Oracle® Fusion Middleware

Enterprise Deployment Guide for Oracle WebCenter Content

Release 14.1.2.0.0
G12337-01
August 2025

ORACLE

Oracle Fusion Middleware Enterprise Deployment Guide for Oracle WebCenter Content, Release 14.1.2.0.0

G12337-01

Primary Author: Oracle Corporation

Contents

11 Configuring the Web Tier for an Enterprise Deployment

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 Service Migration in an Enterprise Deployment

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.

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**.

Enterprise Deployment Guides are the foundation to Maximum Availability Architectures for Oracle Fusion Middleware products. Oracle's Maximum Availability Architecture provides a set of architectures, configurations, and operational best practices that provide High Availability and Disaster Recovery solutions for the entire Oracle Stack. An Enterprise Deployment is the incarnation of these best practices in the scope of a single data center. When combined with the appropriate configuration and operational models for disaster protection, the enterprise deployment will achieve optimal high availability, data protection, and disaster recovery at the lowest cost and complexity while providing the best Recovery Time Objective (RTO) and Recovery Point Objective (RPO).

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.

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

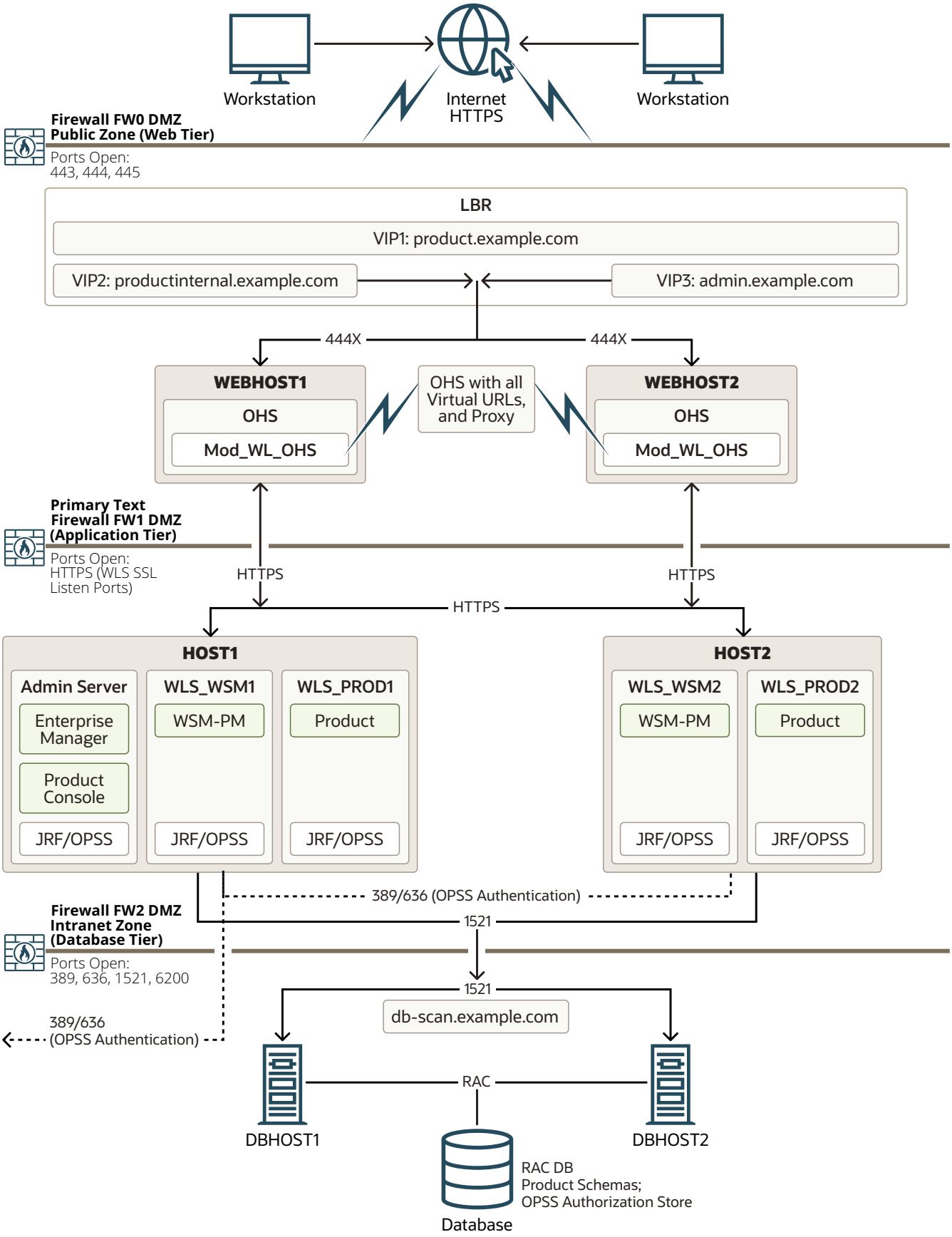
A best practices approach starts with the basic concept of a multi-tiered deployment and standard communications between the different software tiers.

This Enterprise Deployment uses secured communications using SSL all the way from the external clients to the backend WebLogic Servers. Although this eliminates numerous security vulnerabilities and increases the resiliency against different types of attacks, it has implications on the overall performance of the system. These implications vary depending on the applications deployed and the workloads in each invocation. For more information, see Configuring SSL *Administering Security for Oracle WebLogic Server*. An alternative approach, is to terminate SSL at the load balancer. This approach also ensures the communication between the client and the load balancer while maximizing performance within the system components by avoiding SSL overhead in the rest of the tiers. The downside is that this may not offer sufficient security in cloud based applications.

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

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.
- On the firewall protecting an application tier, HTTP ports, and MBean proxy port are open.

  Applications that require external HTTPS 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)

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. 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.

This SSL termination at the load balancer provides a performance gain to applications at the cost of communications being secured in the rest of the tiers only by subnet restrictions in corporate networks. Given the increased security requirements applying nowadays to production deployments, this version of the Enterprise Deployment guide is implementing end-to-end SSL communication all the way from the external clients to the middle tier. The HTTPS protocol is used for the communication between clients and the frontend load balancer, between the frontend load balancer and Oracle HTTP Servers and between Oracle HTTP

Server and the WebLogic Servers . This comes at the cost of performance overhead that is especially relevant in SSL handshakes. Depending on the type of connections used by client requests (long lived, short lived, keep alive settings) this may be a factor in the performance of an Enterprise Deployment. It is considered however that security standards have increased nowadays and this guide will focus on the most secure deployment approach.

> **Note**
>
> For the purpose of providing the most realistic working environment possible this guide uses demo certificates in the web and application tiers. Notice that demonstration digital certificates and keystores are not recommended in production mode. The sample steps provided in this book should be run or substituted with appropriate certificates signed by a formal Certificate Authority in your production environment.

Enterprise Deployment guide environments always use 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

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.

HTTPS or encrypted requests are routed from the load balancer to the web tier. This guide provides instructions for SSL configuration between the load balancer and the web tier and 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 *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 Remote 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

This Enterprise Deployment Guide uses SSL for all the virtual servers, hence the frontend port 80 is no longer used. It is a best practice to assign the main external URL (for example, https://myapplication.example.com) to the SSL port number 443.

> **Note**
>
> If port 80 remains open in the load balancer, then it is recommended to redirect any requests to it (non-SSL protocol) to port 443 (SSL protocol). Refer to your load balancer's specific documentation to implement this redirection. 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

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 faster failover in the event of a WebLogic Server instance failure. The plug-in actively learns about the failed WebLogic Server instance by using the information supplied by its peers. It avoids the failed server until the peers notify the plug-in that it is available. Load balancers are typically more limited and their monitors cause higher overhead.
- 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 provides faster failover in the event of a WebLogic Server instance failure. The plug-in actively learns about the failed WebLogic Server instance by using the information supplied by its peers. It avoids the failed server until the peers notify the plug-in that it is available. Load balancers are typically more limited and their monitors cause higher overhead.
- 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 tier 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.
- The web tier 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.
- The web tier 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.
- A web tier with 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 *Administering Oracle HTTP Server*.

Configuration of Oracle HTTP Server in the Web Tier

This enterprise deployment guide provides information about configuring the Oracle HTTP Server instances as separate standalone domains, one on each web tier host. In each OHS Host the corresponding Node Manager listens only on `localhost` and manages only its local HTTP Server. See About the Oracle HTTP Server Configuration for an Enterprise Deployment. Oracle recommends this approach instead of configuring Oracle HTTP Server instances as part of the application tier domain.

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 HTTPs 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 *Using Oracle WebLogic Server Proxy Plug-Ins*.

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

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

Oracle WebLogic Server 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 *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, Oracle FMW products, such as Oracle SOA Suite, 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.

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.

For scalability and manageability reasons, this Enterprise Deployment Guide uses Per Host Node manager configuration. The sections in the different chapters will provide the required steps for using a single Node manager in each host of the topology.

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 *High Availability Guide*
- One-to-Many Communication Using Unicast in *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 *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 Oracle FMW Products 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 *High Availability Guide*.

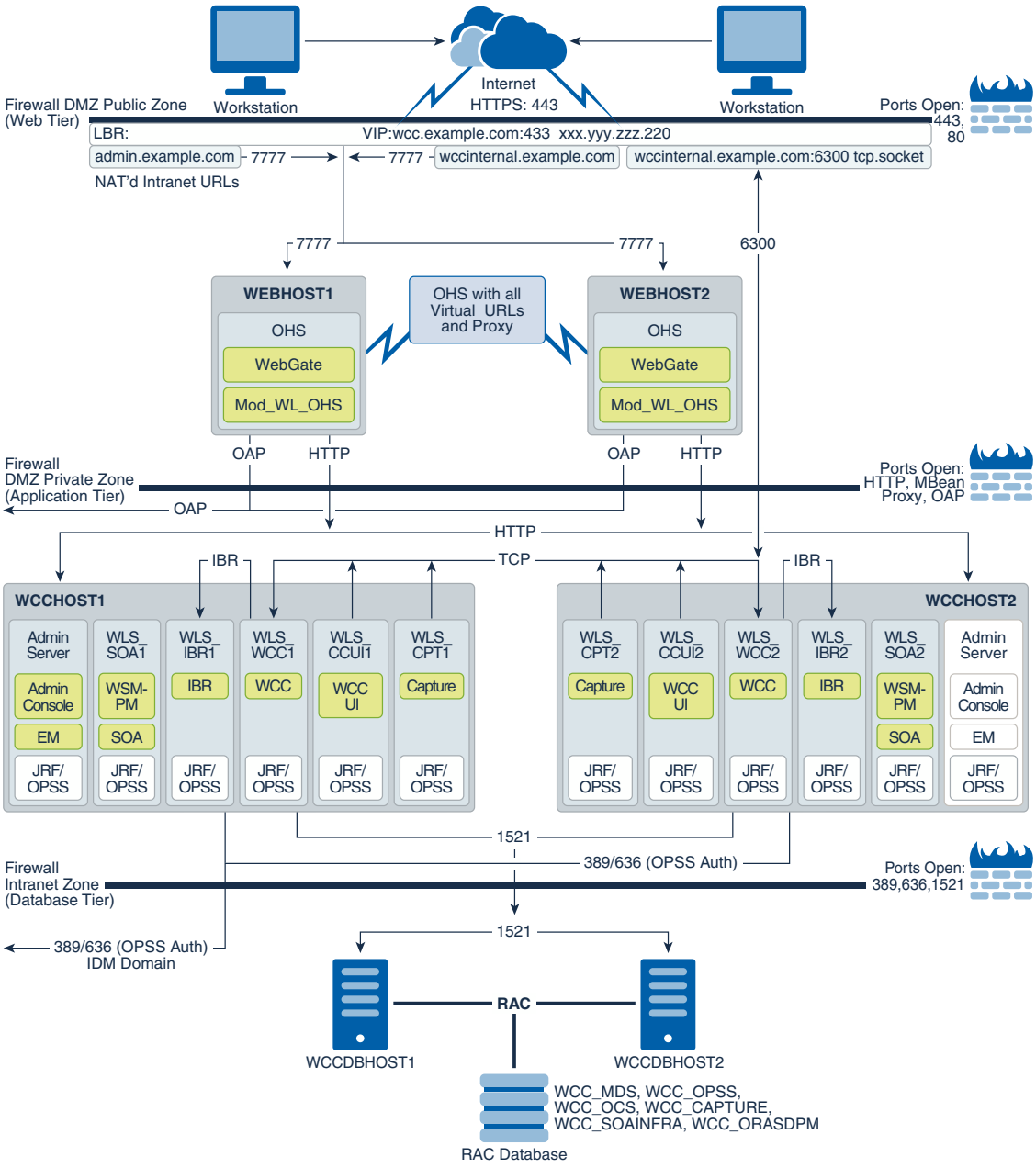
3

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.

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

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 Oracle Enterprise Manager Fusion Middleware Control and Oracle WebLogic Remote Console. As a result, clients access this service by using the following secure address:

  ```
  admin.example.com:445
  ```

  To provide an easier mapping between front-end host names and back-end OHS listeners, this guide presents different listeners (`443`, `444`, and `445`) in the front-end load balancer that map to the `443`, `444`, and `445` listeners in OHS. This allows easier segregation of traffic rules and SLAs for each type of access (LBR listener on `443` for apps, `444` for internal access and `445` for WLS domain administration). Alternatively, the same port (`443`) can be used in the front-end load balancer for all the different front-end host names. However, that OHS cannot host more than one SSL virtual host on the same IP address and port so it will still require separate listeners for each virtual host. For more information about the OHS configuration for the different products, see the following sections.

Instructions later in this guide explain how to:

- 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

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.

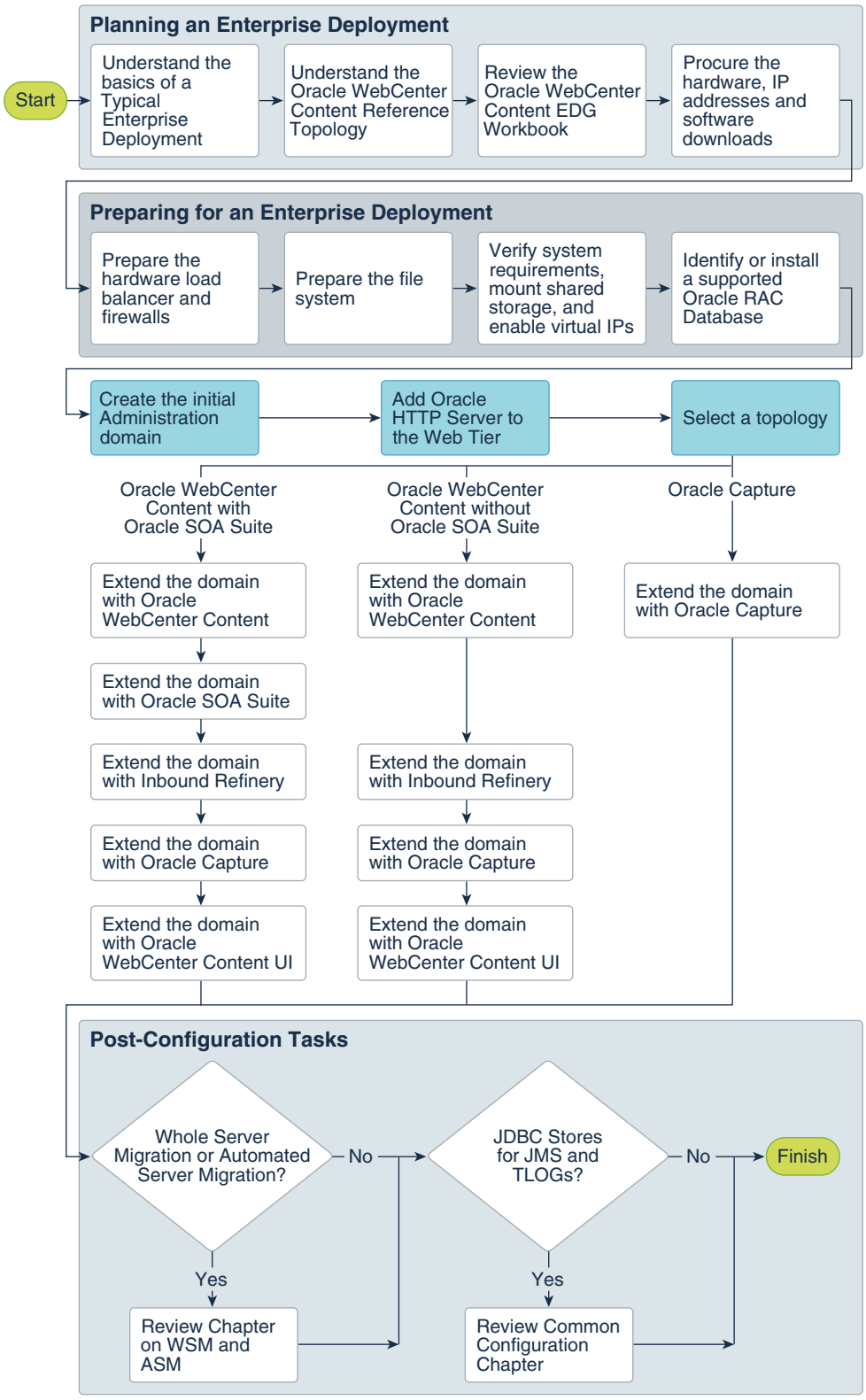
This guide is designed so you can start with a working WebCenter Content domain and then later extend the domain to add additional capabilities.

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

Flow Chart Step	More Information
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	#unique_67

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.

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).

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

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.

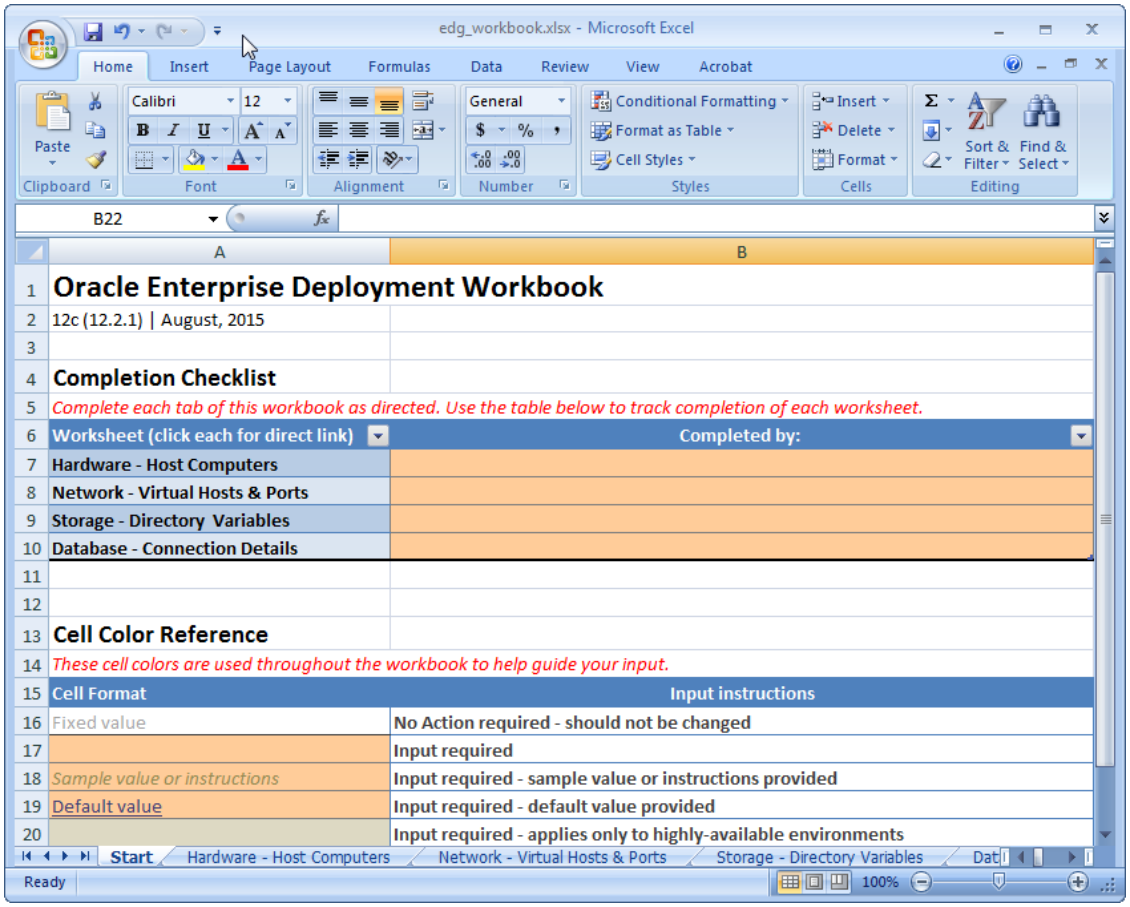
Using the Start Tab

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.

This section includes the following sections.

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 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.
- In this Enterprise Deployment Guide, SSL listeners are used for the Oracle HTTP Servers and Oracle WebLogic Servers. The load balancer should be able to establish SSL communication with the back-end servers in its pools.
- SSL acceleration (this feature is highly recommended, but not required for the enterprise topology).
- The ability to route TCP/IP requests; this is a requirement for Managed File Transfer, which can use sFTP/FTP protocol.

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.

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 14c (14.1.2.0.0) 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 and a virtual host name mapped to this VIP
- 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.

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 *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 Supported System Configurations. If there is a conflict of information between this guide and the certification matrices, then the information in the certification matrices must be considered the correct version, as they are frequently updated.

Distribution	Description
Oracle Fusion Middleware 14*c* (14.1.2.0.0) 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.

Distribution	Description
Oracle HTTP Server 14*c* (14.1.2.0.0)	Download this distribution to install the Oracle HTTP Server software on the Web Tier.
Oracle Fusion Middleware 14*c* (14.1.2.0.0) WebCenter Content	Download this distribution to install the Oracle WebCenter Content.
Oracle Fusion Middleware 14*c* (14.1.2.0.0) 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.

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

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. At the same time, this monitoring should not overload the back-end system with too frequent health requests. In the end, a trade off needs to be made between how fast the death detection occurs and how much overhead is introduced on the systems that are monitored.

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 load balancing between the web hosts, create a pool of servers that would direct requests to hosts WEBHOST1 and WEBHOST2 to each port used in the OHS. For example, a pool to WEBHOST1 and WEBHOST2 to port 4443 for access to applications like WebCenter Content, another pool to WEBHOST1 and WEBHOST2 to port 4444 for internal accesses, and another pool to WEBHOST1 and WEBHOST2 to port 4445 for access to administration consoles.

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

   c. Configure SSL for the virtual server.

   d. Configure SSL for the communication with the pool of servers.

Some load balancers may need to be provided with the back end's certificate (the SSL certificate used by the OHS listeners in the back-end pool) to establish the appropriate SSL communication. In that case, you may need to add the OHS's CA certificate to the load balancer as a trusted certificate. Because this guide uses example certificates based on the WebLogic per-domain CA, you can add this after the domain is created.

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?
admin.example.com:445	WEBHOST1.example.com:4445 WEBHOST2.example.com:4445	HTTPS	No
wcc.example.com:443	WEBHOST1.example.com:4443 WEBHOST2.example.com:4443	HTTPS	No
wccinternal.example.com:444	WEBHOST1.example.com:4444 WEBHOST2.example.com:4444	HTTPS	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 443. If port 80 is used for customer usability, then it is recommended to redirect any requests to it (non-SSL protocol) to port 443 (SSL protocol). Refer to your load balancer's specific documentation to implement this redirection.
- 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 `/management` and `/em` on this virtual server.

  These context strings direct requests to Oracle WebLogic Remote 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.

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 `/management` 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 **Note** You need this option only if redirection from port 80 to port 443 is	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
		used.			
Browser request	FW0	443	HTTPS / 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	FW1	80 **Note** You need this option only if redirection from port 80 to port 443 is	HTTPS / Load Balancer	Outbound (for intranet clients)	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
		used.			
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	HTTPS / 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	444x	HTTPS	n/a	n/a
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.
Database access	FW2	1521	SQL*Net	Both	Timeout depends on database content and on the type of process model used for WebCenter Content.
Coherence for deployment	n/a	9991 Coherence requires the following connectivity between members: • Port 9991 for both UDP and TCP for both multicast and unicast configurations. • TCP port 7. • Ephemereal ports 32768-60999 for both udp and tcp.		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.

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
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	HTTPS / WLS_SOA*n*	Inbound	Timeout varies based on the type of process model used for SOA.
Oracle WebCenter Content Access	FW1	16201	HTTPS / WLS_WCC*n*	Inbound	Browser-based access. Configurable session timeouts.
Oracle WebCenter Enterprise Capture access	FW1	16401	HTTPS / 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	16201	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	16401	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.

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 *High Availability Guide*.

The recommendations in this chapter are based on the concepts and guidelines described in the *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.

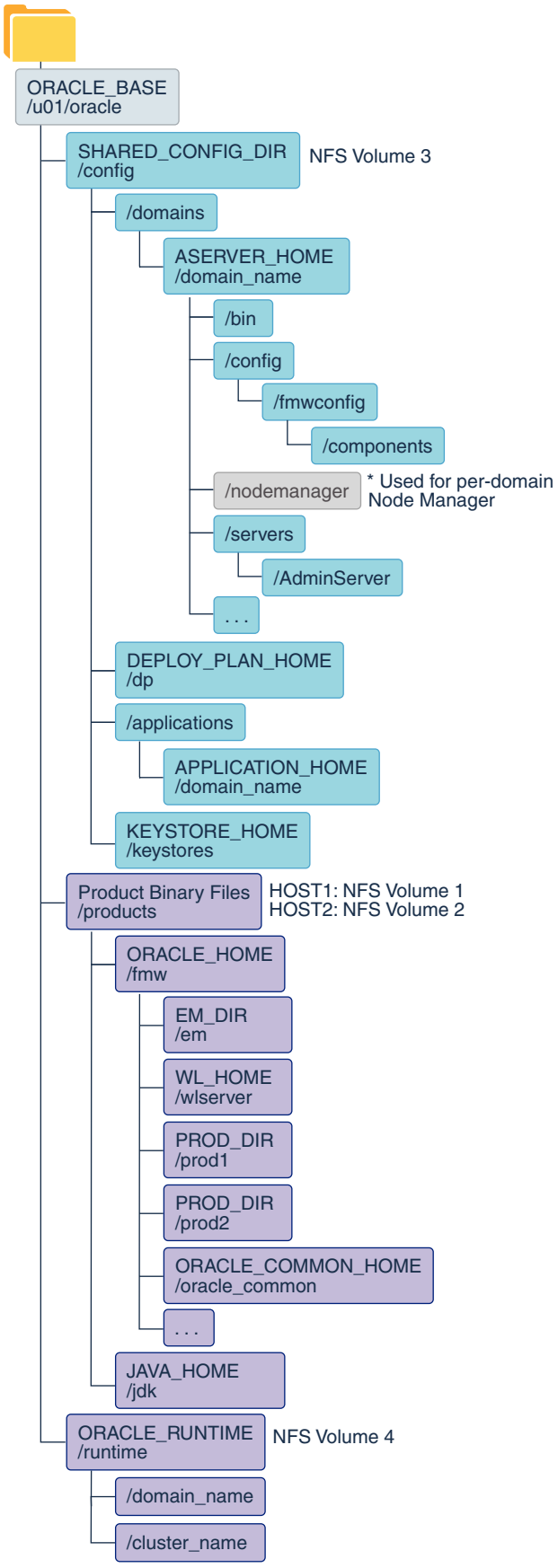
The directories shown in the diagrams contain binary files that are installed on disk by the Oracle Fusion Middleware installers, domain-specific files generated via the domain configuration process, as well as domain configuration files that are propagated to the various host computers via the Oracle 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.

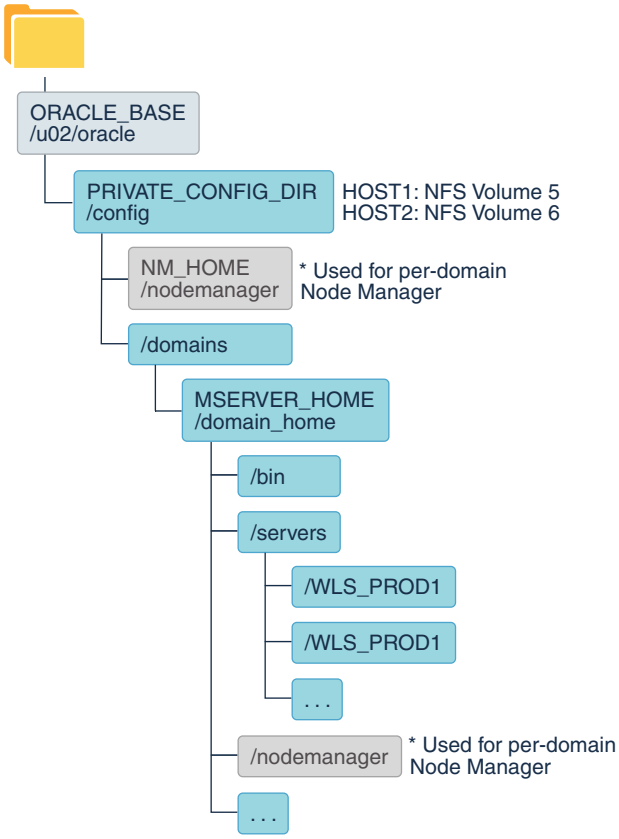
Where applicable, the diagrams also include the standard variables used to reference the directory locations in the installation and configuration procedures in this guide.

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



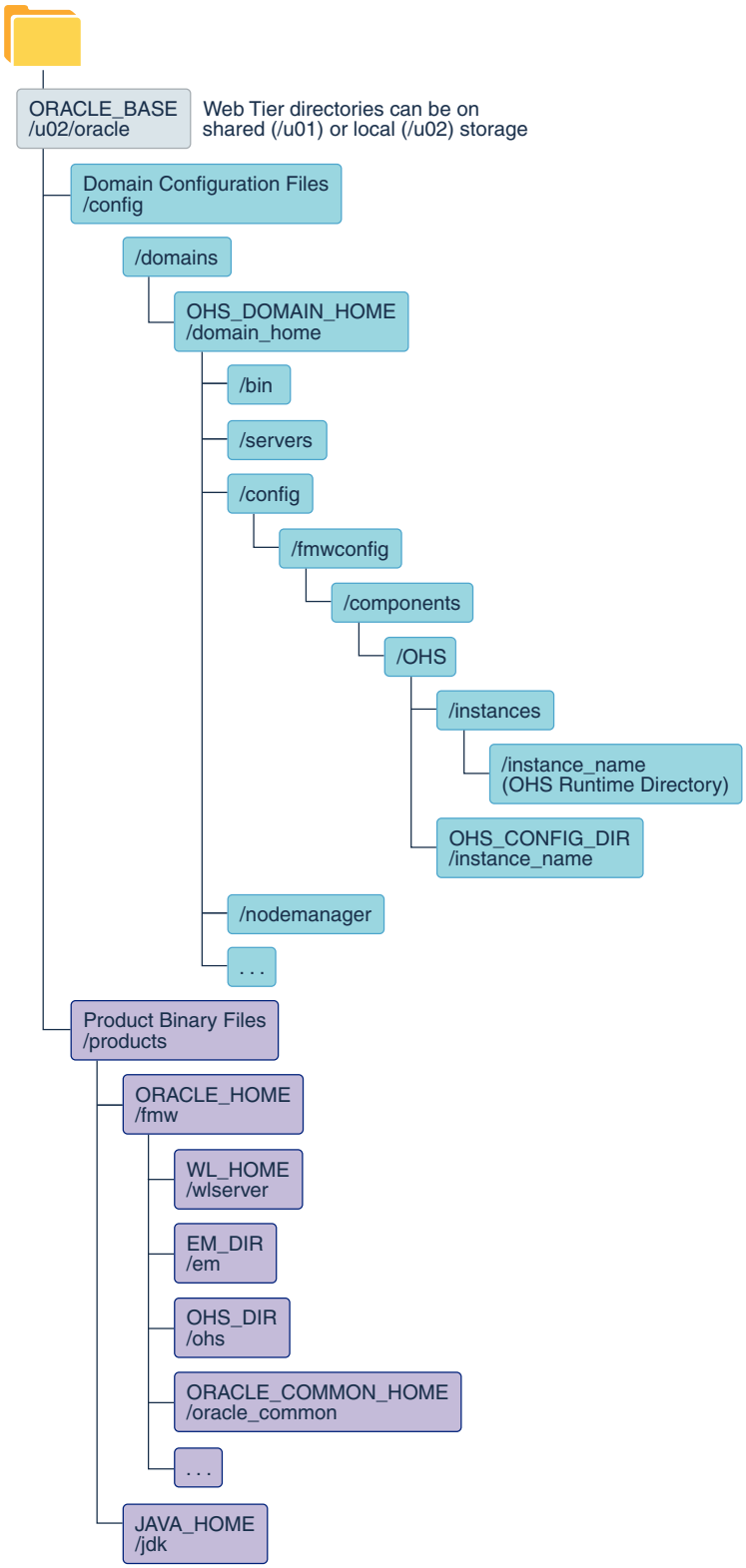
* See About the Node Manager Configuration in a Typical Enterprise Deployment.

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`
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`

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
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.
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.

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_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
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

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

Directory Variable	Description	Sample Value on the Web Tier
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.

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 *Administering Oracle HTTP Server*.
		`/u02/oracle/config/domains`
		`/`*`domain_name`*`/config/fmwconfig`
		`/components/OHS/`*`instance`*`/`
		`/`*`instance_name`*

WEB_KEYSTORE_HOME	If you use Oracle HTTP Server 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.

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

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. See the documentation for your version of the database. For example, 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

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**
   >
   > Ensure that these values are not overridden by any `.conf` file located in `/etc/security/limits.d/` folder.

4. Re-log in to the host computer.
5. Use the following commands to check the current values:

   ```
   echo "soft nofile = $(ulimit -S -n)"
   echo "hard nofile = $(ulimit -H -n)"
   echo "soft nproc = $(ulimit -S -u)"
   echo "hard nproc = $(ulimit -H -u)"
   ```

   Run these commands with user `root` and user `oracle` to check the effective values for each user.

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

You should configure the host to access your corporate DNS hosts.

To do this, update the DNS settings by updating the `/etc/resolv.conf` file.

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.

Configuring a Host to Use an NTP (time) Server

All servers in the deployment must have the same time. The best way to achieve this is to use an NTP server.

Since Oracle Linux 8 and Red Hat 8, the chrony daemon service replaces ntpd for the management of NTP. Chrony is a feature that implements NTP to maintain timekeeping accurately on the network.

To configure a host to use an NTP server with chrony:

1. Determine the name of the NTP servers you wish to use. For security reasons, ensure that these are inside your organization.
2. Log into the host as the root user.
3. Edit the file `/etc/chrony.conf` to include a list of the time servers. After editing, the file appears as follows:

```
server ntphost1.example.com
server ntphost2.example.com
```

4. Use the following systemctl command to check the status the Chrony daemon, chronyd:

   ```
   systemctl status chronyd
   ```

5. Use the following systemctl command to start or restart chronyd:

   ```
   systemctl restart chronyd
   ```

6. Run the following `chronyc -n tracking` command to check chrony tracking:

   ```
   chronyc -n tracking
   ```

7. Ensure the time is set correctly using the date command.
8. To ensure that the server always uses the NTP server to synchronize the time, set the client to start on reboot by using the following command:

   ```
   systemctl enable chronyd
   ```

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.

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

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. Pluggable databases (PDBs) are also supported for Oracle Fusion Middleware schemas, see Interoperability with Supported Databases in *Understanding Interoperability and Compatibility*.

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 *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 19*c*. 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 19*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*.
>
> If the database is a multitenant database, provide the pluggable database (PDB) name when creating the service so that the service is associated with the specified PDB. For example:
>
> ```
> srvctl add service -db wccdb -service wccedg.example.com -preferred
> wccdb1,wccdb2 -pdb PDB1
> ```

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 19*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*.

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`: The default setting prior to 12c. Allows SecureFile LOB storage when the `SECUREFILE` keyword is used. The default storage method is BasicFiles).
- `PREFERRED`: The default setting for 12c and later, which uses SecureFile LOB storage in all cases where LOB storage would otherwise default to BasicFile.
- `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.

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:

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.

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

- *ORACLE_HOME*
- *APPLICATION_HOME*
- *JAVA_HOME*
- *ASERVER_HOME*
- *MSERVER_HOME*
- *KEYSTORE_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

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 host 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

Oracle Fusion Middleware requires that a certified Java Development Kit (JDK) is installed 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:

https://www.oracle.com/java/technologies/downloads/

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 17.0.

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-17.0.10+11_linux-x64_bin.tar.gz
   ```

   > **Note**
   >
   > 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 jdk-17.0.10 /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 `17.0.10`.

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 -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_14.1.2.0.0_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 *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 *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 $ORACLE_HOME
```

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

```
bin
cfgtoollogs
coherence
em
install
inventory
jlib
lib
OPatch
opmn
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 *Creating Schemas with the Repository Creation Utility*.

Complete the following steps to install the required schemas:

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. 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
   ```

   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 *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 *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. As Database Type, select **Oracle Database enabled for edition-based redefinition**.

   > **Note**
   >
   > Oracle Database enabled for edition-based redefinition (EBR) is recommended to support Zero Down Time upgrades. For more information, see https://www.oracle.com/database/technologies/high-availability/ebr.html.

2. In the **Host Name** field, enter the SCAN address of the Oracle RAC Database.
3. Enter the **Port** number of the RAC database scan listener, for example `1521`.
4. Enter the RAC **Service Name** of the database.
5. Enter the **User Name** of a user that has permissions to create schemas and schema objects, for example SYS.
6. Enter the **Password** of the user name that you provided in step 4.
7. If you have selected the SYS user, ensure that you set the role to SYSDBA.
8. 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 *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 FMW1412_

   > **Tip**
   >
   > Make a note of the custom prefix that you choose to enter here; you will 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 *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 *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 *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.

> **Note**
>
> If the database is a pluggable database (PDB), the appropriate `tns` alias that points to the PDB must be used in the `sqlplus` command.

For example:

```
./sqlplus FMW1412_WLS/<WLS_schema_password>
```

```
SQL*Plus: Release 23.0.0.0.0 - for Oracle Cloud and Engineered Systems on Wed Sep 11
14:20:00 2024 Version 23.5.0.24.07
Copyright (c) 1982, 2024, Oracle. All rights reserved.

Enter user-name: FMW1412_WLS
Enter password: WLS_schema_password


Connected to:
Oracle Database 23ai EE Extreme Perf Release 23.0.0.0.0 - for Oracle Cloud and
Engineered Systems
Version 23.5.0.24.07

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

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 - [em]**

Selecting this template automatically selects the following dependencies:

  - **Oracle JRF - [oracle_common]**
  - **WebLogic Coherence Cluster Extension - [wlserver]**
- **Oracle WSM Policy Manager - [oracle_common]**

> **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 (Oracle recommends using a different user name from “WebLogic”) 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.

> **Note**
>
> Ensure that it points to the folder where you have installed the JDK. See Installing the JDK Software.

- In the Enable or Disable Default Ports for the Domain field, use the default values provided for Production Mode:
  - Ensure that **Enable Listen Ports (non-SSL Ports)** is NOT selected.
  - Ensure that **Enable SSL Listen Ports** is selected.
  - Ensure that **Enable Administration Port (SSL Port)** is selected.

> **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*.

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.

- Verify that Vendor is Oracle and Driver is *Oracle's Driver (Thin) for Service Connections; Versions: Any. Verify that Connection Parameters is selected.

> **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 Database Configuration Type for more information about the Database Configuration Type screen.

Field	Description
Host Name	Enter the Single Client Access Name (SCAN) Address for the Oracle RAC database, which you entered in the *Enterprise Deployment Workbook*. For information about the Enterprise Deployment Workbook, see Using the Enterprise Deployment Workbook .
DBMS/Service	Enter the service name for the Oracle RAC database appropriate for this domain where you will install the product schemas. For example: `wccedg.example.com` Specify the service name based on the value configured earlier in Preparing the Database for an Enterprise Deployment.
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 checkbox 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.

Element	Description and Recommended Value
SCAN, Host Name, and Port	Select the **SCAN** checkbox. 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	These values are not required when you are using an Oracle 12c database or higher versions because the ONS list is automatically provided from the database to the driver.
Enable Fan	Verify that the **Enable Fan** checkbox is selected, so the database can receive and process FAN events.

Element	Description and Recommended Value
Service Name	Verify that the service name for the Oracle RAC database is appropriate. For example, wccedg.example.com.

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 add, delete, or modify the Settings for Server Templates, Managed Servers, Clusters, Virtual Targets, and Coherence.

> **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. In the Configure Administration Server Ports section, perform the following steps:
   a. Leave the Enable Listen Port field unchecked. The Listen Port value will be disabled in grey.
   b. Ensure the Enable SSL Listen port field is checked.
   c. Leave the default value as 7002 in the SSL Listen Port field.
   d. Leave the default value as 9002 in the Administration Port.
4. Leave the default value as Unspecified in the Server Group.

Task 12 Configuring Node Manager
Select Manual Node Manager Setup as the Node Manager type.

> **Note**
>
> You can ignore the warning in the bottom pane. This guide provides the required steps for the Manual Node Manager configuration.

> **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 information about Node Manager configurations, see Configuring Node Manager on Multiple Machinesin *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 Type Node Manager Listen Port
ADMINHOST	Enter the value of the ADMINVHN variable.	SSL 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 AdminServer from the Server pane.
2. Select `ADMINHOST` from the Machines pane.
3. Click **Next** to proceed to the next screen.

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.

If you need to make any changes, you can go back to any previous screen by either by using the **Back** button or by selecting the screen in the navigation pane.
Domain creation will 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 *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.

Download and Configure WebLogic Remote Console

This section describes how to download and configure the WebLogic Remote Console.

> **Note**
>
> For the initial configuration steps required in this EDG, you will need access to the AdminServer listen address and administration port. Later on you will configure access from a frontend load balancer.

Perform the following steps to download and configure the WebLogic Remote Console:

1. Uninstall any previous versions of the WebLogic Remote Console from your computer.
2. Download the WebLogic Remote Console. Go to https://github.com/oracle/weblogic-remote-console/releases and download the installer from your operating system.
3. Run the installer.
4. Install the WebLogic Remote Console extension in the WebLogic Server domain. The WebLogic Remote Console extension provides additional functionality when using the WebLogic Remote Console to manage WebLogic domains.

   > **Note**
   >
   > This step is optional.

   a. Create a `management-services-ext` directory under the domain home.

b. Download the latest WebLogic Remote Console extension, `console-rest-ext-<version>.war`, from https://github.com/oracle/weblogic-remote-console/releases and save it inside the `management-services-ext` directory you created in the previous step. If you have an earlier version of the extension already downloaded, delete it and replace it with the latest version.

c. Reboot the Administration Server if it is already running.

5. Launch the WebLogic Remote Console application.

   Example:

   ```
   ./weblogic-remote-console
   ```

   In the next steps you must connect to the EDG domain provider using initially the Admin Servers listen address.

Configuring SSL Certificates for the Domain

This section describes how to configure SSL certificates for the domain.

Creating Certificates and Certificate Stores for the WebLogic Domain

The Enterprise Deployment Guide provides steps to configure a domain that uses SSL listen addresses for all Weblogic Managed Servers, Weblogic Administration Server and Node Managers in the application tier. To achieve this the required certificates for all servers, machines and NM listen addresses must be created and pointed to from the domain and Node Manager configuration.

In Oracle FMW 14.1.2.0, Oracle WebLogic allows the usage of a per-domain Certificate Authority (CA). In this model, the CertGen and ImportPrivateKey utilities are enhanced to use the domain's secret key to encrypt the passphrases and store them in the domain's DemoCerts.props file. A self-signed Demo CA is automatically created for the domain and it is used for signing certificates for the SSL listen addresses used in the EDG. Although in a real production system, standard CAs should be used, the per-domain CA model implements an SSL system using domain specific CA that provides a higher degree of protection than non-ssl configurations. If you want to use your own custom certificates, see About Using Third Party SSL Certificates in the WebLogic and Oracle HTTP Servers in the *Common Configuration and Management Tasks for an Enterprise Deployment* chapter.

Oracle recommends using a shared storage location (protected with the appropriate snapshot or file backup tooling) where all the different certificates and stores can be found by the different servers. Perform the following steps to generate an Identity store and a Trust Store that can be used for enabling SSL listeners in a Weblogic Server using a per-domain CA:

1. Download the `generate_perdomainCACERTS.sh` script in the maa github repo.

   https://github.com/oracle-samples/maa/blob/main/1412EDG/generate_perdomainCACERTS.sh

2. Run the script with the following arguments:

   - WLS_DOMAIN_DIRECTORY: Directory hosting the Weblogic Domain that the Administration Server uses.
   - WL_HOME: The directory within the Oracle home where the Oracle WebLogic Server software binaries are stored. Typically `/u01/oracle/products/fmw/wlserver`.
   - KEYSTORE_HOME: Directory where appIdentity and appTrust stores will be created.

- KEYPASS: Password used for the weblogic administration user (will be reused for certs and stores).

Example:

```
./generate_perdomainCACERTS.sh $ASERVER_HOME $ORACLE_HOME/
wlserver $KEYSTORE_HOME <keypass>
```

The script will traverse the `WLS_DOMAIN_DIRECTORY/config/config.xml` to find all the listen addresses used in the domain, generate certificates for all of them, create a trust store with the domain CA and import certificates into a new Identity store. The aliases used in the import will be the same as the hostname used as listen address. Both the trust store and the identity store will be placed in the `KEYSTORE_HOME` directory.

Run the following command to verify if the "domainca" entry is there as a `trustedCertEntry`:

```
keytool -list -keystore $KEYSTORE_HOME/appTrustKeyStore.pkcs12
```

Run the following command to verify if there is a `PrivateKeyEntry` for each listen address (the values for ADMINVHN, WCCHOST1, and WCCHOST2):

```
keytool -list -keystore $KEYSTORE_HOME/appIdentityKeyStore.pkcs12
```

Adding Certificate Stores Location to the WebLogic Servers Start Scripts

Once the Identity and Trust Stores are created for the domain some Java properties must be added to the WebLogic start scripts. These properties are added to the file `setUserOverridesLate.sh` in `$ASERVER_HOME/bin`. 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.

- If you created the Identity and Trust Stores with the script `generate_perdomainCACERTS.sh`, as explained in Creating Certificates and Certificate Stores for the WebLogic Domain, then the properties are automatically added to the file `setUserOverridesLate.sh` in `$ASERVER_HOME/bin`. Just verify that the file exists and that the `EXTRA_JAVA_PROPERTIES` have been added.
- If you are using your own custom certificates, then manually create the file `setUserOverridesLate.sh` in `$ASERVER_HOME/bin`. Edit the file and add the variable `EXTRA_JAVA_PROPERTIES` to set the `javax.net.ssl.trustStore` and `javax.net.ssl.trustStorePassword` properties with the values used by your EDG system. For example:

```
EXTRA_JAVA_PROPERTIES="${EXTRA_JAVA_PROPERTIES}
 -Djavax.net.ssl.trustStore=/u01/oracle/config/keystores/
appTrustKeyStore.pkcs12
 -Djavax.net.ssl.trustStorePassword=mypassword"
export EXTRA_JAVA_PROPERTIES
```

> **Note**
>
> The order of the extra java properties is relevant. In case that the same property is defined more than once, the later value is used. The custom values must be defined as in the example provided.

Update Server's Security Settings Using the Remote Console

Connecting to the Remote Console Using the Administration Server's Virtual Hostname as Provider

The following procedure temporarily starts the Administration Server with the default start script so to enable you to perform these tasks. After you perform these tasks, you can stop this temporary session and use the Node Manager to start the Administration Server.

> **Note**
>
> For this Remote Console initial access to the Administration Server, it is required that the machine that runs the Remote Console can resolve and connect to the Admin Server's Listen Address. This can be done by starting the Remote Console directly in the node where the Admin Server runs or creating a tunnel to this address from the node where the remote Console is executed.

1. Using the following default start script to start the Administration Server:
   a. Change directory to the following directory:

      ```
      cd $ASERVER_HOME/bin
      ```

   b. Run the start script:

      ```
      ./startWebLogic.sh
      ```

      Monitor the terminal till the following message is displayed:

      ```
      <Server state changed to RUNNING>
      ```

      Also you must verify that the appropriate SSL listener is available, which can be confirmed with the a message like the following displayed in output:

      ```
      <Server> <BEA-002613> <Channel "DefaultSecure" is now listening on
      XXXX:7002 for protocols iiops, t3s, ldaps, https.>
      ```

2. Create a new provider in the WebLogic Remote Console as follows:
   a. Download the domain's trust keystore to the host or laptop where you run the WebLogic Remote Console. For example, when using the per-domain CA steps in previous sections, this would be located at `$KEYSTORE_HOME/appTrustKeyStore.pkcs12`.

b. Open the Remote Console and add the domain trust store to the remote console settings. Click **File** > **Settings** and enter the following values.

   i. Trust Store type - jks

   ii. Trust Store Path - The path to the trust keystore file in the host where the Remote Console runs.

   iii. Trust Store Key - Enter the password provided in the steps above for certificate creation.

   iv. Check **Disable HostName verification** if you are using Demo certificates as described in the steps above.

c. Using the Providers window in the Remote Console, create a new provider by selecting **Add Admin Server Connection Provider**.

   i. In the provider name, enter the name of `wccedg_domain_asvip`. This will identify the type of access.

   ii. Enter the WebLogic Domain Administration username provided in the configuration wizard user name.

   iii. Enter the password used for the domain creation.

   iv. Use https protocol and the admin server listen address used in the configuration wizard as URL for access and specify port 9002.

   For example, `https://ADMINVHN.example.com:9002`.

   v. Check the **Make Insecure Connection** checkbox.

> **Note**
>
> This provider should not be used once the front end and webtier are configured.

The Remote Console Home Window for the domain will be displayed.

Updating the WebLogic Servers Security Settings

Perform the following steps to update the WebLogic Servers Security Settings and Administration Port:

1. Access the Domain provider in the Remote Console and update the Administration Server and WebLogic Servers Security Settings:

   a. Click **Edit Tree**.

   b. Click **Environment** > **Servers** > **AdminServer**.

   c. Click **Security** tab.

   d. Change the keystores dropdown to **Custom Identity and Custom Trust**.

   e. In **Custom Identity Keystore**, enter the fully qualified path to the identity keystore as follows:

   `KEYSTORE_HOME/appIdentityKeyStore.pkcs12`
   Replace `KEYSTORE_HOME` with the value of the folder you use for storing keystore, as described in the Table 7-2.

   f. Set the **Custom Identity Keystore Type** to JKS.

> **Note**
>
> Specifying JKS or PKCS12 is valid both for pkcs12 and jks stores. Both formats can be read and managed if the Customer Key Store Type is set to "JKS".

g. In **Custom Identity Keystore Passphrase**, enter the password `Keystore_Password` you provided in the certificate generation steps.

h. In **Custom Trust Keystore**, enter the fully qualified path to the trust keystore.

`KEYSTORE_HOME/appTrustKeyStore.pkcs12`
Replace `KEYSTORE_HOME` with the value of the folder you use for storing keystore, as described in the Table 7-2.

i. Set the **Custom Trust Keystore Type** to JKS.

> **Note**
>
> Specifying JKS or PKCS12 is valid both for pkcs12 and jks stores. Both formats can be read and managed if the Customer Key Store Type is set to "JKS".

j. In **Custom Trust Keystore Passphrase**, enter the password you provided as the <**keypass**> in the certificate generation steps.

k. Click **Save**.

l. Under **Security** settings, navigate to **SSL** tab.

m. In the **Server Private Key Alias** filed enter the alias provided in the certificate generation steps. If you used the certificate generation script this will be the same as the listen address used for the WLS server.

n. In the **Server Private Key Pass Phrase** field, enter the password provided in the certificate generation steps. If you used the certificate generation script this will be the same as the keystore passphrase.

o. Click **Save**.

The cart on the top right part of the screen will show **full** with a yellow bag inside.

p. Click the Cart icon on the top right and select **Commit Changes**.

Repeat the above steps for each managed server in the domain changing the alias to match the alias used for the certificates.

2. Return to the terminal window where you started the Administration Server with the start script.

3. Press **Ctrl+C** to stop the Administration Server process.

Wait for the Administration Server process to end and for the terminal command prompt to appear.

4. Start the Administration Server again by using the following script:

a. Change directory to the following directory:

```
cd $ASERVER_HOME/bin
```

b. Run the start script:

```
./startWebLogic.sh
```

c. Monitor the output in the terminal till the following output is displayed.

```
<Server state changed to RUNNING>
```

Configuring KSS with Per-domain CA

For consistency purposes and to use a common CA all across the domain artifacts you may want to use the per-domain CA for KSS (store used by OPSS and other components in the WebLogic Infrastructure/JRF Domain.

Perform the following steps to import the domain CA certificate in the KSS trusted store:

1. Download the `import-domainca-into-kss.sh` script in the maa github repo *https://github.com/oracle-samples/maa/blob/main/1412EDG/import-domainca-into-kss.sh*.
2. Edit the script and customize the following variables according to your environment:

   DOMAIN_HOME: Path to the WebLogic domain (ASERVER_HOME in this guide). For example, `/u01/oracle/config/domains/wccedg_domain`.

   MW_HOME: The path to the FMW home. For example, `/u01/oracle/products/fmw`.

   ADMINVHN: Administration Server's listen address. For example, *adminvhn.example.com*.

   ADMINPORT: Administration Server's listen port. For example, 9002.

   DOMAINUSER: Name of the administrator user for the WLS domain. For example, wccedgadmin.

   TRUSTSTOREFILE: Location of the trustore used to connect though SSL to the Admin Server. For example, `/u01/oracle/config/keystores/appTrustKeyStore.pkcs12`.
3. Run the script with the following arguments:
   - DOMAINPASS: WLS domain administrator user's password
   - KEYPASS: Password for the truststore.

   Example

   ```
   ./import-domainca-into-kss.sh adminpassword123 truststorepassword123
   ```

   The script imports the per Domain CA certificate into KSS and assigns it to jps.

   You can verify that the update was successful by inspecting the jps configuration files.

   ```
   grep domainca $ASERVER_HOME/config/fmwconfig/jps-config.xml
   ```

   The result of the command must be similar to the following example:

   ```
   <property name="ca.key.alias" value="domainca-new-24-05-07-16-44-52"/>
   ```
4. Restart the Admin Server.

   If Admin Server was started with the script, perform the following steps:

   a. Press **Ctrl+C** to stop the Administration Server process.

   b. Go to directory `$ASERVER_HOME/bin` and run the following command:

```
./startWebLogic.sh
```

Configuring a Per Host Node Manager for an Enterprise Deployment

For specific enterprise deployments, Oracle recommends that you configure a per-host Node Manager, as opposed to the default per-domain Node Manager.

For more information about the advantages of a per host Node Manager, see About the Node Manager Configuration in a Typical Enterprise Deployment

Creating a Per Host Node Manager Configuration

The step in configuring a per-host Node Manager is to create a configuration directory and two new node manager configuration files. You must also edit the default `startNodeManager.sh` file.

To create a per-host Node Manager configuration, perform the following tasks, first on WCCHOST1, and then on WCCHOST2:

1. Log in to WCCHOST1 and create a directory for the Node Manager configuration files :

   For example:

   ```
   mkdir -p /u02/oracle/config/nodemanager
   ```

   Note that this directory should be on a local disk, because it is specific to the host. This directory location is known as the Node Manager home, and it is identified by the *NM_HOME* directory variable in examples in this guide.

2. Change directory to the Node Manager home directory:

   ```
   cd $NM_HOME
   ```

3. Create a new text file called `nodemanager.properties` and add the values shown in Example: Contents of the nodemanager.properties File to this new file.

   You must repeat similar configuration for the Node Managers in the other node of the domain (WCCHOST2, use the pertaining certificate alias).

   Use the pertaining identity alias for the node that you are configuring. For example, `wcchost1.example.com` in WCCHOST1 and `wcchost2.example.com` in WCCHOST2.

   For more information about the properties that you can add to the `nodemanager.properties` file, see Node Manager Properties in *Administering Node Manager for Oracle WebLogic Server*.

   In the `nodemanager.properties` file, you enable crash recovery for servers as a part of this configuration. See Node Manager and System Crash Recovery in *Administering Node Manager for Oracle WebLogic Server*.

   Example: Contents of the nodemanager.properties File

   ```
   DomainsFile=/u02/oracle/config/nodemanager/nodemanager.domains
   LogLimit=0
   PropertiesVersion=14.1.2.0.0
   AuthenticationEnabled=true
   NodeManagerHome=/u02/oracle/config/nodemanager
   #Include the specific JDK home
   JavaHome=/u01/oracle/products/jdk
   ```

```
LogLevel=INFO
DomainsFileEnabled=true
StartScriptName=startWebLogic.sh
#Leave blank for listening on ANY
ListenAddress=
NativeVersionEnabled=true
ListenPort=5556
LogToStderr=true
SecureListener=true
LogCount=1
StopScriptEnabled=false
QuitEnabled=false
LogAppend=true
StateCheckInterval=500
CrashRecoveryEnabled=true
StartScriptEnabled=true
LogFile=/u02/oracle/config/nodemanager/nodemanager.log
LogFormatter=weblogic.nodemanager.server.LogFormatter
ListenBacklog=50
KeyStores=CustomIdentityAndCustomTrust
CustomIdentityAlias=wcchost1.example.com
CustomIdentityKeyStoreFileName=/u01/oracle/config/keystores/
appIdentityKeyStore.pkcs12
CustomIdentityKeyStorePassPhrase=password
CustomIdentityPrivateKeyPassPhrase=password
```

Notice the values for CustomIdentityAlias. If you used the `generate_perdomainCACERTS.sh` script, this is the hostname used as listen address in the configuration wizard for the Node Manager Machine. If you created the certificates one by one, this would be the alias that you assigned to the certificate import for WCCHOST1. You must also provide the location of the IdentityStore generated in the previous steps and the password for the same.

4. Locate the `startNodeManager.sh` file in the following directory:

   `$WL_HOME/server/bin`

5. Copy the `startNodeManager.sh` file to the Node Manager home directory.

   ```
   cp $WL_HOME/server/bin/startNodeManager.sh $NM_HOME
   ```

6. Edit the new `startNodeManager.sh` file and add the `NODEMGR_HOME` property as follows:

   `NODEMGR_HOME="NM_HOME"`

   In this example, replace *NM_HOME* with the actual path to the Node Manager home.

7. Locate the `stopNodeManager.sh` script in the `WL_HOME/server/bin` directory. Copy it to the Node Manager home directory. Edit the copied file and edit the *NODEMGR_HOME* property pointing to the node manager home (as it has been done for the `startNodemanager.sh` file):

   `NODEMGR_HOME="NM_HOME"`

   In this example, replace *NM_HOME* with the actual path to the Node Manager home.

8. Create another new file in the Node Manager home directory, called `nodemanager.domains`.

The `nodemanager.domains` file provides additional security by restricting Node Manager client access to the domains listed in this file.

9. Perform steps 1 through 8 on WCCHOST2.
10. Add the following entries to the new `nodemanager.domains` files:

    On WCCHOST1, add values for both the Administration Server domain home and the Managed Servers domain home:

    ```
    wccedg_domain=MSERVER_HOME;ASERVER_HOME
    ```

    > **Note**
    >
    > The path that is mentioned first (*MSERVER_HOME*) is considered as the primaryDomainPath and Managed Servers are run from this location.

    On WCCHOST2, add the value for the Managed Servers domain home only:

    ```
    wccedg_domain=MSERVER_HOME
    ```

    In these examples, replace *ASERVER_HOME* and *MSERVER_HOME* with the values of the respective variables, as described in File System and Directory Variables Used in This Guide.

Starting the Node Manager on WCCHOST1

After you manually set up the Node Manager to use a per-host Node Manager configuration, you can start the Node Manager on WCCHOST1, by using the `startNodeManager.sh` script.

To start the Node Manager on WCCHOST1:

1. Change directory to the Node Manager home directory:

   ```
   cd $NM_HOME
   ```

2. Run the following command to start the Node Manager and send the output of the command to an output file, rather than to the current terminal shell:

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

3. Monitor the the `nodemanager.out` file; make sure the NodeManager starts successfully. The output should eventually contain the following strings:

   ```
   <INFO> <Upgrade> <Encrypting NodeManager property:
   CustomIdentityKeyStorePassPhrase>
   <INFO> <Upgrade> <Encrypting NodeManager property:
   CustomIdentityPrivateKeyPassPhrase>
   <Upgrade> <Saving upgraded NodeManager properties to '/u02/oracle/config/
   nodemanager/nodemanager.properties'>
   <INFO> <Loading domains file: /u02/oracle/config/nodemanager/
   nodemanager.domains>
   <INFO> <Loading identity key store: FileName=/u01/oracle/config/keystores/
   appIdentityKeyStore.pkcs12, Type=pkcs12, PassPhraseUsed=true>
   ```

```
<INFO> <Loaded NodeManager configuration properties from '/u02/oracle/
config/nodemanager/nodemanager.properties'>
<INFO> <14.1.2.0.0>
<INFO> <Server Implementation Class:
weblogic.nodemanager.server.NMServer$ClassicServer.>
<INFO> <Secure socket listener started on port 5556>
```

You must check that the plain text used for passwords in `nodemanager.properties` has now been encrypted:

```
[oracle@wcc1onhost1 keystores]$ cat /u02/oracle/config/nodemanager/
nodemanager.properties
#Tue Feb 06 11:53:10 GMT 2024
#Mon Feb 05 17:24:30 GMT 2024
DomainsFile=/u02/oracle/config/nodemanager/nodemanager.domains
LogLimit=0
PropertiesVersion=14.1.2.0.0
AuthenticationEnabled=true
NodeManagerHome=/u02/oracle/config/nodemanager
#Include the specific JDK home
JavaHome=/u01/oracle/products/jdk
LogLevel=INFO
DomainsFileEnabled=true
StartScriptName=startWebLogic.sh
#Leave blank for listening on ANY
ListenAddress=
NativeVersionEnabled=true
ListenPort=5556
LogToStderr=true
SecureListener=true
LogCount=1
StopScriptEnabled=false
QuitEnabled=false
LogAppend=true
StateCheckInterval=500
CrashRecoveryEnabled=true
StartScriptEnabled=true
LogFile=/u02/oracle/config/nodemanager/nodemanager.log
LogFormatter=weblogic.nodemanager.server.LogFormatter
ListenBacklog=50
KeyStores=CustomIdentityAndCustomTrust
CustomIdentityAlias=wcchost1.example.com
CustomIdentityKeyStoreFileName=/u01/oracle/config/keystores/
appIdentityKeyStore.pkcs12
CustomIdentityKeyStorePassPhrase={AES256}EMvPrOCRfN7fyv3d8JcEnttTLyneG9Su+U
VK5DGEmqmqDwLkpLz9nQFZ+fL1Bidc
CustomIdentityPrivateKeyPassPhrase={AES256}O5cEJD8WVYP3aRLp9KAbFZ3CGLyxmmIW
FX1YzVfJvPpl1dc5RbMksAcsBLquKcWW
```

Starting the Node Manager on WCCHOST2

To start the Node Manager on WCCHOST2, follow the steps in Starting the Node Manager on WCCHOST1.

Configuring the Node Manager Credentials

Perform the following steps to set the Node Manager credentials using the Remote Console:

1. Access the Domain provider in the Remote Console.
2. Click **Edit Tree**.
3. Click **Environment** > **Domain**> **Security**.
4. Check the **Show Advanced Fields** field.
5. Set **Node Manager Username** to the same as the Weblogic Administrator, since this username will be used in other tasks mentioned in this guide.
6. Change the NM password. Ensure the **Node Manager password** is set to the same as the Weblogic Administrator since this password will be used in other tasks mentioned in this guide.
7. Click **Save**. The cart on the top right part of the screen will show **full** with a yellow bag inside.
8. Click the Cart Icon on the top right and select **Commit Changes**.

Enrolling the Domain with NM

Perform the following steps in a new terminal window to enroll the domain with Node manager.

> **Note**
>
> You will be unable to connect to the Node Manager and use it to start the servers in the domain without performing this step.

1. Change directory to the following directory:

   ```
   cd $ORACLE_COMMON_HOME/common/bin
   ```

2. Start the WebLogic Server Scripting Tool (WLST). In order to use the certificates created for the appropriate SSL handshake, the location of the stores and password of the same need to be provided to WLST. Use the following command or add these in a script that can be easily invoked:

   ```
   export WLST_PROPERTIES="-Dweblogic.security.TrustKeyStore=CustomTrust -
   Dweblogic.security.CustomTrustKeyStoreFileName=/u01/oracle/config/
   keystores/appTrustKeyStore.pkcs12 -
   Dweblogic.security.CustomTrustKeyStorePassPhrase=storepassword"
   ./wlst.sh
   ```

   > **Note**
   >
   > You must avoid including the password in the script.

3. Connect to the Administration Server by using the following WLST command:

   ```
   connect('admin_user','admin_password','admin_url')
   ```

   For example:

   ```
   connect('weblogic','<password>','t3s://ADMINVHN:9002')
   ```

4. Use the `nmEnroll` command to enable the Node Manager to manage servers in a specified WebLogic domain.

   ```
   nmEnroll('ASERVER_HOME')
   ```

   For example:

   ```
   nmEnroll('/u01/oracle/config/domains/wccedg_domain')
   ```

5. Generate startup properties for the Admin Server using the following WLST command:

   ```
   nmGenBootStartupProps('AdminServer')
   ```

   The `startup.properties` and `boot.properties` files are created in the following directory:

   ```
   $ASERVER_HOME/servers/AdminServer/data/nodemanager/
   ```

Adding Truststore Configuration to Node Manager

It is required to add the corresponding truststore configuration for Node Manager communication with the different WebLogic Server listeners. To do this, edit Node Manager's start script `startNodeManager.sh` located at `$NM_HOME` and add the variable *JAVA_OPTIONS* to set the `javax.net.ssl.trustStore` and `javax.net.ssl.trustStorePassword` properties with the values used by your EDG system. For example:

```
export JAVA_OPTIONS="${JAVA_OPTIONS} -Djavax.net.ssl.trustStore=/u01/oracle/
config/keystores/appTrustKeyStore.pkcs12 -
Djavax.net.ssl.trustStorePassword=mypassword"
```

> **Note**
>
> If you have used the `generate_perdomainCACERTS.sh` script to generate certificates and stores, the `trustStorePassword` is the password provided as "`KEYPASS`" parameter to the script.

Configuring the Domain Directories and Starting the Servers on WCCHOST1

After the domain is created and the node manager is configured, you can then configure the additional domain directories and start the Administration Server and the Managed Servers on WCCHOST1.

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):

```
export WLST_PROPERTIES="
-Dweblogic.security.TrustKeyStore=CustomTrust
-Dweblogic.security.CustomTrustKeyStoreFileName=/u01/oracle/config/keystores/
appTrustKeyStore.pkcs12
-Dweblogic.security.CustomTrustKeyStorePassPhrase=password"

cd $ORACLE_COMMON_HOME/common/bin

./wlst.sh
```

> **Note**
>
> The `weblogic.security.SSL.ignoreHostnameVerification=true` is required when using Demo certificates as the ones provided by the `generateCertifcates` scripts. In an environment with formal CA and certificates, this flag should not be used.

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

```
nmConnect('nodemanager_username','nodemanager_password','ADMINVHN','5556','
domain_name','ASERVER_HOME','SSL')
```

> **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 Remote 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:

`https://ADMINVHN:9002/em`

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

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.

   `https://ADMINVHN:9002/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. Set the following properties:
   - `virtualize` with value `true`
   - `optimize_search` with value `true`

   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 *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

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`

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=com
> 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 Remote 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 the **New** button below the **Authentication Providers** table.

6. 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

7. 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

8. Click **OK** to return to the Providers screen.

9. On the Providers screen, click the newly created authenticator in the table.

10. 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**.

11. Click **Save** to save the control flag settings.

12. Click the **Provider Specific** tab and enter the details specific to your LDAP server, as shown in the following table.

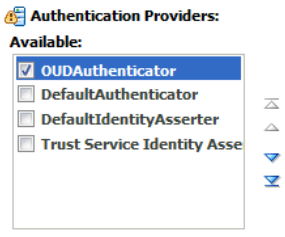
    Note that only the required fields are discussed in this procedure. For information about all the fields on this page, consider the following resources:

    - 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 *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.

Parameter	Sample Value	Value Description
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))`
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.

13. Click **Save** to save the changes.
14. 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.
15. Click **Reorder**, and then use the resulting page to make the Provider you just created first in the list of authentication providers.



16. Click **OK**.
17. On the Providers Page, click **DefaultAuthenticator**.

18. From the Control Flag drop-down, select **SUFFICIENT**.
19. Click **Save** to update the DefaultAuthenticator settings.
20. In the Change Center, click **Activate Changes**.
21. Restart the Administration Server and all managed servers.

    To stop the Managed Servers, sign in to Fusion Middleware Control, select the Managed Servers in the Target Navigator and click **Shut Down** in the toolbar.

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

    a. Start WLST:

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

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

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

    c. Stop the Administration Server:

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

    d. Start the Administration Server:

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

    e. Exit WLST:

    ```
    exit()
    ```

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','SSL')
    ```

    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. In the WebLogic Remote Console, go to **Security Tree**.
   b. Navigate to **Realms** > **myrealm** > **Authentication Providers**.
   c. Expand the new Authentication Provider.
   d. Click **Users** and verify if the administrator user that you provisioned is listed.
   e. Click **Groups** and verify if the administrator group that you provisioned is listed.

Adding the Administration Role to the New Administration Group

After you add the users and groups to your LDAP 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 Remote 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. Select the **Security Data Tree**.

3. Then select **Realms** > *myrealm* > **Role Mappers** > **XACMLRoleMapper** > **Global** > **Roles**.
4. Click the **Admin** role.
5. Click **Add conditions**.
6. Select **Group** from the **Predicate List** drop-down menu, and then click **Next**.
7. Enter `WCCAdministrators` in the **Group Argument Name** field, and then click **OK**.

   `WCCAdministrators` is added to the list box of arguments.
8. Click **Save** to finish adding the **Admin** Role to the `WCCAdministrators` group.
9. Validate that the changes were made by logging in to the WebLogic Remote Console and in to Fusion Middleware Control by using the new `weblogic_wcc` user credentials.

   If you can log in to the Oracle WebLogic Remote 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.

Backing Up the Configuration

After you successfully extended a domain or at another logical point, it is an Oracle best practices recommendation to create a backup.

Create a backup after you verify that the installation is successful so far. This is a quick backup for the purpose of immediate restoration in case of problems in later steps.

Verification of Manual Failover of the Administration Server

After you configure the domain, you should test the failover.

11

Configuring the Web Tier for an Enterprise Deployment

For an enterprise deployment, Oracle HTTP Server must be installed on each of the web tier hosts and configured as Oracle HTTP standalone domains on each host.

In this enterprise deployment, the LBR communicates with OHS over SSL protocol for a more secure configuration. The OHS instances also communicate over SSL protocol with the specific Managed Servers in the application tier. SSL is configured all the way from the LBR to the backend WLS servers.

Before you configure Oracle HTTP Server, be sure to review Understanding the Web Tier.

> **Note**
>
> As of Fusion Middleware 14.1.2.0.0, Oracle Traffic Director has been deprecated. For an enterprise deployment, use Oracle HTTP Server.

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.

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

- *WEB_ORACLE_HOME*
- *WEB_DOMAIN_HOME*
- *WEB_KEYSTORE_HOME*
- *JAVA _HOME*

In addition, you'll be referencing the following virtual IP (VIP) address and host names:

- ADMINVHN
- WEBHOST1
- WEBHOST2
- WCCHOST1
- WCCHOST2

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 simplified management that requires a minimum amount of configuration and a minimum amount of resources to run and maintain. Contrary to the App tier, Node Managers in the Web Tier listen on plain sockets because they are only accessed locally (they listen on `localhost` only).

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

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:

https://www.oracle.com/java/technologies/downloads/

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 17.0.10:

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-17.0.10+11_linux-x64_bin.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 ./jdk-17.0.10 /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 -version
   ```

   The Java version in the output should be displayed as `17.0.10`.

6. Repeat steps 1 through 5 for each web tier host, for example, WEBHOST1 and WEBHOST2.

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

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_14.1.2.0.0_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 *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.
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.

Table 11-1 (Cont.) Oracle HTTP Server Installation Screens

Screen	Description
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 *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:

```
assistants
bin
cfgtoollogs
clone
crs
crypto
css
cv
deinstall
drdaas
env.ora
has
hs
install
instantclient
inventory
javavm
jdbc
jlib
jpub
ldap
lib
network
nls
odbc
ohs
olap
```

```
OPatch
opmn
oracle_common
oracore
oraInst.loc
ord
oss
oui
perl
plsql
plugins
precomp
QOpatch
racg
rdbms
root.sh
schagent.conf
sdk
slax
sqlcl
sqlj
sqlplus
srvm
suptools
ucp
unixODBC
usm
utl
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

> **Note**
>
> In previous chapters, ssl store customizations were added to the `setDomainEnv.sh` scripts in the domain. These customizations are overwritten by the configuration wizard in every domain extension operation. To customize server startup parameters that apply to all servers in a domain, create a file called `setUserOverridesLate.sh` and add in it customizations such as custom libraries, additional JAVA command line options for running the servers, or additional environment variables. Any customizations added to this file are preserved during domain extension operations, and are carried over to remote servers when using the **`pack`** and **`unpack`** commands.

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.

> **Important**
>
> - More information about the Domain home directory can be found in About the Domain Home Directory in *Planning an Installation of Oracle Fusion Middleware*.
> - 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*.
> - For more information about the web tier and the DMZ, see Understanding the Firewalls and Zones of a Typical Enterprise Deployment.
> - For more information about the WEB_DOMAIN_HOME directory variable, see File System and Directory Variables Used in This Guide.

Task 2 Selecting the Configuration Templates
On the Templates screen, select **Oracle HTTP Server (Standalone) - [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.
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 the value of *`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 *Creating WebLogic Domains Using the Configuration Wizard*.
> For information about Node Manager configuration, see Configuring Node Manager on Multiple Machines in *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 *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

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:

   ```
   cd $WEB_DOMAIN_HOME/nodemanager
   ```

2. Modify nodemanager.properties to not use secure listener. Ensure it uses the localhost only as listen address. Your `$WEB_DOMAIN_HOME/nodemanager/nodemanager.properties` should appear like the following:

```
#Mon Feb 26 18:12:34 GMT 2024
#Node manager properties
#Mon Feb 26 18:03:35 GMT 2024
LogAppend=true
DomainsFile=/u02/oracle/config/domains/wccedgohs/nodemanager/
nodemanager.domains
LogLevel=INFO
PropertiesVersion=14.1.2.0.0
ListenBacklog=50
QuitEnabled=false
LogCount=1
LogLimit=0
NodeManagerHome=/u02/oracle/config/domains/wccedgohs/nodemanager
LogToStderr=true
NativeVersionEnabled=true
AuthenticationEnabled=true
CrashRecoveryEnabled=false
weblogic.StopScriptEnabled=false
DomainsFileEnabled=true
weblogic.StartScriptEnabled=true
LogFile=/u02/oracle/config/domains/wccedgohs/nodemanager/nodemanager.log
LogFormatter=weblogic.nodemanager.server.LogFormatter
ListenAddress=localhost
JavaHome=/u02/oracle/products/jdk
weblogic.StartScriptName=startWebLogic.sh
ListenPort=5556
SecureListener=false
StateCheckInterval=500
```

3. 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 &
```

4. Log in to WEBHOST2 and perform steps 1 and 2.

See Advanced Node Manager Configuration in *Administering Node Manager for Oracle WebLogic Server*.

Starting the Oracle HTTP Server Instances

To start the Oracle HTTP Server instances:

1. To start the ohs1 instance in WEBHOST1, enter the following commands:

```
$WEB_ORACLE_HOME/oracle_common/common/bin/wlst.sh

wls:/offline>
nmConnect('ohsdomain_admin_user','ohsdomain_admin_password','localhost','55
56','ohsdomain_name','WEB_DOMAIN_HOME','PLAIN')
```

```
wls:/nm/wccedgohs> nmStart(serverName='ohs1',serverType='OHS')
```

2. Repeat *Step 1* to start the `ohs2` instance on WEBHOST2. See Starting Oracle HTTP Server Instances in *Administering Oracle HTTP Server*.
3. Check the logs in each node at `$WEB_DOMAIN_HOME/servers/ohs1/logs/ohs1.log`.

   This will allow you to validate the appropriate start of OHS on a non-ssl listener. The following steps will guide you through the process for using the SSL listeners for OHS and routing to WLS using SSL.

Setting Front-end Addresses and WebLogic Plug-in for the Administration Server

As a security best practice oracle recommends setting a front-end address for the Administration Server. In the initial domain creation steps, since OHS and the front-end Load Balancer may have not been configured yet, the front-end setting is avoided to allow verifications using the individual server addresses. However, at this point and before configuring OHS (and the front-end load balancer, if not done yet) it is required to add the pertaining addresses.

1. To set the front end and WebLogic Plug-in for the Administration Server, use the WebLogic Remote Console as follows:
   a. Click **Edit Tree**.
   b. Click **Environment**>**Servers**>**AdminServer**.
   c. Select the **Protocol** Tab and then select the **HTTP** tab.
   d. As **Frontend Host**, enter the front end LBR address that is used to access Enterprise management and the Remote Console (*admin.example.com* in the example used in this guide).
   e. Leave the **Frontend HTTP** port set to 0.
   f. Enter the LBR's admin listener port (445) as **Frontend HTTPS port**.
   g. Click **Save**.
   h. Click the cart icon at the top right to commit the changes.
2. Enable the proxy plug-in for the domain using the WebLogic Remote Console as follows:
   a. Click **Edit Tree**.
   b. Click **Environment**>**Domain**.
   c. Select **Web Application** tab.
   d. Click the **WebLogic Plugin Enable** button.
   e. Click **Save**.
   f. Click the cart icon at the top right to commit the changes.

Generate Required Certificates for OHS SSL Listeners

Since the OHS listeners use SSL it is necessary to create the appropriate certificates for them and add also the pertaining SANs for the server names they use. It is required to have

certificates for each WEBHOST address, adding as SAN the different ServerNames that are used in them.

This enterprise deployment uses `wccinternal.example.com`, `wcc.example.com`, and `admin.example.com` as front-end addresses. These addresses are used in the WLS domain configuration as front-end addresses for different clusters and servers.

Oracle recommends using the same Identity and Trust store files for all the CAs and certificates used in the app tier. The OHS nodes, do not use shared storage so the stores need to be copied to their private folders from the app tier. Certificates in a production system should come from formal Certificate Authorities.

In Oracle FMW 14.1.2.0, the Oracle WebLogic allows the usage of a per-domain Certificate Authority (CA). To update the Identity store and a Trust Store for the OHS SSL listeners in a WebLogic Server using a per-domain CA, you can perform the following steps. Run these steps in any of the WLS nodes (because the OHS ones do not install the CerGen and keytool utilities) and then transfer the stores to the OHS nodes:

1. Download the `generate_perdomainCACERTS-ohs.sh` script from the maa github repo *https://github.com/oracle-samples/maa/blob/main/1412EDG/generate_perdomainCACERTS-ohs.sh* to WCCHOST1.
2. Run the script with the following arguments:
   - WLS_DOMAIN_DIRECTORY: Directory hosting the WebLogic Domain that the Administration Server uses (ASERVER variable in this guide).
   - WL_HOME: The directory within the Oracle home where the Oracle WebLogic Server software binaries are stored. Typically `/u01/oracle/products/fmw/wlserver`.
   - KEYSTORE_HOME: Directory where the appIdentity and the appTrust stores are created.
   - KEYPASS: Password used for the WebLogic administration user (reused for certs and stores).
   - LIST_OF_OHS_SSL_VIRTUAL_HOSTS: A space separated list of OHS Virtual host addresses enclosed in single quotes.

   For example:

   ```
   ./generate_perdomainCACERTS-ohs.sh /u01/oracle/config/domains/wccedg_domain /u01/
   oracle/products/fmw/wlserver /u01/oracle/config/keystores "password"
   'ohshost1.example.com ohshost2.example.com'
   ```

   The script performs the following actions:

   a. It traverses the domain configuration and extracts the front-end addresses used by the domain.
   b. It uses the per domain CA to generate certificates for the OHS addresses that are provided as input to the script. The front-end addresses gathered from the domain configuration are added as SAN Subject Alternative Names (SAN) to these certificates.
   c. It connects to the front-end addresses detected in the domain configuration and downloads their public certificates. It adds these certificates to the WebLogic's trust keystore to allow the WebLogic servers establish SSL handshake with the different front-end addresses (used for callbacks, identity access and other redirections).

> **Note**
>
> The node where the `generate_perdomainCACERTS-ohs.sh` script is executed needs to have connectivity to the different front-end addresses included in the domain's `config.xml` to download their certificates.

d. It uses orapki to convert the identity and trust stores in the application tier into the required pkcs wallets used by the different OHS Virtual Hosts.

e. It creates a tar with the corresponding wallets (this needs to be transferred to the OHS nodes for completing the SSL configuration).

3. Transfer the tar generated by the script to the OHS nodes.

   a. Use scp or any sftp tool to copy tar file to the OHS nodes. For consistency with the app tier, place it under `$WEB_KEYSTORE_HOME`.

   b. Untar the contents of the file in that folder as follows:

      i. `cd $WEB_KEYSTORE_HOME`

      ii. `tar -xzvf orapki-ohs.tgz`

      This creates a wallet for WLS access and a directory wallet for each virtual host provided as parameter.

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

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

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.

About the WebLogicCluster Parameter of the <VirtualHost> Directive

A key parameter of the Oracle HTTP Server `<VirtualHost>` directive is the `WebLogicCluster` parameter, which is part of the WebLogic Proxy Plug-In for Oracle HTTP Server. When you configure Oracle HTTP Server for an enterprise deployment, consider the following information when you add this parameter to the Oracle HTTP Server configuration files.

The servers specified in the `WebLogicCluster` parameter are important only at startup time for the plug-in. The list needs to provide at least one running cluster member for the plug-in to discover other members of the cluster. When you start the Oracle HTTP server, the listed cluster member must be running. Oracle WebLogic Server and the plug-in work together to update the server list automatically with new, failed, and recovered cluster members.

Some example scenarios:

- Example 1: If you have a two-node cluster and then add a third member, you do not need to update the configuration to add the third member. The third member is discovered on the fly at runtime.
- Example 2: You have a three-node cluster but only two nodes are listed in the configuration. However, if both listed nodes are down when you start Oracle HTTP Server, then the plug-in would fail to route to the cluster. You must ensure that at least one of the listed nodes is running when you start Oracle HTTP Server.

  If you list all members of the cluster, then you guarantee you can route to the cluster, assuming at least one member is running when Oracle HTTP Server is started.

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. Ensure `ssl.conf` is included in the `httpd` configuration.

```
grep ssl.conf httpd.conf
include "ssl.conf"
```

5. Copy the `ssl.conf` file to a different file name.

> **Note**
>
> This is used as a template for other module conf files.

```
cp $WEB_DOMAIN_HOME/config/fmwconfig/components/OHS/ohs1/
ssl.conf $WEB_DOMAIN_HOME/config/fmwconfig/components/OHS/ohs1/moduleconf/
ssl.template
```

6. Edit the `ssl.conf` file to include only the following lines (remove other content from the file):

```
<IfModule ossl_module>
#
# Some MIME-types for downloading Certificates and CRLs
    AddType application/x-x509-ca-cert .crt
    AddType application/x-pkcs7-crl .crl

# Inter-Process Session Cache:
# Configure the SSL Session Cache: First the mechanism
# to use, second the expiring timeout (in seconds) and third
# the mutex to be used.
    SSLSessionCache "shmcb:${ORACLE_INSTANCE}/servers/${COMPONENT_NAME}/
```

```
logs/ssl_scache(512000)"
    SSLSessionCacheTimeout 300

</IfModule>
```

7. Modify the `$WEB_DOMAIN_HOME/config/fmwconfig/components/OHS/ohs1/mod_wl_ohs.conf` to include the appropriate WLSSWallet file (required to route on SSL to the WLS backends) as follows:

```
# NOTE : This is a template to configure mod_weblogic.

LoadModule weblogic_module   "${PRODUCT_HOME}/modules/mod_wl_ohs.so"

# This empty block is needed to save mod_wl related configuration from EM
to this file when changes are made at the Base Virtual Host Level
<IfModule weblogic_module>

    WLIOTimeoutSecs 900
    KeepAliveSecs 290
    FileCaching OFF
    WLSocketTimeoutSecs 15
    ErrorPage http://www.oracle.com/splash/cloud/index.html
    WLRetryOnTimeout NONE
    WLForwardUriUnparsed On
    SecureProxy On
    WLSSLWallet "/u02/oracle/config/keystores/orapki/"
</IfModule>
```

8. 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. Copy the `ssl.template` file to the `admin_vh.conf` file, this will transfer most of the required SSL configuration:

```
cp $WEB_DOMAIN_HOME/config/fmwconfig/components/OHS/ohs1/moduleconf/
ssl.template $WEB_DOMAIN_HOME/config/fmwconfig/components/OHS/ohs1/
moduleconf/admin_vh.conf
```

3. Edit the file to add the following Listen, VirtualHost, ServerName, AllowEncodedSlashes, and Location directives. Also, change the SSLWallet directory to point to the specific wallet for the virtual host. The `admin_vh.conf` file should resemble the following file.

```
######################################################################
# Oracle HTTP Server mod_ossl configuration file: ssl.conf       #
######################################################################

# The Listen directive below has a comment preceding it that is used
# by tooling which updates the configuration.  Do not delete the comment.
#[Listen] OHS_SSL_PORT
Listen WEBHOST1:4445
##
## SSL Virtual Host Context
##
#[VirtualHost] OHS_SSL_VH
<VirtualHost WEBHOST1:4445>
ServerName admin.example.com:445
AllowEncodedSlashes On
  <IfModule ossl_module>

   #  SSL Engine Switch:
   #  Enable/Disable SSL for this virtual host.
   SSLEngine on

   #  Client Authentication (Type):
   #  Client certificate verification type and depth.  Types are
   #  none, optional and require.
   SSLVerifyClient None

   #  SSL Protocol Support:
   #  Configure usable SSL/TLS protocol versions.
   SSLProtocol TLSv1.2 TLSv1.3

   # Option to prefer the server's cipher preference order
   SSLHonorCipherOrder on

   #  SSL Cipher Suite:
   #  List the ciphers that the client is permitted to negotiate.
   SSLCipherSuite
TLS_AES_128_GCM_SHA256,TLS_AES_256_GCM_SHA384,TLS_CHACHA20_POLY1305_SHA256,
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SH
A384,TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES_256_GCM_S
HA384

   #Path to the wallet
   #SSLWallet "${ORACLE_INSTANCE}/config/fmwconfig/components/$
{COMPONENT_TYPE}/instances/${COMPONENT_NAME}/keystores/default"
   SSLWallet "/u02/oracle/config/keystores/orapki/orapki-vh-WEBHOST1"

   <FilesMatch "\.(cgi|shtml|phtml|php)$">
      SSLOptions +StdEnvVars
   </FilesMatch>

   <Directory "${ORACLE_INSTANCE}/config/fmwconfig/components/$
```

```
{COMPONENT_TYPE}/instances/${COMPONENT_NAME}/cgi-bin">
      SSLOptions +StdEnvVars
   </Directory>

   BrowserMatch "MSIE [2-5]" \
         nokeepalive ssl-unclean-shutdown \
         downgrade-1.0 force-response-1.0

   # Add the following directive to add HSTS
   <IfModule mod_headers.c>
   Header always set Strict-Transport-Security "max-age=63072000; preload;
includeSubDomains"
   </IfModule>


   <Location /em>
   WLSRequest ON
   WebLogicHost ADMINVHN
   WebLogicPort 9002
   </Location>
   <Location /management>
   WLSRequest ON
   WebLogicHost ADMINVHN
   WebLogicPort 9002
   </Location>

</IfModule>
</VirtualHost>
```

4. Repeat similar steps to create a `wccinternal_vh.conf` file with this content (notice the different listen port, virtual host, and WLS settings):

```
########################################################################
# Oracle HTTP Server mod_ossl configuration file: ssl.conf       #
########################################################################

# The Listen directive below has a comment preceding it that is used
# by tooling which updates the configuration.  Do not delete the comment.
#[Listen] OHS_SSL_PORT
Listen WEBHOST1:4444


##
## SSL Virtual Host Context
##
#[VirtualHost] OHS_SSL_VH
<VirtualHost WEBHOST1:4444>
ServerName wccinternal.example.com:444
  <IfModule ossl_module>
   #  SSL Engine Switch:
   #  Enable/Disable SSL for this virtual host.
   SSLEngine on

   #  Client Authentication (Type):
   #  Client certificate verification type and depth.  Types are
```

```
   #  none, optional and require.
   SSLVerifyClient None

   #  SSL Protocol Support:
   #  Configure usable SSL/TLS protocol versions.
   SSLProtocol TLSv1.2 TLSv1.3

   # Option to prefer the server's cipher preference order
   SSLHonorCipherOrder on

   #  SSL Cipher Suite:
   #  List the ciphers that the client is permitted to negotiate.
   SSLCipherSuite
TLS_AES_128_GCM_SHA256,TLS_AES_256_GCM_SHA384,TLS_CHACHA20_POLY1305_SHA256,
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SH
A384,TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES_256_GCM_S
HA384

   #Path to the wallet
   #SSLWallet "${ORACLE_INSTANCE}/config/fmwconfig/components/$
{COMPONENT_TYPE}/instances/${COMPONENT_NAME}/keystores/default"
   SSLWallet "/u02/oracle/config/keystores/orapki/orapki-vh-WEBHOST1"

   <FilesMatch "\.(cgi|shtml|phtml|php)$">
      SSLOptions +StdEnvVars
   </FilesMatch>

   <Directory "${ORACLE_INSTANCE}/config/fmwconfig/components/$
{COMPONENT_TYPE}/instances/${COMPONENT_NAME}/cgi-bin">
      SSLOptions +StdEnvVars
   </Directory>

   BrowserMatch "MSIE [2-5]" \
         nokeepalive ssl-unclean-shutdown \
         downgrade-1.0 force-response-1.0

   # Add the following directive to add HSTS
   <IfModule mod_headers.c>
   Header always set Strict-Transport-Security "max-age=63072000; preload;
includeSubDomains"
   </IfModule>


  </IfModule>
</VirtualHost>
```

5. Restart the ohs1 instance by entering the following commands:

```
$WEB_ORACLE_HOME/oracle_common/common/bin/wlst.sh
```

```
wls:/offline>
nmConnect('ohsdomainadmin','ohsdomainadmin_password','localhost','5556','oh
sdomainname', 'WEB_DOMAIN_HOME','PLAIN')
```

```
wls:/nm/wccedgohs> nmKill(serverName='ohs1',serverType='OHS')
```

```
wls:/nm/wccedgohs> nmStart(serverName='ohs1',serverType='OHS')
```

Watch the $WEB_DOMAIN_HOME/servers/ohs1/logs/ohs1.log file for errors.

6. 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

7. Edit the admin_vh.conf and wccinternal_vh.conf files and change any references from WEBHOST1 to WEBHOST2 in the <VirtualHost> directives.

8. Restart the ohs2 instance by entering the following commands:

```
$WEB_ORACLE_HOME/oracle_common/common/bin/wlst.sh
```

```
wls:/offline>
nmConnect('ohsdomainadmin','ohsdomainadmin_password','localhost','5556','oh
sdomainname','WEB_DOMAIN_HOME','PLAIN')
```

```
wls:/nm/wccedgohs> nmKill(serverName='ohs2',serverType='OHS')
```

```
wls:/nm/wccedgohs> nmStart(serverName='ohs2',serverType='OHS')
```

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.

- https://admin.example.com:445/index.html
- http://wccinternal.example.com:444/index.html

Validating Access to the Management Consoles and Administration Server

To verify the changes that you have made in this chapter:

1. Access the Fusion Middleware Control by using the following URL:

```
https://admin.example.com:445/em
```

Configure a New Provider in the WebLocic Remote Console to Access the Domain Configuration Through the Front-end LBR

Create a new Admin Server Connection Provider that connects through the front-end load balancer and OHS to the domain's Administration Server. To establish this connection, the WebLogic Remote Console must trust the certificate used by the load balancer for the administration front-end address.

1. Ensure that the Trust Store used by the WebLogic Remote Console includes the certificate or the CA certificate used by the front-end load balancer in the admin virtual server.

   > **Tip**
   >
   > If you used the script `generate_perdomainCACERTS-ohs.sh`, you can download the `appTrustKeyStore.pkcs12` file from the domain and use it as the WebLogic Remote Console trust store. It includes the front-end load balancer certificates as a trusted entity.

2. Open the WebLogic Remote Console and click **Add Admin Server Connection Provider**.
3. Use the following values for the new provider:
   - **Connection Provider Name**:

     Use a name identifying the connection. For example, `wccedg_domain_lbrprovider`.
   - **Username** and **Password**:

     Enter the WebLogic Domain Administration user and password.
   - **URL**: Use the front-end address and the port. For example, `https://admin.example.com:445`
   - **Make Insecure Connect**: If the appropriate trust store settings are completed, you do not need to check this field.

     > **Note**
     >
     > If you are using demo certs in the load balancer, you might need to check the **Disable host name verification** field in the WebLogic Remote Console settings.

4. Click **OK** to add the provider.
5. Click the new provider.

   You must able to manage the domain remotely through the front end LBR with these settings.

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.

This section contains the following procedures.

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_14.1.2.0.0_wccontent_generic.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.

Screen	Description
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*.
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

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 *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 *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)

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 *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 **Database Type**, select **Oracle Database enabled for edition-based redefinition**.

> **Note**
>
> Oracle Database enabled for edition-based redefinition (EBR) is recommended to support Zero Down Time upgrades. For more information, see https://www.oracle.com/database/technologies/high-availability/ebr.html.

2. In the **Host Name** field, enter the SCAN address of the Oracle RAC Database.
3. Enter the Port number of the RAC database scan listener, for example 1521.
4. Enter the RAC Service Name of the database.
5. Enter the User Name of a user that has permissions to create schemas and schema objects, for example SYS.
6. Enter the Password of the user name that you provided in *Step 4*.
7. If you have selected the SYS user, ensure that you set the role to **SYSDBA**.
8. 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 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
Click **Next**.
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, therefore, perform this check outside RCU itself.

> **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.

> **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 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 FMW1412_WCCINFRA/<wccinfra_password>

SQL*Plus: Release 23.0.0.0.0 - for Oracle Cloud and Engineered Systems on Wed Sep 11
14:20:00 2024 Version 23.5.0.24.07
Copyright (c) 1982, 2024, Oracle. All rights reserved.


Connected to:
Oracle Database 23ai EE Extreme Perf Release 23.0.0.0.0 - for Oracle Cloud and
Engineered Systems
Version 23.5.0.24.07

SQL>
```

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.

> **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. 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 following sections to extend the domain for Oracle WebCenter Content.

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
- 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, Reviewing Your Configuration Specifications and Configuring the Domain
- Task 18, Writing Down Your Domain Home and Administration Server URL
- Task 19, 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 - 14.1.2.0 [wccontent]**

  In addition, the following additional templates should already be selected, because they were used to create the initial Infrastructure domain:

  - **Oracle Enterprise Manager - 14.1.2.0 [em]**
  - **Oracle JRF - 14.1.2.0 [oracle_common]**
  - **WebLogic Coherence Cluster Extension - 14.1.2.0 [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 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 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**.
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.

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 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 *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)
Service Name	Verify that the service name for the Oracle RAC database is appropriate. For example, `wccedg.example.com`.
ONS Host and Port	These values are not required when you are using an Oracle 12c database or later versions because the ONS list is automatically provided from the database to the driver. .
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**

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	Administration Port	Server Groups
WLS_WCC1	WCCHOST1	Disabled	Checked	16201	9005	UCM-MGD-SVR
WLS_WCC2	WCCHOST2	Disabled	Checked	16201	9005	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. Specify `wcc.example.com` in the Frontend Host field.
4. Leave the Frontend HTTP Port blank and use `443` (or your precise LBS listeners port for application requests) as the Frontend HTTPS port.

> **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. 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*.

Task 15 Creating Machines for WebCenter Content Servers
If required, use the Machines screen to add two new Unix Machines. Use the Machines screen to create new machines in the domain. A machine is required in order for the Node Manager to be able to start and stop the servers.

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.

NNode Manager Type a m e	Node Manager Listen Address	Node Manager Listen Port
WSSL C C H O S T 1	The value of the WCCHOST1 host name variable. For example, `WCCHOST1.example.com`.	5556
WSSL C C H O S T 2	The value of the WCCHOST2 host name variable. For example, `WCCHOST2.example.com`.	5556
ASSL D M I N H O S T	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 Assigning Managed Servers to the Cluster.
Assign AdminServer to ADMINHOST.
Assign WLS_WCC1 Managed Server to WCCHOST1 machine, and assign WLS_WCC2 Managed Server to WCCHOST2 machine.

> **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 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 18 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 Administration Server URL to access the WebLogic Remote Console and Oracle Enterprise Manager Fusion Middleware Control.
Click **Finish** to dismiss the configuration wizard.

Task 19 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.

Re-Configuring SSL Certificates and Updating Servers

Repeat the steps in Configuring SSL Certificates for the Domain to generate new certificates because we have added new host names. In addition, the **Security Settings** must be updated for the newly added servers.

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

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 \
           -log=/tmp/pack_ExtWCC.log \
           -log_priority=debug
   ```

   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 \
       -log_priority=DEBUG \
         -log=/tmp/unpack.log \
       -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.

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 WLS_WCC1 Managed Server

To start the WLS_WCC1 Managed Server:

1. Enter the following URL into a browser to display the Fusion Middleware Control login screen: `https://admin.example.com:445/em`
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_WCC1 Managed Server and click **Start Up** on the Oracle WebLogic Server toolbar.
5. When the startup operation is complete, navigate to the **Domain** home page and verify that the WLS_WCC1 Managed Server is up and running.

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*/wccedg_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/wccedg_domain/WCC_Cluster/cs/
   - **Native File Repository Location**: Set this to *ORACLE_RUNTIME*/*domain_name*/WCC_Cluster/cs/vault/
     For example:/u01/oracle/runtime/wccedg_domain/WCC_Cluster/cs/vault/
   - **WebLayout Folder**: Set this to *ORACLE_RUNTIME*/*domain_name*/WCC_Cluster/cs/weblayout/
     For example:/u01/oracle/runtime/wccedg_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/wccedg_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|WEBHOST1-IP|WEBHOST2-IP|wcc.example.com-IP|wccinternal.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 Fusion Middleware Control.

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 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

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

To configure the Oracle HTTP Server instances in the web tier so that they route requests correctly to the Oracle SOA Suite 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.

This procedure assumes that you performed the Oracle HTTP Server configuration tasks described in Configuring Oracle HTTP Server to Route Requests to the Application Tier.

Create a `wcc_vh.conf` file by copying the existing `admin_vh.conf` file. This will transfer most of the required SSL configuration. Then update it with the entries required by SOA:

1. Log into 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. Copy the existing `admin_vh.conf` file to the `wcc_vh.conf` file. This will transfer most of the required SSL configuration:

   ```
   cp admin_vh.conf wcc_vh.conf
   ```

3. Edit the file and customize with the required values for **Listen**, **ServerName**, **VirtualHost**, **SSLWallet** and **Location** directives (AllowEncodedSlashes not needed here).

   > **Note**
   >
   > For the Listen address you need to specify a different port from the ones used in previous virtual hosts (`admin_vh.conf` and `soainternal_vh.conf`). Otherwise the listeners will conflict.

   ```
   #[Listen] OHS_SSL_PORT

   Listen WEBHOST1:4443
   ##
   ## SSL Virtual Host Context
   ```

```
##
#[VirtualHost] OHS_SSL_VH
<VirtualHost WEBHOST1:4443>
ServerName wcc.example.com:443
  <IfModule ossl_module>

   # SSL Engine Switch:
   # Enable/Disable SSL for this virtual host.
   SSLEngine on

   # Client Authentication (Type):
   # Client certificate verification type and depth. Types are
   # none, optional and require.
   SSLVerifyClient None

   # SSL Protocol Support:
   # Configure usable SSL/TLS protocol versions.
   SSLProtocol TLSv1.2 TLSv1.3

   # Option to prefer the server's cipher preference order
   SSLHonorCipherOrder on

   # SSL Cipher Suite:
   # List the ciphers that the client is permitted to negotiate.
   SSLCipherSuite
TLS_AES_128_GCM_SHA256,TLS_AES_256_GCM_SHA384,TLS_CHACHA20_POLY1305_SHA256,TLS_ECDHE_
ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,TLS_ECDHE_RSA_W
ITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384

   #Path to the wallet
   #SSLWallet "${ORACLE_INSTANCE}/config/fmwconfig/components/${COMPONENT_TYPE}/
instances/${COMPONENT_NAME}/keystores/default"
   SSLWallet "/u02/oracle/config/keystores/orapki/orapki-vh-WEBHOST1"

   <FilesMatch "\.(cgi|shtml|phtml|php)$">
      SSLOptions +StdEnvVars
   </FilesMatch>

   <Directory "${ORACLE_INSTANCE}/config/fmwconfig/components/${COMPONENT_TYPE}/
instances/${COMPONENT_NAME}/cgi-bin">
      SSLOptions +StdEnvVars
   </Directory>
         BrowserMatch "MSIE [2-5]" \
         nokeepalive ssl-unclean-shutdown \
         downgrade-1.0 force-response-1.0

   # Add the following directive to add HSTS
   <IfModule mod_headers.c>
   Header always set Strict-Transport-Security "max-age=63072000; preload;
includeSubDomains"
   </IfModule>

#UCM
<Location /cs>
 WebLogicCluster WCCHOST1:16201,WCCHOST2:16201
 WLSRequest ON
 WLCookieName JSESSIONID
 WLProxySSL ON
 WLProxySSLPassThrough ON
</Location>

<Location /adfAuthentication>
```

```
 WebLogicCluster WCCHOST1:16201,WCCHOST2:16201
 WLSRequest ON
 WLCookieName JSESSIONID
 WLProxySSL ON
 WLProxySSLPassThrough ON
</Location>

<Location /_ocsh>
 WebLogicCluster WCCHOST1:16201,WCCHOST2:16201
 WLSRequest ON
 WLCookieName JSESSIONID
 WLProxySSL ON
 WLProxySSLPassThrough ON
</Location>

  </IfModule>
</VirtualHost>
```

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/OHS/ohs2/moduleconf/`

5. Edit the `wcc_vh.conf` file and change any references to WEBHOST1 to WEBHOST2 in the `<VirtualHost>` directives.

6. Restart the Oracle HTTP servers on WEBHOST1 and WEBHOST2.

> **Note**
>
> If internal invocations are going to be used in the system, add the appropriate locations to the soainternal virtual host.

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

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.

If you have made the following selections in the High Availability Options screen, as recommended in this guide both 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**.

In case you did not select JDBC for JMS and TLOGS persistence in the High Availability Options screen, you can still configure JDBC stores manually in a post step. For more information about how to configure them manually, see Using JDBC Persistent Stores for TLOGs and JMS in an Enterprise Deployment.

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.

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

Synchronizing the System Clocks

Verify that the system clocks on each host computer are synchronized.

To verify the time synchronization, query the NTP service by running the `chronyc -n tracking` command on each host.

Sample output:

```
$chronyc -n tracking
Reference ID : A9FEA9FE (169.254.169.254)
Stratum : 3
```

```
Ref time (UTC) : Tue Jan 14 15:28:01 2025
System time : 0.000043127 seconds fast of NTP time
Last offset : +0.000034640 seconds
...
```

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 -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**
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 *Installing and Configuring the Oracle Fusion Middleware Infrastructure*.

Screen	Description
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

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 *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)

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 *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 *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 **Database Type**, select **Oracle Database enabled for edition-based redefinition**.

> **Note**
>
> Oracle Database enabled for edition-based redefinition (EBR) is recommended to support Zero Down Time upgrades. For more information, see https://www.oracle.com/database/technologies/high-availability/ebr.html.

2. In the **Host Name** field, enter the SCAN address of the Oracle RAC Database.
3. Enter the **Port** number of the RAC database scan listener, for example 1521.
4. Enter the RAC **Service Name** of the database.
5. Enter the **User Name** of a user that has permissions to create schemas and schema objects, for example SYS.
6. Enter the **Password** of the user name that you provided in step 4.
7. If you have selected the SYS user, ensure that you set the role to SYSDBA.
8. 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 FMW1412_WCCINFRA/<wccinfra_password>

SQL*Plus: Release 23.0.0.0.0 - for Oracle Cloud and Engineered Systems on Wed Sep 11
14:20:00 2024 Version 23.5.0.24.07
Copyright (c) 1982, 2024, Oracle. All rights reserved.


Connected to:
Oracle Database 23ai EE Extreme Perf Release 23.0.0.0.0 - for Oracle Cloud and
Engineered Systems
Version 23.5.0.24.07

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.

> **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. 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.

Domain creation and configuration includes the following tasks:

Extending the Domain

Follow the instructions in this section to extend the domain for Oracle SOA Suite.

> **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 *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 Reference Configuration [soa]**

  The following additional templates should already be selected, because they were used to create the initial domain:

  - Oracle Enterprise Manager - 14.1.2.0.0[em]
  - Oracle WSM Policy Manager - 14.1.2.0.0[oracle_common]
  - Oracle JRF - 14.1.2.0.0[oracle_common]
  - WebLogic Coherence Cluster Extension - 14.1.2.0.0[wlserver]

  And the following template should also be selected, because you already configured WebCenter Content:

  - Oracle Universal Content Management - Content Server - 14.1.2.0.0[wccontent]

For more information about the options on this screen, see 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 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 *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 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
Service Name	Verify that the service name for the Oracle RAC database is appropriate. For example, *soaedg.example.com*.
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	These values are not required when you are using an Oracle 12c database or higher versions because 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 *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 *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.

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 *Creating WebLogic Domains Using the Configuration Wizard*.

Server Name	Listen Address	Enable Listen Port	Listen Port	Enable SSL Port	SSL Listen Port	Administration Port	Server Groups
WLS_SOA1	WCCHOST1	Unchecked	Disabled	Checked	8001	9007	SOA-MGD-SVRS-ONLY
WLS_SOA2	WCCHOST2	Unchecked	Disabled	Checked	8001	9007	SOA-MGD-SVRS-ONLY

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.
You also set the **Frontend Host** property for the cluster, which ensures that, when necessary, WebLogic Server redirects 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.

1. Click the **Add** button.
2. Specify `SOA_Cluster` in the **Cluster Name** field.
3. Specify `wcc.example.com` in the **Frontend Host** field.
4. Leave the **Frontend HTTP Port** blank and use `443` (or your precise LBS listeners port for app requests) as the **Frontend HTTPS** port.

> **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 *Administering Clusters for Oracle WebLogic Server*.

For more information about the options on this screen, see Clusters in *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 *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 *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 *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.

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 **Edit Tree** in the Admin 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.

Update Certificates for New Frontend Addresses

This section contains information about certificates for new frontend addresses.

> **Note**
>
> With regard to Certificates for the domain extension, because the SOA servers use the same listen addresses (different port), there is no need to create new certificates and update stores. However, the SOA cluster uses a different front end address that will be added as a trusted endpoint in the Configuring SSL Certificates for the Domain.

Updating the WebLogic Servers Security Settings

Perform the following steps to update the WebLogic Servers Security Settings and Administration Port:

1. Access the Domain provider in the Remote Console and update the Administration Server and WebLogic Servers Security Settings:
   a. Click **Edit Tree**.
   b. Click **Environment** > **Servers** > **AdminServer**.
   c. Click **Security** tab.
   d. Change the keystores dropdown to **Custom Identity and Custom Trust**.

e. In **Custom Identity Keystore**, enter the fully qualified path to the identity keystore as follows:

   `KEYSTORE_HOME/appIdentityKeyStore.pkcs12`
   Replace `KEYSTORE_HOME` with the value of the folder you use for storing keystore, as described in the Table 7-2.

f. Set the **Custom Identity Keystore Type** to JKS.

> **Note**
>
> Specifying JKS or PKCS12 is valid both for pkcs12 and jks stores. Both formats can be read and managed if the Customer Key Store Type is set to "JKS".

g. In **Custom Identity Keystore Passphrase**, enter the password `Keystore_Password` you provided in the certificate generation steps.

h. In **Custom Trust Keystore**, enter the fully qualified path to the trust keystore.

   `KEYSTORE_HOME/appTrustKeyStore.pkcs12`
   Replace `KEYSTORE_HOME` with the value of the folder you use for storing keystore, as described in the Table 7-2.

i. Set the **Custom Trust Keystore Type** to JKS.

> **Note**
>
> Specifying JKS or PKCS12 is valid both for pkcs12 and jks stores. Both formats can be read and managed if the Customer Key Store Type is set to "JKS".

j. In **Custom Trust Keystore Passphrase**, enter the password you provided as the <**keypass**> in the certificate generation steps.

k. Click **Save**.

l. Under **Security** settings, navigate to **SSL** tab.

m. In the **Server Private Key Alias** filed enter the alias provided in the certificate generation steps. If you used the certificate generation script this will be the same as the listen address used for the WLS server.

n. In the **Server Private Key Pass Phrase** field, enter the password provided in the certificate generation steps. If you used the certificate generation script this will be the same as the keystore passphrase.

o. Click **Save**.

   The cart on the top right part of the screen will show **full** with a yellow bag inside.

p. Click the Cart icon on the top right and select **Commit Changes**.

Repeat the above steps for each managed server in the domain changing the alias to match the alias used for the certificates.

2. Return to the terminal window where you started the Administration Server with the start script.

3. Press **Ctrl+C** to stop the Administration Server process.

Wait for the Administration Server process to end and for the terminal command prompt to appear.

4. Start the Administration Server again by using the following script:
   a. Change directory to the following directory:

      ```
      cd $ASERVER_HOME/bin
      ```

   b. Run the start script:

      ```
      ./startWebLogic.sh
      ```

   c. Monitor the output in the terminal till the following output is displayed.

      ```
      <Server state changed to RUNNING>
      ```

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 13-2 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 13-2 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

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 *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 *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 *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.

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

To start the WLS_SOA1 Managed Server:

1. Enter the following URL into a browser to display the Fusion Middleware Control login screen:

   ```
   https://admin.example.com:445/em
   ```

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:

   ```
   https://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/`

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:
   - `https://wcc.example.com:443/soa-infra`
   - `https://wcc.example.com:443/integration/worklistapp`
   - `https://wcc.example.com:443/sdpmessaging/userprefs-ui`
   - `https://wcc.example.com:443/soa/composer`
   - `https://wcc.example.com:443/wsm-pm`

Modifying the Upload and Stage Directories to an Absolute Path

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. See Modifying the Upload and Stage Directories to an Absolute Path in an Enterprise Deployment.

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.

Generate the Required Certificates for OHS SSL Listeners

To add the new fronted address to the certificate stores and update the SAN for the OHS listeners certs, follow the steps described in the Generate Required Certificates for OHS SSL Listeners section in Starting the Oracle HTTP Server Instances.

When asked to replace the existing OHS Virtual Host certificates, answer yes so that they are updated with the new frontend address for the SOA cluster as SAN.

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 cluster.
> - 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.

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

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

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

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 *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 Remote 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`.

To configure adapters, perform the following steps in the WebLogic Remote Console:

1. Create a deployment plan directory on shared storage (if it does not exist) as follows:

   ```
   mkdir -p $DEPLOY_PLAN_HOME/soaedg_domain
   ```

2. Create a fileadapter control directory in the shared runtime folder as follows:

   ```
   mkdir -p /u01/oracle/runtime/soaedg_domain/SOA_Cluster/fadapter
   ```

3. In the **Monitoring Tree**, navigate to **Deployments** > **Application Management** > **File Adapter**.
4. Click **Create Plan** (if it does not already have a plan) and use the `DEPLOY_PLAN_HOME/domain_name/` as its directory.
5. After the new plan is displayed under the **File Adapter**, in the **Monitoring Tree** navigate to **Deployments** > **Application Management** > **File Adapter**.
6. Select **Configuration** > **Outbound Connection Pool Groups**.
7. Navigate to **javax.resource.cci.ConnectionFactory** > **Outbound Connection Pool Instances**.
8. Navigate to **eis/HAFileAdapter** > **Properties**.
9. Modify the values of the properties described in the following table:

Table 13-3 The following table describes modified parameters

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.
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.

10. Redeploy the Adapter using the console.
    a. In the **Monitoring Tree**, navigate to **Deployments** > **Application Management**.
    b. Select the **FileAdapter deployment** check box.

c. Click **Update/Redeploy** > **Redeploy** - **Deployment Source and Plan on Server** (it is not possible to use **Update - Deployment Plan on Server** because these are non-dynamic changes).

Ensure that the deployment plan is correct in the Plan Path filed.

11. Click **Done**.

    Wait for the operation to complete.

12. After the operation is complete, check the values entered in the **Monitoring** > **Deployments** > **Application Management** > **FileAdapter** > **Deployment plan**.

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 13-1.

> **Note**
>
> The location attribute is set to `eis/HAFileAdapter` for the connection factory.

Example 13-1 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 #unique_297/unique_297_Connect_42_BABIFCEI.
- 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 is established to only one random server. If that particular server goes down, no further messages are processed.

To avoid this issue, you can use the “cluster name” syntax in the `FactoryProperties` of the adapter instead of using the static list of members. The cluster name syntax is as follows:

```
cluster:t3s://cluster_name
```

When you use `cluster:t3s://cluster_name`, the invocation fetches the complete list of members in the cluster at any given time, thus avoiding any dependencies on the initial servers and accounting for every member that is alive in the cluster at that point of time. Note that you can use this cluster syntax only when the cluster is in the same domain.

1. Create a deployment plan directory on shared storage (if it does not exist) as follows:

   ```
   mkdir -p $DEPLOY_PLAN_HOME/soaedg_domain
   ```
2. In the **Monitoring Tree** , navigate to **Deployments** > **Application Management** > **JMS Adapter**.
3. Create Plan (if it does not already have a plan) and use the `DEPLOY_PLAN_HOME/domain_name/` as its directory.
4. After the new plan is displayed under the **JMS Adapter**, in the **Monitoring Tree** navigate to **Deployments** > **Application Management** > **JMS Adapter**.
5. Navigate to **Configuration** > **Outbound Connection Pool Groups**.
6. Navigate to **oracle.tip.adapter.jms.IJmsConnectionFactory**> **Outbound Connection Pool Instances**.
7. Click **eis/wls/Queue** > **Properties**.
8. Click the **FactoryProperties** field (click the corresponding cell under Property value), enter the following, all in one line, separated by semicolons. Adjust the values to match your cluster name, username and password:

   ```
   java.naming.factory.initial=weblogic.jndi.WLInitialContextFactory;
   java.naming.provider.url=cluster:t3s://SOA_Cluster;
   java.naming.security.principal=soaedgadmin;
   java.naming.security.credentials=<password>
   ```
9. Click **Save** after you update the properties.

10. Redeploy the Adapter using the console.
    a. Navigate to **Monitoring** > **Deployments** > **Application Management**.
    b. Select the **JMSAdapter deployment** check box.
    c. Click **Update/Redeploy** > **Redeploy** - **Deployment Source and Plan on Server** (not possible to use **Update - Deployment Plan on Server** because these are non dynamic changes)

    Ensure that the deployment plan is correct in the Plan Path filed.
11. Click **Done**.
    Wait for the operation to complete.
12. After the operation is complete, check the values entered in the **Monitoring** > **Deployments** > **Application Management** > **JMSAdapter** > **Deployment plan**.

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.

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 #unique_304.
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 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 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.

Backing Up the Configuration

It is an Oracle best practices recommendation to create a backup after you successfully extended a domain or at another logical point. Create a backup after you verify that the installation so far is successful. This is a quick backup for the express purpose of immediate restoration in case of problems in later steps.

The backup destination is the local disk. You can discard this backup when the enterprise deployment setup is complete. After the enterprise deployment setup is complete, you can initiate the regular deployment-specific Backup and Recovery process.

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.

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.

> **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. 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 - [wccontent]**

  The following additional templates should already be selected, because they were used to create the initial domain:

  - **Oracle Enterprise Manager - [em]**
  - **Oracle SOA Suite - [soa]**
  - **Oracle WebCenter Content - [wcc]**
  - **Oracle WSM Policy Manager - [oracle_common]**

- **Oracle JRF - [oracle_common]**
- **WebLogic Coherence Cluster Extension - [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	SSL Listen Port	AServer Group dmin istration Port
WLS_IBR1	WCCHOST1	16250	16251	9IBR-MGD-SVR 0 0 6
WLS_IBR2	WCCHOST2	16250	16251	9IBR-MGD-SVR 0 0 6

> **Note**
>
> The IBR servers will have a non-SSL port enabled in order to communicate locally with the Content servers. The IBR servers are not accessible externally.

> **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

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 Inbound Refinery Managed Servers

To start the `WLS_IBR1` Managed Server on WCCHOST1:

1. Enter the following URL into a browser to display the Fusion Middleware Control login screen:

```
https://admin.example.com:445/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_IBR1` Managed Server and click **Start Up** on the Oracle WebLogic Server toolbar.
5. When the startup operation is complete, navigate to the Domain home page and verify that the `WLS_IBR1` Managed Server is up and running.
6. 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.

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.

To create an outgoing provider for each Inbound Refinery instance:

1. Log in to Content Server at the following URL:

   `https://WCCHOST1:16201/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 Fusion Middleware Control.

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.

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.

> **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. 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 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, Reviewing Your Configuration Specifications and Configuring the Domain

- Task 17, Writing Down Your Domain Home and Administration Server URL
- Task 18, 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 - [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 - [wccontent]**
  - **Oracle SOA Suite - [soa]**
  - **Oracle Universal Content Management - Inbound Refinery - [wccontent]**
  - **Oracle Enterprise Manager - [em]**
  - **Oracle WSM Policy Manager - [oracle_common]**
  - **Oracle JRF - [oracle_common]**
  - **WebLogic Coherence Cluster Extension - [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** and **ONS Port** fields, these values are not required when you are using an Oracle 12c database or later versions because the ONS list is automatically provided from the database to the driver.
5. In the **ONS Port** field, enter the ONS Remote port (for example, 6200).
6. In the **Enable Fan** field, verify that the **Enable Fan** checkbox is selected, so that the database can receive and process FAN events

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	Administration Port	Server Groups
WLS_CPT1	WCCHOST1	Disabled	No	16401	9008	JRF-MAN-SVR CAPTURE-MGD-SVR
WLS_CPT2	WCCHOST2	Disabled	No	16401	9008	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 `9991`, 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 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 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.

Update the WebLogic Servers Security Settings

This section contains information about WebLogic Servers security settings.

Follow the steps described in Updating the WebLogic Servers Security Settings and update SSL settings for the WLS_WCCUI1 and WLS_WCCUI2 servers.

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 WebLogic Remote Console, and then starting it with the `nmStart` command. Before the restart, stop all Managed Servers in the domain through the WebLogic Remote Console, and then start them after the restart. Log in to the WebLogic Remote 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 WCCHOST1:16401,WCCHOST2:16401
   WLSRequest ON
   WLCookieName JSESSIONID
   WLProxySSL ON
   WLProxySSLPassThrough ON
</Location>

#DC-Client
<Location /dc-client>
   WebLogicCluster WCCHOST1:16401,WCCHOST2:16401
   WLSRequest ON
   WLCookieName JSESSIONID
   WLProxySSL ON
   WLProxySSLPassThrough ON
</Location>
```

2. Restart Oracle HTTP Server on both WEBHOST1 and WEBHOST2:

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`
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 Remote 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 Remote Console.
4. Stop WLS_CPT2 from the WebLogic Remote Console.
5. Access `https://wcc.example.com/dc-console` to verify that it is functioning properly.
6. Start WLS_CPT2 from the WebLogic Remote 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.

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 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
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 - [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 - [wccontent]**
- **Oracle SOA Suite - [soa]**
- **Oracle Universal Content Management - Inbound Refinery - [wccontent]**
- **Oracle Enterprise Manager - [em]**
- **Oracle WSM Policy Manager - [oracle_common]**
- **Oracle JRF - [oracle_common]**
- **WebLogic Coherence Cluster Extension - [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	Administration Port	Server Groups
WLS_IPM1	WCCHOST1	Disabled	No	16001	9009	IPM-MGD-SVR
WLS_IPM2	WCCHOST2	Disabled	No	16001	9009	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 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 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 to ensure the changes you have made to the domain have been applied.

Update the WebLogic Servers Security Settings

This section contains information about WebLogic Servers security settings.

Follow the steps described in Updating the WebLogic Servers Security Settings and update SSL settings for the WLS_WCCUI1 and WLS_WCCUI2 servers.

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

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 WCCHOST1:16001,WCCHOST2:16001
   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 WCCHOST1:16001,WCCHOST2:16001
   WLSRequest ON
   WLCookieName JSESSIONID
   WLProxySSL ON
```

```
   WLProxySSLPassThrough ON
</Location>
```

2. Restart Oracle HTTP Server on both WEBHOST1 and WEBHOST2:

   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`
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.

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.

> **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. 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 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 5, Providing the GridLink Oracle RAC Data Sources Details
- Task 6, Testing the JDBC Data Sources
- Task 7, Providing the GridLink Oracle RAC Component Schema Details
- Task 8, Testing the JDBC Component Schema
- Task 9, Specifying Credentials for wccadfConnectUser
- Task 10, Selecting Advanced Configuration
- Task 11, Configuring Managed Servers
- Task 12, Configuring a Cluster
- Task 13, Assigning Server Templates
- Task 14, Configuring Dynamic Servers
- Task 15, Assigning Managed Servers to the Cluster
- Task 16, Configuring Coherence Clusters
- Task 17, Verifying the Existing Machines
- Task 18, Assigning Servers to Machines
- Task 19, Configuring Virtual Targets
- Task 20, Configuring Partitions
- Task 21, Reviewing Configuration Summary
- Task 22, Reviewing Configuration Progress
- Task 23, Reviewing Your Configuration Specifications and Configuring the Domain
- Task 24, Writing Down Your Domain Home and Administration Server URL

- Task 25, 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 - [wccontent]**

  In addition, the following additional templates should already be selected, because they were used to create the initial domain:

  - **Oracle Universal Content Management - Content Server - [wccontent]**
  - **Oracle SOA Suite - [soa]**
  - **Oracle Universal Content Management - Inbound Refinery - [wccontent]**
  - **Oracle WebCenter Enterprise Capture - [wccapture]**
  - **Oracle Enterprise Manager - [em]**
  - **Oracle WSM Policy Manager - [oracle_common]**
  - **Oracle JRF - [oracle_common]**
  - **WebLogic Coherence Cluster Extension - [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 Specifying the Frontend Host
Specify `wcc.example.com` in the Frontend Host field.
Specify `0` as the Frontend HTTP Port and `443` as the Frontend HTTPS port.

> **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 *Administering Clusters for Oracle WebLogic Server*.

Task 5 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: These values are not required when you are using an Oracle 12c database or later versions because the ONS list is automatically provided from the database to the driver.
Click **Next**.

Task 6 Testing the JDBC Data Sources
Click **Next** to continue.

Task 7 Providing the GridLink Oracle RAC Component Schema Details
Click **Next** to continue.

Task 8 Testing the JDBC Component Schema
Click **Next** to continue.

Task 9 Specifying Credentials for wccadfConnectUser
On the Credentials screen, enter the WebLogic user name(not `weblogic_wcc`) and password.
Click **Next** to continue.

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

Task 11 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	Administration Port	Server Groups
WLS_WCCUI1	WCCHOST1	Disabled	No	16226	9004	UCM-ADF-MGD-SVR
WLS_WCCUI2	WCCHOST2	Disabled	No	16226	9004	UCM-ADF-MGD-SVR

Task 12 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 13 Assigning Server Templates
Click **Next** to proceed to the next screen.

Task 14 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 15 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 16 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*.

Task 17 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 18 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 19 Configuring Virtual Targets
Click **Next** to proceed to the next screen.

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

Task 21 Reviewing Configuration Summary
Click **Update** to proceed to the next screen.

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

Task 23 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 24 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 you need the Administration Server URL to access the WebLogic Remote Console and Oracle Enterprise Manager Fusion Middleware Control.
Click **Finish** to dismiss the configuration wizard.

Task 25 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.

Update the WebLogic Servers Security Settings

This section contains information about WebLogic Servers security settings.

Follow the steps described in Updating the WebLogic Servers Security Settings and update SSL settings for the WLS_WCCUI1 and WLS_WCCUI2 servers.

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 Domain** 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 Domain** 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 Domain** 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 Domain** 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

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 WCCHOST1:16226,WCCHOST2:16226
  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`
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.

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.

Configuration and Management Tasks for All Enterprise Deployments

Complete these common configuration tasks that apply to any Oracle Fusion Middleware enterprise deployment. These tasks include checking the sizing requirements for the deployment, using the JDBC persistence store for web services, and taking backups of the deployment.

Verifying Appropriate Sizing and Configuration for the WLSSchemaDataSource

In Oracle FMW 14.1.2, `WLSRuntimeSchemaDataSource` is the common datasource that is reserved for use by the FMW components for JMS JDBC Stores, JTA JDBC stores, and Leasing services. `WLSRuntimeSchemaDataSource` is used to avoid contention in critical WLS infrastructure services and to guard against dead-locks.

To reduce the `WLSRuntimeSchemaDataSource` 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 *Administering the WebLogic Persistent Store*.

The default `WLSRuntimeSchemaDataSource` 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 WCC 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 `WLSRuntimeSchemaDataSource` 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 or any custom virtual host that maps to a floating IP/VIP. It should not listen on ANY (blank listen address), localhost or any host name that uniquely identifies a single node.

  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 When Using a Per Host Node Manager

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 host Node Manager for the enterprise topology, as described in Creating a Per Host Node Manager Configuration. For more information, 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 on WCCHOST1.
2. Stop the Node Manager on WCCHOST1.

   You can use the script `stopNodeManager.sh` that was created in NM_HOME.
3. Migrate the ADMINVHN virtual IP address to the second host:
   a. Run the following command as root on WCCHOST1 (where X is the current interface used by ADMINVHN) to check the virtual IP address at its CIDR:

```
ip addr show dev ethX
```

For example:

```
ip addr show dev eth0
```

b. Run the following command as root on WCCHOST1 (where *X* is the current interface used by ADMINVHN):

```
ip addr del ADMINVHN/CIDR dev ethX
```

For example:

```
ip addr del 100.200.140.206/24 dev eth0
```

c. Run the following command as root on WCCHOST2:

```
ip addr add ADMINVHN/CIDR dev ethX label ethX:Y
```

For example:

```
ip addr add 100.200.140.206/24 dev eth0 label eth0:1
```

> **Note**
>
> Ensure that the CIDR and interface to be used match the available network configuration in WCCHOST2.

4. Update the routing tables by using `arping`, for example:

```
arping -b -A -c 3 -I eth0 100.200.140.206
```

5. From WCCHOST1, change directory to the Node Manager home directory:

```
cd $NM_HOME
```

6. Edit the `nodemanager.domains` file and remove the reference to *ASERVER_HOME*.

   The resulting entry in the WCCHOST1 `nodemanager.domains` file should appear as follows:

```
wccedg_domain=MSERVER_HOME;
```

7. From WCCHOST2, change directory to the Node Manager home directory:

```
cd $NM_HOME
```

8. Edit the `nodemanager.domains` file and add the reference to *ASERVER_HOME*.

   The resulting entry in the WCCHOST2 `nodemanager.domains` file should appear as follows:

```
wccedg_domain=MSERVER_HOME;ASERVER_HOME
```

9. Start the Node Manager on WCCHOST1 and restart the Node Manager on WCCHOST2.
10. Start the Administration Server on WCCHOST2.
11. Check that you can access the Administration Server on WCCHOST2 and verify the status of components in Fusion Middleware Control using the following URL:

```
https://ADMINVHN:9002/em
```

Validating Access to the Administration Server on WCCHOST2 Through the Load Balancer

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:

- `https://admin.example.com:445/em`

  Where, 445 is the port you use to access to the Fusion Middleware Control in the Load Balancer.

  This URL should display Oracle Enterprise Manager Fusion Middleware Control.
- Verify that you can log into the WebLogic Remote Console through the provider you defined for this domain.

Failing the Administration Server Back to WCCHOST1 When Using a Per Host Node Manager

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.

This procedure assumes that you have configured a per host Node Manager for the enterprise topology, as described in Creating a Per Host Node Manager Configuration. For more information, see About the Node Manager Configuration in a Typical Enterprise Deployment.

1. Stop the Administration Server on WCCHOST2.
2. Stop the Node Manager on WCCHOST2.
3. Run the following command as root on WCCHOST2.

   ```
   ip addr del ADMINVHN/CIDR dev ethX
   ```

   For example:

   ```
   ip addr del 100.200.140.206/24 dev eth0
   ```
4. Run the following command as root on WCCHOST1:

   ```
   ip addr add ADMINVHN/CIDR dev ethX label ethX:Y
   ```

   For example:

   ```
   ip addr add 100.200.140.206/24 dev eth0 label eth0:1
   ```

   > **Note**
   >
   > Ensure that the CIDR and interface to be used match the available network configuration in WCCHOST1.

5. Update the routing tables by using `arping` on WCCHOST1:

   ```
   arping -b -A -c 3 -I eth0 100.200.140.206
   ```

6. From WCCHOST2, change directory to the Node Manager home directory:

   ```
   cd $NM_HOME
   ```

7. Edit the `nodemanager.domains` file and remove the reference to *ASERVER_HOME*.
8. From WCCHOST1, change directory to the Node Manager home directory:

   ```
   cd $NM_HOME
   ```

9. Edit the `nodemanager.domains` file and add the reference to *ASERVER_HOME*.
10. Start the Node Manager on WCCHOST2 and restart the Node Manager on WCCHOST1.
11. Start the Administration Server on WCCHOST1.
12. Test that you can use the WebLogic Remote Console to access the provider defined for this domain.
13. Check that you can access and verify the status of components in the Oracle Enterprise Manager by using the following URL:

    ```
    https://ADMINVHN:9002/em

    https://admin.example.com:445/em
    ```

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.

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 into the WebLogic Remote Console to access the provider of this domain.
2. Open the **Edit Tree**.
3. Expand **Environment**.
4. Expand **Servers**.
5. Click the name of the Managed Server you want to edit. Perform the following steps for each of the Managed Server:
   a. Click the **Advanced** tab.
   b. Click the **Deployment** tab.
   c. Verify that the Staging Directory Name is set to the following:
      `MSERVER_HOME/servers/server_name/stage`

      Replace `MSERVER_HOME` with the full path for the `MSERVER_HOME` directory.

      Update with the correct name of the Managed Server that you are editing.
   d. 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.

e. Click **Save**.

f. Return to the **Summary of Servers** screen.

Repeat the same steps for each of the new managed servers.

6. Navigate to and update the Upload Directory Name value for the **AdminServer**:
   a. Navigate to **Servers** and select the **AdminServer**.
   b. Click the **Advanced** tab.
   c. Click the **Deployment** tab
   d. Verify that the Staging Directory Name is set to the following absolute path: `ASERVER_HOME/servers/AdminServer/stage`
   e. 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.
   f. Click **Save**.
7. When you have modified all the appropriate objects, commit the changes in the shopping cart.
8. Restart all the Servers for the changes to take effect. If you are following the EDG steps in-order and are not going to make any deployments immediately, you can wait until the next restart.

> **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.

About Using Third Party SSL Certificates in the WebLogic and Oracle HTTP Servers

This Oracle WebCenter Content Enterprise Deployment Topology uses SSL all the way from the external clients to the backend WebLogic Servers. The previous chapters in this guide provided scripts (`generate_perdomainCACERTS.sh and generate_perdomainCACERTS-ohs.sh`) to generate the required SSL certificates for the different FMW components.

These scripts generate the different SSL certificates using the WebLogic per domain Certification Authority in the WebLogic domain. These scripts also add the frontend's SSL certificates to the trust keystore. However, in a production environment, you may want to use your own SSL certificates, issued by your own or by a 3rd party certificate authority. This section provides you some guidelines to configure the EDG system with this type of SSL certificates.

Using Third Party SSL Certificates in WebLogic Servers

Here are some guidelines about using custom or third party SSL certificates with the WebLogic Servers:

- The SSL certificate used by each WebLogic server (identity key, private key) must be issued to that server's listen address. For example, if the server *WLS_PROD1* listens in *apphost1.example.com*, the CN of its SSL certificate must be that hostname or wildcard name valid for that hostname.

- Oracle recommends using an identity keystore shared by all the servers in the same domain where you import all the private keys used by the different WebLogic servers each mapped to a different alias.
- Oracle recommends using a trust keystore shared by all the servers in the domain. You must import the Certificate Authority's certificate (and intermediate and root CA if needed) into this trust keystore.
- You must specify the identity keystore, alias of the identity key and the trust keystore for each WebLogic server in the WebLogic domain's configuration. Use WebLogic's Remote Console to configure these SSL settings for each server.
- Start the WebLogic servers using the appropriate java options to point to the trusted keystore so that they can communicate with external SSL endpoints that use the Certificate Authorities included in such a trust store.

The following commands are useful to manage SSL certificates in WebLogic.

- Command to import an SSL certificate (a private key) into the identity keystore:

  Syntax

```
WL_HOME/server/bin/setWLSEnv.sh

java utils.ImportPrivateKey
      -certfile cert_file
      -keyfile private_key_file
      [-keyfilepass private_key_password]
      -keystore keystore
      -storepass storepass
      [-storetype storetype]
      -alias alias
      [-keypass keypass]
```

  Example for a Certificate Issued to `apphost1.example.com`

```
WL_HOME/server/bin/setWLSEnv.sh

java utils.ImportPrivateKey \
-certfile apphost1.example.com_cert.der \
-keyfile apphost1.example.com_key.der \
-keyfilepass keypassword \
-storetype pkcs12 \
-keystore CustomIdentityKeystore.pkcs12 \
-storepass keystorepassword \
-alias apphost1.example.com \
-keypass keypassword
```

- Command to import an SSL certificate (a trusted certificate) into the trusted keystore:

  Syntax

```
keytool -import -v -noprompt -trustcacerts \
-alias <alias_for_trusted_cert> \
-file <certificate>.der \
```

```
-storetype <keystoretype> \
-keystore <customTrustKeyStore> \
-storepass <keystorepassword>
```

Example for Importing a CA Certificate

```
keytool -import -v -noprompt -trustcacerts \
-alias example_ca_cert \
-file example_ca_cert.der \
-storetype pkcs12 \
-keystore CustomTrustKeyStore.pkcs12 \
-storepass keystorepassword
```

Example of the Java Options for Servers to Load Custom Trust Keystore

```
EXTRA_JAVA_PROPERTIES="${EXTRA_JAVA_PROPERTIES}
      -Djavax.net.ssl.trustStore=/u01/oracle/config/keystores/
CustomTrustKeyStore.pkcs12
      -Djavax.net.ssl.trustStorePassword=<keystorepassword>"
export EXTRA_JAVA_PROPERTIES
```

Using Third Party SSL Certificates in Oracle HTTP Servers

Here are some guidelines to use your own SSL certificates in OHS:

- Each OHS virtual host using SSL must use a wallet that contains only one private key. This private key will be used as the OHS server's SSL certificate. It must be issued to the hostname in which the virtual host listens (the hostname value in the "VirtualHost" directive). The private key can also include other hostnames such as Subject Alternative Name (SAN) names (for example, the value of the "ServerName" directive). The virtual host must include the **SSLWallet** directive pointing to this wallet.
- Different OHS virtual hosts can use the same **SSLWallet** (hence, the same private key), as long as they use the same hostname in the VirtualHost directive. The port can be different.
- OHS acts as a client when it connects to the WebLogic servers. Hence, it must trust the certificate authority that issued the WebLogic's certificates. Use the directive **WLSSLWallet** in the `mod_wl_ohs.conf` file to point to the appropriate wallet that contains the WebLogic certificates' CA cert.
- The frontend load balancer acts as a client when it connects to the OHS servers. It must trust the certificate authority that issued the certificates used by OHS. You must check your load balancer documentation to import the OHS's CA as a trusted authority.

The following commands are useful to manage keys and wallets in OHS.

- Command to create a wallet for OHS (orapki):

  Syntax

```
$WEB_ORACLE_HOME/bin/orapki wallet create \
```

```
-wallet wallet \
-auto_login_only
```

Example

```
$WEB_ORACLE_HOME/bin/orapki wallet create \
-wallet /u02/oracle/config/keystores/orapki/ \
-auto_login_only
```

- Command to add a private key to a wallet (orapki) from an identity keystore:

  Syntax

  ```
  $WEB_ORACLE_HOME/bin/orapki wallet jks_to_pkcs12 \
  -wallet wallet \
  -pwd pwd \
  -keystore keystore \
  -jkspwd keystorepassword
  [-aliases [alias:alias..]]
  ```

  Example

  ```
  $WEB_ORACLE_HOME/bin/orapki wallet jks_to_pkcs12 \
  -wallet /u02/oracle/config/keystores/orapki/ \
  -keystore /u02/oracle/config/keystores/customIdentityKeyStore.pkcs12 \
  -jkspwd keystorepassword \
  -aliases ohshost1.example.com
  ```

- Command to add all the trusted keys to a wallet (orapki) from a trusted keystore:

  Example

  ```
  $WEB_ORACLE_HOME/bin/orapki wallet jks_to_pkcs12 \
  -wallet /u02/oracle/config/keystores/orapki/ \
  -keystore /u02/oracle/config/keystores/customTrustKeyStore.pkcs12 \
  -jkspwd password
  ```

- Command to list all the keys of a wallet (orapki):

  Example

  ```
  $WEB_ORACLE_HOME/bin/orapki wallet display \
  -wallet /u02/oracle/config/keystores/orapki/
  ```

Enabling SSL Communication Between the Middle Tier and SSL Endpoints

It is important to understand how to enable SSL communication between the middle tier and the front-end hardware load balancer or any other external SSL endpoints that needs to be

accessed by the WebCenter Content WebLogic Server. For example, for external web services invocations, callbacks, and so on.

> **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 frontend 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.

For example, the following examples are applicable in an Oracle WebCenter Content enterprise deployment:

- Oracle SOA Suite composite applications and services often generate callbacks that need to perform invocations by using the SSL address exposed in the load balancer.
- Oracle SOA Suite composite applications and services often access external webservices using SSL.
- Finally, when you test a SOA Web services endpoint in Oracle Enterprise Manager Fusion Middleware Control, the Fusion Middleware Control software that is running on the Administration Server must access the load balancer frontend to validate the endpoint.

Generating Certificates, Identity Store, and Truststores

Since this Enterprise Deployment Guide uses end to end SSL (except in the access to the Database), certificates have already been generated in the different chapters using a per-domain CA. These have been already added to the pertaining Identity Stores and a Truststore has also been configured to include the per-domain CA. It is expected that through the use of the different generateCerts scripts provided, appropriate certificates exist already in these stores for the different listen addresses used by the WebLogic servers in the domain. On top of this, when the script `generate_perdomainCACERTS-ohs.sh` is executed, it traverses all the front-end addresses in the domain's `config.xml` and adds its pertaining certificates to the trust store used by the domain. By adding these trust stores to the java properties used by the WebLogic Servers in the domain (`-Djavax.net.ssl.trustStore` and `-Djavax.net.ssl.trustStorePassword`), the appropriate SSL handshake is guaranteed when these WebLogic servers acts as client sin SSL invocations.

Importing Other External Certificates into the Truststore

Perform the following steps to add any other SSL end point's certificates to the domain's truststore. These may be external addresses or frontends in other WLS domains used by the applications in the WebCenter Content EDG one:

1. Access the end point's site on SSL with a browser (this adds the server's certificate to the browser's repository).

2. Obtain the certificate from the site. For example, you can obtain a webservice site's certificate using a browser such as Firefox. 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 `site.webservice.com.crt`). Alternatively, you can obtain the certificate using the openssl command. The syntax of the commands is as follows:

```
openssl s_client -connect site.webservice.com -showcerts </dev/null 2>/dev/
null|openssl x509 -outform PEM > $KEYSTORE_HOME/ site.webservice.com.crt
```

3. Use the keytool to import the site's certificate into the truststore:

   For example:

```
keytool -import -file /oracle/certificates/site.webservice.com.crt -v -
keystore appTrustKeyStore.pkcs12 -alias siteWS -storepass password
```

4. Repeat this procedure for each SSL endpoint accessed by your WebLogic Servers.

> **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

Since the trust store's path was already added to the WebLogic start scripts in the chapter where the domain was created, no additional configuration is required. Simply ensure that the new trust store (with the CAs and/or certs for the SSL endpoints added) replaces the existing one.

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.

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.

Adding the Enterprise Deployment Administration User to a Product-Specific Administration Group

For products with a product-specific administration group, use the following procedure to add the enterprise deployment administration user (`weblogic_wcc` to the group. This allows you to manage the product by using the enterprise manager administrator user:

1. Create an **ldif** file called `product_admin_group.ldif` similar to the following:

```
dn: cn=product-specific_group_name, cn=groups, dc=us, dc=oracle, dc=com
displayname: product-specific_group_display_name
objectclass: top
objectclass: groupOfUniqueNames
objectclass: orclGroup
uniquemember: cn=weblogic_wcc,cn=users,dc=us,dc=oracle,dc=com
cn: product-specific_group_name
description: Administrators Group for the Domain
```

Replace *product-specific_group_display_name* with the display name for the group that appears in the management console for the LDAP server and in the Oracle WebLogic Remote Console.

2. Use the **ldif** file to add the enterprise deployment administrator user to the product-specific administration group.

   For Oracle Unified Directory:

```
OUD_INSTANCE_HOME/bin/ldapmodify -a
                                 -D "cn=Administrator"
                                 -X
                                 -p 1389
                                 -f product_admin_group.ldif
```

   For Oracle Internet Directory:

```
OID_ORACLE_HOME/bin/ldapadd -h oid.example.com
                            -p 389
                            -D cn="orcladmin"
                            -w <password>
                            -c
                            -v
                            -f product_admin_group.ldif
```

Using Persistent Stores for TLOGs and JMS in an Enterprise Deployment

The Oracle WebLogic persistent store framework 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. It needs to be noted that the configuration wizard steps provided in the different chapters in this book will already create JDBC persistent stores for the components used. Use the manual steps below for custom stores or for transitioning to JDBC stores from file stores.

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.

> **Note**
>
> Remember that the steps provided for setting up the different components in this EDG (using the configuration wizard) is already configured in JDBC persistent stores for them. Use the following steps for custom persistent stores or when reconfiguring from file stores to JDBC stores (migration of messages from file to JDBC is out of the scope of this EDG).

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 `WLSRuntimeSchemaDataSource` as is for TLOGs and JMS persistent stores under non-high workloads and consider increasing the `WLSRuntimeSchemaDataSource` 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 `WLSRuntimeSchemaDataSource` 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 `WLSRuntimeSchemaDataSource` 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. Log into the WebLogic Remote Console.
2. Navigate to the **Edit Tree**.
3. In the structure tree, expand **Services** and select **Data Sources**.
4. In the Summary of **Data Sources** page, click **New** and select **GridLink Data Source**. Enter the following:

Table 18-1 GridLink Data Source Properties

Properties	Description
Name	Enter a logical name for the data source in the Name field. For example, Leasing.
JNDI Names	Enter a name for JNDI. For example, for the TLOGs store enter jdbc/tlogs. For the JMS store, enter jdbc/jms.
Targets	Select the cluster that is using the persistent store and move to "Chosen".
Data Source Type	Select **GridLink Data Source**.
Database Driver	Select **Oracle's Driver (Thin) for GridLink Connections Versions: Any**.
Global Transaction Protocol	Select **None**.
Listeners	Enter the SCAN address and port for the RAC database, separated by a colon. For example, db-scan.example.com:1521.
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`.
Database username	Enter the user name. For example, 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.

Table 18-1 (Cont.) GridLink Data Source Properties

Properties	Description
Protocol	Leave the default value (TCP).
Fan Enabled	This property must be checked.
ONS Nodes	You can leave this field empty. ONS node list is automatically retrieved when the database is 12.2 or higher version.
ONS Wallet and password	You can leave this field empty.
Test Configuration	You must enable this option.

5. Click **Create**.
6. Commit changes in the shopping cart.
7. Repeat *Step 4* to *Step 6* to create the GridLink Data Source for JMS File Stores.

Assigning the TLOGs JDBC Store to the Managed Servers

If you are going to accomplish data source consolidation, you will reuse the `<PREFIX>_WLS` tablespace and `WLSRuntimeSchemaDataSource` for the TLOG persistent store. Otherwise, ensure that you create the tablespace and user in the database, and you have created the datasource before you assign the TLOG store to each of the required Managed Servers.

1. Log into the Oracle WebLogic Remote Console.
2. In the **Edit Tree**, navigate to **Environment** > **Servers**.
3. Click the name of the Managed Server.
4. Select the **Services** > **JTA** tab.
5. Enable **Transaction Log Store in JDBC**.
6. In the **Data Source** menu, select **WLSSchemaRuntimeDatasource** to accomplish data source consolidation. The `<PREFIX>_WLS` tablespace will be used for TLOGs.
7. In the **Transaction Log Prefix Name** field, specify a prefix name to form a unique **JDBC TLOG** store name for each configured JDBC TLOG store.
8. Click **Save**.
9. Repeat *step 2* to *step 7* for each additional managed server.
10. To activate these changes, commit the changes in the shopping cart.

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 WebLogic Remote Console to create the store.

1. Log into the WebLogic Remote Console.
2. Navigate to the **Edit Tree**.
3. In the structure tree, expand **Services** and select **JDBC Stores**.
4. Click **New**.
5. Enter a persistent store name that easily relates it to the pertaining JMS servers that is using it.

> **Note**
>
> To accomplish data source consolidation, select `WLSRuntimeSchemaDataSource`. The `<PREFIX>_WLS` tablespace is used for JMS persistent stores.

6. Target the store to the migratable target to which the JMS server belongs.
7. Repeat *Step 3* to *Step 7* for each additional JMS server in the cluster.
8. Commit changes in the shopping cart.

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 into the WebLogic Remote Console.
2. Navigate to the Edit Tree.
3. In the structure tree, expand **Services** > **Messaging** > **JMS Servers**.
4. Click the name of the JMS Server that you want to use the persistent store.
5. In the **Persistent Store** property, select the JMS persistent store you created.
6. Click **Save**.
7. Repeat *Step 3* to *Step 6* for each of the additional JMS Servers in the cluster.
8. To activate these changes, commit changes in the shopping cart.

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 #unique_383, 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 Remote Console to edit the existing JDBC Store you created earlier; create the table that is used for the JMS data:
   a. Log into the WebLogic Remote Console.
   b. Navigate to the **Edit Tree**.
   c. In the structure tree, expand **Services** and select **JDBC stores**.
   d. Click the persistent store you created earlier.
   e. Click **Show Advanced Fields**.
   f. Under the **Advanced** options, enter `ORACLE_RUNTIME/domain_name/ddl/jms_custom.ddl` in the **Create Table from DDL File** field.
   g. Click **Save**.
   h. To activate these changes, commit changes in the shopping cart.
5. Restart the Managed Servers.

About JDBC Persistent Stores for Web Services

By default, web services use the WebLogic Server default persistent store for persistence. This store provides high-performance storage solution for web services.

The default web service persistence store is used by the following advanced features:

- Reliable Messaging
- Make Connection
- SecureConversation
- Message buffering

You also have the option to use a JDBC persistence store in your WebLogic Server web service, instead of the default store. For information about web service persistence, see Managing Web Service Persistence.

Best Configuration Practices When Using RAC and Gridlink Data Sources

Oracle recommends that you use GridLink data sources when you use an Oracle RAC database. If you follow the steps described in the Enterprise Deployment guide, the data sources will be configured as GridLink.

GridLink data sources provide dynamic load balancing and failover across the nodes in an Oracle Database cluster, and also receive notifications from the RAC cluster when nodes are added or removed. For more information about GridLink data sources, see Using Active GridLink Data Sources in *Administering JDBC Data Sources for Oracle WebLogic Server*.

Here is a summary of the best practices when using GridLink to connect to the RAC database:

- Use a database service (defined with `srvctl`) different from the default database service
  In order to receive and process notifications from the RAC database, the GridLink needs to connect to a database service (defined with `srvctl`) instead to a default database service. These services monitor the status of resources in the database cluster and generate notifications when the status changes. A database service is used in Enterprise Deployment guide, created and configured as described in Creating Database Services.
- Use the long format database connect string in the data sources
  When Gridlink data sources are used, the long format database connect string must be used. The Configuration Wizard does not set the long format string, it sets the short format instead. You can modify it manually later to set the long format. To update the data sources:
  1. Connect to the WebLogic Remote Console and navigate to **Domain Structure** > **Services** > **Datasources**.
  2. Select a data source, click the **Configuration** tab, and then click the **Connection Pool** tab.
  3. Within the JDBC URL, change the **URL** from `jdbc:oracle:thin:[SCAN_VIP]:[SCAN_PORT]/[SERVICE_NAME]` to
     `jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=[SCAN_VIP])(PORT=[SCAN_PORT])))(CONNECT_DATA=(SERVICE_NAME=[SERVICE_NAME])))`
     For example:

     ```
     jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(LOAD_BALANCE=ON)(ADDRESS=(PROTOCOL=TCP)(HOST=db-scan-address)(PORT=1521)))(CONNECT_DATA=(SERVICE_NAME=wccedg.example.com)))
     ```

- Use auto-ons
  The ONS connection list is automatically provided from the database to the driver. You can leave the ONS Nodes list empty in the data sources configuration.
- Test Connections On Reserve
  Verify that the **Test Connections On Reserve** is checked in the data sources.

  Eventhough the GridLink data sources receive FAN events when a RAC instances becomes unavailable, it is a best practice to enable the **Test Connections On Reserve** in the data source and ensure that the connection returned to the application is good.
- Seconds to Trust an Idle Pool Connection
  For a maximum efficiency of the test, you can also set **Seconds to Trust an Idle Pool Connection** to `0`, so the connections are always verified. Setting this value to zero means that all the connections returned to the application will be tested. If this parameter is set to

10, the result of the previous test will be valid for 10 seconds and if a connection is reused before the lapse of 10 seconds, the result will still be valid.

- Test Frequency
  Verify that the **Test Frequency** parameter value in the data sources is not 0. This is the number of seconds a WebLogic Server instance waits between attempts when testing unused connections. The default value of 120 is normally enough.

Using TNS Alias in Connect Strings

You can create an alias to map the URL information instead of specifying long database connection strings in the jdbc connection pool of a datasource. The connection string information is stored in a tnsnames.ora file with an associated alias name. This alias is used in the connect string of the connection pool.

The following example is of a connect string using tns alias.

```
jdbc:oracle:thin:@wccedg_alias
```

The tnsnames.ora file contains the following details.

```
wccedg_alias =
        (DESCRIPTION=
        (ADDRESS_LIST=
            (LOAD_BALANCE=ON)
            (ADDRESS=(PROTOCOL=TCP)(HOST=wccedgdb-scan)(PORT=1521)))
            (CONNECT_DATA=(SERVICE_NAME=wccedg.example.com))
        )
```

You must specify the oracle.net.tns_admin property in the datasource configuration to point to a specific tnsnames.ora file. For example, <property><name>oracle.net.tns_admin</name><value>/u01/oracle/config/domains/fmw1412edg/config/tnsadmin</value></property></properties>

This is the Maximum Availability and Enterprise Deployment recommended approach for JDBC urls. It simplifies JDBC configurations, facilitates DB configuration aliasing in disaster protection scenarios, and makes database connection changes more dynamic. For more information, see Use a TNS Alias Instead of a DB Connection String in *Administering JDBC Data Sources for Oracle WebLogic Server*.

In Oracle Fusion Middleware 14.1.2, you can use a new type of deployment module to manage the tnsnames.ora files, wallet files, and keystore and truststore files associated with a database connection. These are called DBClientData modules. For more information, see What Are DBClientData Modules in *Administering JDBC Data Sources for Oracle WebLogic Server*. In this EDG, DBClientData type of module is used to maintain the database client information. However, wallets and SSL configuration is not used to access the database so the DBClientData module contains only the appropriate tnsnames.ora.

The following steps are required to use a TNS alias in the different Datasources used by FMW and WLS schemas:

1. Create a tnsnames.ora with the pertaining alias and mapping URLs used in the connection pools. Copy the connect string from one of the existing datasource configuration files. For example,

> **Note**
>
> This is an example using the short jdbc URL.

```
[oracle@soahost1~]$  grep url  /u01/oracle/config/domains/wccedgdomain/
config/jdbc/opss-datasource-jdbc.xml
    <url>jdbc:oracle:thin:@drdbrac12a-
scan.dbsubnet.vcnlon80.oraclevcn.com:1521/wccedg.example.com</url>
[oracle@soahost1~]$
```

Use the information in the connect string to add a long URL entry to a `tnsnames.ora` file. Use an alias name that identifies your connection. Notice that in order to deploy the `tsnnames.ora` as DBCLient module the location of the deployment module needs to be two levels down under the domain config directory if it resides on the WLS Administration Server node. The file can also be created in the node that runs the WebLogic Remote Console and can also be uploaded (as an application ear or war file).

```
[oracle@soahost1~]$  cat /u01/oracle/config/tnsadmin/tnsnames.ora

wccedg_alias =
        (DESCRIPTION=
        (ADDRESS_LIST=
            (LOAD_BALANCE=ON)
            (ADDRESS=(PROTOCOL=TCP)(HOST= drdbrac12a-
scan.dbsubnet.vcnlon80.oraclevcn.com)(PORT=1521)))
            (CONNECT_DATA=(SERVICE_NAME=wccedg.example.com))
        )
```

2. Deploy the directory containing the `tnsnames.ora` as a DBClientData module.
   a. Access the domain provider in the WebLogic Remote Console.
   b. Click **Edit Tree**.
   c. Click **Environment** > **Deployments** >**Database Client Data Directories**.
   d. Click **New**.
   e. Enter a name for the dbclient directory deployment. For example, `dbclientdata_modulename`.

      If the directory containing the `tnsnames.ora` file resides on your local computer, uncheck the **Upload** checkbox.
   f. Click **Create**.
   g. Click **Save**.

      The cart on the top right part of the screen will display full with a yellow bag inside.
   h. Click the **Cart** icon and select **Commit Changes**.

      This will create a `tnsnames/dbclient` module under domain dir `/u01/oracle/config/domains/wccedgdomain/config/ dbclientdata/dbclientdata_modulename`.

You can also perform the deployment of a database client module using the `deploy` command in wlst.

3. Update the different Datasources and fmwconfig files to use the alias instead of the explicit URLS.

> **Note**
>
> To update a datasource to use the tns alias, the datasource configuration needs to include both a pointer to the `tsnames.ora` file and the alias itself in the jdbc URL.

You must perform the following steps to include a pointer to the `tnsnames.ora` file include the property `oracle.net.tns_admin` in the datasource properties.

a. Access the domain provider in the WebLogic Remote Console.
b. Click **Edit Tree**.
c. Click **Services** > **Datasources** > **Datasource_name**.
d. In the navigation tree on the left, select **Properties** for the precise Datasource.
e. Click **New**.
f. Enter `oracle.net.tns_admin` as the property name.
g. Click **Create**.
h. In the next screen with the property details, enter as **value** the directory for the `dbclientdata_modulename` that is `/u01/oracle/config/domains/wccedgdomain/config/ dbclientdata/dbclientdata_modulename` in the example above.
i. Click **Save**.

   The cart on the top right part of the screen will display full with a yellow bag inside.
j. In the navigation tree on the left, click the **Datasource** name.
k. Select the **Connection Pool** tab.
l. In the URL, replace the URL with the alias syntax as shown below:

   ```
   jdbc:oracle:thin:@wccedg_alias
   ```
m. Click **Save**.
n. Click the **Cart** icon and select **Commit Changes**.

   If you check the datasource configuration file, it should reflect the following under the `<jdbc-driver-params> <properties>` entries:

   ```
   <property>
   <name>oracle.net.tns_admin</name>
   <value>/u01/oracle/config/domains/wccedgdomain/config/dbclientdata/
   dbclientdata_modulename</value>
   </property>
   ```

   The datasource configuration file should reflect as JDBC URL under `<jdbc-driver-params>` as shown below:

   ```
   <url>jdbc:oracle:thin:@wccedg_alias</url>
   ```

4. To update the FMW jps config to use the tns alias, the `domain_path/config/fmwconfig/jps-config.xml` and `domain_path/config/fmwconfig/jps-config-jse.xml` files need to be updated and both a pointer to the `tsnames.ora` file and the alias itself must be included in the jdbc url, that is replace the information in the propertySet for the DB with the updated URL and the tnsadmin pointer.

```
<property name="oracle.net.tns_admin" value="/u01/oracle/config/domains/
wccedgdomain/config/dbclientdata/dbclientdata_modulename "/>
<property name="jdbc.url" value="jdbc:oracle:thin:@wccedg_alias "/>
```

Restart the Administration Server for all the changes to be applied.

Alternatively, you can use the `https://github.com/oracle-samples/maa/tree/main/1412EDG/fmw1412_change_to_tns_alias.sh` script instead of the steps 1, 2, 3 and 4 to deploy the corresponding DBClientData module and replace all urls in the jdbc and jps configuration with the pertaining alias.

However, using the script is only recommended when all domain extensions have been completed and all the required datasources are present in the domain configuration because the script is configured to exit if an existing tnsadmin already exists in the configuration files. This behavior is intentional to avoid conflicts with other DBClient modules in the domain.

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 *Administering Oracle Fusion Middleware*:

- Backing Up Your Environment
- Recovering Your Environment

Table 18-2 lists the static artifacts to back up in a typical Oracle WebCenter Content enterprise deployment.

Table 18-2 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

Table 18-2 (Cont.) Static Artifacts to Back Up in the Oracle WebCenter Content Enterprise Deployment

Type	Host	Tier
Installation-related files	WEBHOST1, WEHOST2, and shared storage	N/A

Table 18-3 lists the runtime artifacts to back up in a typical Oracle WebCenter Content enterprise deployment.

Table 18-3 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 Configuration directory	WEBHOST1 and WEBHOST2	Web Tier

Online Domain Run-Time Artifacts Backup/Recovery Example

This section describes an example procedure to implement a backup of the WebLogic domain artifacts. This approach can be used during the EDG configuration process, for example, before extending the domain to add a new component.

This example has the following features:

- App tier Runtime Artifacts are backed up/recovered in this example:

Artifact	Host	Tier
Administration Server domain home (ASERVER_HOME)	WCCHOST1 (or NAS Filer)	Application Tier
Application home (APPLICATION_HOME)	WCCHOST1 (or NAS Filer)	Application Tier
Deployment Plan home (DEPLOY_PLAN_HOME)	WCCHOST1 (or NAS Filer)	Application Tier
Runtime artifacts (adapter control files) (ORACLE_RUNTIME)	WCCHOST1 (or NAS Filer)	Application Tier
Scripts and Customizations	Per host	Application Tier

- This backup procedure is suitable for cases when a major configuration change is done to the domain (that is, domain extension). If something goes wrong, or if you make incorrect selections, you can restore the domain configuration to the earlier state.
  Database backup/restore is not mandatory for this sample procedure, but steps to backup/restore the database are included as optional.

Artifact	Host	Tier
Oracle RAC database (optional)	Oracle RAC database (optional)	Data Tier

- Operating system tools are used in this example. Some of the run-time artifacts listed in this section are hosted from Network Attached Storage (NAS). If possible, do the backup and recovery of these volumes from the NAS filer directly rather than from the application servers.
- Managed servers are running during the backup. MSERVER_HOME is not backed up and pack/unpack procedure is used later to recover MSERVER_HOME. Therefore, managed server lock files are not included in the backup.
- AdminServer can be running during the backup if `.lok` files are excluded from the backup. To avoid an inconsistent backup, do not make any configuration changes until the backup is complete. To ensure that no changes are made in the WebLogic Server domain, you can lock the WebLogic Server configuration.

> **Note**
>
> Excluding these:
>
> – `AdminServer/data/ldap/ldapfiles/EmbeddedLDAP.lok`
>
> – `AdminServer/tmp/AdminServer.lok`

Back Up the Domain Run-Time Artifacts

To backup the domain runtime artifacts, perform the following steps:

1. Log in to WCCHOST1 with user *oracle* and ensure that you define and export the following variables:

Variable	Example Value	Description
BAK_TAG	BEFORE_BPM	Descriptive tag used in the names of the backup files and database restore point.
BAK_DIR	/backups	Host folder where backup files are stored.
DOMAIN_NAME	wccedg_domain	Domain name

For example:

```
export BAK_TAG=BEFORE_BPM
export DOMAIN_NAME=wccedg_domain
export BAK_DIR=/backups
```

2. Ensure that the following domain variables are set with the values of the domain:

Variable	Example Value
ASERVER_HOME	`/u01/oracle/config/domains/wccedg_domain`
DEPLOY_PLAN_HOME	`/u01/oracle/config/dp`

Variable	Example Value
APPLICATION_HOME	`/u01/oracle/config/applications/wccedg_domain`
ORACLE_RUNTIME	`/u01/oracle/runtime`

See Table 7-2.

3. Before you make the backup, lock the domain configuration, so you prevent other accounts from making changes during your edit session. To lock the domain configuration from Fusion Middleware Control:
   a. Log in to `https://admin.example.com:445/em`.
   b. Locate the Change Center at the top of Fusion Middleware Control.
   c. From the **Changes** menu, select **Lock & Edit** to lock the configuration edit for the domain.

> **Note**
>
> To avoid an inconsistent backup, do not make any configuration changes until the backup is complete.

4. Log in to WCCHOST1 and clean the logs and backups applications before the backup:

```
find ${ASERVER_HOME}/servers/AdminServer/logs -type f -name "*.out0*" ! -
size 0c -print -exec rm -f {} \+
find ${ASERVER_HOME}/servers/AdminServer/logs -type f -name "*.log0*" ! -
size 0c -print -exec rm -f {} \+
find ${APPLICATION_HOME} -type f -name "*.bak*" -print -exec rm -f {} \;
```

5. Perform the backup of each artifact by using `tar`:

```
tar -cvzf  ${BAK_DIR}/backup_aserver_home_${DOMAIN_NAME}_${BAK_TAG}.tgz   $
{ASERVER_HOME} --exclude ".lok"

tar -cvzf ${BAK_DIR}/backup_dp_home_${DOMAIN_NAME}_${BAK_TAG}.tgz    $
{DEPLOY_PLAN_HOME}/${DOMAIN_NAME}

tar -cvzf ${BAK_DIR}/backup_app_home_${DOMAIN_NAME}_${BAK_TAG}.tgz   $
{APPLICATION_HOME}

tar -cvzf ${BAK_DIR}/backup_runtime_${DOMAIN_NAME}_${BAK_TAG}.tgz         $
{ORACLE_RUNTIME}/${DOMAIN_NAME}

ls --format=single-column ${BAK_DIR}/backup_aserver_*.tgz
ls --format=single-column ${BAK_DIR}/backup_dp_*.tgz
ls --format=single-column ${BAK_DIR}/backup_app_*.tgz
ls --format=single-column ${BAK_DIR}/backup_runtime_*.tgz
```

6. Release the domain lock.
   a. Log in to `https://admin.example.com:445/em`.
   b. Locate the Change Center at the top of Fusion Middleware Control.

c. From the **Changes** menu, select **Release Configuration** to release the configuration edit for the domain.

7. Backup your scripts and customizations, if needed.
8. (**Optional**) Log in to the database and create a flashback database restore point:

> **Note**
>
> Flash database technology is used in this example for database recovery. Check your database version's documentation for more information about Flashback.

a. Create flashback guaranteed checkpoint.

```
sqlplus / as sysdba
SQL> create restore point BEFORE_BPM guarantee flashback database;
SQL> alter system switch logfile;
```

b. Verify.

```
SQL> set linesize 300
SQL> column name format a30
SQL> column time format a32
SQL> column storage_size format 999999999999
SQL> SELECT name, guarantee_flashback_database, time, storage_size FROM
v$restore_point ORDER BY time;

Example:
NAME                           GUA TIME
STORAGE_SIZE
------------------------------ --- --------------------------------
-------------
SOAEDG_BEFORE_BPM              YES 12-MAY-17 03.29.28.000000000 AM
8589934592
exit
```

Restore the Domain Run-Time Artifacts

To recover the domain to the point where the backups where made, follow these steps:

1. Log in to WCCHOST1 using the oracle user.
2. Stop all the servers in the domain, including the AdminServer.
3. Ensure that the following domain variables are set with the values of the domain:

Variable	Example Value
ASERVER_HOME	`/u01/oracle/config/domains/wccedg_domain`
DEPLOY_PLAN_HOME	`/u01/oracle/config/dp`
APPLICATION_HOME	`/u01/oracle/config/applications/wccedg_domain`
ORACLE_RUNTIME	`/u01/oracle/runtime`

4. Remove the current folders by renaming them. You can remove these folders completely at the end of the process after you have verified the recovered domain.

   a. In WCCHOST1:

   ```
   mv  ${ASERVER_HOME}  ${ASERVER_HOME}_DELETE
   mv  ${DEPLOY_PLAN_HOME}/${DOMAIN_NAME}  ${DEPLOY_PLAN_HOME}/$
   {DOMAIN_NAME}_DELETE
   mv  ${APPLICATION_HOME}   ${APPLICATION_HOME}_DELETE
   mv  ${ORACLE_RUNTIME}/${DOMAIN_NAME} ${ORACLE_RUNTIME}/$
   {DOMAIN_NAME}_DELETE
   ```

   b. In each WCCHOSTn:

   ```
   mv   ${MSERVER_HOME}  ${MSERVER_HOME}_DELETE
   ```

5. Locate and identify the backups in the backup folder. Ensure that you define and export the following variables with the correct values of the backup you want to recover:

Variable	Example Value	Description
BAK_TAG	BEFORE_BPM	Descriptive tag used in the names of the backup files and database restore point.
BAK_DIR	/backups	Host folder where backup files are stored.
DOMAIN_NAME	wccedg_domain	Domain name

   For example:

   ```
   export BAK_TAG=BEFORE_BPM
   export DOMAIN_NAME=wccedg_domain
   export BAK_DIR=/backups
   ```

6. Perform the recovery of the files by extracting the files.

   > **Note**
   >
   > TAR files will recreate the structure beginning with `/`, so you need to go to `/` folder.

   ```
   cd  /
   tar -xzvf ${BAK_DIR}/backup_aserver_home_${DOMAIN_NAME}_${BAK_TAG}.tgz
   tar -xzvf ${BAK_DIR}/backup_dp_home_${DOMAIN_NAME}_${BAK_TAG}.tgz
   tar -xzvf ${BAK_DIR}/backup_app_home_${DOMAIN_NAME}_${BAK_TAG}.tgz
   tar -xzvf ${BAK_DIR}/backup_runtime_${DOMAIN_NAME}_${BAK_TAG}.tgz
   ```

7. (**Optional**) If you need to recover the database to the flashback recovery point, perform the following steps:

   a. Log in to DBHOST with oracle user and stop the database:

   ```
   srvctl stop database -database wccedgdb
   ```

b. Log in to the database and flashback database to the restore point:

```
sqlplus / as sysdbaSQL>
          startup mountSQL>
          FLASHBACK DATABASE TO RESTORE POINT BEFORE_BPM;
    Flashback complete.
```

c. Start database with this command:

```
SQL> ALTER DATABASE OPEN RESETLOGS;
```

8. Start AdminServer:

```
${ORACLE_COMMON_HOME}/common/bin/wlst.sh
wls:/offline> nmConnect('nodemanager','password','ADMINVHN','5556',
'domain_name','ASERVER_HOME','PLAIN')
Connecting to Node Manager ...
Successfully Connected to Node Manager.
wls:/nm/domain_name > nmStart('AdminServer')
```

9. Propagate the domain to the Managed Servers.

a. Sign in to WCCHOST1 and run the `pack` command to create the template, as follows:

```
cd ${ORACLE_COMMON_HOME}/common/bin
./pack.sh -managed=true
          -domain=ASERVER_HOME \
          -template=/full_path/recover_domain.jar \
          -template_name=recover_domain_template \
          -log_priority=DEBUG \
          -log=/tmp/pack.log
```

- 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 where you want to create the domain template jar file.
- `recover_domain.jar` is an example of the name for the jar file that you are creating.
- `recover_domain_template` is an example of the name for the jar file that you are creating.

b. Run the `unpack` command in every WCCHOST, as follows:

```
cd $ORACLE_COMMON_HOME/common/bin

./unpack.sh -domain=MSERVER_HOME \
            -overwrite_domain=true \
            -template=/full_path/recover_domain.jar
            -log_priority=DEBUG \
            -log=/tmp/unpack.log \
            -app_dir=APPLICATION_HOME \
```

- 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/ recover_domain.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.

10. Recover/perform customizations, if needed.
11. Start the servers and verify the domain.
12. After checking that everything is correct, you can delete the previous renamed folders:
    a. In WCCHOST1:

```
rm -rf   ${ASERVER_HOME}_DELETE
rm -rf   ${KEYSTORE_HOME}_DELETE
rm -rf   ${DEPLOY_PLAN_HOME}/${DOMAIN_NAME}_DELETE
rm -rf   ${APPLICATION_HOME}_DELETE
rm -rf   ${ORACLE_RUNTIME}/${DOMAIN_NAME}_DELETE
```

    b. In every WCCHOSTn:

```
rm -rf   ${MSERVER_HOME}_DELETE
```

19

Using Service Migration in an Enterprise Deployment

The Oracle WebLogic Server migration framework supports Whole Server Migration and Service Migration. Whole Server Migration requires more resources and a full start of a managed server, so it involves a higher failover latency than Service Migration. The products included in this EDG support Service Migration. Hence, Service Migration is recommended and this guide explains how to use Service Migration in an Oracle Fusion Middleware enterprise topology. Whole Server migration is out of the scope of this guide.

About 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

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 *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 *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

Automatic Service Migration requires a data source for the leasing table, which is a table that resides in a tablespace, and schema 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 data source is already configured with the `FMW1412_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 WebLogic Remote Console.
2. Navigate to the **Edit Tree**.
3. In the structure tree, expand **Services** and select **Data Sources**.
4. On the Summary of **Data Sources** page, click **New** and select **GridLink Data Source**. Enter the following:

Table 19-1 GridLink Data Source Properties

Properties	Description
Name	Enter a logical name for the data source in the Name field. For example, Leasing.
JNDI Names	Enter a name for JNDI. For example, jdbc/leasing.
Targets	Select the cluster that you are configuring for Automatic Service Migration and move to "Chosen".
Data Source Type	Select **GridLink Data Source**.
Database Driver	Select **Oracle's Driver (Thin) for GridLink Connections Versions: Any**.
Global Transaction Protocol	Select **None**.
Listeners	Enter the SCAN address and port for the RAC database, separated by a colon. For example, `db-scan.example.com:1521`.
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, `soaedg.example.com`.

Table 19-1 (Cont.) GridLink Data Source Properties

Properties	Description
Database username	Enter the user name of the WLS Runtime schema. For example, `FMW1412_WLS_RUNTIME`. In this example, `FMW1412` is the prefix you used when you created the schemas as you prepared to configure the domain. **Note** 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.
Protocol	Leave the default value (TCP).
Fan Enabled	You must check this option.
ONS Nodes	Leave it empty. ONS node list is automatically retrieved when the database is 12.2 or higher version.
ONS Wallet and password	You can leave this field empty.
Test Configuration	You must enable this option.

5. Click **Create**.
6. Commit changes in the shopping cart.

Configuring Automatic Service Migration in an Enterprise Deployment

The services used by the different SOA components in this Enterprise Deployment Guide are already configured with Automatic Service Migration when following the Configuration Wizard steps provided in this guide. For any other custom services, you can use the following steps to configure service migration.

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.

> **Note**
>
> To accomplish data source consolidation and connection usage reduction, you can reuse the `WLSRuntimeSchemaDataSource` datasource as is for database leasing. This datasource is already configured with the `FMW1412_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 `WLSRuntimeSchemaDataSource` or a custom datasource that you created as described in Creating a GridLink Data Source for Leasing.

1. Log into the WebLogic Remote Console.
2. Navigate to the **Edit Tree**.
3. In the structure tree, expand **Environment** > **Clusters**.
4. The Summary of **Clusters** page appears. 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. In 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 `WLSRuntimeSchemaDataSource` for data source consolidation.
8. Click **Save**.
9. Commit changes in the shopping cart
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 JTA Migration Settings for the Managed Servers in the Cluster

After you set the leasing mechanism and data source for the cluster, you can 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. Log into the WebLogic Remote Console.
2. Navigate to the **Edit Tree**.
3. In the structure tree, expand the **Environment** node and click **Servers**.

   The **Summary of Servers** page appears.
4. Expand the name of the server you want to modify.
5. Navigate to **JTA Migratable Target**.
6. In the the **JTA Migration Policy** drop-down menu, select **Failure Recovery**.

In the **JTA Candidate Servers** section of the page, leave the **Chosen** list box empty. If you do not select any servers from the **Available** list box, all the available servers in the cluster become candidates for service migration.

7. Click **Save**.
8. Commit the changes in the shopping cart.
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 **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 **failure-recovery** policy.

See Policies for Manual and Automatic Service Migration in *Administering Clusters for Oracle WebLogic Server*.

Setting the Service Migration Policy for Each Managed Server in the Cluster

After you modify the JTA 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 Remote Console:

1. Log into the WebLogic Remote Console.
2. Navigate to the **Edit Tree**.
3. In the structure tree, expand **Environment** > **Migratable Targets**.
4. Click the name of the first Managed Server in the cluster.
5. In the **Service Migration Policy** drop-down menu, select the appropriate policy for the cluster. See About Selecting a Service Migration Policy.
6. In the **Candidate** tab, leave the **Chosen** list box empty. If you do not select any servers from the **Available** list box, all the available servers in the cluster become candidates for service migration.
7. Click **Save**.
8. Repeat *Step 2* to *Step 6* for each of the additional Managed Servers in the cluster.
9. Commit changes in the shopping cart.
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.

Validating Automatic Service Migration

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 into the WebLogic Remote Console.
2. Navigate to the **Monitoring Tree**.
3. In the structure tree, expand **Environment** > **Migration**.
4. Click **Service Migration Data Runtimes**.

   The console displays a list of migratable targets and their current hosting server.
5. In the **Migratable Targets** table, select a row for one of the migratable targets.
6. Note the value in the **Migrated To** property.
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 WebLogic Remote Console and refresh the table of Service Migration Data Runtimes; verify that the migratable targets are transferred to the remaining, running Managed Server in the cluster:

   - Verify that the **Migrated to** value 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. Log into the **WebLogic Remote Console** and navigate to the **Monitoring Tree**.
>
> b. Navigate to **Dashboards** and click **JMS Destinations**.
>
> c. Order by Destination Name and look for the destination.
>
> > **Tip**
> >
> > You can copy this dashboard and create a custom one to filter by specific destination name.
>
> d. Review the **Messages Current Count** and **Messages Pending Count** values.

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, use WLST migrate command. For more information, see *WLST Command Reference for Oracle WebLogic Server*.