Oracle® Fusion Middleware

Enterprise Deployment Guide for Oracle WebCenter Portal

Release 14.1.2.0.0
G12338-01
August 2025

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

G12338-01

Primary Author: Oracle Corporation

Contents

11 Configuring Oracle HTTP Server for an Enterprise Deployment

12 Extending the Domain to Include Oracle WebCenter Content

13 Extending the Domain to Include Inbound Refinery

14 Extending the Domain with Oracle SOA Suite

15 Extending the Domain with Oracle WebCenter Portal

16 Integrating WebCenter Portal Workflows with Oracle SOA Suite in the Same Domain

Part IV Common Configuration and Management Procedures for an Enterprise Deployment

17 Common Configuration and Management Tasks for an Enterprise Deployment

18 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.

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 Portal 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:

- Use the instructions in one of the product-specific installation guides to install and configure a **standard installation topology** for a selected set of Oracle Fusion Middleware products.

  A standard installation topology can be installed on a single host for evaluation purposes, but it can also serve as a starting point for scaling out to a more complex production environment.

  For Oracle WebCenter Portal, see:

  – Installing and Configuring Oracle WebCenter Portal

  For Oracle WebCenter Content , see:

  – Installing and Configuring Oracle WebCenter Content

  For Oracle SOA Suite, see:

  – *Installing SOA Suite and Business Process Management Suite Quick Start for Developers*

- 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.

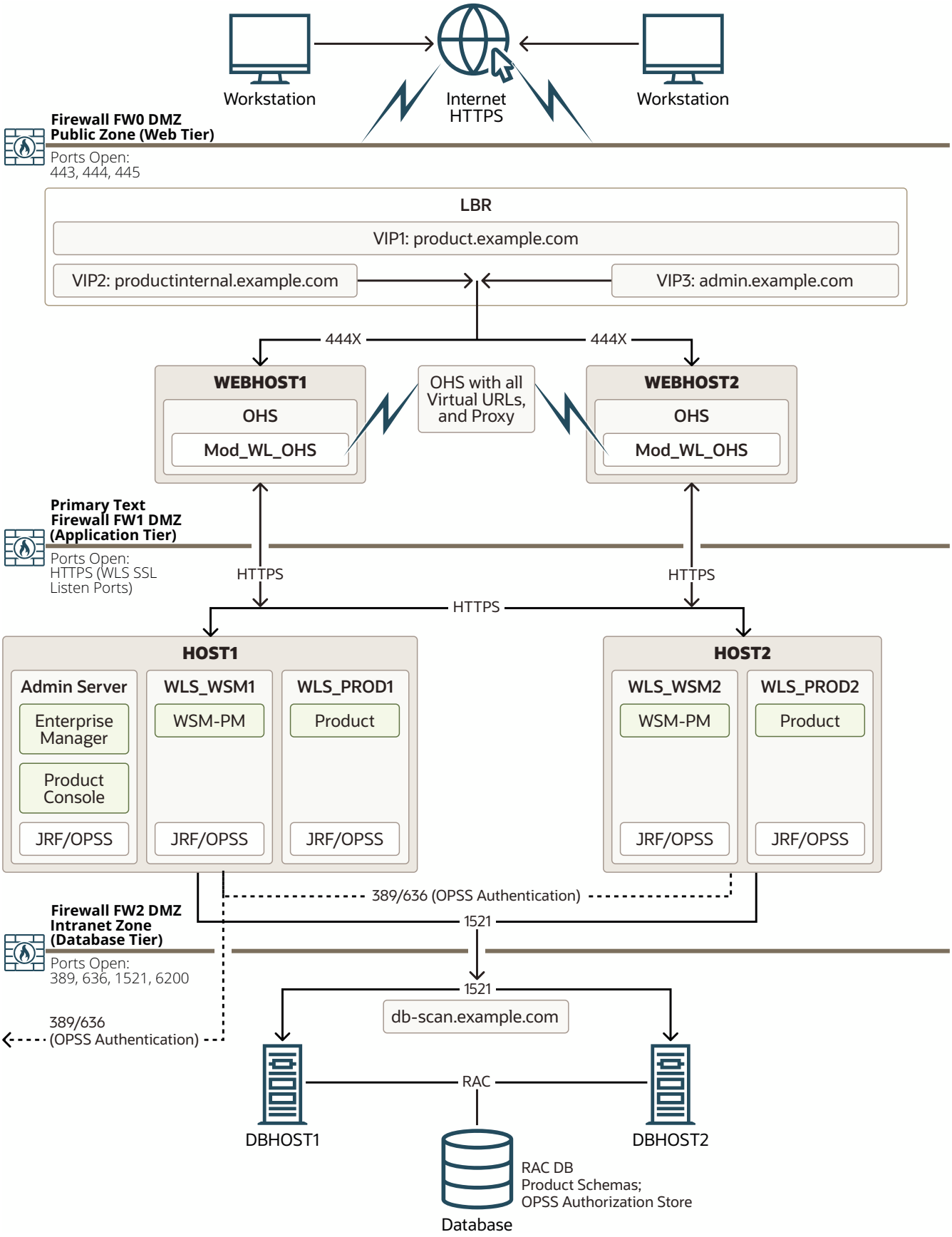
A best practices approach starts with the basic concept of a multi-tiered deployment and standard communications between the different software tiers.

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 in *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 DMZ public zone, which is a common requirement in security audits. If a load balancer routes directly to the WebLogic Server, requests move from the load balancer to the application tier in one single HTTP jump, which can cause security concerns.
- The Web tier allows the WebLogic Server cluster membership to be reconfigured (new servers added, others removed) without having to change the Web server configuration (as long as at least some of the servers in the configured list remain alive).
- Oracle HTTP Server delivers static content more efficiently and faster than WebLogic Server; it also provides FTP services, which are required for some enterprise deployments,

as well as the ability to create virtual hosts and proxies via the Oracle HTTP Server configuration files.

- Oracle HTTP Server provides HTTP redirection over and above what WebLogic Server provides. You can use Oracle HTTP Server as a front end against many different WebLogic Server clusters, and in some cases, control the routing via content based routing.
- Oracle HTTP Server provides the ability to integrate single sign-on capabilities into your enterprise deployment. For example, you can later implement single sign-on for the enterprise deployment, using Oracle Access Manager, which is part of the Oracle Identity and Access Management family of products.
- Oracle HTTP Server provides support for WebSocket connections deployed within WebLogic Server.

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

Alternatives to Using Oracle HTTP Server in the Web Tier

Although Oracle HTTP Server provides a variety of benefits in an enterprise topology, Oracle also supports routing requests directly from the hardware load balancer to the Managed Servers in the middle tier.

This approach provide the following advantages:

- Lower configuration and processing overhead than using a front-end Oracle HTTP Server Web tier front-end.
- Monitoring at the application level since the LBR can be configured to monitor specific URLS for each Managed Server (something that is not possible with OHS).

  Note that this enables routing to the Managed Servers only when all composites are deployed, and you must use the appropriate monitoring software.

Configuration of Oracle HTTP Server in the Web Tier

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

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

About Mod_WL_OHS

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

See What are Oracle WebLogic Server Proxy Plug-Ins? in *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.

Using Oracle Web Services Manager in the Application Tier

Oracle Web Services Manager (Oracle WSM) provides a policy framework to manage and secure web services in the Enterprise Deployment topology.

In most enterprise deployment topologies, the Oracle Web Services Manager Policy Manager runs on Managed Servers in a separate cluster, where it can be deployed in an active-active highly available configuration.

You can choose to target Oracle Web Services Manager and Fusion Middleware products or applications to the same cluster, as long as you are aware of the implications.

The main reasons for deploying Oracle Web Services Manager on its own managed servers is to improve performance and availability isolation. Oracle Web Services Manager often provides policies to custom web services or to other products and components in the domain. In such a case, you do not want the additional Oracle Web Services Manager activity to affect the performance of any applications that are sharing the same managed server or cluster as Oracle Web Services Manager.

The eventual process of scaling out or scaling up is also better addressed when the components are isolated. You can scale out or scale up only the Fusion Middleware application Managed Servers where your products are deployed or only the Managed Servers where Oracle Web Services Manager is deployed, without affecting the other product.

Best Practices and Variations on the Configuration of the Clusters and Hosts on the Application Tier

In a typical enterprise deployment, you configure the Managed Servers in a cluster on two or more hosts in the application tier. For specific Oracle Fusion Middleware products, the enterprise deployment reference topologies demonstrate best practices for the number of Managed Servers, the number of clusters, and what services are targeted to each cluster.

These best practices take into account typical performance, maintenance, and scale-out requirements for each product. The result is the grouping of Managed Servers into an appropriate set of clusters within the domain.

Variations of the enterprise deployment topology allow the targeting of specific products or components to additional clusters or hosts for improved performance and isolation.

For example, you can consider hosting the Administration Server on a separate and smaller host computer, which allows the FMW components and products to be isolated from the Administration Server.

These variations in the topology are supported, but the enterprise deployment reference topology uses the minimum hardware resources while keeping high availability, scalability and security in mind. You should perform the appropriate resource planning and sizing, based on the system requirements for each type of server and the load that the system needs to sustain. Based on these decisions, you must adapt the steps to install and configure these variations accordingly from the instructions presented in this guide.

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, the Oracle SOA Suite products can be configured for Automated Service Migration, rather than Whole Server Migration. This means the Managed Servers, as well as the Administration Server, can take advantage of Crash Recovery, so there is no need to apply different properties to the Administration Server and Managed Server domain directories.

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

About Using a Per Host Node Manager Configuration

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

A per host Node Manager configuration allows all Node Managers to use a listen address of ANY, so they listen on all addresses available on the host. This means that when the Administration Server fails over to a new host, no additional configuration is necessary.

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 Coherence Clusters In a Typical Enterprise Deployment

The standard Oracle Fusion Middleware enterprise deployment includes a Coherence cluster that contains storage-enabled Managed Coherence Servers. Oracle FMW products add their clusters as members to this default coherence cluster during domain creation or extension.

This configuration is a good starting point for using Coherence. Depending upon your specific requirements, you can consider tuning and reconfiguring Coherence to improve performance in a production environment

> **Note**
>
> Most Oracle Fusion Middleware products include Coherence GAR deployments. These deployments may have specific requirements pertaining to the default Coherence Cluster configuration (for example, local caches versus distributed). Consult the appropriate product installation and administration guides for specific limitations or processes regarding Coherence cluster configuration changes.

When reviewing port assignments, note that the Oracle Fusion Middleware products and components default to a Well Known Address (WKA) list that uses the port specified on the Coherence Clusters screen of the Configuration Wizard. The WKA list also uses the listen address of all servers that participate in the coherence cluster as the listen address for the WKA list. These settings can be customized by using the Oracle WebLogic Remote Console.

With respect to listen addresses, a Coherence cluster uses different services and protocols for network communications. The following are the out of the box services and their bind points:

- **Discovery Service** - Responsible for discovering other services including the cluster, defaults to a wildcard address, that is, listens on all addresses. It is configurable via operational configuration coherence/cluster-config/unicast-listener/discovery-address (generally left unset).
- **Clustering/TCMP** - Responsible for intra-cluster communication, defaults to whatever local address is routable to the WKA list, which are WCPHOST1 and WCPHOST2 ips in an

enterprise deployment topology. It is configurable via operational configuration coherence/cluster-config/unicast-listener/address (generally left unset).

- **Extend Proxy** - Responsible for communication with non-clustered clients, defaults to the discovery address. It is configurable via cache cache-config/caching-schemes/proxy-scheme/acceptor-config/tcp-acceptor/local-address (generally left unset).

For more information, refer to the following resources:

- For information about Coherence clusters, see Configuring and Managing Coherence Clusters in *Administering Clusters for Oracle WebLogic Server*.
- For information about tuning Coherence, see Performance Tuning in *Administering Oracle Coherence*.
- For information about storing HTTP session data in Coherence, see Using Coherence*Web with WebLogic Server in *Administering HTTP Session Management with Oracle Coherence*Web*.
- For more information about creating and deploying Coherence applications, see Creating Coherence Applications for WebLogic Server and Deploying Coherence Applications for WebLogic Server in *Developing Oracle Coherence Applications for Oracle WebLogic Server*.
- For information about the coherence listen addresses, see Element Reference and Configuring Caches in *Developing Applications with Oracle Coherence*.

About the Data Tier

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

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

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

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

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

3

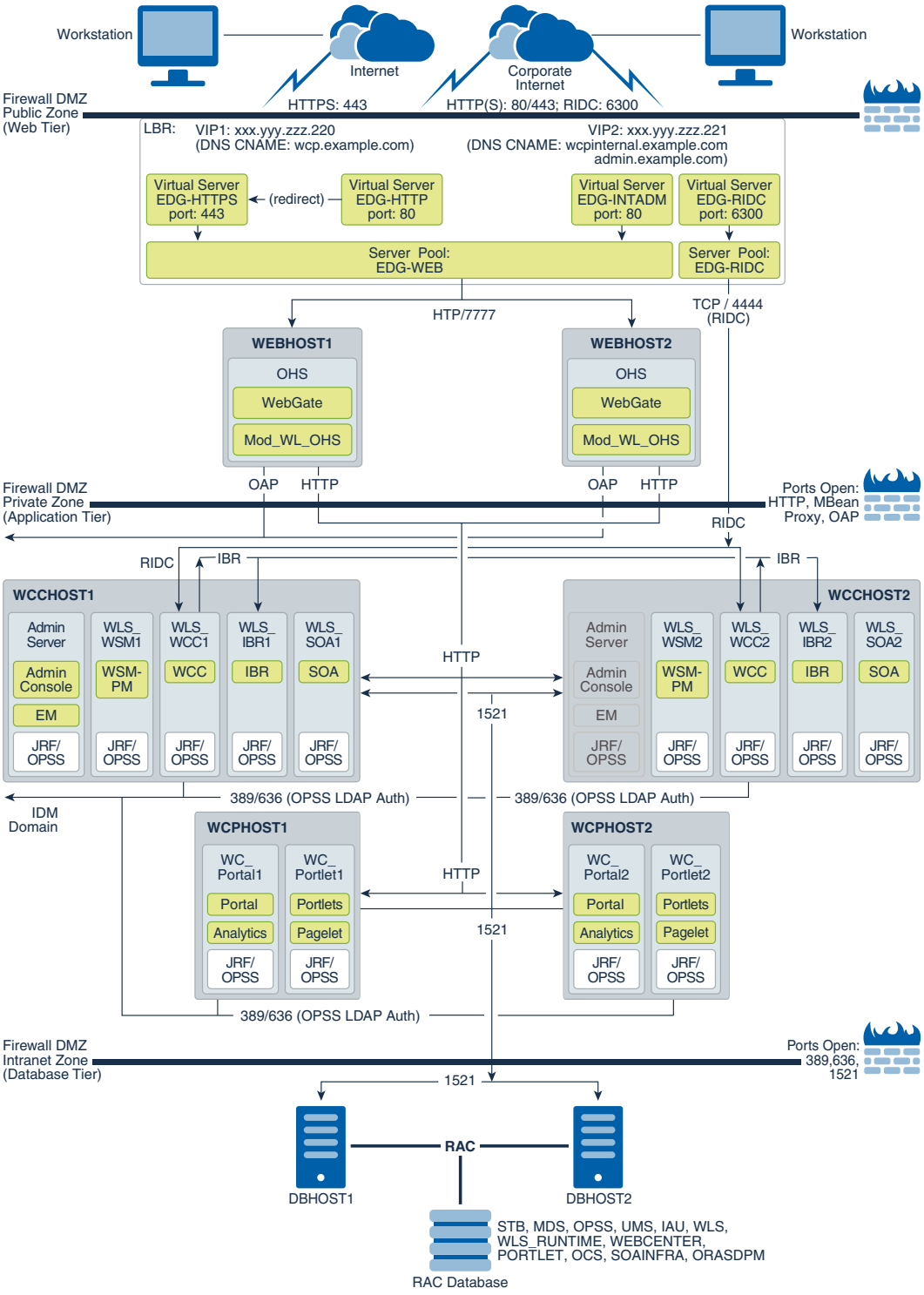
Understanding the WebCenter Portal Enterprise Deployment Topology

This chapter describes the WebCenter Portal deployment topologies. These topologies represent specific reference implementations of the concepts described in Understanding a Typical Enterprise Deployment.

This chapter provides information on primary WebCenter Portal topology diagrams.

Diagram of the WebCenter Portal Enterprise Deployment Topology

The following diagram shows the primary Oracle WebCenter Portal enterprise deployment topology, which is described in this guide.



Understanding the Primary WebCenter Portal Topology Diagrams

The Oracle WebCenter Portal topology follows a standard approach to an enterprise topology based on Oracle-recommended best practices. The standard elements of an Oracle Fusion Middleware enterprise topology are described in detail in Understanding a Typical Enterprise Deployment.

Before you review the information here, 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.

See the following sections for information about the elements that are unique to the topology described in this chapter:

Summary of the WebCenter Portal 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 Portal topologies:

- `wcp.example.com` — This virtual server name is used for all incoming end-user client traffic. It acts as the access point for all HTTP traffic to the run-time WebCenter Portal components. The load balancer listens for all requests to this virtual server name over SSL. Optionally, an additional virtual server configured on port 80 can be set up with a rule to automatically redirect all traffic to port 443. As a result, clients access this environment's services using the following secure address:

  ```
  https://wcp.example.com:443
  ```

- wcpinternal.example.com — This virtual server name is used for internal calls to Oracle WebCenter services. There are two virtual servers configured at the hardware load balancer. One (listening on port 80) is used for web-based services. The second (listening on port 6300) is for WebCenter Conent Remote Intradoc Client (RIDC) API calls.

  Incoming traffic from clients is not SSL-enabled. Clients accessing the HTTP services use the following address. These requests are forwarded to port 7777 on WEBHOST1 and WEBHOST2:

  ```
  http://wcpinternal.example.com:80/
  ```

  WebCenter Content RIDC-enabled applications, such as the WebCenter Content Desktop Integration Suite (DIS) and WebCenter Portal Document Services, use port 6300 for TCP connections to WebCenter Content Server's RIDC API endpoints. These requests are forwarded to port 4444 on WCCHOST1 and WCCHOST2:

  ```
  wcpinternal.example.com:6300
  ```

- `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. As a result, clients access this service by using the following secure address:

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

Use instructions later in this guide to perform the following tasks:

- 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 Portal 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 may include the clusters shown in Table 3-1. The servers in these clusters provide a single-site actively clustered high availability configuration.

> **Note**
>
> Beginning with 12*c* (12.2.1.3.0), Oracle WebCenter Portal has deprecated support for Jive features (announcements and discussions). If you are upgrading from a prior release, these features remain available only to support your existing portals that use announcements and discussions. Oracle recommends that you do not include these features in new portals. See Managing Announcements and Discussions in the *Administering Oracle WebCenter Portal* guide.

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

Product Component	Cluster	Managed Servers
Oracle Web Services Manager	WSM-PM_Cluster	WLS_WSM1, WLS_WSM2
Oracle WebCenter Portal	Portal_Cluster	WLS_Portal1, WLS_Portal2
Oracle WebCenter Portlet	Portlet_Cluster	WLS_Portlet1, WLS_Portlet2
Oracle WebCenter Content	WCC_Cluster	WLS_WCC1, WLS_WCC2
Oracle Inbound Refinery	IBR_Servers	WLS_IBR1, WLS_IBR2
Oracle SOA Suite	SOA_Cluster	WLS_SOA1, WLS_SOA2

Table 3-2 Upgrade-only Clusters for the Oracle WebCenter Portal Enterprise Deployment Topology

Product Component	Cluster	Managed Servers
Oracle WebCenter Collaboration	Collab_Cluster	WLS_Collaboration1, WLS_Collaboration2

Flow Charts and Roadmaps for Implementing the Primary WebCenter Portal Enterprise Topologies

The following topics summarize the high-level steps you must perform to install and configure the enterprise topology described in this chapter.

Flow Chart of the Steps to Install and Configure the WebCenter Portal Enterprise Topologies

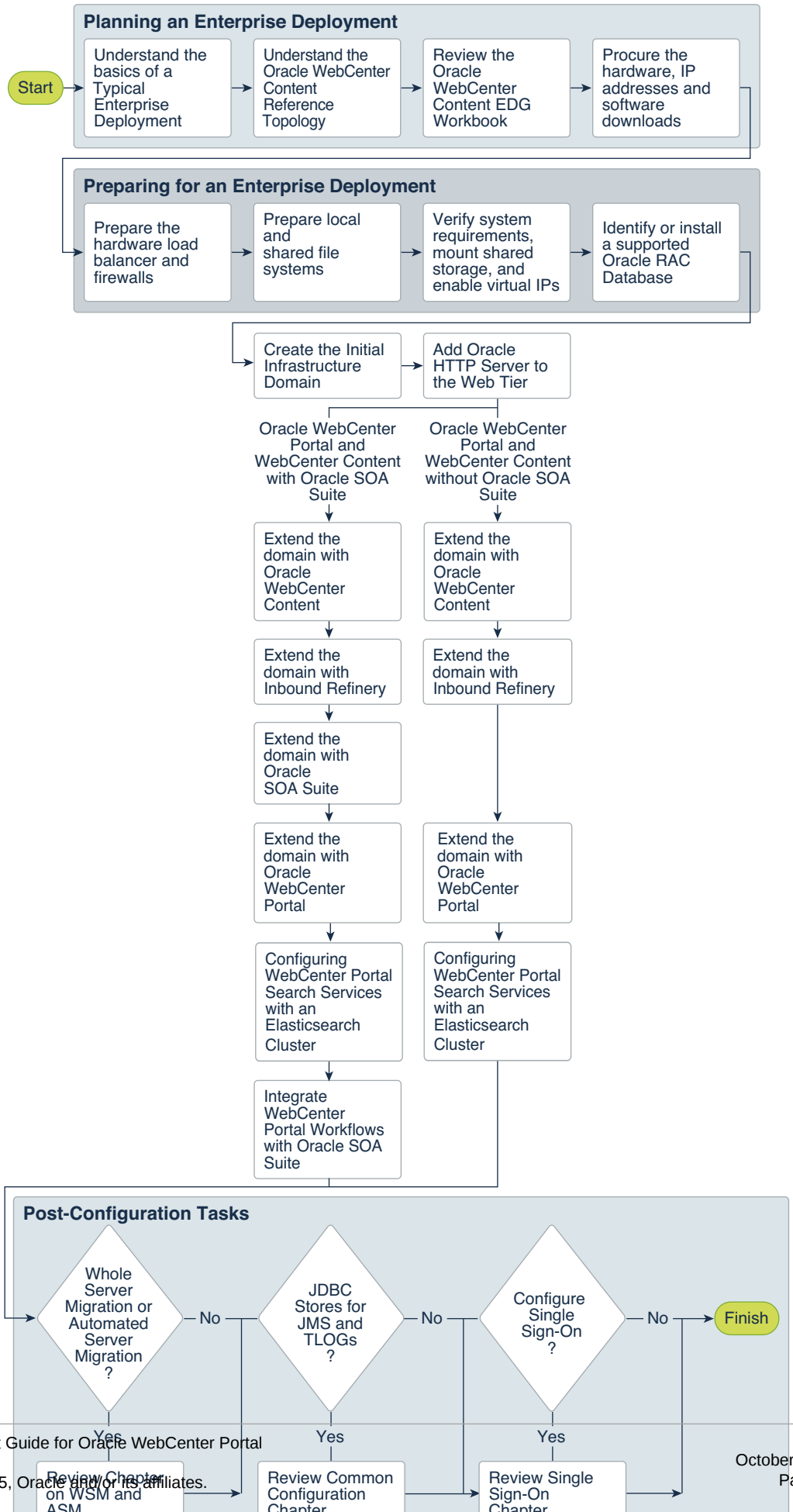
The following flow chart shows the steps required to install and configure the primary enterprise deployment topologies described in this chapter. The sections following the flow chart explain each step in the flow chart.

This guide is designed so you can start with a working WebCenter Portal 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 Portal 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.



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 Portal 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 Portal Topology

Roadmap Table for Configuring the WebCenter Portal Enterprise Topology describes each of the configuration steps required when configuring the topology shown in Diagram of the WebCenter Portal Enterprise Deployment Topology.

These steps correspond to the steps shown in the flow chart in Flow Chart of the Steps to Install and Configure the WebCenter Portal Enterprise Topologies.

Table 3-3 Roadmap Table for Configuring the Oracle WebCenter Portal Topology

Flow Chart Step	More Information
Create the initial Infrastructure domain	Creating the Initial Infrastructure Domain for an Enterprise Deployment

Table 3-3 (Cont.) Roadmap Table for Configuring the Oracle WebCenter Portal Topology

Flow Chart Step	More Information
Extend the domain to Include the Web Tier	Configuring the Web Tier for an Enterprise Deployment
Extend the domain with Oracle WebCenter Content	Extending the Domain to Include Oracle WebCenter Content
Extend the domain with Inbound Refinery	Extending the Domain to Include Inbound Refinery
Extend the domain with Oracle SOA Suite	Extending the Domain with Oracle SOA Suite
Extend the domain with Oracle WebCenter Portal	Extending the Domain with Oracle WebCenter Portal
Integrating WebCenter Portal with Oracle SOA Suite in the Same Domain	Integrating WebCenter Portal Workflows with Oracle SOA Suite in the Same Domain

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 Portal Enterprise Deployment Workbook

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

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

Locating the Oracle WebCenter Portal Enterprise Deployment Workbook

The Oracle WebCenter Portal 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 Portal Enterprise Deployment Workbook

The following sections describe the contents of the Oracle WebCenter Portal 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.

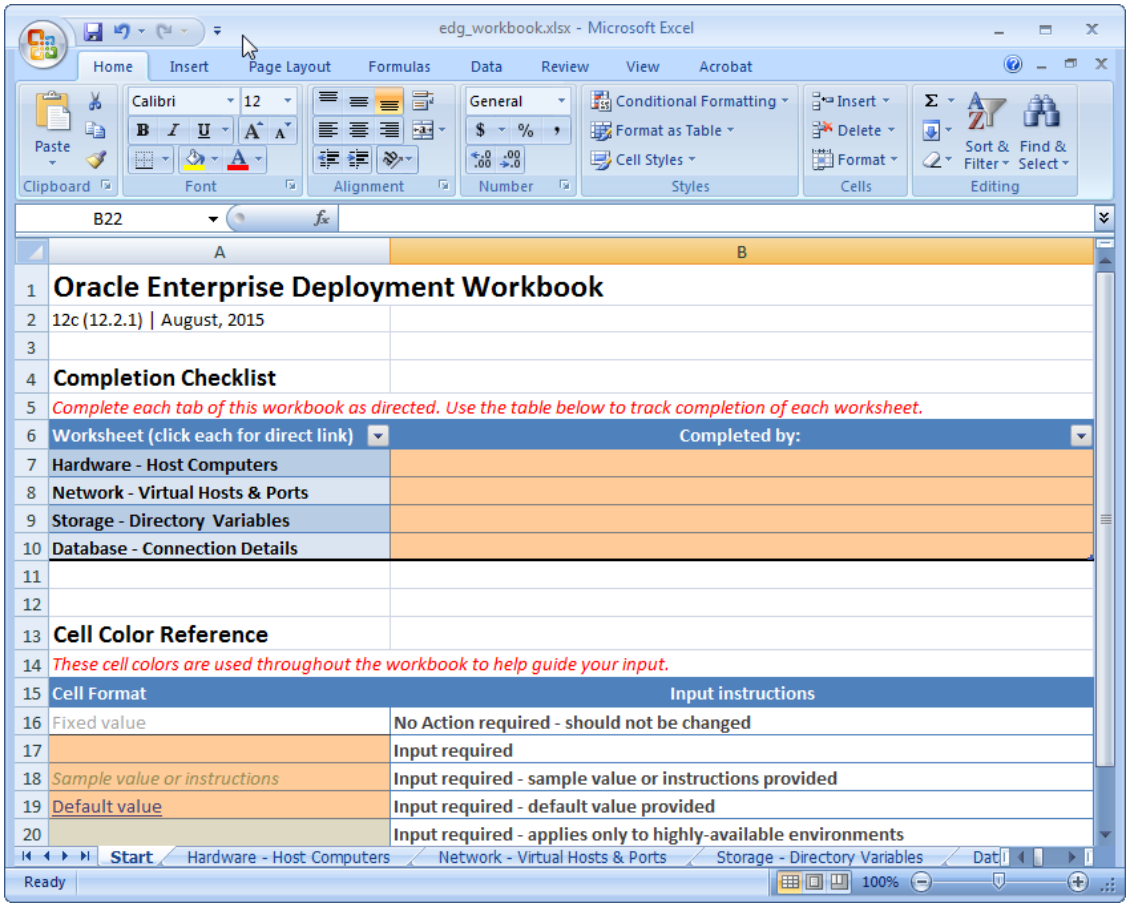
Using the Start Tab

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 Portal 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 WCPHOST1, you can then replace the WCPHOST1 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 WCPHOST1 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 Portal 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 HTTP and HTTPS traffic.
  - The virtual server names must be associated with IP addresses and be part of your DNS. Clients must be able to access the external load balancer through the virtual server names.
- Ability to detect node failures and immediately stop routing traffic to the failed node.
- It is highly recommended that you configure the load balancer to be in fault-tolerant mode.
- It is highly recommended that you configure the load balancer virtual server to return immediately to the calling client when the backend services to which it forwards traffic are

unavailable. This is preferred over the client disconnecting on its own after a timeout based on the TCP/IP settings on the client machine.

- Ability to maintain sticky connections to components. Examples of this include cookie-based persistence, IP-based persistence, and so on.
- In this Enterprise Deployment Guide, SSL listeners are used for the Oracle HTTP Servers and the Oracle WebLogic Servers. The load balancer should hence be able to establish SSL communication with the back-end servers in its pools.
- SSL acceleration (this feature is recommended, but not required for the enterprise topology).
- The ability to route TCP/IP requests; this is a requirement for Oracle SOA Suite for healthcare integration, which uses the Minimum Lower Layer Protocol (MLLP) over TCP.

Host Computer Hardware Requirements

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

It includes the following topics.

General Considerations for Enterprise Deployment Host Computers

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

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 Portal 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 that are 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 Portal 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 WCPHOST1, as depicted in the reference topologies.

When you procure systems, use the information in the **Approximate Top Memory** column as a guide when determining the minimum physical memory that 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 that 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_WSM	3.0 GB	2000	130
WLS_Portal	4.0 GB	3100	240
WLS_Portlet	4.0 GB	2200	180
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	18.5 GB*	13130	770

* 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 14*c* (14.1.2.0.0) products, including the Oracle WebCenter Portal 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 Portal 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
WCPHOST*n*	10 GB*
WCCHOST*n*	20 GB*

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

Operating System Requirements for an Enterprise Deployment Topology

This section provides details about the operating system requirements.

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

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

Reserving the Required IP Addresses for an Enterprise Deployment

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

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

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

For Fusion Middleware 12*c* products that support Automatic Service Migration, VIPs for the Managed Servers are typically not necessary.

- A unique virtual host name to be mapped to each VIP.

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

For more information, see the following topics.

What is a Virtual IP (VIP) Address?

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

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 Portal 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 certification matrixes. If there is a conflict of information between this guide and the certification matrixes, then the information in the certification matrixes must be considered the correct version, as they are frequently updated.

Distribution	Installer File Name	Description
Oracle Fusion Middleware 14*c* (14.1.2.0.0) Infrastructure	`fmw_14.1.2.0.0_infrastructure.jar`	Download this distribution to install the Oracle Fusion Middleware Infrastructure, which includes Oracle WebLogic Server and Java Required Files software required for Oracle Fusion Middleware products. This distribution also installs the Repository Creation Utility (RCU), which in previous Oracle Fusion Middleware releases was packaged in its own distribution.
Oracle HTTP Server 14*c* (14.1.2.0.0)	`fmw_14.1.2.0.0_ohs_linux64.bin`	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 Portal	`fmw_14.1.2.0.0_wcportal.jar`	Download this distribution to install the Oracle WebCenter Portal software.
Oracle Fusion Middleware 14*c* (14.1.2.0.0) WebCenter Content	`fmw_14.1.2.0.0_wccontent.jar`	Download this distribution if you plan to add Oracle WebCenter Content to the WebCenter Portal topology.
Oracle Fusion Middleware 14*c* (14.1.2.0.0) SOA Suite and Business Process Management	`fmw_14.1.2.0.0_soa.jar`	Download this distribution if you plan to add Oracle SOA Suite to the WebCenter Portal 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.

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 Portal, 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 admin 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 backend's certificate (the SSL certificate used by the OHS listeners in the backend 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. Since 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 Portal enterprise topology:

Virtual Host	Server Pool	Protocol	SSL Termination?	External?
`admin.example.com:445`	`WEBHOST1.example.com:4445` `WEBHOST2.example.com 4445`	HTTPS	No	No
`wcp.example.com:443`	`WEBHOST1.example.com:4443` `WEBHOST2.example.com 4443`	HTTPS	Yes	Yes
`wcpinternal.example.com:444`	`WEBHOST1.example.com:4444` `WEBHOST2.example.com 4444`	HTTPS	No	No

Additional Instructions for admin.example.com

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

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

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

Additional Instructions for wcp.example.com

This section provides additional instructions for configuring the virtual server—wcp.example.com.

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.
- 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 the Oracle WebLogic Remote Console and to the Oracle Enterprise Manager Fusion Middleware Control and should be used only when accessing the system from `admin.example.com`.

Configuring the Firewalls and Ports for an Enterprise Deployment

As an administrator, it is important that you become familiar with the port numbers that are 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:

> **Note**
>
> The ports and port ranges specified for application access at firewall `FW1` in the following table apply to the static cluster use-case. Ports and port ranges will vary for dynamic clusters based on scalability needs.

Firewall notation:

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

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

Type	Firewall	Port and Port Range	Protocol / Application	Inbound / Outbound	Other Considerations and Timeout Guidelines
Browser request	FW0	80	HTTP / Load Balancer **Note** You need this option only if redirection from port 8	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
			0 to port 443 is used.		
Browser request	FW0	44x	HTTPS / Load Balancer	Inbound	Timeout depends on the size and type of HTML content.
Browser request	FW1	80	HTTP / Load Balancer	Outbound (for intranet clients)	Timeout depends on the size and type of HTML content.
Browser request	FW1	44x	HTTPS / Load Balancer	Outbound (for intranet clients)	Timeout depends on the size and type of HTML content.
Callbacks and Outbound invocations	FW1	80	HTTP / Load Balancer	Outbound	Timeout depends on the size and type of HTML content.
Callbacks and Outbound invocations	FW1	44x	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.

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
WebLogic Remote Console and Enterprise Manager Console	FW1	9002	HTTPS / Remote Console and Enterprise Manager t3s	Both	You should tune this timeout based on the type of access to the admin console (whether you plan to use the Oracle WebLogic Remote Console from the application tier clients or clients external to the application tier).
Database access	FW2	1521	SQL*Net	Both	Timeout depends on database content and on the type of process model used for WebCenter Portal.
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	n/a
Oracle Unified Directory access	FW2	389 636 (SSL)	LDAP or LDAP/ssl	Inbound	You should tune the directory server's parameters based on load balancer, and not the other way around.
Oracle Notification Server (ONS)	FW2	6200	ONS	Both	Required for Gridlink. An ONS server runs on each database server.

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

Type	Firewall	Port and Port Range	Protocol / Application	Inbound / Outbound	Other Considerations and Timeout Guidelines
WSM-PM access	FW1	7010 Range: 7010 - 7999	HTTPS / WLS_WSM-PM*n*	Inbound	Set the timeout to 60 seconds.
Portal Server access	FW1	9001	HTTP / WLS_Portal*n*	Inbound	Set the timeout to a short period (5–10 seconds).
RIDC API requests	FW1	6300	TCP/WLS_WCC*n*	Inbound	n/a
SOA Server access	FW1*	8001 Range: 8000 - 8010	HTTP / WLS_SOA*n*	Inbound	Timeout varies based on the type of process model used for SOA.

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.
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:

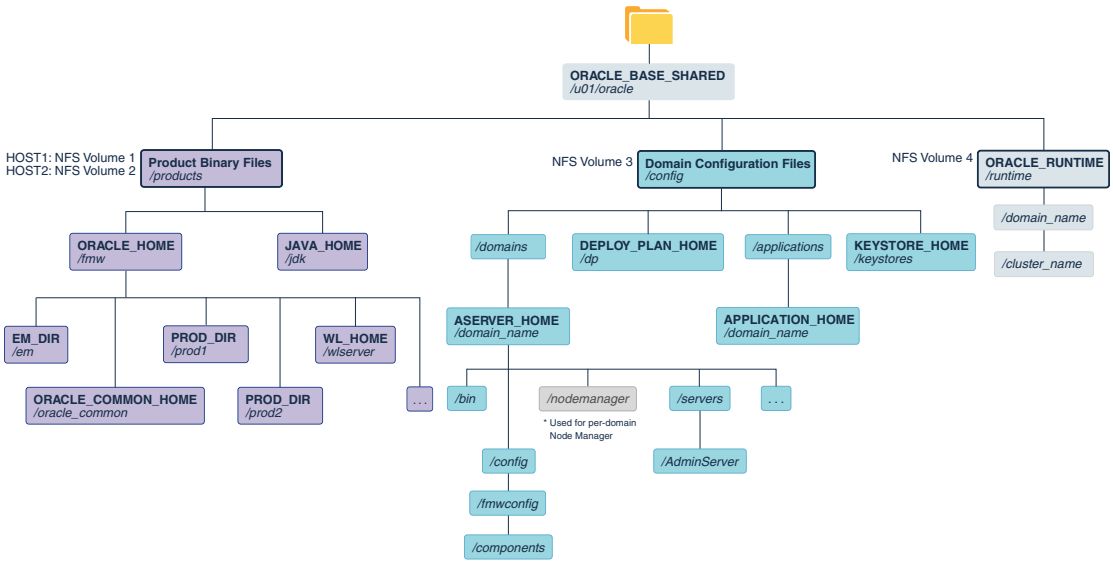
- Figure 7-1 shows the resulting directory structure on the shared storage device after you have installed and configured a typical Oracle Fusion Middleware enterprise deployment. The shared storage directories are accessible by the application tier host computers.
- Figure 7-2 shows the resulting directory 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.

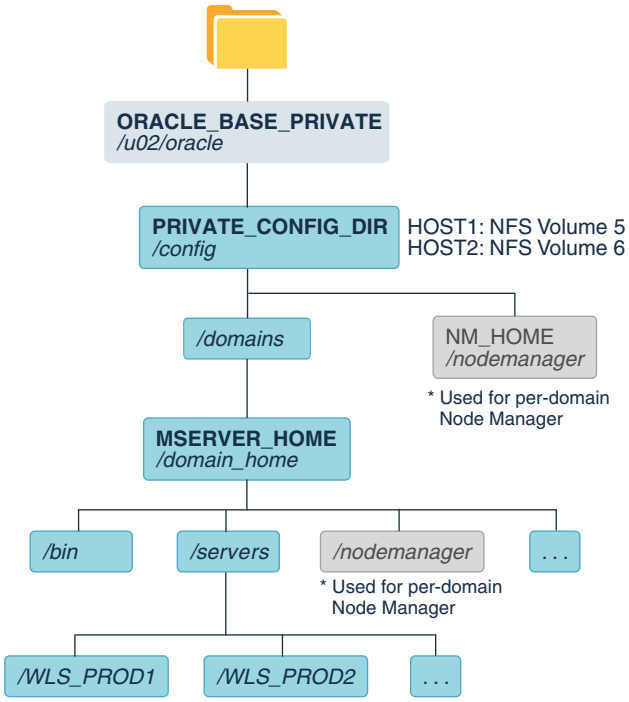
Where applicable, the diagrams also include the standard variables used to reference the directory locations in the installation and configuration procedures in this guide.

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



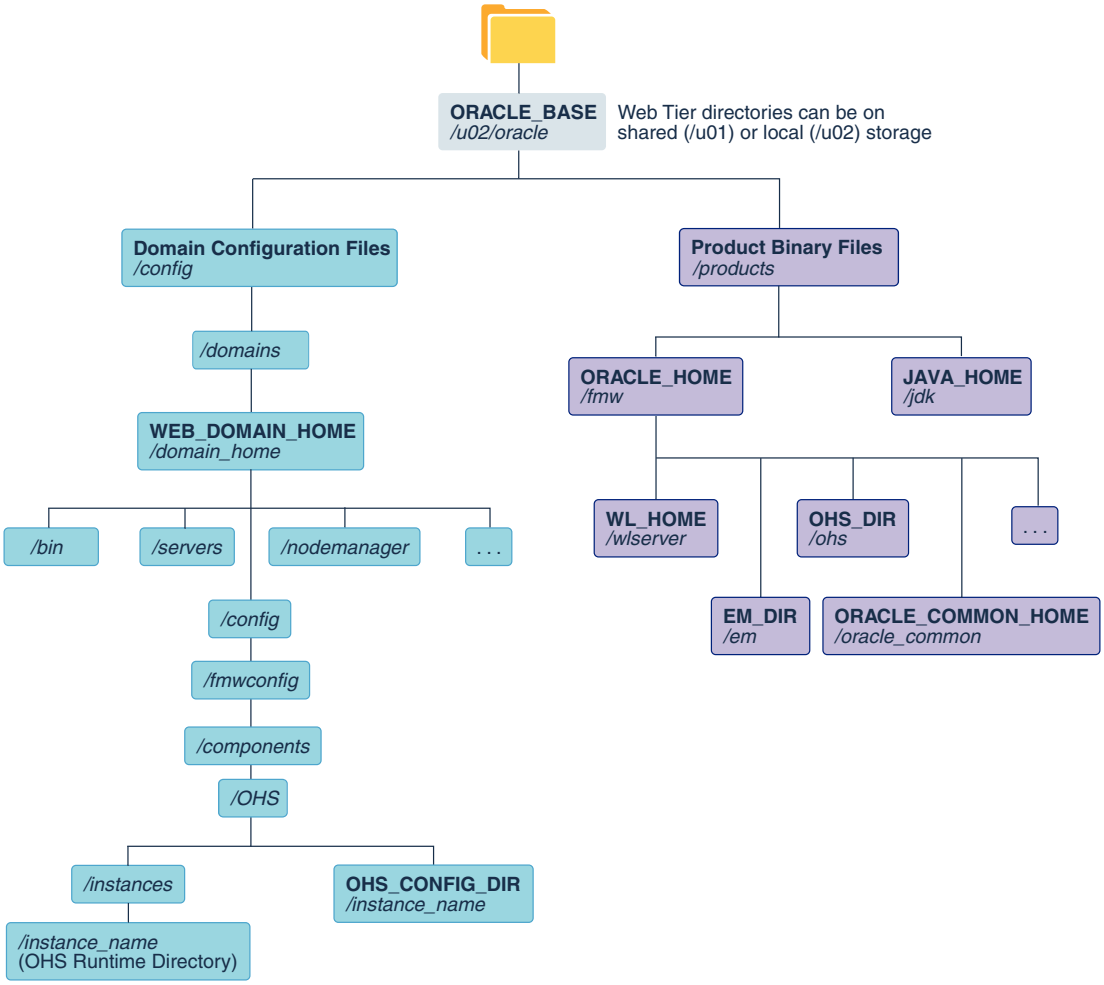
* See About the Node Manager Configuration in a Typical Enterprise Deployment.

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`
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`

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
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

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

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

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 WCPHOST2. Be sure the directory path to the binaries on WCPHOST2 is identical to the directory path on WCPHOST1.

  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 (WCPHOST1 and WCPHOST2), 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	WCPHOST1 WCCHOST1	`/u01/oracle/products/`	Storage for the product binaries to be used by WCPHOST1; this is where the Oracle home directory and product directories are installed. Used initially by WCPHOST1, but can be shared with other hosts when scaling-out the topology.
NFS Volume 2	WCPHOST2 WCCHOST2	`/u01/oracle/products/`	Storage for the product binaries to be used by WCPHOST2; this is where the Oracle home directory and product directories are installed. Used initially by WCPHOST2, but can be shared with other hosts when scaling-out the topology.

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 3	WCCHOST1 WCCHOST2	`/u01/oracle/config/`	Administration Server domain configuration, mounted to all hosts; used initially by WCPHOST1, but can be failed over to any host.
NFS Volume 4	WCPHOST1 WCPHOST2 WCCHOST1 WCCHOST2	`/u01/oracle/runtime/`	The runtime directory mounted to all hosts contains application runtime artifacts or any other files that need to be shared by all the members of the cluster and that are generated during the execution of an application or integration flow.
NFS Volume 5	WCPHOST1	`/u02/oracle/config/`	Local storage for the Managed Server domain directory to be used by WCPHOST1, if the private Managed Server domain directory resides on shared storage.
NFS Volume 6	WCPHOST2	`/u02/oracle/config/`	Local storage for the Managed Server domain directory to be used by WCPHOST2, if the private Managed Server domain directory resides on shared storage.
NFS Volume 7	WEBHOST1	`/u02/oracle/`	Local storage for the Oracle HTTP Server 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 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.
NFS Volume 9	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.

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 10	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.

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

4. Re-login into the host computer.
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)"
   ```

   Execute 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
```

Oracle also recommends that you use aliases to map to different IPs in different data centers in preparation for disaster recovery. You can also use these aliases to configure the listen address for some of the components.

In this guide, the abstract hostnames that are provided on the **Hardware - Host Computers** tab of the workbook (WCCHOST*n*, WCPHOST*n*, WEBHOST*n*, and ADMINVHN) are used for these aliases, so the `/etc/hosts` can be similar to this example:

```
# EDG VIPs for Application-Tier Hosts
10.229.188.204 host1-vip.example.com host1-vip ADMINVHN
# EDG Application-Tier Hosts
10.229.188.205 host1.example.com host1 WCCHOST1
10.229.188.206 host2.example.com host2 WCCHOST2
10.229.188.207 host3.example.com host3 WCPHOST1
10.229.188.208 host4.example.com host4 WCPHOST2
# EDG DMZ Web-Tier Hosts
10.229.150.121 host5.example.com host5 WEBHOST1
10.229.150.122 host6.example.com host6 WEBHOST2
```

Setting the DNS Settings

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

Configuring Operating System Users and Groups

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

Groups

You must create the following groups on each node.

- `oinstall`
- `dba`

Users

You must create the following user on each node.

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

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

Enabling Unicode Support

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

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

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

If the operating system is configured to use a non-UTF-8 encoding, Oracle WebCenter Portal 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 WCPHOST1 and WCPHOST2 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 WCPHOST1, 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 WCPHOST2 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 WCPHOST1. 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

To prepare for the enterprise deployment, you must enable the virtual IP (VIP) address required by the WebLogic Administration Server in the node where it will run by default.

See Reserving the Required IP Addresses for an Enterprise Deployment.

This guide recommends using Service Migration instead of Server Migration for high availability of services across the members of a WLS cluster. A Virtual IP is required only for the Administration Server's listen address so that it can be failed over manually to a different node in a loss of host scenario. It is assumed that this VIP and its mapping virtual host name have been provisioned and enabled by your network administrator so that the Administration Server can use it as a valid listen address.

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
```

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 server(s) you wish to use. For security reasons, ensure that these are inside your organization.
2. Log in to 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
```

Configuring a Host to Use an NIS/YP Host

If you are using NFS Version 4, configure a directory service or an NIS (Network Information Server). If your organization does not have one already, use the built-in one on the ZFS

storage appliance. See Configuring NFS Version 4 (NFSv4) on Exalogic in the *Oracle Fusion Middleware Exalogic Machine Owner's Guide* for more information.

Once you have configured your NIS host, configure each compute node to use it. Before beginning, determine the host names of the NIS servers you are going to use.

1. Login to the host as root.
2. Edit the `/etc/idmapd.conf` configuration file:

   ```
   vi /etc/idmapd.conf
   ```

   Set the domain value, as in the following example:

   ```
   Domain = example.com
   ```

3. Restart the `rpcidmapd` service:

   ```
   service rpcidmapd restart
   ```

4. Update the `/etc/yp.conf` configuration file, and set the correct domain value, as in the following example:

   ```
   vi /etc/yp.conf
   ```

   Add the following line:

   ```
   domain example.com server NIS_Server_hostname_or_IP
   ```

   Where `example.com` is the example domain and *NIS_Server_hostname_or_IP* is the host name or IP address of the NIS host. You must replace these sample values with values appropriate for your environment.

5. Set NIS domain name on the command line:

   ```
   domainname NIS_DOMAIN_NAME
   ```

   For example:

   ```
   domainname nisdomain.example.com
   ```

6. Edit the `/etc/nsswitch.conf` configuration file:

   ```
   vi /etc/nsswitch.conf
   ```

   Add `nis` to each of the following entries:

   > **Note**
   >
   > The first value may be `compat` or `files` depending on your OS and enterprise requirements.

   ```
   passwd:     files nis
   shadow:     files nis
   group:      files nis
   automount:  files nis nisplus
   aliases:    files nis nisplus
   ```

7. Restart the `rpcidmapd` service:

   ```
   service rpcidmapd restart
   ```

8. Restart the `ypbind` service by running the following command:

```
service ypbind restart
```

9. Check the `yp` service by running this command:

   ```
   ypwhich
   ```

10. Verify if you can access Oracle user accounts:

    ```
    ypcat passwd
    ```

11. Add `ypbind` to your boot sequence, so that it starts automatically after rebooting.

    ```
    chkconfig ypbind on
    ```

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 and Oracle Oracle Additional Considerations for the Restricted Use Licenses Included in Oracle WebCenter Portal in *Oracle Fusion Middleware Licensing Information*.

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*.

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**
>
> Creating a service for each specific product in the environment provides significant flexibility for tuning. For the WebCenter Enterprise Deployment Guide, it is usually sufficient to create a single DB service for each domain. Consult your Database Administrators for any local practices in this regard.
>
> 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 wcpdb -service wcpedg.example.com -preferred wcpdb1,wcpdb2
   ```

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

2. Start the service:

   ```
   srvctl start service-db wcpdb -service wcpedg.example.com
   ```

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

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

Use the following resources in the Oracle Database 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 wcpdb -service wcpedg.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 wcpdb -service wcpedg.example.com -rlbgoal SERVICE_TIME -
clbgoal SHORT
```

4. Restart the service:

```
srvctl stop service -db wcpdb -service wcpedg.example.com

srvctl start service -db wcpdb -service wcpedg.example.com
```

5. Verify the change in the configuration:

```
srvctl config service -db wcpdb -service wcpedg.example.com
Runtime Load Balancing Goal: SERVICE_TIME
  Service name: wcpedg.example.com
  Service is enabled
  Server pool: wcpdb_wcpedg.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 12*c*. Allows SecureFile LOB storage when the SECUREFILE keyword is used. The default storage method is BasicFile.

- `PREFERRED`: The default setting for 12*c* 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.

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

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

Part III contains the following chapters:

10

Creating the Initial Infrastructure Domain for an Enterprise Deployment

The following topics describe how to install and configure an initial domain, which can be used as the starting point for an enterprise deployment. Later chapters in this guide describe how to extend this initial domain with the various products and components that comprise the enterprise topology that you are deploying.

Variables Used When Creating the Infrastructure Domain

As you perform the tasks in this chapter, you reference the directory variables that are listed in this section.

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

- ORACLE_HOME
- ASERVER_HOME
- MSERVER_HOME
- APPLICATION_HOME
- JAVA_HOME
- NM_HOME
- KEYSTORE_HOME

In addition, you reference the following virtual IP (VIP) addresses and host names that are defined in Physical and Virtual IP Addresses Required by the Enterprise Topology:

- ADMINVHN
- WCCHOST1
- WCCHOST2
- WCPHOST1
- WCPHOST2
- DBHOST1
- DBHOST2
- SCAN Address for the Oracle RAC Database (`DB-SCAN.example.com`)

About the Initial Infrastructure Domain

Before you create the initial Infrastructure domain, be sure to 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 Domain

The following table lists some of the key characteristics of the domain that you are about to create. Reviewing these characteristics helps you to understand the purpose and context of the procedures that are 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
Uses a separate virtual IP (VIP) address for the Administration Server.	Configuration of the Administration Server and Managed Servers Domain Directories
Uses separate domain directories for the Administration Server and the Managed Servers in the domain.	Configuration of the Administration Server and Managed Servers Domain Directories
Includes a dedicated cluster for Oracle Web Services Manager	Using Oracle Web Services Manager in the Application Tier
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 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 /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
   ```

2. Navigate to the following directory on WCCHOST1:

   ```
   cd $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.

> **Note**
>
> Custom prefixes must be 10 characters or less when including WebCenter Portal schemas even though the RCU limit is 12 characters. This is to avoid RCU errors when validating the full WebCenter Portal schemas names. Maximum schema user name length is limited to 30 characters total. WebCenter Portal schema suffixes use up to 20 characters.

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.

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.


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

The following topics provide instructions for creating a WebLogic Server domain using the Fusion Middleware Configuration wizard.

For more information on the other methods that are available for creating a domain, 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 the following sections to create and configure the domain for the topology.

- Task 20, Assigning Servers to Machines
- Task 21, Reviewing Your Configuration Specifications and Configuring the Domain
- Task 22, Writing Down Your Domain Home and Administration Server URL

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

> **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
On the Database Configuration Type screen:

- 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 have to manually fill in the parameters for your schema on the JDBC Component Schema screen.

After you select **RCU Data**, fill in the fields as shown in the following table:

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: `wcpedg.example.com` Specify the service name based on the value configured earlier in the Preparing the Database for an Enterprise Deployment section.
Port	Enter the port number on which the database listens. For example, 1521.

Field	Description
Schema Owner Schema Password	Enter the user name and password to connect 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 finish specifying the database connection information. The following output in the Connection Result Log indicates that the operation is successful.

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

Successfully Done.
```

Click **Next** 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 following table.

Element	Description and Recommended Value
SCAN, Host Name, and Port	Select the **SCAN** check box. In the **Host Name** field, enter the Single Client Access Name (SCAN) Address for the Oracle RAC database. In the **Port** field, enter the SCAN listening port for the database (for example, 1521).
ONS Host and Port	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 the database can receive and process FAN events.
Service Name	Verify that the service name for the Oracle RAC database is appropriate. For example, `wcpedg.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 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, and 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 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.

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.

> **Warning**
>
> 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 Machines in *Administering Node Manager for Oracle WebLogic Server*.

Task 13 Configuring Managed Servers
Use the Managed Servers screen to create two new Managed Servers:

1. Click the **Add** button to create a new Managed Server.
2. Specify `WLS_WSM1` in the **Server name** column.
3. In the **SSL Listen Address** column, enter `WCCHOST1`. Ensure you enter the host name that corresponds to WCCHOST1 and do not use the IP address.
4. Ensure that **Enable Listen** is unchecked, and Listen Port is "Disabled" (grayed out).
5. Ensure that **Enable SSL Port** is checked for all servers.
6. Set **SSL Listen Port** to `7010`.
7. Set **Administration Port** to `9003`.
8. In the **Server Groups** drop-down list, select **JRF-MAN-SVR** and **WSMPM-MAN-SVR**.

   These server groups ensure that the Oracle JRF and Oracle Web Services Manager (OWSM) services are targeted to the Managed Servers that you are creating.

Server groups target Fusion Middleware applications and services to one or more servers by mapping defined groups of application services to each defined server group. Any application services that are mapped to a given server group are automatically targeted to all servers that are assigned to that group. See Application Service Groups, Server Groups, and Application Service Mappings in *Domain Template Reference*.

> **Note**
>
> Nonce caching for Oracle Web Services is initialized automatically by the WSM-CACHE-SVR server group and is suitable for most custom applications. This initialization is automatically performed in SOA, OSB, and other FMW servers that run JRF and create a coherence cluster. Nonce is a unique number that can be used only once in a SOAP request and is used to prevent replay attacks. Nonce caching naturally scales with the number of added Managed Servers that run Web service applications.
>
> For information about advanced caching configurations, see Caching the Nonce with Oracle Coherence in *Securing Web Services and Managing Policies with Oracle Web Services Manager*, which provides additional guidance for the use of nonce caching and the WSM-CACHE-SVR server-group in custom WLS servers.

9. Repeat this process to create a second Managed Server named `WLS_WSM2`.

Server Name	Listen Address	Enable Listen Port	Listen Port	Enable SSL Port	SSL Listen Port	Administration Port	Server Groups
`WLS_WSM1`	`WCCHOST1`	Unchecked	Disabled	Checked	`7010`	`9003`	`JRF-MAN-SVR` and `WSMPM-MAN-SVR`
`WLS_WSM2`	`WCCHOST2`	Unchecked	Disabled	Checked	`7010`	`9003`	`JRF-MAN-SVR` and `WSMPM-MAN-SVR`

The Managed Server names suggested in this procedure (WLS_WSM1 and WLS_WSM2) are referenced throughout this document; if you choose different names then be sure to replace them as needed.

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

Task 14 Configuring a Cluster

Use the Clusters screen to create a new cluster:

1. Click the **Add** button.

2. Specify `WSM-PM_Cluster` in the **Cluster Name** field.
3. From the **Dynamic Server Groups** drop-down list, select `Unspecified`.
4. Click **Next**.

> **Tips**
>
> For more information about the options on this screen, see Clusters in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 15 Assigning Server Templates
Click **Next**.

Task 16 Configuring Dynamic Servers
Verify that all dynamic server options are disabled for clusters that are to remain as static clusters.

1. Confirm that the **Calculated Listen Port** and **Calculated Machine Names** checkboxes on this screen are unchecked.
2. Confirm that the **Server Template** selection and **Dynamic Server Groups** are Unspecified.
3. Click **Next**.

Task 17 Assigning Managed Servers to the Cluster
Use the Assign Servers to Clusters screen to assign `WLS_WSM1` and `WLS_WSM2` to the new cluster `WSM-PM_Cluster`:

1. In the **Clusters** pane, select the cluster to which you want to assign the servers; in this case, `WSM-PM_Cluster`.
2. In the **Servers** pane, assign `WLS_WSM1` to `WSM-PM_Cluster` by doing one of the following:
   - Click once on `WLS_WSM1` to select it, and then click on the right arrow to move it beneath the selected cluster (`WSM-PM_Cluster`) in the Clusters pane.

     or
   - Double-click on `WLS_WSM1` to move it beneath the selected cluster (`WSM-PM_Cluster`) in the clusters pane.
3. Repeat these steps to assign the WLS_WSM2 Managed Server to the WSM-PM_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 18 Configuring Coherence Clusters
Use the Coherence Clusters screen to configure the Coherence cluster that is automatically added to the domain.
In the **Cluster Listen Port**, enter `9991`.

> **Note**
>
> For Coherence licensing information, see Oracle Coherence Products in *Oracle Fusion Middleware Licensing Information User Manual*.

Task 19 Creating 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. Select the **Unix Machine** tab.
2. Click the **Add** button to create new UNIX machines.

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

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

Name	Node Manager Listen Address	Node Manager Listen Port	NodeManagerType
ADMINHOST	Enter the value of the ADMINVHN variable.	5556	SSL
WCCHOST1	The value of the WCCHOST1 host name variable or WCCHOST1 alias. For example, `WCCHOST1.example.com`.	5556	SSL
WCCHOST2	The value of the WCCHOST2 host name variable or WCCHOST2 alias. For example, `WCCHOST2.example.com`.	5556	SSL
WCPHOST1	The value of the WCPHOST1 host name variable. For example, `WCPHOST1.example.com`.	5556	SSL
WCPHOST2	The value of the WCPHOST2 host name variable. For example, `WCPHOST2.example.com`.	5556	SSL

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

Task 20 Assigning Servers to Machines

Use the Assign Servers to Machines screen to assign any statically defined managed servers to the appropriate machines. Servers that are part of a dynamic cluster are assigned to the calculated machine names automatically.

The Assign Servers to Machines screen is similar to the Assign Managed Servers to Clusters screen. Select the target machine in the Machines column, select the server name in the left column, and click the right arrow to assign the server to the appropriate machine.

Assign the servers as follows:

- Assign the AdminServer to the ADMINHOST machine.
- Assign the WLS-WSM1 Managed Server to the WCCHOST1 machine.
- Assign the WLS-WSM2 Managed Server to the 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 21 Reviewing Your Configuration Specifications and Configuring the Domain

The Configuration Summary screen contains the 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 22 Writing Down Your Domain Home and Administration Server URL

The Configuration Success screen shows the following items about the domain you just configured:

- Domain Location
- Administration Server URL

You must make a note of both items as you need them later; the domain location is needed to access the scripts that are used to start 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 #unique_152 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 `wcpedg_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/wcpedg_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, soaedgadmin.

TRUSTSTOREFILE: Location of the truststore 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.

   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:

    ```
    wcpedg_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:

    ```
    wcpedg_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@wcchost1 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
```

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/wcpedg_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. Set the WLST Properties.

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

2. Start the WebLogic Scripting Tool (WLST):

   > **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.

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

3. 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`

4. 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.>
> ```

5. 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 Separate Domain Directory for Managed Servers on WCCHOST1

When you initially create the domain for enterprise deployment, the domain directory resides on a shared disk. This default domain directory is used to run the Administration Server. You

can now create a copy of the domain on the local storage for both WCCHOST1 and WCCHOST2. The domain directory on the local (or private) storage is used to run the Managed Servers.

Placing the MSERVER_HOME on local storage is recommended to eliminate the potential contention and overhead caused by servers writing logs to shared storage. It is also faster to load classes and jars need from the domain directory, so any temporary or cache data that the Managed Servers use from the domain directory is processed quicker.

As described in Preparing the File System for an Enterprise Deployment, the path to the Administration Server domain home is represented by the ASERVER_HOME variable, and the path to the Managed Server domain home is represented by the MSERVER_HOME variable.

To create the Managed Server domain directory:

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

   ```
   cd $ORACLE_COMMON_HOME/common/bin

   ./pack.sh -managed=true \
             -domain=ASERVER_HOME \
             -template=/full_path/create_domain.jar \
             -template_name=wcp_domain_template \
         -log_priority=DEBUG \
             -log=/tmp/pack.log
   ```

   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 location where you want to create the domain template jar file. You need to reference this location when you copy or unpack the domain template jar file. It is recommended to choose a shared volume other than *ORACLE_HOME*, or write to `/tmp/` and copy the files manually between servers.

     You must specify a full path for the template jar file as part of the `-template` argument to the pack command:

     ```
     SHARED_CONFIG_DIR/domains/template_filename.jar
     ```

   - The `create_domain.jar` file is a sample name for the jar file that you create, which contains the domain configuration files.
   - The `wcp_domain_template` label is the label is assigned to the template data stored in the template file.

2. Make a note of the location of the `create_domain.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*.

3. If you have not 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.

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/create_domain.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/create_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 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 Managed Server directory and verify that the domain configuration files were copied to the correct location on the WCCHOST1 local storage device.

Starting and Validating the WLS_WSM1 Managed Server on WCCHOST1

After you have configured Node Manager and created the Managed Server domain directory, you can use Oracle Enterprise Manager Fusion Middleware Control to start the WLS_WSM1 Managed Server on WCCHOST1.

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

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

   In this example:

   - Replace *ADMINVHN* with the host name assigned to the ADMINVHN Virtual IP address in Identifying and Obtaining Software Downloads for an Enterprise Deployment.
   - Port `9002` is the typical port used for the Administration Server console and Fusion Middleware Control. However, you should use the actual URL that was displayed at the end of the Configuration Wizard session when you created the domain.

   > **Tip**
   >
   > For more information about managing Oracle Fusion Middleware by using Oracle Enterprise Manager Fusion Middleware, see Getting Started Using Oracle Enterprise Manager Fusion Middleware Control in *Administering Oracle Fusion Middleware*.

2. Sign-in to the Fusion Middleware Control by using the administrator's account. For example: `weblogic`.
3. Select the **Servers** pane to view the Managed Servers in the domain.
4. Select only the **WLS_WSM1** Managed Server, and note the assigned port number.
5. Click **Control** > **Start** on the tool bar to start the selected **WLS_WSM1** Managed Server.
6. To verify that the Managed Server is working correctly, open your browser and enter the following URL:

   ```
   https://WCCHOST1:7010/wsm-pm/
   ```

   Enter the domain admin user name and password when prompted.

Configuring Web Services Manager

This section describes how to configure Web Services Manager.

Updating WebServices Domain Configuration

1. Log into the Fusion Middleware Control by using the administrator’s account.
2. In the **Weblogic Domain** drop down menu, select **WebServices** > **WSM Domain Configuration**.
3. Click **Policies Access** tab.
4. Select the **Auto Discover** and **Use SSL Only** check boxes.

5. In the **SSL Setup** section, select **Oneway**.
6. In KeyStore Type, select JKS (Java Key Store).

> **Note**
>
> You must select JKS if you are using the certificates and stores created in previous steps.

7. In the Truststore Path enter the location of the truststore used in previous sections as follows:

   ```
   /u01/oracle/config/keystores/appTrustKeyStore.pkcs12
   ```

8. In the **Key** field, enter a name to uniquely identify the password used for the truststore.
9. In the **password** field, enter the password used for the truststore in previous sections (same as domain admin).
10. Click **Apply**.

Bootstrapping WSM

In a new terminal window, perform the following steps to bootstrap WSMPM.

> **Note**
>
> If this task is not performed, the WSMPM does not work properly .

1. Change directory to the following directory as follows:

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

2. Start the WebLogic Server Scripting Tool (WLST). 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 (avoid including the password in the script):

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

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

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

   For example:

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

4. Use the `setWSMBootstrapConfig` to enable `WSM pm.url`.

   ```
   setWSMBootstrapConfig('domain_name','domainpath','ConfigManager','pm.url','
   auto-ssl')
   ```

   For example:

   ```
   setWSMBootstrapConfig('wcpedg_domain','/u01/oracle/config/domains/
   wcpedg_domain','ConfigManager','pm.url','auto-ssl')
   ```

   Check that the appropriate entry is created in the `$ASERVER_HOME/config/fmwconfig/wsm-config.xml` for the domain. For example:

   ```
   cat /u01/oracle/config/domains/wcpedg_domain/config/fmwconfig/wsm-
   config.xml
   <?xml version="1.0" encoding="UTF-8"?>
   <orares:properties xmlns:orares="http://xmlns.oracle.com/wsm/resources"
   xmlns:wsp15="http://www.w3.org/ns/ws-policy" xmlns:orawsp="http://
   schemas.oracle.com/ws/2006/01/policy" orares:type="CONFIGURATION"
   orares:resource="/">
      <orares:property orares:category="ConfigManager" orares:name="pm.url">
         <orares:value>auto-ssl</orares:value>
      </orares:property>
   </orares:properties>
   ```

Propagating the Domain and Starting the Servers on WCCHOST2

After you start and validate the Administration Server and WLS_WSM1 Managed Server on WCCHOST1, you can then perform the following tasks on WCCHOST2.

Unpacking the Domain Configuration on WCCHOST2

Now that you have the Administration Server and the first WLS_WSM1 Managed Server running on WCCHOST1, you can configure the domain on WCCHOST2.

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 `create_domain.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/create_domain.jar \
            -log_priority=DEBUG \
            -log=/tmp/unpack.log \
            -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 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 the Node Manager on WCCHOST2

After you manually set up the Node Manager to use a per host Node Manager configuration, you can start the Node Manager by using the following commands on WCCHOST2:

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 &
   ```

Starting and Validating the WLS_WSM2 Managed Server on WCCHOST2

Use the procedure in Starting and Validating the WLS_WSM1 Managed Server on WCCHOST1 to start and validate the WLS_WSM2 Managed Server on WCCHOST2.

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.

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 Remote 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 Portal enterprise deployment domain, then create a user called **`weblogic_wcp`** and an administration group called **`WCPAdministrators`**.

About the Domain Connector User

Oracle recommends that you create a separate domain connector user (for example, `wcpLDAP`) 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.

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 Remote 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`

8. Click **OK**.
9. Click **OK** again to persist the change.
10. 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 Portal

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 `WCPAdministrators` group. After they are added to the `WCPAdministrators` 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 `WCPAdministrators` 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_wcp,cn=users,dc=example,dc=com
  ```

- Admin Group DN:

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

- Product-specific LDAP Connector User:

  ```
  cn=wcpLDAP,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
  ```

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

Complete the prerequisites required to create an authentication provider and provision users and groups. Backup the relevant backup files and then enable authentication provider.

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

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 **Edit Tree**.

3. In the left navigational bar, click **Security** > **Realms** > **myrealm** > **Authentication Providers**.

> **Note**
>
> The **DefaultAuthenticator** provider is configured for the realm. This is the default WebLogic Server authentication provider.

4. Click **New button**.
5. 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
6. Select the authenticator type from the **Type** drop-down list.

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

   - `OracleUnifiedDirectoryAuthenticator` for Oracle Unified Directory
   - `OracleInternetDirectoryAuthenticator` for Oracle Internet Directory
7. Select **SUFFICIENT** from the **Control Flag** drop-down menu for the newly created authenticator provider.

   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 will fall 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`** and make sure that its control flag is set to **SUFFICIENT**.
8. Click **Create**.
9. Click the **Authenticator Parameters** 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=wcpLDAP, cn=systemids,dc=example,dc=com`	The LDAP user DN used to connect to the LDAP server.

Parameter	Sample Value	Value Description
Credential	Enter LDAP password.	The password used to connect to the LDAP server.
SSL Enabled	Unchecked (clear)	Specifies whether SSL protocol is used when connecting to the LDAP server.
User Base DN	For example: `cn=users,dc=example,dc=com`	Specify the DN under which your users start.
All Users Filter	`(&(uid=*)(objectclass=person))`	Instead of a default search criteria for **All Users Filter**, search all users based on the `uid` value. If the **User Name Attribute** for the user object class in the LDAP directory structure is a type other than `uid`, then change that type in the **User From Name Filter** field. For example, if the **User Name Attribute** type is `cn`, then this field should be set to: `(&(cn=*)(objectclass=person)))`
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.
Use Retrieved User Name as Principal	Checked	Must be turned on.
Group Base DN	For example: `cn=groups,dc=example,dc=com`	Specify the DN that points to your Groups node.
GUID Attribute	`entryuuid`	This value is prepopulated with `entryuuid` when `OracleUnifiedDirectoryAuthenticator` is used for OUD. Check this value if you are using Oracle Unified Directory as your authentication provider.

10. Click **Save** to save the changes.
11. Commit changes in the shopping cart.
12. Navigate to **Authenticator Providers** under **Security** > **Realms** > **myrealm**.
13. Check the **Authenticator Providers** you just created and move up to the first position.
14. On the **Authentication Providers** screen, click **DefaultAuthenticator**.
15. From the Control Flag drop-down, select **SUFFICIENT**.
16. Click **Save** to update the DefaultAuthenticator settings.
17. Commit the changes in the shopping cart.
18. 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:

```
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
```

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()
```

To start the Managed Servers, log in to Fusion Middleware Control, select the Managed Servers, and click **Start Up** in the toolbar.

> **Note**
>
> If you plan to log in to the system immediately by using the central LDAP user role, you can skip the restart until you have assigned the Administration role to the new enterprise deployment administration group. For more information, see Adding the Administration Role to the New Administration Group.

19. 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 they are correct and try saving and restarting the Administration Server again.

20. 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:

a. In the Remote Console, go to the **Security Tree**.

b. Navigate to **Realms** > **myrealm** > **Authentication Providers**.

c. Expand the new Authentication Provider.

d. Click **Users** and then click **Groups**.

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_wcp`** and a group called **`WCPAdministrators`**.

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

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=WCPAdministrators,cn=Groups,dc=example,dc=com
changetype: add
objectclass: top
```

```
objectclass: groupOfUniqueNames
objectclass: orclGroup
uniquemember: cn=weblogic_wcp,cn=users,dc=example,dc=com
cn: WCPAdministrators
displayname: WCPAdministrators
description: Administrators Group for the Oracle WebCenter Portal 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 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 Oracle Internet Directory, the group must be assigned the Administration role within the WebLogic domain security realm. This enables all users that belong to the group to be administrators for the domain.

To assign the Administration role to the new enterprise deployment administration group:

1. Log in to the WebLogic 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. Click **Security Data Tree**.
3. Click **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 `WCPAdministrators` in the **Group Argument Name** field, and then click **Add**.

   `WCPAdministrators` is added to the list box of arguments.

8. Click **Save** to finish adding the **Admin** Role to the `WCPAdministrators` group.

9. Validate that the changes were made by logging into the WebLogic Remote Console and into the Fusion Middleware Control by using the new `weblogic_wcp` user credentials.

   If you can log into 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.

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

- Admin Group DN:

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

- Product-specific LDAP Connector User:

  `cn=wcpLDAP,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
  ```

Adding the wsm-pm Role to the Administrators Group

After you configure a new LDAP-based Authorization Provider and restart the Administration Server, add the enterprise deployment administration LDAP group (`WCPAdministrators`) as a member to the `policy.Updater` role in the `wsm-pm` application stripe.

1. Sign in to the Fusion Middleware Control by using the administrator's account. For example: `weblogic_wcp`.
2. From the **WebLogic Domain** menu, select **Security**, and then **Application Roles**.
3. Select the **wsm-pm** application stripe from the Application Stripe drop-down menu.
4. Click the triangular icon next to the role name text box to search for all role names in the wsm-pm application stripe.
5. Select the row for the **policy.Updater** role to be edited.
6. Click the Application Role **Edit** icon to edit the role.
7. Click the Application Role **Add** icon on the Edit Application Role page.
8. In the Add Principal dialog box, select **Group** from the **Type** drop-down menu.

9. To search for the enterprise deployment administrators group, enter the group name `WCPAdministrators` in the **Principal Name Starts With** field and click the right arrow to start the search.
10. Select the appropriate administrators group in the search results and click **OK**.
11. Click **OK** on the Edit Application Role page.

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.

For information about backing up your configuration, see Performing Backups and Recoveries for an Enterprise Deployment.

Verification of Manual Failover of the Administration Server

After you configure the domain, test failover by following the steps that are described in Verifying Manual Failover of the Administration Server.

11

Configuring Oracle HTTP Server 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.

About the Oracle HTTP Server 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.

Variables Used When Configuring the Oracle HTTP Server

As you perform the tasks in this chapter, you reference the directory variables that are listed in this topic.

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 reference the following virtual IP (VIP) address and host names:

- ADMINVHN
- WEBHOST1
- WEBHOST2

- WCCHOST1
- WCCHOST2

Installing Oracle HTTP Server on WEBHOST1

It is important to understand the procedure for installing the Oracle HTTP Server software on 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 HTTP Server requires that you install a certified Java Development Kit (JDK) on your system.

You must install the JDK in 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.

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 17.0.10.

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`.

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 the Oracle 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 an Oracle HTTP Server Domain on WEBHOST1

The following topics describe how to create a new Oracle HTTP Server standalone domain on the first web tier host.

Starting the Configuration Wizard on WEBHOST1

To start the Configuration Wizard, navigate to the following directory and start the WebLogic Server Configuration Wizard, as follows:

```
cd $WEB_ORACLE_HOME/oracle_common/common/bin
./config.sh
```

Navigating the Configuration Wizard Screens for an Oracle HTTP Server Domain

Oracle recommends that you create a standalone domain for the Oracle HTTP Server instances on each web tier host.

The following topics describe how to create a new standalone Oracle HTTP Server domain:

- Task 1, Selecting the Domain Type and Domain Home Location
- Task 2, Selecting the Configuration Templates
- Task 3, Selecting the JDK for the Web Tier Domain.
- Task 4, Configuring System Components
- Task 5, Configuring OHS Server
- Task 7, Reviewing Your Configuration Specifications and Configuring the Domain
- Task 8, Writing Down Your Domain Home

Task 1 Selecting the Domain Type and Domain Home Location
On the Configuration Type screen, select **Create a new domain**.
In the **Domain Location** field, enter the value assigned to the *WEB_DOMAIN_HOME* variable.
Note the following:

- The Configuration Wizard creates the new directory that you specify here.
- Create the directory on local storage, so the web servers do not have any dependencies on storage devices outside the DMZ.

> **Tip**
>
> - 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. Use the **Restart Interval Seconds** field to specify the number of seconds to wait before you attempt a restart if an application is not responding.
4. Use the **Restart Delay Seconds** field to specify the number of seconds to wait between restart attempts.

Task 5 Configuring OHS Server
Use the OHS Server screen to configure the OHS servers in your domain:

1. Select **ohs1** from the **System Component** drop-down menu.
2. In the **Listen Address** field, enter the value of *`WEBHOST1`*.

   All the remaining fields are prepopulated, but you can change the values as required for your organization. The non-ssl listener will be disabled manually later in this guide. 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 an Oracle HTTP Server Domain on WEBHOST2

After you install Oracle HTTP Server and configure a domain on WEBHOST1, then you must also perform the same tasks on WEBHOST2.

1. Log in to WEBHOST2 and install Oracle HTTP Server by using the instructions in Installing Oracle HTTP Server on WEBHOST1.
2. Configure a new standalone domain on WEBHOST2 by 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/wcedgohs/nodemanager/
   nodemanager.domains
   LogLevel=INFO
   PropertiesVersion=14.1.2.0.0
   ListenBacklog=50
   QuitEnabled=false
   LogCount=1
   LogLimit=0
   NodeManagerHome=/u02/oracle/config/domains/wcedgohs/nodemanager
   LogToStderr=true
   ```

```
NativeVersionEnabled=true
AuthenticationEnabled=true
CrashRecoveryEnabled=false
weblogic.StopScriptEnabled=false
DomainsFileEnabled=true
weblogic.StartScriptEnabled=true
LogFile=/u02/oracle/config/domains/wcedgohs/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/soaedgohs> 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 Frontend Addresses and WebLogic Plugin for the WSM_PM Cluster and the Administration Server

As a security best practice oracle recommends setting a frontend address for the Administration Server and the WSM-PM cluster. In the initial domain creation steps, since OHS and the frontend Load Balancer may have not been configured yet, the frontend setting is avoided to allow verifications using the individual server addresses. However, at this point and

before configuring OHS (and the frontend load balancer, if not done yet) it is required to add the pertaining addresses.

1. To set the frontend and WebLogic Plugin 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. To set the frontend for the WSM-PM Cluster, use the Remote Console as follows:
   a. Click **Edit Tree**.
   b. Click **Environment**>**Clusters**>**WSM-PM_Cluster**.
   c. Select the **HTTP** tab.
   d. As **Frontend Host**, enter the front end LBR address that will be used to access internally to the Enteprise Deployment system (services like WSMPM and *internal.example.com* in the example used in this guide).
   e. Leave the **Frontend HTTP** port set to 0.
   f. Enter the LBR's internal listener (if your LBR does not allow using the same port from different listener this will have to be a different one from the admin port used for the Admin Server access) as **Frontend HTTPS port** (444).
   g. Click **Save**.
   h. Click the cart icon at the top right to commit the changes.

      These changes requires a restart of the AdminServer and the `WSM-PM_Cluster` to be effective (a notification appears in the WebLogic Remote Console about restart being required).
3. Enable the proxy plugin 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 *wcpinternal.example.com*, *wcp.example.com*, and *admin.example.com* as frontend addresses. These addresses are used in the WLS domain configuration as frontend 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/wcedg_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).

Since all virtual hosts in this EDG use SSL, the original `ssl.conf` file is used as a template for them. This Enterprise Deployment Guide segregates the listeners and certificates that are used by the different endpoints exposed through OHS. It uses different certificates and listeners for the external, internal and administration virtual hosts. This permits segregating the traffic and the encryption quality for each type of access and provides a well-structured mapping of front ends, Virtual Hosts and listeners.

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 from 2 to 7, 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 `wcpinternal_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 wcpinternal.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>

   <Location /wsm-pm>
   WLSRequest ON
   WebLogicCluster WCCHOST1:7010,WCCHOST2:7010
   </Location>

  </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/soaedgohs> nmKill(serverName='ohs1',serverType='OHS')
```

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

```
wls:/nm/soaedgohs> 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`
- `https://wcpinternal.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 Frontend LBR

Create a new Admin Server Connection Provider that connects through the frontend 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 frontend address.

1. Ensure that the Trust Store used by the WebLogic Remote Console includes the certificate or the CA certificate used by the frontend 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 frontend 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, `wcedg_domain_lbrprovider`.
   - **Username** and **Password**:

     Enter the WebLogic Domain Administration user and password.
   - **URL**: Use the frontend address and the port. For example, `https://admin.example.com:445`.
   - **Make Insecure Connect**: If the 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.

Synchronizing the System Clocks

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

Oracle recommends the use of the Network Time Protocol (NTP). See Configuring a Host to Use an NTP (time) Server.

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 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 Oracle WebCenter Content 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 example below.

```
$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
Installation Inventory Screen	If you did not create a central inventory when you installed the Oracle Fusion Middleware Infrastructure software, then this dialog box appears. Edit the **Inventory Directory** field so it points to the location of your local inventory, and then click **OK**.
Welcome	This screen introduces you to the product installer.
Auto Updates	Use this screen to automatically search My Oracle Support for available patches or automatically search a local directory for patches that you've already downloaded for your organization.
Installation Location	Use this screen to specify the location of your Oracle home directory. For more information about Oracle Fusion Middleware directory structure, see "Selecting Directories for Installation and Configuration" in *Planning an Installation of Oracle Fusion Middleware*.
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.
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.

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

Click **Next** to proceed, then click **OK** on the dialog window confirming that connection to the database was successful.

Task 4 Specifying a Custom Prefix and Selecting Schemas
Select **existing prefix**, and select the prefix you created while configuring the initial domain.
From the list of schemas, expand the **WebCenter Content** schema section and select only the **Oracle WebCenter Content Server - Complete** schema.
The custom prefix is used to logically group these schemas together for use in this domain only; you must create a unique set of schemas for each domain as schema sharing across domains is not supported.

> **Tip**
>
> For more information about custom prefixes, see "Understanding Custom Prefixes" in *Creating Schemas with the Repository Creation Utility*.
> For more information about how to organize your schemas in a multi-domain environment, see "Planning Your Schema Creation" in *Creating Schemas with the Repository Creation Utility*.

> **Tip**
>
> You must make a note of the custom prefix you choose to enter here; you will need this later on during the domain creation process.

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

Task 5 Specifying Schema Passwords
Specify how you want to set the schema passwords on your database, then specify and confirm your passwords.

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

Task 6 Verifying the Tablespaces for the Required Schemas
On the Map Tablespaces screen, review the information, and then click **Next** to accept the default values.
Click **OK** in the confirmation dialog box.

Task 7 Completing Schema Creation
Navigate through the remainder of the RCU screens to complete schema creation. When you reach the Completion Summary screen, click **Close** to dismiss RCU.

Task 8 Verifying the Schema Creation
To verify that the schemas were created successfully, and to verify the database connection details, use SQL*Plus or another utility to connect to the database, using the OCS schema name and the password you provided.
For example:

```
./sqlplus FMW1412_WLS/<wcpinfra_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.
>
> For more information about using the `setUserOverridesLate` script with this Enterprise Deployment Guide, see Customizing Server Parameters with the setUserOverridesLate Script.

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

Domain extension 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 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 - [wccontent]**

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

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

  – **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 Configuring High Availability Options

This screen appears for the first time when you create a cluster that uses Automatic Service Migration or JDBC stores or both. After you select HA Options for a cluster, all subsequent clusters that are added to the domain by using the Configuration Wizard, automatically apply HA options (that is, the Configuration Wizard creates the JDBC stores and configures ASM for them).

On the High Availability Options screen:

- Select **Enable Automatic Service Migration** with **Database Basis**.
- Set **JTA Transaction Log Persistence** to **JDBC TLog Store**.
- Set **JMS Server Persistence** to **JMS JDBC Store**.

> **Note**
>
> Oracle recommends that you use JDBC stores, which leverage the consistency, data protection, and high availability features of an oracle database and makes resources available for all the servers in the cluster. So, the Configuration Wizard steps assume that the JDBC persistent stores are used along with Automatic Service Migration. When you choose JDBC persistent stores, additional unused File Stores are automatically created but are not targeted to your clusters. Ignore these File Stores. If, for any reason, you want to use Files Stores, you can retain the default values for TLOGs and JMS persistent store options in this screen and configure them in a shared location later. See Task 8, Selecting Advanced Configuration. Shared location is required to resume JMS and HA in a failover scenario.
>
> You can also configure TLOGs and JMS persistent stores manually in a post step. For information about the differences between JDBC and Files Stores, and for specific instructions to configure them manually, see Using Persistent Stores for TLOGs and JMS in an Enterprise Deployment.

Click **Next**.

Task 4 Specifying the Database Configuration Type

On the Database Configuration Type screen, select **RCU Data**.

All fields are pre-populated, because you already configured the domain to reference the Fusion Middleware schemas that are required for the Infrastructure domain.

Verify and ensure that credentials in all the fields are the same that you have provided while configuring the Oracle Fusion Middleware Infrastructure.

Click **Get RCU Configuration** after you finish verifying the database connection information. The following output in the Connection Result Log indicates that the operating succeeded:

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

Successfully Done.
```

> **Tip**
>
> For more information about the **RCU Data** option, see "Understanding the Service Table Schema" in *Creating Schemas with the Repository Creation Utility*.
> For more information about the other options on this screen, see "Datasource Defaults" in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 5 Specifying JDBC Component Schema Information
On the JDBC Component Schema screen, select all the UCM schemas (for WebCenter Content) in the table.
When you select the schemas, the fields on the page are activated and the database connection fields are populated automatically.
Click **Convert to GridLink** and click **Next**.

Task 6 Providing the GridLink Oracle RAC Database Connection Details
On the GridLink Oracle RAC Component Schema screen, provide the information that is required to connect to the RAC database and component schemas, as shown in the following table.

Element	Description and Recommended Value
SCAN, Host Name, and Port	Select the **SCAN** check box. In the **Host Name** field, enter the Single Client Access Name (SCAN) Address for the Oracle RAC database. In the **Port** field, enter the SCAN listening port for the database (for example, 1521)
ONS Host and Port	These values are not required when you are using an Oracle 12*c* 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	16251	9005	UCM-MGD-SVR
WLS_WCC2	WCCHOST2	Disabled	Checked	16251	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.
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 `wcp.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 using unicast. If you want to change your cluster communications to use multicast, refer to Considerations for Choosing Unicast or Multicast in *Administering Clusters for Oracle WebLogic Server*.

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

Task 11 Assigning Server Templates
Click **Next** to proceed to the next screen.

Task 12 Configuring Dynamic Servers
Verify that all dynamic server options are disabled for clusters that are to remain as static clusters.

1. Confirm that the **Dynamic Cluster**, **Calculated Listen Port**, and **Calculated Machine Names** checkboxes on this screen are unchecked.
2. Confirm the **Server Template** selection is **Unspecified**.
3. Click **Next**.

Task 13 Assigning Managed Servers to the Cluster
Use the Assign Servers to Clusters screen to assign `WLS_WCC1` and `WLS_WCC2` to the new cluster, `WCC_Cluster`:

1. In the Clusters pane, select the cluster to which you want to assign the servers; in this case, `WCC_Cluster`.
2. In the Servers pane, assign WLS_WCC1 to `WCC_Cluster` by doing one of the following:

- Click once on the WLS_WCC1 Managed Server to select it, then click on the right arrow to move it beneath the selected cluster in the Clusters pane.
- Double-click on WLS_WCC1 to move it beneath the selected cluster in the clusters pane.

3. Repeat to assign WLS_WCC2 to WCC_Cluster.
4. Click **Next** to proceed to the next screen.

> **Tip**
>
> More information about the options on this screen can be found in Assign Servers to Clusters in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 14 Configuring Coherence Clusters
Use the Coherence Clusters screen to configure the Coherence cluster that is automatically added to the domain. Update the port number value to **9991**.

> **Note**
>
> For Coherence licensing information, refer to *Oracle Coherence* in *Oracle Fusion Middleware Licensing Information*.

Task 15 Creating Machines for WebCenter Content Servers
If required, use the Machines screen to add two new Unix Machines:

1. On the Unix Machines tab, click the **Add** button.
2. Enter WCCHOST1 in the **Name** field.
3. Enter the host name of WCCHOST1 for the Node Manage Listener address. Leave the Node Manager port to the default value of 5556.
4. Repeat the above steps for WCCHOST2.

Under the **Unix Machine** tab, verify the names of the machines you created when creating the initial Infrastructure domain, as shown in the following table.
Click **Next** to proceed.

Name	Node Manager Listen Address	Node Manager Listen Port
WCCHOST1	The value of the WCCHOST1 host name variable. For example, WCCHOST1.example.com.	5556
WCCHOST2	The value of the WCCHOST2 host name variable. For example, WCCHOST2.example.com.	5556

Task 16 Assigning Servers to Machines
Use the Assign Servers to Machines screen to assign the Oracle WebCenter Content Managed Servers you just created to the corresponding machines in the domain. Use a similar process as when assigning managed servers to the cluster. See Task 13, Assigning Managed Servers to the Cluster.
Assign WLS_WCC1 to WCCHOST1, and assign WLS_WCC2 to WCCHOST2.

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

Task 17 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 URL is needed to access the Administration Server.
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.

After you have completed extending the domain with static clusters, go to Completing Postconfiguration and Verification Tasks for WebCenter Content.

Update Certificates for New Frontend Addresses

This section contains information about certificates for new frontend addresses.

> **Note**
>
> About Certificates for the domain extension. Since the WCC and WSM servers use the same listen addresses (different port), there is no need to create new certificates and update stores. However, the WCC cluster uses a different front end address that will be added as a trusted endpoint in the Configuring the Web Tier for the Extended Domain.

Update the WebLogic Servers Security Settings

This section contains information about WebLogic Servers security settings.

Follow the steps described in the Updating the WebLogic Servers Security Settings and update SSL settings for the WLS_WCC1 and WLS_WCC2 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

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 WebCenter Content Managed Servers, complete the following steps:

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=/full_path/edgdomaintemplateExtWCC.jar
             -template_name=edgdomain_templateExtWCC
   ```

   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 location where you want to create the domain template jar file. You will need to reference this location when you copy or unpack the domain template jar file. It is recommended to choose a shared volume other than *ORACLE_HOME*, or write to `/tmp/` and copy the files manually between servers.

     You must specify a full path for the template jar file as part of the `-template` argument to the pack command:

     ```
     SHARED_CONFIG_DIR/domains/template_filename.jar
     ```

   - `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 the Oracle HTTP Server instances.
   - `edgdomain_templateExtWCC` is the name assigned to the domain template file.
3. Run the following `unpack` command on WCCHOST1 to propagate the template created in the preceding step to the *MSERVER_HOME* directory:

```
cd ORACLE_COMMON_HOME/common/bin

./unpack.sh -domain=MSERVER_HOME
            -template=/full_path/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 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 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.
- Replace *APPLICATION_HOME* with the complete path to the Application directory for the domain on local 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. If the full path to the packed jar file is on a shared volume available to the other servers, skip this step. Otherwise, run the following command on WCCHOST1 to copy the template pack created in step 1 to WCCHOST2. The WCPHOST*n* servers do not need the domain configuration at this time.

   ```
   scp /full_path/edgdomaintemplateExtWCC.jar oracle@WCCHOST2:/full_path/
   ```

5. Run the following `unpack` command on each of the remote hosts to deploy the domain template copied in the preceding step to the *MSERVER_HOME* directory:

   ```
   cd ORACLE_COMMON_HOME/common/bin

   ./unpack.sh -domain=MSERVER_HOME
               -template=/full_path/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 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 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.
- Replace *`APPLICATION_HOME`* with the complete path to the Application directory for the domain on local 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*.

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.

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_wcp`.
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. 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 `https://wcchost1:16201/cs` using the `weblogic` user name and password to display a configuration page.

   The Oracle WebCenter Content configuration files are on a shared disk so that all members of the cluster can access them. The shared disk location of the Oracle WebCenter Content enterprise deployment is at *`ORACLE_RUNTIME`*`/wcpedg_domain/WCC_Cluster`.

3. Change the following values on the server configuration page:

   Make sure that the **Is new Content Server Instance?** check box is selected.

   - **Content Server Instance Folder**: Set this to *`ORACLE_RUNTIME/domain_name/`*`WCC_Cluster/cs/`
     For example:`/u01/oracle/runtime/wcpedg_domain/WCC_Cluster/cs/`
   - **Native File Repository Location**: Set this to *`ORACLE_RUNTIME/domain_name/`*`WCC_Cluster/cs/vault/`
     For example:`/u01/oracle/runtime/wcpedg_domain/WCC_Cluster/cs/vault/`
   - **WebLayout Folder**: Set this to *`ORACLE_RUNTIME/domain_name`*`/WCC_Cluster/cs/weblayout/`
     For example:`/u01/oracle/runtime/wcpedg_domain/WCC_Cluster/cs/weblayout/`
   - **User Profile Folder**: Set this to *`ORACLE_RUNTIME/domain_name`*`/WCC_Cluster/cs/data/users/profiles/`
     For example:`/u01/oracle/runtime/wcpedg_domain/WCC_Cluster/cs/data/users/profiles/`
   - **Server Socket Port**: 4444
   - **Incoming Socket Connection Address Security Filter**: A pipe-delimited list of the local host and the server IP addresses:

     ```
     127.0.0.1|0:0:0:0:0:0:0:1|WCCHOST1-IP|WCCHOST2-IP|WCPHOST1-IP|WCPHOST2-IP|WEBHOST1-IP|WEBHOST2-IP|wcp.example.com-IP|wcpinternal.example.com-IP|load-balancer-host-IP
     ```

     > **Note**
     >
     > Must use IP addresses for all entries, including the load-balancer IP addresses for the internal virtual server and the primary interface depending on network address translation configuration settings at the load-balancer.

   - **WebServer HTTP/HTTPS Address:** `wcp.example.com:443`
   - **Web Address is HTTPS**: Select this checkbox.
   - **Server Instance Name**: `WCC_Cluster`
   - **Server Instance Label**: `WCC_Cluster`
   - **Server Instance Description:** WebCenter Content cluster
   - **Auto_Number Prefix**: `WCC_Cluster-`

4. Click **Submit** when finished.
5. Restart the Managed Server by using the WebLogic Remote Console.

Updating the cwallet File in the Administration Server

Content Server updates the `cwallet.sso` file located in the *`MSERVER_HOME`*`/config/fmwconfig` directory when it starts. This change needs to be propagated back to the Administration Server.

When adding WebCenter Content to a WebCenter Portal enterprise deployment, you will have to copy the cwallet file from the WC Content host (WCCHOST1) to `ASERVER_HOME` configuration directory on the server where the Administration Server is running. In the case of the EDG topology, both locations are on WCCHOST1 in this release.

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 and validate the WLS_WCC2 Managed Server, use the procedure in Starting and Validating the WLS_SOA1 Managed Server for WLS_SOA2 Managed Server.

Configuring the Content Server on WLS_WCC2 Managed Server

To configure Content Server:

1. Log in to WLS_WCC2 at `https://wcchost2:16201/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 Remote Console.

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 Granting the WebCenter Content Administrative Roles via Credential Map section.

> **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 `https://WCCHOST1:16201/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**.

   These changes will take effect after a restart of all WebCenter Content Managed Servers, at the end of the procedure described in Granting the WebCenter Content Administrative Roles via Credential Map section.

> **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 the WebCenter Content Administrative Roles through Credential Map

You must configure the Credential map to grant the Content Server administrative roles to the **`WCPAdministrators`** LDAP group.

The **`WCPAdministrators`** 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 **`WCPAdministrators`** 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 WCPAdministrators
group
     WCPAdministrators, admin
     WCPAdministrators, sysmanager
     WCPAdministrators, refineryadmin
     WCPAdministrators, rmaadmin
     WCPAdministrators, pcmadmin
     WCPAdministrators, ermadmin
# Allow existing roles to propagate without mappings
     |#all|             , %%
#
# Comment the following if you are not implementing Accounts in Content
Server
     WCPAdministrators, @#all(RWDA)
     WCPAdministrators, @#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_wcp` 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 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.

Generate the Required Certificates for OHS SSL Listeners

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

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

Configuring Oracle HTTP Server for the WLS_WCC Managed Servers

To configure the Oracle HTTP Server instances in the web tier so they route requests correctly to the Oracle WebCenter Content cluster, use the following procedure to create an additional Oracle HTTP Server configuration file that creates and defines the parameters of the `wcp.example.com` virtual server. To configure Oracle HTTP Server for the WLS_WCC Managed Servers:

1. Log in to WEBHOST1 and change directory to the configuration directory for the first Oracle HTTP Server instance (`ohs1`).
2. Copy the existing `admin_vh.conf` file to the `wcp_vh.conf` file. This will transfer most of the required SSL configuration:

   ```
   cp admin_vh.conf wcp_vh.conf
   ```

3. Edit the file and and customize with the required values for **Listen**, **ServerName**, **VirtualHost**, **SSLWallet** and **Location** directives.

   ```
   #[Listen] OHS_SSL_PORT

   Listen WEBHOST1:4443
   ##
   ## SSL Virtual Host Context
   ##
   #[VirtualHost] OHS_SSL_VH
   <VirtualHost WEBHOST1:4443>
   ServerName wcp.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_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 /cs>
        WebLogicCluster WCCHOST1:16201,WCCHOST2:16202
        WLSRequest ON
        WLCookieName JSESSIONID
        WLProxySSL ON
        WLProxySSLPassThrough ON
     </Location>

     <Location /adfAuthentication>
        WebLogicCluster WCCHOST1:16201,WCCHOST2:16202
        WLSRequest ON
        WLCookieName JSESSIONID
        WLProxySSL ON
        WLProxySSLPassThrough ON
     </Location>

     <Location /_ocsh>
        WebLogicCluster WCCHOST1:16201,WCCHOST2:16202
        WLSRequest ON
        WLCookieName JSESSIONID
        WLProxySSL ON
        WLProxySSLPassThrough ON
     </Location>
```

```
</IfModule>
</VirtualHost>
```

4. Copy the `wcp_vh.conf` file to the configuration directory for the second Oracle HTTP Server instance (`ohs2`):

   ```
   WEB_DOMAIN_HOME/config/fmwconfig/components/ohs2/moduleconf/
   ```

5. Edit the `wcp_vh.conf` and change any references of WEBHOST1 to WEBHOST2 in the `<VirtualHost>` directives.
6. Restart the Oracle HTTP server instances on WEBHOST1 and WEBHOST2.

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 Remote Console.
2. Access `https://wcp.example.com/cs` to verify that it is functioning properly.
3. Start WLS_WCC1 from the WebLogic Remote Console.
4. Stop WLS_WCC2 from the WebLogic Remote Console.
5. Access `https://wcp.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://wcp.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.

Configuring Oracle WebCenter Content for WebCenter Portal

This section describes tasks required for configuring Oracle WebCenter Content Server for use with WebCenter Portal.

This section includes the following topics.

Enabling Mandatory Content Server Components

For WebCenter Portal, you must enable the following Content Server components:

- **WebCenterConfigure** - Enable it to configure an instance of Content Server for WebCenter Portal.
- **Folders_g or FrameworkFolders** - Enable either of these components to specify the folder service configured on Content Server.
  - **Folders_g** - Provides a hierarchical folder interface to content in Content Server. For an Oracle WebCenter Portal instance patched from an earlier release that used the Folders_g component, you can continue to use Folders_g or choose to migrate to the FrameworkFolders interface. Oracle recommends that you migrate to the FrameworkFolders interface for better performance and so that you can use any new Content Server features.
  - **FrameworkFolders** - Provides a hierarchical folder interface similar to a conventional file system, for organizing and locating some or all of the content in the repository. FrameworkFolders is a scalable, enterprise solution and is intended to replace Folders_g as the folder service for Content Server. For new installations of Oracle WebCenter Portal, it is recommended that you enable the FrameworkFolders component on Content Server.

> **Note**
>
> Make sure you enable AutoSuggestConfig component before you enable FrameworkFolders component.

For detailed steps, see Enabling Mandatory Components in *Administering Oracle WebCenter Portal*.

> **Note**
>
> If Oracle WebCenter Portal is configured to use the Folders_g component, and Folders_g is not enabled, the following exception displays:
>
> ```
> SEVERE: UCM feature folders is not installed on server. at
> oracle.webcenter.content.integration.spi.ucm.UCMBridge.getBridge(UCMBridge.java:3
> 49) ....
> ```
>
> If Oracle WebCenter Portal is configured to use the FrameworkFolders component, and FrameworkFolders is not enabled, the following message is displayed:
>
> ```
> Foldering service from content server Folders_g and Portal Server Configuration
> FrameworkFolders do not match
> ```

Enabling and Configuring the Dynamic Converter Component

This task is optional, but strongly recommended.

This configuration is required for the Slide Previewer capability in WebCenter Portal, which makes use of the HTML renditions generated on the fly by the Dynamic Converter.

The configuration for the Dynamic Converter consists of two steps: enabling the Dynamic Converter, and defining the file types for which the Dynamic Converter is available. For detailed steps, see the Configuring the Dynamic Converter Component section in *Administering Oracle WebCenter Portal*.

Configuring Additional Content Server Features

There are several other Content Server features that, while not mandatory, can provide additional functionality in your WebCenter Portal enterprise deployment. For example, you can enable features such as Site Studio, OracleTextSearch, and so on.

To find out more, and for detailed steps, see the Configuration Roadmap for Content Server section in *Administering Oracle WebCenter Portal*.

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.

For information about backing up your configuration, see Performing Backups and Recoveries for an Enterprise Deployment.

13

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.
>
> For more information about using the `setUserOverridesLate` script with this Enterprise Deployment Guide, see Customizing Server Parameters with the setUserOverridesLate Script.

To start the Configuration Wizard:

1. From the Oracle WebLogic Remote Console, stop any managed servers that are modified by this domain extension. Managed Servers that are not effected can remain on-line.
2. For any managed servers to be modified, verify that the managed server shutdown has completed.
3. Stop the Administration Server once all managed servers are in a steady state.
4. Navigate to the following directory and start the WebLogic Server Configuration Wizard.

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

Navigating the Configuration Wizard Screens to Extend the Domain

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

> **Note**
>
> You can use the same procedure described in this section to extend an existing domain. If your needs do not match the instructions given in the procedure, be sure to make your selections accordingly, or refer to the supporting documentation for additional details.

Domain creation and configuration includes the following tasks:

Task 1 Selecting the Domain Type and Domain Home Location
On the Configuration Type screen, select **Update an existing domain**.
In the **Domain Location** field, select the value of the *ASERVER_HOME* variable, which represents the complete path to the initial Administration Server domain home you created. For more information about the directory location variables, see File System and Directory Variables Used in This Guide.

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

Click **Next** to proceed.

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

- **Oracle Universal Content Management - Inbound Refinery - [wccontent]**

  The Infrastructure templates, WebCenter Portal templates, and WebCenter Content templates should already be selected, because they were used to create and update the initial domain.

> **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 dministrationPort
WLS_IBR1	WCCHOST1	16250	16251	9IBR-MGD-SVR 00 6
WLS_IBR2	WCCHOST2	16250	16251	9IBR-MGD-SVR 00 6

> **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
Verity that all dynamic server options are disabled and unchecked for the `IBR_Servers` static (configured) cluster.

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:

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:

   `https://wcchostN:16251/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/wcpedg_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/wcpedg_domain/IBR_Servers/ibr1/vault

- **WebLayout Folder**: Set this to *ORACLE_RUNTIME/domain_name/*IBR_Servers/ibr*N*/weblayout

  For example: /u01/oracle/runtime/wcpedg_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/wcpedg_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/wcpedg_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/wcpedg_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 Remote 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).
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://wcp.example.com/cs`
2. From the **Administration** tray or menu, choose **Admin Server**, then **Component Manager**.
3. On the Component Manager page, select **Inbound Refinery**, then select components that you want to enable under Inbound Refinery, such as **XMLConverterSupport**, and then click **Update**.
4. Restart both Content Servers by restarting the WebCenter Content Managed Servers, using the 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://wcp.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*.

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.

For information about backing up your configuration, see Performing Backups and Recoveries for an Enterprise Deployment.

14

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 are 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
- WEB_DOMAIN_HOME
- JAVA_HOME
- ORACLE_RUNTIME

In addition, you reference the following virtual IP (VIP) address that are defined in Reserving the Required IP Addresses for an Enterprise Deployment:

- ADMINVHN

Actions in this chapter are performed on the following host computers:

- WCPHOST1
- WCPHOST2
- WEBHOST1
- WEBHOST2
- WCCHOST1
- WCCHOST2

Synchronizing the System Clocks

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

Oracle recommends the use of the Network Time Protocol (NTP). See Configuring a Host to Use an NTP (time) Server.

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 -d64 -jar Installer File Name
   ```

   Be sure to replace the JDK location in these examples with the actual JDK location on your system.

   Replace *`Installer File Name`* with the name of the actual installer file for your product listed in Identifying and Obtaining Software Distributions for an Enterprise Deployment.

When the installation program appears, you are ready to begin the installation.

Navigating the Installation Screens

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

If you need additional help with any of the installation screens, click the screen name.

Screen	Description
Welcome	This screen introduces you to the product installer.
Auto Updates	Use this screen to automatically search My Oracle Support for available patches or automatically search a local directory for patches that you have already downloaded for your organization.
Installation Location	Use this screen to specify the location of your Oracle home directory. For more information about Oracle Fusion Middleware directory structure, see Selecting Directories for Installation and Configuration in *Planning an Installation of Oracle Fusion Middleware*.

Screen	Description
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*.
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
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 5*.
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**

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 `LARGE` for the **Database Profile** custom variable; if you plan to use Oracle Healthcare, then enter `YES` 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_SOAINFRA/<soainfra_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>
```

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

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**
>
> SSL store customizations were added to the `setUserOverridesLate.sh` in the domain creation chapter. Any customizations added to this file are preserved when a domain is extended and are carried over to remote servers when using the pack and unpack commands.However, if you added any additional customizations to the `setDomainEnv.sh` script in the domain (such as custom libraries, JAVA command line options for starting the servers or environment variables), those will be overwritten by the configuration wizard when you extend the domain. Add all the startup parameters that apply to all servers in a domain to the `setUserOverridesLate.sh` file. This will preserve them across extensions.

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 with static clusters.

Extending the Domain with Static Clusters

Follow the instructions in this section to extend the domain for Oracle SOA Suite, with static clusters.

> **Note**
>
> This procedure assumes that you are extending an existing domain. If your needs do not match the instructions given in the procedure, ensure that you make your selections accordingly, or refer to the supporting documentation for additional details.

Domain creation and configuration includes the following tasks:

- Task 1, Selecting the Domain Type and Domain Home Location
- Task 2, Selecting the Configuration Template
- Task 3, Configuring High Availability Options
- Task 4, Specifying the Database Configuration Type
- Task 5, Specifying JDBC Component Schema Information
- Task 6, Providing the GridLink Oracle RAC Database Connection Details
- Task 7, Testing the JDBC Connections
- Task 8, Keystore
- Task 9, Selecting Advanced Configuration
- Task 10, Configuring Managed Servers
- Task 11, Configuring a Cluster
- Task 12, Assigning Server Templates
- Task 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, Reviewing Your Configuration Specifications and Configuring the Domain
- Task 19, Writing Down Your Domain Home and Administration Server URL
- Task 20, Start the Administration Server

Task 1 Selecting the Domain Type and Domain Home Location
On the Configuration Type screen, select **Update an existing domain**.
In the **Domain Location** field, select the value of the *ASERVER_HOME* variable, which represents the complete path to the Administration 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 - 14.1.2.0.0 [soa]**

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 12*c* 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.
> If you choose File Stores in Task 3, Configuring High Availability Options, you have to select the File Stores option here to configure them in a shared location in `ORACLE_RUNTIME/domain_name/SOA_Cluster/jms`. Shared location is required to resume JMS and JTA in a failover scenario.

Task 10 Configuring Managed Servers

On the Managed Servers screen, a new Managed Server for Oracle SOA Suite appears in the list of servers. This server was created automatically by the Oracle SOA Suite configuration template that you selected in Task 2, Selecting the Configuration Template.
Perform the following tasks to modify the default Oracle SOA Suite Managed Server and create a second Oracle SOA Suite Managed Server:

1. Rename the default Oracle SOA Suite Managed Server to `WLS_SOA1`.
2. Click **Add** to create a new Oracle SOA Suite Managed Server, and name it `WLS_SOA2`.

> **Tip**
>
> The server names recommended here are used throughout this document; if you choose different names, be sure to replace them as needed.

3. Use the information in the following table to fill in the rest of the columns for each Oracle SOA Suite Managed Server.

For more information about the options on the Managed Server screen, see Managed Servers in *Creating WebLogic Domains Using the Configuration Wizard*.

Server Name	Listen Address	Listen Port	SSL Listen Port	Server Groups
WLS_SOA1	WCCHOST1	Disabled	8001	SOA-MGD-SVRS-ONLY
WLS_SOA2	WCCHOST2	Disabled	8001	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.
Use the Clusters screen to create a new cluster:

1. Click the **Add** button.

2. Specify `SOA_Cluster` in the **Cluster Name** field.
3. From the **Dynamic Server Groups** drop-down list, select `Unspecified`.

> **Note**
>
> By default, server instances in a cluster communicate with one another by using unicast. If you want to change your cluster communications to use multicast, refer to Considerations for Choosing Unicast or Multicast in *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
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 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 19 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 Remote 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 20 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** > **Deployments** > **App Deployments**.
2. Locate and click the name of the adapter in the Summary of the Deployments table.
3. In the **Targets** tab, select **SOA_Cluster** and move it to the **Chosen** pane.

> **Note**
>
> If you are deploying MFT, select MFT_Cluster as the target.

4. **Add The cart** on the top right part of the screen will show full with a yellow bag inside.
5. Click the **Cart** icon on the top right and select **Commit Changes**.
6. In the **Navigation Tree** pane of the console, navigate to **Monitoring Tree** > **Deployments** > **Ap Development Runtimes** and verify that the adapter is in the Active state.

Propagating the Extended Domain to the Domain Directories and Machines

After you have extended the domain with the Oracle WebCenter Portal instances, and you have restarted the Administration Server on WCCHOST1, you must then propagate the domain changes to the domain directories and machines.

Table 14-2 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 14-2 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 `wcpdomaintemplateExtSOA.jar`.	Packing Up the Extended Domain on WCPHOST1
Unpack the Domain in the Managed Servers directory on WCPHOST1	Unpack the template JAR file in the Managed Servers directory on WCPHOST1 local storage.	Unpacking the Domain in the Managed Servers Domain Directory on WCPHOST1
Unpack the Domain on WCPHOST2 Unpack the Domain on WCCHOST1 and WCCHOST2	Unpack the template JAR file in the Managed Servers directory on the WCPHOST2local storage. Unpack the JAR file, later, in the Managed Services directory on WCCHOST1 and WCCHOST2 local storage.	Unpacking the Domain on WCPHOST2

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/wcpdomaintemplateExtSOA.jar \
          -template_name=wcp_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.
   - `wcpdomaintemplateExtSOA.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.
   - `wcp_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/wcpdomaintemplateExtSOA.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/wcpdomaintemplateExtSOA.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 that you have copied the file which you created earlier into a location that is accessible from each application-tier host, whether that is local to each host, or on shared storage.

1. Log in to WCCHOST2.
2. If you have not 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 that the `create_domain.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/create_domain.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/create_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 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.
6. Repeat this procedure for WCPHOST1 and WCPHOST2.

Restarting and Validating Pre-existing Managed Servers

It is necessary to restart the managed servers for the pre-existing components after the domain has been extended and unpacked to the *MSERVER_HOME* directories on all of the servers.

Restart the managed servers for the pre-existing components now that the domain has been extended and unpacked to the *MSERVER_HOME* directories on all of the servers.

1. From the WebLogic Server Console, restart the WLS_WSM*n* Managed Servers for the WebServices Manager Policy Manager.

2. From another browser window, verify the WSM-PM application is responding by successfully loading the URL:

   ```
   http://wcpinternal.example.com/wsm-pm/validator
   ```

3. Restart other pre-existing managed servers as necessary. Other product functionality will not be needed at this stage.

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 in to the WebLogic Remote Console to access the provider of this domain.
2. In the left navigation tree, expand **Domain**, and then **Environment**.
3. Click **Lock & Edit**.
4. Navigate to and edit the appropriate objects for your cluster type.
   a. For Static Clusters, navigate to **Servers** and click the name of the Managed Server you want to edit.
   b. For Dynamic Clusters, navigate to **Clusters** > **Server Templates**, and click on the name of the server template to be edited.
5. For each new Managed Server or Server Template to be edited:
   a. Click the **Configuration** tab, and then click the **Deployment** tab.
   b. Verify that the **Staging Directory Name** is set to the following:

      MSERVER_HOME/servers/*server_or_template_name*/stage

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

      If you use static clusters, update with the correct name of the Managed Server that you are editing.

      If you use dynamic clusters, leave the template name intact. For example: /u02/oracle/config/domains/wcpedg_domain/servers/*XYZ-server-template*/stage
   c. Update the **Upload Directory Name** to the following value:

      ASERVER_HOME/servers/AdminServer/upload

      Replace *ASERVER_HOME* with the directory path for the *ASERVER_HOME* directory.
   d. Click **Save**.
   e. Return to the Summary of Servers or Summary of Server Templates screen as applicable.

6. Repeat the previous steps for each of the new managed servers.
7. Navigate to and update the Upload Directory Name value for the AdminServer:
   a. Navigate to **Servers**, and select the AdminServer.
   b. Click the **Configuration** tab, and then click the **Deployment** Tab.
   c. Verify that the **Staging Directory Name** is set to the following absolute path:

      ASERVER_HOME/servers/AdminServer/stage

   d. Update the **Upload Directory Name** to the following absolute path:

      ASERVER_HOME/servers/AdminServer/upload

      Replace *ASERVER_HOME* with the directory path for the *ASERVER_HOME* directory.
   e. Click **Save**.
8. When you have modified all the appropriate objects, click **Activate Changes**.

> **Note**
>
> If you continue directly with further domain configurations, a restart to enable the stage and upload directory changes is not strictly necessary at this time.

Starting and 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

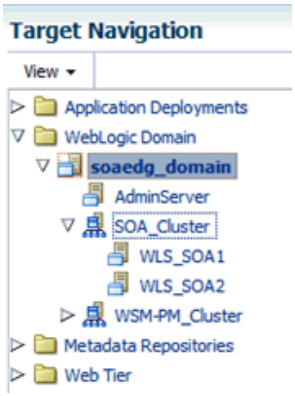
To start the WLS_SOA1 Managed Server:

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

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

   > **Note**
   >
   > If you have already configured web tier, use `http://admin.example.com/console`.

2. Sign in to the Fusion Middleware Control by using the administrator's account. For example: `weblogic_wcp`.
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 (`WCPAdministrators`).

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_wcp`).

   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_wcp`):

`https://WCCHOSt2:8001/soa-infra`

Configuring the Web Tier for the Extended Domain

The following sections describe how to configure the Oracle HTTP Server instances so they route requests for both public and internal URLs to the proper clusters in the enterprise topology.

Configuring Oracle HTTP Server for SOA in an Oracle WebCenter Portal Enterprise Deployment

When integrating SOA with WebCenter Portal for workflow functionality both internal and end-user access to the SOA Suite resources should be configured.

Use the following procedure to configure the Oracle HTTP Server instances in the web tier, so they route requests correctly to the Oracle SOA Suite cluster. This procedure assumes that you have performed the Oracle HTTP Server configuration tasks described in Configuring Oracle HTTP Server to Route Requests to the Application Tier.

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 `wcp_vh.conf` file and add the following directives inside the `<VirtualHost>` tags:

> **Note**
>
> - The URL entry for `/workflow` is optional. It is for workflow tasks associated with Oracle ADF task forms. The `/workflow` URL itself can be a different value, depending on the form.
> - Configure the port numbers appropriately as assigned for your static or dynamic cluster. Dynamic clusters with the Calculate Listen Port option selected will have incremental port numbers for each dynamic managed server created.
> - The WebLogicCluster directive needs only a sufficient number of redundant `server:port` combinations to guarantee initial contact in case of a partial outage. The actual total list of cluster members is retrieve automatically upon first contact with any given node.

```
# soa-infra
<Location /soa-infra>
    WLSRequest ON
    WebLogicCluster WCCHOST1:8001,WCCHOST2:8001
    WLProxySSL OFF
    WLProxySSLPassThrough OFF
</Location>

# SOA inspection.wsil
<Location /inspection.wsil>
    WLSRequest ON
    WebLogicCluster WCCHOST1:8001,WCCHOST2:8001
    WLProxySSL OFF
    WLProxySSLPassThrough OFF
</Location>

# Worklist
<Location /integration>
    WLSRequest ON
    WebLogicCluster WCCHOST1:8001,WCCHOST2:8001
    WLProxySSL OFF
    WLProxySSLPassThrough OFF
</Location>

# UMS prefs
<Location /sdpmessaging/userprefs-ui>
    WLSRequest ON
    WebLogicCluster WCCHOST1:8001,WCCHOST2:8001
    WLProxySSL OFF
    WLProxySSLPassThrough OFF
</Location>

# Default to-do taskflow
<Location /DefaultToDoTaskFlow>
    WLSRequest ON
    WebLogicCluster WCCHOST1:8001,WCCHOST2:8001
    WLProxySSL OFF
    WLProxySSLPassThrough OFF
</Location>

# Workflow
<Location /workflow>
```

```
    WLSRequest ON
    WebLogicCluster WCCHOST1:8001,WCCHOST2:8001
    WLProxySSL OFF
    WLProxySSLPassThrough OFF
</Location>

#Required if attachments are added for workflow tasks
 <Location /ADFAttachmentHelper>
    WLSRequest ON
    WebLogicCluster WCCHOST1:8001,WCCHOST2:8001
    WLProxySSL OFF
    WLProxySSLPassThrough OFF
</Location>

# SOA composer application
 <Location /soa/composer>
    WLSRequest ON
    WebLogicCluster WCCHOST1:8001,WCCHOST2:8001
    WLProxySSL OFF
    WLProxySSLPassThrough OFF
</Location>

 <Location /frevvo>
    WLSRequest ON
    WebLogicCluster WCCHOST1:8001,WCCHOST2:8001
    WLProxySSL OFF
    WLProxySSLPassThrough OFF
</Location>
</VirtualHost>
```

3. Copy the `wcpinternal_vh.conf`and `wcp_vh.conf` files to the configuration directory for the second Oracle HTTP Server instance (`ohs2`):

   `WEB_DOMAIN_HOME``/config/fmwconfig/components/`**`ohs2`**`/moduleconf/`

4. Edit the `wcpinternal_vh.conf` and `wcp_vh.conf` to change any references to WEBHOST1 to WEBHOST2 in the `<VirtualHost>` directives.

5. Restart both Oracle HTTP servers.

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 WebLogic Remote 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://wcp.example.com:443/soa-infra`
   - `https://wcp.example.com:443/integration/worklistapp`
   - `https://wcp.example.com:443/sdpmessaging/userprefs-ui`
   - `https://wcp.example.com:443/soa/composer`

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 WebLogic 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 14-3 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 14-1.

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

Example 14-1 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_322/unique_322_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/wcpedg_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:t3://cluster_name
```

When you use `cluster:t3://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:
   Copy

   ```
   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 Creating the boot.properties File.
2. In the left navigation tree, expand **WebLogic Domain**, and then click **System MBean Browser**.
3. Navigate to **Application Defined Mbean** > **oracle.as.soainfra.config** > **WLS_SOA1** > **WorkflowConfig** > **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.

4. On the right panel, look for the **FusionAppsFrontendHostUrl** attribute.
5. For the **FusionAppsFrontendHostUrl** attribute, specify the value `*=https://wcp.example.com:443`.
6. Click **Apply**.

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.

For information about backing up your configuration, see Performing Backups and Recoveries for an Enterprise Deployment.

15

Extending the Domain with Oracle WebCenter Portal

The following topics describes how to extend the enterprise deployment domain with the Oracle WebCenter Portal software.

Variables Used When Extending the Domain for WebCenter Portal

As you perform the tasks in this chapter, you will be referencing the directory variables listed in this section.

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

- ORACLE_HOME
- ASERVER_HOME
- APPLICATION_HOME
- MSERVER_HOME
- DEPLOY_PLAN_HOME
- WEB_DOMAIN_HOME
- JAVA_HOME

In addition, you'll be referencing the following virtual IP (VIP) addresses and host names defined in Physical and Virtual IP Addresses Required by the Enterprise Topology:

- ADMINVHN
- WCCHOST1
- WCCHOST2
- WCPHOST1
- WCPHOST2
- DBHOST1
- DBHOST2
- SCAN Address for the Oracle RAC Database (DB-SCAN.example.com)

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 WebCenter Portal 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 example below.

   ```
   $JAVA_HOME/bin/java  -jar fmw_14.1.2.0.0_wcportal.jar
   ```

   Be sure to replace the JDK location in these examples with the actual JDK location on your system.

   For information about downloading the software and locating the actual installer file name for your product, see 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'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*.
Installation Type	Use this screen to select the type of installation and consequently, the products and feature sets you want to install. Select **WebCenter Portal**, then click **Next**. **Notes:** • WebCenter Portal Worklist integration requires that the BPEL Services provided by SOA Suite share the same WebTier, SSO, and Identity Store. • For this Enterprise Deployment Guide, SOA Suite is installed and configured in the same WebLogic Server Domain and included in the WebTier, SSO, and Directory configurations. • Whether the installation has SOA Suite installed in the same or separate *ORACLE_HOME*, the WebCenter Portal SOA Composites option is required as a separate installation. For instructions about that installation see, Installing the Oracle WebCenter Portal SOA Composites.
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 following documents in the Roadmap for Verifying Your System Environment section in *Planning Your Oracle Fusion Middleware Infrastructure Installation.*

Screen	Description
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 Portal on WCCHOST2

If you have followed the EDG shared storage recommendations, there is a separate shared storage volume for product installations mounted on the `*HOST2` hosts. You must also install the software on to the second products volume. It is recommended to execute the installs consistently from the `WCCHOST`*n* hosts. See Shared Storage Recommendations When Installing and Configuring an Enterprise Deployment.

Creating the Oracle WebCenter Portal Database Schemas

Before you can configure an Oracle WebCenter Portal domain, you must install the required schemas on a certified database for use with this release of Oracle Fusion Middleware.

The following topics describe how to install the required 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

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

> **Note**
>
> 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.

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

Task 4 Specifying a Custom Prefix and Selecting Schemas
Select **Select existing prefix**, and then select the prefix you used when you created the initial domain in Creating the Initial Infrastructure Domain for an Enterprise Deployment.
From the list of schemas, select **WebCenter Portal** . This will automatically select the required WebCenter Portal schemas. In addition, the following dependent schemas have already been installed with the Infrastructure and are grayed out:

- Metadata Services
- Audit Services
- Audit Services Append
- Audit Services Viewer
- Oracle Platform Security Services
- User Messaging Service

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 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.

> **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.

Task 6 Specifying Custom Variables
For an enterprise deployment, Oracle recommends that you enter “Y” to enable partitioning of the Analytics data.
This feature partitions the analytics data by month. In a partitioned environment, the recommended method for purging data is simply to drop the month-based partitions that are no longer required.
For information about partitioning analytics data, see "Partitioning Oracle WebCenter Portal's Analytics Data" in the *Oracle Fusion Middleware Administrator's Guide*.

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.

Task 8 Completing Schema Creation
Navigate through the remainder of the RCU screens to complete schema creation. When you reach the Completion Summary screen, click **Close** to dismiss RCU.

Task 9 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 WCPINFRA schema name and the password you provided.
For example:

```
./sqlplus FMW1412_WCPINFRA/<wcpinfra_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 Enterprise Deployment Domain with Oracle WebCenter Portal

This section provides instructions for extending the existing enterprise deployment domain with the Oracle WebCenter Portal software.

Extending the domain involves the following:

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.
>
> For more information about using the `setUserOverridesLate` script with this Enterprise Deployment Guide, see Customizing Server Parameters with the setUserOverridesLate Script.

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 Portal

Follow the instructions in these sections to extend the domain for Oracle WebCenter Portal.

> **Note**
>
> This procedure assumes that you are extending an existing domain. If your needs do not match the instructions given in the procedure, ensure that you make your selections accordingly, or refer to the supporting documentation for additional details.

Domain creation and configuration includes the following tasks.

- Task 1, Selecting the Domain Type and Domain Home Location
- Task 2, Selecting the Configuration Template
- Task 3, Specifying the Database Configuration Type
- Task 4, Specifying JDBC Component Schema Information
- Task 5, Providing the GridLink Oracle RAC Database Connection Details
- Task 6, Testing the JDBC Connections
- Task 7, Selecting Advanced Configuration
- Task 8, Configuring Managed Servers
- Task 9, Assign Servers to Clusters
- Task 10, Configuring Clusters
- 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, Verifying the Existing Machines
- Task 16, Assigning Servers to Machines
- Task 17, Deployments Targeting
- Task 18, Services Targeting
- Task 19, Reviewing Your Configuration Specifications and Configuring the Domain
- Task 20, Writing Down Your Domain Home and Administration Server URL
- Task 21, Start the Administration Server

Task 1 Selecting the Domain Type and Domain Home Location
On the Configuration Type screen, select **Update an existing domain**.
In the **Domain Location** field, select the value of the ASERVER_HOME variable, which represents the complete path to the Administration Server domain home you created in Creating the Initial Infrastructure Domain for an Enterprise Deployment.

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 Portal – 14.1.2.0.0 [wcportal]**
- **Oracle WebCenter Pagelet Producer – 14.1.2.0.0 [wcportal]**
- **Oracle WebCenter Portlet Producers - 14.1.2.0.0 [wcportal]**
- **Oracle WebCenter Analytics Collector - 14.1.2.0.0 [wcportal]**

In addition, the following additional templates should already be selected, because they were used to create the initial domain in Creating the Initial Infrastructure Domain for an Enterprise Deployment:

- **Oracle Enterprise Manager - 12.2.1.3.0 [em]**
- **Oracle WSM Policy Manager - 12.2.1.3.0 [oracle_common]**
- **Oracle JRF - 12.2.1.3.0 [oracle_common]**
- **WebLogic Coherence Cluster Extension - 12.2.1.3.0 [wlserver]**

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

Task 3 Specifying the Database Configuration Type
On the Database Configuration Type screen, select **RCU Data**.
All fields are pre-populated, because you already configured the domain to reference the Fusion Middleware schemas that are required for the Infrastructure domain.
Verify and ensure that credentials in all the fields are the same that you have provided while configuring Oracle Fusion Middleware Infrastructure.
Click **Get RCU Configuration** after you finish verifying the database connection information.
The following output in the Connection Result Log indicates that the 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.
```

> **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
On the JDBC Component Schema screen, select all the WebCenter Portal schemas in the table.
When you select the schemas, the fields on the page are activated and the database connection fields are populated automatically.
Click **Convert to GridLink** and click **Next**.

Task 5 Providing the GridLink Oracle RAC Database Connection Details
On the GridLink Oracle RAC Component Schema screen, provide the information 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, `wcpedg.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** checkbox is selected, so the database can receive and process FAN events.

Task 6 Testing the JDBC Connections
Use the JDBC Component Schema Test screen to test the data source connections you have just configured.
A green check mark in the Status column indicates a successful test. If you encounter any issues, see the error message in the Connection Result Log section of the screen, fix the problem, then try to test the connection again.

> **Tip**
>
> For more information about the other options on this screen, see Test Component Schema in *Creating WebLogic Domains Using the Configuration Wizard*

Task 7 Selecting Advanced Configuration
To complete domain configuration for the topology, select the following options on the Advanced Configuration screen:

- **Topology**
- **Deployments and Services**

Task 8 Configuring Managed Servers

On the Managed Servers screen, new Managed Servers for Oracle WebCenter Portal and Portlets appear in the list of servers along with the other Managed Servers that were created earlier. These servers are created automatically by the Oracle WebCenter Portal configuration template you selected earlier in the Configuration Wizard session.
Perform the following tasks to modify the default Managed Servers and create a second Managed Server for each server type:

1. Select WC_Portal and rename it to WC_Portal1.
2. Select **Clone** to create another managed server. Rename the new server to WC_Portal2.
3. Repeat the above two steps to edit and create WC_Portlet1 and WC_Portlet2.

> **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
WC_Portal1	WCPHOST1	Disabled	Checked	8002	9010	WebCenter Portal Managed Server WebCenter Portal Analytics Managed Server
WC_Portal2	WCPHOST2	Disabled	Checked	8002	9010	WebCenter Portal Managed Server WebCenter Portal Analytics Managed Server
WC_Portlet1	WCPHOST1	Disabled	Checked	8003	9010	WebCenter Portal Pagelet Producer Managed Server WebCenter Portal Portlet Producer Managed Server
WC_Portlet2	WCPHOST2	Disabled	Checked	8003	9010	WebCenter Portal Pagelet Producer Managed Server WebCenter Portal Portlet Producer Managed Server

Task 9 Assign Servers to Clusters
Confirm that all static configured Managed Servers are assigned to the correct Clusters. Click **Next** to proceed to the next screen.

Task 10 Configuring Clusters
In the Configure Clusters screen, add the following new clusters:

- Portal_Cluster
- Portlet_Cluster

For all three clusters, leave the default values for **Cluster Address**, **Frontend Host**, and **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 Managed Servers to their respective cluster:

1. In the Clusters pane, select the cluster to which you want to assign the servers; in this case, `Portal_Cluster`.
2. In the Servers pane, assign WC_Portal1 to `Portal_Cluster` by doing one of the following:
   - Click once on `WC_Portal1` 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 `WC_Portal1` to move it beneath the selected cluster in the clusters pane.
3. Repeat to assign `WC_Portal2` to `Portal_Cluster`.
4. Repeat steps 1-3 to assign `WC_Portlet1` and `WC_Portlet2` to `Portlet_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 14 Configuring Coherence Clusters
Use the Coherence Clusters screen to configure the Coherence cluster that is automatically added to the domain. Leave the port number value at 9991, as it was defined during the initial Infrastructure domain creation.

> **Note**
>
> For Coherence licensing information, see Oracle Coherence Products in *Oracle Fusion Middleware Licensing Information User Manual*.

Task 15 Verifying the Existing Machines
Click **Next** to proceed.

Task 16 Assigning Servers to Machines
Use the Assign Servers to Machines screen to assign the Managed Servers you just created to the corresponding machines in the domain.
Assign `WC_Portal1`, `WC_Portlet1` to `WCPHOST1`.
Assign `WC_Portal2`, `WC_Portlet2` to `WCPHOST2`.

> **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 Deployments Targeting
With the Oracle Web Services Manager Policy Manager deployed to a separate cluster, the default targeting of the WSM-PM application to the Portal and Portlet clusters should be removed.
For each of the Portal_Cluster and Portlet_Cluster in the Targets panel:
Select the wsm-pm application entry within the Application folder and click the left arrow button to remove it from the targets list.

Task 18 Services Targeting
With the Oracle Web Services Manager Policy Manager deployed to a separate cluster, the default targeting of the service resources needed by the WSM-PM application can be removed from the Portal and Portlet clusters.
For each of the Portal_Cluster and Portlet_Cluster in the Targets panel:
Select and remove the following resource from the targets list:

- mds-owsm

Task 19 Reviewing Your Configuration Specifications and Configuring the Domain
The Configuration Summary screen contains the detailed configuration information for the domain you are about to create. Review the details of each item on the screen and verify that the information is correct.
You can go back to any previous screen if you need to make any changes, either by using the **Back** button or by selecting the screen in the navigation pane.

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 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

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 Remote Console and Oracle Enterprise Manager Fusion Middleware Control.
Click **Finish** to dismiss the configuration wizard.

Task 21 Start the Administration Server
Start the Administration Server to ensure the changes you have made to the domain have been applied.

After you have completed extending the domain with static clusters, go to Propagating the Extended Domain to the Domain Directories and Machines.

Propagating the Extended Domain to the Domain Directories and Machines

After you have extended the domain with the Oracle WebCenter Portal instances, and you have started the Administration Server on WCCHOST1, you must then propagate the domain changes to the domain directories and machines.

1. Make a backup 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=/full_path/wcpdomaintemplateExtWCP.jar
             -template_name=wcpdomain_templateExtWCP
   ```

   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 location where you want to create the domain template jar file. You will need to reference this location when you copy or unpack the domain template jar file. It is recommended to choose a shared volume other than *ORACLE_HOME*, or write to `/tmp/` and copy the files manually between servers.

You must specify a full path for the template jar file as part of the `-template` argument to the pack command:

`SHARED_CONFIG_DIR/domains/template_filename.jar`

- `wcpdomaintemplateExtWCP.jar` is a sample name for the JAR file you are creating, which will contain the domain configuration files, including the configuration files for the Oracle HTTP Server instances.
- `wcpdomain_templateExtWCP` is the name assigned to the domain template file.

3. Run the following `unpack` command on WCCHOST1 to propagate the template created in the preceding step to the *MSERVER_HOME* directory:

```
cd $ORACLE_COMMON_HOME/common/bin

./unpack.sh -domain=MSERVER_HOME
            -template=/full_path/wcpdomaintemplateExtWCP.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.
- `wcpdomaintemplateExtWCP.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 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.
- Replace *APPLICATION_HOME* with the complete path to the Application directory for the domain on local 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. If the full path to the packed jar file is on a shared volume available to the other servers, skip this step, otherwise, run the following command on WCCHOST1 to copy the template pack created in step 1 to WCCHOST2, WCPHOST1, and WCPHOST2:

```
scp /full_path/wcpdomaintemplateExtWCP.jar oracle@WCCHOST2:/full_path/
scp /full_path/wcpdomaintemplateExtWCP.jar oracle@WCPHOST1:/full_path/
scp /full_path/wcpdomaintemplateExtWCP.jar oracle@WCPHOST2:/full_path/
```

5. Run the following `unpack` command on WCCHOST2, WCPHOST1, and WCPHOST2 to deploy the domain template copied in the preceding step to the local *MSERVER_HOME* domain directory:

```
cd $ORACLE_COMMON_HOME/common/bin

./unpack.sh -domain=MSERVER_HOME
            -template=/full_path/wcpdomaintemplateExtWCP.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.
- `wcpdomaintemplateExtWCP.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 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.
- Replace *APPLICATION_HOME* with the complete path to the Application directory for the domain on local 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*.

Restoring customizations to setDomainEnv.sh after Unpacking the Domain

If any customizations have been made earlier to the `setDomainEnv.sh` files in *ASERVER_HOME* and *MSERVER_HOME*, then these customizations will need to be repeated after any domain extension.

> **Note**
>
> Modifying the `setDomainEnv` script is not recommended. For more information, see Customizing Domain Wide Server Parameters in *Administering Server Startup and Shutdown for Oracle WebLogic Server*.
>
> For WebCenter Enterprise Deployments, see Customizing Server Parameters with the setUserOverridesLate Script.

Verify that all customizations have been restored before starting NodeManager or WebLogic Server instances.
On WCCHOST1:

1. Verify and update *ASERVER_HOME*`/bin/setDomainEnv.sh`.
2. Verify and update *MSERVER_HOME*`/bin/setDomainEnv.sh`.
3. Copy *MSERVER_HOME*`/bin/setDomainEnv.sh` to the other hosts (Example: WCCHOST2, WCPHOST1, and WCPHOST2).

> **Note**
>
> There are unique differences in parameter values stored in the `ASERVER_HOME` and `MSERVER_HOME` `setDomainEnv.sh` configuration files. The same file cannot be copied into both locations and should be edited separately. *`MSERVER_HOME`*`/bin/setDomainEnv.sh` can be copied across the environment consistently.

Updating the NodeManager Configuration After Unpacking the Domain

When extending a domain, the `nodemanager.properties` file in *MSERVER_HOME* may be overwritten with some values from the `nodemanager.properties` file for *ASERVER_HOME*. Specifically, the `ListenAddress` and/or `CustomIdentityAlias` values can be reset.

> **Notes:**
>
> - The `ListenAddress` may typically get reset on the *MSERVER_HOME* nodemanager residing on the same host as the *ASERVER_HOME* nodemanager. In this topology, WCCHOST1.
> - For domain extensions prior to Enabling SSL Communication Between the SOA Servers and the Hardware Load Balancer, steps 2 through 4 regarding the `CustomIdentityAlias` may not be applicable.

For the *`MSERVER_HOME`*`/nodemanager/nodemanager.properties` file on each host:

1. Verify the correct `ListenAddress` parameter value and reset it, if required.

   ```
   grep ListenAddress MSERVER_HOME/nodemanager/nodemanager.properties
   ```

2. Confirm the list of configured Identity Aliases from the domain configuration file as a reference for the next command.

   ```
   grep server-private-key-alias ASERVER_HOME/config/config.xml | sort | uniq
   ```

   > **Note**
   >
   > Use the appropriate host-specific certificate identity aliases when updating the `nodemanger.properties CustomIdentityAlias` property in the next instruction.

3. Verify the current `nodemanager.properties CustomIdentityAlias` parameter value matches the alias for the host.

   ```
   grep CustomIdentityAlias MSERVER_HOME/nodemanager/nodemanager.properties
   ```

4. Reset the `CustomIdentityAlias` parameter value to the correct alias string appropriate for the current host, if required.

5. Restart the `nodemanager` process:

```
kill `ps -eaf | grep weblogic.NodeManager | grep MSERVER_HOME | grep -v
grep | awk '{print $2}' `
nohup MSERVER_HOME/bin/startNodeManager.sh > MSERVER_HOME/nodemanager/
nodemanager.out 2>&1 &
```

> **Note**
>
> For more information about the `CustomIdentityAlias` parameter, see Configuring Node Manager to Use the Custom Keystores.

Restarting and Validating Pre-existing Managed Servers

Restart the managed servers for the pre-existing components now that the domain has been extended and unpacked to the *MSERVER_HOME* directories on all of the servers.

1. From the Fusion Middleware Control, restart the WLS_WSM*n* Managed Servers for the WebServices Manager Policy Manager.
2. From another browser window, verify the WSM-PM application is responding by successfully loading the URL:

   ```
   https://wcpinternal.example.com:444/wsm-pm
   ```

3. Restart all other pre-existing managed servers before continuing. This can be done in a rolling manner as necessary to avoid application outages.

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 in to the WebLogic Remote Console to access the provider of this domain.
2. In the left navigation tree, expand **Domain**, and then **Environment**.
3. Click **Lock & Edit**.
4. Navigate to and edit the appropriate objects for your cluster type.
   a. For Static Clusters, navigate to **Servers** and click the name of the Managed Server you want to edit.
   b. For Dynamic Clusters, navigate to **Clusters** > **Server Templates**, and click on the name of the server template to be edited.
5. For each new Managed Server or Server Template to be edited:

a. Click the **Configuration** tab, and then click the **Deployment** tab.

b. Verify that the **Staging Directory Name** is set to the following:

   `MSERVER_HOME/servers/server_or_template_name/stage`

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

   If you use static clusters, update with the correct name of the Managed Server that you are editing.

   If you use dynamic clusters, leave the template name intact. For example: `/u02/oracle/config/domains/wcpedg_domain/servers/XYZ-server-template/stage`

c. Update the **Upload Directory Name** to the following value:

   `ASERVER_HOME/servers/AdminServer/upload`

   Replace *ASERVER_HOME* with the directory path for the *ASERVER_HOME* directory.

d. Click **Save**.

e. Return to the Summary of Servers or Summary of Server Templates screen as applicable.

6. Repeat the previous steps for each of the new managed servers.
7. Navigate to and update the Upload Directory Name value for the AdminServer:
   a. Navigate to **Servers**, and select the AdminServer.
   b. Click the **Configuration** tab, and then click the **Deployment** Tab.
   c. Verify that the **Staging Directory Name** is set to the following absolute path:

      `ASERVER_HOME/servers/AdminServer/stage`

   d. Update the **Upload Directory Name** to the following absolute path:

      `ASERVER_HOME/servers/AdminServer/upload`

      Replace *ASERVER_HOME* with the directory path for the *ASERVER_HOME* directory.

   e. Click **Save**.
8. When you have modified all the appropriate objects, click **Activate Changes**.

> **Note**
>
> If you continue directly with further domain configurations, a restart to enable the stage and upload directory changes is not strictly necessary at this time.

Starting the Node Manager on WCPHOST1

After you unpack the extended domain on WCPHOST1, you can start the Node Manager from the Managed Server directory on WCPHOST1.

1. Navigate to the following directory on WCPHOST1:

   `MSERVER_HOME/bin`

2. Use the following command to start the Node Manager:

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

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

Starting the Node Manager on WCPHOST2

After you have propagated the domain configuration to WCPHOST2, you can start the Node Manager for the *MSERVER_HOME* domain directory.

1. If you haven't already, log in to WCPHOST2.
2. Change directory to the following location:

   MSERVER_HOME/bin

3. Use the following command to start the Node Manager on WCPHOST2:

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

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

Starting and Validating the WC_Portal1 and WC_Portlet1 Managed Servers

Now that you have extended the domain, restarted the Administration Server, and propagated the domain to the other hosts, you can start the newly configured Oracle WebCenter Portal servers.

This process involves three tasks:

Starting the Managed Servers on WCPHOST1

To start the WC_Portal1 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_wcp`.
3. In the **Target Navigation** pane, expand the domain to view the Managed Servers in the domain.
4. Select only the **WC_Portal1** Managed Server and click **Start Up** on the Oracle WebLogic Server toolbar.

   > **Note**
   >
   > WebCenter Portal 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 WebCenter Portal servers are started.

5. When the startup operation is complete, navigate to the Domain home page and verify that the WC_Portal1 Managed Server is up and running.

6. Repeat the above steps to start the WC_Portlet1 Managed Server on WCPHOST1.

Adding the WCPAdmin Role to the Portal Administrators Group

Before you validate the Oracle WebCenter Portal configuration on WC_Portal1 Managed Server, grant the WebCenter Portal administrator role to the `WCPAdministrators` LDAP group.

To perform this task, refer to Configuring Roles for Administration of Oracle WebCenter Portal Products.

Granting the Administrator Role for WebCenter Portal Using WLST

This section describes how to grant WebCenter Portal's Administrator role using WLST.

To grant the WebCenter Portal Administrator role using WLST:

1. Create a group in the LDAP store named **WCPAdministrators**.

   This group will be assigned the Administrator role in WebCenter Portal.

   For more information on how to create a user, see Provisioning an Enterprise Deployment Administration User and Group and Adding the Administration Role to the New Administration Group.

2. Navigate to your WebCenter Portal Oracle home directory and invoke the WLST script:

   ```
   (UNIX) ORACLE_HOME/wcportal/common/bin/wlst.sh

   (Windows) ORACLE_HOME\wcportal\common\bin\wlst.cmd
   ```

3. Connect to the Administration Server for the target domain with the following command:

   ```
   wls:/offline>connect("user_name","password", "host_name:port_number")
   ```

4. Grant the WebCenter Portal Administrator application role to the WCPAdministrators group in LDAP using the `grantAppRole` command.

   ```
   grantAppRole(appStripe="webcenter",
   appRoleName="s8bba98ff_4cbb_40b8_beee_296c916a23ed#-#Administrator",
   principalClass="weblogic.security.principal.WLSGroupImpl",
   principalName="WCPAdministrators")
   ```

   Where **WCPAdministrators** is the name of the portal administration group you created earlier.

5. Restart the `WC_Portal1` Managed Server.

   ```
   shutdown('WC_Portal1', block='true', force='true')
   start('WC_Portal1', block='true')
   ```

6. To test the new account, log in to WebCenter Portal using the new account name.

   Open WebCenter Portal in your browser using `http://WCPHOST1:9001/webcenter`. After logging in, the **Administration** link should appear, and you should be able to perform all administrative operations.

Granting the Administrator Role for WebCenter Portal Using Fusion Middleware Control

This section describes how to grant WebCenter Portal's **Administrator** role to a user account other than the default **weblogic** account.

To grant WebCenter Portal's Administrator role using Fusion Middleware Control:

1. Sign-in to the Fusion Middleware Control by using the administrator's account. For example: `weblogic_wcp`.
2. From the **WebLogic Domain** menu, select **Security**, and then **Application Roles**.
3. Search for WebCenter Portal's Administrator role:
   a. Select the webcenter Application Stripe.
   b. Change the search form's Role Name option from **Starts With** to **Includes**.
   c. In the text box next to the **Includes** option, enter `#Administrator`, then click the Search Application Roles icon .
4. Click the returned row to select the Administrators role.
5. Click the Edit icon to open the Edit Application Role view.
6. Click the Add icon .

   The form to add members is displayed.
7. Change the search type from **Application Role** to **Group**.
8. Use the Search function to search for the **WCPAdministrators** LDAP group created earlier in Provisioning an Enterprise Deployment Administration User and Group.
9. Click to select the correct row of search results for the **WCPAdministrators** group.
10. Click **OK** to add the selected group to the role.
11. On the **Edit Application Role** page, verify the updated list of members includes the newly added group.
12. Click **OK** to save the changes to the Application Role.
13. Restart the **WC_Portal1** Managed Server.

    When you log in to WebCenter Portal as a member of the WCPAdministrators group, the Administration link should appear and you should be able to perform all administrative operations.

Enabling SSL Communication Between the WebCenter Portal and Portlet Managed Servers and the Hardware Load Balancer

After you extend the domain with Oracle WebCenter Portal, ensure that the Administration Server and Managed Servers can access the front-end, SSL URL of the hardware load balancer.

This allows applications and web services to invoke callbacks and other communications with the front-end, secure URL.

See Enabling SSL Communication Between the Middle Tier and SSL Endpoints.

Configuring Session Persistence for WebCenter Portal

Applications that are deployed to a cluster in WebLogic Server can optionally take advantage of HTTP session persistence and replication to provide high-availability for the user's experience (their session data) in case of the loss or outage of one or more application server instances within the cluster. This additional session state replication between server instances incurs a performance penalty for the application.

By default, the HTTP session persistence is disabled in WebCenter Portal, and can be enabled if required.

For technical reference, see Using Sessions and Session Persistence in *Developing Web Applications, Servlets, and JSPs for Oracle WebLogic Server*, and HTTP State Replication in *Administering Clusters for Oracle WebLogic Server*.

To enable HTTP session persistence for WebCenter Portal, configure and apply a customized deployment plan as follows:

1. Create a new deployment plan XML file for the webcenter portal application. The deployment plan should be in *DEPLOY_PLAN_HOME* on the shared filesystem available to all hosts. Set the example path/filename to:

   DEPLOY_PLAN_HOME/*DOMAIN_NAME*/webcenterPlan.xml

   > **Note**
   >
   > Substitute the full path to *DEPLOY_PLAN_HOME* and *DOMAIN_NAME* in the <*config-root*> element value.

```
<?xml version='1.0' encoding='UTF-8'?>
<deployment-plan xmlns="http://www.bea.com/ns/weblogic/90"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.bea.com/ns/weblogic/90 http://
www.bea.com/ns/weblogic/90/weblogic-deployment-plan.xsd" global-
variables="false">
<application-name>webcenter</application-name>
<variable-definition>
    <variable>
      <name>addPersistentStore</name>
      <value>replicated_if_clustered</value>
    </variable>
  </variable-definition>

<module-override>
     <module-name>spaces.war</module-name>
     <module-type>war</module-type>
     <module-descriptor external="false">
        <root-element>weblogic-web-app</root-element>
        <uri>WEB-INF/weblogic.xml</uri>
        <variable-assignment>
            <name>addPersistentStore</name>
            <xpath>/weblogic-web-app/session-descriptor/persistent-store-
type</xpath>
            <operation>add</operation>
```

```
        </variable-assignment>
      </module-descriptor>
</module-override>
<config-root>DEPLOY_PLAN_HOME/DOMAIN_NAME/webcenterPlan.xml</config-root>
</deployment-plan>
```

> **Note**
>
> *DEPLOY_PLAN_HOME* and *DOMAIN_NAME* are placeholder substitution values. Provide the actual paths or names as required.

The *DEPLOY_PLAN_HOME* value should be replaced with the full path to a shared filesystem directory as specified in Shared Storage Recommendations When Installing and Configuring an Enterprise Deployment.

2. Sign in to the WebLogic Server Console as an administrative user. For example: `weblogic_wcp`.
3. Click **Deployments** in the Domain Structure panel.
4. Click **Lock & Edit**.
5. Select the checkbox for the webcenter application.
6. Click **Update**.
7. Observe the current **Deployment Plan Path** value as (no value specified).

> **Note**
>
> If a custom deployment plan has previously been specified, then integrate the differences based on the code in step 1 into the existing deployment plan file instead to preserve any pre-existing deployment customizations.

8. Click **Change Path** associated with the Deployment plan path field.
9. Enter the full and complete path to the custom deployment plan XML file, and click **Next**.

   DEPLOY_PLAN_HOME`/`*DOMAIN_NAME*`/webcenterPlan.xml`

10. Click **Finish**. The WebCenter Portal application deployment must be redeployed and cannot be updated in-place. Do not change the selection of the redeploy option.
11. Click **Activate Changes** in the Change Center panel.
12. Restart all managed servers in the `Portal_Cluster`.
13. Validate the updated plan enhancements are listed in the webcenter application deployment section of *ASERVER_HOME*`/config/config.xml`. Look for the <*plan-dir*> and <*plan-path*> elements.

    For example:

```
<app-deployment>
    <name>webcenter</name>
    <target>WC_Portal</target>
    <module-type>ear</module-type>
    <source-path>/u01/oracle/products/fmw/wcportal/archives/applications/
```

```
webcenter.ear</source-path>
    <deployment-order>400</deployment-order>
    <plan-dir xsi:nil="true"></plan-dir>
    <plan-path>/u01/oracle/config/wcpedg_domain/webcenterPlan.xml</plan-
path>
    <security-dd-model>DDOnly</security-dd-model>
    <staging-mode>nostage</staging-mode>
    <parallel-deploy-modules>false</parallel-deploy-modules>
  </app-deployment>
```

Configuring Analytics

In the enterprise deployment reference architecture, analytics collectors are usually configured to communicate with the local WebCenter Portal application in a 1-1 relationship (the collectors listen on localhost on the default port). Additional analytics collector configuration is required if the *Portal_Cluster* will be scaled-up vertically with multiple managed servers per WCPHOST, either manually with traditional cluster configuration or through the use of the WebLogic Server Dynamic Clustering features.

While multiple collectors on the same host can be configured to avoid port conflicts to allow vertical scale-up of Portal and analytics services, there can be only one active common Analytics collector registration in the portal. This active registration can reference only a single hostname and port combination. As such, it remains recommended to configure the WebCenter Portal applications to send event messages to localhost with the default analytics collector port.

> **Note**
>
> Clustered analytics collectors are not supported for collecting WebCenter Portal events.
>
> Additional configuration to avoid analytics collector port conflicts would be required if scaling-up more than one portal managed server per host. In this case, for more information about configuring collector maximum port values, see Configuring Analytics Collector Settings in the *Administering Oracle WebCenter Portal* guide.
>
> Portal run-time configuration only supports a single active Analytics collector service registration with a single port. Only the collector started per host using the port matching the portal registration will collect data. If that collector or managed server is down, Analytics event collection will not occur for any of the remaining portal servers on that host.
>
> Use of the local hardware load balancer to centrally distribute analytics traffic across co-located collectors in scale-up scenarios has not been validated as of this Enterprise Deployment Guide release. Use of this potential topology option should be thoroughly tested in a non-production environment before finalizing your architecture in this regard.

Registering a Default Analytics Connection for WebCenter Portal

Connect the WebCenter Portal application to the analytics collector.

To connect the WebCenter Portal application to the analytics collector, complete the following steps:

1. Connect to the Administration Server for the domain using WLST as the weblogic_wcp user, for example:

   ```
   ORACLE_COMMON_HOME/common/bin/wlst.sh
   connect("weblogic_admin_username", "weblogic_admin_pwd", "t3s://
   ADMINVHN:9002")
   ```

2. Create a connection between the WebCenter Portal application and the Analytics Collector and make it the default connection (default=1).

   > **Note**
   >
   > The commands in this section need to be executed only once for the clustered Webcenter application, even though there is a specific server name listed for the connection. When this configuration is set, it applies to all servers.

   For example:

   ```
   createAnalyticsCollectorConnection(appName="webcenter",
   server="WC_Portal1", connectionName="Collector31314", collectorPort=31314,
   isEnabled=1, default=1, isUnicast=1, collectorHost="localhost", timeout=30)
   ```

   See also, the createAnalyticsCollectorConnection section in the *WebLogic Scripting Tool Command Reference*.

3. Verify the changes made:

   ```
   listDefaultAnalyticsCollectorConnection(appName="webcenter", server="WC_Portal1")
   ```

   For example:

   ```
   wls:/wcpedg_domain/serverConfig/>
   listDefaultAnalyticsCollectorConnection(appName="webcenter",
   server="WC_Portal1")
   --------------
   Collector31314
   --------------
   Cluster Name / Host Name: localhost
   Port: 31314
   Timeout: 30
   Unicast: true
   Enabled: true
   ```

   See also the listDefaultAnalyticsCollectorConnection section in the *WebLogic Scripting Tool Command Reference*.

Configuring Analytics to Support Scale-Up of the Portal Managed Servers

Additional configuration and management processes may be needed for Analytics if Portal managed servers are to be scaled-up beyond one managed server per host. Scale-up

operations may be manual in the case of traditional static clusters. Scale-up may be manual or elastic (policy-based) when using Dynamic Cluster features of the WebLogic Server.

> **Note**
>
> This section applies only when there are multiple portal managed servers per host.
>
> If the topology follows the default enterprise deployment guide reference architecture with only one portal managed server per host, or is enhanced to include only horizontal scale-out scenarios, then this section does not apply and can be skipped.

Two additional configurations must be addressed to support scale-up:

1. The Analytics Collector Port range must be set to allow multiple collector services to run on a single host and listen on separate unique ports.
2. WebCenter Portal registrations for the additional collectors will need to be added. These registrations can be quickly activated later and portal servers restarted if the Analytics Collector on the default port is unavailable. Once restarted, the portal servers will attempt to use the newly activated collector registration.

The requirements for reliable Analytics high-availability, are as follows:

- There must be an Analytics collector process listening on the port specified in the default and active registration within the portal.
- There must be an Analytics collector process available on each port within the configured Analytics Collector port range on every WCPHOST.
- Every WCPHOST must have an identical number of scaled-up portal managed servers, otherwise some portal servers may not find the collector on the currently default/active port on localhost for the selected registration.

The following sections advise on the additional configuration required for analytics when accommodating for scale-up:

Configuring the Analytics Collector Port Range

The Analytics collector settings are environment-wide and cannot be customized on a server-by-server basis. The maxPort parameter provides an effective range above the defaultPort for multiple Analytics Collector instances to use unique ports on a per-host basis. If a collector process cannot bind to the default port, it retries every port number until it reaches the maxPort value. If no ports are available, the portal managed server will not start correctly.

To update the Analytics maxPort value, use the following WLST method when connected to the administration server for the domain:

1. Connect to the Administration Server for the domain using WLST, for example:

   ```
   ORACLE_COMMON_HOME/common/bin/wlst.sh
   connect("weblogic_admin_username", "weblogic_admin_pwd", "t3s://
   ADMINVHN:9002")
   ```

2. List the current Analytics Collector configuration with the following WLST command:

   ```
   listAnalyticsCollectorConfig(appName='analytics-collector',
   server='WC_Portal1')
   ```

> **Note**
>
> - Verify the Collector Default Port and the Collector Maximum Port assignments. Out-of-the-box, they should be the same port number.
> - Cluster values should be ignored.
> - Analytics clustering is not supported and can cause faults.

For example:

```
wls:/wcpedg_domain/serverConfig/>
listAnalyticsCollectorConfig(appName='analytics-collector',
server='WC_Portal1')
Location changed to domainRuntime tree. This is a read-only tree with
DomainMBean as the root MBean.
For more help, use help('domainRuntime')

Collector Hostname = localhost
Collector Default Port = 31314
Collector Maximum Port = 31314
Broadcast Type = Multicast
Collector Heartbeat Frequency = 10
Cluster Enabled? = 0
Cluster Name =
```

3. Calculate the required new Collector Maximum Port number.

   Consider the number of portal managed servers that you decide are needed for scalability, Then round up that number to an even multiple of the number of `WCPHOST`*n* machines you have assigned in the environment. This value is the total number of possible collector instances per `WCPHOST`. The actual number of portal managed servers with collectors may be less than this. In this case, the maximum port number must still accommodate the highest number of co-located Analytics collectors or portal managed servers. Subtract one from this multiple and add to the Analytics Collector Default Port number to calculate the Analytics Collector maxPort parameter value.

   Example Scenario:

   - Environmental Requirements/Facts:
     - SLA allows for degraded performance (fewer servers) upon host failure
     - Number of Portal managed servers needed for scalability/performance: 5
     - Number of WCPHOST machines purchased: 2
     - Default Analytics Collector port not changed (31314)
     - Portal and Analytics Collector deployed to same managed server (WCP 12.2.1 out-of-box topology)
   - Analysis:
     - Derive max servers per host by evenly rounding up the number of servers needed divided by number of hosts:

       ROUNDUP(5/2) = 3 servers per host (max @ required scalability)
     - Must support an analytics collector port range of 3 collectors per WCPHOST.

- Analytics Collector maxPort: 31316

  defaultPort +(#ofPortalSvrsPerHost) - 1 = (31314+3-1)

4. Update the Collector configuration settings to set the **maxPort** value.

   The **maxPort** value is the incremented port number greater than or equal to the Collector Default Port.

   > **Note**
   >
   > The commands in this section need to be executed only once for the clustered WebCenter application, even though there is a specific server name listed for the connection. When this configuration is set, it applies to all servers.

   For example:

   ```
   setAnalyticsCollectorConfig(appName='analytics-collector', server='WC_Portal1',
   maxPort=31316)
   ```

   See also, the createAnalyticsCollectorConnection section in the *WebLogic Scripting Tool Command Reference*.

5. Verify the change to the Collector Maximum Port value using the `listAnalyticsCollectorConfig()` WLST method.

   For example:

   ```
   listAnalyticsCollectorConfig(appName='analytics-collector', server='WC_Portal1')

   Collector Hostname  =  localhost
   Collector Default Port  =  31314
   Collector Maximum Port  =  31316
   Broadcast Type  =  Multicast
   Collector Heartbeat Frequency  =  10
   Cluster Enabled?  =  0
   Cluster Name  =
   ```

   See also, the listDefaultAnalyticsCollectorConnection section in the *WebLogic Scripting Tool Command Reference*.

6. Restart the portal servers after completing Registering additional Analytics Collectors in WebCenter Portal.

Registering additional Analytics Collectors in WebCenter Portal

For each port in the configured Analytics Collector port range, register additional non-default collectors with unique connection names and ports based on the port-range assigned in the Analytics Collector configuration. This will allow for a simplified maintenance process to swap collector registrations in a persistent, well-qualified manner.

For example:

```
createAnalyticsCollectorConnection(appName="webcenter", server="WC_Portal1",
connectionName="NAME", collectorPort=PORTNUMBER, isEnabled=1, default=0,
isUnicast=1, collectorHost="localhost", timeout=30)
```

Complete the following steps:

1. Connect to the Administration Server for the domain using WLST, for example:

   ```
   ORACLE_COMMON_HOME/common/bin/wlst.sh
   connect("weblogic_admin_username", "weblogic_admin_pwd", "t3s://
   ADMINVHN:9002")
   ```

2. List the current Analytics Collector configuration with the following WLST command:

   ```
   listAnalyticsCollectorConfig(appName='analytics-collector',
   server='WC_Portal1')
   ```

   For example:

   ```
   Collector Hostname  =  localhost
   Collector Default Port  =  31314
   Collector Maximum Port  =  31316
   Broadcast Type  =  Multicast
   Collector Heartbeat Frequency  =  10
   Cluster Enabled?  =  0
   Cluster Name  =
   ```

   > **Note**
   >
   > You will need to repeat the next step for every port number within the default-to-maximum port range, including the maximum port number. The registration for the default port should have already been completed earlier.

3. Add registrations through WLST for each port above the default port up to the maximum port number, changing the **connectionName** and **collectorPort** values for each registration.

   For example:

   ```
   createAnalyticsCollectorConnection(appName="webcenter",
   server="WC_Portal1", connectionName="Collector31315", collectorPort=31315,
   isEnabled=1, default=0, isUnicast=1, collectorHost="localhost", timeout=30)

   createAnalyticsCollectorConnection(appName="webcenter",
   server="WC_Portal1", connectionName="Collector31316", collectorPort=31316,
   isEnabled=1, default=0, isUnicast=1, collectorHost="localhost", timeout=30)
   ```

4. List and verify the details for all of the Analytics registrations by using WLST.

   ```
   listAnalyticsCollectorConnections(appName="webcenter", server="WC_Portal1")

   ------------------
   Collector31314
   ------------------
   Cluster Name / Host Name: localhost
   Port: 31314
   Timeout: 30
   Unicast: true
   Enabled: true
   ------------------
   ```

```
Collector31315
------------------
Cluster Name / Host Name: localhost
Port: 31315
Timeout: 30
Unicast: true
Enabled: true
------------------
Collector31316
------------------
Cluster Name / Host Name: localhost
Port: 31316
Timeout: 30
Unicast: true
Enabled: true
```

5. List the details for the default (currently active) Analytics registrations by using WLST.

   For example:

```
listDefaultAnalyticsCollectorConnection(appName="webcenter", server="WC_Portal1")

------------------
Collector31314
------------------
Cluster Name / Host Name: localhost
Port: 31314
Timeout: 30
Unicast: true
Enabled: true
```

> **Note**
>
> This example indicates Port: 31314, the default port. Multiple registrations have been made, but the portal's Analytics registration has not been failed-over to an alternate connection/port yet. The first portal managed server to be started per machine will have the Analytics Collector listening on the default port.

6. Restart all WC_Portal*n* managed servers for the changes to take effect. This can be all-at-once or done in a rolling fashion.

   Example - all at once:

```
shutdown('Portal_Cluster', 'Cluster', force='true', block='true')
start('Portal_Cluster',  'Cluster', block='true')
```

   Example - rolling, no outage:

```
shutdown('WC_Portal1', force='true', block='true')
start('WC_Portal1', block='true')

shutdown('WC_Portal2', force='true', block='true')
start('WC_Portal2', block='true')
```

Failing-over the Portal's Default Analytics Registration

In the case of a failure of a portal managed server that includes the Analytics Collector service listening on the default port, an alternate analytics connection should be selected in the portal configuration as the default connection and the surviving portal servers restarted as soon as possible.

Until this occurs, analytics collection for the remaining portal instances on that machine will not be able to contact the collector on the default port to log analytics metrics. Only one registration is default and actively used at a time.

The fail-over of the default analytics registration is a manual process. The change will be environment-wide but will only take effect when each portal server instance is restarted.

> **Note**
>
> This section is applicable only when there are multiple portal managed servers per host.
>
> This sections is not applicable if the topology follows the default enterprise deployment guide reference architecture with only one portal managed server per host or is enhanced to include only horizontal scale-out scenarios.

To change the default Analytics registration, complete the following steps:

1. Connect to the Administration Server for the domain using WLST, for example:

   ```
   ORACLE_COMMON_HOME/common/bin/wlst.sh
   connect("weblogic_admin_username", "weblogic_admin_pwd", "t3s://
   ADMINVHN:9002")
   ```

2. List the details for the default (currently active) Analytics registrations by using WLST:

   ```
   listDefaultAnalyticsCollectorConnection(appName="webcenter",
   server="WC_Portal1")

   ------------------
   Collector31314
   ------------------
   Cluster Name / Host Name: localhost
   Port: 31314
   Timeout: 30
   Unicast: true
   Enabled: true
   ```

3. List the available Analytics registrations via WLST, and choose a different collector name to set as default.

   ```
   listAnalyticsCollectorConnections(appName="webcenter", server="WC_Portal1")

   ------------------
   Collector31314
   ------------------
   ```

```
Cluster Name / Host Name: localhost
Port: 31314
Timeout: 30
Unicast: true
Enabled: true
------------------
Collector31315
------------------
Cluster Name / Host Name: localhost
Port: 31315
Timeout: 30
Unicast: true
Enabled: true
------------------
Collector31316
------------------
Cluster Name / Host Name: localhost
Port: 31316
Timeout: 30
Unicast: true
Enabled: true
```

4. Update the default Analytics registration by using WLST.

```
setDefaultAnalyticsCollectorConnection(appName="webcenter",
name="Collector31315", server="WC_Portal1")
```

5. Restart all WC_Portal*n* managed servers for the changes to take effect. This can be all-at-once or done in a rolling fashion.

   Example - all at once:

```
shutdown('Portal_Cluster', 'Cluster', force='true', block='true')
start('Portal_Cluster', 'Cluster', block='true')
```

   Example - rolling, no outage:

```
shutdown('WC_Portal1', force='true', block='true')
start('WC_Portal1', block='true')

shutdown('WC_Portal2', force='true', block='true')
start('WC_Portal2', block='true')
```

Confirming REST API Configuration

This section describes the procedure to configure REST APIs.

With release 12.2.1.0, the WebCenter REST APIs are pre-configured with the credential map configuration.
If you want to confirm the seed entries in the credential store that enable the REST security tokens to function properly, run the following WLST commands:

> **Note**
>
> If the credential maps already exist, JPS-01007 exceptions messages will be returned. This confirms the configuration.

1. Connect to the Administration Server using the command-line Oracle WebLogic Scripting Tool (WLST), for example:

```
ORACLE_COMMON_HOME/common/bin/wlst.sh
connect('weblogic_admin_username','weblogic_admin_pwd','t3s://
ADMINVHN:9002')
```

2. Run the following WLST commands to configure the credential store:

```
createCred(map="o.webcenter.jf.csf.map", key="keygen.algorithm",
     user="keygen.algorithm", password="AES")
createCred(map="o.webcenter.jf.csf.map", key="cipher.transformation",
     user="cipher.transformation", password="AES/CBC/PKCS5Padding")
```

Configuring the Web Tier for the Extended Domain

The following sections describe how to configure the Oracle HTTP Server instances so they route requests for both public and internal URLs to the proper clusters in the enterprise topology.

Configuring Oracle HTTP Server for the Oracle WebCenter Portal Clusters

To configure the Oracle HTTP Server instances in the Web tier so they route requests correctly to the Oracle WebCenter Portal clusters, use the following procedure to create an additional Oracle HTTP Server configuration file that creates and defines the parameters of the `wcp.example.com` virtual server.

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

Complete the following steps to update the virtual host configuration file so that requests are routed properly:

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 `wcp_vh.conf` file and add the following directives near the end of the file just above the last line: `<VirtualHost>`:

> **Note**
>
> Configure the port numbers appropriately as assigned for your static or dynamic cluster. Dynamic clusters with the Calculate Listen Port option selected will have incremental port numbers for each dynamic managed server created.
>
> The WebLogicCluster directive needs only a sufficient number of redundant *server*:*port* combinations to guarantee initial contact in case of a partial outage. The actual total list of cluster members is retrieved automatically upon first contact with any given node.

```
# WebCenter Portal Application (previously called Spaces)
<Location /webcenter>
    WLSRequest ON
    WebLogicCluster WCPHOST1:8002,WCPHOST2:8002
    WLProxySSL ON
    WLProxySSLPassThrough ON
</Location>

<Location /webcenterhelp>
    WLSRequest ON
    WebLogicCluster WCPHOST1:8002,WCPHOST2:8002
    WLProxySSL ON
    WLProxySSLPassThrough ON
</Location>

<Location /rss>
    WLSRequest ON
    WebLogicCluster WCPHOST1:8002,WCPHOST2:8002
    WLProxySSL ON
    WLProxySSLPassThrough ON
</Location>

<Location /rest>
    WLSRequest ON
    WebLogicCluster WCPHOST1:8002,WCPHOST2:8002
    WLProxySSL ON
    WLProxySSLPassThrough ON
</Location>

<Location /portalTools>
    WLSRequest ON
    WebLogicCluster WCPHOST1:8003,WCPHOST2:8003
    WLProxySSL ON
    WLProxySSLPassThrough ON
</Location>

<Location /wsrp-tools>
    WLSRequest ON
    WebLogicCluster WCPHOST1:8003,WCPHOST2:8003
    WLProxySSL ON
    WLProxySSLPassThrough ON
</Location>

</VirtualHost>
```

3. Copy the `wcp_vh.conf` file into the configuration directory for the second Oracle HTTP Server instance (ohs2) on WEBHOST2:

```
scp WEB_DOMAIN_HOME/config/fmwconfig/components/OHS/ohs1/moduleconf/wcp_vh.conf \
WEBHOST2:/WEB_DOMAIN_HOME/config/fmwconfig/components/OHS/ohs2/moduleconf/wcp_vh.conf
```

4. Log in to WEBHOST2 and change directory to the configuration directory for the second Oracle HTTP Server instance (ohs2):

   ```
   cd WEB_DOMAIN_HOME/config/fmwconfig/components/OHS/ohs2/moduleconf/
   ```

5. Edit the `wcp_vh.conf` and change any references to WEBHOST1 to WEBHOST2 in the `<VirtualHost>` directives.
6. Restart both Oracle HTTP servers.

Validating the Oracle WebCenter Portal Public Services 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 public service requests through the Oracle HTTP Server instances to the application tier:

1. Verify that the server status is reported as **Running** in the WebLogic Remote Console or in the Fusion Middleware Control.

   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://wcp.example.com/webcenter
   - https://wcp.example.com/webcenterhelp
   - https://wcp.example.com/rss
   - https://wcp.example.com/rest/api/resourceIndex
   - https://wcp.example.com/portalTools
   - https://wcp.example.com/wsrp-tools/info

Configuring HTTP Server for Internal WebCenter Services

To configure the Oracle HTTP Server instances in the Web tier so they route requests correctly to the internal Oracle WebCenter services, use the following procedure to edit existing `wcpinternal_vh.conf` file.

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

Complete the following steps to update the virtual host configuration file so that requests are routed properly:

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 `wcpinternal_vh.conf` file and add the following directives near the end of the file just above the last line: `</VirtualHost>`

> **Note**
>
> Configure the port numbers appropriately as assigned for your static or dynamic cluster. Dynamic clusters with the Calculate Listen Port option selected will have incremental port numbers for each dynamic managed server created.
>
> The WebLogicCluster directive needs only a sufficient number of redundant *server*:*port* combinations to guarantee initial contact in case of a partial outage. The actual total list of cluster members is retrieve automatically upon first contact with any given node.

```
# WebCenter Portal Application Services

<Location /webcenter>
    WLSRequest ON
    WebLogicCluster WCPHOST1:8002,WCPHOST2:8002
    WLProxySSL OFF
    WLProxySSLPassThrough OFF
</Location>

<Location /webcenterhelp>
    WLSRequest ON
    WebLogicCluster WCPHOST1:8002,WCPHOST2:8002
    WLProxySSL OFF
    WLProxySSLPassThrough OFF
</Location>

<Location /rss>
    WLSRequest ON
    WebLogicCluster WCPHOST1:8002,WCPHOST2:8002
    WLProxySSL OFF
    WLProxySSLPassThrough OFF
</Location>

<Location /rsscrawl>
    WLSRequest ON
    WebLogicCluster WCPHOST1:8002,WCPHOST2:8002
    WLProxySSL OFF
    WLProxySSLPassThrough OFF
</Location>

<Location /sesUserAuth>
    WLSRequest ON
    WebLogicCluster WCPHOST1:8002,WCPHOST2:8002
    WLProxySSL OFF
    WLProxySSLPassThrough OFF
</Location>

<Location /rest>
    WLSRequest ON
    WebLogicCluster WCPHOST1:8002,WCPHOST2:8002
    WLProxySSL OFF
    WLProxySSLPassThrough OFF
</Location>

# Portlets
<Location /pagelets>
    WLSRequest ON
```

```
    WebLogicCluster WCPHOST1:8003,WCPHOST2:8003
    WLProxySSL OFF
    WLProxySSLPassThrough OFF
</Location>

<Location /portalTools>
    WLSRequest ON
    WebLogicCluster WCPHOST1:8003,WCPHOST2:8003
    WLProxySSL OFF
    WLProxySSLPassThrough OFF
</Location>

<Location /wsrp-tools>
    WLSRequest ON
    WebLogicCluster WCPHOST1:8003,WCPHOST2:8003
    WLProxySSL OFF
    WLProxySSLPassThrough OFF
</Location>

# Collector
<Location /collector>
    WLSRequest ON
    WebLogicCluster WCPHOST1:8002,WCPHOST2:8002
    WLProxySSL OFF
    WLProxySSLPassThrough OFF
</Location>

</VirtualHost>
```

3. Copy the `wcpinternal_vh.conf` file into the configuration directory for the second Oracle HTTP Server instance (ohs2) on WEBHOST2:

   ```
   scp WEB_DOMAIN_HOME/config/fmwconfig/components/OHS/ohs1/moduleconf/
   wcpinternal_vh.conf \
   WEBHOST2:/WEB_DOMAIN_HOME/config/fmwconfig/components/OHS/ohs2/moduleconf/
   wcpinternal_vh.conf
   ```

4. Log in to WEBHOST2 and change directory to the configuration directory for the second Oracle HTTP Server instance (ohs2):

   ```
   cd WEB_DOMAIN_HOME/config/fmwconfig/components/OHS/ohs2/moduleconf/
   ```

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

6. Restart both Oracle HTTP servers.

Validating the Oracle WebCenter Portal Internal Services 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 internal service requests through the Oracle HTTP Server instances to the application tier:

1. Verify that the server status is reported as **Running** in the WebLogic Remote Console or in the Fusion Middleware Control.

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

2. Verify that you can access these URLs:
   - http://wcpinternal.example.com/webcenter
   - http://wcpinternal.example.com/webcenterhelp
   - http://wcpinternal.example.com/rss
   - http://wcpinternal.example.com/rest/api/resourceIndex
   - http://wcpinternal.example.com/pagelets
   - http://wcpinternal.example.com/portalTools
   - http://wcpinternal.example.com/wsrp-tools
   - http://wcpinternal.example.com/collector

Configuring WebCenter Portal for External Services

This section describes how to configure external tools and services for WebCenter Portal applications by using Fusion Middleware Control or WLST commands. For most external services, you must set up a connection between the WebCenter Portal application and the backend server.

Each of the following service configurations require various Managed Servers to be restarted for the changes take effect. Some of the restarts are required at the point during the process as indicated. Other restarts may be optionally delayed to the end of the section if completing this section from start-to-finish, otherwise all restarts should be done within the individual subtopics. Restarts that may be delayed to the end of the service configurations section are mentioned as a note.

Configuring Default Web Service Policies for WebCenter Portlet Producer Applications

After installing Oracle WebCenter Portal, you must attach the default Oracle Web Services Manager (OWSM) security policy to the following:

- WSRP Tools Producer (wsrp-tools)

These steps are required because security policies for these Web service end points are not configured out-of-the-box.

To attach the default Web service security policy:

1. Ensure that `WC_Portal1`, `WC_Portal2`, `WC_Portlet1`, and `WC_Portlet2` managed servers are up and running.
2. Start the WebLogic Scripting Tool:

   ```
   WCPHOST1> ORACLE_COMMON_HOME/common/bin/wlst.sh
   ```

3. Connect to the Administration Server as an administrator.

   For example

   ```
   connect("weblogic_wcp","admin password","t3://ADMINVHN:7001")
   ```

   For information, see the Running Oracle WebLogic Scripting Tool (WLST) Commands section in *Administering Oracle WebCenter Portal*.

4. Run WLST commands to attach the default OWSM security policy (oracle/wss10_saml_token_service_policy) to each of the following:
   - WSRP Tools Producer (wsrp-tools)

     > **Note**
     >
     > In the examples in this topic, configure the application parameter value with the domain name and managed server names for your environment.
     >
     > For more information on attaching WebService policies, see Web Services Custom WLST Commands in Oracle Fusion Middleware WLST Command Reference for Infrastructure Components.

   - Run the following WLST commands to attach the default OWSM security policy to the WSRP Tools Producer's Web service end point. One execution will configure the policy for all servers in the cluster (execute against `WC_Portlet1`). Set your domain name accordingly in the following command:

     ```
     attachWebServicePolicy(application='/domain_name/WC_Portlet1/wsrp-
     tools', moduleName='wsrp-tools', moduleType='web',
     serviceName='WSRP_v2_Service', subjectName='WSRP_v2_Markup_Service',
     policyURI='oracle/wss10_saml_token_service_policy')
     ```

     For more information on attaching WebService policies, see Web Services Custom WLST Commands in Oracle Fusion Middleware WLST Command Reference for Infrastructure Components the link directly to the attachWebServicePolicy() reference.

     For more information on which WS-Security policy to choose here, see WSRP Producer Security Connection Parameters in *Administering Oracle WebCenter Portal* . The same token policy attached here will also need to be used consistently in the next section when registering the portlet producers.

5. Restart all managed servers in the Portal and Portlet clusters.

   > **Note**
   >
   > These restarts should be completed now, before the service connections are configured.

Registering Portlet Producers

Several out-of-the-box portlet producers can be registered with the WebCenter Portal application. In the WebCenter Portal Enterprise Deployment, the required producer URLs are as follows:

- WSRP Producer URL: **http://wcpinternal.example.com/wsrp-tools/portlets/wsrp2?WSDL**
- WebClipping Producer URL: **http://wcpinternal.example.com/portalTools/webClipping/providers**

- OmniPortlet Producer URL: **http://wcpinternal.example.com/portalTools/omniPortlet/providers**

You can register portlet producers using Fusion Middleware Control or WLST commands.

Registering Out-of-the-Box Portlet Producers using Fusion Middleware Control

For details on how to register portlet producers using Fusion Middleware Control, see Managing Portlet Producers in *Administering Oracle WebCenter Portal*.

Registering Out-of-the-Box Portlet Producers Using WLST

To register out-of-the-box portlet producers using WLST:

1. Start the WebLogic Scripting Tool:

   ```
   WCPHOST1> ORACLE_HOME/oracle_common/common/bin/wlst.sh
   ```

2. In WLST, connect as the administrator.
3. Register the out-of-the-box OmniPortlet and WSRP portlet producers.

   For example:

   ```
   registerOOTBProducers(producerHost='wcpinternal.example.com',producerPort=4
   43, appName='webcenter', server='WC_Portal1')
   ```

   Where:

   - The `producerHost` value is set to the internal load-balancer fully-qualified domain name.
   - The `server` value is the name of the first portal managed server.
   - The `appName` value is `webcenter`, the name of the WebCenter Portal application.

   See also, registerOOTBProducers in the *WebCenter WLST Command Reference*.
4. List the new WSRP Producer to display the WSDL URL needed for the next step.

   ```
   listWSRPProducers(appName='webcenter', server='WC_Portal1')
   ```

5. Configure the WSRP Producer service for the WS-Security Policy attached earlier in the Configuring Default Web Service Policies for WebCenter Portlet Producer Applications.
   - Update the `url` parameter based on the WSRP URL displayed in the previous step.
   - Update the `tokenType` parameter with the appropriate value for the needed policy as provided for the setWSRPProducer() tokenType details in *WebCenter WLST Command Reference*.

     ```
     setWSRPProducer(appName='webcenter', server='WC_Portal1', name='wc-
     WSRPTools', url='http://wcpinternal.exaple.com:80/wsrp-tools/portlets/
     wsrp2?WSDL', enforcePolicyURI='1', tokenType='WSS10_SAML_TOKEN_ONLY')
     ```

> **Note**
>
> The `tokenType` parameter value must be the appropriate match with the WS-Security policy attached in the previous section.

6. List the WSRP Producer to confirm updated TokenType configuration.

```
listWSRPProducers(appName='webcenter', server='WC_Portal1')
```

7. Restart all servers in the Portal_Cluster.

> **Note**
>
> If additional services are being configured, the portal restart can be done once at the end of the section.

Registering the Pagelet Producer

If you want to expose WSRP and Oracle JPDK portlets and OpenSocial gadgets as pagelets in WebCenter Portal, you must register the pagelet producer. In the WebCenter Portal Enterprise Deployment, the required pagelet producer URL is:

https://wcpinternal.example.com/pagelets

You can register the pagelet producer using Fusion Middleware Control or WLST commands.

Registering Pagelet Producer Using Fusion Middleware Control

To register Pagelet Producer using Fusion Middleware Control:

1. Sign-in to the Fusion Middleware Control by using the administrator's account (for example: `weblogic_wcp`), and navigate to the WebCenter Portal home page.

> **Note**
>
> When administering WebCenter resources in the Enterprise Manager Fusion Middleware Control, it is recommended to use the WebCenter-specific administrative user created in the remote LDAP authenticator (for example, `weblogic_wcp`). See Configuring Roles for Administration of an Enterprise Deployment.

   See Navigating to the Home page for WebCenter Portal in *Administering Oracle WebCenter Portal* .

2. From the **WebCenter Portal** menu, select **Register Producer**.
3. Enter connection details for Pagelet Producer, as shown in the following table.

Field	Description
Connection Name	A unique name to identify this Pagelet Producer instance within the application. The name must be unique across all WebCenter Portal connection types. The name specified here appears in Composer under the UI Components > Pagelet Producers folder (by default).
Producer Type	Select **Pagelet Producer**.

Server URL	The URL to Pagelet Producer. The URL must include a fully-qualified domain name. Use the following syntax:
	`<protocol>://<host_name>:<port_number>/pagelets/`
	For example:
	`http://wcpinternal.example.com/pagelets/`
	If pagelets contain secure data, the registered URL must use the `https` protocol. For example:
	`https://wcp.example.com/pagelets/`
	The context root can be changed from /pagelets/ if necessary; for details, see “Redeploying Pagelet Producer to a Different Context” in *Administering Oracle WebCenter Portal*.
	Note: In WebCenter Portal, if the Pagelet Producer URL is protected by OAM, the URL to the pagelet catalog must be excluded (mapped directly without access control), or the catalog will appear to be empty when using REST. The pagelet catalog URL is `http://<host_name>:<port_number>/pagelets/api/v2/ensemble/pagelets`

4. Click **OK**. The new producer appears in the connection table.

Registering Pagelet Producer Using WLST

To register out-of-the-box pagelet producers using WLST:

1. Start the WebLogic Scripting Tool:

   ```
   WCPHOST1> ORACLE_COMMON_HOME/common/bin/wlst.sh
   ```

2. In WLST, connect as the administrator.
3. Register the pagelet producer by entering the following command.

   ```
   registerPageletProducer(appName='webcenter', name='PageletProducer',
   url='https://wcpinternal.example.com/pagelets', server='WC_Portal1')
   ```

For command syntax and examples, see `registerPageletProducer` in *WebCenter WLST Command Reference*.

You can also use WLST to list or edit the current connection details.

For information on how to run WLST commands, see Running Oracle WebLogic Scripting Tool (WLST) Commands in *Administering Oracle WebCenter Portal*.

Configuring Search Services

You can configure Oracle Secure Enterprise Search (Oracle SES) services and crawlers using procedures in the Managing Oracle Secure Enterprise Search in WebCenter Portal section in *Administering Oracle WebCenter Portal*.

> **Note**
>
> WebCenter Portal provides basic integration with Elasticsearch. The configuration of a robust highly-available Elasticsearch deployment with the WebCenter Enterprise Deployment topology has not been fully tested as of this release. For basic configuration, see Managing Search in WebCenter Portal with Elasticsearch in *Administering Oracle WebCenter Portal*. Significant additional analysis and configuration related to clustering, HA, and security/accessibility is needed for use a WebCenter Enterprise Deployment Guide topology.

Ensure that:

- Oracle Secure Enterprise Search is registered with Oracle Internet Directory and the WebCenter Portal application is configured as an Oracle SES trusted entity, as described in the Managing Oracle Secure Enterprise Search in WebCenter Portal section in *Administering Oracle WebCenter Portal*.
- Connection exists between the WebCenter Portal application and Oracle Secure Enterprise Search, as described in the Setting Up Oracle SES Connections section in *Administering Oracle WebCenter Portal*.

Add the following additional URL location paths for the `wcpinternal.example.com` virtual host in the `wcinternal_vh.conf` file, and then restart each OHS service.

```
<Location /rsscrawl>
    WLSRequest ON
    WebLogicCluster WCPHOST1:8002,WCPHOST2:8002
    WLProxySSL OFF
    WLProxySSLPassThrough OFF
</Location>

<Location /sesUserAuth>
    WLSRequest ON
    WebLogicCluster WCPHOST1:8002,WCPHOST2:8002
    WLProxySSL OFF
    WLProxySSLPassThrough OFF
</Location>
```

Configuring Oracle WebCenter Portal Notifications for the SMTP Mail Server

In a WebCenter Portal Enterprise Deployment, if you choose to send notifications using mail, you must configure a connection to your corporate mail server and specify several unique parameters for the sent emails to appear correctly.

To ensure sufficient configuration for your mail server and business requirements, before completing this task, review Managing Mail in *Administering Oracle WebCenter Portal* for details on the required and optional configurations and parameters.

You can register a mail server using Fusion Middleware Control or WLST commands:

Registering Mail Servers Using Fusion Middleware Control

For details on how to register a mail server using Fusion Middleware Control, see Registering Mail Servers Using Fusion Middleware Control in *Administering Oracle WebCenter Portal*.

Registering Mail Servers Using WLST

Use the WLST command `createMailConnection` to create a mail server connection.

1. Start the WebLogic Scripting Tool:

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

2. Connect to the Administration Server.
3. Register an External Application Connection for the mail server:

   ```
   createMailExtAppConnection(appName='webcenter', name='CorpMailServer',
   displayName='Corporate Mail Server', server='WC_Portal1')
   ```

   > **Note**
   >
   > Ignore the warning: `Another application named "webcenter" exists....`

4. Register a Mail Server Connection:

   ```
   createMailConnection(appName='webcenter', name='NotificationSharedConn',
   smtpHost='mail.example.com', smtpPort=25, smtpSecured=0,
   timeout=10, default=1,appId='CorpMailServer', server='WC_Portal1')
   ```

   > **Note**
   >
   > Additional capabilities are available and might be required depending on your mail server and business needs, such as the ability to notify distribution lists. See Managing Mail in *Administering Oracle WebCenter Portal*.

5. Set the required mail server connection properties.

   These properties ensure that a specific mail address is the same in the external application and in the mail server. These properties are added to the mail connection and are used by mail for the **From**, **Display Name**, and **Reply To** fields.

   For example:

   ```
   setMailConnectionProperty(appName='webcenter',
   name='NotificationSharedConn', key='mail.user.emailAddress',
   value='john.doe@example.com')

   setMailConnectionProperty(appName='webcenter',
   name='NotificationSharedConn', key='mail.user.displayName', value='John
   ```

```
Doe')

setMailConnectionProperty(appName='webcenter',
name='NotificationSharedConn', key='mail.user.replyToAddress',
value='feedback@example.com')
```

6. **For Exchange 2007 only**, create a universal distribution list. To do this, you must update the value of the `mail.exchange.dl.group.type` property from 2 (default value) to 8.

   Specify a value of 8 for the `mail.exchange.dl.group.type` mail property, as follows:

   ```
   setMailServiceProperty(appName='webcenter',
   property='mail.exchange.dl.group.type', value='8')
   ```

   If your application offers a self-registration page with the facility to mail user ID information on request, then you must ensure that public credentials are configured for the external application selected here. If public credentials are not defined, then mail cannot be sent to users on their request. WebCenter Portal offers this feature on its default self-registration page.

7. Restart all servers in the `Portal_Cluster`.

> **Note**
>
> If additional services are being configured, the portal restart can be done once at the end of the section.

Configuring the Content Server Connection

Oracle WebCenter Portal supports content management and storage capabilities, including file upload, file and folder creation and management, file check out, and versioning.

To provide content integration in WebCenter Portal, you must configure at least one WebCenter Content Server connection and mark it as the default connection (sometimes referred to as the active or primary connection). For more information on the requirements, see the Oracle WebCenter Content Server Requirements section in the Oracle Fusion Middleware Installing and Configuring Oracle WebCenter Portal guide.

Additional configuration is needed to assure high-availablity for the WebCenter Portal Content Manager component file uploads.

Registering Oracle WebCenter Content with the WebCenter Portal Application

Provides steps to register Oracle WebCenter Content Server with the WebCenter Portal application.

> **Note**
>
> For more information about Content Server registration, see the Configuring Back-end Data Repositories for Tools and Services section in *Oracle Fusion Middleware Administrator's Guide for Oracle WebCenter Portal*.

Complete the following steps:

1. Sign-in to the Fusion Middleware Control by using the administrator's account (for example: `weblogic_wcp`), and navigate to the home page for your application.

   For example, to navigate to the home page for WebCenter Portal, expand **WebCenter** > **Portal** > **Server** > **WebCenter Portal (WC_Portal1)**.

   > **Note**
   >
   > Multiple **WebCenter Portal** entries will appear, one for each WebLogic Managed Server. Choose any one. The application registration in this section will apply to the entire portal.

2. From the **WebCenter Portal** menu, select **Settings**, and then **Service Configuration**.
3. From the list of services on the WebCenter Service Configuration page, select **Content Repository**.
4. To connect to a new content repository, click **Add**.
5. Enter a unique name for this connection, specify the content repository type, and indicate whether this connection is the active (or default) connection for the application.
   - **Connection Name**

     Enter a name for this content repository connection.

     – Name must be unique across all connection types within the WebCenter Portal application.

     – Name should be consistent between portal environments in order to maintain references when exporting or importing portals and content folders, and so on.
   - **Repository Type**

     Select the type of repository to which you want to connect: **Oracle Content Server**.
   - **Active Connection**

     Select this checkbox.

     You must designate one content repository registration as the active connection to serve as the default repository to use for WebCenter Portal functionality.

     You can connect your WebCenter Portal to multiple content repositories. All connections will be used, however one must be designated the active connection to serve as a default selection.
6. Enter the following additional content repository details after selecting the **Active Connection** checkbox. These values are required on the default repository connection:

   > **Note**
   >
   > The WebCenter Portal application will re-configure the default/active WebCenter Content repository upon the next restart based on these settings to support proper Portal functionality.

   - **Content Administrator**

     The default value of `sysadmin` is recommended. If you wish to customize this value, configure a valid WCC administrative user name here. Administrative privileges are

required for this connection so that operations can be performed on behalf of WebCenter Portal users, including creating and maintaining folders for WebCenter Portal content and managing content access rights.

- **Portal Server Identifier**

  Enter the root folder under which all WebCenter Portal content is stored. Specify a content repository folder that does not yet exist and use the format: `/foldername`. For example: `/MyWebCenterPortal`. The Root Folder cannot be /, the root itself, and it must be unique across applications. The folder specified is created for you when the application starts up. Invalid entries include: `/`, `/foldername/`, `/foldername/subfolder`.

- **Security Group**

  Enter a unique name for this WebCenter Portal application within this content repository. For example: **MyWebCenterPortalApp**

  The name must begin with an alphabetical character, followed by any combination of alphanumeric characters or the underscore character. The string must be less than or equal to fourteen characters.

  This name is used to separate data when multiple WebCenter Portal applications share the same content repository and should be unique across applications. It is also used to name document-related workflows, the security group in which all data created in that Portals application is stored, security roles, as well as to stripe user permissions and default attributes for a particular WebCenter Portal instance.

7. Enter connection details for the content repository:

   - **RIDC Socket Type**

     Select **Socket** - Use an intradoc socket connection to connect to Content Server.

     The client IP address must be added to the list of authorized addresses in the Content Server. In this case, the client is the machine on which Oracle WebCenter Portal is running.

   - **Server Host**

     Enter the Load Balancer address, **wcpinternal.example.com**, so that requests to `/cs` use any available Content Server node.

     The IP address for the virtual host configured on the load balancer and the Self-IP of the load balancer must be added to the Content Server's Incoming Socket Connection Address Security Filter.

     > **Note**
     >
     > If you have not done so already, add a rule to your Load Balancer that specifies how to route WebCenter Content Remote Intradoc Client (RIDC) API traffic, for example:
     >
     > ```
     > (LBR)10.110.10.135:6300 -> 10.110.10.23:4444 (WCCHOST1) &
     > 10.110.10.24:4444 (WCCHOST2)
     > ```

   - **Server Port**

     Enter the port on which the Content Server listens: **6300**

   - **Connection Timeout (ms)**

Specify the length of time allowed to log in to Content Server (in milliseconds) before issuing a connection timeout message. If no timeout is set, there is no time limit for the login operation. Select a reasonable timeout depending on your environment. For example: **60000**.

- **Authentication Method**

  Select **Identity Propagation** - In this enterprise deployment, Content Server and the WebCenter Portal application both use the same identity store to authenticate users.

- **Web Context Root**

  Enter **/cs** as the Web server context root for Content Server.

- **Administrator User Name**

  Enter a user name with administrative rights for this Oracle Content Server instance. This user will be used to fetch content type information based on profiles and track document changes for WebCenter Portal cache invalidation. The default value of `sysadmin` can be used as-is.

- **Administrator Password**

  Leave Empty. An Administrator Password value is only required when the `socketType` is set to `web`.

8. Click **OK** to save this connection.
9. Restart all servers in the `Portal_Cluster`.

> **Note**
>
> If additional services are being configured, the portal restart can be done once at the end of the section.

Configure WebCenter Portal Content Manager MBeans for High Availability

The WebCenter Portal Content Manager component and task flows utilize the WebCenter Content remote UI (RUI) APIs to provide content integration capabilities. While these libraries are directly included with the Portal installation, specific MBean configuration settings need to be modified for fail-safe runtime within a High Availability architecture.

Modify and validate the following attributes for the `ADFConfig:WccAdfConfiguration` and `ADFConfig:ADFcConfig` application-defined MBeans on the webcenter application:

- Application: webcenter:
  - ADFConfig: ADFcConfig
    - AdfScopeHaSupport
  - ADFConfig:WccAdfConfiguration:
    - ClusterCompatible
    - TemporayDirectory

The upload temporary directory specified must be configured to a common directory location on a shared disk volume across all portal nodes when the portal is clustered in a High Availability environment.

1. Create a unique folder on the `ORACLE_RUNTIME` shared volume for the Portal Content Manager component upload location.

   ```
   mkdir -p ORACLE_RUNTIME/DOMAIN_NAME/Portal_Cluster/wccAdfUpload
   ```

   For example,

   ```
   mkdir -p /u01/oracle/runtime/wcedg/Portal_Cluster/wccAdfUpload
   ```

2. Sign-in to the Fusion Middleware Control by using the administrator's account. For example: `weblogic_wcp`.
3. From the WebLogic Domain menu, select **System MBean Browser**.
4. Click the green search filter icon on the System MBean Browser navigation menu and search for the following MBean:

   ```
   oracle.adf.share.config:ApplicationName=webcenter,Location=WC_Portal1,name=WccAdfConf
   iguration,type=ADFConfig,Application=webcenter,ADFConfig=ADFConfig
   ```

   > **Note**
   >
   > Update the Location value with your first portal Managed Server name if different than `WC_Portal1`.

5. Verify the `ClusterCompatible` attribute has a value of: **true**.
6. Set the Temporary Directory attribute to the directory created above on shared storage.

   The Temporary Directory attribute must be set to a directory so that the uploaded files stored under that directory can be accessed by both WCPHOST1 and WCPHOST2.

   For example,

   ```
   /u01/oracle/runtime/wcedg/Portal_Cluster/wccAdfUpload
   ```

7. Click **Apply**.
8. Wait for the following confirmation message:

   ```
   Attributes "TemporaryDirectory" have been updated successfully.
   ```

9. Click the green search filter icon on the System MBean Browser navigation menu and search for the following MBean.

   > **Note**
   >
   > Update the Location value with your first portal Managed Server name if different than `WC_Portal1`.

   ```
   oracle.adf.share.config:ApplicationName=webcenter,Location=WC_Portal1,name=ADFcConfig
   uration,type=ADFConfig,Application=webcenter,ADFConfig=ADFConfig
   ```

10. Verify the `AdfScopeHASupport` attribute value is set to true, update if necessary.
11. If the value was updated, click **Apply**, then verify that a confirmation message appears at the top of the page.

12. Click the green search filter icon on the **System MBean Browser** navigation menu and search for the following MBean:

> **Note**
>
> Update the location value with your first portal Managed Server name if different than `WC_Portal1`.

```
oracle.adf.share.config:ApplicationName=webcenter,Location=WC_Portal1,name=ADFConfig,
type=ADFConfig,Application=webcenter
```

13. Click the **Operations** tab.
14. Click on the **save** link in the Name column on the Operations tab view.
15. On the Operation:save page, click **Invoke** to commit all the MBean changes made since the last save operation.
16. Verify a confirmation message appears at the top of the page, then click **Return**.
17. Restart the portal server used in the location for the MBean configuration in prior steps. This server must be the first to restart for the MBean change to take effect properly.

    For example, restart `WC_Portal1` based on the MBean search example above.
18. Restart all other managed servers in the `Portal_Cluster`.

> **Note**
>
> If additional services are being configured, the remaining portal restarts can be done at the end of the section. With these MBean modifications, `WC_Portal1` server should be restarted first to assure the saved MBean attribute settings are consistently applied.

Restarting the Portal Managed Servers to Activate all Service Configuration Changes

If the optional restarts in the sections above have been deferred, restart the portal managed servers now.

Use the WLST to stop and start the Portal Cluster:

1. Start the WebLogic Scripting Tool:

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

2. Connect to the Administration Server.
3. Stop the Portal Cluster and wait for it to shutdown.

   ```
   shutdown('Portal_Cluster', 'Cluster', force='true', block='true')
   ```

4. Start the Portal Cluster and wait for it to reach a *RUNNING* state.

   ```
   start('Portal_Cluster', 'Cluster', block='true')
   ```

5. Confirm the Portal_Cluster has fully started.

```
state('Portal_Cluster', 'Cluster')
```

6. Exit the WLST session.

```
exit()
```

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.

For information about backing up your configuration, see Performing Backups and Recoveries for an Enterprise Deployment.

16

Integrating WebCenter Portal Workflows with Oracle SOA Suite in the Same Domain

WebCenter Portal provides several prebuilt workflows that handle portal membership notifications, portal subscription requests, and so on. WebCenter Portal workflows rely on the Oracle BPM Worklist, which is installed as a component of Oracle SOA Suite.

WebCenter Portal Worklist integration requires that the BPEL Services provided by SOA Suite share the same WebTier, SSO, and Identity Store with the portal. For this Enterprise Deployment Guide, SOA Suite is installed and configured in the same WebLogic Server Domain and included in the WebTier, SSO, and directory configurations.

The SOA Suite may optionally be deployed to a separate WebLogic Server Domain, however the shared WebTier, SSO, and identity Store requirements must be met for the Portal Workflow integration task flows to function correctly.

For more information on Oracle BPM Worklist features, see Using Oracle BPM Worklist in *Developing SOA Applications with Oracle SOA Suite*.

The tasks that must be performed to enable the WebCenter Portal workflow functionality in WebCenter Portal are as follows.

> **Note**
>
> Integrating BPM functionality into portal pages using the BPM Process Portal Resource Catalog requires additional steps not covered in this guide. See Integrating BPM Functionality into WebCenter Portal.

Backing Up the Installation

Back up the environment before re-configuring to include the Portal Workflow Integration with SOA Suite.

This is a quick backup for the express purpose of immediate restore in case of problems executing this chapter. The backup destination is the local disk. You can discard this backup once the enterprise deployment setup is complete. At that point, the regular deployment-specific backup and recovery process can be initiated. The Oracle Fusion Middleware Administrator's Guide provides further details.

To back up the environment:

1. Shutdown the domain resources in order: Managed Servers, Admin Server, Node Managers.
2. Back up the database. A full database backup (either hot or cold) is strongly recommended.

3. Clear the WebLogic Server and NodeManager logs to aid in troubleshooting this chapter and reduce backup size.

```
find ASERVER_HOME/servers/AdminServer/logs -type f ! -size 0c -print -exec
rm -f {} \+
find ASERVER_HOME/nodemanager -type f \( -name '*.log' -or -name '*.out'
\) -print -exec rm -f {} \+

find MSERVER_HOME/servers/*/logs -type f ! -size 0c -print -exec rm -f {} \
+
find MSERVER_HOME/nodemanager -type f \( -name '*.log' -or -name '*.out'
\) -print -exec rm -f {} \+
```

4. Backup up the Administration Server domain directory.

```
tar -czf ASERVER_HOME/../backup_aserver_home_DOMAIN_NAME_WCPEDG_CH16_1.tgz
ASERVER_HOME &
```

5. Clean up and backup the Applications Directory.

```
find APPLICATION_HOME -type f -name "*.bak*" -print -exec rm -f {} \;
tar -czf APPLICATION_HOME/../backup_app_home_DOMAIN_NAME_ch16_1.tgz
APPLICATION_HOME &
```

6. Backup other shared configuration and runtime folders.

```
tar -czf KEYSTORE_HOME/../backup_keystore_home_DOMAIN_NAME_ch16_1}.tgz
KEYSTORE_HOME &
tar -czf DEPLOY_PLAN_HOME/../backup_dp_home_DOMAIN_NAME_ch16_1.tgz
DEPLOY_PLAN_HOME &
tar -czf ORACLE_RUNTIME/backup_runtime_DOMAIN_NAME_ch16_1.tgz
ORACLE_RUNTIME/DOMAIN_NAME &
```

7. Backup the shared product *ORACLE_HOME* binaries and Oracle Inventory on WCCHOST1 and WCCHOST2.

```
tar -czf ORACLE_HOME/../backup_fmw_ch16_1.tgz ORACLE_HOME &
tar -czf /u01/oracle/products/backup_oraInv_ch16_1.tgz /u01/oracle/
products/oraInventory &
```

8. Restart the Node Manager for *AdminServer* on WCCHOST1.

```
nohup ASERVER_HOME/bin/startNodeManager.sh > ASERVER_HOME/nodemanager/
nodemanager.out 2>&1 &
sleep 5 && grep "started on port" ASERVER_HOME/nodemanager/nodemanager.log
```

> **Note**
>
> Do not start the *AdminServer* or any other domain services at this time, as additional product installations and a domain extension occur in later sections.

9. Restart the Node Manager for *MSERVER_HOME* on each of the application tier hosts.

```
nohup MSERVER_HOME/bin/startNodeManager.sh > MSERVER_HOME/nodemanager/
nodemanager.out 2>&1 &
sleep 5 && grep "started on port" MSERVER_HOME/nodemanager/nodemanager.log
```

Installing Oracle SOA Suite

To support workflows, WebCenter Portal relies on the BPEL server, which is included with Oracle SOA Suite. For information about installing Oracle SOA Suite as part of this domain, see Extending the Domain with Oracle SOA Suite .

Installing the Oracle WebCenter Portal SOA Composites

To use workflows in WebCenter Portal, you must install WebCenter Portal SOA Composites by using the portal installer after SOA Suite is installed.

To install WebCenter Portal SOA Composites:

1. Execute the WebCenter Portal Installer a second time for each shared ORACLE_HOME, selecting an Installation Type of WebCenter Portal SOA Composites.
2. Verify that the WebCenter Portal composite archive has been installed.

Starting the Oracle WebCenter Portal 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 example below.

   ```
   $JAVA_HOME/bin/java  -jar fmw_14.1.2.0.0_wcportal.jar
   ```

   Be sure to replace the JDK location in these examples with the actual JDK location on your system.

   For information about downloading the software and locating the actual installer file name for your product, see 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.

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*.
Installation Type	Use this screen to select the type of installation and consequently, the products and feature sets you want to install. To install the SOA Composites for WebCenter Portal, select: • **WebCenter Portal SOA Composites**
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 following documents in the Roadmap for Verifying Your System Environment section in *Planning Your Oracle Fusion Middleware Infrastructure Installation.*
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 you want to skip this step, clear the check box and verify your selection in the follow-up dialog box.
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.

Verifying the Installed Files

Once the installation has been completed, verify that the WebCenter Portal SOA Composite and worklist details application archives have been written to the correct directory structure within ORACLE_HOME as follows:

```
ls ORACLE_HOME/wcportal/common/soa-composite/wcp/sca_CommunityWorkflows.jar
ls ORACLE_HOME/wcportal/webcenter/applications/WebCenterWorklistDetailApp.ear
```

Performing the Installation on WCCHOST2

The installation should be repeated once for each shared file system containing a unique *ORACLE_HOME*. See Summary of the Shared Storage Volumes in an Enterprise Deployment.

Extending the Domain to Deploy the WebCenter Portal Workflows

WebCenter Portal workflows are deployed to Oracle SOA cluster. You must extend the domain in which Oracle SOA is installed with the template: `oracle.wc_composite_template.jar`.

When executing the configuration wizard, the domain's AdminServer and any managed servers that will be modified by the selected components/templates must be shutdown. All other unaffected managed servers may stay running.
It is recommended that for this section, all managed servers, including those in the `WSM-PM_Cluster` be shutdown. This will save having to stop and restart them anyway later.

To extend the domain:

1. Shutdown all managed servers through EM, WLS console, or WLST.
2. Shutdown the AdminServer for the domain to be extended.
3. Run the following command to start the configuration wizard.

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

4. 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.

   ```
   /u01/oracle/config/domains/domain_name/
   ```

5. On the Templates screen, select the template in either of the following ways:
   - Select Update Domain Using Product Templates, then select:
     Oracle WebCenter Portal Composites - 14.1.2.0.0 [wcportal]
   - Select Update Domain Using Custom Template, and specify the following path in the Template location field:
     ORACLE_HOME`/wcportal/common/templates/wls/oracle.wc_composite_template.jar`

   The `oracle.wc_composite_template.jar` template automatically deploys:
   - `WebCenterWorklistDetailApp.ear`, the ADF application that displays invitations and messages.
   - `sca_CommunityWorkflows.jar`, the BPEL composite that manages the WebCenter Portal membership mechanism.
6. Review and click **Next** to continue through the next few screens until you reach the Advanced Configuration view.
7. On the Advanced Configuration screen, select **Deployments and Services**, then click **Next**.
8. Deployments Targeting.

   The `WebCenterWorklistDetailApp` application deployment should be targeted to the `SOA_Cluster` only.

Validate the application deployments targeted to each cluster and remove the `WebCenterWorklistDetailApp` from any cluster it is targeted to except for the `SOA_Cluster`. Ensure it remains targeted to the `SOA_Cluster`.

9. Services Targeting.

   With the Oracle Web Services Manager Policy Manager deployed to a separate cluster, the default targeting of the WSM-PM service resources to the Portal, Portlet, In-Bound Refinery, Content, and SOA clusters should be removed.

   For each of the `Portal_Cluster`, `Portlet_Cluster`, `SOA_Cluster`, `WCC_Cluster`, and `IBR_Servers` in the Targets panel, select and remove the following resource from the targets list:

   - mds-owsm

10. 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 of the previous screen if you need to make any changes, either by selecting **Back** or by selecting the required 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*.

11. 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.

Propagating the Extended Domain to the Domain Directories and Machines

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, complete the following steps:

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=/full_path/wcpdomaintemplateExtComposites.jar
             -template_name=wcp_domain_template_extension_composites
   ```

   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 location where you want to create the domain template jar file. You will need to reference this location when you copy or unpack the domain template jar file. It is recommended to choose a shared volume other than *ORACLE_HOME*, or write to `/tmp/` and copy the files manually between servers.

  You must specify a full path for the template jar file as part of the `-template` argument to the pack command:

  ```
  SHARED_CONFIG_DIR/domains/template_filename.jar
  ```

- `wcpdomaintemplateExtComposites.jar` is a sample name for the JAR file you are creating, which will contain the domain configuration files, including the configuration files for the Oracle HTTP Server instances.
- `wcp_domain_template_extension_composites` is the name assigned to the domain template file.

3. Run the following `unpack` command on WCCHOST1 to propagate the template created in the preceding step to the *MSERVER_HOME* directory:

   ```
   cd ORACLE_COMMON_HOME/common/bin

   ./unpack.sh -domain=MSERVER_HOME
               -template=/full_path/wcpdomaintemplateExtComposites.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.
   - `wcpdomaintemplateExtComposites.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 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.
   - Replace *APPLICATION_HOME* with the complete path to the Application directory for the domain on local 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. If the full path to the packed jar file is on a shared volume available to the other servers, skip this step, otherwise, run the following command on WCCHOST1 to copy the template pack created in step 1 to WCCHOST2, WCPHOST1, and WCPHOST2:

```
scp /full_path/wcpdomaintemplateExtComposites.jar oracle@WCCHOST2:/full_path/
scp /full_path/wcpdomaintemplateExtComposites.jar oracle@WCPHOST1:/full_path/
scp /full_path/wcpdomaintemplateExtComposites.jar oracle@WCPHOST2:/full_path/
```

5. Run the following `unpack` command on each of the remote hosts to deploy the domain template copied in the preceding step to the *MSERVER_HOME* directory:

```
cd ORACLE_COMMON_HOME/common/bin

./unpack.sh -domain=MSERVER_HOME
            -template=/full_path/wcpdomaintemplateExtComposites.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.
- `wcpdomaintemplateExtComposites.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 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.
- Replace *APPLICATION_HOME* with the complete path to the Application directory for the domain on local 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*.

Restoring customizations to setDomainEnv.sh after Unpacking the Domain

If any customizations have been made earlier to the `setDomainEnv.sh` files in *ASERVER_HOME* and *MSERVER_HOME*, then these customizations will need to be repeated after any domain extension.

> **Note**
>
> Modifying the `setDomainEnv` script is not recommended. For more information, see Customizing Domain Wide Server Parameters in *Administering Server Startup and Shutdown for Oracle WebLogic Server*.
>
> For WebCenter Enterprise Deployments, see Customizing Server Parameters with the setUserOverridesLate Script.

Verify that all customizations have been restored before starting NodeManager or WebLogic Server instances.
On WCCHOST1:

1. Verify and update *ASERVER_HOME*/bin/setDomainEnv.sh.
2. Verify and update *MSERVER_HOME*/bin/setDomainEnv.sh.
3. Copy *MSERVER_HOME*/bin/setDomainEnv.sh to the other hosts (Example: WCCHOST2, WCPHOST1, and WCPHOST2).

> **Note**
>
> There are unique differences in parameter values stored in the ASERVER_HOME and MSERVER_HOME setDomainEnv.sh configuration files. The same file cannot be copied into both locations and should be edited separately. *MSERVER_HOME*/bin/setDomainEnv.sh can be copied across the environment consistently.

Updating the NodeManager Configuration After Unpacking the Domain

When extending a domain, the nodemanager.properties file in *MSERVER_HOME* may be overwritten with some values from the nodemanager.properties file for *ASERVER_HOME*. Specifically, the ListenAddress and/or CustomIdentityAlias values can be reset.

> **Notes:**
>
> - The ListenAddress may typically get reset on the *MSERVER_HOME* nodemanager residing on the same host as the *ASERVER_HOME* nodemanager. In this topology, WCCHOST1.
> - For domain extensions prior to Enabling SSL Communication Between the SOA Servers and the Hardware Load Balancer, steps 2 through 4 regarding the CustomIdentityAlias may not be applicable.

For the *MSERVER_HOME*/nodemanager/nodemanager.properties file on each host:

1. Verify the correct ListenAddress parameter value and reset it, if required.

   ```
   grep ListenAddress MSERVER_HOME/nodemanager/nodemanager.properties
   ```

2. Confirm the list of configured Identity Aliases from the domain configuration file as a reference for the next command.

   ```
   grep server-private-key-alias ASERVER_HOME/config/config.xml | sort | uniq
   ```

> **Note**
>
> Use the appropriate host-specific certificate identity aliases when updating the nodemanger.properties CustomIdentityAlias property in the next instruction.

3. Verify the current `nodemanager.properties CustomIdentityAlias` parameter value matches the alias for the host.

```
grep CustomIdentityAlias MSERVER_HOME/nodemanager/nodemanager.properties
```

4. Reset the `CustomIdentityAlias` parameter value to the correct alias string appropriate for the current host, if required.
5. Restart the `nodemanager` process:

```
kill `ps -eaf | grep weblogic.NodeManager | grep MSERVER_HOME | grep -v
grep | awk '{print $2}' `
nohup MSERVER_HOME/bin/startNodeManager.sh > MSERVER_HOME/nodemanager/
nodemanager.out 2>&1 &
```

> **Note**
>
> For more information about the `CustomIdentityAlias` parameter, see Configuring Node Manager to Use the Custom Keystores.

Starting the Domain and Validating the WebCenter Portal SOA Composite Domain Extension

Start the entire domain and use Enterprise Manager to verify the deployment of the Portal SOA Composites and WebCenter Worklist Detail application.

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. Set the WLST Properties.

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

2. Start the WebLogic Scripting Tool (WLST):

> **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.

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

3. 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`

4. 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.>
> ```

5. Exit WLST:

```
exit()
```

Start and confirm all Managed Servers are running

Managed servers created or modified by the latest domain extension should now be started. Managed servers that remained running during the domain extension should be confirmed as running.

Table 16-1 Managed Servers

Cluster	Managed Servers	Initial State	Action
WSM-PM_Cluster	`WLS_WSM`*n*	SHUTDOWN	Start and verify all managed servers
SOA_Cluster	`WLS_SOA`*n*	SHUTDOWN	Start and verify all managed servers
IBR_Servers	`WLS_IBR`*n*	SHUTDOWN	Start and verify all managed servers
WCC_Cluster	`WLS_WCC`*n*	SHUTDOWN	Start and verify all managed servers
Portlet_Cluster	`WC_Portlet`*n*	SHUTDOWN	Start and verify all managed servers
Portal_Cluster	`WC_Portal`*n*	SHUTDOWN	Start and verify all managed servers

Verifying the WebCenter Portal SOA Composites Deployment

Two deployments are added when the domain is extended with the WebCenter Portal SOA Composites. These include one enterprise application archive and one SOA composites archive. These resources must be successfully deployed and validated before continuing with this domain extension. The SOA composites rely on the application for the human tasks included in the workflows. They are deployed separately, using different processes.

- `WebCenterWorklistDetailApp.ear` — A standard Java EE web application located in *`ORACLE_HOME`*`/wcportal/webcenter/applications`
- `sca_CommunityWorkflows.jar` — A SOA Composite located in *`ORACLE_HOME`*`/wcportal/common/soa-composite/wcp`

This section contains instructions for both the validation and deployment processes.

Confirming the WebCenter Portal SOA Composite and Application Deployments

To validate the `WebCenterWorklistDetailApp.ear` application deployment:

1. Connect to Enterprise Manager as the `weblogic_wcp` administrative user.
2. Verify that the `WebCenterWorklistDetailApp` application is listed **Target Navigation** > **Application Deployments**. If the application link is not listed, see section Deploying the WebCenterWorklistDetailApp Application to the SOA_Cluster.
3. If the `WebCenterWorklistDetailApp` application is listed, click on the link and validate that the State is listed as **Active** and the Health is **OK** in the Summary view. Also, validate that the **SOA_Cluster** is listed in the Targets column of the Deployments view.

   If the `WebCenterWorklistDetailApp` does not show the SOA_Cluster in the Targets column, complete the following steps:

a. From the Domain Application Deployment drop-down list, select **Administration** > **Targets**.

b. From the lock icon in the upper-right corner, select **Lock & Edit**.

c. Select **WebCenterWorklistDetailApp EAR**, then click **Change Targets**.

d. In the pop-up window, select **SOA_Cluster** and then select the **All configured Servers in this cluster** option.

e. Click **OK**. After the changes are complete a confirmation message is displayed

f. If an information panel is displayed with the following message, click **Create New Deployment Plan**.

Information: The configuration changes will be saved in the deployment plan. This application does not currently have a deployment plan. In order to save the configuration changes, you need to first create a new deployment plan for this application.

g. In a separate command shell, create a shared deployment plan folder for the `WebCenterWorklistDetailApp` in the `DEPLOY_PLAN_HOME` folder. See Understanding the Recommended Directory Structure for an Enterprise Deployment.

```
mkdir -p DEPLOY_PLAN_HOME/WebCenterWorklistDetailApp
```

h. In the Save Deployment Plan dialog, enter or browse to the full path to the new app-specific deployment plan folder, plus the file name plan.xml.

```
/u01/oracle/config/dp/WebCenterWorklistDetailApp/plan.xml
```

i. Click the **Save Deployment Plan** button.

j. From the lock icon in the upper-right corner, select **Activate Changes**.

> **Note**
>
> If the Activate Chagnes selection is unavailable, click the reload icon next to the date immediately below the lock icon.

k. Restart the managed servers in the SOA_Cluster.

4. Expand the **Target Navigation** panel and navigate to the **WebLogic Domain** > **domain-name** > **SOA_Cluster**.

5. From the WebLogic Cluster drop-down menu, select **Deployments**.

6. Verify that the `WebCenterWorklistDetailApp` is listed with a green *up* arrow status and a state of **Active**.

   If the `WebCenterWorklistDetailApp` is not deployed to the SOA_Cluster, perform the deployment after all validation steps are completed. See Deploying the WebCenterWorklistDetailApp Application to the SOA_Cluster.

To validate the `sca_CommunityWorkflows.jar` SOA composites deployment:

1. Expand the **Target Navigation** panel and navigate to the **SOA** > **soa-infra (WLS_SOA1)** service

2. Click the **Deployed Composites** tab.

3. Verify that the CommunitWorkflows [12.2.1.3.0] composite is listed with a green *up* arrow for status.

   See Deploying the CommunityWorkflows SOA Composite to the SOA service.

Deploying the WebCenterWorklistDetailApp Application to the SOA_Cluster

If the `WebCenterWorklistDetailApp` application needs to be deployed, perform the following steps:

1. Connect to Enterprise Manager as the weblogic_wcp administrative user .
2. Expand the **Target Navigation** panel and navigate to the **WebLogic Domain** > ***domain-name*** > **SOA_Cluster**
3. From the WebLogic Cluster drop-down menu, select **Deployments**.
4. Verify that the `WebCenterWorklistDetailApp` is not listed.
5. From the Deployment drop-down menu, select **Deploy**.
6. Select an appropriate deployment scope. For out-of-box configurations, the appropriate scope is **global**.
7. Select **Archive on the server where Enterprise Manager is running**.
8. Enter: *`ORACLE_HOME`*`/wcportal/webcenter/applications/WebCenterWorklistDetailApp.ear`.
9. Select **Create a new deployment plan when deployment configuration is done** .
10. Select **Deploy this archive or exploded directory as an application**.
11. Click **Next**.
12. On the Select Target view, make sure that only the **SOA_Cluster** is checked and the **All configured Servers** in this cluster option is selected.
13. On the Application Attributes view, change only the distribution option to: **Install and start application (servicing all requests)**. Do not alter any other application attributes
14. Click **Deploy**. The remaining application deployment configurations do not need to be modified.
15. Observe the progress messages provided in the **Processing: Deploy** modal dialog box that appears and wait for it to complete.
16. Observe that the dialog box is updated with a *Deployment Succeeded* message.
17. Close the dialog box.
18. Verify that the new application deployment is listed with a green *up* arrow status and a state of **Active**.

See Deploying Java EE Applications Using Fusion Middleware Control in *Administering Oracle Fusion Middleware* for details on how to deploy the enterprise application archive.

Deploying the CommunityWorkflows SOA Composite to the SOA service

If the `CommunityWorkflows SOA` composite needs to be deployed, perform the following steps:

1. Connect to Enterprise Manager as the `weblogic_wcp` administrative user .
2. Expand the **Target Navigation** panel and navigate to the **SOA** > **soa-infra (WLS_SOA1)** service.
3. Click on the **Deployed Composites** tab.
4. Verify the CommunityWorkflows composite is listed as *Up* and *Active*. If not listed, or the list says *No Composites Found*, then continue with these deployment steps. If status is down, select and start the CommunityWorkflows composite.

5. Click **Deploy**.
6. Select **Archive on the server where Enterprise Manager is running**.
7. Enter: *ORACLE_HOME*`/wcportal/common/soa-composite/wcp/sca_CommunityWorkflows.jar`
8. Select **No external configuration plan is required**.
9. Click **Next**.
10. Confirm the deployment target is **/Domain_< domain_name >/< domain_name >/ SOA_Cluster**
11. Choose the appropriate SOA Folder. For out-of-box configurations, the appropriate folder to select is **default**.
12. Click **Next**.
13. Confirm the **Deploy as default revision** selection as this is the first time the composites are getting deployed.
14. Click **Deploy**.
15. Observe the progress messages provided in the **Processing: Deploy** modal dialog box that appears and wait for it to complete.
16. Observe that the dialog box is updated with a **Deployment Succeeded** message.
17. Close the dialog box.
18. Observe that Enterprise Manager now displays the `CommunityWorkflows[12.2.1.3.0]` SOA composite dashboard view with several components and services.

See Deploying SOA Composite Applications in *Administering Oracle SOA Suite and Oracle Business Process Management Suite* for details on how to deploy the composite.

Configuring WS-Security for Oracle SOA and WebCenter Portal

WebCenter Portal Web services, deployed to Oracle WebCenter Portal, facilitate communication between WebCenter Portal and the SOA server. You must secure these Web service calls.

> **Note**
>
> Some of the key aliases and other properties values used in this configuration are specifically required by the deployed products. This process has been tuned specifically for a combined topology with Oracle SOA and WebCenter Portal in the same domain. Customizing this process beyond the provided instructions is not recommended.
> For more information on configuration for a two-domain topology, see Oracle SOA and WebCenter Portal - WS-Security Configuration in *Installing and Configuring Oracle WebCenter Portal*.

Set up WS-Security by creating a security application stripe and keystore for WebCenter Portal and the SOA Suite BPEL Server to use. Oracle Fusion Middleware 12c implements these keystores using the Keystore Security Service (KSS) configured via Enterprise Manager or WLST commands.

See Configuring Keystores for Message Protection in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

For syntax and reference information about the KSS commands, see OPSS Keystore Service Commands in *Oracle Fusion Middleware Infrastructure Security WLST Command Reference*.

Creating the WebCenter Portal Keystore via WLST

To create a new WebCenter Portal keystore, complete the following steps:

1. Start WLST and connect to the Administration Server as the administrative LDAP user.

   ```
   ORACLE_COMMON_HOME/common/bin/wlst.sh
   connect("weblogic_wcp", "weblogic_admin_pwd", "t3s://ADMINVHN:9002")
   ```

2. Use the following WLST command to get an OPSS service command object:

   ```
   svc = getOpssService(name='KeyStoreService')
   ```

3. Create the keystore using the following WLST command:

   ```
   svc.createKeyStore(appStripe='WCPortalStripe', name='producer',
   password='password', permission=true)
   ```

   Where:

   - *appstripe* — The keystore stripe name. Keys and certificates created in the keystore reside in an application stripe or product, and each stripe in a domain is uniquely named
   - *keystore_name* — The name of the keystore you are creating; typically, you use a name to help identify the component or endpoint being secured
   - *password* — Enter the password you want to use for this keystore.
   - *permission* — false if protected by both permission and password (true if keystore is protected by permission only)

4. Generate the key pair for this newly created keystore, supplying an appropriate password and setting the domain name, organization, location(city), state, and country appropriately:

   ```
   svc.generateKeyPair(appStripe='WCPortalStripe', name='producer',
   password='password', dn='CN=Producer, OU=WCPortalServices, OU=domain_name,
   O=MyOrganization, L=MyTown, ST=MyState, C=US', keysize='2048',
   alias='producer', keypassword='keypassword')
   ```

   Where:

   - *appstripe* — The keystore stripe name. Keys and certificates created in the keystore reside in an application stripe or product, and each stripe in a domain is uniquely named
   - *keystore_name* — The name of the keystore you are creating; typically, you use a name to help identify the component or endpoint being secured
   - *password* — The password you defined when you created the keystore.
   - *dn* — Distinguished Name; used to uniquely identify and organize the key pair within a hierarchical naming structure. Update the `OU=`*domain_name*, `O=`*MyOrg*, `L=`*MyTown*, `ST=`*MyState*, `C=`*US*, portions of the DN to match your environment and organization appropriately.

- *keysize* - number of bits for the encryption key, should be at least 2048
- *alias* — Public Key Alias
- *keypassword* — Enter a password for new public key that you are creating.

5. Export the producer certificate (which will be used by the consumer):

```
svc.exportKeyStoreCertificate(appStripe='WCPortalStripe', name='producer',
password='password', alias='producer',
type='TrustedCertificate',filepath='KEYSTORE_HOME/
ksscert_wcportalproducer.crt')
```

Where:

- *appstripe* — The keystore stripe name. Keys and certificates created in the keystore reside in an application stripe or product, and each stripe in a domain is uniquely named
- *name* — Keystore name
- *password* — Keystore password
- *alias* — Public Key Alias
- *keypassword* — Password for new public key
- *filepath* — Certificate path

6. Import the certificate exported by the producer for use by the consumer web service:

```
svc.importKeyStoreCertificate(appStripe='WCPortalStripe', name='producer',
password='password', alias='webcenter_spaces_ws',
keypassword='keypassword', type='TrustedCertificate',
filepath='KEYSTORE_HOME/ksscert_wcportalproducer.crt')
```

Where:

- *password* — Keystore password
- *keypassword* — Password for new public key
- *filepath* — Certificate path

> **Note**
>
> The alias for the `importKeyStoreCertificate` command must always be set to `webcenter_spaces_ws`. Do not attempt to change this alias; otherwise, the web services communications will fail. This alias is used in the default configuration of the Web services policy security.

7. Register the producer stripe:

```
configureWSMKeystore('/WLS/domain_name','KSS', 'kss://WCPortalStripe/
producer', signAlias='producer', cryptAlias='producer',
signAliasPassword='password', cryptAliasPassword='password')
```

Where:

- *domain_name* – The name of the WebCenter Portal domain; the "/WLS/" prefix is required to provide context for the Web Services Manager manager
- KSS — The keystore type
- `kss://WCPortalStripe/producer`— The location of the keystore: keys and certificates created in the keystore reside in an application stripe or product, and each stripe in a domain is uniquely named
- *signAlias* – The alias of the signature key; the value must match the value in the keystore, in this case, 'producer'
- *cryptAlias* — The alias of the encryption key. The value that you specify here must match the value in the keystore, in this case, 'producer'
- *signAliasPassword*— The password for the certificate specified for the *signAlias* as configured earlier
- *cryptAliasPassword*— The password for the certificate specified for the *cryptAlias* as configured earlier

> **Note**
>
> When using a KSS keystore type, the `configureWSMKeystore()` command may issue warnings that the passwords are not required. In this specific case, when services are using policies that mandate message protection, the passwords are required, otherwise the services cannot use the certificates to encrypt and decrypt appropriately. Be sure to include the password parameters as indicated in the example.

See Configuring the OWSM Keystore Using WLST in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

8. Grant Keystore Permission for the newly created `producer` keystore in the WCPortalStripe stripe:

```
grantPermission(permClass="oracle.security.jps.service.keystore.KeyStoreAcc
essPermission",
permTarget="stripeName=WCPortalStripe,keystoreName=producer,alias=*",
permActions="read")
```

> **Note**
>
> The `StripeName` and `keystoreName` values in the `permTarget` value must match values used in earlier steps. No other changes are required to this command.

Verifying Application Roles

Before you configure WebCenter Portal with SOA Suite, understand and verify the `SOAAdmin` and `BPMWorkflowAdmin` application roles.

The memberships of the SOAAdmin and BPMWorkflowAdmin application roles can be listed as follows:

```
ORACLE_COMMON_HOME/common/bin/wlst.sh
connect('weblogic_wcp','password', 't3://ADMINVHN:7001')
listAppRoleMembers(appStripe="soa-infra", appRoleName="SOAAdmin")
listAppRoleMembers(appStripe="soa-infra", appRoleName="BPMWorkflowAdmin")
```

To revoke a user or group membership from an application role, consider following WLST examples:

```
revokeAppRole(appStripe="soa-infra", appRoleName="BPMWorkflowAdmin",
principalClass="weblogic.security.principal.WLSUserImpl",
principalName="weblogic")
revokeAppRole(appStripe="soa-infra", appRoleName="BPMWorkflowAdmin",
principalClass="weblogic.security.principal.WLSGroupImpl",
principalName="Administrators")
```

To specifically grant a user membership to an application role, follow this example:

```
grantAppRole(appStripe="soa-infra", appRoleName="BPMWorkflowAdmin",
principalClass="weblogic.security.principal.WLSUserImpl",
principalName="weblogic_wcp")
```

For the LDAP group configurations in the enterprise deployment environment, it is not necessary to grant the `weblogic_wcp` user specific access. The `WCPAdministrators` group should already be part of the `SOAAdmin` application role and inherit membership to the `BPMWorkflowAdmin` application role.

For more information about configuring a remote LDAP server for the enterprise deployment, see Creating a New LDAP Authenticator and Provisioning Enterprise Deployment Users and Group.

For more information about SOA application role configurations, see Configuring Roles for Administration of an Enterprise Deployment.

Creating the Connection to the BPEL Server

WebCenter Portal uses BPEL server to host internal workflows, such as worklists, membership notifications, subscription requests, and so on. BPEL Services are configured on the SOA Managed Servers. To enable workflow functionality for WebCenter Portal, a connection to the BPEL service is required.

> **Note**
>
> WebCenter Portal workflows must be deployed on the SOA managed server that WebCenter Portal is configured to use. See also, Back-End Requirements for WebCenter Portal Workflows in *Installing and Configuring Oracle WebCenter Portal*.

To configure a connection for worklist notifications:

1. Sign-in to the Fusion Middleware Control by using the administrator's account (for example: `weblogic_wcp`), and navigate to the home page for your application.

   See “Navigating to the Home Page for WebCenter Portal” in *Administering Oracle WebCenter Portal*.
2. From the **WebCenter Portal** menu, select **Settings**, then **Application Configuration** .
3. In the BPEL SOAP URL field, specify the internal load-balanced URL.

   For example:

   ```
   http://wcp-internal.example.com:80
   ```
4. In the Link URL field, specify the public front-end load-balanced URL for the environment.

   For example:

   ```
   https://wcp.example.com:443
   ```
5. Select **Enable WebCenter Portal Workflows**.
6. Click **Apply**.
7. Restart the Portal_Cluster for this change to take effect.

   See “Starting and Stopping Managed Servers for WebCenter Portal Application Deployments” in *Administering Oracle WebCenter Portal*.

Validating the Connection to the BPEL Server

After you create the connection to the BPEL Server, validate the connection to be sure it is working properly.

Use the WLST command `listWorklistConnections` to display the configured connections and validate the connection details.

Use the `getSpacesWorkflowConnectionName()` to confirm the name of the active workflow connection.

For example:

```
listWorklistConnections(appName='webcenter', server='WC_Portal1', verbose=1)
getSpacesWorkflowConnectionName(appName='webcenter', server='WC_Portal1')
```

Use the listed URL property value to construct a valid worklist application URL and validate access using a browser. Append the listed URL property value with the path `/integration/worklistapp`, to generate an appropriate URL for testing.

Configuring WebCenter Portal Workflow Notifications to be Sent by Email

WebCenter Portal can use human workflows (requiring human interaction), which are integrated with SOA workflows. The SOA server can configure email so that notifications are delivered to a user's inbox, where the user can accept or reject the notification.

This topic briefly explains how to enable email notifications and set your mail server details to have WebCenter Portal workflow notifications sent through email. Both outbound and incoming

email addresses or mailboxes that are dedicated to portal workflow notification and reply processing are needed for full functionality. For a more detailed description, see Configuring Human Workflow Notification Properties in *Administering Oracle SOA Suite and Oracle Business Process Management Suite*.

1. Sign-in to the Fusion Middleware Control by using the administrator's account. For example: `weblogic_wcp`.
2. Expand the Target Navigation panel and navigate to the **SOA** > **soa-infra (WLS_SOA1) service**.
3. From the **SOA Infrastructure** drop-down menu, select **SOA Administration** > **Workflow Properties**.
4. Set the Notification Mode to: **Email**
5. Provide the correct email addresses for the Notification Service.
6. Click **Apply** and then confirm when prompted. Verify the returned message that confirms changes have been applied.
7. Click the **Go to the Messaging Driver page** link.
8. In the Associated Drivers section, Click the **Configure Driver** icon for the **User Messaging Email Driver**.
9. Click **Create** to configure an email driver, if one does not already exist.

   For instructions on how to configure the email driver for notifications, see Configuring an Email Driver for Notifications in *Using Oracle Managed File Transfer*.
10. Once all required email driver configurations are completed, click **Test** and validate successful test status.
11. Click **OK** to save the email driver configuration.
12. Verify that the new UMS driver configuration shows up in the **Email Driver Properties table** view.
13. Restart the SOA Cluster. No configuration or restart is required for WebCenter Portal.

Testing the Oracle BPM Worklist Application in WebCenter Portal

Testing of the WebCenter Portal invitation and membership workflows and email notifications can be performed using end-user accounts and requires specific portal run-time configuration to set up the test case.

To access BPEL worklist task details sent from WebCenter Portal, without incurring additional login prompts, WebCenter Portal and Oracle SOA Suite servers must be configured to a shared Oracle Single Sign-On server. This testing can be more easily validated after completing the instructions in the Configuring Single Sign-On for an Enterprise Deployment section.

See Task 7 in the Configuration Roadmap for WebCenter Portal Workflows section in the *Administering Oracle WebCenter Portal Guide*.

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.

For information about backing up your configuration, see Performing Backups and Recoveries for an Enterprise Deployment.

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.

17

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

These are some of the typical configuration and management tasks you are likely need to perform on an Oracle Fusion Middleware enterprise 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.

To reduce the `WLSSchemaDataSource` connection usage, you can change the JMS JDBC and TLOG JDBC stores connection caching policy from *Default* to *Minimal* by using the respective connection caching policy settings. When there is a need to reduce connections in the back-end database system, Oracle recommends that you set the caching policy to *Minimal* . Avoid using the caching policy *None* because it causes a potential degradation in performance. For a detailed tuning advice about connections that are used by JDBC stores, see Configuring a JDBC Store Connection Caching Policy in *Administering the WebLogic Persistent Store*.

The default `WLSSchemaDataSource` connection pool size is 75 (size is double in the case of a GridLink DataSource). You can tune this size to a higher value depending on the size of the different FMW clusters and the candidates that are configured for migration. For example, consider a typical SOA EDG deployment with the default number of worker threads per store. If more than 25 JDBC Stores or TLOG-in-DB instances or both can fail over to the same Weblogic server, and the Connection Caching Policy is not changed from *Default* to *Minimal*, possible connection contention issues could arise. In these cases, increasing the default `WLSSchemaDataSource` pool size (maximum capacity) becomes necessary (each JMS store uses a minimum of two connections, and leasing and JTA are also added to compete for the pool).

Verifying Manual Failover of the Administration Server

In case a host computer fails, you can fail over the Administration Server to another host. The steps to verify the failover and failback of the Administration Server from WCCHOST1 and WCCHOST2 are detailed in the following sections.

Assumptions:

- The Administration Server is configured to listen on ADMINVHN, and not on localhost or on any other host's address.

  For more information about the ADMINVHN virtual IP address, see Reserving the Required IP Addresses for an Enterprise Deployment.

- These procedures assume that the Administration Server domain home (*ASERVER_HOME*) has been mounted on both host computers. This ensures that the Administration Server domain configuration files and the persistent stores are saved on the shared storage device.
- The Administration Server is failed over from WCCHOST1 to WCCHOST2, and the two nodes have these IPs:
  - WCCHOST1: 100.200.140.165
  - WCCHOST2: 100.200.140.205
  - ADMINVHN : 100.200.140.206. This is the Virtual IP where the Administration Server is running, assigned to a virtual sub-interface (for example, eth0:1), to be available on WCCHOST1 or WCCHOST2.
- Oracle WebLogic Server and Oracle Fusion Middleware components have been installed in WCPHOST2 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 domain Node Manager for the enterprise topology. See About the Node Manager Configuration in a Typical Enterprise Deployment

To fail over the Administration Server to a different host:

1. Stop the Administration Server 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 to check the virtual IP address at its CIDR:

      ```
      ip addr show dev ethX
      ```

      Where, `X` is the current interface used by ADMINVHN.

      For example:

      ```
      ip addr show dev eth0
      ```

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

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

Where, X:Y is the current interface used by ADMINVHN.

For example:

```
ip addr del 100.200.140.206/24 dev eth0:1
```

c. Run the following command as root on WCCHOST2:

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

Where, X:Y is the current interface used by ADMINVHN.

For example:

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

> **Note**
>
> Ensure that the CIDR representing the netmask and interface to be used to match the available network configuration in WCPHOST2.
>
> The name of the network interface device may something other than eth*X*, especially on systems with redundant bonded interfaces.

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

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

5. 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:

```
wcpedg_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:

```
wcpedg_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 WCPHOST2:

- `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.

- `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.

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.

1. Stop the Administration Server on WCCHOST2.
2. Stop the Node Manager on WCCHOST2.
3. Migrate the ADMINVHN virtual IP address to the second host:
   a. Run the following command as root on WCCHOST2 to check the virtual IP address at its CIDR:

      ```
      ip addr show dev ethX
      ```

      Where, `X` is the current interface used by ADMINVHN.

      For example:

      ```
      ip addr show dev eth0
      ```

   b. Run the following command as root on WCCHOST2 (where X:Y is the current interface used by ADMINVHN):

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

      Where, `X:Y` is the current interface used by ADMINVHN.

      For example:

      ```
      ip addr del 100.200.140.206/24 dev eth0:1
      ```

   c. Run the following command as root on WCCHOST1:

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

Where, `X:Y` is the current interface used by ADMINVHN.

For example:

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

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

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

   ```
   arping -b -A -c 3 -I eht0 100.200.140.206
   ```

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

   ```
   cd $NM_HOME
   ```

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

   ```
   cd $NM_HOME
   ```

8. Edit the `nodemanager.domains` file and add the reference to ASERVER_HOME.
9. Start the Node Manager on WCCHOST2 and restart the Node Manager on WCCHOST1.
10. Start the Administration Server on WCCHOST1.
11. Test that you can access the Administration Server on WCCHOST1 as follows:
    a. Test that you can use the WebLogic Remote Console to access the provider defined for this domain.
    b. Check that you can access and verify the status of components in Fusion Middleware Control using the following URL:

       ```
       https://ADMINVHN:9002/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 in to the WebLogic Remote Console to access the provider of this domain.
2. In the left navigation tree, expand **Domain**, and then **Environment**.
3. Click **Lock & Edit**.

4. Navigate to and edit the appropriate objects for your cluster type.
   a. For Static Clusters, navigate to **Servers** and click the name of the Managed Server you want to edit.
   b. For Dynamic Clusters, navigate to **Clusters** > **Server Templates**, and click on the name of the server template to be edited.
5. For each new Managed Server or Server Template to be edited:
   a. Click the **Configuration** tab, and then click the **Deployment** tab.
   b. Verify that the **Staging Directory Name** is set to the following:

      `MSERVER_HOME/servers/server_or_template_name/stage`

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

      If you use static clusters, update with the correct name of the Managed Server that you are editing.

      If you use dynamic clusters, leave the template name intact. For example: `/u02/oracle/config/domains/wcpedg_domain/servers/XYZ-server-template/stage`
   c. Update the **Upload Directory Name** to the following value:

      `ASERVER_HOME/servers/AdminServer/upload`

      Replace *ASERVER_HOME* with the directory path for the *ASERVER_HOME* directory.
   d. Click **Save**.
   e. Return to the Summary of Servers or Summary of Server Templates screen as applicable.
6. Repeat the previous steps for each of the new managed servers.
7. Navigate to and update the Upload Directory Name value for the AdminServer:
   a. Navigate to **Servers**, and select the AdminServer.
   b. Click the **Configuration** tab, and then click the **Deployment** Tab.
   c. Verify that the **Staging Directory Name** is set to the following absolute path:

      `ASERVER_HOME/servers/AdminServer/stage`
   d. Update the **Upload Directory Name** to the following absolute path:

      `ASERVER_HOME/servers/AdminServer/upload`

      Replace *ASERVER_HOME* with the directory path for the *ASERVER_HOME* directory.
   e. Click **Save**.
8. When you have modified all the appropriate objects, click **Activate Changes**.

> **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.

Setting the Front End Host and Port for a WebLogic Cluster

You must set the front-end HTTP host and port for the Oracle WebLogic Server cluster that hosts the Oracle SOA Suite servers. You can specify these values in the Configuration Wizard while you are specifying the properties of the domain. However, when you add a SOA Cluster as part of an Oracle WebCenter Portal enterprise deployment, Oracle recommends that you perform this task after you verify the SOA Managed Servers.

To set the frontend host and port from the WebLogic Remote Console:

1. Log in to the WebLogic Remote Console.
2. Open the **Edit Tree**.
3. Expand **Environment**.
4. Expand **Clusters**. On the **Clusters** page, click the cluster that you want to modify and then select the **HTTP** tab.
5. Set the following values:
   - **Frontend Host**: `wcp.example.com`
   - **Frontend HTTP Port**: `0`
   - **Frontend HTTPS Port**: `443`
6. Click **Save**.
7. Commit the changes in the shopping cart.
8. Restart the managed servers of the cluster.

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 the Frontend 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 SOA Suite enterprise deployment:

- Oracle Business Process Management and SOA Composer require access to the frontend load balancer URL when they attempt to retrieve role and security information through specific web instances. Some of these invocations require not only that the LBR certificate is added to the WebLogic Server's trust store but also that the appropriate identity key certificates are created for the SOA server's listen addresses.
- Oracle Service Bus performs invocations to endpoints exposed in the Load Balancer SSL virtual servers.
- 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 SOA 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.

Summary of Products with Specific Administration Roles

The following table lists the Fusion Middleware products that have specific administration roles, which must be added to the enterprise deployment administration group (`WCPAdministrators`), which you defined in the LDAP Authorization Provider for the enterprise deployment.

Use the information in the following table and the instructions in Adding a Product-Specific Administration Role to the Enterprise Deployment Administration Group to add the required administration roles to the enterprise deployment Administration group.

Product	Application Stripe	Administration Role to be Assigned
Oracle Web Services Manager	wsm-pm	policy.updater
WebCenter Portal	webcenter	s8bba98ff_4cbb_40b8_beee_296c916a23ed#-#Administrator
SOA Infrastructure	soa-infra	SOAAdmin

Summary of Oracle SOA Suite Products with Specific Administration Groups

Table 17-1 lists the Oracle SOA Suite products that need to use specific administration groups.

For each of these components, the common enterprise deployment Administration user must be added to the product-specific Administration group; otherwise, you won't be able to manage the product resources by using the enterprise manager administration user that you created in Provisioning an Enterprise Deployment Administration User and Group.

Use the information in Table 17-1 and the instructions in Adding the Enterprise Deployment Administration User to a Product-Specific Administration Group to add the required administration roles to the enterprise deployment Administration group.

Table 17-1 Oracle SOA Suite Products with a Product-Specific Administration Group

Product	Product-Specific Administration Group
Oracle Business Activity Monitoring	BAMAdministrator
Oracle Business Process Management	Administrators
Oracle Service Bus Integration	IntegrationAdministrators
MFT	OracleSystemGroup

> **Note**
>
> MFT requires a specific user, namely OracleSystemUser, to be added to the central LDAP. This user must belong to the OracleSystemGroup group. You must add both the user name and the user group to the central LDAP to ensure that MFT job creation and deletion work properly.

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_wcp`), 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_wcp`) 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, `WCPAdministrators`) 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_wcp` 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_wcp,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 persistent store provides a built-in, high-performance storage solution for WebLogic Server subsystems and services that require persistence.

For example, the JMS subsystem stores persistent JMS messages and durable subscribers, and the JTA Transaction Log (TLOG) stores information about the committed transactions that are coordinated by the server but may not have been completed. The persistent store supports persistence to a file-based store or to a JDBC-enabled database. Persistent stores' high availability is provided by server or service migration. Server or service migration requires that all members of a WebLogic cluster have access to the same transaction and JMS persistent stores (regardless of whether the persistent store is file-based or database-based).

For an enterprise deployment, Oracle recommends using JDBC persistent stores for transaction logs (TLOGs) and JMS.

This section analyzes the benefits of using JDBC versus File persistent stores and explains the procedure for configuring the persistent stores in a supported database. If you want to use File persistent stores instead of JDBC stores, the procedure for configuring them is also explained in this section.

Products and Components that use JMS Persistence Stores and TLOGs

Determining which installed FMW products and components utilize persistent stores can be done through the WebLogic Server Console in the Domain Structure navigation under ***DomainName*** > **Services** > **Persistent Stores**. The list indicates the name of the store, the store type (FileStore and JDBC), and the target of the store. The stores listed that pertain to MDS are outside the scope of this chapter and should not be considered.

These components (as applicable) use stores by default:

Component/Product	JMS Stores	TLOG Stores
SOA	Yes	Yes
WCC	Yes	Yes
WCP	No	No
WSM	No	No

Component/Product	JMS Stores	TLOG Stores
OAM	No	No
OIM	Yes	Yes

Typically, for an Oracle WebCenter Portal environment which includes Oracle WebCenter Content and Oracle SOA, the managed servers in their respective clusters will be the targets for the JMS and TLOGS data sources and new JDBC Persistent Stores.

JDBC Persistent Stores vs. File Persistent Stores

Oracle Fusion Middleware supports both database-based and file-based persistent stores for Oracle WebLogic Server transaction logs (TLOGs) and JMS. Before you decide on a persistent store strategy for your environment, consider the advantages and disadvantages of each approach.

> **Note**
>
> Regardless of which storage method you choose, Oracle recommends that for transaction integrity and consistency, you use the same type of store for both JMS and TLOGs.

About JDBC Persistent Stores for JMS and TLOGs

When you store your TLOGs and JMS data in an Oracle database, you can take advantage of the replication and high availability features of the database. For example, you can use Oracle Data Guard to simplify cross-site synchronization. This is especially important if you are deploying Oracle Fusion Middleware in a disaster recovery configuration.

Storing TLOGs and JMS data in a database also means that you do not have to identity a specific shared storage location for this data. Note, however, that shared storage is still required for other aspects of an enterprise deployment. For example, it is necessary for Administration Server configuration (to support Administration Server failover), for deployment plans, and for adapter artifacts, such as the File and FTP Adapter control and processed files.

If you are storing TLOGs and JMS stores on a shared storage device, then you can protect this data by using the appropriate replication and backup strategy to guarantee zero data loss, and you potentially realize better system performance. However, the file system protection is always inferior to the protection provided by an Oracle Database.

For more information about the potential performance impact of using a database-based TLOGs and JMS store, see Performance Considerations for TLOGs and JMS Persistent Stores.

Performance Considerations for TLOGs and JMS Persistent Stores

One of the primary considerations when you select a storage method for Transaction Logs and JMS persistent stores is the potential impact on performance. This topic provides some guidelines and details to help you determine the performance impact of using JDBC persistent stores for TLOGs and JMS.

Performance Impact of Transaction Logs Versus JMS Stores

For transaction logs, the impact of using a JDBC store is relatively small, because the logs are very transient in nature. Typically, the effect is minimal when compared to other database operations in the system.

On the other hand, JMS database stores can have a higher impact on performance if the application is JMS intensive.

Factors that Affect Performance

There are multiple factors that can affect the performance of a system when it is using JMS DB stores for custom destinations. The main ones are:

- Custom destinations involved and their type
- Payloads being persisted
- Concurrency on the SOA system (producers on consumers for the destinations)

Depending on the effect of each one of the above, different settings can be configured in the following areas to improve performance:

- Type of data types used for the JMS table (using raw versus lobs)
- Segment definition for the JMS table (partitions at index and table level)

Impact of JMS Topics

If your system uses Topics intensively, then as concurrency increases, the performance degradation with an Oracle RAC database will increase more than for Queues. In tests conducted by Oracle with JMS, the average performance degradation for different payload sizes and different concurrency was less than 30% for Queues. For topics, the impact was more than 40%. Consider the importance of these destinations from the recovery perspective when deciding whether to use database stores.

Impact of Data Type and Payload Size

When you choose to use the RAW or SecureFiles LOB data type for the payloads, consider the size of the payload being persisted. For example, when payload sizes range between 100b and 20k, then the amount of database time required by SecureFiles LOB is slightly higher than for the RAW data type.

More specifically, when the payload size reach around 4k, then SecureFiles tend to require more database time. This is because 4k is where writes move out-of-row. At around 20k payload size, SecureFiles data starts being more efficient. When payload sizes increase to more than 20k, then the database time becomes worse for payloads set to the RAW data type.

One additional advantage for SecureFiles is that the database time incurred stabilizes with payload increases starting at 500k. In other words, at that point it is not relevant (for SecureFiles) whether the data is storing 500k, 1MB or 2MB payloads, because the write is asynchronized, and the contention is the same in all cases.

The effect of concurrency (producers and consumers) on the queue's throughput is similar for both RAW and SecureFiles until the payload sizes reach 50K. For small payloads, the effect on varying concurrency is practically the same, with slightly better scalability for RAW. Scalability is better for SecureFiles when the payloads are above 50k.

Impact of Concurrency, Worker Threads, and Database Partioning

Concurrency and worker threads defined for the persistent store can cause contention in the RAC database at the index and global cache level. Using a reverse index when enabling multiple worker threads in one single server or using multiple Oracle WebLogic Server clusters can improve things. However, if the Oracle Database partitioning option is available, then global hash partition for indexes should be used instead. This reduces the contention on the index and the global cache buffer waits, which in turn improves the response time of the application. Partitioning works well in all cases, some of which will not see significant improvements with a reverse index.

Using JDBC Persistent Stores for TLOGs and JMS in an Enterprise Deployment

This section explains the guidelines to use JDBC persistent stores for transaction logs (TLOGs) and JMS. It also explains the procedures to configure the persistent stores in a supported database.

Recommendations for TLOGs and JMS Datasource Consolidation

To accomplish data source consolidation and connection usage reduction, use a single connection pool for both JMS and TLOGs persistent stores.

Oracle recommends you to reuse the `WLSSchemaDatasource` as is for TLOGs and JMS persistent stores under non-high workloads and consider increasing the `WLSSchemaDatasource` pool size. Reuse of datasource forces to use the same schema and tablespaces, and so the `PREFIX_WLS_RUNTIME` schema in the `PREFIX_WLS` tablespace is used for both TLOGs and JMS messages.

High stress (related with high JMS activity, for example) and contention in the datasource can cause stability and performance problems. For example:

- High contention in the DataSource can cause persistent stores to fail if no connections are available in the pool to persist JMS messages.
- High Contention in the DataSource can cause issues in transactions if no connections are available in the pool to update transaction logs.

For these cases, use a separate datasource for TLOGs and stores and a separate datasource for the different stores. You can still reuse the `PREFIX_WLS_RUNTIME` schema but configure separate custom datasources to the same schema to solve the contention issue.

Roadmap for Configuring a JDBC Persistent Store for TLOGs

The following topics describe how to configure a database-based persistent store for transaction logs.

1. Creating a User and Tablespace for TLOGs
2. Creating GridLink Data Sources for TLOGs and JMS Stores
3. Assigning the TLOGs JDBC Store to the Managed Servers

> **Note**
>
> Steps 1 and 2 are optional. To accomplish data source consolidation and connection usage reduction, you can reuse `PREFIX_WLS` tablespace and `WLSSchemaDatasource` as described in Recommendations for TLOGs and JMS Datasource Consolidation.

Roadmap for Configuring a JDBC Persistent Store for JMS

The following topics describe how to configure a database-based persistent store for JMS.

1. Creating a User and Tablespace for JMS
2. Creating GridLink Data Sources for TLOGs and JMS Stores
3. Creating a JDBC JMS Store

4. Assigning the JMS JDBC store to the JMS Servers
5. Creating the Required Tables for the JMS JDBC Store

> **Note**
>
> Steps 1 and 2 are optional. To accomplish data source consolidation and connection usage reduction, you can reuse `PREFIX_WLS` tablespace and `WLSSchemaDatasource` as described in Recommendations for TLOGs and JMS Datasource Consolidation.

Creating a User and Tablespace for TLOGs

Before you can create a database-based persistent store for transaction logs, you must create a user and tablespace in a supported database.

1. Create a tablespace called `tlogs`.

   For example, log in to SQL*Plus as the `sysdba` user and run the following command:

   ```
   SQL> create tablespace tlogs
           logging datafile 'path-to-data-file-or-+asmvolume'
           size 32m autoextend on next 32m maxsize 2048m extent management local;
   ```

2. Create a user named `TLOGS` and assign to it the `tlogs` tablespace.

   For example:

   ```
   SQL> create user TLOGS identified by password;

   SQL> grant create table to TLOGS;

   SQL> grant create session to TLOGS;

   SQL> alter user TLOGS default tablespace tlogs;

   SQL> alter user TLOGS quota unlimited on tlogs;
   ```

Creating a User and Tablespace for JMS

Before you can create a database-based persistent store for JMS, you must create a user and tablespace in a supported database.

1. Create a tablespace called `jms`.

   For example, log in to SQL*Plus as the `sysdba` user and run the following command:

   ```
   SQL> create tablespace jms
           logging datafile 'path-to-data-file-or-+asmvolume'
           size 32m autoextend on next 32m maxsize 2048m extent management local;
   ```

2. Create a user named `JMS` and assign to it the `jms` tablespace.

   For example:

   ```
   SQL> create user JMS identified by password;

   SQL> grant create table to JMS;
   ```

```
SQL> grant create session to JMS;

SQL> alter user JMS default tablespace jms;

SQL> alter user JMS quota unlimited on jms;
```

Creating GridLink Data Sources for TLOGs and JMS Stores

Before you can configure database-based persistent stores for JMS and TLOGs, you must create two data sources: one for the TLOGs persistent store and one for the JMS persistent store.

For an enterprise deployment, you should use GridLink data sources for your TLOGs and JMS stores. To create a GridLink data source:

1. Sign in to the 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**, and enter the following:

Table 17-2 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, `soaedg.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.
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.

Table 17-2 (Cont.) GridLink Data Source Properties

Properties	Description
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 `WLSSchemaDatasource` 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 in to 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 in to 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**
>
> The length of the prefix name must not exceed 30 characters for DB versions that are below 12.2.x.x.x.

> **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 steps 3 through 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 in to 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 steps 3 to 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 Performance Considerations for TLOGs and JMS Persistent Stores, and decide which table features are appropriate for your environment.

   There are three Oracle DB schema definitions provided in this release and were extracted for review in the previous step. The basic definition includes the RAW data type without any partition for indexes. The second uses the blob data type, and the third uses the blob data type and secure files.
2. Create a domain-specific well-named folder structure for the custom `DDL` file on shared storage. The *ORACLE_RUNTIME* shared volume is recommended so it is available to all servers.

Example:

```
mkdir -p ORACLE_RUNTIME/domain_name/ddl
```

3. Create a `jms_custom.ddl` file in new shared `ddl` folder based on your requirements analysis.

   For example, to implement an optimized schema definition that uses both secure files and hash partitioning, create the `jms_custom.ddl` file with the following content:

   ```
   CREATE TABLE $TABLE (
     id     int  not null,
     type   int  not null,
     handle int  not null,
     record blob not null,
   PRIMARY KEY (ID) USING INDEX GLOBAL PARTITION BY HASH (ID) PARTITIONS 8)
   LOB (RECORD) STORE AS SECUREFILE (ENABLE STORAGE IN ROW);
   ```

   This example can be compared to the default schema definition for JMS stores, where the RAW data type is used without any partitions for indexes.

   Note that the number of partitions should be a power of two. This ensures that each partition is of similar size. The recommended number of partitions varies depending on the expected table or index growth. You should have your database administrator (DBA) analyze the growth of the tables over time and adjust the tables accordingly. See Partitioning Concepts in *Database VLDB and Partitioning Guide*.

4. Use the WebLogic Remote Console to edit the existing JDBC Store you created earlier; create the table that is used for the JMS data:
   a. Log in to 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.

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 Portal 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 17-3 lists the static artifacts to back up in a typical Oracle WebCenter Portal enterprise deployment.

Table 17-3 Static Artifacts to Back Up in the Oracle WebCenter Portal Enterprise Deployment

Type	Host	Tier
Database Oracle home	DBHOST1 and DBHOST2	Data Tier
Oracle Fusion Middleware Oracle home	WEBHOST1 and WEBHOST2	Web Tier
Oracle Fusion Middleware Oracle home	WCCHOST1 and WCCHOST2 (or NAS Filer)	Application Tier
Installation-related files	WEBHOST1, WEHOST2, and shared storage	N/A

Table 17-4 lists the runtime artifacts to back up in a typical Oracle WebCenter Portal enterprise deployment.

Table 17-4 Run-Time Artifacts to Back Up in the Oracle WebCenter Portal 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

Table 17-4 (Cont.) Run-Time Artifacts to Back Up in the Oracle WebCenter Portal Enterprise Deployment

Type	Host	Tier
OHS/OTD Configuration directory	WEBHOST1 and WEBHOST2	Web Tier

Configuration and Management Tasks for an Oracle WebCenter Portal Enterprise Deployment

These are some of the key configuration and management tasks that you likely need to perform on an Oracle WebCenter Portal enterprise deployment.

Deploying Oracle SOA Suite Composite Applications to an Enterprise Deployment

Oracle SOA Suite applications are deployed as composites, consisting of different kinds of Oracle SOA Suite components. SOA composite applications include the following:

- Service components such as Oracle Mediator for routing, BPEL processes for orchestration, BAM processes for orchestration (if Oracle BAM Suite is also installed), human tasks for workflow approvals, spring for integrating Java interfaces into SOA composite applications, and decision services for working with business rules.
- Binding components (services and references) for connecting SOA composite applications to external services, applications, and technologies.

These components are assembled into a single SOA composite application.

When you deploy an Oracle SOA Suite composite application to an Oracle SOA Suite enterprise deployment, be sure to deploy each composite to a specific server or cluster address and not to the load balancer address (`soa.example.com`).

Deploying composites to the load balancer address often requires direct connection from the deployer nodes to the external load balancer address. As a result, you have to open additional ports in the firewalls.

For more information about Oracle SOA Suite composite applications, see the following sections in *Administering Oracle SOA Suite and Oracle Business Process Management Suite*:

- Deploying SOA Composite Applications
- Monitoring SOA Composite Applications
- Managing SOA Composite Applications

Using Shared Storage for Deployment Plans and SOA Infrastructure Applications Updates

When you redeploy a SOA infrastructure application or resource adapter within the SOA cluster, the deployment plan along with the application bits should be accessible to all servers in the cluster.

SOA applications and resource adapters are installed using nostage deployment mode. Because the administration sever does not copy the archive files from their source location when the nostage deployment mode is selected, each server must be able to access the same deployment plan.

To ensure deployment plan location is available to all servers in the domain, use the Deployment Plan home location described in File System and Directory Variables Used in This Guide and represented by the *DEPLOY_PLAN_HOME* variable in the *Enterprise Deployment Workbook*.

Managing Database Growth in an Oracle SOA Suite Enterprise Deployment

When the amount of data in the Oracle SOA Suite database grows very large, maintaining the database can become difficult, especially in an Oracle SOA Suite enterprise deployment where potentially many composite applications are deployed.

See the following sections in *Administering Oracle SOA Suite and Oracle Business Process Management Suite*:

- Developing a Database Growth Management Strategy
- Managing Database Growth

Managing the JMS Messages in a SOA Server

There are several procedures to manage JMS messages in a SOA server. You may need to perform these procedures in some scenarios, for example, to preserve the messages during a scale-in operation.

This section explains some of these procedures in detail.

Draining the JMS Messages from a SOA Server

The process of draining the JMS messages helps you clear out the messages from a particular WebLogic server. A basic approach to drain stores consists of stopping the message production in the appropriate JMS Servers and allowing the applications to consume the messages.

This procedure, however, is application dependent, and could take an unpredictable amount of time. As an alternative, general instructions are provided here for saving the current messages from their current JMS destinations and, when/if required, importing them into a different server.

The draining procedure is useful in scale-in/down scenarios, where the size of the cluster is reduced by removing one or more servers. You can ensure that no messages are lost by draining the messages from the server that you delete, and then importing them into another server in the cluster.

You can also use this procedure in some disaster recovery maintenance scenarios, when the servers are started in a secondary location by using an Snapshot Standby database. In this case, you may need to drain the messages from the domain before starting it in the secondary location to avoid their consumption in the standby domain when you start the domain (otherwise, duplicate executions could take place). You cannot import messages in this scenario.

To drain the JMS messages from a server, perform the following steps:

1. Stop a new workload by pausing production for the JMS Server. You must do this activity for each JMS Server of the server that is affected in the operation:

a. In the WebLogic Remote Console, open the **Monitoring Tree**.

b. Navigate to **Environment** > **Servers**.

c. In the server that you want to delete, navigate to the **Services** > **Messaging** > **JMS Runtime** > **JMS Servers**.

d. Select the *JMS Server* of the server that you want to delete.

e. Click **Production**, and then click **Pause**.

2. Drain the messages from the destinations. To drain the JMS messages, you can let applications consume the pending messages. However, this task is application dependent and may take time. Hence, Oracle recommends you to export the messages of each destination. Verify which destinations have messages:

   a. In the WebLogic Remote Console, open the **Monitoring Tree**.

   b. Navigate to **Environment** > **Servers**.

   c. In the server that you want to delete, navigate to **Services** > **Messaging** > **JMS Runtime** > **JMS Servers**.

   d. For each JMS Server, look whether the destination members have current messages. Identify the destination name, its JMS Module and JMS Server.

   e. Repeat this activity for each JMS Server that is running in the server that you want to delete.

   - **Drain messages from queues**: For those queue destinations that have current messages:

     a. In the WebLogic Remote Console, open the **Monitoring Tree**.

     b. Navigate to **Dashboards** and click **JMS Destinations dashboard**.

     c. Select the queue that you want to export messages from.

     d. In the **Messages** tab, select **Export** > **Export All** and export the messages to a file. Make a note of the file name for later use.

     e. Delete the exported messages by using the **Delete All** option. This step is important to avoid message duplications.

   - **Drain messages from topics**

     Oracle recommends you to drain and import messages from topics only if they have a critical business impact. See Table 17-5 for details about the purpose and business impact for each topic. Only the loss of messages in the topic **dist_EDNTopic_auto**, used by EDN, has a business impact.

Table 17-5 Details of the Purpose and Business Impact for Each Topic of a Component

Component	JMS Module	JMS Topic Name	Purpose	Business Impact of Message Loss
BPM	BPMJMSModule	dist_MeasurementTopic_auto	Used for publishing process metrics messages to the internal process star schema.	Low impact. Will affect some dashboard number appearing in the PCS workspace dashboards and BAM dashboards based on the process star schema data object.
BPM	BPMJMSModule	dist_PeopleQueryTopic_auto	Used for updating logical group memberships.	Low impact. The group membership will be recalculated based on a scheduler.
SOA	SOAJMSModule	dist_B2BBroadcastTopic_auto	Used by B2B, messages are meant to be consumed immediately.	No impact.
SOA	SOAJMSModule	dist_EDNTopic_auto	Used for EDN, contains event messages for applications.	Business impact. Applications that consume these EDN event messages will lose them.
SOA	SOAJMSModule	dist_TenantTopic_auto	No longer used.	No impact.
SOA	SOAJMSModule	dist_XmlSchemaChangeNotificationTopic_auto	No longer used.	No impact.
Insight	ProcMonJMSModule	dist_ProcMonActivationTopic_auto	Used by Insight for lifecycle operations - for activating an insight model across different nodes of the cluster.	No impact.
BAM	BAMJMSSystemResource	dist_oracle.beam.cqs.activedata_auto	Not used in production.	No impact.

Table 17-5 (Cont.) Details of the Purpose and Business Impact for Each Topic of a Component

Component	JMS Module	JMS Topic Name	Purpose	Business Impact of Message Loss
BAM	BAMJMSSystem Resource	dist_oracle.beam.persistence.activedata_auto	Data change notifications sent from persistence to the continuous query processor in support of active-data queries.	Low impact. Message loss could only cause incorrect data to be displayed in the active-data dashboards. Refreshing the dashboards or restarting the active-query will restore the correct data.
BAM	BAMJMSSystem Resource	dist_oracle.beam.server.event.reportcache.changelist_auto	Data changes sent from the report cache to the active-data dashboards.	
BAM	BAMJMSSystem Resource	dist_oracle.beam.server.metadatachange_auto	Metadata changes sent to the downstream listeners if artifacts (queries, views, dashboards) are modified.	
MFT	MFTJMSModule	dist_MFTSystemEventTopic_auto	Used for publishing events that require synch in all the nodes, such as activation of the listening source, adding the PGP key, Mbean property changes, and so on.	Low impact. These messages are very short lived and their frequency is low. If there is any message loss, a restart ensures that all nodes in sync.

Follow these steps drain messages from the topics:

a. In the WebLogic Remote Console, open the **Monitoring Tree**.

b. Navigate to **Dashboards** and click the **JMS Destinations dashboard**.

c. Select the topic that you want to delete and navigate to its **Subscribers**.

d. Select the Durable Subscriber that has current messages and click**Show Messages**.

e. Click **Export** > **Export All** and export the messages to a file. Make a note of the file name for later use.

f. Delete the exported messages from the subscriber by clicking **Delete** > **Delete All**. This step is important to avoid message duplications.

g. Repeat the export process for any subscriber in the topic that has current messages.

18

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 Service Migration in an Enterprise Deployment

Using Automatic Service Migration (ASM) in an Enterprise Deployment has implications in the infrastructure and configuration requirements.

The implications are:

- The resources used by servers must be accessible to both the original and failover system

  In its initial status, resources are accessed by the original server or service. When a server or service is failed over/restarted in another system, the same resources (such as external resources, databases, and stores) must be available in the failover system. Otherwise, the service cannot resume the same operations. It is for this reason, that both whole server and service migration require that all members of a WebLogic cluster have access to the same transaction and JMS persistent stores.

  Oracle allows you to 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. Alternatively, you can use shared storage. When you configure persistent stores properly in the database or in shared storage, you must ensure that if a failover occurs (whole server migration or service migration), the failover system is able to access the same stores without any manual intervention.

- Leasing Datasource

  Service migration requires the configuration of a leasing datasource that is used by servers to store *alive* timestamps. These timestamps are used to determine the health of a server or service, and are key to the correct behavior of server and service migration (they are used to marks servers or services as *failed* and trigger failover).

> **Note**
>
> Oracle does not recommend that you use consensus leasing for HA purposes.

Understanding Which Products and Components Require Whole Server Migration and Service Migration

The following table summarizes the list of FMW products and components that benefit from use of a migration capability and indicates the best-practice recommendation for this release. Components listed as migratable can use either Whole Server or Automatic 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 Web Services Manager (OWSM)	NO	NO
Oracle WebCenter Portal	NO	NO
Oracle WebCenter Portal Portlets and Pagelet Producers	NO	NO
Oracle WebCenter Content	YES	YES (Recommended)
Oracle WebCenter Inbound Refinery	NO	NO

Component	Whole Server Migration (WSM)	Automatic Service Migration (ASM)
Oracle SOA Suite	YES	YES (Recommended)
Oracle Enterprise Scheduler	NO	NO

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 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, 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 18-1 (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

You may need to configure automatic service migration for specific services in an enterprise deployment.

Note

The Automatic Service Migration feature is part of Oracle Integration Continuous Availability. For more details on Oracle SOA Suite for Middleware Options, see *Oracle Fusion Middleware Licensing Information*.

Setting the Leasing Mechanism and Data Source for an Enterprise Deployment Cluster

Before you can configure automatic service migration, you must verify the leasing mechanism and data source that is used by the automatic service migration feature. You must configure the leasing mechanism and datasource for both static and dynamic clusters.

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

1. Log in to the WebLogic Remote Console.
2. Navigate to the **Edit Tree**.
3. In the structure tree, expand **Environment** > **Clusters**.

   The **Summary of Clusters** page appears.
4. 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.

After you complete the database leasing configuration, continue with the configuration of the service migration:

- See Configuring Automatic Service Migration

Configuring Automatic Service Migration

After you have configured the leasing for the cluster as described in Setting the Leasing Mechanism and Data Source for an Enterprise Deployment Cluster, you can configure automatic service migration for specific services in an enterprise deployment. The following sections explain how to configure and validate Automatic Service Migration for static clusters.

Changing the Migration Settings for the Managed Servers in the Cluster

After you set the leasing mechanism and data source for the cluster, you can then enable automatic JTA migration for the Managed Servers that you want to configure for service migration. Note that this topic applies only if you are deploying JTA services as part of your enterprise deployment.

To change the migration settings for the Managed Servers in each cluster:

1. Log in to 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.

Based on these guidelines, the following policies are recommended for an Oracle WebCenter enterprise topology:

- SOA_Cluster: **failure-recovery**
- WCC_Cluster: **failure-recovery**

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 in to the WebLogic Remote Console.
2. Navigate to the **Edit Tree**.
3. In the structure tree, expand **Environment** > **Migratable Targets**.

4. In the **Service Migration Policy** drop-down menu, select the appropriate policy for the cluster. See About Selecting a Service Migration Policy.
5. 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.
6. Click **Save**.
7. Repeat *Step 2* to *Step 6* for each of the additional Managed Servers in the cluster.
8. Commit changes in the shopping cart.
9. 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 in to 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 the 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 Oracle WebLogic Remote Console and refresh the table of migratable targets; verify that the migratable targets are transferred to the remaining, running Managed Server in the cluster:
   - Verify that the 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 in to 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.
> 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.