Oracle Portal Enterprise Deployment Guide: 11.1.1.2

Oracle Maximum Availability Architecture White Paper
December 2009

Maximum Availability Architecture

Oracle Best Practices For High Availability

ORACLE

Enterprise Deployment Overview

What is an Enterprise Deployment?

An enterprise deployment is an Oracle best practices blueprint based on proven Oracle high-availability and security technologies and recommendations for Oracle Fusion Middleware. The high-availability best practices described in this document make up one of several components of high-availability best practices for all Oracle products across the entire technology stack—Oracle Database, Oracle Fusion Middleware, Oracle Applications, Oracle Collaboration Suite, and Oracle Grid Control.

An Oracle Fusion Middleware enterprise deployment:

- Considers various business service level agreements (SLA) to make high-availability best practices as widely applicable as possible
- Leverages database grid servers and storage grid with low-cost storage to provide highly resilient, lower cost infrastructure
- Uses results from extensive performance impact studies for different configurations to ensure that the high-availability architecture is optimally configured to perform and scale to business needs
- Enables control over the length of time to recover from an outage and the amount of acceptable data loss from a natural disaster
- Evolves with each Oracle version and is completely independent of hardware and operating system

For more information on high availability practices, visit:

http://www.oracle.com/technology/deploy/availability/htdocs/maa.htm

Terminology

Table 1-1 provides definitions for some of the terms that define the architecture of an Oracle Fusion Middleware environment:

Table 1-1 Oracle Fusion Middleware Architecture Terminology

Term	Definition
Oracle Base	Oracle Mount point, all binaries and configuration information are in relation to this mount point.
Oracle Fusion Middleware home	A Middleware home consists of the Oracle WebLogic Server home, and, optionally, one or more Oracle homes.
WebLogic Server home	A WebLogic Server home contains installed files necessary to host a WebLogic Server. The WebLogic Server home directory is a peer of other Oracle home directories underneath the Middleware home directory.
Oracle home	An Oracle home contains installed files necessary to host a specific product. For example, the Oracle Identity Management Oracle home contains a directory that contains binary and library files for Oracle Identity Management. An Oracle home resides within the directory structure of the Middleware home. Each Oracle home can be associated with multiple Oracle instances or Oracle WebLogic Server

	domains.
Oracle instance	An Oracle instance contains one or more system components, such as Oracle Web Cache, Oracle HTTP Server, or Oracle Internet Directory. The system components in an Oracle instance must reside on the same machine. An Oracle instance directory contains updatable files, such as configuration files, log files, and temporary files. The directory structure of an Oracle instance is separate from the directory structure of the Oracle home. It can reside anywhere; it need not be within the Middleware home directory.
Oracle WebLogic Server domain	A WebLogic Server domain is a logically related group of Java components. A WebLogic Server domain includes a special WebLogic Server instance called the Administration Server, which is the central point from which you configure and manage all resources in the domain. Usually, you configure a domain to include additional WebLogic Server instances called Managed Servers. You deploy Java components, such as Web applications, EJBs, and Web services, and other resources to the Managed Servers and use the Administration Server for configuration and management purposes only. Managed Servers in a WebLogic Server domain can be grouped together into a cluster.
Oracle Fusion Middleware farm	Oracle Enterprise Manager Fusion Middleware

	Control is a Web browser-based, graphical user interface that you can use to monitor and administer an Oracle Fusion Middleware farm. An Oracle Fusion Middleware farm is a collection of components managed by Fusion Middleware Control. It can contain a WebLogic Server domain, one or more Managed Servers and the Oracle Fusion Middleware system components that are installed, configured, and running in the domain.

Benefits of Oracle Recommendations

The Oracle Fusion Middleware configurations discussed in this guide are designed to ensure security of all transactions, maximize hardware resources, and provide a reliable, standards-compliant system for enterprise computing with a variety of applications. The security and high availability benefits of the Oracle Fusion Middleware configurations are realized through isolation in firewall zones and replication of software components.

Built-in Security

The Enterprise Deployment architectures are secure because every functional group of software components is isolated in its own DMZ, and all traffic is restricted by protocol and port. The following characteristics ensure security at all needed levels, as well as a high level of standards compliance:

- All external communication received on port 80 is redirected to port 443.
- Communication from external clients does not go beyond the load balancing router level.
- No direct communication from the load balancing router to the data tier DMZ is allowed.

- Components are separated between DMZs on the Web Tier, application tier, and the directory tier.
- Direct communication between two firewalls at any one time is prohibited.
- If a communication begins in one firewall zone, it must end in the next firewall zone.
- Oracle Internet Directory is isolated in the directory tier DMZ.
- Identity Management components are in the DMZ.
- All communication between components across DMZs is restricted by port and protocol, according to firewall rules.

High Availability

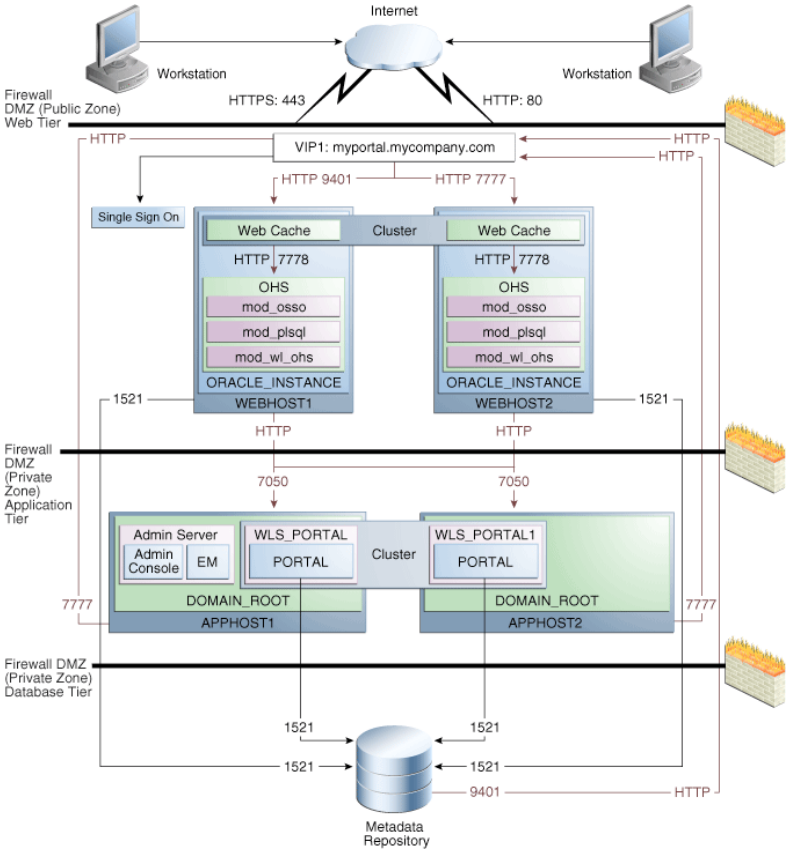
The Enterprise Deployment architectures are highly available, because each component or functional group of software components is replicated on a different computer, and configured for component-level high availability.

The Enterprise Deployment Reference Topology

The instructions and diagrams in this guide describe a reference topology, to which variations may be applied.

This guide provides instructions for creating the Application and Web Server tiers of the myPortal company architecture, distributing the software components into the Enterprise Deployment architecture depicted below.

At the end of this document the following infrastructure will have been configured.



Understanding the Web Tier

The Web Tier is in the DMZ Public Zone. Web Cache and HTTP Servers are deployed in the Web Tier.

Web Cache is the first point on entry into the site, it performs two functions; Its primary function is to serve static web content from its cache, much faster than could be achieved by the Oracle HTTP Servers alone. If Web Cache does not have a cacheable page in its cache or that page is not current, then Web Cache will request the page from the attached Oracle HTTP server(s).

The second function of Web Cache is to load balance requests between several Oracle HTTP Servers.

The Oracle HTTP Server is responsible for assembling pages requested by the user. Page assembly is not always straightforward however. Depending on how the page is made up the Oracle HTTP Server will perform one of the following:

- If the page is a simple HTML document, then the Web Tier will find and return the document.
- If the web page needs to be assembled by executing a Java J2EE application then the Oracle Web Tier will route the request to Oracle WebLogic server, which after processing the request will send the result back to the user via the Oracle Web Tier.
- If the web page needs to be assembled by executing some other application such as PLSQL or CGI then the Oracle Web Tier will route the request to the appropriate application, and once that application has processed the request, it will send the result back to the user via the Oracle Web Tier.
- If the page being requested is protected, then the Oracle Web Server will invoke Oracle Identity Management (Single Sign On) to ensure that the user is authorized to view the requested page.

The Oracle HTTP Server uses an Apache module called mod_wl_ohs to route requests to WebLogic Managed Servers. In this implementation the WebLogic managed Servers

WLS_PORTAL and WLS_PORTAL1 are clustered together and mod_wl_ohs will load balance requests amongst them.

When a request needs authorization the Oracle HTTP Server will intercept the request and if necessary redirect the browser to the Oracle Single Sign Server(s) for authentication.

The Oracle Web Caches are clustered together to provide a global cache which is consistent across nodes.

In this implementation user requests are received at the load balancer on port 443. These requests are passed on to the Oracle Web Caches using the HTTP protocol on port 7777. If the originating request is using the SSL protocol (HTTPS) then the load balancer will strip off the encryption prior to sending it into the site. It will encrypt traffic returning to the user. This enables the site to operate in the most efficient manner possible.

Understanding the Application Tier

The application tier is where the main application logic resides. Oracle WebLogic servers resident in this tier, are responsible for the application logic. Sometimes this application logic takes the form of C processes, which are started by the WebLogic application. In this scenario WebLogic is responsible for starting/stopping and channeling work to these C processes. An example of this behavior is the Forms runtime process.

Requests are routed to the application tier from the Oracle Web Tier by mod_wl_ohs.

Understanding the Database Tier

Oracle Portal is an application built mainly in PLSQL. The Oracle application tier, interacts with the Portal Metadata repository to construct web pages, this metadata is stored within an Oracle Database, along with user content. Because the database is such an integral part of the infrastructure, this database also needs to be highly available. Oracle therefore recommends that the metadata repository be placed into an Oracle Real Application Clusters database.

Approach

Installing an Enterprise deployment is complex; to simplify this and to provide intermediary checkpoints this guide uses the following approach:

1. Install Oracle Portal on APPHOST1.
2. Fully configure APPHOST1 to support access via the load balancer.
3. Install APPHOST2.
4. Fully configure APPHOST2 to support access via the load balancer.

The above steps include configuring Oracle Web Cache and the Oracle HTTP server. If a simple HA configuration is desired then no further steps are necessary. If however the full enterprise deployment as described above is being implemented the following must also be done.

5. Install Oracle Web Tier on WEBHOST1.
6. Fully configure Web Tier on WEBHOST1.
7. Install Oracle Web Tier on WEBHOST2.
8. Fully configure Web Tier on WEBHOST2.
9. Disable/Remove the Web tier components on APPHOST1.
10. Disable/Remove the Web tier components on APPHOST2.

What to Install

The following table identifies the source for installation of each software component:

Component	CD
Oracle Database	Oracle Database CS (10.2.0.4 or 11.1.0.7 and 11.2)
Oracle WebLogic Server	WebLogic Server 10.3 CD

Oracle Portal	Oracle Portal, Forms, Reports and Discoverer CD (11.1.1.2.0)
Repository Creation Utility	Oracle Fusion Middleware Repository Creation Utility CD (11.1.1.2.0)
Oracle Web Tier	Oracle Fusion Middleware Web Tier and Utilities CD (11.1.1.2.0)

Third Party Components of Enterprise Deployments

load balancer

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

- Ability to load-balance traffic to a pool of real servers through a virtual host name: Clients access services using the virtual host name (instead of using actual host names). The load balancer can then load balance requests to the servers in the pool.
- Port translation configuration
- Monitoring of ports (HTTP and HTTPS)
- Virtual servers and port configuration: Ability to configure virtual server names and ports on your external load balancer, and the virtual server names and ports must meet the following requirements:
- The load balancer should allow configuration of multiple virtual servers. For each virtual server, the load balancer should allow configuration of traffic management on more than one port. For example, for OracleAS Clusters, 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.
- Resource monitoring / port monitoring / process failure detection: The load balancer must be able to detect service and node failures (through notification or some other means) and to stop directing non-Oracle Net traffic to the failed node. If your external load balancer has the ability to automatically detect failures, you should use it.
- Fault tolerant mode: It is highly recommended that you configure the load balancer to be in fault-tolerant mode.
- Other: 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.
- SSL acceleration (this feature is recommended, but not required)

Managing port numbers

Many Oracle Fusion Middleware components and services use ports. As an administrator, it is important to know the port numbers used by these services, and to ensure that the same port number is not used by two services on your host.

Most port numbers are assigned during installation.

Note: It is important that any traffic going from the Oracle HTTP servers to the WebLogic servers has access through any firewalls.

Oracle Single Sign On

The Oracle Portal topology requires access to a highly available Enterprise Deployment Identity Management. Oracle Portal uses Single Sign On 10g (minimum version 10.1.4.3). Creating a highly available Identity Management topology is beyond the scope of this document. Further information can be found at:

http://download.oracle.com/docs/cd/B14099_19/core.1012/b13998/selecting.htm#sthref75

With specific installation instructions located at:

http://download.oracle.com/docs/cd/B14099_19/core.1012/b13998/security.htm#CDDFHGCF

Other variants of the above topology using 11g stack (for example for OID) are possible and supported to work with this configuration but detailed description of these is out of scope of this topology

Understanding the Directory Structure

Once the installation is complete the following directory structure will exist:

Directory	Shared	Purpose
/u01/app/oracle	N	Oracle Base Directory
/u01/app/oracle/product/fmw	N	Middleware Home Directory
/u01/app/oracle/product/fmw/Portal	N	Oracle Home (application tier)
/u01/app/oracle/product/fmw/web	N	Oracle Home (Web Tier)
/u01/app/oracle/product/fmw/user_projects	N	Domain Home Directory
/u01/app/oracle/admin/Portal1	N	Oracle Instance (APPHOST1)
/u01/app/oracle/admin/Portal2	N	Oracle Instance (APPHOST2)
/u01/app/oracle/admin/web1	N	Oracle Instance (WEBHOST1)
/u01/app/oracle/admin/web2	N	Oracle Instance (WEBHOST2)

Special Installation and Configuration Considerations

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

The table below lists the ports used in the Oracle Portal topology, including the ports that need to be opened on the firewalls in the topology.

Firewall notation:

- FW0 refers to the outermost firewall.
- FW1 refers to the firewall between the Web Tier and the application tier.
- FW2 refers to the firewall between the application tier and the directory tier.

Type	Firewall	Ports	Protocol	Inbound/ Outbound	Comments
Browser request	FW0	443	HTTPS/LBR1	In/out	
Browser request	FW0	80	HTTP/LBR1	In/out	
LBR to WC	FW1	7777 9401 9402	HTTP	NA	NA
WC to OHS	NA	7778	HTTP	In/out	
OHS to WLS	FW2	7050	HTTP	In/out	
Admin Console Access	Depends	7001	HTTP/Admin Server-EM and t3	In/out	Admin console. However, administrators will not be allowed to access the Admin Console from anywhere. It is unlikely for example that administrators will be allowed to access the Admin Console from outside of the organisation.

Database Access	FW2	1521	SQLNET	In/out	
WC Invalidation Requests	FW3	9401	HTTP	Out	Database sends invalidation requests to loadbalancer.
Node Manager	NA	NA	TCP/IP		NA

Assumptions

For the remainder of this document the following assumptions have been made, when building an Enterprise deployment, the values listed below (especially usernames/passwords) should be changed.

Site Names

The following site names are used by this Enterprise Deployment:

Name	Purpose
myPortal.mycompany.com	Portal Site Name
login.mycompany.com	Single Sign On

Ports

The following Ports are assumed for the purposes of this document. All of these ports can be changed during the installation.

Purpose	Host(s)	Port	Comment
myPortal.mycompany.com	load balancer	443	SSL port on the load balancer
myPortal.mycompany.com	load balancer	7777	HTTP port on load balancer

Web Cache HTTP	WEBHOST1 WEBHOST2	7777	Web Cache HTTP Port
Web Cache HTTPS	WEBHOST1 WEBHOST2	4443	Web Cache HTTPS Port
Web Cache Invalidation	WEBHOST1 WEBHOST2	9401	Web Cache Invalidation Port
Web Cache Admin	WEBHOST1 WEBHOST2	9400	Web Cache Administration Port
HTTP Server (OHS) - HTTP	WEBHOST1 WEBHOST2	7778	OHS HTTP Listening Port
HTTP Server (OHS) – HTTPS	WEBHOST1 WEBHOST2	4444	OHS HTTPS Listening Port
HTTP Server Admin Port	WEBHOST1 WEBHOST2	8889	OHS Administration Port
OPMN Local Port	WEBHOST1 WEBHOST2 APPHOST1 APPHOST2	1880	OPMN Management Port
WebLogic Admin Port	APPHOST1	7001	WebLogic Administration Server Port
WLS_PORTAL	APPHOST1	7050	WebLogic Managed Server Port
WLS_PORTAL_2	APPHOST2	7050	WebLogic Managed Server Port
Internet Directory	SSOHOST	389/4443	OID HTTP/HTTPS port

Single Sign On	SSOHOST	7777	Single Sign on Listening Port.

WebLogic

The following have been assumed for the purposes of this paper, although it is recommended that these values be changed for your environment:

Purpose	Value	Comment
Web logic Domain Name	Portal	Name assigned to the WebLogic domain
WebLogic Admin User	WebLogic	WebLogic Administrator User Name

Installation Overview

Creating an enterprise deployment is a complicated process. This section summarizes the steps that need to be undertaken to create such a deployment:

1. If it does not already exist create an enterprise identity management deployment with Oracle Single Sign-on
2. Configure Network and load balancer
3. Create a Highly Available Database to store the portal metadata.
4. Create a portal metadata repository in the newly created database using the Repository Creation Utility.
5. Install WebLogic Server on APPHOST1.
6. Install and initial configuration of Oracle Portal on APPHOST1.
7. Configure Oracle HTTP Server on APPHOST1.
8. Configure Oracle Web Cache on APPHOST1.
9. Rewire portal to use the load balancer.
10. Configure Portal Parallel page Engine.

11. Create a Database Wallet.
12. Register Portal with Oracle Single Sign On.
13. Configure Host Assertion in Oracle WebLogic Server.
14. Install Oracle WebLogic Server on APPHOST2
15. Install and perform initial configuration of Oracle Portal on APPHOST2
16. Copy Files from APPHOST1 to APPHOST2
17. Introduce APPHOST2 to Web Cache.
18. Cluster Web Cache Instances on APPHOST1 an APPHOST2
19. Install Oracle Web Tier on WEBHOST1 and WEBHOST2
20. Introduce WEBHOST1 and WEBHOST1 to Web Cache Cluster.
21. Copy files from APPHOST1 to WEBHOST1 and WEBHOST2.
22. Tidy up installation.

Configuring the Network for Enterprise Deployments

This section describes some of the network prerequisites for the enterprise deployment.

Oracle Portal uses an external load balancer, which must support:

- Virtual server name and port configuration
- Process failure detection

Many Oracle Fusion Middleware components and services use ports. When configuring an enterprise deployment, it is important to know which port numbers are used by these services, and to ensure that the same port number is not used by two services. The Oracle installer will check to make sure that the ports you wish to use are not in use already.

Configure Virtual Server Names and Ports for the Load Balancer.

If you are using a load balancing router, it must be configured to enable the following:

- A virtual IP address (VIP1) that listens for requests to myPortal.mycompany.com on port 443 (an HTTPS listening port), and balances them to the application tier Oracle Web Caches running on WEBHOST1 and WEBHOST2 port 7777 (an HTTP listening port). You must configure the load balancing router to perform protocol conversion.
- The virtual IP address VIP1 listens for requests to myPortal.mycompany.com on port 7777 (an HTTP listening port), and balances them to the application tier Oracle Web Caches on WEBHOST1 and WEBHOST2 port 7777 (an HTTP listening port). Port 7777 on the load balancing router receives the HTTP loop-back requests made by the Parallel Page Engine. The 7777 port also receives requests from the Portal Metadata Repository for web provider design time messages. This configuration may require a Network Address Translation (NAT) rule in the load balancing router in order for the loop-back request from the PPE to succeed.

  Note: For security reasons, port 7777 on the load balancing router should not be visible to external users.

- The virtual IP address VIP1 listens for requests to myportal.mycompany.com on port 9401 (an HTTP listening port), and balances them to the application tier Oracle Web Caches on WEBHOST1 and WEBHOST2 port 9401 (an HTTP listening port). Port 9401 port on the load balancing router receives invalidation messages from the Oracle Portal Repository when content that is cached in Oracle Web Cache becomes stale. This configuration might require a Network Address Translation (NAT) rule in the load balancing router in order for the invalidation requests from the Oracle Portal repository to succeed.

- HTTP monitoring of OracleAS Web Cache. The load balancing router must be configured to detect an inoperative computer and stop routing requests to it until it is functioning again. Two OracleAS Web Cache ports must be monitored: the HTTP request port and the invalidation port.

To monitor port 7777, use the following URL in the load balancing router configuration:

hostname:port/_oracle_http_server_Web Cache_static_.html

For example:

http://webhost1.mycompany.com:7777/_oracle_http_server_Web Cache_static_.html

If the load balancing router receives a response from this URL, then the OracleAS Web Cache instance is running. If not, then the process or the server is down, and the load balancing router will forward all requests to the surviving computer.

To monitor port 9401, use the following URL in the load balancing router configuration:

http://hostname.domain.com:9401/x-oracle-cache-invalidate-ping

For example:

http://apphost1.mycompany.com:9401/x-oracle-cache-invalidate-ping

The load balancing router sends an HTTP request to this URL; the response header resembles the following:

HTTP/1.0

The load balancing router must be configured to detect the string HTTP in the first line of the response header. Thus, when the load balancing router detects HTTP in the first line of the response header, the invalidation port is available. If not, then all invalidation requests are routed to the surviving computer.

If a proxy server is being used, follow the instructions in Section "Configuring a Reverse Proxy for OracleAS Portal and OracleAS Single Sign-On".

Note:

The sqlnet.ora file must be updated to prevent connection time outs related to the load balancing router and firewall. See Section 4.1.5, "Configuring the Time out Value in the sqlnet.ora File".

Summary

To summarize, the load balancer requires the following configuration:

Configuring the Database for Enterprise Deployments

The myPortal.mycompany.com application requires a database to store its information in. This database should be a highly available Real Application Clusters database with the following characteristics:

Before beginning to install and configure the Portal components, the following steps must be performed:

- Install and configure the Oracle database repository.
- Create the Oracle Portal Management schemas in the database using the Repository Creation Utility (RCU).

Database versions supported

- Oracle Database 10g Release 2 (10.2.0.4)
- Oracle Database 11g Release 1 (11.1.0.7)

To determine the database version, execute this query:

SQL>select version from sys.product_component_version where product like 'Oracle%';

Real Application Clusters

The database used to store the metadata repository should be highly available in its own right, for maximum availability Oracle recommends the use of an Oracle Real Application Clusters (RAC) database.

Ideally the database will use Oracle ASM for the storage of data, however this is not necessary.

If using ASM, then ASM should be installed into its own Oracle Home and have two disk groups:

- 1 for the Database Files.
- 1 for the Flash Recovery Area.

If using Oracle ASM it is recommended that Oracle Managed Files also be used.

Installing and Configuring the Database Repository

Oracle Clusterware

- For 10g Release 2 (10.2), see the Oracle Database Oracle Clusterware and Oracle Real Application Clusters Installation Guide.
- For 11g Release 1 (11.1), see Oracle Clusterware Installation Guide.

Automatic Storage Management

- For 10g Release 2 (10.2), see Oracle Database Oracle Clusterware and Oracle Real Application Clusters Installation Guide.
- For 11g Release 1 (11.1), see Oracle Clusterware Installation Guide.

- When the installer is run, select the Configure Automatic Storage Management option in the Select Configuration page to create a separate Automatic Storage Management home.

Oracle Real Application Clusters

- For 10g Release 2 (10.2), see Oracle Database Oracle Clusterware and Oracle Real Application Clusters Installation Guide.
- For 11g Release 1 (11.1), see Oracle Real Application Clusters Installation Guide.

Configuring the Database for Oracle FMW 11g Metadata

Create a Real Applications Clusters Database with the following characteristics:

- Database should be in archive log mode to facilitate backup and recovery.
- Optionally Flashback should be enabled.
- Database is created with ALT32UTF8 character set.
- Database block size of 8K
- In addition the database will have the following minimum initialization parameters defined:

Parameter	Value
aq_tm_processes	1
dml_locks	200
job_queue_processes	10
open_cursors	400
session_max_open_files	50
sessions	400
processes	500

sga_target	512Mb
sga_max_size	800Mb
pga_aggregate_target	100Mb

Database Services

Oracle recommends using the Oracle Enterprise Manager Cluster Managed Services Page to create database services that client applications will use to connect to the database. For complete instructions on creating database services, see the chapter on Workload Management in the Oracle Database Oracle Clusterware and Oracle Real Application Clusters Administration and Deployment Guide.

SQL*Plus can be used to configure your RAC database to automate failover for Oracle Portal using the following instructions:

1. Use the CREATE_SERVICE subprogram to both create the database service and enable high-availability notification and configure server-side Transparent Application Failover (TAF) settings:

```
prompt> sqlplus "sys/password as sysdba"

SQL> EXECUTE DBMS_SERVICE.CREATE_SERVICE
(SERVICE_NAME => 'portal.mycompany.com',
NETWORK_NAME => 'portal.mycompany.com',
AQ_HA_NOTIFICATIONS => TRUE,
FAILOVER_METHOD => DBMS_SERVICE.FAILOVER_METHOD_BASIC,
FAILOVER_TYPE => DBMS_SERVICE.FAILOVER_TYPE_SELECT,
FAILOVER_RETRIES => 5, FAILOVER_DELAY => 5);
```

2. Add the service to the database and assign it to the instances using srvctl:

```
prompt> srvctl add service -d portal -s portal -r racnode1,racnode2
```

3. Start the service using srvctl:

```
prompt> srvctl start service -d portal -s  portal
```

Note:

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

If you already have a service created in the database, make sure that it is enabled for high-availability notifications and configured with the proper server-side Transparent Application Failover (TAF) settings. Use the DBMS_SERVICE package to modify the service to enable high availability notification to be sent through Advanced Queuing (AQ) by setting the AQ_HA_NOTIFICATIONS attribute to TRUE and configure server-side Transparent Application Failover (TAF) settings, as shown below:

```
prompt> sqlplus "sys/password as sysdba"

SQL> EXECUTE DBMS_SERVICE.MODIFY_SERVICE
(SERVICE_NAME => 'portal.mycompany.com',
AQ_HA_NOTIFICATIONS => TRUE,
FAILOVER_METHOD => DBMS_SERVICE.FAILOVER_METHOD_BASIC,
FAILOVER_TYPE => DBMS_SERVICE.FAILOVER_TYPE_SELECT,
FAILOVER_RETRIES => 5, FAILOVER_DELAY => 5);
```

For more information about the DBMS_SERVICE package, see the Oracle Database PL/SQL Packages and Types Reference.

Executing the Repository Creation Utility

The Repository Creation Utility (RCU) ships on its own CD as part of the Oracle Fusion Middleware 11g kit.

You run RCU to create the collection of schemas used by Identity Management and Management Services.

Issue this command:

```
prompt> RCU_HOME/bin/rcu &
```

Screen	Action
Welcome	Click **Next**.

Create Repository	Select **Create** Click **Next**.
Specify Installation Location	Specify the following values: Fusion Middleware Home Location (Installation Location) for example: `/u01/app/oracle/product/FMW/RCU`
Database Connection Details	Specify the following values: Database Type: Oracle Database Host Name: Enter one of the RAC nodes (use the VIP name) Port: Enter the listener port Service Name: Enter the service name of the RAC database. User Name: Enter sys Password: Enter the sys user password. Role: Select SYSDBA Click **Next**.
Check Pre-Requisites	Click OK when the pre-requisites have been validated.
Select Components	Specify the following values: Create New Prefix: Enter a prefix to be added to database schemas. For example MYP Components: Check AS Common Schemas -> Metadata Services Portal and BI -> Portal Webcenter Suite -> Webcenter portlets

	All other components should be unchecked.
	Click **Next**
Check Pre-Requisites	Click OK when the pre-requisites have been validated.
Schema Passwords	Enter passwords for each of the portal schemas or use the same password for all schemas.
	Click **Next**
Map Tablespaces	Click **Next** to accept the defaults
Create Tablespaces	Select **Yes** to allow the RCU to create any missing tablespaces.
Creating tablespaces	Select **OK** to acknowledge Table space creation.
Summary	Click **Create** to begin the creation process.

Configuring Single Sign On for Enterprise Deployments

Prior to starting this installation a highly available Oracle Single Sign On (Identity Management) needs to be in place and configured. Configuration of Oracle Identity Management is beyond the scope of this document.

Install and Configure application tier

Install application tier on APPHOST1

Install WebLogic Server

The first step in the installation procedure is to install WebLogic Server binaries

On UNIX issue the command: `server103_linux32.bin`

On Windows issue the command: `server103_win32.exe`

Screen	Action
Welcome	Click **Next**.
Choose Middleware Home Directory	Select **Create a New Middleware Home** Enter a value for the Middleware Home directory. This will be known henceforth MW_HOME. For example /u01/app/oracle/product/FMW Click **Next**.
Register for Security Updates	Choose whether or not to receive security updates from Oracle Support. If desired enter an email address and the appropriate Oracle Support Password. Click **Next**
Choose Install Type	Select **Typical** Click **Next**.
Choose Product Installation Directories	Click **Next**.
Installation Summary	Click **Next**.
Installation Complete	Uncheck runQuickstart and Click **Done.**

Install Oracle Portal Software

The next step in the installation procedure is to install Oracle Portal binaries into the MW_HOME created above

On UNIX issue the command: `runInstaller`

On Windows issue the command: `setup.exe`

Note: Before starting the install ensure that the following environment variables (UNIX) are not set:

- LD_ASSUME_KERNEL
- ORACLE_BASE
- LD_LIBRARY_PATH

Screen	Action
Welcome	Click **Next**.
Installation Type	Install Software and Configure Click **Next**.
Prerequisite Checks	Once all checks have passed. Click **Next**
Specify Installation Location	Enter the following Values:Middleware Home: Enter the value for MW_HOME For example /u01/app/oracle/product/FMW Oracle Home: Enter the installation directory for Portal. ** Note this will be placed under the MW_HOME directory. For example `Portal` WebLogic Server Directory: Enter the installation directory for Oracle WebLogic server. This should be MW_HOME/wlserver_10.3 For example `/u01/app/oracle/product/FMW/wlserver_10.3` Oracle Instance Location: Enter the directory where the Oracle Configuration files will be placed. This should be outside of the Oracle

	Home. This will be known henceforth as ORACLE_INSTANCE For example `/u01/app/oracle/admin/PortalDomain/Portal1` Oracle Instance Name: Portal1 Click **Next**
Select Domain	Select Create New Domain and enter the values: User Name: Name of user to log into the WebLogic domain. User Password: Password for the domain. Confirm Password: The same as above Domain Name: Name for the Domain: PortalDomain Click **Next**
Configure Components	As a minimum ensure that the following values are checked: Server Components – Oracle Portal Management Components – Enterprise Manager Ensure that the clustered box is ticked. Click **Next**.

Configure Ports	Select Specify Ports using Configuration File
	In HA implementations whilst not mandatory it makes life simpler if all of the ports used by the various components are synchronized across hosts. Oracle allows the bypassing of Automatic port Configuration by specifying ports to be used in a file.
	Select a File Name and then click **View/Edit**. The file will look like this:
	`[DOMAIN]`
	`#This port indicates the Domain port no`

```
Domain Port No = 7001

[OHS]

#Listen port for OHS component

Oracle HTTP Server Port No = 7780


[WEB CACHE]

#Port no for WebCache component (also used for virtual server
port)

Oracle Web Cache Port No = 7777

#Adminstration port no for WebCache component

Oracle Web Cache Administration Port No = 9400

#STATISTICS port no for WebCache component

Oracle Web Cache Statistics Port No = 9402

#INVALIDATION port no for WebCache component

Oracle Web Cache Invalidation Port No = 9401

[OPMN]

#Process Manager Local port no

Oracle Process Manager Local Port No = 1880

[MANAGEDSERVER]

#Port no for Portal Managed Server

Oracle WLS Portal Managed Server Port No = 7050
```

You can find a sample staticports.ini file on installation Disk1 in the stage/Response directory.

Save the file and click **Next**

Specify Schema	Specify the following values: Database Connect String in the format: racnode1-vip:ListenerPort:racnode2-vip:ListenerPort@mydb.mycompany.com Portal Schema Name: MYP_PORTAL Portal Schema Password: Enter password entered in RCU Click **Next**
Specify Portlet Schema	Specify the following values: Portlet Schema Name: MYP_PORTLET Portlet Schema Password: Enter password entered in RCU Click **Next**
Specify Application Identity Store	Specify the following values: Hostname: Name of oid server: login.mycompany.com Port: OID port:389 User Name: cn=orcladmin Password: OID's orcladmin password. Click **Next**
Summary	Click **Install** to begin the creation process. When prompted the script oracleRoot.sh needs to be run as the root user – UNIX installations only.

Validation

Validate the initial Portal installation by performing the following tests.

Test	URL	Result
Test Portal	http://APPHOST1.mycompany.com:7777/portal/pls/portal/	Portal Home Page Displayed
Test Portal Login	http://myPortal.mycompany.com/portal/pls/portal	Log into Portal using the user account orcladmin
Test WebLogic Admin Console	http://APPHOST1.mycompany.com:7001/console	Login using admin credentials specified above
Test EM	http://APPHOST1.mycompany.com:7001/em	Login using admin credentials specified above
Test Webcache Admin	http://APPHOST1.mycompany.com:9400/Web Cacheadmin	Login using admin credentials administrator/administrator

Configure APPHOST1

Create boot.properties file

Create a boot.properties file for the Administration Server on APPHOST1. The boot.properties file enables the Administration Server to start without prompting you for the administrator username and password.

In a text editor, create a file called boot.properties in the directory DOM_HOME/servers/AdminServer/security, and enter the following lines in the file:

username=<adminuser>

password=<password>

Restarting the Administration Server will encrypt the values in the above file, for that reason it is recommended that the Administration Server be restarted on each node, which can host it.

The Administration Server is stopped using the script stopWebLogic.sh which is located in DOM_HOME/bin and started using the script startWebLogic also located in DOM_HOME/bin

Set Admin Server Listen Address

To do this, login to the WebLogic console using the URL:

http://apphost1.mycompany.com:7001/console

Select Environment – Servers from the Domain Structure Menu

Click on AdminServer(admin)

Click on Lock and Edit from the Change Center.

Set the listen address to the DNS name referring to the network card you wish to use. This is generally the public server name.

Click **Save**

Click **Activate Changes** from the Change Center.

Restart the Administration server to enable the changes.

The Administration Server is stopped using the script stopWebLogic.sh which is located in DOM_HOME/bin and started using the script startWebLogic also located in DOM_HOME/bin

Configure sqlnet.ora

Create a file called sqlnet.ora in the directory ORACLE_INSTANCE/config/ and add the following entry to the file:

```
TCP.CONNECT_TIMEOUT=10
```

This ensures that database connections time out after a reasonable time.

Configure Virtual Hosts

In order for Portal to work with the load balancer two virtual hosts need to be created.

Create a file called **virtual_hosts.conf** in

ORACLE_INSTANCE/config/OHS/ohs1/moduleconf

Add the following entries to the file:

```
NameVirtualHost *:7778
<VirtualHost *:7778>
    ServerName https://myPortal.mycompany.com:443
    RewriteEngine On
    RewriteOptions inherit
    UseCanonicalName On
</VirtualHost>

<VirtualHost *:7778>
    ServerName apphost1.mycompany.com:7777
    RewriteEngine On
    RewriteOptions inherit
    UseCanonicalName On
</VirtualHost>
```

Configure Web Cache

Log into the Enterprise Manager Administration Console

Log into the Enterprise Manager Console using the URL:

http://apphost1.mycompany.com:7001/em

Default User Name and Password are the same as the domain username and password entered during the installation.

Create Site

In the Navigator window, expand the Web Tier tree.

Click on the component wc1

From the drop down list at the top of the page select **Administration - Sites**

Select **Create Site**

Enter the following information to add the following site:

Site: myPortal.mycompany.com	
Host Name	myPortal.mycompany.com
Port	443
Default site	Yes
Site Wide Compression	Yes
Site Alias – Host Name	myPortal.mycompany.com
Site Alias - Port	7777
Site Alias – Host Name	myPortal.mycompany.com
Site Alias - Port	80

Leave everything else at the default. and then click **Submit**.

Select **OK** to save each entry

Remove all other site entries from the list.

Create Site to Server Mapping

On the same page select Create in the Site-to-server Mapping section.

Enter the following information to add the site

Host Pattern	myPortal.mycompany.com
Port Pattern	443
Selected Origin Servers	Apphost1.mycompany.com:7778

Click **OK** to store the site.

Remove all other site entries from the list by clicking on each entry and then clicking the Delete button.

Ensure that the site APPHOST1.mycompany.com:443 appears first in the list of site to server mappings.

Click **Apply** to save the changes.

Enable Session Binding

The session binding feature in OracleAS Web Cache is used to bind user sessions to a given origin server to maintain state for a period of time. Although almost all components running in a default OracleAS portal mid-tier are stateless, session binding is required for two reasons:

- The Web Clipping Studio, used by both the OracleAS Web Clipping Portlet and the Web Page Data Source on OmniPortlet uses HTTP Sessions to maintain state, for which session binding must be enabled.
- Enabling session binding forces all the user requests to go to a specific OracleAS Portal middle tier, resulting in a better cache hit ratio for the portal cache.

Follow these steps to enable session binding:

From the drop down list at the top of the page select Administration – Session Configuration

Select the site myPortal.mycompany.com:443 from the drop down list.

In the Session Binding session select Cookie based Session Binding with any Set Cookie

Select **Apply** to save the changes.

Change Web Cache Passwords

The Web Cache invalidation and admin passwords are randomly generated, however they are required later. It is therefore recommended that these passwords be changed from the default value to a new known value.

This is achieved by:

In the Navigator window, expand the Web Tier tree.

Click on the component **wc1**

From the drop down list at the top of the page select **Administration – Passwords**

Enter a new invalidation password and administration passwords, confirm and click **Apply**

Restart Web Tier (OHS and Web Cache)

Having made the above changes the Web Tier components need to be restarted. This can be achieved by issuing the commands:

Restart the Oracle HTTP Server using the commands:

```
opmnctl stopall
opmnctl startall
```

Validate Configuration

In order to validate the configuration the following tests should be performed:

Test	URL	Result
Test load balancer	http://myPortal.mycompany.com/	Home page displayed
Test load balancer via SSL	https://myPortal.mycompany.com/	Home page displayed
Test load balancer Termination (**)	https://myPortal.mycompany.com/portal/pls/portal/owa_util.print_cgi_env	REQUEST_PROTOCOL value of HTTPS

(**) Note: owa_util.print_cgi_env needs to be enabled by:

Adding: PlsqlExclusionList "#None#" in the portal_dads.conf. file located in DOM_HOME/config/fmwconfig/servers/WLS_PORTAL/applications/portal/configuration/portal_dads.conf

From the database servers check that it is possible to contact the Webcache page invalidator. From each database host issue the command:

telnet myPortal.mycompany.com 9401

Ensure that no connection error messages are returned.

Rewire Portal Repository

Log into the domain via Enterprise Manager using the URL:

http://apphost1.us.oracle.com:7001/em

Expand the Fusion Middleware Menu on the left hand side.

Expand the Portal menu (under Fusion Middleware Menu)

Click on Portal and then right click on Portal again.

Select settings Wire Configuration

Enter the following information:

Portal Midtier	
host	Enter the DNS name of the load balancer For example myportal.mycompany.com
Port	Enter the SSL port that the load balancer is listening on. for example 443
SSL Protocol	Ensure that this is ticked. This will ensure that when portal needs to generate URLs it generates them using the format: https://myportal.mycompany.com:443/

WebCache	
host	Enter the DNS name of the load balancer for example myportal.mycompany.com
Invalidation Port	Enter the Portal Invalidation port as configured at the load balancer e.g. 9401
Invalidation User Name	invalidator
Invalidation Password	Password for the above account.

Click **Apply** to start the rewire.

After the rewire is complete click on the Portal Menu option again, and ensure that the Portal URL now shows:

https://myportal.mycompany.com:443/portal/pls/portal

Configure Parallel Page Engine Look-Back with load balancer

The purpose of the Parallel Page Engine (PPE Servlet) is to construct pages that have been requested by users. It does this by receiving the page request from a user, making its own new requests to fetch all the pieces of the page "in parallel", assembling these pieces into a single page file and then sending the page content back to the end user (or back to the client browser).

These internal requests should be kept inside of the organization, and be served using the HTTP protocol. To enable this:

Log into the Enterprise Manager as described above. Select Fusion Middleware -> Classic -> Portal from the object browser on the left.

Right click on Portal, and select Settings -> Page Engine

In the Advanced Properties section add the following information:

UsePort	Select the internal loopback port number for example: 7777
Use Scheme	http
HTTPS Ports	443

Click **Apply** to save the settings.

Restart the WebLogic Managed Server from the WebLogic admin console:

Connect to the console using the URL: **http://APPHOST1.mycompany.com:7001/console**

Select Servers, and then select the Control tab. Select the box next to WLS_PORTAL, select Shutdown then Startup.

Create a Database Wallet

Portal requires a wallet in the database in which the portal schema resides. The certificate of the load balancer is stored in this wallet.

Before starting this process it is necessary to copy the certificate to the database servers.

Each browser does this in a slightly different way below are the instructions for the Internet Explorer 7 and Firefox browsers:

Use the browser to access the URL https://myportal.mycompany.com.

Follow the browsers prompts to save or import the certificate.

Firefox

Go to **Firefox -> Preferences – Advanced – Encryption – view certificates**

Highlight the certificate for myportal.mycompany.com select export and give the file a name.

Internet Explorer 7

Go to Internet options -> Content – certificates

Find the certificate in the various certificate stores (the location will depend on where you requested it to be stored when you imported the certificate when you accessed the site).

Highlight the certificate

Click on Export

The Export wizard will be started.

Click **Next**

Select DER encoded binary X.509 (.CER), which is the default.

Click **Next**

Specify a file name

Click **Next**

Click **Finish**.

Import Certificate into database wallet.

Copy this file to the database server.

Save the certificate if requested to do so.

Having obtained a copy of the certificate the next step is to create a wallet on each of the database servers and import this certificate. This is achieved using the Oracle Wallet Manager from the database server. Note this has to be performed on all of the RAC nodes:

type owm to invoke the Oracle Wallet Manager

Select Wallet -> New

Select No to NOT create the wallet in the default location.

Enter a password for the wallet (keep a note of this as it will be required later).

Set the wallet type to standard.

Select No to the question "Do you want to create a certificate at this time?"

In Oracle Wallet Manager select Operations – Import Trusted certificate.

Select "Select a file that contains the certificate" and Click **OK**

Select the certificate file selected above and click import.

Select Wallet and Save As

Select a location for the wallet for example $ORACLE_BASE/admin/DB_NAME/wallet

Repeat for successive nodes.

Identify the Wallet to Portal

Now that the certificate is stored inside the database wallet, the location of the wallet has to be stored within the Portal repository. This is achieved by running the sqlplus script secwc.sql which is located in the directory:

ORACLE_HOME/portal/admin/plsql/wwc

For example

sqlplus myp_portal/mypasswd@mydb.mycompany.com

Note:

It may be necessary to create a database entry in the file tnsnames.ora located in ORACLE_HOME/network/admin

SQL> @secwc 'file:$ORACLE_BASE/admin/DB_NAME/wallet' 'walletpassword'

Notes:

Use the absolute path to the wallet - do not use environment variables

walletpassword is the password for the wallet.

Use the path to the wallet directory not the wallet file itself.

Register with SSO

These steps must be carried out from the Single Sign-On (SSO) server:

4. set the ORACLE_HOME variable to the SSO *ORACLE_HOME* location

5. Execute ORACLE_HOME/sso/bin/ssoreg.sh (ssoreg.bat on Windows) with the following parameters

```
-site_name myPortal.mycompany.com
-mod_osso_URL https://myPortal.mycompany.com
-config_mod_osso TRUE
-oracle_home_path ORACLE_HOME
-config_file /tmp/osso.conf
-admin_info cn=orcladmin
-virtualhost
-remote_midtier
```

6. Copy /tmp/osso.conf to the Portal mid-tier home location $ORACLE_INSTANCE/config/OHS/ohs1
7. Restart Oracle HTTP Server by issuing the command ORACLE_HOME/opm/bin/opmnctl restartproc process-type=OHS
8. Log into the Single Sign-On Server via the URL http://login.mycompany.com/pls/orasso
9. Go to the administration page and then Administer Partner applications. Delete the entry for apphost1.mycompany.com

Restart Web Tier (OHS and Web Cache)

Having made the above changes the Web Tier components need to be restarted. This can be achieved by issuing the commands:

Restart the Oracle Web Tier components using these commands:

```
opmnctl stopall
opmnctl startall
```

Note: Prior to issuing these commands ensure that then environment variable ORACLE_INSTANCE is set to the value that was entered during the install above.

Change Host Assertion in WebLogic

Because the Oracle HTTP Server acts as a proxy for WebLogic, by default certain CGI environment variables are not passed through to WebLogic. These include the host and port. WebLogic needs to be told that it is using a virtual site name and port so that it can generate internal URLs appropriately.

Log into the WebLogic administration console using the following URL

http://apphost1.mycompany.com:7001/console

Select Clusters from the home page or alternatively **Environment -> Clusters from the Domain** structure menu.

Click Lock and Edit in the Change Center Window to enable editing.

Click on the Cluster Name (cluster_portal)

Select HTTP and enter the following values:

Parameter	**Value**
Frontend Host	myportal.mycompany.com
Frontend HTTP Port	80
Frontend HTTPS Port	443

This ensures that any HTTPS URLs created from within WebLogic are directed to port 443 on the load balancer.

Click **Activate Changes** in the Change Center window to enable editing.

Restart the WLS_PORTAL Managed Server by:

Select Servers from the home page or alternatively Environment -> Servers from the Domain structure menu.

Select the Control tab

Select the box next to WLS_PORTAL

Select Shutdown -> Force Shutdown Now

Click **Yes** to shutdown the managed server.

Once the server is shutdown

Select the box next to WLS_PORTAL

Click on **Start**

Click **Yes** to start the managed server

Validate Configuration

In order to validate the configuration the following tests should be performed:

Test	URL	Result
Test load balancer SSL Termination	https://myPortal.mycompany.com/portal/pls/portal/owa_util.print_cgi_env	REQUEST_PROTOCOL value of HTTPS
Test Portal via load balancer	https://myPortal.mycompany.com/portal/pls/portal	Portal Home Page Displayed
Test Portal Login vi load balancer	https://myPortal.mycompany.com/portal/pls/portal	Should be able to login using account orcladmin

Troubleshooting

WWC-0000

Sometimes after performing the above steps a WWC-0000 message is displayed along with error text which starts something like:

@ ;i=pls%2Forasso%2Forasso.wwsso_app_admin.fapp_process_login%3Fp_app_id%3D;

Accept=text/html Accept-Charset=ISO-8859-1,utf-8;q=0.7,*;q=0.7…..

This error text indicates that the load balancers certificate is not correctly stored in the database wallet and identified correctly to Portal.

Redo sections:

Create a Database Wallet

Identify the Wallet to Portal

Install Application Tier on APPHOST2

Install WebLogic Server

The first step in the installation procedure is to install WebLogic Server binaries onto APPHOST2

On UNIX issue the command: `server103_linux32.bin`

On Windows issue the command: `server103_win32.exe`

Screen	Action
Welcome	Click **Next**.
Choose Middleware Home Directory	Select **Create a New Middleware Home** Enter a value for the Middleware Home directory. This will be known henceforth as MW_HOME. For example /u01/app/oracle/product/FMW Click **Next**.
Register for	Choose whether or not to receive security updates from Oracle Support. If

Security Updates	desired enter an email address and the appropriate Oracle Support Password. Click **Next**
Choose Install Type	Select **Typical** Click **Next**.
Choose Product Installation Directories	Click **Next**.
Installation Summary	Click **Next**.
Installation Complete	Uncheck runQuickstart and Click **Done.**

Install Oracle Portal Software

The next step in the installation procedure is to install Oracle Portal binaries into the MW_HOME created above

On UNIX issue the command: `runInstaller`

On Windows issue the command: `setup.exe`

Note: Before starting the install ensure that the following environment variables (UNIX) are not set:

- LD_ASSUME_KERNEL
- ORACLE_BASE
- LD_LIBRARY_PATH

Screen	Action
Welcome	Click **Next**.

Installation Type	Install Software and Configure Click **Next**.
Prerequisite Checks	Once all checks have passed Click **Next**
Specify Installation Location	Enter the following Values: Middleware Home: Enter the value for MW_HOME for example: /u01/app/oracle/product/FMW Oracle Home: Enter the installation directory for Portal. ** Note this will be placed under the MW_HOME directory. For example `Portal` WebLogic Server Directory: Enter the installation directory for Oracle WebLogic server. This should be MW_HOME/wlserver_10.3 For example `/u01/app/oracle/product/FMW/wlserver_10.3` Oracle Instance Location: Enter the directory where the Oracle Configuration files will be placed. This should be outside of the Oracle Home. This will be known henceforth as ORACLE_INSTANCE For example `/u01/app/oracle/admin/PortalDomain/portal2` Oracle Instance Name: Portal2 Click **Next**
Select Domain	Select Expand Cluster and enter the values: Host Name: Name of host running WebLogic Admin Server: APPHOST1.mycompany.com

	Port: Port Admin Server is using for example: 7001 User Name: Admin Server administrator account name. Password: Admin Server Password Click **Next**
Configure Components	At a minimum ensure that the following values are checked (Note this should be the same list as that selected for APPHOST1: Server Components – Oracle Portal Click **Next**.
Configure Ports	Select Specify Ports using Configuration File Select the same file used for APPHOST1 and click **Next**
Specify Application Identity Store	Specify the following values: Hostname: Name of oid server:login.mycompany.com Port: OID port: 389 User Name: cn=orcladmin Password: OID's orcladmin password. Click **Next**
Summary	Click **Install** to begin the creation process. When prompted the script oracleRoot.sh needs to be run as the root user – UNIX installations only.

Configure the WebLogic Domain for APPHOST2

The following steps are performed on the Administration Server on APPHOST1. These steps will create a second managed server and tell WebLogic that it will be running on APPHOST2

Configure application tier on APPHOST2

Introduce WLS_PORTAL1 to Oracle HTTP Server on APPHOST1

Now that the managed server WLS_PORTAL1is up and running, the Oracle HTTP Server (OHS) on APPHOST1 needs to be told of its existence, so that it can route requests to it.

Update Oracle HTTP Server configuration to be cluster aware.

When the installation was first created it was configured all WebLogic requests are directed to the managed server WLS_PORTAL residing on APPHOST1. Now that a WebLogic cluster has been created, these requests need to be directed to the cluster.

On APPHOST1, edit the file
ORACLE_INSTANCE/config/OHS/ohs1/moduleconf/portal.conf

Edit the above file and change the following entries for the blocks beginning with:

- /portal
- /portalTools
- /wsrp-tools
- /portalHelp
- /portalHelp2

Remove the lines beginning WebLogicHost and WebLogic port and add in a line which looks like:

WebLogicCluster apphost1.mycompany.com:9001,apphost2.mycompany.com:9001

For example

Change

```
<Location /portal>
    SetHandler WebLogic-handler
    WebLogicHost apphost1.mycompany.com
    WebLogicPort 9001
</Location>
```

to:

```
<Location /portal>
    SetHandler WebLogic-handler
    WebLogicCluster apphost1.mycompany.com:9001,apphost2.mycompany.com:9001
</Location>
```

Restart the Oracle HTTP Server using the command:

opmnctl restartproc process-type=OHS

Copy Configuration Information from APPHOST1

Even though the expand cluster has created a new WebLogic managed server and associated machine it is still necessary to copy some configuration information from APPHOST1 to APPHOST2.

Copy the following files located on APPHOST1

File	location APPHOST1	Location APPHOST2
appConfig.xml	MW_HOME/user_projects/domains/PortalDomain/config/fmwconfig/servers/WLS_PORTAL/applications/portal/configuration/	MW_HOME/user_projects/domains/PortalDomain/config/fmwconfig/servers/WLS_PORTAL1/applications/portal/configuration
portal_cache.conf		
portal_dads.conf		
portal_plsql.conf		
mod_oradav.conf	ORACLE_INSTANCE/config/OHS/ohs1/moduleconf	ORACLE_INSTANCE/config/OHS/ohs1/moduleconf
mod_osso.conf		
plsql.conf		
portal.conf		
virtual_hosts.conf		

osso.conf	ORACLE_INSTANCE/config/OHS/ohs1	ORACLE_INSTANCE/config/OHS/ohs1
sqlnet.ora	ORACLE_INSTANCE/config/	ORACLE_INSTANCE/config/

Configure Virtual Hosts

Edit the file ORACLE_INSTANCE/config/OHS/ohs1/moduleconf/virtual_hosts.conf on APPHOST2

Remove the virtual Host entry for APPHOST1 and add one for APPHOST2 so that the file looks like:

```
NameVirtualHost *:7778
<VirtualHost *:7778>
    ServerName https://myPortal.mycompany.com:443
    RewriteEngine On
    RewriteOptions inherit
    UseCanonicalName On
</VirtualHost>

<VirtualHost *:7778>
    ServerName apphost2.mycompany.com:7777
    RewriteEngine On
    RewriteOptions inherit
    UseCanonicalName On
</VirtualHost>
```

Restart the Oracle HTTP Server using the commands:

```
ORACLE_HOME/opmn/bin/opmnctl stopall
ORACLE_HOME/opmn/bin/opmnctl startall
```

Create Portal Directories

Create the following directories on APPHOST2 to allow the storage of the Oracle Portal Cache:

ORACLE_INSTANCE/portal/cache

ORACLE_INSTANCE/diagnostics/logs/portal

Update Instance Paths

Two of the copied files have hard coded entries for the above directories; these files need amending to reflect the paths above.

Edit the files:

portal_cache.conf – Change PlsqlCacheDirectory

portal_plsql.conf – Change PlsqlLogDirectory

The files are located in the directory:

$DOM_HOME/config/fmwconfig/servers/WLS_PORTAL1/applications/portal/configuration

Start WLS_PORTAL1

Now that the application files have been copied across it should be possible to start the managed server WLS_PORTAL1.

Log into the Administration Server on APPHOST1 using the URL:

http://APPHOST1.mycompany.com:7001/console

Provide the WebLogic administration console login credentials.

Select Environment -> Servers from the Domain structure menu.

Select the Control tab

Select the box next to the managed server WLS_PORTAL1 and click Shutdown – Force Shutdown Now.

Click on **Yes** to confirm the operation. This will reset the server's status.

Wait for the operation to complete.

Select the box next to the managed server WLS_PORTAL1 and click **Start**

Click on **Yes** to confirm the operation.

Wait for the operation to complete.

Configure Web Cache

Log into the Enterprise Manager Administration Console

Log into the Enterprise Manager Console using the URL:

http://apphost1.mycompany.com:7001/em

Default User Name and Password are the same as the domain username and password entered during the installation.

Change Web Cache Passwords

The Web Cache invalidation and admin passwords are randomly generated, however they are required later. It is therefore recommended that these passwords be changed from the default value to a new known value.

This is achieved by:

In the Navigator window, expand the Web Tier tree.

Click on the component wc1

From the drop down list at the top of the page select Administration – Passwords

Enter a new invalidation password and administration passwords, confirm and click **Apply**

NOTE: Use the same passwords as used in APPHOST1.

Restart Web Cache, for the changes to take effect using the commands:

```
opmnctl restartproc ias-component=wc1
```

Create Origin Server

In the Navigator window, expand the Web Tier tree.

Click on the component wc1 (make sure it is the one associated with APPHOST1)

From the drop down list at the top of the page select Administration – Origin Servers

Select **Create**

Enter the following information to add the origin server

Host	APPHOST2.mycompany.com

Port	7778
Capacity	100
Protocol	HTTP
Failover Threshold	5
Ping URL	/
Ping Frequency	10

And select **OK** to save the changes.

Select **Apply** to save the changes.

Add Origin Server Site to Server Mapping

In the Navigator window, expand the Web Tier tree.

Click on the component wc1 (make sure it is the one associated with APPHOST1)

From the drop down list at the top of the page select Administration – Sites

In the Site to Server Mapping section click on the Host:port

myPortal.mycompany.com:443

Click on **Edit**

Select the origin server APPHOST2.mycompany.com:7778 and move it to the selected Origin servers list.

Click **OK** to save the changes.

Select **Apply** to save the changes.

Cluster Web Cache on Hosts APPHOST1 and APPHOST2

In the Navigator window, expand the Web Tier tree.

Click on the component wc1 (make sure it is the one associated with APPHOST1)

From the drop down list at the top of the page select Administration – Cluster

Click on **Add**

The Web Cache from APPHOST2 will automatically be added.

Select **Apply** to apply the changes

Click on the newly created Web Cache entry (be sure not to click on the URL part of it)

Click on **Synchronize** to copy the configuration to the Web Cache on APPHOST2.

Click **Yes** when prompted to confirm that you wish you perform the operation.

Click **Apply** to apply the new configuration

Restart the Web Caches on both APPHOST1 and APPHOST2 by issuing the following command on each server:

```
opmnctl restartproc ias-component=wc1
```

Validate Configuration

In order to validate the configuration the following tests should be performed:

Before starting the tests, Shutdown all the processes on APPHOST1 including:

- Managed Server WLS_PORTAL
- Web Cache
- Oracle HTTP Server.

Once the validation tests have been performed start the above processes on APPHOST1 and retry.

Test	URL	Result
Test load balancer	http://myPortal.mycompany.com/	Home page displayed
Test load balancer via SSL	https://myPortal.mycompany.com/	Home page displayed

Test load balancer Termination	https://myPortal.mycompany.com/portal/pls/portal/owa_util.print_cgi_env	REQUEST_PROTOCOL value of HTTPS

Setting up Node Manager

This section describes how to configure Node Manager per the EDG recommendations. Oracle Fusion Middleware EDG recommends using host name verification for the communications between Node Manager and the Administration Server. This requires the use of certificates for the different addresses communicating with the Administration Server. In this section, the steps for configuring APPHOST1 and APPHOST2 certificates for host name verification are provided.

This section includes the following subsections:

About the Node Manager

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

About Passwords

The passwords used in this guide are used only as examples. Use secure passwords in a production environment. For example, use passwords that include both uppercase and lowercase characters as well as numbers.

Enabling Host Name Verification for Node Manager - APPHOST1

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

Step 1: Generating Self-Signed Certificates Using the utils.CertGen Utility

Step 2: Creating an Identity Keystore Using the utils.ImportPrivateKey Utility

Step 3: Creating a Trust Keystore Using the keytool Utility

Step 4: Configuring Node Manager to Use the Custom Keystores

Generating Self-Signed Certificates Using the utils.CertGen Utility

Follow these steps to create self-signed certificates on APPHOST1.mycompany.com. These certificates should be created using the network name/alias. For information on using trust CA certificates instead, see "Configuring Identity and Trust" in Oracle Fusion Middleware Securing Oracle WebLogic Server.

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

   In the Bourne shell, run the following command:

   ```
   APPHOST1> . setWLSEnv.sh
   ```

Verify that the CLASSPATH environment variable is set:

```
APPHOST1> echo $CLASSPATH
```

2. Create a user-defined directory for the certificates. For example, create a directory called certs under the ORACLE_BASE/product/FMW/ directory. Note that certificates can be shared across WLS domains.

   ```
   APPHOST1> cd ORACLE_BASE/product/FMW
   APPHOST1> mkdir certs
   ```

3. Change directory to the user-defined directory.

   ```
   APPHOST1> cd certs
   ```

4. Run the utils.CertGen tool from the user-defined directory to create the certificates for APPHOST1.

Syntax:

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

Examples:

```
APPHOST1> java utils.CertGen welcome1 APPHOST1_cert APPHOST1_key
            domestic APPHOST1.mycompany.com
```

Creating an Identity Keystore Using the utils.ImportPrivateKey Utility

Follow these steps to create an Identity Keystore on APPHOST1.mycompany.com.

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

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

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

   Import the certificate and private key for APPHOST1 into the Identity Store.

   Make sure that you use a different alias for each of the certificate/key pair imported.

   Syntax:

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

   Examples:

   ```
   APPHOST1> java utils.ImportPrivateKey appIdentityKeyStore.jks welcome1
               appIdentity1 welcome1
               ORACLE_BASE/product/FMW/certs/APPHOST1_cert.pem
               ORACLE_BASE/product/FMW/certs/APPHOST1_key.pem
   ```

Creating a Trust Keystore Using the keytool Utility

Follow these steps to create the Trust Keystore on APPHOST1.mycompany.com.

1. Create a new trust keystore called appTrustKeyStore using the keytool utility:

```
APPHOST1> keytool -keystore appTrustKeyStore.jks -genkey -keyalg RSA -
alias appTrustKey -dname "cn=appTrustKey,ou=FOR TESTING
ONLY,o=MyOrganization,L=MyTown,ST=MyState,C=US"

Enter keystore password:
Re-enter new password:
Enter key password for <appTrustKey>
RETURN if same as keystore password):
```

   Note:

   Use the standard Java keystore to create the new trust keystore because it already contains most of the needed root CA certificates. Do not to modify the standard Java trust key store directly.

2. You will be asked a series of questions. The keystore is created after you respond to these questions.

   Tip: Make a note of the information that you provide on the command line and in the subsequent dialog box, because you will need this information to define gateway policy steps.

3. Change the default password for the standard Java keystore utility using the keytool utility. Use the following syntax to change the default password:

```
keytool -storepasswd -keystore <TrustKeyStore>
```

4. Copy the standard Java keystore called cacerts, which is located in the ORACLE_BASE/product/FMW/wlserver_10.3/server/lib directory, to the same directory as the certificates. Copy cacerts as follows:

```
APPHOST1> cp ORACLE_BASE/product/FMW/wlserver_10.3/server/lib/cacerts
          ORACLE_BASE/product/FMW/certs/appTrustKeyStore.jks
```

5. Import the CA certificate called CertGenCA.der into the appTrustKeyStore using the keytool utility. This certificate, which is located in the ORACLE_BASE/product/FMW/wlserver_10.3/server/lib directory, is used to sign all

certificates generated by utils.CertGen tool. Import CertGenCA.der using the following syntax:

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

Configuring Node Manager to Use the Custom Keystores

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

```
KeyStores=CustomIdentityAndCustomTrust
CustomIdentityKeyStoreFileName=<Identity KeyStore>
CustomIdentityKeyStorePassPhrase=<Identity KeyStore Passwd>
CustomIdentityAlias=<Identity Key Store Alias>
CustomIdentityPrivateKeyPassPhrase=<Private Key used when creating Certificate>
```

Make sure to use the correct value for CustomIdentityAlias on each node. For example on APPHOST1, use appIdentity1.

Example for Node 1:

```
KeyStores=CustomIdentityAndCustomTrust
CustomIdentityKeyStoreFileName=ORACLE_BASE/product/FMW/certs/
appIdentityKeyStore.jks
CustomIdentityKeyStorePassPhrase=welcome1
CustomIdentityAlias=appIdentity1
CustomIdentityPrivateKeyPassPhrase=welcome1
```

The passphrase entries in the nodemanager.properties file get encrypted when you start Node Manager as described in the next section, "Starting the Node Manager on APPHOST1" For security reasons, you want to minimize the time the entries in the nodemanager.properties file are left unencrypted. After you edit the file, you should start Node Manager as soon as possible so that the entries get encrypted.

Starting the Node Manager on APPHOST1

Run these commands to start Node Manager on APPHOST1:

```
APPHOST1> cd ORACLE_BASE/product/FMW/wlserver_10.3/server/bin
```

```
APPHOST1> ./startNodeManager.sh
```

Enabling Host Name Verification for Node Manager - APPHOST2

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

Step 1: Generating Self-Signed Certificates Using the utils.CertGen Utility

Step 2: Creating an Identity Keystore Using the "utils.ImportPrivateKey" Utility

Step 3: Creating a Trust Keystore Using the keytool Utility

Step 4: Configuring Node Manager to Use the Custom Keystores

Generating Self-Signed Certificates Using the utils.CertGen Utility

Follow these steps to create self-signed certificates on APPHOST2.mycompany.com. These certificates should be created using the network name/alias.

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

   In the Bourne shell, run the following command:

   ```
   APPHOST2> . setWLSEnv.sh
   ```

   Verify that the CLASSPATH environment variable is set:

   ```
   APPHOST2> echo $CLASSPATH
   ```

2. Create a user-defined directory for the certificates. For example, create a directory called certs under the ORACLE_BASE/product/FMW/ directory. Note that certificates can be shared across WLS domains.

   ```
   APPHOST2> cd ORACLE_BASE/product/FMW
   APPHOST2> mkdir certs
   ```

3. Change directory to the user-defined directory.

```
APPHOST2> cd certs
```

4. Run the utils.CertGen tool from the user-defined directory to create the certificates for APPHOST2.

   Syntax:

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

   Examples:

   ```
   APPHOST2> java utils.CertGen welcome1 APPHOST2_cert APPHOST2_key domestic
   APPHOST2.mycompany.com
   ```

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

Follow these steps to create an Identity Keystore on APPHOST2.mycompany.com.

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

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

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

Import the certificate and private key for APPHOST2 into the Identity Store. Make sure that you use a different alias for each of the certificate/key pair imported.

Syntax:

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

Example:

```
APPHOST2> java utils.ImportPrivateKey appIdentityKeyStore.jks welcome1
appIdentity1 welcome1 ORACLE_BASE/product/FMW/certs/APPHOST2_cert.pem
ORACLE_BASE/product/FMW/certs/APPHOST2_key.pem
```

Creating a Trust Keystore Using the keytool Utility

Follow these steps to create the Trust Keystore on APPHOST2.mycompany.com.

1. Create a new trust keystore called appTrustKeyStore using the keytool utility:

```
APPHOST2>keytool -keystore appTrustKeyStore.jks -genkey -keyalg RSA -alias
app TrustKey -dname "cn=appTrustKey,ou=FOR TESTING
ONLY,o=MyOrganization,L=MyTown,ST=MyState,C=US"

Enter keystore password:
Re-enter new password:
Enter key password for <appTrustKey>
RETURN if same as keystore password):
```

   Note:

   Use the standard Java keystore to create the new trust keystore because it already contains most of the needed root CA certificates. Do not to modify the standard Java trust key store directly.

2. You will be asked a series of questions. The keystore is created after you respond to these questions.

   Tip:

   Make a note of the information that you provide on the command line and in the subsequent dialog box, because you will need this information to define gateway policy steps.

3. Change the default password for the standard Java keystore utility using the keytool utility. Use the following syntax to change the default password:

```
keytool -storepasswd -keystore <TrustKeyStore>
```

4. Copy the standard Java keystore called cacerts, which is located in the ORACLE_BASE/product/FMW/wlserver_10.3/server/lib directory, to the same directory as the certificates. Copy cacerts as follows:

```
APPHOST2> cp ORACLE_BASE/product/FMW/wlserver_10.3/server/lib/cacerts
ORACLE_BASE/product/FMW/certs/appTrustKeyStore.jks
```

5. Import the CA certificate called CertGenCA.der into the appTrustKeyStore using the keytool utility. This certificate, which is located in the ORACLE_BASE/product/product/FMW/wlserver_10.3/server/lib directory, is used to

sign all certificates generated by utils.CertGen tool. Import CertGenCA.der using the following syntax:

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

Configuring Node Manager to Use the Custom Keystores

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

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

```
KeyStores=CustomIdentityAndCustomTrust
CustomIdentityKeyStoreFileName=<Identity KeyStore>
CustomIdentityKeyStorePassPhrase=<Identity KeyStore Passwd>
CustomIdentityAlias=<Identity Key Store Alias>
CustomIdentityPrivateKeyPassPhrase=<Private Key used when creating
Certificate>
```

Make sure to use the correct value for CustomIdentityAlias on each node. For example on APPHOST2, use "appIdentity2.

Example for Node 1:

```
KeyStores=CustomIdentityAndCustomTrust

CustomIdentityKeyStoreFileName=ORACLE_BASE/product/FMW/certs/appIdentityKe
yStore.jks
CustomIdentityKeyStorePassPhrase=welcome1
CustomIdentityAlias=appIdentity1
CustomIdentityPrivateKeyPassPhrase=welcome1
```

Note:

The passphrase entries in the nodemanager.properties file get encrypted when you start Node Manager, as described in the next section, "Starting the Node Manager on APPHOST2"

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

Starting the Node Manager on APPHOST2

Run these commands to start Node Manager on APPHOST2:

```
APPHOST2> cd ORACLE_BASE/product/FMW/wlserver_10.3/server/bin
APPHOST2> ./startNodeManager.sh
```

Install and Configure the Web Tier

At this point a highly available Portal configuration is now available. However for an Enterprise deployment the next stage is to separate the Web Components (Web Cache and Oracle HTTP Server) to separate servers to add security and flexibility.

Follow these steps to install the Oracle HTTP Server onto Webhost1 and Webhost2

Install and Configure the First Oracle Web Tier on Webhost1

Install Oracle HTTP Server on Webhost1

Start the Oracle Universal Installer as follows:

On UNIX, issue this command: `runInstaller`

On Windows, double-click `setup.exe`

Before Starting the install ensure that the following environments are not set.

- LD_ASSUME_KERNEL
- ORACLE_INSTANCE

Screen	Action
Welcome	Click **Next**.
Select Installation Type	Select **Install and Configure**. Click **Next**.

Prerequisite Checks	Click **Next**.
Specify Installation Location	Specify the following values: Fusion Middleware Home Location (Installation Location) for example: `/u01/app/oracle/product/FMW/web`
Configure Components	Select: Oracle HTTP Server Oracle Web Cache Associate Selected Components with WebLogic Domain Click **Next**.

Specify WebLogic Domain Details
(Optional)
	Specify the following values:
	Domain Host Name (Machine Hosting WebLogic Admin Server) for example:
	`APPHOST1.mycompany.com`
	Domain port Number (WebLogic Administration server Port) for example:
	`7001`
	Username (WebLogic Admin Server user) for example:
	`WebLogic`
	Password (Password for above account)
	Click **Next**.

Specify Component Details	Specify the following values: Instance Home Location: /u01/app/oracle/admin/web1 AS Instance Name: web1

	OHS Component Name: http1 WebCache Component Name: Web Cache1 Click **Next**.
WebCache Administrator Password	Specify a value for the Webcache administrator password. Confirm the password and click **Next**

Configure Ports	In HA implementations whilst not mandatory it makes life simpler if all of the ports used by the various components are synchronized across hosts. Oracle allows the bypassing of Automatic port Configuration by specifying ports to be used in a file.
	
	Select a File Name and then click **View/Edit**. The file will look like:
	
	`[DOMAIN]`
	
	`#This port indicates the Domain port no`
	
	`Domain Port No = 7001`
	
	`[OHS]`
	
	`#Listen port for OHS component`
	
	`OHS Port = 7780`
	
	`[WEBCACHE]`
	
	`#Port no for WebCache component (also used for virtual server port)`
	
	`Web Cache Listen Port= 7777`
	
	`#Adminstration port no for WebCache component`
	
	`Web Cache Admin Port= 9400`
	
	`#STATISTICS port no for WebCache component`
	
	`Web Cache Statistics Port = 9402`

	`#INVALIDATION port no for WebCache component`
	`Web Cache Invalidation Port = 9401`
	`[OPMN]`
	`#Process Manager Local port no`
	`Oracle Process Manager Local Port No = 1880`
	You can find a sample staticports.ini file on installation Disk1 in the stage/Response directory.
	Click **Next**.

Specify Security Updates	Choose whether or not to receive security updates from Oracle Support. Click **Next**.
Installation Summary	Review the selections to ensure that they are correct (if they are not, click **Back** to modify selections on previous screens), and click **Install**. When prompted the script oracleRoot.sh needs to be run as the root user – UNIX installations only.
Configuration	Multiple configuration assistants are launched in succession; this process can be lengthy. When it completes, click **Next** and the **Installation Complete** screen appears. Click **Finish** to confirm your choice to exit.

Validate the Installation

Once the installation is completed check that the it is possible to access the Oracle HTTP Server home page using the following URL:

http://webhost1.mycompany.com:7777/

Copy Portal Specific Files from APPHOST1

The Web Tier needs certain files such as images and configuration information to be able to display the Portal pages correctly. Copy the following directories from APPHOST1 to WEBHOST2

APPHOST1	WEBHOST1
ORACLE_HOME/portal	ORACLE_HOME/portal
ORACLE_INSTANCE/config/OHS/ohs1 /moduleconf	ORACLE_INSTANCE/config/OHS/http1 /moduleconf
ORACLE_INSTANCE/config/OHS/ohs1 /osso.conf	ORACLE_INSTANCE/config/OHS/http1 /osso.conf

Configure Virtual Hosts

On WEBHOST1 edit the file

ORACLE_INSTANCE/config/OHS/http1/moduleconf/virtual_hosts

Remove the Virtual Host entry for APPHOST1 and add an entry for WEBHOST1. The resulting file should look like:

```
NameVirtualHost *:7778
<VirtualHost *:7778>
    ServerName https://myPortal.mycompany.com:443
    RewriteEngine On
    RewriteOptions inherit
    UseCanonicalName On
</VirtualHost>

<VirtualHost *:7778>
    ServerName webhost1.mycompany.com:7777
    RewriteEngine On
    RewriteOptions inherit
    UseCanonicalName On
</VirtualHost>
```

Configure Web Cache

Log into the Enterprise Manager Administration Console

Log into the Enterprise Manager Console using the URL:

http://apphost1.mycompany.com:7001/em

Default User Name and Password are the same as the domain username and password entered during the installation.

Create Origin Server

In the Navigator window, expand the Web Tier tree.

Click on the component wc1 (make sure it is the one associated with APPHOST1)

From the drop down list at the top of the page select Administration – Origin Servers

Select **Create**

Enter the following information to add the origin server

Host	WEBHOST1.mycompany.com
Port	7778
Capacity	100
Protocol	HTTP
Failover Threshold	5
Ping URL	/
Ping Frequency	10

Select **OK** to save the changes.

Select **Apply** to save the changes.

Add Origin Server Site to Server Mapping

In the Navigator window, expand the Web Tier tree.

Click on the component wc1 (make sure it is the one associated with APPHOST1)

From the drop down list at the top of the page select Administration – Sites

In the Site to Server Mapping section click on the Host:port

myPortal.mycompany.com:443

Click on **Edit**

Select the origin server WEBHOST1.mycompany.com:7778 and move it to the selected Origin servers list.

Click **OK** to save the changes.

Select **Apply** to save the changes.

Cluster Web Cache

In the Navigator window, expand the Web Tier tree.

Click on the component wc1 (make sure it is the one associated with APPHOST1)

From the drop down list at the top of the page select Administration – Cluster

Click on **Add**

The Web Cache from WEBHOST1 will automatically be added.

Click **Apply** to apply the changes

Click on the newly created Web Cache entry (be sure not to click on the URL part of it)

Click on **Synchronize** to copy the configuration to the Web Cache on WEBHOST1.

Click **Yes** when prompted to confirm that you wish you perform the operation.

Click **Apply** to apply the new configuration

Click on the Web Cache entry (be sure not to click on the URL part of it) associated with APPHOST2

Click on **Synchronize** to copy the configuration to the Web Cache on APPHOST2.

Click **Yes** when prompted to confirm that you wish you perform the operation.

Click **Apply** to Apply the new configuration

Restart Web Tier (OHS and Web Cache)

Having made the above changes the Web Tier components need to be restarted. This can be achieved by issuing the commands:

Restart the Oracle Web Tier components using the commands:

opmnctl stopall

opmn/bin/opmnctl startall

Note: Prior to issuing these commands ensure that the environment variable ORACLE_INSTANCE is set to the value that was entered during the install above.

Note: For validation purposes only restart the Web Tier on WEBHOST1, leave the others shutdown.

Validate Configuration

In order to validate the configuration the following tests should be performed:

Test	URL	Result
Test load balancer	http://myPortal.mycompany.com/	Home page displayed
Test load balancer via SSL	https://myPortal.mycompany.com/	Home page displayed
Test load balancer Termination	https://myPortal.mycompany.com/portal/pls/portal/owa_util.print_cgi_env	REQUEST_PROTOCOL value of HTTPS

Install and configure the second Oracle Web Tier on Webhost2

This process is the same as that for installing the first Web Tier:

Install Oracle HTTP Server on Webhost2

Start the Oracle Universal Installer as follows:

On UNIX, issue this command: `runInstaller`

On Windows, double-click `setup.exe`

Before Starting the install ensure that the following environments are not set.

- LD_ASSUME_KERNEL
- ORACLE_INSTANCE

Screen	Action
Welcome	Click **Next**.
Select Installation Type	Select **Install and Configure**. Click **Next**.
Prerequisite Checks	Click **Next**.
Specify Installation Location	Specify the following values: Fusion Middleware Home Location (Installation Location) for example: `/u01/app/oracle/product/FMW/web`
Configure Components	Select: Oracle HTTP Server Oracle Web Cache Associate Selected Components with WebLogic Domain Click **Next**.
Specify WebLogic Domain Details	Specify the following values: Domain Host Name (Machine Hosting WebLogic Admin Server) for

	example:
	`wladmin.mycompany.com`
	Domain port Number (WebLogic Administration server Port) for example:
	`7001`
	Username (WebLogic Admin Server user) for example:
	`WebLogic`
	Password (Password for above account)
	Click **Next**.

Specify Component Details	Specify the following values: Instance Home Location: /u01/app/oracle/admin/web1 AS Instance Name: web2 OHS Component Name: http2 WebCache Component Name: Web Cache2 Click **Next**.
WebCache Administrator Password	Specify a value for the Webcache administrator password. Confirm the password and click **Next**
Configure Ports	Select Specify Ports using Configuration File Select the same file used for WEBHOST1 and click **Next**
Specify security updates	Choose whether or not to receive security updates from Oracle support. Click **Next**.
Installation Summary	Review the selections to ensure that they are correct (if they are not, click **Back** to modify selections on previous screens), and click **Install**. When prompted the script oracleRoot.sh needs to be run as the root user –

	UNIX installations only.
Configuration	Multiple configuration assistants are launched in succession; this process can be lengthy. When it completes, click **Next** and the **Installation Complete** screen appears. Click **Finish** to confirm your choice to exit.

Validate the Installation

Once the installation is completed check that it is possible to access the Oracle HTTP Server home page using the following URL:

http://webhost1.mycompany.com:7777/

Copy Portal Specific Files from APPHOST1

The Web Tier needs certain files such as images and configuration information to be able to display the Portal pages correctly. Copy the following directories from APPHOST1 to WBHOST2

APPHOST1	**WEBHOST2**
ORACLE_HOME/portal	ORACLE_HOME/portal
ORACLE_INSTANCE/config/OHS/ohs1 /moduleconf	ORACLE_INSTANCE/config/OHS/http 2/moduleconf
ORACLE_INSTANCE/config/OHS/ohs1	ORACLE_INSTANCE/config/OHS/http

Configure Virtual Hosts

On WEBHOST2 edit the file

ORACLE_INSTANCE/config/OHS/http1/moduleconf/virtual_hosts

Remove the Virtual Host entry for APPHOST1 and add an entry for WEBHOST2. The resulting file should look like:

```
NameVirtualHost *:7778
<VirtualHost *:7778>
    ServerName https://myPortal.mycompany.com:443
    RewriteEngine On
    RewriteOptions inherit
    UseCanonicalName On
</VirtualHost>

<VirtualHost *:7778>
    ServerName webhost2.mycompany.com:7777
    RewriteEngine On
    RewriteOptions inherit
    UseCanonicalName On
</VirtualHost>
```

Configure Web Cache

Log into the Enterprise Manager Administration Console

Log into the Enterprise Manager Console using the URL:

http://apphost1.mycompany.com:7001/em

Default User Name and Password are the same as the domain username and password entered during the installation.

Create Origin Server

In the Navigator window, expand the Web Tier tree.

Click on the component Web Cache1

From the drop down list at the top of the page select Administration – Origin Servers

Select **Create**

Enter the following information to add the origin server

Host	WEBHOST2.mycompany.com

Port	7778
Capacity	100
Protocol	HTTP
Failover Threshold	5
Ping URL	/
Ping Frequency	10

Select **OK** to save the changes.

Select Apply to save the changes.

Add Origin Server Site to Server Mapping

In the Navigator window, expand the Web Tier tree.

Click on the component Web Cache1

From the drop down list at the top of the page select Administration – Sites

In the Site to Server Mapping section click on the Host:port

myPortal.mycompany.com:443

Click on **Edit**

Select the origin server WEBHOST2.mycompany.com:7778 and move it to the selected Origin servers list.

Click **OK** to save the changes.

Select **Apply** to save the changes.

Cluster Web Cache

In the Navigator window, expand the Web Tier tree.

Click on the component Web Cache1

From the drop down list at the top of the page select Administration – Cluster

Click on **Add**

The Web Cache from WEBHOST2 will automatically be added.

Select **Apply** to apply the changes

Click on the newly created Web Cache entry (be sure not to click on the URL part of it)

Click on **Synchronize** to copy the configuration to the Web Cache on WEBHOST2.

Click **Yes** when prompted to confirm that you wish you perform the operation.

Click **Apply** to apply the new configuration

Click on the Web Cache entry (be sure not to click on the URL part of it) associated with APPHOST1

Click on **Synchronize** to copy the configuration to the Web Cache on APPHOST1.

Click **Yes** when prompted to confirm that you wish you perform the operation.

Click on the Web Cache entry (be sure not to click on the URL part of it) associated with APPHOST2

Click on **Synchronize** to copy the configuration to the Web Cache on APPHOST2.

Click **Yes** when prompted to confirm that you wish you perform the operation.

Click **Apply** to apply the new configuration

Restart Web Tier (OHS and Web Cache)

Having made the above changes the Web Tier components need to be restarted. This can be achieved by issuing the commands:

Restart the Oracle HTTP Server using the commands:

opmnctl stopall

opmnctl startall

Note: Prior to issuing these commands ensure that the environment variable ORACLE_INSTANCE is set to the value that was entered during the install above.

Note: For Validation purposes only restart the Web Tier on WEBHOST2. Leave the others shutdown.

Once validation is complete, restart the Web Tier components on WEBHOST1.

Validate Configuration

In order to validate the configuration the following tests should be performed:

Note: For ease of testing ensure that the Oracle HTTP Server on Webhost2 is the only one running.

Test	URL	Result
Test load balancer	http://myPortal.mycompany.com/	Home page displayed
Test load balancer via SSL	https://myPortal.mycompany.com/	Home page displayed
Test load balancer Termination	https://myPortal.mycompany.com/portal/pls/portal/owa_util.print_cgi_env	REQUEST_PROTOCOL value of HTTPS

Tidy up APPHOST1 and APPHOST2

Now that Web Cache and the Oracle HTTP Servers are configured and running on WEBHOST1 and WEBHOST2 there is no need for them to be started on APPHOST1 and APPHOST2. Additionally the Web Caches on these nodes should be removed from the cluster.

Remove Origin Servers from Site to Server Mapping

In the Navigator window, expand the Web Tier tree.

Click on the component Web Cache1

From the drop down list at the top of the page select Administration – Sites

In the Site to Server Mapping section click on the Host:port

myPortal.mycompany.com:443

Click on **Edit**

Select the origin servers APPHOST1.mycompany.com:7778 and APPHOST1.mycompany.com:7778 and remove them from the selected Origin servers list.

Click **OK** to save the changes.

Select **Apply** to save the changes.

Remove Origin Servers

In the Navigator window, expand the Web Tier tree.

Click on the component Web Cache1

From the drop down list at the top of the page select Administration – Origin Servers

Click on the Origin Servers APPHOST1 and APPHOST2 and click **Delete**.

Select **Apply** to save the changes.

Remove APPHOST1 and APPHOST2 from Web Cache Cluster

In the Navigator window, expand the Web Tier tree.

Click on the component Web Cache1

From the drop down list at the top of the page select Administration – Cluster

Click on the Web Caches associated with APPHOST1 and APPHOST2 and click **Delete**.

Select **Apply** to apply the changes

Click on the Web Cache entry Web Cache2 (be sure not to click on the URL part of it)

Click on **Synchronize** to copy the configuration to the Web Cache on WEBHOST2.

Click **Yes** when prompted to confirm that you wish you perform the operation.

Remove Web Cache and Oracle HTTP Server

Now that the Web Caches and Oracle HTTP Servers have been disassociated, they can be deleted form APPHOST1 and APPHOST2

APPHOST1

Before starting this operation ensure that ORACLE_HOME and ORACLE_INSTANCE are set appropriately for this host:

ORACLE_HOME=/u01/app/oracle/product/FMW/Portal

ORACLE_INSTANCE=/u01/app/oracle/admin/PortalDomain/Portal1

Issue the following command to remove the Oracle Web Cache:

opmnctl `deletecomponent` -componentName wc1 -adminHost APPHOST1 -adminPort 7001 -adminUsername WebLogic

Enter the WebLogic Administration Password when requested.

Issue the following command to remove the Oracle HTTP Server:

opmnctl deletecomponent -componentName ohs1 -adminHost APPHOST1 -adminPort 7001 -`adminUsername` WebLogic

Enter the WebLogic Administration Password when requested.

APPHOST2

Before starting this operation ensure that ORACLE_HOME and ORACLE_INSTANCE are set appropriately for this host:

ORACLE_HOME=/u01/app/oracle/product/FMW/Portal

ORACLE_INSTANCE=/u01/app/oracle/admin/PortalDomain/Portal2

Issue the following command to remove the Oracle Web Cache:

```
opmnctl deletecomponent -componentName wc1 -adminHost APPHOST1 -adminPort 7001 -
adminUsername WebLogic
```

Enter the WebLogic Administration Password when requested.

Issue the following command to remove the Oracle HTTP Server:

```
opmnctl deletecomponent -componentName ohs1 -adminHost APPHOST1 -adminPort 7001 -
adminUsername WebLogic
```

Enter the WebLogic Administration Password when requested.

Restart Web Tier (OHS and Web Cache)

Having made the above changes the Web Tier components on WEBHOST1 and WEBHOST2 need to be restarted. This can be achieved by issuing the commands:

Restart the Oracle HTTP Server using the commands:

opmnctl stopallopmnctl startall

Note: Prior to issuing these commands ensure that the environment variable ORACLE_INSTANCE is set to the value that was entered during the install above.

Scale Out

This deployment is extremely scalable. The steps to scale out the architecture are the same as those described for APPHOST2 and WEBHOST2, depending on whether it is the application or Web Tier, which is being extended.

References

1. Oracle Maximum Availability Architecture Web site
   http://www.otn.oracle.com/goto/maa

White Paper Title
November 2009
Author: Michael Rhys

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com

Oracle is committed to developing practices and products that help protect the environment

0109