCenter Content User Interface Cluster
- Validating Access Through the Load Balancer

Configuring Oracle HTTP Server for the WLS_WCCUI Managed Servers

To configure Oracle HTTP Server for the WLS_WCCUI Managed Servers:

1. For each of the web servers on WEBHOST1 and WEBHOST2, add the following lines to the `ORACLE_INSTANCE/config/OHS/ohs1/moduleconf/wcc_vh.conf` and `ORACLE_INSTANCE/config/OHS/ohs2/moduleconf/wcc_vh.conf` files:

```
# ADF UI
<Location /wcc>
  WebLogicCluster WCCHOST1:16225,WCCHOST2:16225
  WLSRequest ON
  WLCookieName WCCSID
  WLProxySSL ON
  WLProxySSLPassThrough ON
</Location>
```

2. Restart both Oracle HTTP Servers.

Setting the Front-End HTTP Host and Port for the WebCenter Content User Interface Cluster

To set the front-end HTTP host and port for the WCC UI cluster:

1. Log in to the WebLogic Server Administration Console.
2. Go to the Change Center section and click **Lock & Edit**.
3. Expand the **Environment** node in the **Domain Structure** tree on the left.
4. Click **Clusters**.
5. On the Summary of Clusters page, select **WCCUI_Cluster**.
6. Open the **HTTP** tab.
7. Set the following values:
   - **Frontend Host:** `wcc.example.com`
   - **Frontend HTTPS Port:** `443`
   - **Frontend HTTP Port:** `80`
8. Click **Save**.
9. Click **Activate Changes** in the Change Center section of the Administration Console.
10. Restart the servers to make the front-end host directive in the cluster take effect.

Validating Access Through the Load Balancer

Verify URLs to ensure that appropriate routing and failover is working from the HTTP Server to WCCUI_Cluster. To verify the URLs:

1. While WLS_WCCUI2 is running, stop WLS_WCCUI1 from the WebLogic Server Administration Console.
2. Access `https://wcc.example.com/wcc` to verify that it is functioning properly. (You will not be able to retrieve reports or data because the Capture server is down.)
3. Start WLS_WCCUI1 from the WebLogic Server Administration Console.
4. Stop WLS_WCCUI2 from the WebLogic Server Administration Console.
5. Access `https://wcc.example.com/wcc` to verify that it is functioning properly.

You can verify the cluster node to which you were directed after the traffic balancing provided through your load balancer and then again through the web tier.

Completing the Workflow Configuration

To complete the workflow configuration for the WebCenter Content user interface, you need to restart the Managed Servers and verify the configuration. The `UseDatabaseWfInQueue` configuration variable enables the WebCenter Content user interface to filter workflows assigned to a user. The `EmailNotificationType` configuration variable specifies where the links in notification emails point for workflows and subscriptions in different Content Server user interfaces, and its default value is `NativeWebUI`.

To complete the workflow configuration:

1. Make sure that the `WCCHOST1/ucm/cs/config/config.cfg` file contains the *EmailNotificationType* variable with either of the following settings:
   - To generate emails with links that point only to the WebCenter Content user interface, set `EmailNotificationType=ContentUI` in `config.cfg`.
   - To generate emails with links that point to both the WebCenter Content user interface and the native 11*g* user interface, set `EmailNotificationType=ContentUI,NativeWebUI` in `config.cfg`.
2. Restart the Content Server Managed Server, using the WebLogic Server Administration Console.
3. Click the alert that appears on the Content Server home page after restart: **Click to complete workflow setup**.

   Ensure that Content Server returns a success message: `The setup for workflow in queue was successful`.
4. Restart the WebCenter Content user interface Managed Server, using the WebLogic Server Administration Console for the WebCenter Content user interface domain.

or more information about workflows, see ”Managing Workflows” in *Oracle Fusion Middleware Managing Oracle WebCenter Content*.

Part IV

Common Configuration and Management Procedures for an Enterprise Deployment

There are certain configuration and management procedures that are recommended for a typical enterprise deployment.

The following topics contain configuration and management procedures that are required for a typical enterprise deployment.

- Common Configuration and Management Tasks for an Enterprise Deployment
  The configuration and management tasks that may need to be performed on the enterprise deployment environment are detailed in this section.
- Using Whole Server Migration and Service Migration in an Enterprise Deployment
  The Oracle WebLogic Server migration framework supports Whole Server Migration and Service Migration. The following sections explain how these features can be used in an Oracle Fusion Middleware enterprise topology.
- Configuring Single Sign-On for an Enterprise Deployment
  You need to configure the Oracle HTTP Server WebGate in order to enable single sign-on with Oracle Access Manager.

18

Common Configuration and Management Tasks for an Enterprise Deployment

The configuration and management tasks that may need to be performed on the enterprise deployment environment are detailed in this section.

- Verifying Appropriate Sizing and Configuration for the WLSSchemaDataSource
  `WLSSchemaDataSource` is the common datasource that is reserved for use by the FMW components for JMS JDBC Stores, JTA JDBC stores, and Leasing services. `WLSSchemaDataSource` is used to avoid contention in critical WLS infrastructure services and to guard against dead-locks.
- Verifying Manual Failover of the Administration Server
  In case a host computer fails, you can fail over the Administration Server to another host. The steps to verify the failover and failback of the Administration Server from WCCHOST1 and WCCHOST2 are detailed in the following sections.
- Configuring Listen Addresses in Dynamic Cluster Server Templates
  The default configuration for dynamic managed servers in dynamic clusters is to listen on all available network interfaces. In most cases, this may be undesirable.
- Enabling SSL Communication Between the Middle Tier and the Hardware Load Balancer
  It is important to understand how to enable SSL communication between the middle tier and the hardware load balancer.
- Configuring Roles for Administration of an Enterprise Deployment
  In order to manage each product effectively within a single enterprise deployment domain, you must understand which products require specific administration roles or groups, and how to add a product-specific administration role to the Enterprise Deployment Administration group.
- Using Persistent Stores for TLOGs and JMS in an Enterprise Deployment
  The persistent store provides a built-in, high-performance storage solution for WebLogic Server subsystems and services that require persistence.
- Performing Backups and Recoveries for an Enterprise Deployment
  It is recommended that you follow the below mentioned guidelines to make sure that you back up the necessary directories and configuration data for an Oracle WebCenter Content enterprise deployment.
- Modifying the Upload and Stage Directories to an Absolute Path in an Enterprise Deployment
  After you configure the domain and unpack it to the Managed Server domain directories on all the hosts, verify and update the upload and stage directories for Managed Servers in the new clusters. Also, update the upload directory for the AdminServer to have the same absolute path instead of relative, otherwise deployment issues can occur. If you implement dynamic clusters, the configuration of the server template assigned to each newly added cluster should be verified and updated, otherwise, verify and update every statically-defined Managed Server for the newly added clusters.

- Starting and Stopping Servers in Dynamic Clusters
  You can start and stop server instances in dynamic clusters by using the same methods you use to start and stop server instances in configured static clusters.
- Expanding or Reducing Dynamic Clusters
  When you create a dynamic cluster, WebLogic Server generates the number of dynamic servers you specify. Before you decide upon the number of server instances, ensure you have the performance capacity to handle the desired number.

Verifying Appropriate Sizing and Configuration for the WLSSchemaDataSource

`WLSSchemaDataSource` is the common datasource that is reserved for use by the FMW components for JMS JDBC Stores, JTA JDBC stores, and Leasing services. `WLSSchemaDataSource` is used to avoid contention in critical WLS infrastructure services and to guard against dead-locks.

To reduce the `WLSSchemaDataSource` connection usage, you can change the JMS JDBC and TLOG JDBC stores connection caching policy from *Default* to *Minimal* by using the respective connection caching policy settings. When there is a need to reduce connections in the back-end database system, Oracle recommends that you set the caching policy to *Minimal* . Avoid using the caching policy *None* because it causes a potential degradation in performance. For a detailed tuning advice about connections that are used by JDBC stores, see Configuring a JDBC Store Connection Caching Policy in *Oracle Fusion Middleware Administering the WebLogic Persistent Store*.

The default `WLSSchemaDataSource` connection pool size is 75 (size is double in the case of a GridLink DataSource). You can tune this size to a higher value depending on the size of the different FMW clusters and the candidates that are configured for migration. For example, consider a typical SOA EDG deployment with the default number of worker threads per store. If more than 25 JDBC Stores or TLOG-in-DB instances or both can fail over to the same Weblogic server, and the Connection Caching Policy is not changed from *Default* to *Minimal*, possible connection contention issues could arise. In these cases, increasing the default `WLSSchemaDataSource` pool size (maximum capacity) becomes necessary (each JMS store uses a minimum of two connections, and leasing and JTA are also added to compete for the pool).

Verifying Manual Failover of the Administration Server

In case a host computer fails, you can fail over the Administration Server to another host. The steps to verify the failover and failback of the Administration Server from WCCHOST1 and WCCHOST2 are detailed in the following sections.

Assumptions:

- The Administration Server is configured to listen on ADMINVHN, and not on localhost or on any other host's address.

  For more information about the ADMINVHN virtual IP address, see Reserving the Required IP Addresses for an Enterprise Deployment.
- These procedures assume that the Administration Server domain home (*ASERVER_HOME*) has been mounted on both host computers. This ensures that

the Administration Server domain configuration files and the persistent stores are saved on the shared storage device.

- The Administration Server is failed over from WCCHOST1 to WCCHOST2, and the two nodes have these IPs:
  - WCCHOST1: 100.200.140.165
  - WCCHOST2: 100.200.140.205
  - ADMINVHN : 100.200.140.206. This is the Virtual IP where the Administration Server is running, assigned to a virtual sub-interface (for example, eth0:1), to be available on WCCHOST1 or WCCHOST2.
- Oracle WebLogic Server and Oracle Fusion Middleware components have been installed in WCCHOST2 as described in the specific configuration chapters in this guide.

  Specifically, both host computers use the exact same path to reference the binary files in the Oracle home.

The following topics provide details on how to perform a test of the Administration Server failover procedure.

- Failing Over the Administration Server to a Different Host
  The following procedure shows how to fail over the Administration Server to a different node (WCCHOST2). Note that even after failover, the Administration Server will still use the same Oracle WebLogic Server *machine* (which is a logical machine, not a physical machine).
- Validating Access to the Administration Server on WCCHOST2 Through Oracle HTTP Server
  If you have configured the web tier to access AdminServer, it is important to verify that you can access the Administration Server after you perform a manual failover of the Administration Server, by using the standard administration URLs.
- Failing the Administration Server Back to WCCHOST1
  After you have tested a manual Administration Server failover, and after you have validated that you can access the administration URLs after the failover, you can then migrate the Administration Server back to its original host.

Failing Over the Administration Server to a Different Host

The following procedure shows how to fail over the Administration Server to a different node (WCCHOST2). Note that even after failover, the Administration Server will still use the same Oracle WebLogic Server *machine* (which is a logical machine, not a physical machine).

This procedure assumes you've configured a per domain Node Manager for the enterprise topology. See About the Node Manager Configuration in a Typical Enterprise Deployment

To fail over the Administration Server to a different host:

1. Stop the Administration Server.
2. Stop the Node Manager in the Administration Server domain directory (ASERVER_HOME).
3. Migrate the ADMINVHN virtual IP address to the second host:

a. Run the following command as root on WCCHOST1 to check the virtual IP address at its CIDR:

```
ip addr show dev ethX
```

Where, `X` is the current interface used by ADMINVHN.

For example:

```
ip addr show dev eth0
```

b. Run the following command as root on WCCHOST1 (where X:Y is the current interface used by ADMINVHN):

```
ip addr del ADMINVHN/CIDR dev ethX:Y
```

Where, `X:Y` is the current interface used by ADMINVHN.

For example:

```
ip addr del 100.200.140.206/24 dev eth0:1
```

c. Run the following command as root on WCCHOST2:

```
ip addr add ADMINVHN/CIDR dev ethX:Y
```

Where, `X:Y` is the current interface used by ADMINVHN.

For example:

```
ip addr add 100.200.140.206/24 dev eth0:1
```

> **Note:**
>
> Ensure that the CIDR representing the netmask and interface to be used to match the available network configuration in WCCHOST2.
>
> The name of the network interface device may something other than eth*X*, especially on systems with redundant bonded interfaces.

4. Update the routing tables using `arping`, for example:

```
arping -b -A -c 3 -I eth0 100.200.140.206
```

5. Start the Node Manager in the Administration Server domain home on WCCHOST2.
6. Start the Administration Server on WCCHOST2.
7. Test that you can access the Administration Server on WCCHOST2 as follows:

a. Ensure that you can access the Oracle WebLogic Server Administration Console using the following URL:

```
http://ADMINVHN:7001/console
```

b. Check that you can access and verify the status of components in Fusion Middleware Control using the following URL:

```
http://ADMINVHN:7001/em
```

Validating Access to the Administration Server on WCCHOST2 Through Oracle HTTP Server

If you have configured the web tier to access AdminServer, it is important to verify that you can access the Administration Server after you perform a manual failover of the Administration Server, by using the standard administration URLs.

From the load balancer, access the following URLs to ensure that you can access the Administration Server when it is running on WCCHOST2:

- `http://admin.example.com/console`

  This URL should display the WebLogic Server Administration console.

- `http://admin.example.com/em`

  This URL should display Oracle Enterprise Manager Fusion Middleware Control.

Failing the Administration Server Back to WCCHOST1

After you have tested a manual Administration Server failover, and after you have validated that you can access the administration URLs after the failover, you can then migrate the Administration Server back to its original host.

1. Stop the Administration Server.
2. Stop the Node Manager in the Administration Server domain directory (ASERVER_HOME).
3. Migrate the ADMINVHN virtual IP address to the second host:
   a. Run the following command as root on WCCHOST2 to check the virtual IP address at its CIDR:

      ```
      ip addr show dev ethX
      ```

      Where, `X` is the current interface used by ADMINVHN.

      For example:

      ```
      ip addr show dev eth0
      ```

   b. Run the following command as root on WCCHOST2 (where X:Y is the current interface used by ADMINVHN):

      ```
      ip addr del ADMINVHN/CIDR dev ethX:Y
      ```

      Where, `X:Y` is the current interface used by ADMINVHN.

      For example:

      ```
      ip addr del 100.200.140.206/24 dev eth0:1
      ```

   c. Run the following command as root on WCCHOST1:

      ```
      ip addr add ADMINVHN/CIDR dev ethX:Y
      ```

      Where, `X:Y` is the current interface used by ADMINVHN.

      For example:

```
ip addr add 100.200.140.206/24 dev eth0:1
```

> **Note:**
>
> Ensure that the CIDR representing the netmask and interface to be used to match the available network configuration in WCCHOST1.

4. Update the routing tables using `arping`, for example:

   ```
   arping -b -A -c 3 -I eht0 100.200.140.206
   ```

5. Start the Node Manager in the Administration Server domain home on WCCHOST1.
6. Start the Administration Server on WCCHOST1.
7. Test that you can access the Administration Server on WCCHOST1 as follows:
   a. Ensure that you can access the Oracle WebLogic Server Administration Console using the following URL:

      `http://ADMINVHN:7001/console`

   b. Check that you can access and verify the status of components in Fusion Middleware Control using the following URL:

      `http://ADMINVHN:7001/em`

Configuring Listen Addresses in Dynamic Cluster Server Templates

The default configuration for dynamic managed servers in dynamic clusters is to listen on all available network interfaces. In most cases, this may be undesirable.

In preparation for disaster recovery, Oracle recommends that you use host name aliases that can be mapped to different IPs in different data centers (for example, WCCHOST1, WCCHOST2) to set each server's listen address to a specific network interface. With dynamic clusters, each server cannot be configured specifically. There is only one listen address configuration in the cluster's server-template. To effectively set the listen-address properly for each dynamic server in the cluster, a calculated macro must be used.
WebLogic Server provides the "${id}" macro which corresponds to the index number of the dynamic server in the cluster. This index starts at the numeral one ("1") and increments to the current managed server count for the cluster. This sequentially-numbered server ID macro can be used with the recommended host naming pattern to have the Listen address calculated for each Dynamic Server to listen on a specific network interface.

This approach is recommended for enterprise deployment environments where there is only one managed server per host per cluster and the cluster is expected to scale-out horizontally only.

To configure the server-template Listen Address using the ${id} macro:

1. Verify that the required WCCHOST*n*, WEBHOST*n*, entries in `/etc/hosts` are configured to the appropriate IP address for the intended machines.

For example:

```
10.229.188.205 host1.example.com host1 WCCHOST1
10.229.188.206 host2.example.com host2 WCCHOST2
10.229.188.207 host3.example.com host5 WEBHOST1
10.229.188.208 host4.example.com host6 WEBHOST2
```

For information about the requirements for name resolution, see Verifying IP Addresses and Host Names in DNS or hosts File.

2. Browse to the Oracle WebLogic Server Administration console, and sign in with your administrative credentials.

   ```
   http://adminvhn:7001/console
   ```

3. **Lock & Edit** the domain.
4. Navigate to **Clusters** > **Server Templates**, and select the server template to be modified.
5. Set the Listen Address value to the appropriate abstracted listener hostname, with the variable assignment as written.

   For example:

   ```
   wsmpm-server-template  : Listen Address = WCCHOST${id}
   UCM-server-template    : Listen Address = WCCHOST${id}
   soa-server-template    : Listen Address = WCCHOST${id}
   CPT-server-template    : Listen Address = WCCHOST${id}
   IPM-server-template    : Listen Address = WCCHOST${id}
   ```

6. Click **Save**.
7. Repeat from step 4 if additional server templates need to be modified.
8. Click **Activate Changes**.
9. Restart the servers that use the template, for the changes to be effective.

- Configuring Server Template Listen Addresses Using the Machine Name

Configuring Server Template Listen Addresses Using the Machine Name

If your host naming or aliasing convention does not follow a sequential numbering pattern starting at 1, to correlate to the internal ID number of each dynamic server, or you desire the cluster to scale-up with multiple managed servers per host per cluster, then an alternative configuration may be preferred. In this case, you can use the `${machineName}` macro value to specify the listen address instead of using a host name prefix and server ID macro pattern. The `${machineName}` macro will use the display name of the machine that is dynamically assigned to the server, and requires that the machine name be resolvable to an IP address.

To configure the server-template Listen Address with the `${machineName}` macro:

1. Browse to the Oracle WebLogic Server Administration console, and sign in with your administrative credentials:

   ```
   http://adminvhn:7001/console
   ```

2. Navigate to **Machines** to review the list of machine names.
3. Validate to ensure that these names are resolvable as network addresses, using the OS commands such as `ping`.
4. **Lock & Edit** the domain.
5. Navigate to **Clusters** and then **Server Templates**, and select the server-template that you want to modify.
6. Set the Listen Address value to `${machineName}` as written here. Do not substitute any other value.
7. Click **Save**.
8. Repeat from step 5 if you want to modify additional server-templates.
9. Click **Activate Changes**.
10. Restart the servers that use the modified server-template, for the changes to be effective.

Enabling SSL Communication Between the Middle Tier and the Hardware Load Balancer

It is important to understand how to enable SSL communication between the middle tier and the hardware load balancer.

> **Note:**
>
> The following steps are applicable if the hardware load balancer is configured with SSL and the front-end address of the system has been secured accordingly.

When is SSL Communication Between the Middle Tier and Load Balancer Necessary?

In an enterprise deployment, there are scenarios where the software running on the middle tier must access the front-end SSL address of the hardware load balancer. In these scenarios, an appropriate SSL handshake must take place between the load balancer and the invoking servers. This handshake is not possible unless the Administration Server and Managed Servers on the middle tier are started by using the appropriate SSL configuration.

Generating Self-Signed Certificates Using the utils.CertGen Utility

This section describes the procedure to create self-signed certificates on WCCHOST1. Create certificates for every app-tier host by using the network name or alias of each host.

The directory where keystores and trust keystores are maintained must be on shared storage that is accessible from all nodes so that when the servers fail over (manually or with server migration), the appropriate certificates can be accessed from the failover node. Oracle recommends that you use central or shared stores for the certificates used for different purposes (for example, SSL set up for HTTP invocations). See the information on filesystem specifications for the KEYSTORE_HOME location provided in Understanding the Recommended Directory Structure for an Enterprise Deployment.

For information on using trust CA certificates instead, see the information about configuring identity and trust in *Oracle Fusion Middleware Administering Security for Oracle WebLogic Server*.

About Passwords

The passwords used in this guide are used only as examples. Use secure passwords in a production environment. For example, use passwords that include both uppercase and lowercase characters as well as numbers.

To create self-signed certificates:

1. Temporarily, set up your environment by running the following script:

   ```
   . WL_HOME/server/bin/setWLSEnv.sh
   ```

   Note that there is a dot(.) and space( ) preceding the script name in order to source the shell script in the current shell.

2. Verify that the *CLASSPATH* environment variable is set:

   ```
   echo $CLASSPATH
   ```

3. Verify that the shared configuration directory folder has been created and mounted to shared storage correctly, as described in Preparing the File System for an Enterprise Deployment.

   For example, use the following command to verify that the shared configuration directory is available to each host:

   ```
   df -h | grep -B1 SHARED_CONFIG_DIR
   ```

Replace *SHARED_CONFIG_DIR* with the actual path to your shared configuration directory.

You can also do a listing of the directory to ensure that it is available to the host:

```
ls -al SHARED_CONFIG_DIR
```

4. Create the keystore home folder structure if does not already exist.

   For example:

   ```
   cd SHARED_CONFIG_DIR
   mkdir keystores
   chown oracle:oinstall keystores
   chmod 750 keystores
   export KEYSTORE_HOME=SHARED_CONFIG_DIR/keystores
   ```

5. Change directory to the keystore home:

   ```
   cd KEYSTORE_HOME
   ```

6. Run the `utils.CertGen` tool to create the certificates for hostnames or aliases used by the managed servers and node managers, one per host.

   > **Note:**
   >
   > You must run the `utils.CertGen` tool to create certificates for all the other hosts that run the Manager Servers.

   Syntax:

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

   Examples:

   ```
   java utils.CertGen password ADMINVHN.example.com_cert \
         ADMINVHN.example.com_key domestic ADMINVHN.example.com

   java utils.CertGen password WCCHOST1.example.com_cert \
         WCCHOST1.example.com_key domestic WCCHOST1.example.com
   ```

7. Repeat the above step for all the remaining hosts used in the system (for example, WCCHOST2, WCCHOST1, and WCCHOST2).

8. For Dynamic clusters, in addition to `ADMINVHN` and one certificate for each host, a certificate matching a wildcard URL should also be generated.

   For example:

   ```
   java utils.CertGen password WILDCARD.example.com_cert \
   WILDCARD.example.com_key domestic \*.example.com
   ```

Creating an Identity Keystore Using the utils.ImportPrivateKey Utility

This section describes how to create an Identity Keystore on `WCCHOST1.example.com`.

In previous sections you have created certificates and keys that reside on shared storage. In this section, the certificate and private keys created earlier for all hosts and ADMINVHN are imported into a new Identity Store. Make sure that you use a different alias for each of the certificate and key pair imported.

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

1. Import the certificate and private key for ADMINVHN and WCCHOST1 into the Identity Store. Make sure that you use a different alias for each of the certificate and key pair imported.

   Syntax:

   ```
   java utils.ImportPrivateKey
         -certfile cert_file
         -keyfile private_key_file
         [-keyfilepass private_key_password]
         -keystore keystore
         -storepass storepass
         [-storetype storetype]
         -alias alias
         [-keypass keypass]
   ```

   > **Note:**
   >
   > The default keystore_type is `jks`.

   Examples:

   ```
   java utils.ImportPrivateKey\
        -certfile KEYSTORE_HOME/ADMINVHN.example.com_cert.pem\
        -keyfile KEYSTORE_HOME/ADMINVHN.example.com_key.pem\
        -keyfilepass password\
        -keystore appIdentityKeyStore.jks\
        -storepass password\
        -alias ADMINVHN\
        -keypass password

   java utils.ImportPrivateKey\
        -certfile KEYSTORE_HOME/WCCHOST1.example.com_cert.pem\
        -keyfile KEYSTORE_HOME/WCCHOST1.example.com_key.pem\
        -keyfilepass password\
        -keystore appIdentityKeyStore.jks\
        -storepass password\
   ```

```
-alias WCCHOST1\
-keypass password
```

2. Repeat the `java importPrivateKey` command for each of the remaining host-specific certificate and key pairs. (for example, for WCCHOST1, WCCHOST2).

> **Note:**
>
> Make sure to use a unique alias for each certificate and key pair imported.

3. For Dynamic clusters, import the wildcard certificate and private key pair by using the custom id alias of `WILDCARD`.

   Example:

```
${JAVA_HOME}/bin/java utils.ImportPrivateKey \
-certfile ${KEYSTORE_HOME}/WILDCARD.example.com_cert.pem \
-keyfile ${KEYSTORE_HOME}/WILDCARD.example.com_key.pem \
-keyfilepass password \
-keystore ${KEYSTORE_HOME}/appIdentityKeyStore.jks \
-storepass password \
-alias WILDCARD \
-keypass password
```

Creating a Trust Keystore Using the Keytool Utility

To create the Trust Keystore on WCCHOST1.example.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 KEYSTORE_HOME/appTrustKeyStore.jks
```

2. Use the keytool utility to change the default password.

   The default password for the standard Java keystore is `changeit`. Oracle recommends that you always change the default password, as follows:

```
keytool -storepasswd -new NewPassword -keystore TrustKeyStore -storepass
Original_Password
```

   For example:

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

3. Import the CA certificate into the `appTrustKeyStore` by using the keytool utility.

   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.

   Use the following syntax to import the certificate:

```
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 password
```

Importing the Load Balancer Certificate into the Truststore

For the SSL handshake to act properly, the load balancer's certificate must be added to the WLS servers truststore. To add a load balancer's certificate:

1. Access the site on SSL with a browser (this adds the server's certificate to the browser's repository).
2. From the browser's certificate management tool, export the certificate to a file that is on the server's file system (with a file name such as `soa.example.com.crt`).
3. Use the keytool to import the load balancer's certificate into the truststore:

   For example:

   ```
   keytool -import -file /oracle/certificates/soa1.cz.example.com -v -keystore
   appTrustKeyStore.jks -alias aliasSOA -storepass password
   keytool -import -file /oracle/certificates/soa1-osb.cz.example.com.crt -v -
   keystore appTrustKeyStore.jks -alias aliasOSB -storepass password
   ```

> **Note:**
>
> The need to add the load balancer certificate to the WLS server truststore applies only to self-signed certificates. If the load balancer certificate is issued by a third-party CA, you have to import the public certificates of the root and the intermediate CAs into the truststore.

Adding the Updated Trust Store to the Oracle WebLogic Server Start Scripts

The `setDomainEnv.sh` script is provided by Oracle WebLogic Server and is used to start the Administration Server and the Managed Servers in the domain. To ensure that each server accesses the updated trust store, edit the `setDomainEnv.sh` script in each of the domain home directories in the enterprise deployment.

1. Log in to WCCHOST1 and open the following file with a text editor:

   ASERVER_HOME`/bin/setDomainEnv.sh`

2. Replace reference to the existing `DemoTrust.jks` entry with the following entry:

> **Note:**
>
> All the values for `EXTRA_JAVA_PROPERTIES` must be on one line in the file, followed by the export command on a new line.

```
EXTRA_JAVA_PROPERTIES="-Djavax.net.ssl.trustStore=/u01/oracle/config/
keystores/appTrustKeyStore.jks ${EXTRA_JAVA_PROPERTIES} ......."
export EXTRA_JAVA_PROPERTIES
```

3. Make the same change to the `setDomainEnv.sh` file in the *`MSERVER_HOME`*`/bin` directory WCCHOST1, WCCHOST2.

> **Note:**
>
> The `setDomainEnv.sh` file cannot be copied between *`ASERVER_HOME`*`/bin` and *`MSERVER_HOME`*`/bin` as there are differences in the files for these two domain home locations. The*`MSERVER_HOME`*`/bin/setDomainEnv.sh` file can be copied between hosts.
>
> WebLogic Serverautomatically overwrites the `setDomainEnv.sh` file after each domain extension. Some patches may also replace this file. Verify your customizations to `setDomainEnv.sh` after each of these types of maintenance operations.

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` files located both in *`ASERVER_HOME`*`/nodemanager` and *`MSERVER_HOME`*`/nodemanager` directories in all nodes:

```
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` for Node Manager's listen address. For example, in the WCCHOST1 *MSERVER_HOME*, use the alias WCCHOST1 and in the *ASERVER_HOME* on WCCHOST1, use the alias *ADMINVHN* according to the steps in Creating an Identity Keystore Using the utils.ImportPrivateKey Utility.

```
Example for WCCHOST1:
KeyStores=CustomIdentityAndCustomTrust
CustomIdentityKeyStoreFileName=KEYSTORE_HOME/appIdentityKeyStore.jks
CustomIdentityKeyStorePassPhrase=password
CustomIdentityAlias=WCCHOST1
CustomIdentityPrivateKeyPassPhrase=password
```

The passphrase entries in the `nodemanager.properties` file are encrypted when you start Node Manager. For security reasons, minimize the time the entries in the `nodemanager.properties` file are left unencrypted. After you edit the file, restart Node Manager as soon as possible so that the entries are encrypted.

> **Note:**
>
> The `CustomIdentityAlias` value will need to be corrected every time the domain is extended after this configuration is performed. An unpack operation will replace the `CustomIdentityAlias` with the Administration Server's value when the domain configuration is written.

Configuring WebLogic Servers to Use the Custom Keystores

Configure the WebLogic Servers to use the custom keystores using the Oracle WebLogic Server Administration Console. Complete this procedure for the Administration Server and the Managed Servers that require access to the front end LBR on SSL.

To configure the identity and trust keystores:

1. Log in to the Administration Console, and click **Lock & Edit**.
2. In the left pane, expand **Environment**, and select **Servers**.
3. Click the name of the server for which you want to configure the identity and trust keystores.
4. Select **Configuration**, and then **Keystores**.
5. In the **Keystores** field, click **Change**, and select **Custom Identity and Custom Trust** method for storing and managing private keys/digital certificate pairs and trusted CA certificates, and click Save.
6. In the Identity section, define attributes for the identity keystore.
   - Custom Identity Keystore: Enter the fully qualified path to the identity keystore:

     KEYSTORE_HOME`/appIdentityKeyStore.jks`
   - Custom Identity Keystore Type: Leave this field blank, it defaults to JKS.
   - Custom Identity Keystore Passphrase: Enter the password Keystore_Password you provided in Creating an Identity Keystore Using the utils.ImportPrivateKey Utility

     This attribute may be optional or required depending on the type of keystore. All keystores require the passphrase in order to write to the keystore. However, some keystores do not require the passphrase to read from the keystore. WebLogic Server reads only from the keystore, so whether or not you define this property depends on the requirements of the keystore.
7. In the Trust section, define properties for the trust keystore:
   - Custom Trust Keystore: Enter the fully qualified path to the trust keystore:

     KEYSTORE_HOME`/appTrustKeyStore.jks`
   - Custom Trust Keystore Type: Leave this field blank, it defaults to JKS.
   - Custom Trust Keystore Passphrase: The password you provided as the New_Password value in `Creating a Trust Keystore Using the Keytool Utility`.

As mentioned in the previous step, this attribute may be optional or required depending on the type of keystore.

8. Click **Save**.
9. To activate these changes, in the Change Center of the Administration Console, click **Activate Changes**.
10. Click **Lock & Edit**.
11. Select **Configuration**, then **SSL**.
12. In the Private Key Alias field, enter the alias you used for the host name the managed server listens on.

    In the Private Key Passphrase and the Confirm Private Key Passphrase fields, enter the password for the keystore that you created in Creating an Identity Keystore Using the utils.ImportPrivateKey Utility
13. Click **Save**.
14. Click **Activate Changes** in the Administration Console's Change Center to make the changes take effect.
15. Restart the Administration Server.
16. Restart the Managed Servers where the keystore has been updated.

> **Note:**
>
> The fact that servers can be restarted using the Administration Console/ Node Manager is a good verification that the communication between Node Manager, Administration Server, and the managed servers is correct.

Testing Composites Using SSL Endpoints

Once SSL has been enabled, composites endpoints can be verified on SSL from Oracle Enterprise Manager FMW Control. To test an SSL endpoint, follow these steps:

1. Enter the following URL into a browser to display the Fusion Middleware Control login screen:

   ```
   http://ADMINVHN:7001/em
   ```

   In this example:

   - Replace ADMINVHN with the host name that is assigned to the ADMINVHN Virtual IP address in Identifying and Obtaining Software Distributions for an Enterprise Deployment.
   - Port `7001` is the typical port used for the Administration Server console and Fusion Middleware Control. However, you should use the actual URL that was displayed at the end of the Configuration Wizard session when you created the domain.
2. Log in to Fusion Middleware Control by using the administrative user credentials.

3. From the tree on the left, expand SOA, then click **soa-infra** (WLS_SOA1).
4. Click the **Deployed Composites** navigation tab link.
5. Click **Composite** to open the composite's dashboard view.
6. Click the **Test** button and select one of the services from drop-down.
7. In the WSDL or WADL address, replace the base URL (`http://WCCHOST1:8001`) with the front-end load balancer base url (`https://wcc.example.com:443`) keeping the URI resource path and query string intact.
8. Click **Parse WSDL or WADL**.
9. Verify that the Endpoint URL shown is SSL, and no errors are returned.
10. Test the composite. If the response is as expected for the web service, the SSL communication between the Administration Server and the Load Balancer has been configured properly.

Configuring Roles for Administration of an Enterprise Deployment

In order to manage each product effectively within a single enterprise deployment domain, you must understand which products require specific administration roles or groups, and how to add a product-specific administration role to the Enterprise Deployment Administration group.

Each enterprise deployment consists of multiple products. Some of the products have specific administration users, roles, or groups that are used to control administration access to each product.

However, for an enterprise deployment, which consists of multiple products, you can use a single LDAP-based authorization provider and a single administration user and group to control access to all aspects of the deployment. See Creating a New LDAP Authenticator and Provisioning a New Enterprise Deployment Administrator User and Group.

To be sure that you can manage each product effectively within the single enterprise deployment domain, you must understand which products require specific administration roles or groups, you must know how to add any specific product administration roles to the single, common enterprise deployment administration group, and if necessary, you must know how to add the enterprise deployment administration user to any required product-specific administration groups.

For more information, see the following topics.

- Summary of Products with Specific Administration Roles
- Adding a Product-Specific Administration Role to the Enterprise Deployment Administration Group

Summary of Products with Specific Administration Roles

The following table lists the Fusion Middleware products that have specific administration roles, which must be added to the enterprise deployment administration group (`WCCAdministrators`), which you defined in the LDAP Authorization Provider for the enterprise deployment.

Use the information in the following table and the instructions in Adding a Product-Specific Administration Role to the Enterprise Deployment Administration Group to add the required administration roles to the enterprise deployment Administration group.

Product	Application Stripe	Administration Role to be Assigned
SOA Infrastructure	soa-infra	SOAAdmin

Adding a Product-Specific Administration Role to the Enterprise Deployment Administration Group

For products that require a product-specific administration role, use the following procedure to add the role to the enterprise deployment administration group:

1. Sign-in to the Fusion Middleware Control by using the administrator's account (for example: `weblogic_wcc`), and navigate to the home page for your application.

   These are the credentials that you created when you initially configured the domain and created the Oracle WebLogic Server Administration user name (typically, `weblogic_wcc`) and password.
2. From the **WebLogic Domain** menu, select **Security**, and then **Application Roles**.
3. For each production-specific application role, select the corresponding application stripe from the **Application Stripe** drop-down menu.
4. Click Search Application Roles icon to display all the application roles available in the domain.
5. Select the row for the application role that you are adding to the enterprise deployment administration group.
6. Click the Edit icon to edit the role.
7. Click the Add icon on the Edit Application Role page.
8. In the Add Principal dialog box, select **Group** from the **Type** drop-down menu.
9. Search for the enterprise deployment administrators group, by entering the group name (for example, `WCCAdministrators`) in the **Principal Name Starts With** field and clicking the right arrow to start the search.
10. Select the administrator group in the search results and click **OK**.
11. Click **OK** on the Edit Application Role page.

Using Persistent Stores for TLOGs and JMS in an Enterprise Deployment

The persistent store provides a built-in, high-performance storage solution for WebLogic Server subsystems and services that require persistence.

For example, the JMS subsystem stores persistent JMS messages and durable subscribers, and the JTA Transaction Log (TLOG) stores information about the committed transactions that are coordinated by the server but may not have been completed. The persistent store supports persistence to a file-based store or to a JDBC-enabled database. Persistent stores' high availability is provided by server or service migration. Server or service migration requires that all members of a WebLogic cluster have access to the same transaction and JMS persistent stores (regardless of whether the persistent store is file-based or database-based).

For an enterprise deployment, Oracle recommends using JDBC persistent stores for transaction logs (TLOGs) and JMS.

This section analyzes the benefits of using JDBC versus File persistent stores and explains the procedure for configuring the persistent stores in a supported database. If you want to use File persistent stores instead of JDBC stores, the procedure for configuring them is also explained in this section.

Products and Components that use JMS Persistence Stores and TLOGs

Determining which installed FMW products and components utilize persistent stores can be done through the WebLogic Server Console in the Domain Structure navigation under ***DomainName*** > **Services** > **Persistent Stores**. The list indicates the name of the store, the store type (FileStore and JDBC), and the target of the store. The stores listed that pertain to MDS are outside the scope of this chapter and should not be considered.

These components (as applicable) use stores by default:

Component/Product	JMS Stores	TLOG Stores
B2B	Yes	Yes
BAM	Yes	Yes
BPM	Yes	Yes
ESS	No	No
HC	Yes	Yes
Insight	Yes	Yes
MFT	Yes	Yes
OSB	Yes	Yes
SOA	Yes	Yes
WSM	No	No

Component/Product	JMS Stores	TLOG Stores
OAM	No	No

Component/Product	JMS Stores	TLOG Stores
OIM	Yes	Yes

JDBC Persistent Stores vs. File Persistent Stores

Oracle Fusion Middleware supports both database-based and file-based persistent stores for Oracle WebLogic Server transaction logs (TLOGs) and JMS. Before you decide on a persistent store strategy for your environment, consider the advantages and disadvantages of each approach.

> **Note:**
>
> Regardless of which storage method you choose, Oracle recommends that for transaction integrity and consistency, you use the same type of store for both JMS and TLOGs.

- About JDBC Persistent Stores for JMS and TLOGs
- Performance Considerations for TLOGs and JMS Persistent Stores

About JDBC Persistent Stores for JMS and TLOGs

When you store your TLOGs and JMS data in an Oracle database, you can take advantage of the replication and high availability features of the database. For example, you can use Oracle Data Guard to simplify cross-site synchronization. This is especially important if you are deploying Oracle Fusion Middleware in a disaster recovery configuration.

Storing TLOGs and JMS data in a database also means that you do not have to identity a specific shared storage location for this data. Note, however, that shared storage is still required for other aspects of an enterprise deployment. For example, it is necessary for Administration Server configuration (to support Administration Server failover), for deployment plans, and for adapter artifacts, such as the File and FTP Adapter control and processed files.

If you are storing TLOGs and JMS stores on a shared storage device, then you can protect this data by using the appropriate replication and backup strategy to guarantee zero data loss, and you potentially realize better system performance. However, the file system protection is always inferior to the protection provided by an Oracle Database.

For more information about the potential performance impact of using a database-based TLOGs and JMS store, see Performance Considerations for TLOGs and JMS Persistent Stores.

Performance Considerations for TLOGs and JMS Persistent Stores

One of the primary considerations when you select a storage method for Transaction Logs and JMS persistent stores is the potential impact on performance. This topic

provides some guidelines and details to help you determine the performance impact of using JDBC persistent stores for TLOGs and JMS.

Performance Impact of Transaction Logs Versus JMS Stores

For transaction logs, the impact of using a JDBC store is relatively small, because the logs are very transient in nature. Typically, the effect is minimal when compared to other database operations in the system.

On the other hand, JMS database stores can have a higher impact on performance if the application is JMS intensive.

Factors that Affect Performance

There are multiple factors that can affect the performance of a system when it is using JMS DB stores for custom destinations. The main ones are:

- Custom destinations involved and their type
- Payloads being persisted
- Concurrency on the SOA system (producers on consumers for the destinations)

Depending on the effect of each one of the above, different settings can be configured in the following areas to improve performance:

- Type of data types used for the JMS table (using raw versus lobs)
- Segment definition for the JMS table (partitions at index and table level)

Impact of JMS Topics

If your system uses Topics intensively, then as concurrency increases, the performance degradation with an Oracle RAC database will increase more than for Queues. In tests conducted by Oracle with JMS, the average performance degradation for different payload sizes and different concurrency was less than 30% for Queues. For topics, the impact was more than 40%. Consider the importance of these destinations from the recovery perspective when deciding whether to use database stores.

Impact of Data Type and Payload Size

When you choose to use the RAW or SecureFiles LOB data type for the payloads, consider the size of the payload being persisted. For example, when payload sizes range between 100b and 20k, then the amount of database time required by SecureFiles LOB is slightly higher than for the RAW data type.

More specifically, when the payload size reach around 4k, then SecureFiles tend to require more database time. This is because 4k is where writes move out-of-row. At around 20k payload size, SecureFiles data starts being more efficient. When payload sizes increase to more than 20k, then the database time becomes worse for payloads set to the RAW data type.

One additional advantage for SecureFiles is that the database time incurred stabilizes with payload increases starting at 500k. In other words, at that point it is not relevant (for SecureFiles) whether the data is storing 500k, 1MB or 2MB payloads, because the write is asynchronized, and the contention is the same in all cases.

The effect of concurrency (producers and consumers) on the queue's throughput is similar for both RAW and SecureFiles until the payload sizes reach 50K. For small

payloads, the effect on varying concurrency is practically the same, with slightly better scalability for RAW. Scalability is better for SecureFiles when the payloads are above 50k.

Impact of Concurrency, Worker Threads, and Database Partioning

Concurrency and worker threads defined for the persistent store can cause contention in the RAC database at the index and global cache level. Using a reverse index when enabling multiple worker threads in one single server or using multiple Oracle WebLogic Server clusters can improve things. However, if the Oracle Database partitioning option is available, then global hash partition for indexes should be used instead. This reduces the contention on the index and the global cache buffer waits, which in turn improves the response time of the application. Partitioning works well in all cases, some of which will not see significant improvements with a reverse index.

Using JDBC Persistent Stores for TLOGs and JMS in an Enterprise Deployment

This section explains the guidelines to use JDBC persistent stores for transaction logs (TLOGs) and JMS. It also explains the procedures to configure the persistent stores in a supported database.

- Recommendations for TLOGs and JMS Datasource Consolidation
  To accomplish data source consolidation and connection usage reduction, use a single connection pool for both JMS and TLOGs persistent stores.
- Roadmap for Configuring a JDBC Persistent Store for TLOGs
  The following topics describe how to configure a database-based persistent store for transaction logs.
- Roadmap for Configuring a JDBC Persistent Store for JMS
  The following topics describe how to configure a database-based persistent store for JMS.
- Creating a User and Tablespace for TLOGs
  Before you can create a database-based persistent store for transaction logs, you must create a user and tablespace in a supported database.
- Creating a User and Tablespace for JMS
  Before you can create a database-based persistent store for JMS, you must create a user and tablespace in a supported database.
- Creating GridLink Data Sources for TLOGs and JMS Stores
  Before you can configure database-based persistent stores for JMS and TLOGs, you must create two data sources: one for the TLOGs persistent store and one for the JMS persistent store.
- Assigning the TLOGs JDBC Store to the Managed Servers
  After you create the tablespace and user in the database, and you have created the datasource, you can then assign the TLOGs persistence store to each of the required Managed Servers.
- Creating a JDBC JMS Store
  After you create the JMS persistent store user and table space in the database, and after you create the data source for the JMS persistent store, you can then use the Administration Console to create the store.

- Assigning the JMS JDBC store to the JMS Servers
  After you create the JMS tablespace and user in the database, create the JMS datasource, and create the JDBC store, then you can assign the JMS persistence store to each of the required JMS Servers.
- Creating the Required Tables for the JMS JDBC Store
  The final step in using a JDBC persistent store for JMS is to create the required JDBC store tables. Perform this task before you restart the Managed Servers in the domain.

Recommendations for TLOGs and JMS Datasource Consolidation

To accomplish data source consolidation and connection usage reduction, use a single connection pool for both JMS and TLOGs persistent stores.

Oracle recommends you to reuse the `WLSSchemaDatasource` as is for TLOGs and JMS persistent stores under non-high workloads and consider increasing the `WLSSchemaDatasource` pool size. Reuse of datasource forces to use the same schema and tablespaces, and so the `PREFIX_WLS_RUNTIME` schema in the `PREFIX_WLS` tablespace is used for both TLOGs and JMS messages.

High stress (related with high JMS activity, for example) and contention in the datasource can cause stability and performance problems. For example:

- High contention in the DataSource can cause persistent stores to fail if no connections are available in the pool to persist JMS messages.
- High Contention in the DataSource can cause issues in transactions if no connections are available in the pool to update transaction logs.

For these cases, use a separate datasource for TLOGs and stores and a separate datasource for the different stores. You can still reuse the `PREFIX_WLS_RUNTIME` schema but configure separate custom datasources to the same schema to solve the contention issue.

Roadmap for Configuring a JDBC Persistent Store for TLOGs

The following topics describe how to configure a database-based persistent store for transaction logs.

1. Creating a User and Tablespace for TLOGs
2. Creating GridLink Data Sources for TLOGs and JMS Stores
3. Assigning the TLOGs JDBC Store to the Managed Servers

> **Note:**
>
> Steps 1 and 2 are optional. To accomplish data source consolidation and connection usage reduction, you can reuse `PREFIX_WLS` tablespace and `WLSSchemaDatasource` as described in Recommendations for TLOGs and JMS Datasource Consolidation.

Roadmap for Configuring a JDBC Persistent Store for JMS

The following topics describe how to configure a database-based persistent store for JMS.

1. Creating a User and Tablespace for JMS
2. Creating GridLink Data Sources for TLOGs and JMS Stores
3. Creating a JDBC JMS Store
4. Assigning the JMS JDBC store to the JMS Servers
5. Creating the Required Tables for the JMS JDBC Store

> **Note:**
>
> Steps 1 and 2 are optional. To accomplish data source consolidation and connection usage reduction, you can reuse `PREFIX_WLS` tablespace and `WLSSchemaDatasource` as described in Recommendations for TLOGs and JMS Datasource Consolidation.

Creating a User and Tablespace for TLOGs

Before you can create a database-based persistent store for transaction logs, you must create a user and tablespace in a supported database.

1. Create a tablespace called `tlogs`.

   For example, log in to SQL*Plus as the `sysdba` user and run the following command:

   ```
   SQL> create tablespace tlogs
           logging datafile 'path-to-data-file-or-+asmvolume'
           size 32m autoextend on next 32m maxsize 2048m extent management local;
   ```

2. Create a user named `TLOGS` and assign to it the `tlogs` tablespace.

   For example:

   ```
   SQL> create user TLOGS identified by password;

   SQL> grant create table to TLOGS;

   SQL> grant create session to TLOGS;

   SQL> alter user TLOGS default tablespace tlogs;

   SQL> alter user TLOGS quota unlimited on tlogs;
   ```

Creating a User and Tablespace for JMS

Before you can create a database-based persistent store for JMS, you must create a user and tablespace in a supported database.

1. Create a tablespace called `jms`.

   For example, log in to SQL*Plus as the `sysdba` user and run the following command:

   ```
   SQL> create tablespace jms
           logging datafile 'path-to-data-file-or-+asmvolume'
           size 32m autoextend on next 32m maxsize 2048m extent management local;
   ```

2. Create a user named `JMS` and assign to it the `jms` tablespace.

   For example:

   ```
   SQL> create user JMS identified by password;

   SQL> grant create table to JMS;

   SQL> grant create session to JMS;

   SQL> alter user JMS default tablespace jms;

   SQL> alter user JMS quota unlimited on jms;
   ```

Creating GridLink Data Sources for TLOGs and JMS Stores

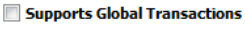
Before you can configure database-based persistent stores for JMS and TLOGs, you must create two data sources: one for the TLOGs persistent store and one for the JMS persistent store.

For an enterprise deployment, you should use GridLink data sources for your TLOGs and JMS stores. To create a GridLink data source:

1. Sign in to the Oracle WebLogic Server Administration Console.
2. If you have not already done so, in the **Change Center**, click **Lock & Edit**.
3. In the **Domain Structure** tree, expand **Services**, then select **Data Sources**.
4. On the Summary of Data Sources page, click **New** and select **GridLink Data Source**, and enter the following:
   - Enter a logical name for the data source in the **Name** field.

     For the TLOGs store, enter TLOG; for the JMS store, enter JMS.
   - Enter a name for **JNDI**.

     For the TLOGs store, enter `jdbc/tlogs`; for the JMS store, enter `jdbc/jms`.
   - For the Database Driver, select **Oracle's Driver (Thin) for GridLink Connections Versions: Any**.
   - Click **Next**.
5. In the Transaction Options page, clear the **Supports Global Transactions** check box, and then click **Next**.



6. In the GridLink Data Source Connection Properties Options screen, select **Enter individual listener information** and click **Next**.

7. Enter the following connection properties:

   - **Service Name**: Enter the service name of the database with lowercase characters. For a GridLink data source, you must enter the Oracle RAC service name. For example:

     ```
     wccedg.example.com
     ```

   - **Host Name and Port**: Enter the SCAN address and port for the RAC database, separated by a colon. For example:

     ```
     db-scan.example.com:1521
     ```

     Click **Add** to add the host name and port to the list box below the field.

Figure 18-1 Adding Host Name and Port Details for the RAC Database

You can identify the SCAN address by querying the appropriate parameter in the database using the TCP Protocol:

```
SQL>show parameter remote_listener;

NAME                 TYPE        VALUE

-------------------------------------------------

remote_listener     string      db-scan.example.com
```

> **Note:**
>
> For Oracle Database 11*g* Release 1 (11.1), use the virtual IP and port of each database instance listener, for example:
>
> ```
> dbhost1-vip.example.com (port 1521)
> ```
>
> and
>
> ```
> dbhost2-vip.example.com (1521)
> ```

   - **Database User Name**: Enter the following:

     For the TLOGs store, enter `TLOGS`; for the JMS persistent store, enter `JMS`.

   - **Password**: Enter the password that you used when you created the user in the database.

- **Confirm Password**: Enter the password again and click **Next**.

8. On the Test GridLink Database Connection page, review the connection parameters and click **Test All Listeners**.

   Here is an example of a successful connection notification:

   ```
   Connection test
   for jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)
   (HOST=db-scan.example.com)
   (PORT=1521)))(CONNECT_DATA=(SERVICE_NAME=wccedg.example.com))) succeeded.
   ```

   Click **Next**.

9. In the ONS Client Configuration page, do the following:

   - Select **FAN Enabled** to subscribe to and process Oracle FAN events.
   - Enter the SCAN address: ONS remote port for the RAC database and the ONS remote port as reported by the database (see the following example) and click **Add**:

     ```
     [orcl@db-scan1 ~]$ srvctl config nodeapps -s

     ONS exists: Local port 6100, remote port 6200, EM port 2016
     ```

   - Click **Next**.

   > **Note:**
   >
   > For Oracle Database 11*g* Release 1 (11.1), use the hostname and port of each database's ONS service, for example:
   >
   > ```
   > custdbhost1.example.com (port 6200)
   > ```
   >
   > and
   >
   > ```
   > custdbhost2.example.com (6200)
   > ```

10. On the Test ONS Client Configuration page, review the connection parameters and click **Test All ONS Nodes**.

    Here is an example of a successful connection notification:

    ```
    Connection test for db-scan.example.com:6200 succeeded.
    ```

    Click **Next**.

11. In the Select Targets page, select the cluster that is using the persistent store, and then select **All Servers in the cluster**.
12. Click **Finish**.
13. To activate these changes, in the Change Center of the Administration Console, click **Activate Changes**.
14. Repeat step 4 through step 13 to create the GridLink Data Source for JMS File Stores.

Assigning the TLOGs JDBC Store to the Managed Servers

After you create the tablespace and user in the database, and you have created the datasource, you can then assign the TLOGs persistence store to each of the required Managed Servers.

1. Login in to the Oracle WebLogic Server Administration Console.
2. In the **Change Center**, click **Lock and Edit**.
3. In the Domain Structure tree, expand **Environment**, then **Servers**.
4. Click the name of the Managed Server you want to use the TLOGs store.
5. Select the **Configuration** > **Services** tab.
6. Under **Transaction Log Store**, select **JDBC** from the **Type** menu.
7. From the **Data Source** menu, select the data source you created for the TLOGs persistence store.
8. In the **Prefix Name** field, specify a prefix name to form a unique JDBC TLOG store name for each configured JDBC TLOG store
9. Click **Save**.
10. Repeat steps 3 to 7 for each of the additional Managed Servers in the cluster.
11. To activate these changes, in the Change Center of the Administration Console, click **Activate Changes**.

Creating a JDBC JMS Store

After you create the JMS persistent store user and table space in the database, and after you create the data source for the JMS persistent store, you can then use the Administration Console to create the store.

1. Log in to the Oracle WebLogic Server Administration Console.
2. If you have not already done so, in the **Change Center**, click **Lock & Edit**.
3. In the **Domain Structure** tree, expand **Services**, then select **Persistent Store**.
4. Click **New**, and then click **JDBC Store**.
5. Enter a persistent store name that easily relates it to the pertaining JMS servers that will be using it.
6. Specify a unique prefix qualifying the installation and cluster and associate it with the data source you created for the JMS persistent store.
7. Target the store to the entity that will host the JTA services.
8. To activate these changes, in the Change Center of the Administration Console, click **Activate Changes**.

Assigning the JMS JDBC store to the JMS Servers

After you create the JMS tablespace and user in the database, create the JMS datasource, and create the JDBC store, then you can assign the JMS persistence store to each of the required JMS Servers.

To assign the JMS persistence store to the JMS servers:

1. Log in to the Oracle WebLogic Server Administration Console.
2. In the **Change Center**, click **Lock and Edit**.
3. In the Domain Structure tree, expand **Services**, then **Messaging**, and then **JMS Servers**.
4. Click the name of the JMS Server that you want to use the persistent store.
5. From the **Persistent Store** menu, select the JMS persistent store you created earlier.
6. Click **Save**.
7. Repeat steps 3 to 6 for each of the additional JMS Servers in the cluster.
8. To activate these changes, in the Change Center of the Administration Console, click **Activate Changes**.

Creating the Required Tables for the JMS JDBC Store

The final step in using a JDBC persistent store for JMS is to create the required JDBC store tables. Perform this task before you restart the Managed Servers in the domain.

1. Review the information in Performance Considerations for TLOGs and JMS Persistent Stores, and decide which table features are appropriate for your environment.

   There are three Oracle DB schema definitions provided in this release and were extracted for review in the previous step. The basic definition includes the RAW data type without any partition for indexes. The second uses the blob data type, and the third uses the blob data type and secure files.

2. Create a domain-specific well-named folder structure for the custom `DDL` file on shared storage. The *ORACLE_RUNTIME* shared volume is recommended so it is available to all servers.

   Example:

   ```
   mkdir -p ORACLE_RUNTIME/domain_name/ddl
   ```

3. Create a `jms_custom.ddl` file in new shared `ddl` folder based on your requirements analysis.

   For example, to implement an optimized schema definition that uses both secure files and hash partitioning, create the `jms_custom.ddl` file with the following content:

   ```
   CREATE TABLE $TABLE (
     id     int  not null,
     type   int  not null,
     handle int  not null,
   ```

```
  record blob not null,
PRIMARY KEY (ID) USING INDEX GLOBAL PARTITION BY HASH (ID)
PARTITIONS 8)
LOB (RECORD) STORE AS SECUREFILE (ENABLE STORAGE IN ROW);
```

This example can be compared to the default schema definition for JMS stores, where the RAW data type is used without any partitions for indexes.

Note that the number of partitions should be a power of two. This ensures that each partition is of similar size. The recommended number of partitions varies depending on the expected table or index growth. You should have your database administrator (DBA) analyze the growth of the tables over time and adjust the tables accordingly. See Partitioning Concepts in *Database VLDB and Partitioning Guide*.

4. Use the Administration Console to edit the existing JDBC Store you created earlier; create the table that is used for the JMS data:
   a. Login in to the Oracle WebLogic Server Administration Console.
   b. In the **Change Center**, click **Lock and Edit**.
   c. In the Domain Structure tree, expand **Services**, then **Persistent Stores**.
   d. Click the persistent store you created earlier.
   e. Under the **Advanced** options, enter *`ORACLE_RUNTIME`*`/`*`domain_name`*`/ddl/jms_custom.ddl` in the **Create Table from DDL File** field.
   f. Click **Save**.
   g. To activate these changes, in the Change Center of the Administration Console, click **Activate Changes**.
5. Restart the Managed Servers.

Using File Persistent Stores for TLOGs and JMS in an Enterprise Deployment

This section explains the procedures to configure TLOGs and JMS File persistent stores in a shared folder.

- Configuring TLOGs File Persistent Store in a Shared Folder
- Configuring JMS File Persistent Store in a Shared Folder

Configuring TLOGs File Persistent Store in a Shared Folder

Oracle WebLogic Server uses the transaction logs to recover from system crashes or network failures.

- Configuring TLOGs File Persistent Store in a Shared Folder with a Static Cluster
- Configuring TLOGs File Persistent Store in a Shared Folder with a Dynamic Cluster
- Validating the Location and Creation of the Transaction Logs

Configuring TLOGs File Persistent Store in a Shared Folder with a Static Cluster

To set the location for the default persistence stores for each managed server in a static cluster, complete the following steps:

1. Log into the Oracle WebLogic Server Administration console:

   `ADMINVHN:7001/console`

   > **Note:**
   >
   > If you have already configured web tier, use `http://admin.example.com/console`.

2. In the Change Center section, click **Lock & Edit**.
3. For each of the Managed Servers in the cluster:
   a. In the Domain Structure window, expand the **Environment** node, and then click the **Servers** node.

      The Summary of Servers page appears.

   b. Click the name of the server (represented as a hyperlink) in the **Name** column of the table.

      The settings page for the selected server appears and defaults to the Configuration tab.

   c. On the **Configuration** tab, click the **Services** tab.

   d. In the Default Store section of the page, enter the path to the folder where the default persistent stores stores its data files.

      For the enterprise deployment, use the *ORACLE_RUNTIME* directory location. This subdirectory serves as the central, shared location for transaction logs for the cluster. See File System and Directory Variables Used in This Guide.

      For example:

      `ORACLE_RUNTIME/domain_name/cluster_name/tlogs`

      In this example, replace *ORACLE_RUNTIME* with the value of the variable for your environment. Replace *domain_name* with the name you assigned to the domain. Replace *cluster_name* with the name of the cluster you just created.

   e. Click **Save**.
4. Complete step 3 for all servers in the SOA_Cluster.
5. Click **Activate Changes**.

> **Note:**
>
> You validate the location and the creation of the transaction logs later in the configuration procedure.

Configuring TLOGs File Persistent Store in a Shared Folder with a Dynamic Cluster

To set the location for the default persistence stores for a dynamic cluster, update the server template:

1. Log into the Oracle WebLogic Server Administration Console:

   `ADMINVHN:7001/console`

   > **Note:**
   >
   > If you have already configured web tier, use `http://admin.example.com/console`.

2. In the Change Center section, click **Lock & Edit**.
3. Navigate to the server template for the cluster:
   a. In the Domain Structure window, expand the **Environment and Clusters** nodes, and then click the **Server Templates** node.

      The Summary of Server Templates page appears.

   b. Click the name of the server template (represented as a hyperlink) in the **Name** column of the table.

      The settings page for the selected server template appears and defaults to the **Configuration** tab.

   c. On the **Configuration** tab, click the **Services** tab.
   d. In the Default Store section of the page, enter the path to the folder where the default persistent stores stores its data files.

      For the enterprise deployment, use the *ORACLE_RUNTIME* directory location. This subdirectory serves as the central, shared location for transaction logs for the cluster. See File System and Directory Variables Used in This Guide.

      For example:

      `ORACLE_RUNTIME/domain_name/cluster_name/tlogs`

      In this example, replace *ORACLE_RUNTIME* with the value of the variable for your environment. Replace *domain_name* with the name that you assigned to the domain. Replace *cluster_name* with the name of the cluster you just created.

   e. Click **Save**.
4. Click **Activate Changes**.

> **Note:**
>
> You validate the location and the creation of the transaction logs later in the configuration procedure.

Validating the Location and Creation of the Transaction Logs

After the WLS_SERVER_TYPE1 and WLS_SERVER_TYPE2 managed Servers are up and running, verify that the transaction log directory and transaction logs are created as expected, based on the steps that you performed in Configuring TLOGs File Persistent Store in a Shared Folder with a Static Cluster and Configuring TLOGs File Persistent Store in a Shared Folder with a Dynamic Cluster:

ORACLE_RUNTIME/domain_name/OSB_Cluster/tlogs

- _WLS_WLS_SERVER_TYPE1000000.DAT
- _WLS_WLS_SERVER_TYPE2000000.DAT

Configuring JMS File Persistent Store in a Shared Folder

If you have already configured and extended your domain, the JMS Persistent Files are already configured in a shared location. If you need to change any other persistent store file to the shared folder, perform the following steps:

1. Log in to the Oracle WebLogic Server Administration Console.
2. Navigate to **Domain > Services > Persistent Store** and click the name of the persistent store that you want to move to the shared folder.

   The **Configuration: General** tab is displayed.
3. Change the directory to ORACLE_RUNTIME/domain_name/WCC_cluster/jms.
4. Click **Save**.
5. Click **Activate Changes**.

Performing Backups and Recoveries for an Enterprise Deployment

It is recommended that you follow the below mentioned guidelines to make sure that you back up the necessary directories and configuration data for an Oracle WebCenter Content enterprise deployment.

> **Note:**
>
> Some of the static and runtime artifacts listed in this section are hosted from Network Attached Storage (NAS). If possible, backup and recover these volumes from the NAS filer directly rather than from the application servers.

For general information about backing up and recovering Oracle Fusion Middleware products, see the following sections in *Oracle Fusion Middleware Administering Oracle Fusion Middleware*:

- Backing Up Your Environment
- Recovering Your Environment

Table 18-1 lists the static artifacts to back up in a typical Oracle WebCenter Content enterprise deployment.

Table 18-1 Static Artifacts to Back Up in the Oracle WebCenter Content Enterprise Deployment

Type	Host	Tier
Database Oracle home	DBHOST1 and DBHOST2	Data Tier
Oracle Fusion Middleware Oracle home	WEBHOST1 and WEBHOST2	Web Tier
Oracle Fusion Middleware Oracle home	WCCHOST1 and WCCHOST2 (or NAS Filer)	Application Tier
Installation-related files	WEBHOST1, WEHOST2, and shared storage	N/A

Table 18-2 lists the runtime artifacts to back up in a typical Oracle WebCenter Content enterprise deployment.

Table 18-2 Run-Time Artifacts to Back Up in the Oracle WebCenter Content Enterprise Deployment

Type	Host	Tier
Administration Server domain home (ASERVER_HOME)	WCCHOST1 (or NAS Filer)	Application Tier
Application home (APPLICATION_HOME)	WCCHOST1 (or NAS Filer)	Application Tier
Oracle RAC databases	DBHOST1 and DBHOST2	Data Tier
Scripts and Customizations	Per host	Application Tier
Deployment Plan home (DEPLOY_PLAN_HOME)	WCCHOST1 (or NAS Filer)	Application Tier
OHS/OTD Configuration directory	WEBHOST1 and WEBHOST2	Web Tier

Modifying the Upload and Stage Directories to an Absolute Path in an Enterprise Deployment

After you configure the domain and unpack it to the Managed Server domain directories on all the hosts, verify and update the upload and stage directories for Managed Servers in the new clusters. Also, update the upload directory for the AdminServer to have the same absolute path instead of relative, otherwise deployment issues can occur. If you implement dynamic clusters, the configuration of the server template assigned to each newly added cluster should be verified and updated, otherwise, verify and update every statically-defined Managed Server for the newly added clusters.

> **Note:**
>
> This option is applicable only for static clusters.

This step is necessary to avoid potential issues when you perform remote deployments and for deployments that require the stage mode.

To update the directory paths for the Deployment Stage and Upload locations, complete the following steps:

1. Log in to the Oracle WebLogic Server Administration Console.
2. In the left navigation tree, expand **Domain**, and then **Environment**.
3. Click **Lock & Edit**.
4. Navigate to and edit the appropriate objects for your cluster type.
   a. For Static Clusters, navigate to **Servers** and click the name of the Managed Server you want to edit.
   b. For Dynamic Clusters, navigate to **Clusters** > **Server Templates**, and click on the name of the server template to be edited.
5. For each new Managed Server or Server Template to be edited:
   a. Click the **Configuration** tab, and then click the **Deployment** tab.
   b. Verify that the **Staging Directory Name** is set to the following:

      `MSERVER_HOME/servers/server_or_template_name/stage`

      Replace *MSERVER_HOME* with the full path for the *MSERVER_HOME* directory.

      If you use static clusters, update with the correct name of the Managed Server that you are editing.

      If you use dynamic clusters, leave the template name intact.
      For example: `/u02/oracle/config/domains/wccedg_domain/servers/XYZ-server-template/stage`
   c. Update the **Upload Directory Name** to the following value:

      `ASERVER_HOME/servers/AdminServer/upload`

      Replace *ASERVER_HOME* with the directory path for the *ASERVER_HOME* directory.
   d. Click **Save**.
   e. Return to the Summary of Servers or Summary of Server Templates screen as applicable.
6. Repeat the previous steps for each of the new managed servers or dynamic cluster server templates.
7. Navigate to and update the Upload Directory Name value for the AdminServer:
   a. Navigate to **Servers**, and select the AdminServer.
   b. Click the **Configuration** tab, and then click the **Deployment** Tab.
   c. Verify that the **Staging Directory Name** is set to the following absolute path:

      `ASERVER_HOME/servers/AdminServer/stage`
   d. Update the **Upload Directory Name** to the following absolute path:

      `ASERVER_HOME/servers/AdminServer/upload`

      Replace *ASERVER_HOME* with the directory path for the *ASERVER_HOME* directory.

e. Click **Save**.

8. When you have modified all the appropriate objects, click **Activate Changes**.
9. Restart all Managed Servers for the changes to take effect.

> **Note:**
>
> If you continue directly with further domain configurations, a restart to enable the stage and upload directory changes is not strictly necessary at this time.

Starting and Stopping Servers in Dynamic Clusters

You can start and stop server instances in dynamic clusters by using the same methods you use to start and stop server instances in configured static clusters.

Methods to start and stop server instances in configured clusters:

- WebLogic Server Administration Console
- Fusion Middleware Control
- WLST start and shutdown commands
- Node Manager
- Start scripts

Depending on which startup method you choose and the tasks you have already performed, you may have to follow several other procedures before you can start server instances. See Starting and Stopping Servers section in *Oracle Fusion Middleware Administering Server Startup and Shutdown for Oracle WebLogic Server*.

> **Note:**
>
> Before you begin, ensure that WebLogic Server is installed on all hosts where you want to run your server instances. If you want to use Node Manager to start and stop your server instances, then you must also run Node Manager on these hosts.

Expanding or Reducing Dynamic Clusters

When you create a dynamic cluster, WebLogic Server generates the number of dynamic servers you specify. Before you decide upon the number of server instances, ensure you have the performance capacity to handle the desired number.

The number of dynamic server instances available are based on the configured maximum specified in the server template for a given dynamic cluster. Transient changes in capacity requirements can be easily met by starting or stopping some of the available managed servers within the cluster, keeping in mind that a minimum of two or three are required to maintain high-availability.

If you need additional server capacity on top of the number of server instances you originally specified, you can increase the maximum number of dynamic servers in

the dynamic cluster configuration. To reduce the number of server instances in the dynamic cluster, decrease the value of the maximum number of dynamic servers attribute. Before lowering this value, shut down the server instances you plan to remove.

You can also use the WLST `scaleUp` and `scaleDown` commands to manage your dynamic cluster. To increase the number of dynamic servers in the dynamic cluster, use the `scaleUp` command and enable the `updateConfiguration` argument. WLST will increase the maximum size of the cluster by the specified number of servers and start the server instances.

The `scaleUp` command increases the number of running servers for the specified dynamic cluster. The non-running server instance with the lowest server ID starts first, followed by the next highest non-running server ID, until the specified number of server instances is started.

You can start one, all, or any number of server instances in the dynamic cluster by specifying the desired number with the `numServers` argument in the `scaleUp` command. If all available server instances are already running, the `scaleUp` command increases the size of the cluster to the minimum number of requested server instances before starting the specified number of servers.

To decrease the maximum size of the dynamic cluster, use the scaleDown command and enable the `updateConfiguration` argument. WLST will gracefully shut down the specified number of running server instances and remove them from the dynamic cluster. See scaleUp and scaleDown in *Oracle Fusion Middleware WLST Command Reference for WebLogic Server*. The `scaleDown` command gracefully shuts down the specified number of running servers. The server instance with the highest server ID shuts down first, followed by the next highest ID, until the specified number of server instances is shut down.

> **Note:**
>
> You can only use the WLST `scaleUp` and `scaleDown` commands with dynamic server instances. In a mixed cluster, containing both manually configured and dynamic server instances, the `scaleUp` and `scaleDown` commands ignore the configured servers. You must manually start and stop configured server instances in a mixed cluster.
>
> For example, a cluster contains two running dynamic servers and two non-running configured servers. If you use the `scaleUp` command, WLST adds one additional dynamic server instance to your cluster and starts the dynamic server.

The WLST `scaleUp` and `scaleDown` commands provide ways to manually scale your dynamic cluster. For automatic scaling, you can configure elasticity for your dynamic cluster. Elasticity enables a dynamic cluster to perform scaling and re-provisioning operations automatically in response to demand or on a calendar based schedule. WebLogic Server provides elasticity for dynamic clusters through the Policies and Actions system of the WebLogic Diagnostic Framework (WLDF). See Configuring Elasticity in Dynamic Clusters for Oracle WebLogic Server.

19

Using Whole Server Migration and Service Migration in an Enterprise Deployment

The Oracle WebLogic Server migration framework supports Whole Server Migration and Service Migration. The following sections explain how these features can be used in an Oracle Fusion Middleware enterprise topology.

- About Whole Server Migration and Automatic Service Migration in an Enterprise Deployment
  Oracle WebLogic Server provides a migration framework that is an integral part of any highly available environment. The following sections provide more information about how this framework can be used effectively in an enterprise deployment.
- Creating a GridLink Data Source for Leasing
  Whole Server Migration and Automatic Service Migration require a data source for the leasing table, which is a tablespace created automatically as part of the Oracle WebLogic Server schemas by the Repository Creation Utility (RCU).
- Configuring Whole Server Migration for an Enterprise Deployment
  After you have prepared your domain for whole server migration or automatic service migration, you can configure Whole Server Migration for specific Managed Servers within a cluster.
- Configuring Automatic Service Migration in an Enterprise Deployment
  You may need to configure automatic service migration for specific services in an enterprise deployment.

About Whole Server Migration and Automatic Service Migration in an Enterprise Deployment

Oracle WebLogic Server provides a migration framework that is an integral part of any highly available environment. The following sections provide more information about how this framework can be used effectively in an enterprise deployment.

- Understanding the Difference between Whole Server and Service Migration
- Implications of Using Whole Server Migration or Service Migration in an Enterprise Deployment
- Understanding Which Products and Components Require Whole Server Migration and Service Migration

Understanding the Difference between Whole Server and Service Migration

The Oracle WebLogic Server migration framework supports two distinct types of automatic migration:

- **Whole Server Migration**, where the Managed Server instance is migrated to a different physical system upon failure.

  Whole server migration provides for the automatic restart of a server instance, with all its services, on a different physical machine. When a failure occurs in a server that is part of a cluster which is configured with server migration, the server is restarted on any of the other machines that host members of the cluster.

  For this to happen, the servers must use a floating IP as listen address and the required resources (transactions logs and JMS persistent stores) must be available on the candidate machines.

  See Whole Server Migration in *Oracle Fusion Middleware Administering Clusters for Oracle WebLogic Server*.

- **Service Migration**, where specific services are moved to a different Managed Server within the cluster.

  To understand service migration, it's important to understand *pinned services*.

  In a WebLogic Server cluster, most subsystem services are hosted homogeneously on all server instances in the cluster, enabling transparent failover from one server to another. In contrast, pinned services, such as JMS-related services, the JTA Transaction Recovery Service, and user-defined singleton services, are hosted on individual server instances within a cluster—for these services, the WebLogic Server migration framework supports failure recovery with service migration, as opposed to failover.

  See Understanding the Service Migration Framework in *Oracle Fusion Middleware Administering Clusters for Oracle WebLogic Server*.

Implications of Using Whole Server Migration or Service Migration in an Enterprise Deployment

When a server or service is started in another system, the required resources (such as services data and logs) must be available to both the original system and to the failover system; otherwise, the service cannot resume the same operations successfully on the failover system.

For this reason, both whole server and service migration require that all members of the cluster have access to the same transaction and JMS persistent stores (whether the persistent store is file-based or database-based).

This is another reason why shared storage is important in an enterprise deployment. When you properly configure shared storage, you ensure that in the event of a manual failover (Administration Server failover) or an automatic failover (whole server migration or service migration), both the original machine and the failover machine can access the same file store with no change in service.

In the case of an automatic service migration, when a pinned service needs to be resumed, the JMS and JTA logs that it was using before failover need to be accessible.

In addition to shared storage, Whole Server Migration requires the procurement and assignment of a virtual IP address (VIP). When a Managed Server fails over to another machine, the VIP is automatically reassigned to the new machine.

Note that service migration does not require a VIP.

Understanding Which Products and Components Require Whole Server Migration and Service Migration

Note that the table lists the recommended best practice. It does not preclude you from using Whole Server or Automatic Server Migration for those components that support it.

Component	Whole Server Migration (WSM)	Automatic Service Migration (ASM)
Oracle WebCenter Content	YES	YES (Recommended)
Oracle SOA Suite	YES	YES (Recommended)
Oracle Enterprise Capture	YES	YES (Recommended)

Creating a GridLink Data Source for Leasing

Whole Server Migration and Automatic Service Migration require a data source for the leasing table, which is a tablespace created automatically as part of the Oracle WebLogic Server schemas by the Repository Creation Utility (RCU).

> **Note:**
>
> To accomplish data source consolidation and connection usage reduction, you can reuse the `WLSSchemaDatasource` as is for database leasing. This datasource is already configured with the `FMW1221_WLS_RUNTIME` schema, where the leasing table is stored.

For an enterprise deployment, you should create a GridLink data source:

1. Log in to the Oracle WebLogic Server Administration Console.
2. If you have not already done so, in the **Change Center**, click **Lock & Edit**.
3. In the **Domain Structure** tree, expand **Services**, then select **Data Sources**.
4. On the Summary of Data Sources page, click **New** and select **GridLink Data Source**, and enter the following:
   - Enter a logical name for the data source in the **Name** field. For example, **Leasing**.
   - Enter a name for **JNDI**. For example, **jdbc/leasing**.
   - For the Database Driver, select **Oracle's Driver (Thin) for GridLink Connections Versions: Any**.
   - Click **Next**.
5. In the Transaction Options page, clear the **Supports Global Transactions** check box, and then click **Next**.

6. In the GridLink Data Source Connection Properties Options screen, select **Enter individual listener information** and click **Next**.

7. Enter the following connection properties:

   - **Service Name**: Enter the service name of the database with lowercase characters. For a GridLink data source, you must enter the Oracle RAC service name. For example:

     ```
     wccedg.example.com
     ```

   - **Host Name and Port**: Enter the SCAN address and port for the RAC database, separated by a colon. For example:

     ```
     db-scan.example.com:1521
     ```

     Click **Add** to add the host name and port to the list box below the field.

Figure 19-1 Specifying SCAN Address for the RAC Database

You can identify the SCAN address by querying the appropriate parameter in the database using the TCP Protocol:

```
SQL>show parameter remote_listener;

NAME                 TYPE        VALUE

-----------------------------------------------------

remote_listener     string      db-scan.example.com
```

> **Note:**
>
> For Oracle Database 11*g* Release 1 (11.1), use the virtual IP and port of each database instance listener, for example:
>
> ```
> dbhost1-vip.mycompany.com (port 1521)
> ```
>
> and
>
> ```
> dbhost2-vip.mycompany.com (1521)
> ```
>
> For Oracle Database 10*g*, use multi data sources to connect to an Oracle RAC database. For information about configuring multi data sources, see Using Multi Data Sources with Oracle RAC.

   - **Database User Name**: Enter the following:

```
FMW1221_WLS_RUNTIME
```

In this example, FMW1221 is the prefix you used when you created the schemas as you prepared to configure the initial enterprise manager domain.

Note that in previous versions of Oracle Fusion Middleware, you had to manually create a user and tablespace for the migration leasing table. In Fusion Middleware 12*c* (12.2.1), the leasing table is created automatically when you create the WLS schemas with the Repository Creation Utility (RCU).

- **Password**: Enter the password you used when you created the WLS schema in RCU.
- **Confirm Password**: Enter the password again and click **Next**.

8. On the Test GridLink Database Connection page, review the connection parameters and click **Test All Listeners**.

   Here is an example of a successful connection notification:

   ```
   Connection test
   for jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)
   (HOST=db-scan.example.com)
   (PORT=1521)))(CONNECT_DATA=(SERVICE_NAME=wccedg.example.com))) succeeded.
   ```

   Click **Next**.

9. In the ONS Client Configuration page, do the following:
   - Select **FAN Enabled** to subscribe to and process Oracle FAN events.
   - Enter the SCAN address in the **ONS Host and Port** field, and then click **Add**.

     This value should be the ONS host and ONS remote port for the RAC database. To find the ONS remote port for the database, you can use the following command on the database host:

     ```
     [orcl@db-scan1 ~]$ srvctl config nodeapps -s

     ONS exists: Local port 6100, remote port 6200, EM port 2016
     ```

   - Click **Next**.

   > **Note:**
   >
   > For Oracle Database 11*g* Release 1 (11.1), use the hostname and port of each database's ONS service, for example:
   >
   > ```
   > custdbhost1.example.com (port 6200)
   > ```
   >
   > and
   >
   > ```
   > custdbhost2.example.com (6200)
   > ```

10. On the Test ONS Client Configuration page, review the connection parameters and click **Test All ONS Nodes**.

    Here is an example of a successful connection notification:

    ```
    Connection test for db-scan.example.com:6200 succeeded.
    ```

    Click **Next**.

11. In the Select Targets page, select the cluster that you are configuring for Whole Server Migration or Automatic Service Migration, and then select **All Servers in the cluster**.
12. Click **Finish**.
13. Click **Activate Changes**.

Configuring Whole Server Migration for an Enterprise Deployment

After you have prepared your domain for whole server migration or automatic service migration, you can configure Whole Server Migration for specific Managed Servers within a cluster.

> **Note:**
>
> As mentioned earlier, for migration to work, servers must use a virtual hostname that matches a floating IP, as the listen address. You can specify the listen address directly in the Configuration Wizard or update it in the administration console.

- Editing the Node Manager's Properties File to Enable Whole Server Migration
- Setting Environment and Superuser Privileges for the wlsifconfig.sh Script
- Configuring Server Migration Targets
- Testing Whole Server Migration

Editing the Node Manager's Properties File to Enable Whole Server Migration

Use the section to edit the Node Manager properties file on the two nodes where the servers are running.

1. Locate and open the following file with a text editor:

   MSERVER_HOME`/nodemanager/nodmeanager.properties`

2. If not done already, set the `StartScriptEnabled` property in the `nodemanager.properties` file to true.

   This is required to enable Node Manager to start the managed servers.

3. Add the following properties to the `nodemanager.properties` file to enable server migration to work properly:

   - `Interface`

     ```
     Interface=eth0
     ```

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

> **Note:**
>
> Do not specify the sub interface, such as `eth0:1` or `eth0:2`. This interface is to be used without the `:0`, or `:1`.
>
> The Node Manager's scripts traverse the different `:X` enabled IPs to determine which to add or remove. For example, the valid values in Linux environments are `eth0`, `eth1`, or, `eth2`, `eth3`, `ethn`, depending on the number of interfaces configured.

- `NetMask`

  ```
  NetMask=255.255.255.0
  ```

  This property specifies the net mask for the interface for the floating IP.

- `UseMACBroadcast`

  ```
  UseMACBroadcast=true
  ```

  This property specifies whether to use a node's MAC address when sending ARP packets, that is, whether to use the `-b` flag in the arping command.

4. Restart the Node Manager.
5. Verify in the output of Node Manager (the shell where the Node Manager is started) that these properties are in use. Otherwise, problems may occur during migration. The output should be similar to the following:

   ```
   ...
   SecureListener=true
   LogCount=1
   eth0=*,NetMask=255.255.255.0
   ...
   ```

Setting Environment and Superuser Privileges for the wlsifconfig.sh Script

Use this section to set the environment and superuser privileges for the `wlsifconfig.sh` script, which is used to transfer IP addresses from one machine to another during migration. It must be able to run `ifconfig`, which is generally only available to superusers.

For more information about the `wlsifconfig.sh` script, see Configuring Automatic Whole Server Migration in *Oracle Fusion Middleware Administering Clusters for Oracle WebLogic Server*.

Refer to the following sections for instructions on preparing your system to run the `wlsifconfig.sh` script.

- Setting the PATH Environment Variable for the wlsifconfig.sh Script
- Granting Privileges to the wlsifconfig.sh Script

Setting the PATH Environment Variable for the wlsifconfig.sh Script

Ensure that the commands listed in the following table are included in the PATH environment variable for each host computers.

File	Directory Location
`wlsifconfig.sh`	*`MSERVER_HOME`*`/bin/server_migration`
`wlscontrol.sh`	*`WL_HOME`*`/common/bin`
`nodemanager.domains`	*`MSERVER_HOME`*`/nodemanager`

Granting Privileges to the wlsifconfig.sh Script

Grant *sudo* privilege to the operating system user (for example, `oracle`) with no password restriction, and grant execute privilege on the `/sbin/ifconfig` and `/sbin/arping` binaries.

> **Note:**
>
> For security reasons, `sudo` should be restricted to the subset of commands required to run the `wlsifconfig.sh` script.
>
> Ask the system administrator for the *sudo* and system rights as appropriate to perform this required configuration task.

The following is an example of an entry inside */etc/sudoers* granting *sudo* execution privilege for `oracle` to run `ifconfig` and `arping`:

```
Defaults:oracle !requiretty
oracle ALL=NOPASSWD: /sbin/ifconfig,/sbin/arping
```

Configuring Server Migration Targets

To configure migration in a cluster:

1. Sign in to the Oracle WebLogic Server Administration Console.
2. In the Domain Structure window, expand **Environment** and select **Clusters**. The Summary of Clusters page is displayed.
3. Click the cluster for which you want to configure migration in the Name column of the table.
4. Click the **Migration** tab.
5. Click **Lock & Edit**.
6. Select **Database** as Migration Basis. From the drop-down list, select **Leasing** as Data Source For Automatic Migration.

7. Under **Candidate Machines For Migratable Server**, in the Available filed, select the Managed Servers in the cluster and click the right arrow to move them to **Chosen**.
8. Click **Save**.
9. Set the Candidate Machines for Server Migration. You must perform this task for all of the managed servers as follows:
    a. In Domain Structure window of the Oracle WebLogic Server Administration Console, expand **Environment** and select **Servers**.
    b. Select the server for which you want to configure migration.
    c. Click the **Migration** tab.
    d. Select **Automatic Server Migration Enabled** and click **Save**.

       This enables the Node Manager to start a failed server on the target node automatically.

       For information on targeting applications and resources, see Using Multi Data Sources with Oracle RAC.
    e. In the **Available** field, located in the Migration Configuration section, select the machines to which to allow migration and click the right arrow.

       In this step, you are identifying the host to which the Managed Server should failover if the current host is unavailable. For example, for the Managed Server on the HOST1, select HOST2; for the Managed Server on HOST2, select HOST1.

> **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 is different from the configuration if the server is migrated automatically.

10. Click **Activate Changes**.
11. Restart the Administration Server and the servers for which server migration has been configured.

Testing Whole Server Migration

Perform the steps in this section to verify that automatic whole server migration is working properly.

To test from Node 1:

1. Stop the managed server process.

```
kill -9 pid
```

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

```
ps -ef | grep WLS_WCC1
```

2. Watch the Node Manager console (the terminal window where you performed the kill command): you should see a message indicating that the managed server's floating IP has been disabled.
3. Wait for the Node Manager to try a second restart of the Managed Server. Node Manager waits for a period of 30 seconds before trying this restart.
4. After node manager restarts the server and before it reaches *Running* state, kill the associated process again.

   Node Manager should log a message indicating that the server will not be restarted again locally.

> **Note:**
>
> The number of restarts required is determined by the `RestartMax` parameter in the following configuration file:
>
> *`MSERVER_HOME`*`/servers/WLS_WCC1/data/nodemanager/startup.properties`
>
> The default value is `RestartMax=2`.

To test from Node 2:

1. Watch the local Node Manager console. After 30 seconds since the last try to restart the managed server on Node 1, Node Manager on Node 2 should prompt that the floating IP for the managed server is being brought up and that the server is being restarted in this node.
2. Access a product URL by using the same IP address. If the URL is successful, then the migration was successful.

Verification From the Administration Console

You can also verify migration using the Oracle WebLogic Server Administration Console:

1. Log in to the Administration Console.
2. Click **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.

> **Note:**
>
> After a server is migrated, to fail it back to its original machine, stop the managed server from the Oracle WebLogic Administration Console and then start it again. The appropriate Node Manager starts the managed server on the machine to which it was originally assigned.

Configuring Automatic Service Migration in an Enterprise Deployment

You may need to configure automatic service migration for specific services in an enterprise deployment.

Setting the Leasing Mechanism and Data Source for an Enterprise Deployment Cluster

Before you can configure automatic service migration, you must verify the leasing mechanism and data source that is used by the automatic service migration feature. You must configure the leasing mechanism and datasource for both static and dynamic clusters.

> **Note:**
>
> To accomplish data source consolidation and connection usage reduction, you can reuse the `WLSSchemaDatasource` datasource as is for database leasing. This datasource is already configured with the `FMW1221_WLS_RUNTIME` schema, where the leasing table is stored.
>
> The following procedure assumes that you have configured the Leasing data source either by reusing the `WLSSchemaDatasource` or a custom datasource that you created as described in Creating a GridLink Data Source for Leasing.

1. Log in to the Oracle WebLogic Server Administration Console.
2. Click **Lock & Edit**.
3. In the Domain Structure window, expand **Environment** and select **Clusters**.

   The Summary of Clusters page appears.
4. In the **Name** column of the table, click the cluster for which you want to configure migration.
5. Click the **Migration** tab.
6. Verify that **Database** is selected in the **Migration Basis** drop-down menu.

7. From the **Data Source for Automatic Migration** drop-down menu, select the Leasing data source that you created in Creating a GridLink Data Source for Leasing. Select the `WLSSchemaDatasource` for data source consolidation.
8. Click **Save**.
9. Activate changes.
10. Restart the managed servers for the changes to be effective. If you are configuring other aspects of ASM in the same configuration change session, you can use a final unique restart to reduce downtime.

Changing the Migration Settings for the Managed Servers in the Cluster

After you set the leasing mechanism and data source for the cluster, you can then enable automatic JTA migration for the Managed Servers that you want to configure for service migration. Note that this topic applies only if you are deploying JTA services as part of your enterprise deployment.

For example, this task is not required for Oracle WebCenter Content enterprise deployments.

To change the migration settings for the Managed Servers in each cluster:

1. If you haven't already, log in to the Administration Console, and click **Lock & Edit**.
2. In the Domain Structure pane, expand the **Environment** node and then click **Servers**.

   The Summary of Servers page appears.
3. Click the name of the server you want to modify in **Name** column of the table.

   The settings page for the selected server appears and defaults to the Configuration tab.
4. Click the **Migration** tab.
5. From the **JTA Migration Policy** drop-down menu, select **Failure Recovery**.
6. In the **JTA Candidate Servers** section of the page, select the Managed Servers in the **Available** list box, and then click the move button to move them into the **Chosen** list box.
7. In the **JMS Service Candidate Servers** section of the page, select the Managed Servers in the **Available** list box, and then click the move button to move them into the **Chosen** list box.
8. Click **Save**.
9. Restart the managed servers and the Administration Server for the changes to be effective. If you are configuring other aspects of ASM in the same configuration change session, you can use a final unique restart to reduce downtime.

About Selecting a Service Migration Policy

When you configure Automatic Service Migration, you select a Service Migration Policy for each cluster. This topic provides guidelines and considerations when selecting the Service Migration Policy.

For example, products or components running singletons or using Path services can benefit from the **Auto-Migrate Exactly-Once** policy. With this policy, if at least one Managed Server in the candidate server list is running, the services hosted by this migratable target are active somewhere in the cluster if servers fail or are administratively shut down (either gracefully or forcibly). This can cause multiple homogenous services to end up in one server on startup.

When you use this policy, you should monitor the cluster startup to identify what servers are running on each server. You can then perform a manual failback, if necessary, to place the system in a balanced configuration.

Other Fusion Middleware components are better suited for the **Auto-Migrate Failure-Recovery Services** policy.

Based on these guidelines, you should use **Auto-Migration Failure-Recovery Services** for the clusters in an Oracle WebCenter Content enterprise deployment.

See Policies for Manual and Automatic Service Migration in *Oracle Fusion Middleware Administering Clusters for Oracle WebLogic Server*.

Setting the Service Migration Policy for Each Managed Server in the Cluster

After you modify the migration settings for each server in the cluster, you can then identify the services and set the migration policy for each Managed Server in the cluster, using the WebLogic Administration Console:

1. If you have not already, log in to the Administration Console, and click **Lock & Edit**.
2. In the Domain Structure pane, expand **Environment**, then expand **Clusters**, then select **Migratable Targets**.
3. Click the name of the first Managed Server in the cluster.
4. Click the **Migration** tab.
5. From the **Service Migration Policy** drop-down menu, select the appropriate policy for the cluster.

   See About Selecting a Service Migration Policy.
6. In the Constrained Candidate Servers section of the page, select both Managed Servers in the **Available** list box and move them to the **Chosen** list box.
7. Click **Save**.
8. Repeat steps 2 through 7 for each of the additional Managed Servers in the cluster.
9. In the Domain Structure pane, select **Environment** > **Clusters** > *cluster_name* > **Configuration** tab > **General** tab > **Advanced**.
10. In the Service Activation Request Response Timeout field, type `0`.
11. Activate the changes.
12. Restart the managed servers for the changes to be effective. If you are configuring other aspects of ASM in the same configuration change session, you can use a final unique restart to reduce downtime.

Validating Automatic Service Migration in Static Clusters

After you configure automatic service migration for your cluster and Managed Servers, validate the configuration, as follows:

1. If you have not already done so, log in to the Administration Console.
2. In the Domain Structure pane, expand **Environment**, and then expand **Clusters**.
3. Click **Migratable Targets**.
4. Click the **Control** tab.

   The console displays a list of migratable targets and their current hosting server.
5. In the Migratable Targets table, select a row for the one of the migratable targets.
6. Note the value in the **Current Hosting Server** column.
7. Use the operating system command line to stop the first Managed Server.

   Use the following command to end the Managed Server Process and simulate a crash scenario:

   ```
   kill -9 pid
   ```

   In this example, replace *pid* with the process ID (PID) of the Managed Server. You can identify the PID by running the following UNIX command:

   ```
   ps -ef | grep managed_server_name
   ```

   > **Note:**
   >
   > After you kill the process, the Managed Server might be configured to start automatically. In this case, you must kill the second process using the `kill –9` command again.

8. Watch the terminal window (or console) where the Node Manager is running.

   You should see a message indicating that the selected Managed Server has failed. The message is similar to the following:

   ```
   <INFO> <domain_name> <server_name>
   <The server 'server_name' with process id 4668 is no longer alive; waiting
   for the process to die.>
   <INFO> <domain_name> <server_name>
   <Server failed during startup. It may be retried according to the auto
   restart configuration.>
   <INFO> <domain_name> <server_name>
   <Server failed but will not be restarted because the maximum number of
   restart attempts has been exceeded.>
   ```

9. Return to the Oracle WebLogic Server Administration Console and refresh the table of migratable targets; verify that the migratable targets are transferred to the remaining, running Managed Server in the cluster:
   - Verify that the Current Hosting Server for the process you killed is now updated to show that it has been migrated to a different host.
   - Verify that the value in the **Status of Last Migration** column for the process is *Succeeded*.

10. Open and review the log files for the Managed Servers that are now hosting the services; look for any JTA or JMS errors.

> **Note:**
>
> For JMS tests, it is a good practice to get message counts from destinations and make sure that there are no stuck messages in any of the migratable targets:
>
> For example, for uniform distributed destinations (UDDs):
>
> a. Access the JMS Subdeployment module in the Administration Console:
>
> In the Domain Structure pane, select **Services**, then **Messaging**, and then **JMS Modules**.
>
> b. Click the JMS Module.
>
> c. In the Summary of Resources table, click **Destinations**, and then click the **Monitoring** tab.
>
> d. Review the **Messages Total** and **Messages Pending** values. Click **Customize table** to add these columns to the table, if these values do not appear in the table.

Failing Back Services After Automatic Service Migration

When Automatic Service Migration occurs, Oracle WebLogic Server does not support failing back services to their original server when a server is back online and rejoins the cluster.

As a result, after the Automatic Service Migration migrates specific JMS services to a backup server during a fail-over, it does not migrate the services back to the original server after the original server is back online. Instead, you must migrate the services back to the original server manually.

To fail back a service to its original server, follow these steps:

1. If you have not already done so, in the Change Center of the Administration Console, click **Lock & Edit.**
2. In the Domain Structure tree, expand **Environment**, expand **Clusters**, and then select **Migratable Targets**.
3. To migrate one or more migratable targets at once, on the Summary of Migratable Targets page:
   a. Click the **Control** tab.
   b. Use the check boxes to select one or more migratable targets to migrate.
   c. Click **Migrate**.
   d. Use the **New hosting server** drop-down to select the original Managed Server.
   e. Click **OK**.

A request is submitted to migrate the JMS-related service. In the Migratable Targets table, the Status of Last Migration column indicates whether the requested migration has succeeded or failed.

f. Release the edit lock after the migration is successful.

20

Configuring Single Sign-On for an Enterprise Deployment

You need to configure the Oracle HTTP Server WebGate in order to enable single sign-on with Oracle Access Manager.

- About Oracle HTTP Server Webgate
  Oracle HTTP Server WebGate is a web server plug-in that intercepts HTTP requests and forwards them to an existing Oracle Access Manager instance for authentication and authorization.
- General Prerequisites for Configuring Oracle HTTP Server WebGate
  Before you can configure Oracle HTTP Server WebGate, you must have installed and configured a certified version of Oracle Access Manager.
- Enterprise Deployment Prerequisites for Configuring OHS 12c Webgate
  When you are configuring Oracle HTTP Server Webgate to enable single sign-on for an enterprise deployment, consider the prerequisites mentioned in this section.
- Configuring Oracle HTTP Server 12c WebGate for an Enterprise Deployment
  You need to perform the following steps in order to configure Oracle HTTP Server 12*c* WebGate for Oracle Access Manager on both WEBHOST1 and WEBHOST2.
- Registering the Oracle HTTP Server WebGate with Oracle Access Manager
  You can register the WebGate agent with Oracle Access Manager using the Oracle Access Manager Administration console.
- Setting Up the WebLogic Server Authentication Providers
  To set up the WebLogic Server authentication providers, back up the configuration files, set up the Oracle Access Manager Identity Assertion Provider and set the order of providers.
- Configuring Oracle ADF and OPSS Security with Oracle Access Manager
  Some Oracle Fusion Middleware management consoles use Oracle Application Development Framework (Oracle ADF) security, which can integrate with Oracle Access Manager Single Sign On (SSO). These applications can take advantage of Oracle Platform Security Services (OPSS) SSO for user authentication, but you must first configure the domain-level `jps-config.xml` file to enable these capabilities.

About Oracle HTTP Server Webgate

Oracle HTTP Server WebGate is a web server plug-in that intercepts HTTP requests and forwards them to an existing Oracle Access Manager instance for authentication and authorization.

For Oracle Fusion Middleware 12*c*, the Oracle WebGate software is installed as part of the Oracle HTTP Server 12*c* software installation. See Registering and Managing OAM 11g Agents in *Adminstrator's Guide for Oracle Access Management*.

General Prerequisites for Configuring Oracle HTTP Server WebGate

Before you can configure Oracle HTTP Server WebGate, you must have installed and configured a certified version of Oracle Access Manager.

For the most up-to-date information, see the certification document for your release on the *Oracle Fusion Middleware Supported System Configurations* page.

For WebGate certification matrix, click and open http://www.oracle.com/technetwork/middleware/id-mgmt/downloads/oam-webgates-2147084.html, then click the *Certification Matrix for 12c Access Management WebGates* link to download the certification matrix spreadsheet.

> **Note:**
>
> For production environments, it is highly recommended that you install Oracle Access Manager in its own environment and not on the machines that are hosting the enterprise deployment.

For more information about Oracle Access Manager, see the latest Oracle Identity and Access Management documentation, which you can find in the **Middleware** documentation on the Oracle Help Center.

Enterprise Deployment Prerequisites for Configuring OHS 12*c* Webgate

When you are configuring Oracle HTTP Server Webgate to enable single sign-on for an enterprise deployment, consider the prerequisites mentioned in this section.

- Oracle recommends that you deploy Oracle Access Manager as part of a highly available, secure, production environment. For more information about deploying Oracle Access Manager in an enterprise environment, see the Enterprise Deployment Guide for your version of Oracle Identity and Access Mangement.
- To enable single sign-on for the WebLogic Server Administration Console and the Oracle Enterprise Manager Fusion Middleware Control, you must add a central LDAP-provisioned administration user to the directory service that Oracle Access Manager is using (for example, Oracle Internet Directory or Oracle Unified Directory). For more information about the required user and groups to add to the LDAP directory, follow the instructions in Creating a New LDAP Authenticator and Provisioning Enterprise Deployment Users and Group.

> **Note:**
>
> It is recommended that you use the WebGate version that is certified with your Oracle Access Manager deployment.

Configuring Oracle HTTP Server 12*c* WebGate for an Enterprise Deployment

You need to perform the following steps in order to configure Oracle HTTP Server 12*c* WebGate for Oracle Access Manager on both WEBHOST1 and WEBHOST2.

In the following procedure, replace the directory variables, such as *WEB_ORACLE_HOME* and *WEB_CONFIG_DIR*, with the values, as defined in File System and Directory Variables Used in This Guide.

1. Perform a complete backup of the web tier domain.
2. Change directory to the following location in the Oracle HTTP Server Oracle home:

   ```
   cd WEB_ORACLE_HOME/webgate/ohs/tools/deployWebGate/
   ```

3. Run the following command to create the WebGate Instance directory and enable WebGate logging on OHS Instance:

   ```
   ./deployWebGateInstance.sh -w WEB_CONFIG_DIR -oh WEB_ORACLE_HOME
   ```

4. Verify that a `webgate` directory and subdirectories was created by the `deployWebGateInstance` command:

   ```
   ls -lat WEB_CONFIG_DIR/webgate/
   total 16
   drwxr-x---+ 8 orcl oinstall 20 Oct  2 07:14 ..
   drwxr-xr-x+ 4 orcl oinstall  4 Oct  2 07:14 .
   drwxr-xr-x+ 3 orcl oinstall  3 Oct  2 07:14 tools
   drwxr-xr-x+ 3 orcl oinstall  4 Oct  2 07:14 config
   ```

5. Run the following command to ensure that the `LD_LIBRARY_PATH` environment variable contains *WEB_ORACLE_HOME*`/lib` directory path:

   ```
   export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:WEB_ORACLE_HOME/lib
   ```

6. Change directory to the following directory

   ```
   WEB_ORACLE_HOME/webgate/ohs/tools/setup/InstallTools
   ```

7. Run the following command from the `InstallTools` directory.

   ```
   ./EditHttpConf -w WEB_CONFIG_DIR -oh WEB_ORACLE_HOME -o
   output_file_name
   ```

   > **Note:**
   >
   > The `-oh` *WEB_ORACLE_HOME* and `-o` *output_file_name* parameters are optional.

   This command:

- Copies the `apache_webgate.template` file from the Oracle HTTP Server Oracle home to a new `webgate.conf` file in the Oracle HTTP Server configuration directory.
- Updates the `httpd.conf` file to add one line, so it includes the `webgate.conf`.
- Generates a WebGate configuration file. The default name of the file is `webgate.conf`, but you can use a custom name by using the `-o` *`output_file_name`* argument to the command.

Registering the Oracle HTTP Server WebGate with Oracle Access Manager

You can register the WebGate agent with Oracle Access Manager using the Oracle Access Manager Administration console.

See Registering an OAM Agent Using the Console in *Administrator's Guide for Oracle Access Management*.

Locating and Preparing the RREG Tool

To set up the RREG tool, complete the following steps:

1. Sign in to one of the Oracle Access Manager hosts in the Application tier.
2. Change to the following directory in the Oracle Access Manager Oracle home:

   `OAM_ORACLE_HOME``/oam/server/rreg/client`

> **Note:**
>
> The location is required only for the out-of-band mode.

In this example, *OAM_ORACLE_HOME* refers to the Oracle home on the system where the Oracle Access Manager software was installed.

> **Note:**
>
> If the Oracle Enterprise Deployment Guide for IDM was used, *OAM_ORACLE_HOME* may be `/u01/oracle/products/access/iam`.

> **Note:**
>
> If you do not have privileges or access to the Oracle Access Manager server, then you can use out-of-band mode to generate the required files and register the WebGate with Oracle Access Manager. See About RREG In-Band and Out-of-Band Mode.

3. Decompress the `RREG.tar.gz` file to the required directory.
4. From the unzipped directory, open the `oamreg.sh` file and set the following environment variables in the file, as follows:
   - Set `OAM_REG_HOME` to the absolute path to the directory in which you extracted the contents of RREG archive.

     Set `JAVA_HOME` to the absolute path of the directory in which a supported JDK is installed on your machine.

About RREG In-Band and Out-of-Band Mode

You can run the RREG Tool in one of the two modes: in-band and out-of-band.

Use **in-band** mode when you have the privileges to access the Oracle Access Manager server and run the RREG tool yourself from the Oracle Access Manager Oracle home. You can then copy the generated artifacts and files to the web server configuration directory after you run the RREG Tool.

Use **out-of-band** mode if you do *not* have privileges or access to the Oracle Access Manager server. For example, in some organizations, only the Oracle Access Manager server administrators have privileges to access the server directories and perform administration tasks on the server. In out-of-band mode, the process can work as follows:

1. The Oracle Access Manager server administrator provides you with a copy of the RREG archive file (RREG.tar.gz).

   The server administrator can find it in the location described in Locating and Preparing the RREG Tool.
2. Untar the `RREG.tar.gz` file that was provided to you by the server administrator.

   For example:

   ```
   gunzip RREG.tar.gz
   ```

   ```
   tar -xvf RREG.tar
   ```

   After you unpack the RREG archive, you can find the tool for registering the agent in the following location:

   RREG_HOME`/bin/oamreg.sh`

In this example, *RREG_Home* is the directory in which you extracted the contents of RREG archive.

3. Use the instructions in Updating the Standard Properties in the OAM11gRequest.xml File to update the `OAM11GRequest.xml` file, and send the completed `OAM11GRequest.xml` file to the Oracle Access Manager server administrator.
4. The Oracle Access Manager server administrator then uses the instructions in Running the RREG Tool in Out-Of-Band Mode to run the RREG Tool and generate the *AgentID*`_response.xml` file.
5. The Oracle Access Manager server administrator sends the *AgentID*`_response.xml` file to you.
6. Use the instructions in Running the RREG Tool in Out-Of-Band Mode to run the RREG Tool with the *AgentID*`_response.xml` file and generate the required artifacts and files on the client system.

Updating the Standard Properties in the OAM11gRequest.xml File

Before you can register the Webgate agent with Oracle Access Manager, you must update some required properties in the `OAM11gRequest.xml` file.

> **Note:**
>
> - If you plan to use the default values for most of the parameters in the provided XML file, then you can use the shorter version (`OAM11gRequest_short.xml`, in which all non-listed fields take a default value.
> - In the primary server list, the default names are mentioned as OAM_SERVER1 and OAM_SERVER2 for OAM servers. Rename these names in the list if the server names are changed in your environment.

To perform this task:

1. If you are using in-band mode, then change directory to the following location on one of the OAM Servers:

   OAM_ORACLE_HOME`/oam/server/rreg/input`

   If you are using out-of-band mode, then change directory to the location where you unpacked the RREG archive on the WEBHOST1 server.
2. Make a copy of the `OAM11GRequest.xml` file template with an environment-specific name.

   ```
   cp OAM11GRequest.xml OAM11GRequest_edg.xml
   ```
3. Review the properties listed in the file, and then update your copy of the `OAM11GRequest.xml` file to make sure that the properties reference the host names and other values specific to your environment.

Table 20-1 Fields in the OAM11GRequest.xml file.

OAM11gRequest.xml Property	Set to...
`serverAddress`	The host and the port of the Administration Server for the Oracle Access Manager domain.
`agentName`	Any custom name for the agent. Typically, you use a name that identifies the Fusion Middleware product that you are configuring for single sign-on.
`applicationDomain`	A value that identifies the web tier host and the FMW component you are configuring for single sign-on.
`security`	Must be set to the security mode configured on the Oracle Access Management server. This is one of the three modes: open, simple, or certificate. **Note:** For an enterprise deployment, Oracle recommends simple mode, unless additional requirements exist to implement custom security certificates for the encryption of authentication and authorization traffic. In most cases, avoid using open mode, because in open mode, traffic to and from the Oracle Access Manager server is not encrypted. For more information using certificate mode or about Oracle Access Manager supported security modes in general, see Securing Communication Between OAM Servers and WebGates in *Administrator's Guide for Oracle Access Management*.
`cachePragmaHeader`	private
`cacheControlHeader`	private

Table 20-1 (Cont.) Fields in the OAM11GRequest.xml file.

OAM11gRequest.xml Property	Set to...
`ipValidation`	0 `<ipValidation>0</ipValidation>` If ipValidation is set to '1', the IP address stored in the cookie must match the client's IP address, otherwise, the SSO cookie is rejected and the user must reauthenticate. This can cause problems with certain Web applications. For example, Web applications managed by a proxy server typically change the user's IP address, substituting the IP address of the proxy. Setting to '0' Disables IP validation.

`ipValidationExceptions`	Can be empty when ipValidation is '0'.
	If IP Validation is true, the IP address is compared to the IP Validation Exceptions list. If the address is found on the exceptions list, it does not need to match the IP address stored in the cookie. You can add as many IP addresses as needed. For example, the IP address of the front end load balancer:
	
	`<ipValidationExceptions>`
	`<ipAddress>130.35.165.42</ipAddress>`
	`</ipValidationExceptions>`

`agentBaseUrl`	Fully-qualified URL with the host and the port of the front-end Load Balancer VIP in front of the WEBHOST*n* machines on which Oracle HTTP 12*c* WebGates are installed.
	For example:
	
	`<agentBaseUrl>`
	`https://wcc.example.com:443`
	`</agentBaseUrl>`

`virtualHost`	Set to **true** when protecting more than the `agentBaseUrl`, such as SSO protection for the administrative VIP.

Table 20-1 (Cont.) Fields in the OAM11GRequest.xml file.

OAM11gRequest.xml Property	Set to...

`hostPortVariationsList`	Add `hostPortVariation` host and port elements for each of the load-balancer URLs that are protected by the WebGates.
	For example:
	
	`<hostPortVariationsList>`
	`<hostPortVariations>`
	`<host>wccinternal.example.com</host>`
	`<port>80</port>`
	`</hostPortVariations>`
	`<hostPortVariations>`
	`<host>admin.example.com</host>`
	`<port>80</port>`
	`</hostPortVariations>`
	`<hostPortVariations>`
	`<host>osb.example.com</host>`
	
	`<port>443</port>`
	`</hostPortVariations>`
	`</hostPortVariationsList>`

`logOutUrls`	Leave it empty. The Logout URL triggers the logout handler, which removes the cookie and requires the user to re-authenticate the next time the user accesses a resource protected by Access Manager. If Logout URL is not configured, the request URL is checked for *logout*. and, if found (except *logout.gif* and *logout.jpg*), also triggers the logout handler. If a value is set to this property, all used logout URLs must be added.

`primaryServerList`	Verify that the host and the port of the OAM Managed Servers matches with this list. Example:
	
	`<primaryServerList>`
	`<Server>`
	`<host>wls_oam1</host>`
	`<port>14100</port>`
	`<numOfConnections>1</numOfConnections>`
	`</Server>`
	`<Server>`
	`<host>wls_oam2</host>`
	`<port>14100</port>`
	`<numOfConnections>2</numOfConnections>`
	`</Server>`
	`</primaryServerList>`

Updating the Protected, Public, and Excluded Resources for an Enterprise Deployment

When you set up an Oracle Fusion Middleware environment for single sign-on, you identify a set of URLs that you want Oracle Access Manager to protect with single sign-on. You identify these using specific sections of the `OAM11gRequest.xml` file. To identify the URLs:

1. If you have not already opened the copied `OAM11GRequest_edg.xml` file for editing, locate, and open the file in a text editor.

   See Locating and Preparing the RREG Tool

   See Updating the Standard Properties in the OAM11gRequest.xml File

2. Remove the sample entries from the file, and then enter the list of protected, public, and excluded resources in the appropriate sections of the file, as shown in the following example.

> **Note:**
>
> If you are using Oracle Access Manager 11*g* Release 2 (11.1.2.2) or later, then note that the entries with the wildcard syntax ("`.../*`") are included in this example for backward compatibility with previous versions of Oracle Access Manager.

```
<protectedResourcesList>
        <resource>/adfAuthentication</resource>
        <resource>/ibr/adfAuthentication</resource>
        <resource>/imaging/faces</resource>
        <resource>/em</resource>
        <resource>/console</resource>
        <resource>/DefaultToDoTaskFlow</resource>
        <resource>/sdpmessaging/userprefs-ui</resource>
        <resource>/integration/worklistapp</resource>
        <resource>/workflow/sdpmessagingsca-ui-worklist</resource>
        <resource>/soa/composer</resource>
        <resource>/soa-infra/deployer</resource>
        <resource>/soa-infra/events/edn-db-log</resource>
        <resource>/soa-infra/cluster/info</resource>
        <resource>/soa-infra</resource>
        <resource>/dc-console/adfAuthentication</resource>
        <resource>/dc-client/adfAuthentication</resource>
        <resource>/wcc/adfAuthentication</resource>
    </protectedResourcesList>
    <publicResourcesList>
        <resource>/cs</resource>
        <resource>/_ocsh</resource>
        <resource>/_dav</resource>
        <resource>/imaging</resource>
        <resource>/soa-infra/directWSDL</resource>
        <resource>/dc-console</resource>
```

```
        <resource>/wcc</resource>
    </publicResourcesList>
    <excludedResourcesList>
        <resource>/imaging/jnlp</resource>
        <resource>/wsm-pm</resource>
        <resource>/soa-infra/services</resource>
        <resource>/ucs/messaging/webservice</resource>
        <resource>/imaging/ws</resource>
        <resource>/idcnativews</resource>
        <resource>/imaging/lib</resource>
        <resource>/cs/common/idcapplet.jar</resource>
        <resource>/cs/common/checkoutandopen*.jar </resource>
        <resource>/cs/images</resource>
        <resource>/dc-client</resource>
        <resource>/axf-ws</resource>
    </excludedResourcesList>
```

3. Save and close the `OAM11GRequest_edg.xml` file.

> **Note:**
>
> For 11*g* Webgate, a user-defined parameter `filterOAMAuthnCookie` (default set to `true`) can be used to prevent the `OAMAuthnCookie` from being passed to downstream applications for security consideration. Some products such as IPM require to set the `filterOAMAuthnCookie` parameter to `false`.

Running the RREG Tool

The following topics provide information about running the RREG tool to register your Oracle HTTP Server Webgate with Oracle Access Manager.

- Running the RREG Tool in In-Band Mode
- Running the RREG Tool in Out-Of-Band Mode

Running the RREG Tool in In-Band Mode

To run the RREG Tool in in-band mode:

1. Change to the RREG home directory.

   If you are using in-band mode, the RREG directory is inside the Oracle Access Manager Oracle home:

   OAM_ORACLE_HOME`/oam/server/rreg`

   If you are using out-of-band mode, then the RREG home directory is the location where you unpacked the RREG archive.

2. Change to the following directory:

   - (UNIX) *RREG_HOME*`/bin`
   - (Windows) *RREG_HOME*`\bin`

   `cd` *RREG_HOME*`/bin/`

3. Set the permissions of the `oamreg.sh` command so that you can execute the file:

   ```
   chmod +x oamreg.sh
   ```

4. Enter the following command:

   ```
   ./oamreg.sh inband RREG_HOME/input/OAM11GRequest_edg.xml
   ```

In this example:

- It is assumed that the edited `OAM11GRequest.xml` file is located in the *`RREG_HOME`*`/input` directory.
- The output from this command is saved to the following directory:

  ```
  RREG_HOME/output/
  ```

The following example shows a sample RREG session:

```
Welcome to OAM Remote Registration Tool!
Parameters passed to the registration tool are:
Mode: inband
Filename: /u01/oracle/products/fmw/iam_home/oam/server/rreg/client/rreg/
input/OAM11GRequest_edg.xml
Enter admin username:weblogic_idm
Username: weblogic_iam
Enter admin password:
Do you want to enter a Webgate password?(y/n):
n
Do you want to import an URIs file?(y/n):
n

----------------------------------------
Request summary:
OAM11G Agent Name:SOA12213_EDG_AGENT
Base URL: https://soa.example.com:443
URL String:null
Registering in Mode:inband
Your registration request is being sent to the Admin server at: http://
host1.example.com:7001
----------------------------------------

Jul 08, 2015 7:18:13 PM oracle.security.jps.util.JpsUtil disableAudit
INFO: JpsUtil: isAuditDisabled set to true
Jul 08, 2015 7:18:14 PM oracle.security.jps.util.JpsUtil disableAudit
INFO: JpsUtil: isAuditDisabled set to true
Inband registration process completed successfully! Output artifacts
are created in the output folder.
```

Running the RREG Tool in Out-Of-Band Mode

To run the RREG Tool in out-of-band mode on the WEBHOST server, the administrator uses the following command:

```
RREG_HOME/bin/oamreg.sh outofband input/OAM11GRequest.xml
```

In this example:

- Replace *RREG_HOME* with the location where the RREG archive file was unpacked on the server.
- The edited `OAM11GRequest.xml` file is located in the *RREG_HOME*`/input` directory.
- The RREG Tool saves the output from this command (the *AgentID*`_response.xml` file) to the following directory:

  RREG_HOME`/output/`

  The Oracle Access Manager server administrator can then send the *AgentID*`_response.xml` to the user who provided the `OAM11GRequest.xml` file.

To run the RREG Tool in out-of-band mode on the web server client machine, use the following command:

RREG_HOME`/bin/oamreg.sh outofband input/AgentID_response.xml`

In this example:

- Replace *RREG_HOME* with the location where you unpacked the RREG archive file on the client system.
- The *AgentID*`_response.xml` file, which was provided by the Oracle Access Manager server administrator, is located in the *RREG_HOME*/input directory.
- The RREG Tool saves the output from this command (the artifacts and files required to register the Webgate software) to the following directory on the client machine:

  RREG_HOME`/output/`

Files and Artifacts Generated by RREG

The files that get generated by the RREG Tool vary, depending on the security level you are using for communications between the WebGate and the Oracle Access Manager server. See Securing Communication Between OAM Servers and WebGates in *Administrator's Guide for Oracle Access Management*.

Note that in this topic any references to *RREG_HOME* should be replaced with the path to the directory where you ran the RREG tool. This is typically the following directory on the Oracle Access Manager server, or (if you are using out-of-band mode) the directory where you unpacked the RREG archive:

OAM_ORACLE_HOME`/oam/server/rreg/client`

The following table lists the artifacts that are always generated by the RREG Tool, regardless of the Oracle Access Manager security level.

File	Location
`cwallet.sso`	• *RREG_HOME*`/output/Agent_ID/` - For WebGate 11*g* (11.1.2.3). • *RREG_HOME*`/output/Agent_ID/wallet` - For WebGate 11*g* (11.1.2.2) and OHS 12*c*.
`ObAccessClient.xml`	*RREG_HOME*`/output/Agent_ID/`

The following table lists the additional files that are created if you are using the SIMPLE or CERT security level for Oracle Access Manager:

File	Location
aaa_key.pem	*RREG_HOME*/output/Agent_ID/
aaa_cert.pem	*RREG_HOME*/output/Agent_ID/
password.xml	*RREG_HOME*/output/Agent_ID/

Note that the password.xml file contains the obfuscated global passphrase to encrypt the private key used in SSL. This passphrase can be different than the passphrase used on the server.

You can use the files generated by RREG to generate a certificate request and get it signed by a third-party Certification Authority. To install an existing certificate, you must use the existing aaa_cert.pem and aaa_chain.pem files along with password.xml and aaa_key.pem.

Copying Generated Artifacts to the Oracle HTTP Server WebGate Instance Location

After the RREG Tool generates the required artifacts, manually copy the artifacts from the *RREG_Home*/output/*agent_ID* directory to the Oracle HTTP Server configuration directory on the web tier host.

The location of the files in the Oracle HTTP Server configuration directory depends upon the Oracle Access Manager security mode setting (OPEN, SIMPLE, or CERT).

The following table lists the required location of each generated artifact in the Oracle HTTP Server configuration directory, based on the security mode setting for Oracle Access Manager. In some cases, you might have to create the directories if they do not exist already. For example, the wallet directory might not exist in the configuration directory.

> **Note:**
>
> For an enterprise deployment, Oracle recommends simple mode, unless additional requirements exist to implement custom security certificates for the encryption of authentication and authorization traffic. The information about using open or certification mode is provided here as a convenience.
>
> Avoid using open mode, because in open mode, traffic to and from the Oracle Access Manager server is not encrypted.
>
> For more information about using certificate mode or about Oracle Access Manager supported security modes in general, see Securing Communication Between OAM Servers and WebGates in *Administrator's Guide for Oracle Access Management*.

Table 20-2 Web Tier Host Location to Copy the Generated Artifacts

File	Location When Using OPEN Mode	Location When Using SIMPLE Mode	Location When Using CERT Mode
wallet/cwallet.sso[1]	*WEB_CONFIG_DIR*/webgate/config/wallet	*WEB_CONFIG_DIR*/webgate/config/wallet/ By default the wallet folder is not available. Create the wallet folder under *WEB_CONFIG_DIR*/webgate/config/.	*WEB_CONFIG_DIR*/webgate/config/wallet/
ObAccessClient.xml	*WEB_CONFIG_DIR*/webgate/config	*WEB_CONFIG_DIR*/webgate/config/	*WEB_CONFIG_DIR*/webgate/config/
password.xml	N/A	*WEB_CONFIG_DIR*/webgate/config/	*WEB_CONFIG_DIR*/webgate/config/
aaa_key.pem	N/A	*WEB_CONFIG_DIR*/webgate/config/simple/	*WEB_CONFIG_DIR*/webgate/config/
aaa_cert.pem	N/A	*WEB_CONFIG_DIR*/webgate/config/simple/	*WEB_CONFIG_DIR*/webgate/config/

[1] Copy `cwallet.sso` from the wallet folder and not from the output folder. Even though there are 2 files with the same name they are different. The one in the wallet sub directory is the correct one.

> **Note:**
>
> If you need to redeploy the `ObAccessClient.xml` to `WEBHOST1` and `WEBHOST2`, delete the cached copy of `ObAccessClient.xml` and its lock file, `ObAccessClient.xml.lck` from the servers. The cache location on `WEBHOST1` is:
>
> *WEB_DOMAIN_HOME*/servers/ohs1/cache/
>
> And you must perform the similar step for the second Oracle HTTP Server instance on `WEBHOST2`:
>
> *WEB_DOMAIN_HOME*/servers/ohs2/cache/

Insert OHS SimpleCA Certificate into the Wallet Artifact

If the OHS servers have been configured with an 11*g* or earlier version of the OAM server, there is a need to insert the OHS SimpleCA certificate into the wallet file artifact that was deployed in Copying Generated Artifacts to the Oracle HTTP Server WebGate Instance Location.

Complete the following steps:

1. On WEBHOST1, go to the following directory:

   ```
   WEB_CONFIG_DIR/webgate/config/wallet
   ```

2. Run the following command to insert the SimpleCA certificate into the wallet file:

```
WEB_ORACLE_HOME/oracle_common/bin/orapki wallet add -wallet ./
-trusted_cert -cert WEB_ORACLE_HOME/webgate/ohs/tools/openssl/
simpleCA/cacert.pem -auto_login_only
```

The following output is displayed:

```
  simpleCA/cacert.pem -auto_login_only
  Oracle PKI Tool : Version 12.2.1.3.0
 Copyright (c) 2004, 2017, Oracle and/or its affiliates. All rights
reserved.

 Operation is successfully completed.
```

3. Validate the certificate insertion with the following command:

```
WEB_ORACLE_HOME/oracle_common/bin/orapki wallet display -wallet ./
```

The following output is displayed:

```
  Oracle PKI Tool : Version 12.2.1.3.0
  Copyright (c) 2004, 2017, Oracle and/or its affiliates. All rights
reserved.

  Requested Certificates:
  User Certificates:
  Oracle Secret Store entries: OAMAgent@#3#@wcedgRwse01Env1Ps3_Key
  Trusted Certificates:
  Subject: CN=NetPoint Simple Security CA - Not for General
Use,OU=NetPoint,O=Oblix\, Inc.,L=Cupertino,ST=California,C=US
```

4. Repeat steps 1 through 3 on WEBHOST2.

Enable MD5 Certificate Signatures for the Oracle HTTP Server Instances

Some releases of Oracle Access Management Server implement simple mode security certificates by using MD5 signatures unless upgraded or patched appropriately. Oracle Recommends that, if possible, the OAM certificates are upgraded to SHA-2 certificates. This might not be possible for customers who have several versions of Oracle HTTP server to contend with.

If upgrading the certificates is not possible, support for MD5 signatures must be enabled manually to make Oracle HTTP server 12.2.1.*x* work with Oracle Access Manager 11*g*'s MD5 certificates when you use a webgate in simple security mode.

To enable MD5 certificate signatures on each OHS instance, complete the following steps:

1. On WEBHOST1, change to the following directory:

   WEB_DOMAIN_HOME/config/fmwconfig/components/OHS/ohs1

2. Open the `ohs.plugins.nodemanager.properties` file, add the following line, and save the file.

   ```
   environment.ORACLE_SSL_ALLOW_MD5_CERT_SIGNATURES = 1
   ```

3. Repeat steps 1 and 2 for all other instances on the WEBHOST*n* servers.

   For example, the `ohs2` instance on WEBHOST2

> **Note:**
>
> The change takes effect when the instances are restarted in the next topic.

Restarting the Oracle HTTP Server Instance

For information about restarting the Oracle HTTP Server instance, see Restarting Oracle HTTP Server Instances by Using WLST in *Oracle Fusion Middleware Administering Oracle HTTP Server*.

If you have configured Oracle HTTP Server in a WebLogic Server domain, you can also use Oracle Fusion Middleware Control to restart the Oracle HTTP Server instances. See Restarting Oracle HTTP Server Instances by Using Fusion Middleware Control in *Oracle Fusion Middleware Administering Oracle HTTP Server*.

Setting Up the WebLogic Server Authentication Providers

To set up the WebLogic Server authentication providers, back up the configuration files, set up the Oracle Access Manager Identity Assertion Provider and set the order of providers.

The following topics assumes that you have already configured the LDAP authenticator by following the steps in Creating a New LDAP Authenticator and Provisioning Enterprise Deployment Users and Group. If you have not already created the LDAP authenticator, then do so before you continue with this section.

- Backing Up Configuration Files
- Setting Up the Oracle Access Manager Identity Assertion Provider
- Updating the Default Authenticator and Setting the Order of Providers

Backing Up Configuration Files

To be safe, you should first back up the relevant configuration files:

```
ASERVER_HOME/config/config.xml
ASERVER_HOME/config/fmwconfig/jps-config.xml
ASERVER_HOME/config/fmwconfig/system-jazn-data.xml
```

Also back up the `boot.properties` file for the Administration Server:

```
ASERVER_HOME/servers/AdminServer/security/boot.properties
```

Setting Up the Oracle Access Manager Identity Assertion Provider

Set up an Oracle Access Manager identity assertion provider in the Oracle WebLogic Server Administration Console.

To set up the Oracle Access Manager identity assertion provider:

1. Log in to the WebLogic Server Administration Console, if not already logged in.
2. Click **Lock & Edit**.
3. Click **Security Realms** in the left navigation bar.
4. Click the **myrealm** default realm entry.
5. Click the **Providers** tab.
6. Click **New**, and select the asserter type **OAMIdentityAsserter** from the drop-down menu.
7. Name the asserter (for example, *OAM ID Asserter*), and click **OK**.
8. Click the newly added asserter to see the configuration screen for the Oracle Access Manager identity assertion provider.
9. Set the control flag to *REQUIRED*.
10. Under Chosen types, select both the **ObSSOCookie** and **OAM_REMOTE_USER** options, if they are not selected by default.
11. Click **Save** to save the settings.
12. Click **Activate Changes** to propagate the changes.

Updating the Default Authenticator and Setting the Order of Providers

Set the order of identity assertion and authentication providers in the WebLogic Server Administration console.

To update the default authenticator and set the order of the providers:

1. Log in to the WebLogic Server Administration Console, if not already logged in.
2. Click **Lock & Edit**.
3. From the left navigation, select **Security Realms**.
4. Click the **myrealm** default realm entry.
5. Click the **Providers** tab.
6. From the table of providers, click the **DefaultAuthenticator**.
7. Set the Control Flag to `SUFFICIENT`.
8. Click **Save** to save the settings.
9. From the navigation breadcrumbs, click **Providers** to return to the list of providers.
10. Click **Reorder**.
11. Sort the providers to ensure that the OAM Identity Assertion provider is first and the DefaultAuthenticator provider is last.

Table 20-3 Sort order

Sort Order	Provider	Control Flag
1	OAMIdentityAsserter	REQUIRED
2	LDAP Authentication Provider	SUFFICIENT
3	DefaultAuthenticator	SUFFICIENT
4	Trust Service Identity Asserter	N/A
5	DefaultIdentityAsserter	N/A

12. Click **OK**.
13. Click **Activate Changes** to propagate the changes.
14. Shut down the Administration Server, Managed Servers, and any system components, as applicable.
15. Restart the Administration Server.
16. If you are going to configure ADF consoles with SSO, you can keep the managed servers down and restart them later. If not, you need to restart managed servers now.

Configuring Oracle ADF and OPSS Security with Oracle Access Manager

Some Oracle Fusion Middleware management consoles use Oracle Application Development Framework (Oracle ADF) security, which can integrate with Oracle Access Manager Single Sign On (SSO). These applications can take advantage of Oracle Platform Security Services (OPSS) SSO for user authentication, but you must first configure the domain-level `jps-config.xml` file to enable these capabilities.

The domain-level `jps-config.xml` file is located in the following location after you create an Oracle Fusion Middleware domain:

ASERVER_HOME`/config/fmwconfig/jps-config.xml`

> **Note:**
>
> The domain-level `jps-config.xml` should not be confused with the `jps-config.xml` that is deployed with custom applications.

To update the OPSS configuration to delegate SSO actions in Oracle Access Manager, complete the following steps:

1. Change to the following directory:

   ORACLE_COMMON_HOME`/common/bin`

2. Start the WebLogic Server Scripting Tool (WLST):

   ```
   ./wlst.sh
   ```

3. Connect to the Administration Server, by using the following WLST command:

   ```
   connect('admin_user','admin_password','admin_url')
   ```

   For example:

   ```
   connect('weblogic_wcc','mypassword','t3://ADMINVHN:7001')
   ```

4. Run the `addOAMSSOProvider` command, as shown:

   ```
   addOAMSSOProvider(loginuri="/${app.context}/
   adfAuthentication", logouturi="/oamsso/logout.html")
   ```

   The following table defines the expected value for each argument in the `addOAMSSOProvider` command.

Table 20-4 Expected Values for the Argument in the `addOAMSSOProvider` command

Argument	Definition
loginuri	Specifies the URI of the login page **Note:** For ADF security enabled applications, "/*context-root*/adfAuthentication" should be provided for the 'loginuri' parameter. For example: `/${app.context}/adfAuthentication` **Note:** `${app.context}` must be entered as shown. At runtime, the application replaces the variable appropriately. Here is the flow: a. User accesses a resource that has been protected by authorization policies in OPSS, fox example. b. If the user is not yet authenticated, ADF redirects the user to the URI configured in *loginuri*. c. Access Manager, should have a policy to protect the value in *loginuri*: for example, "/*context-root*/adfAuthentication". d. When ADF redirects to this URI, Access Manager displays a Login Page (depending on the authentication scheme configured in Access Manager for this URI).

Table 20-4 (Cont.) Expected Values for the Argument in the `addOAMSSOProvider` command

Argument	Definition
logouturi	Specifies the URI of the logout page. The value of the *loginurl* is usually `/oam/logout.html`.
autologinuri	Specifies the URI of the autologin page. This is an optional parameter.

5. Disconnect from the Administration Server by entering the following command:

   ```
   disconnect()
   ```

6. Restart the Administration Server and the managed servers.

A

Using Multi Data Sources with Oracle RAC

Oracle recommends that you use GridLink data sources when you develop new Oracle RAC applications. However, if you are using legacy applications and databases that do not support GridLink data sources, refer to the information in this appendix.

This appendix provides information about multi data sources and Oracle RAC and procedure for configuring multi data sources for an Enterprise Deployment.

- About Multi Data Sources and Oracle RAC
  A multi data source provides an ordered list of data sources to use to satisfy connection requests.
- Typical Procedure for Configuring Multi Data Sources for an Enterprise Deployment
  You need to configure data sources when you configure a domain. If you want to use Multi Data Sources instead of GridLink data sources, replace the GridLink instructions with the instructions provided in this section.

About Multi Data Sources and Oracle RAC

A multi data source provides an ordered list of data sources to use to satisfy connection requests.

Normally, every connection request to this kind of multi data source is served by the first data source in the list. If a database connection test fails and the connection cannot be replaced, or if the data source is suspended, a connection is sought sequentially from the next data source on the list.

For more information about configuring Multi Data Sources with Oracle RAC, see Using Multi Data Sources with Oracle RAC in *Oracle Fusion Middleware Administering JDBC Data Sources for Oracle WebLogic Server*.

Typical Procedure for Configuring Multi Data Sources for an Enterprise Deployment

You need to configure data sources when you configure a domain. If you want to use Multi Data Sources instead of GridLink data sources, replace the GridLink instructions with the instructions provided in this section.

For example, when you are configuring the initial Administration domain for an Enterprise Deployment reference topology, you use the configuration wizard to define the characteristics of the domain, as well as the data sources.

The procedures for configuring the topologies in this Enterprise Deployment Guide include specific instructions for defining GridLink data sources with Oracle RAC. If you want to use Multi Data Sources instead of GridLink data sources, replace the GridLink instructions with the following:

1. In the Configure JDBC Component Schema screen:

a. Select the appropriate schemas.

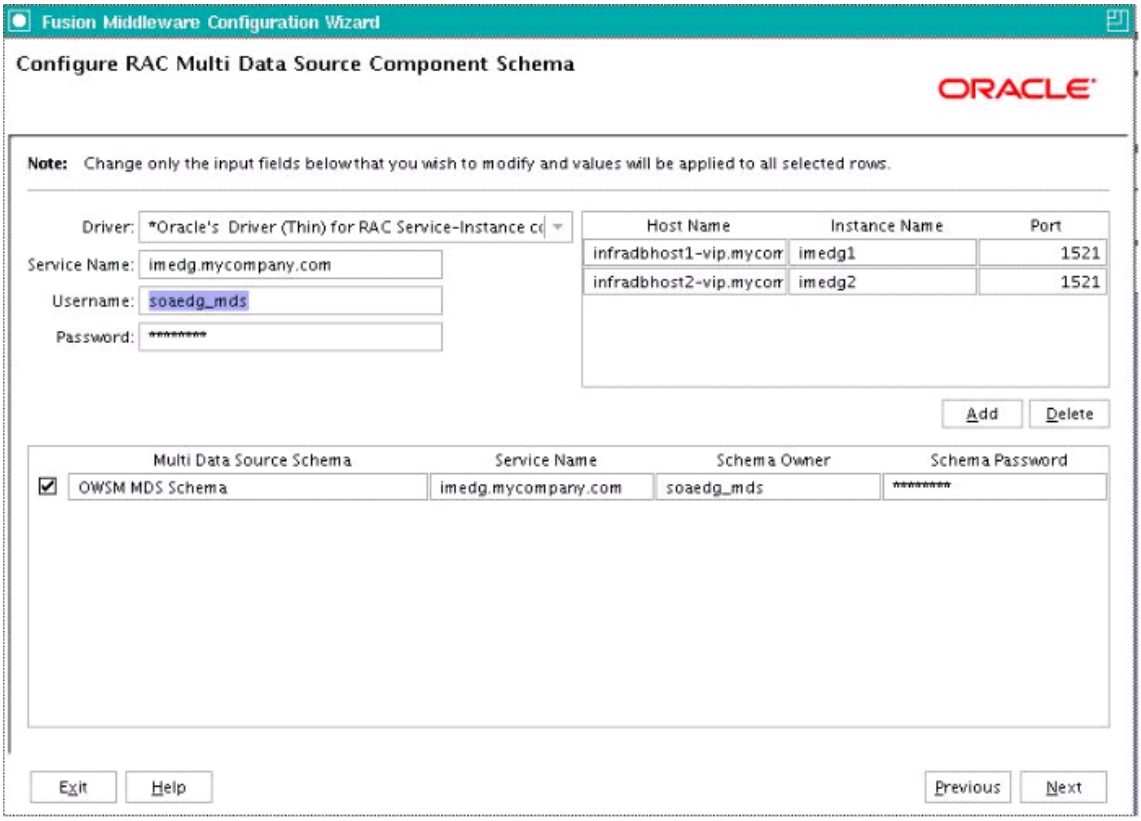
b. For the RAC configuration for component schemas, **Convert to RAC multi data source**.

c. Ensure that the following data source appears on the screen with the schema prefix when you ran the Repository Creation Utility.

d. Click **Next**.

2. The Configure RAC Multi Data Sources Component Schema screen appears (Figure A-1).

Figure A-1 Configure RAC Multi Data Source Component Schema Screen



In this screen, do the following:

a. Enter values for the following fields, specifying the connect information for the Oracle RAC database that was seeded with RCU.

- **Driver:** Select **Oracle driver (Thin) for RAC Service-Instance connections, Versions:10, 11**.
- **Service Name:** Enter the service name of the database.
- **Username:** Enter the complete user name (including the prefix) for the schemas.
- **Password:** Enter the password to use to access the schemas.

b. Enter the host name, instance name, and port.

c. Click **Add**.

d. Repeat this for each Oracle RAC instance.

e. Click **Next**.

3. In the Test JDBC Data Sources screen, the connections are tested automatically. The **Status** column displays the results. Ensure that all connections were successful. If not, click **Previous** to return to the previous screen and correct your entries.

   Click **Next** when all the connections are successful.