Oracle Forms, Reports and Discoverer Enterprise Deployment Guide: 11.1.1.2.0

Oracle Maximum Availability Architecture White Paper
December 2009

Maximum
Availability
Architecture

Oracle Best Practices For High Availability

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 book 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

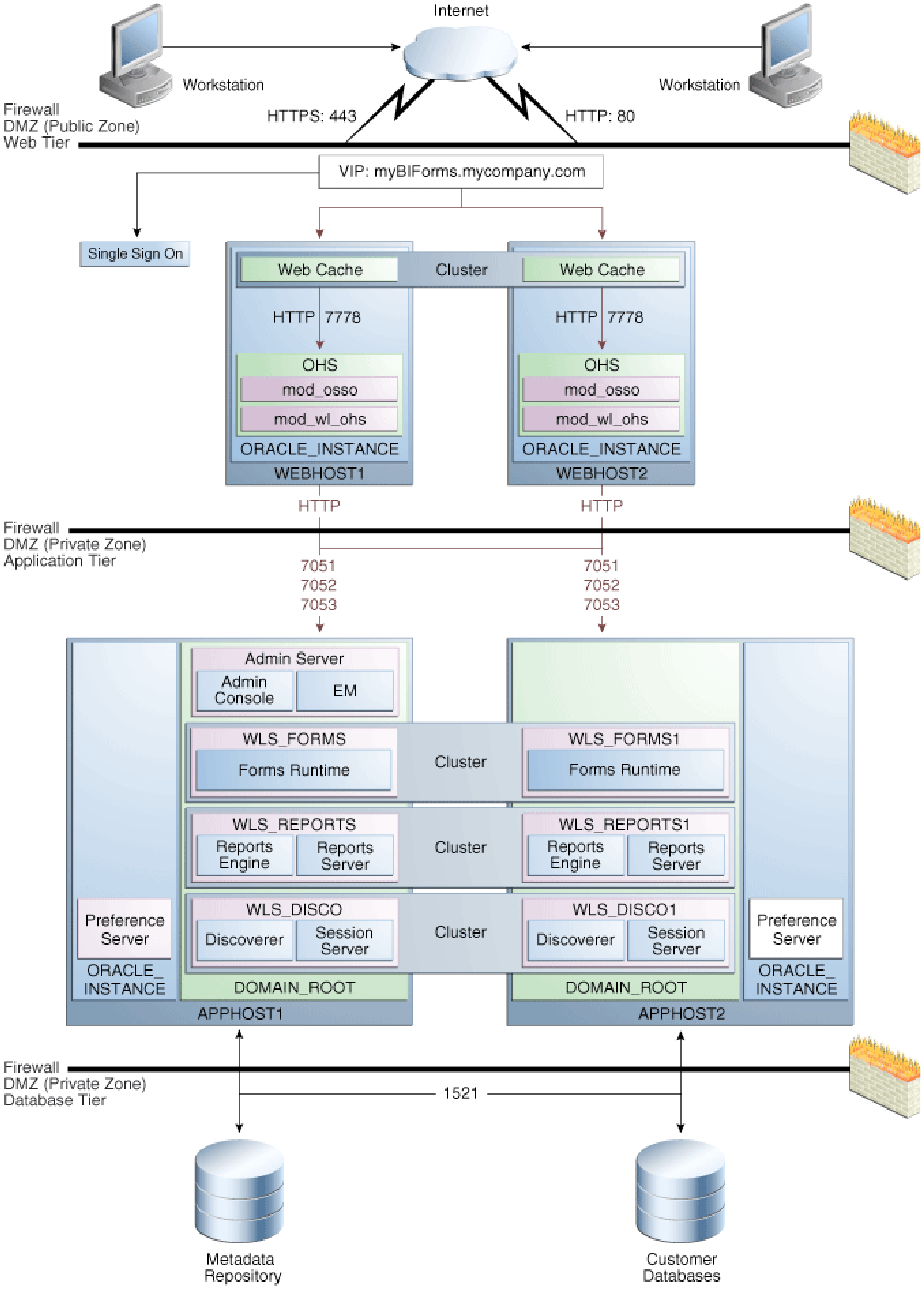
The Enterprise Deployment architectures are highly available, because each component or functional group of software components is replicated on a different computer, and configured for component-level high availability.

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 myBIForms 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.


Internet
Workstation
Workstation
Firewall
DMZ (Public Zone)
Web Tier
HTTPS: 443
HTTP: 80
VIP: myBIForms.mycompany.com
Single Sign On
Web Cache
Cluster
Web Cache
HTTP 7778
HTTP 7778
OHS
mod_osso
mod_wl_ohs
ORACLE_INSTANCE
WEBHOST1
OHS
mod_osso
mod_wl_ohs
ORACLE_INSTANCE
WEBHOST2
HTTP
HTTP
Firewall
DMZ (Private Zone)
Application Tier
7051
7052
7053
7051
7052
7053
Admin Server
Admin Console
EM
WLS_FORMS
Forms Runtime
Cluster
WLS_FORMS1
Forms Runtime
WLS_REPORTS
Reports Engine
Reports Server
Cluster
WLS_REPORTS1
Reports Engine
Reports Server
WLS_DISCO
Discoverer
Session Server
Cluster
WLS_DISCO1
Discoverer
Session Server
Preference Server
ORACLE_ INSTANCE
DOMAIN_ROOT
APPHOST1
DOMAIN_ROOT
Preference Server
ORACLE_ INSTANCE
APPHOST2
Firewall
DMZ (Private Zone)
Database Tier
1521
Metadata Repository
Customer Databases

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 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_FORMS and WLS_FORMS1 are clustered together and mod_wl_ohs will load balance requests amongst them. The same is true for WLS_REPORTS and WLS_REPORTS1 and WLS_DISCO and WLS_DISCO1.

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 or port 80. 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 deployed 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

Each of the products, Forms, Reports and Discoverer interact with databases. In the case of Forms and Reports, these products interact with customer applications. Discoverer also requires access to its own metadata. This metadata is used to provide a real worldview on the raw customer data, which users use for their business intelligence applications.

The Oracle Discoverer metadata consists of:

- Discoverer End User Layer
- Discoverer Work Books
- Analytic Workspaces

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 or 11.2)
Oracle WebLogic Server	WebLogic Server 10.3 CD
Oracle Forms, Reports and Discoverer	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 Forms, Reports and Discoverer 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/BIForms	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/BIForms1	N	Oracle Instance (APPHOST1)
/u01/app/oracle/admin/BIForms2	N	Oracle Instance (APPHOST2)
/u01/app/oracle/admin/web1	N	Oracle Instance (WEBHOST1)
/u01/app/oracle/admin/web2	N	Oracle Instance (WEBHOST2)
/shared/reports	Y	Reports Output Cache
/shared/disco/pref	Y	Discoverer Portlet Preference Store.

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.

Table below lists the ports used in the Oracle Forms, Reports and Discoverer 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.

Component	Ports	Firewall	Protocol	Inbound/ Outbound	Comments
Browser request	443	FW0	HTTPS/LBR1	In/out	
Browser request	80	FW0	HTTP/LBR1	In/out	
LBR to WC	7777 9401	FW1	HTTP	NA	NA

	9402				
WC to OHS	7778	NA	HTTP	In/out	
OHS to WLS	7051 7052 7053	FW2	HTTP	In/out	
Admin Console Access	7001	Depends	HTTP/Admin Server-EM and t3	In/out	Administrators need access to the 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	1521	FW2	SQLNET	In/out	
WC Invalidation Requests	9401	FW3	HTTP	Out	
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

In order to configure the myBIForms.mycompany.com web site a site name is required. This site name (myBIForms.mycompany.com) must be defined in DNS and be associated with the Virtual IP address assigned to the Load balancer.

The following site names are used in this enterprise deployment:

Name	Purpose

myBIForms.mycompany.com	Forms 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, if required.

Purpose	Host(s)	Port	Comment
myBIForms.mycompany.com	Load Balancer	443	SSL port on the Load Balancer
myBIForms.mycompany.com	Load Balancer	80	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_REPORTS	APPHOST1	7051	WebLogic Managed Server Port
WLS_REPORTS1	APPHOST2	7051	WebLogic Managed Server Port
WLS_FORMS	APPHOST1	7052	WebLogic Managed Server Port
WLS_FORMS1	APPHOST2	7052	WebLogic Managed Server Port
WLS_DISCO	APPHOST1	7053	WebLogic Managed Server Port
WLS_DISCO1	APPHOST2	7053	WebLogic Managed Server Port
Internet Directory	SSOHOST	389/ 4443	OID HTTP/HTTPS ports
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
WebLogic Domain Name	BIForms	Name assigned to the WebLogic domain
WebLogic Admin User	WebLogic	WebLogic Administrative 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. Configure Shared Storage, for the Discoverer Portlet Preference store and the Report Cache.
4. Create a Highly Available Database to store the Reports and Discoverer metadata.

5. Create a Discoverer metadata repository in the newly created database using the Repository Creation Utility.
6. Install WebLogic Server on APPHOST1.
7. Install and perform initial configuration of Oracle Forms, Reports and Discoverer on APPHOST1.
8. Configure Oracle HTTP Server on APPHOST1.
9. Configure Oracle Web Cache on APPHOST1 (If required).
10. Register with Oracle Single Sign On.
11. Configure host Assertion in Oracle WebLogic Server.
12. Create a global reports queue in the database.
13. Configure reports to use the global reports queue and shared storage for the reports cache.
14. Configure Oracle Discoverer.
15. Install Oracle WebLogic Server on APPHOST2.
16. Install and perform initial configuration of Oracle Forms, Reports and Discoverer on APPHOST2.
17. Copy Files from APPHOST1 to APPHOST2.
18. Introduce APPHOST2 to Web Cache (If required).
19. Cluster Web Cache Instances on APPHOST1 an APPHOST2.
20. Make Oracle HTTP Server WebLogic Cluster aware.
21. Create a Reports Server Cluster.
22. Configure Discoverer preference store.
23. Configure Discoverer Portlet preference store.
24. Install Oracle Web Tier on WEBHOST1 and WEBHOST2.
25. Introduce WEBHOST1 and WEBHOST1 to Web Cache Cluster (if required).
26. Copy files from APPHOST1 to WEBHOST1 and WEBHOST2.
27. Tidy up installation.

Configuring the Network for Enterprise Deployments

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

Oracle Forms, Reports and Discoverer 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.

Configuring the load balancer differs depending on whether or not Oracle Web Cache is used in the implementation.

Configuring the Load Balancer with Web Cache

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 myBIForms.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 myBIForms.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 is used to propagate web cache invalidation requests.
- 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_webcache_static_.html

For example:

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

Note:

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

To summarize, the load balancer requires the following configuration:

Virtual Host	Virtual Port	Server Pool	Server	Port	Comments
myBIForms.mycompany.com	443	MyBIForms	WEBHOST1	7777	Protocol Conversion Required
			WEBHOST2	7777	
myBIForms.mycompany.com	9401	Invalidation	WEBHOST1	9401	Invisible to the external clients.
			WEBHOST2	9401	

Configuring the Load Balancer without Web Cache

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 myBIForms.mycompany.com on port 443 (an HTTPS listening port), and balances them to the application tier Oracle HTTP Servers running on WEBHOST1 and WEBHOST2 port 7777 (an HTTP listening port). You must configure the load balancing router to perform protocol conversion.
- HTTP monitoring of Oracle HTTP Server. The load balancing router must be configured to detect an inoperative computer and stop routing requests to it until it is functioning again.

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

  hostname:port/

  For example:

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

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

  HTTP/1.0

  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".

To Summarize, the load balancer requires the following configuration:

Virtual Host	Virtual Port	Server Pool	Server	Port	Comments
myBIForms.mycompany.com	443	MyBIForms	WEBHOST1	7777	Protocol Conversion Required
			WEBHOST2	7777	

Configuring the Database for Enterprise Deployments

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

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

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

Database versions supported

Minimum:

- 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) and 11g Release 2 (11.2), 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.
- For 11g Release 2 (11.2), 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, Reports and Discoverer using the following instructions:

Note: Forms doesn't support full failover because pl/sql package state doesn't migrate/failover. If the connection an Oracle Forms session is using is lost. The user must establish a new connection to a surviving node to continue. Oracle Forms can however integrate with high availability notification, see below for details.

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 => 'biforms.mycompany.com',
   NETWORK_NAME => 'biforms.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);
   ```

   Oracle Forms also integrates with high availability event notification. To enable this feature for Oracle Forms:

Use the Oracle Enterprise Manager Cluster Managed Services Page to create database services. For Oracle Forms, set the Oracle RAC DBMS_SERVICE property values according to the following table. database.

Table 4-5 Oracle Forms Database Services Property Settings

Property Name	Value
AQ_HA_NOTIFICATIONS	TRUE
FAILOVER_METHOD	DBMS_SERVICE.FAILOVER_METHOD_NONE
FAILOVER_TYPE	DBMS_SERVICE.FAILOVER_TYPE_NONE

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

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

3. Start the service using srvctl:

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

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 => 'biforms.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);
```

Note:

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	Specify the following values:

Components	Create New Prefix: Enter a prefix to be added to database schemas. For example: MYD Components: Check AS Common Schemas -> Metadata Services Portal and BI -> Discoverer 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 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 Shared Storage for Enterprise Deployments

Create a Shared Directory for Report Output

Each report server in a highly available configuration needs to have access to common directory where report output can be placed. This is required so that report output is available to all report servers. So in the event of the failure of the report server that generated the report, another server can still access the generated output.

This folder should be writeable from all servers hosting a reports server.

This document will assume that /shared/reports will be used for this purpose.

Create a Shared Directory for the Portlet Preference Store

The portlet preference store is used for persisting consumer registration handles and portlet preference data.

Discoverer WSRP portlet producer uses a file-based preference store. In a clustered environment, for the file-based preference store, all Discoverer WSRP portlet producers running within the same Oracle WebLogic Server must use the same path for the preference store.

This document will assume that /shared/disco/pref will be used.

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 as 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**.
JDK Selection	Click **Next**.
Choose Product Installation Directories	Click **Next**.
Installation Summary	Click **Next**.
Installation Complete	Uncheck runQuickstart and Click **Done.**

Install Oracle Forms, Reports and Discoverer Software

The next step in the installation procedure is to install Oracle 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**
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 For example BIFormsDomain 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 the Oracle Binaries. ** Note this will be placed under the MW_HOME directory. For example `BIForms` 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/BIForms1` Oracle Instance Name: BIForms1 Click **Next**
Configure Components	As a minimum ensure that the following values are checked: Server Components – Oracle Forms Oracle Reports Oracle Discoverer 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

```
[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]
Oracle WLS Reports Managed Server port No = 7051
```

	`Oracle WLS Forms Managed Server port No = 7052`
	`Oracle WLS Discoverer Managed Server port No = 7053`
	You can find a sample staticports.ini file on installation Disk1 in the stage/Response directory.
	Save the file and click **Next**

	Save the file and click **Next**
Specify Application Identity Store	Specify the following values: Hostname: Name of oid server For example login.mycompany.com Port: OID port For example 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.

Validate Configuration

Validate the initial installation by performing the following tests.

Test	URL	Result
Forms	http://apphost1.mycompany.com:7777/forms/frmservlet	Test Form is displayed
Discoverer	http://apphost1.mycompany.com:7777/discoverer/viewer	Discoverer Viewer Home Page displayed
Reports Queue	http://apphost1.mycompany.com:7777/reports/rwservlet/showjobs	Job Queue is displayed.

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.

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

Create a TNSNAMES entries for Customer Databases

If the application is to access one or more databases then an entry for each database being accessed must be placed into the file tnsnames.ora.

Edit the file ORACLE_INSTANCE/config/tnsnames.ora and add an entry similar to the one below:

```
mydb.mycompany.com =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = mydbnode1-vip)(PORT = 1521))
    (ADDRESS = (PROTOCOL = TCP)(HOST = mydbnode2-vip)(PORT = 1521))
    (LOAD_BALANCE = yes)
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = mydb.mycompany.com)
    )
  )
```

NOTE: This is a RAC database connect string.

Save the file and test that the database connection is configured correctly using the command:

tnsping mydb.mycompany.com

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 Forms, Reports and Discoverer to work with the load balancer two virtual hosts need to be defined.

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://myBIForms.mycompany.com:443
    RewriteEngine On
    RewriteOptions inherit
    UseCanonicalName On
</VirtualHost>

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

Save the file and Restart the Web Tier using the commands:

```
opmnctl restartproc ias-component=ohs1
```

Configure Web Cache

Web Cache is not strictly required for Forms, Reports and Discoverer to operate. However many Enterprise Deployments co-exist with other components such as Portal which do require Web Cache. By including this section, the steps required to integrate Web Cache are listed. If the environment being configured won't use Web Cache then ignore the Web Cache configuration steps below.

Log into 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 and administration passwords, confirm and click **Apply**

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: myBIForms.mycompany.com

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

Leave everything else at the default, and then click submit.

Select **OK** to save each entry

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	myBIForms.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.

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 Web Tier components using these 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.

Register with SSO

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

1. Set the ORACLE_HOME variable to the SSO *ORACLE_HOME* location.
2. Execute ORACLE_HOME/sso/bin/ssoreg.sh (ssoreg.bat on Windows) with the following parameters

   ```
   -site_name myBIForms.mycompany.com
   -mod_osso_url https://myBIForms.mycompany.com
   -config_mod_osso TRUE
   -oracle_home_path /u01/app/oracle/product/fmw/BIForms
   -config_file /tmp/osso.conf
   -admin_info cn=orcladmin
   -virtualhost
   -remote_midtier
   ```

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

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.

Login to 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 each of the following Cluster Names in turn cluster_disco, cluster_Forms and cluster_reports.

Select HTTP and enter the following values:

Parameter	Value
Frontend Host	myBIForms.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.

Configure Oracle Forms

There are no specific configuration steps for node 1 and Forms.

Restart WLS_FORMS

Having made the above changes the WebLogic managed server WLS_FORMS needs to be restarted.

This is achieved by logging into the WebLogic administration console using the following URL

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

Select Environment -> Servers from the Domain Structure menu.

Select the Control tab

Select the box next to the server WLS_FORMS

Select Shutdown -> Force shutdown now.

Click yes in the confirmation dialogue box.

When the server is shutdown restart it by following these steps:

Select the box next to the server WLS_FORMS

Select Start.

Validate Configuration

Validate the installation by performing the following tests.

Test	URL	Result
Test Load balancer	http://myBIForms.mycompany.com/	Home page displayed
Test Load Balancer via SSL	https://myBIForms.mycompany.com/	Home page displayed
Forms	https://myBIForms.mycompany.com/forms/frmservlet	Test Form is displayed

Configure Oracle Reports

Create Reports Queue in Database

To maintain a consistent reports queue across multiple Reports server instances and to be resilient to the failure of a reports server it is necessary to create the reports queue in a highly available Real Application Clusters database.

This is achieved by running the sqlplus script rw_server.sql against the database.

This script is located in ORACLE_HOME/reports/admin/sql

First a user needs to be created to hold the report queue in the database by issuing the commands:

```
sql> create user report_queue identified by mypassword;
sql> grant connect, resource,create view to report_queue;
```

Now connect to the reports user and execute the above script.

```
cd ORACLE_HOME/reports/admin/sql/
sqlplus report_queue/mypasswd
```

```
sql> @ rw_server.sql
```

Create a TNSNAMES entry for Reports Queue

Oracle Reports uses entries in the tnsnames.ora file to determine database connection information. It is therefore required that an entry be placed into this file for the reports queue database.

Edit the file ORACLE_INSTANCE/config/tnsnames.ora and add an entry similar to the one below:

```
myrepq.mycompany.com =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = mydbnode1-vip)(PORT = 1521))
    (ADDRESS = (PROTOCOL = TCP)(HOST = mydbnode2-vip)(PORT = 1521))
    (LOAD_BALANCE = yes)
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = myrepq.mycompany.com)
    )
  )
```

Save the file and test that the database connection is configured correctly using the command:

tnsping myrepq.mycompany.com

Create a Security Key for the Reports Queue

Reports security is performed using an indirect model. Before the reports server can be configured to use the database reports queue, an entry needs to be made in WebLogic to hold the reports queue password.

This is achieved by logging into Enterprise Manager using the URL:

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

Provide the WebLogic Administration user and password when prompted to do so.

In the navigation tree on the left hand side, expand WebLogicDomain and click on the Name of the domain for example ReportsDomain. The ReportsDomain Overview page will be displayed.

From the drop down menu at the top of the page select Security -> Credentials.

Click **Create Key**

Provide the following information:

Select Map	reports

Key	queuePassword
Type	Password
User Name	report_queue
Password	password for the reports queue account
Description	Password for the reports queue account

Click **OK** when finished.

Configure the database job repository for in-process Reports Servers

Now that the reports queue has been placed into the database, the reports server needs to be told how to access it.

This is achieved by using the Enterprise Manager Fusion Middleware Console:

Log into Enterprise Manger Fusion Middleware Console.

Expand the Reports tab on the left hand side and click on reports (<version>)(WLS_REPORTS).

From the Reports menu at the top of the screen Select Administration -> Advanced Configuration.

Select Enable Job Repository DB

Enter User Name:report_queue

Enter Password Key: csf:reports:queuePassword

Enter Database: `myrepq.mycompany.com`

Click **Apply** to save the changes.

Create a Shared Directory for Report Output

Each report server in a highly available configuration needs to have access to a common directory where report output can be placed. This is required so that report output is available to all report servers. So in the event of the failure of the report server that generated the report, another server can still access the generated output.

This folder should be writeable from all servers hosting a reports server.

Configure the Reports Server to Access Shared output directory.

Add the CacheDir or JOCCacheDir property to the <cache> element of the file rwserver.conf which is located in: DOM_HOME/config/fmwconfig/servers/WLS_REPORTS/applications/reports_<version>/configuration/

For example:

```
<property name="JOCCacheDir" value="folder_name"/>
 <property name="CacheDir" value="folder_name"/>
```

on UNIX:

```
<cache class="oracle.reports.cache.RWCache">
      <property name="cacheSize" value="50"/>
      <property name="JOCcacheDir" value="/shared/reports"/>
      <property name="cacheDir" value="/shared/reports"/>
</cache>
```

Restart WLS_REPORTS

Having made the above changes the WebLogic managed server WLS_REPORTS needs to be restarted.

This is achieved by logging into the WebLogic administration console using the following URL

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

Select Environment -> Servers from the Domain Structure menu.

Select the Control tab

Select the box next to the server WLS_REPORTS

Select Shutdown -> Force shutdown now.

Click yes in the confirmation dialogue box.

When the server is shutdown restart it by following these steps:

Select the box next to the server WLS_REPORTS

Select Start.

Validation

Validate the initial Reports installation by performing the following tests.

Test	URL	Result
Reports	https://myBIForms.mycompany.com	Reports Queue is

Queue	/reports/rwservlet/showjobs	displayed.

An alternative test is to download a sample report from OTN and run it.

Configure Oracle Discoverer

Update configuration.xml

The configuration.xml file stores information about the discoverer configuration. Edit this file which is located in:DOM_HOME/config/fmwconfig/servers/WLS_DISCO/applications/discoverer_<version>/configuration/

Update the line beginning:

applicationURL= and change the URL to:

https://myBIForms.mycompany.com/discoverer

For example:

```
applicationURL='https://myBIForms.mycompany.com/discoverer">
```

Save the file.

Discoverer Viewer and Web Cache

By default Discoverer viewer is not configured to make full use of Oracle Web Cache. When enabled significant performance gains can be attained. However it is not always appropriate to enable this functionality.

For details on when and how to enable Discoverer viewer with Web Cache see:

Oracle Business Intelligence Discoverer Configuration Guide 11g.

Restart WLS_DISCO Managed Servers

Restart the WLS_DISCO 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_DISCO

Select Shutdown -> Force Shutdown Now

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

Once the server is shutdown

Select the box next to WLS_DISCO

Click on Start

Click **Yes** to start the managed server

Validate Configuration

Validate the installation by performing the following tests.

Test	URL	Result
Test Load balancer	http://myBIForms.mycompany.com/	Home page displayed
Test Load Balancer via SSL	https://myBIForms.mycompany.com/	Home page displayed
Discoverer	https://myBIForms.mycompany.com/discoverer/viewer	Discoverer Viewer Home Page displayed

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_linux32.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 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**.
JDK Selection	Click **Next**.
Choose Product Installation Directories	Click **Next**.
Installation Summary	Click **Next**.
Installation Complete	Uncheck runQuickstart and click **Done.**

Install Oracle Forms, Reports and Discoverer Software

The next step in the installation procedure is to install Oracle Forms 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**
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:7001 User Name: Admin Server administrator account name. Password: Admin Server Password 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 ** Note this will be placed under the MW_HOME directory. For example `BIForms` 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/BIForms2`

	Oracle Instance Name: BIForms2 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 Forms Oracle Reports Oracle Discoverer 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 For example login.mycompany.com Port: OID port For example 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.

Validate Configuration

Validate the installation by performing the following tests.

Test	URL	Result
Forms	http://apphost2.mycompany.com:7777/forms/frmservlet	Test Form is displayed

Discoverer	http://apphost2.mycompany.com:7777/discoverer/viewer	Discoverer Viewer Home Page displayed
Reports Queue	http://apphost2.mycompany.com:7777/reports/rwservlet/showjobs	Job Queue is displayed.

Configure application tier on APPHOST2

Create a TNSNAMES entries for Customer Databases

If the application is to access one or more databases, then an entry for each database being accessed must be placed into the file tnsnames.ora.

Edit the file ORACLE_INSTANCE/config/tnsnames.ora and add an entry similar to the one below:

```
mydb.mycompany.com =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = mydbnode1-vip)(PORT = 1521))
    (ADDRESS = (PROTOCOL = TCP)(HOST = mydbnode2-vip)(PORT = 1521))
    (LOAD_BALANCE = yes)
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = mydb.mycompany.com)
    )
  )
```

NOTE: This is a RAC database connect string.

Save the file and test that the database connection is configured correctly using the command:

tnsping mydb.mycompany.com

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 Oracle Forms, Reports and Discoverer to work with the load balancer completely two virtual hosts need to be defined.

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://myBIForms.mycompany.com:443
    RewriteEngine On
    RewriteOptions inherit
    UseCanonicalName On
</VirtualHost>

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

Save the file

Restart Web Processes on APPHOST1 and APPHOST2

Restart the Web Tier components on APPHOST1 and APPHOST2 by issuing the following commands on each of the servers:

```
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.

Configure Web Cache

Log into Enterprise Manager Administration Console

Log into 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.

Default User Name and Password are administrator/administrator

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 and administration passwords, confirm and click **Apply**

NOTE: Use the same passwords as used in APPHOST1.

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 Interval	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

myBIForms.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:

ORACLE_HOME/opmn/bin/opmnctl restartproc ias-component=wc1

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.

Configure Oracle Forms

Update Oracle HTTP Server configuration to be cluster aware.

When the installation was first created it was configured so that all WebLogic requests are directed to the managed server WLS_FORMS residing on APPHOST1. Now that APPHOST2 exists, both the Oracle HTTP Servers on APPHOST1 and APPHOST2 need to be made aware of each other.

Edit the file ORACLE_INSTANCE/config/OHS/ohs1/moduleconf/forms.conf

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

/forms

Change the line beginning with WebLogicCluster to look something like.

WebLogicCluster apphost1:9001,apphost2:9001

For example

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

Copy Forms Configuration Files

Copy the following configuration directories/files from APPHOST1 to APPHOST2

APPHOST1	APPHOST2
$DOM_HOME/config/fmwconfig/servers/WLS_FORMS/applications/formsapp_<version>/config	$DOM_HOME/config/fmwconfig/ser IS1/applications/formsapp_<version>/c
ORACLE_INSTANCE/config/FormsComponent/forms	ORACLE_INSTANCE/config/FormsComponent/forms
ORACLE_INSTANCE/config/FRComponent/frcommon	ORACLE_INSTANCE/config/FRComponent/frcommon
ORACLE_INSTANCE/config/OHS/ohs1/moduleconf/forms.conf	ORACLE_INSTANCE/config/OHS/ohs1/moduleconf/forms.conf

Update default.env

Having copied the above files, the file default.env located in $DOM_HOME/config/fmwconfig/servers/WLS_FORMS/applications/formsapp_<version>/config needs to be updated with the correct values for APPHOST2. In particular the following entries need to be changed:

ORACLE_INSTANCE

TNS_ADMIN

FORMS_PATH

WEBUTIL_CONFIG

Typically these entries will have the following values:

```
ORACLE_INSTANCE=/u01/app/oracle/admin/BIForms1
TNS_ADMIN=/u01/app/oracle/admin/BIForms1/config
FORMS_PATH=/u01/app/oracle/product/fmw/BIForms/BIforms:/u01/app/oracle/admin/BIFor
ms1/FormsComponent/forms
WEBUTIL_CONFIG=/u01/app/oracle/admin/BIForms1/config/FormsComponent/forms/server/w
ebutil.cfg
```

And will need to be changed to:

```
ORACLE_INSTANCE=/u01/app/oracle/admin/BIForms2
TNS_ADMIN=/u01/app/oracle/admin/BIForms2/config
FORMS_PATH=/u01/app/oracle/product/fmw/BIForms/forms:/u01/app/oracle/admin/BIForms
2/FormsComponent/forms
WEBUTIL_CONFIG=/u01/app/oracle/admin/BIForms2/config/FormsComponent/forms/server/w
ebutil.cfg
```

Configure Oracle Reports

Update Oracle HTTP Server configuration to be cluster aware.

When the installation was first created it was configured with all WebLogic requests are directed to the managed server WLS_REPORTS residing on APPHOST1. Now that APPHOST2 exists, both the Oracle HTTP Servers on APPHOST1 and APPHOST2 need to be made aware of each other.

Edit the file ORACLE_INSTANCE/config/OHS/ohs1/moduleconf/reports_ohs.conf

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

/Reports

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

WebLogicCluster apphost1:9001,apphost2:9001

For example

Change

```
<Location /reports>
    SetHandler WebLogic-handler
    WebLogicHost apphost1.mycompany.com
    WebLogicport 9001
</Location>
```

to:

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

Create a TNSNAMES entry for Reports Queue

Oracle Reports uses entries in the tnsnames.ora file to determine database connection information. It is therefore required that an entry must be placed into this file for the reports queue database.

Edit the file ORACLE_INSTANCE/config/tnsnames.ora and add an entry similar to the one below:

```
myrepq.mycompany.com =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = mydbnode1-vip)(PORT = 1521))
    (ADDRESS = (PROTOCOL = TCP)(HOST = mydbnode2-vip)(PORT = 1521))
    (LOAD_BALANCE = yes)
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = myrepq.mycompany.com)
    )
  )
```

Save the file and test that the database connection is configured correctly using the command:

tnsping myreportq.mycompany.com

Configure the Reports Server to Access Shared output directory.

Add the CacheDir or JOCCacheDir property to the <cache> element of the file rwserver.conf which is located in DOM_HOME/config/fmwconfig/servers/WLS_REPORTS/applications/reports_<version>/configuration

For example:

<property name="JOCCacheDir" value="folder_name"/>

<property name="CacheDir" value="folder_name"/>

on UNIX:

```
<cache class="oracle.reports.cache.RWCache">
  <property name="cacheSize" value="50"/>
  <property name="JOCCacheDir" value="/shared/reports"/>
  <property name="CacheDir" value="/shared/reports"/>
```

```
        </cache>
```

Configure Database job repository for in-process Reports Servers

Now that the reports queue has been placed into the database, the reports server needs to be told how to access it.

This is achieved by using the Enterprise Manager Fusion Middleware Console:

Log into Enterprise Manger Fusion Middleware Console.

Expand the Reports tab on the left hand side and click on reports (<version>)(WLS_REPORTS1).

From the Reports menu at the top of the screen Select Administration -> Advanced Configuration.

Select Enable Job Repository DB

Enter User Name:report_queue

Enter Password Key: csf:reports:queuePassword

Enter Database: `myrepq.mycompany.com`

Click **Apply** to save the changes.

Create a Reports Server Cluster

By creating a Reports cluster with a database reports queue it is possible to link all of the reports servers to the same queue. The benefit is that when a report server has spare capacity it can take and execute the next report in the queue thereby distributing the load. It also ensures that if a cluster member becomes unavailable another report server can detect this and run any reports the downed server was working on.

Creating a Reports cluster is achieved by adding the a cluster entry into the file rwservlet.properties. This needs to be done on both APPHOST1 and APPHOST2

Cluster APPHOST1

Edit the file rwservlet.properties which is located at:
DOM_HOME/config/fmwconfig/servers/WLS_REPORTS1/applications/reports_<version>/configuration

Add the following line:

```
<cluster clustername="cluster_reports"
clusternodes="rep_wls_reports1_APPHOST2_reports2"/>
```

Note: The value of clusternodes will be the value which appears in the <server> tag in the file rwservlet.properties which exists on APPHOST2.

Note: The clusternodes parameter should list all of the reports servers in the cluster (comma separated) EXCEPT the local report server.

Cluster APPHOST2

Edit the file rwservlet.properties which is located at:
DOM_HOME/config/fmwconfig/servers/WLS_REPORTS1/applications/reports_<version>/configuration

Add the following line:

```
<cluster clustername="cluster_reports"
clusternodes="rep_wls_reports_APPHOST1_reports1"/>
```

Note: The value of clusternodes will be the value which appears in the <server> tag in the file rwservlet.properties which exists on APPHOST1.

Note: The clusternodes parameter should list all of the reports servers in the cluster (comma separated) EXCEPT the local report server.

Configure Oracle Discoverer

Update Oracle HTTP Server Configuration to be Cluster Aware.

When the installation was first created, it was configured so that all WebLogic requests were directed to the managed server WLS_DISCO 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/module_disco.conf

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

/discoverer

Change the WebLogicCluster diective. Initially the entry will look like:

WebLogicCluster apphost1:7052

but this needs changing to include apphost2

WebLogicCluster apphost1:7052,apphost2:7052

For example

Change

```
<Location /Discoverer>
    SetHandler WebLogic-handler
    WebLogicCluster apphost1.mycompany.com
</Location>
```

to:

```
<Location /Discoverer>
    SetHandler WebLogic-handler
    WebLogicCluster
apphost1.mycompany.com:7052,apphost2.mycompany.com:7052
</Location>
```

Restart the Oracle HTTP Server using the command:

```
opmnctl restartproc process-type=OHS
```

Copy Discoverer Configuration Files

Prior to performing this step make a backup of the file configuration.xml located in DOM_HOME/servers/WLS_DISCO/stage/discoverer/<version>/discoverer/configuration/

This is necessary because some of the contents are required later on.

Copy the file configuration.xml from APPHOST1 to APPHOST2

Copy the contents of ORACLE_INSTANCE/config/OHS/ohs1/moduleconf from APPHOST1 to APPHOST2

Update configuration.xml

Update the configuration.xml file copied above and replace the following values with the values located in configuration.xml.orig

<oracleInstance>

<discovererComponentName>

Preference Store

In an enterprise deployment there should only be one Discoverer preference store active at one time. Changing the preference store is a two-stage process.

1. Change the Preference store on APPHOST2 to point to the preference store on APPHOST1
2. Disable the preference store on APPHOST2

Identify Preference Store Host Name and Port

The nominated preference server in this example will be configured to run on APPHOST1.

When installed onto the server the Preference server will have been assigned a port. This value needs to be located before proceeding to the next steps.

On APPHOST1 open the file opmn.xml which is located in:

ORACLE_INSTANCE/config/OPMN/opmn

Search the file for the entry PREFERENCE_PORT. The value= shows the port assigned to the preference server.

Specifying a Discoverer Preferences Server on Other Machines

Having identified the host name and port number of the machine that is going to run

the Preferences component (that is, the Discoverer Preferences server machine), you must now ensure that other machines use the Preferences component on the

Discoverer Preferences server machine.

To modify the opmn.xml file of other machines to use the Discoverer Preferences

server machine, perform the following steps on every other machine in the

installation:

1. On each machine except the Preferences server machine, open the `opmn.xml` file in a text editor (or XML editor) – The file is located in ORACLE_INSTANCE/config/OPMN/opmn

2. Locate the PREFERENCE_HOST variable, and change its value to the host name of the Discoverer Preferences server machine, as follows:

   ```
   <variable id="PREFERENCE_HOST" value="hostname">
   ```

3. Locate the PREFERENCE_PORT variable, and change its value to the port number of the Discoverer Preferences server machine, as follows:

   <variable id="PREFERENCE_PORT" value="port">

4. Locate the PreferenceServer process type, and change its status to disabled, as follows:

   ```
   <process-type id="PreferenceServer" module-
   id="Disco_PreferenceServer" working-dir="$DC_LOG_DIR"
   status="disabled">
   ```

5. Save the opmn.xml file.

Restart Web Processes on APPHOST1 and APPHOST2

Restart the Web Tier components on APPHOST1 and APPHOST2 by issuing the following commands on each of the servers:

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.

Setting up Discoverer WSRP Portlet Producer in a Clustered Environment

The portlet preference store is used for persisting consumer registration handles and portlet preference data.

Discoverer WSRP portlet producer uses a file-based preference store and the location of preference store is defined by the value of the *discoWsrpPrefStoreSharedPath* variable of the Discoverer deployment plan (an XML file). The default value of the *discoWsrpPrefStoreSharedPath* variable is portletData.

In a clustered environment, for the file-based preference store, all Discoverer WSRP portlet producers running within the same Oracle WebLogic Server must use the same path for the *discoWsrpPrefStoreSharedPath* variable in the deployment plan. Therefore, the value of the *discoWsrpPrefStoreSharedPath* variable must be changed to a shared path in the deployment plan.

When you change the `discoWsrpPrefStoreSharedPath` variable, if you want to migrate the preference store content from the existing preference store to a shared path, you must run the migration utility to transfer preference data from the source path to the destination path as described in the section "Using the Migration Utility to Transfer Preference Store Content".

Setting up the Preference Store

To view and edit the Discoverer deployment plan by using WebLogic Server Administration Console:

1. Log into Oracle WebLogic Server Administration Console.
2. In the left pane, click **Deployments**.
3. In the right pane, select the Discoverer application for which you want to update the deployment plan. The Discoverer: Overview page appears.

The Deployment Plan field on the Overview page displays the path of the Discoverer application deployment plan (an XML file). Modify the deployment plan as described the following procedure:

1. Open the deployment plan from the location (as displayed on the Discoverer: Overview page) in an XML editor.
2. Navigate to the **`variable-definition`** section:

```
<variable-definition>
  <variable>
    <name>discoWsrpPrefStoreSharedPath</name>
    <value>portletData</value>
  </variable>
<variable-definition>
```

Note:□ The *discoWsrpPrefStoreSharedPath* variable is defined in the **variable-definition** and **variable-assignment** sections of the deployment plan. You should modify only the *discoWsrpPrefStoreSharedPath* variable, which is defined in the **variable-definition** section.

3. Change the **value** of the *discoWsrpPrefStoreSharedPath* variable to a shared directory which all Managed Servers can access:

```
<variable-definition>
  <variable>
    <name>discoWsrpPrefStoreSharedPath</name>
    <value>/shared/disco/pref</value>
  </variable>
```

```
<variable-definition>
```

4. Save your changes and close the XML file.

Once the deployment plan is updated with the new value for the *discoWsrpPrefStoreSharedPath* variable, you must update the Discoverer application as described in the "Update (redeploy) an Enterprise application" section of Oracle Fusion Middleware Oracle WebLogic Server Administration Console Online Help.

Restart WebLogic Managed Servers

Restart the WLS_FORMS1, WLS_REPORTS1 and WLS_DISCO1 managed servers by:

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

Select the Control tab

Select the boxes next to WLS_FORMS1, WLS_REPORTS1 and WLS_DISCO1

Select Shutdown -> Force Shutdown Now

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

Once the server is shutdown

Select the boxes next to WLS_FORMS1, WLS_REPORTS1 and WLS_DISCO1

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	http://myBIForms.mycompany.com/	Home page displayed
Test Load Balancer via SSL	https://myBIForms.mycompany.com/	Home page displayed

Forms	https://myBIForms.mycompany.com/forms/frmservlet	Test Form is displayed
Discoverer	https://myBIForms.mycompany.com/discoverer/viewer	Discoverer Viewer Home Page displayed
Reports Queue	https://myBIForms.mycompany.com/reports/rwservlet/showjobs	Job Queue is displayed.

Install and Configure the Web Tier

Follow these steps to install the Oracle Web Tier on to 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`

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 all components. Click **Next**.

	Note: If this installation is going to be associated with a WebLogic Domain then select the box “Associate Selected Components with WebLogic Domain” If this installation is not going to be associated with a WebLogic domain then deselct this box. If required this association can be done post-installation.

Specify WebLogic Domain Details
(Optional)
	Specify the following values:
	Domain Host Name (Machine Hosting WebLogic Admin Server) For example
	`wladmin.mycompany.com`
	Domain port Number (WebLogic Administration server Port) e.g.
	`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: webcache1 Click **Next**.
WebCache Administrator Password	Specify a value for the Web Cache administrator password. Confirm the password and click **Next**

Specify Web Tier port Details	In HA Implementations whilst not mandatory it makes life simpler if all of the ports used by the various components are synchronised 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 Oracle Configuration Manager Details	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://webhost1:7777/

Configure Virtual Hosts

In order for Oracle Forms, Reports and Discoverer to work with the load balancer two virtual hosts need to be defined, the file virtual_hosts.conf can be copied from apphost1.mycompany.com

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://myBIForms.mycompany.com:443
    RewriteEngine On
    RewriteOptions inherit
    UseCanonicalName On
</VirtualHost>

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

Save the file

Copy Forms Specific Files from APPHOST1

The Web Tier needs certain files such as images and configuration information to be able to display the Forms pages correctly. All of these files exist either in the ORACLE_HOME or ORACLE_INSTANCE of the Forms installation.

Copy the entire contents of the directory

ORACLE_INSTANCE/config/OHS/ohs1/moduleconf to the equivalent directory on WEBHOST1 (ORACLE_INSTANCE/config/OHS/http1/moduleconf).

Copy Reports Specific Files from APPHOST1

The Web Tier needs certain files such as images and configuration information to be able to display the Reports pages correctly. All of these files exist either in the ORACLE_HOME or ORACLE_INSTANCE of the Reports installation.

Copy the entire contents of the directory

ORACLE_INSTANCE/config/OHS/ohs1/moduleconf to the equivalent directory on WEBHOST1 (ORACLE_INSTANCE/config/OHS/http1/moduleconf).

Copy Discoverer Specific Files from APPHOST1

The Web Tier needs certain files such as images and configuration information to be able to display the Discoverer pages correctly. All of these files exist either in the ORACLE_HOME or ORACLE_INSTANCE of the Discoverer installation.

Copy the entire contents of the directory

ORACLE_INSTANCE/config/OHS/ohs1/moduleconf to the equivalent directory on WEBHOST2 (ORACLE_INSTANCE/config/OHS/http1/moduleconf).

Copy SSO configuration File

Copy osso.conf from ORACLE_INSTANCE/config/OHS/ohs1 on APPHOST1 to the equivalent directory on WEBHOST1 (ORACLE_INSTANCE/config/OHS/http1).

Configure Web Cache

Log into Enterprise Manager Administration Console

Log into 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.

Default User Name and Password are administrator/administrator

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 Interval	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.

myBIForms.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 Caches

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.

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 WEBHOST1.

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

Click **Apply** to apply the new configuration.

Restart Oracle Web Tier

Shutdown the Web Tier components on APPHOST1,APPHOST2 and WEBHOST1 by issuing the following command on each server:

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.

Validate Configuration

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

Test	URL	Result
Test Load balancer	http://myBIForms.mycompany.com/	Home page displayed
Test Load Balancer via SSL	https://myBIForms.mycompany.com/	Home page displayed
Forms	https://myBIForms.mycompany.com/forms/frmservlet	Test Form is displayed

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`

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 all components. Click **Next**. Note: If this installation is going to be associated with a WebLogic Domain then select the box "Associate Selected Components with WebLogic Domain" If this installation is not going to be associated with a WebLogic domain then deselect this box. If required this association can be done post installation.
Specify WebLogic Domain Details (Optional)	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: webcache2 Click **Next**.
WebCache Administrator Password	Specify a value for the Web Cache administrator password. Confirm the password and click **Next**
Specify Web Tier port Details	Select Specify Ports using Configuration File Select the same file used for WEBHOST1 and click **Next**
Specify Oracle Configuration Manager Details	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://webhost1:7777/

Configure Virtual Hosts

In order for Forms to work with the load balancer completely two virtual hosts need defined.

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://myBIForms.mycompany.com:443
    RewriteEngine On
    RewriteOptions inherit
    UseCanonicalName On
</VirtualHost>

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

Save the file

Copy Forms Specific Files from APPHOST1

The Web Tier needs certain files such as images and configuration information to be able to display the Forms pages correctly. All of these files exist either in the ORACLE_HOME or ORACLE_INSTANCE of the Forms installation.

Copy the entire contents of the directory ORACLE_INSTANCE/config/OHS/ohs1/moduleconf to the equivalent directory on WEBHOST2 (ORACLE_INSTANCE/config/OHS/http2/moduleconf)

Copy Reports Specific Files from APPHOST1

The Web Tier needs certain files such as images and configuration information to be able to display the Reports pages correctly. All of these files exist either in the ORACLE_HOME or ORACLE_INSTANCE of the Reports installation.

Copy the entire contents of the directory ORACLE_INSTANCE/config/OHS/ohs1/moduleconf to the equivalent directory on WEBHOST1 (ORACLE_INSTANCE/config/OHS/http2/moduleconf).

Copy Discoverer Specific Files from APPHOST1

The Web Tier needs certain files such as images and configuration information to be able to display the Discoverer pages correctly. All of these files exist either in the ORACLE_HOME or ORACLE_INSTANCE of the Discoverer installation.

Copy the entire contents of the directory ORACLE_INSTANCE/config/OHS/ohs1/moduleconf to the equivalent directory on WEBHOST2 (ORACLE_INSTANCE/config/OHS/http2/moduleconf)

Copy SSO Configuration File

Copy osso.conf from ORACLE_INSTANCE/config/OHS/ohs1 on APPHOST1 to the equivalent directory on WEBHOST1 i.e. ORACLE_INSTANCE/config/OHS/http1

Configure Web Cache

Log into Enterprise Manager Administration Console

Log in to 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.

Default User Name and Password are administrator/administrator

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	WEBHOST2.mycompany.com
Port	7778
Capacity	100
Protocol	HTTP
Failover Threshold	5
Ping URL	/
Ping Interval	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.

myBIForms.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 Caches

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 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 WEBHOST1.

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

Click **Apply** to apply the new configuration.

Restart Oracle HTTP Server

Shutdown the Web Tier components on APPHOST1, APPHOST2, WEBHOST1 and WEBHOST2 by issuing the following command on each server

ORACLE_HOME/opmn/bin/opmnctl stopall

Restart the Web Tier components on WEBHOST1 and WEBHOST2 by issuing the following command on each server :

ORACLE_HOME/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.

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://myBIForms.mycompany.com/	Home page displayed
Test Load Balancer via SSL	https://myBIForms.mycompany.com/	Home page displayed

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 webcache1

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

myBIForms.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 webcache1

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 webcache1

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 webcache2 (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 from APPHOST1 and APPHOST2

APPHOST1

Before starting this operation ensure that ORACLE_HOME and ORACLE_INSTANCE are set appropriately for this host i.e.

ORACLE_HOME=/u01/app/oracle/product/fmw/BIForms

ORACLE_INSTANCE=/u01/app/oracle/admin/BIForms1

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 i.e.

ORACLE_HOME=/u01/app/oracle/product/fmw/BIForms

ORACLE_INSTANCE=/u01/app/oracle/admin/BIForms2

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.

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: 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.

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 both 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 Section 0 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: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
   ```

1. 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
   ```

2. Change directory to the user-defined directory.

   ```
   APPHOST2> cd certs
   ```

3. Run the utils.CertGen tool from the user-defined directory to create the certificates for both 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 both 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 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
```

Failover of the Preference Server

In the event of the server running the preference server failing over, the server needs to be started on one of the surviving nodes. This is accomplished by:

Identify Preference Store Host Name and Port

The nominated preference server in this example will be configured to run on APPHOST2.

When installed onto the server the Preference server will have been assigned a port. This value needs to be located before proceeding to the next steps.

On APPHOST2 open the file opmn.xml which is located in:

ORACLE_INSTANCE/config/OPMN/opmn

Search the file for the entry PREFERENCE_PORT. The value= shows the port assigned to the preference server.

Enable the Preference Store on APPHOST2

Now that APPHOST2 is using the preference store on APPHOST1, it should be disabled on APPHOST2.

This is achieved by editing the file opmn.xml which is located in the directory ORACLE_INSTANCE/config/OPMN/opmn

Locate the following line:

```
<process-type id="PreferenceServer" module-id="Disco_PreferenceServer" working-
dir="$DC_LOG_DIR" status="disabled">
```

Change status=disabled to status=enabled
For example

```
<process-type id="PreferenceServer" module-id="Disco_PreferenceServer" working-
dir="$DC_LOG_DIR" status="enabled">
```

Start the Preference Server on APPHOST2

Start the preference server on APPHOST2 using the command:

opmnctl startproc process-type=PreferenceServer

Change Any Surviving Servers to use new Preference Store

Now that the preference server has been started on APPHOST2 all discoverer instances need to be amended to use this preference server including APPHOST2.

The preference store being used is determined at the startup of the WebLogic Managed server WLS_DISCO1. In order to get WLS_DISCO1 to use a different preference store the startup parameters need changing.

This is achieved by logging into the WebLogic administration console using the following URL

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

Select Environment -> Servers from the Domain Structure menu.

Click on Lock and Edit in the Change Center window.

Click on the server WLS_DISCO1

Click on the Configuration Tab and the Server Start Sub tab.

In the arguments field append the following:

```
-Doracle.disco.activation.preferencePort=<portno>
-Doracle.disco.activation.preferenceHost=<hostname>
```

Where port number is the value of PREFERENCE_PORT above.

Where hostname is the name of the host on which the preference server is started ie. APPHOST2

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 scaled out.

Best Practice

Oracle Fusion Middleware 11g does not replicate configuration information automatically between nodes. It is important therefore that any changes to any of the following be manually propagated to the other servers in the deployment:

ORACLE_INSTANCE

DOMAIN_HOME/servers/WLS_FORMS/stage/formsapp/<version>/formsapp/config

DOM_HOME/user_projects/domains/ReportsDomain/servers/WLS_REPORTS1/stage/reports/reports/configuration

DOMAIN_HOME/servers/WLS_DISCO1/stage/discoverer/<version>/discoverer/configuration/configuration.xml

Preference Server

The preference server contains details about user preferences, which can be updated on a frequent basis. In the event of the failure of the Discoverer preference server, the preference server will need to be started on another server. Therefore the preference server information needs to be copied to these potential hosts on a regular basis.

To this end the following file needs to be propagated to other servers which could host the preference server on a regular basis:

ORACLE_INSTANCE/config/PreferenceServer/Discoverer_*Instance*/.reg_key.dc

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