Best Practices for Oracle WebCenter Custom Portal Apps in an Enterprise Topology

Oracle Maximum Availability Architecture White Paper
September 2012

Maximum Availability Architecture

Oracle Best Practices For High Availability

ORACLE

Introduction

Oracle WebCenter Portal is a popular platform for rich applications that can easily incorporate tools such as Discussion Forums, Blogs, Wikis and Portlets. Included with Oracle WebCenter Portal is a built-in application, Oracle WebCenter Spaces, which provides a ready-to-go community platform.

In an Enterprise environment, WebCenter Portal applications should be configured to be both Highly Available themselves as well as configured with external services that are also robust and available. For the Oracle WebCenter Portal Suite and for Oracle WebCenter Spaces, configuration guidelines are provided in the *Enterprise Deployment Guide for Oracle WebCenter Portal.* For customers, however, who are deploying their own custom Portal applications, there is still some additional configuration required in order for their applications to be successfuly deployed in this environment.

This paper provides the guidelines for preparing an Enterprise topology suitable for deploying custom Portal applications as well as guidelines on configuring the application for availability and best practices in the deployment and ongoing maintenance of the application.

The responsibility for the preparation of the application itself and the responsibility for preparing the production environment may fall to separate organizations. Therefore it is understood that this paper may have two audiences. The section on ‘Setting up the Enterprise Toplogy’ is meant as a complement to information already in the *Enterprise*

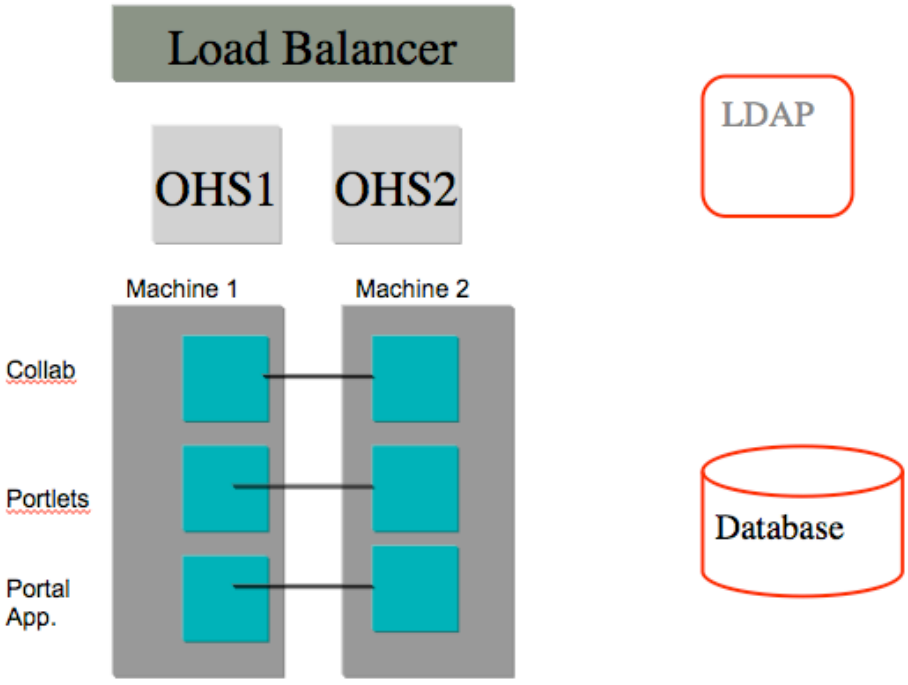
Deployment Guide for Oracle WebCenter Portal. Architects can use this information in planning their production environments.

The section on 'Preparing and Deploying the Application' can be used by Developers to ensure that the application is ready to be moved over to a production environment.

Setting up the Enterprise Topology

An Enterprise topology is one that is available, reliable, and performant. It also includes support for a production database, identity management scheme, and load balancing mechanisms.

The topology into which we will be deploying a Custom Portal application is shown below. At minimum, it includes two separate machines for running Oracle WebCenter Portal, an external identity management database, an external Database, and front-ended by Oracle HTTP Servers which, in turn, are front-ended by a Load Balancer.



The rest of this section covers the steps necessary to create this topology, including creating the Servers where the Custom Portal application will be deployed and provisioning those servers with the necessary external applications.

Setting up the Initial Enterprise Environment

This paper does not go into the full set of steps for deploying Oracle WebCenter Portal Enterprise architecture. Many of the necessary steps are covered well by the *Enterprise Deployment*

Guide for Oracle WebCenter Portal (EDG). Here, we'll outline the required steps as well as pointing out what the minimum recommendations are for customers who are only interested in running Oracle Webcenter Portal Custom applications with as few external requirements as possible.

Preparing the Environment (Network, File System)

The Network and File system should be prepared, if necessary, in accordance with the guidelines provided in the EDG. This includes preparing the directory structures on the filesystem to support a clustered environment.

If a load balancer will be used, then the load balancer should be configured as stated in Chapter 3 of the EDG. No Virtual IPs are required for WebCenter Portal Custom applications.

In order to support AdminServer failover, a shared disk is required to hold the Administration Server Domain Home. The configuration of the shared disk is outline in Chapter 4 of the EDG.

Preparing the Database

The Database should be installed and have its Services configured. This is assumed to be an Oracle RAC Database. Connections to this database can be multi-data sources but we recommend Active Gridlink for even greater Avalability. The steps for configuring Active GridLink are provided later in this paper.

At this point, The Oracle Fusion Metadata Repository should be loaded into the Database in accordance with the EDG.

The full set of schemas would include those for all of the Oracle WebCenter Portal applications, as outlined in the *Enterprise Deployment Guide*. The required schemas depend on which other WebCenter Portal Suite applications are required. At the very least the schemas `WebCenterDS` and `ActivitiesDS` should be installed.

At this point, it is not necessary to create the schemas for the application (including any custom application schemas and MDS) They will be created later. Instructions are provided in the next section for creating these additional schemas.

Installing the Software

The WebLogic and the Oracle WebCenter Portal software should be installed. This can be a separate, identical installation for each machine or one installation on shared storage that is shared by multiple machines.

Installing a Web Tier

If a Web Tier is not already present, then Oracle HTTP Server should be installed to act as an HTTP gateway to the Oracle WebCenter Portal Suite applications. For now, it is sufficient to install and start the HTTP Server. Later, we'll configure routing.

Creating a WebCenter Portal Domain

The domain should be created in accordance with the guidelines set out in the EDG. This means configuring a cluster of managed servers across all the machines and configuring the datasources to point to the schemas created earlier.

The required servers again depend on which servers will be required by the Custom Portal Application. This may include the Portlet servers and the Collaboration servers, for example. Or none of these may be required. If none are required, then it is not necessary to configure the domain yet.

The Managed Servers that will run the Custom Portal application should not be created yet. They will be created in the next set of steps.

Creating a Custom WebCenter Portal Cluster

At this point, we create the required schemas and then create a cluster of managed servers that will run the Custom Portal application.

Create the necessary datasources

It is recommended that a Custom application use its own Metadata Service Repository (MDS) and not share existing repositories, especially those being used by Oracle WebCenter Portal Spaces.

To create a new MDS for the application, launch the Repository Creation Utility (RCU) and follow these steps:

1) Select **Create** to create a new schema and click **Next**.
2) Enter the Database information on the **Database Connection Details** page and click **Next.**
3) On the **Select Components** page, enter a prefix to use for your MDS schema and select **Metadata Services** under **AS Common Schemas**. Nothing else should be selected. Click **Next**.
4) On the **Schema Passwords** page, enter a password for your schema and click **Next.**
5) Click **Nex**t on the **Map Tablespaces** page.
6) Click **OK** to create the Tablespace.
7) Click **Create** to create the schema.

Any other schemas required by the application should also be created at this time.

Create the Custom Portal Cluster

The Custom Portal cluster is created by extending the existing Domain to add new managed servers that have the correct set of JRF libraries applied. This is accomplished by using a preconfigured WebLogic template. (If this is a new Domain, then the **Create** option should be used for the Domain and then the template should be selected.)

Run the Configuration Wizard and choose to **Extend an Existing WebLogic Domain**. Then, choose the WebCenter Portal domain.

1) On the **Select Extension Source** screen, choose to **Extend my domain using an existing extension template**

2) Find and choose the template *FMW_HOME/Oracle_WC1/common/templates/applications/oracle.wc_custom_portal_template_11.1.1.jar*

3) In the **Configure JDBC Data Sources** screen, you should see the following three new datasources:

   - mds-CustomPortalDS – This should be configured to point to the MDS schemas created earlier for the application.
   - WebCenter-CustomPortalDS – This should be configured to point to the existing WebCenter schema
   - Activities-CustomPortalDS – This should be configured to point to the existing Activities schema

4) For RAC configuration you can select **Convert to GridLink** and click **Next** (if not using GridLink, the next step can be skipped)

5) In the **Configure GridLink RAC Component Schema** screen:

   - Select a GridLink data source schema: mds-CustomPortalDS schema
   - Follow the instructions in Appendix A: Creating GridLink datasources to create the datasources.
   - Repeat for the other schemas

6) In the **Test JDBC Data Sources** screen, select the newly created datasources and then click **Test Connections**. Click **Next** when all the connections are successful.

7) In the **Select Optional Configuration** screen, select:

   - Managed Servers, Clusters and Machines
   - Deployments and Services

8) In the **Configure Managed Servers** screen, add two new Managed Servers:

Name	Server	Listen Port	SSL Port	SSL Enabled
WC_CustomPortal1	Host1	9010	n/a	No
WC_CustomPortal2	Host2	9010	n/a	No

9) In the **Configure Clusters** screen, add a new Cluster for the new Managed Servers:

Name	Messaging Mode	Multicast Address	Multicast Port	Cluster Address
CustomPortal_Cluster	Unicast	n/a	n/a	Leave empty

10) In the **Assign Servers to Clusters** screen, assign the newly created servers to the cluster.

11) In the **Configure Machines** screen, add Machines if required.

12) In the **Assign Servers to Machines** screen, place each of the Custom Portal Managed Servers.

13) Click **Next** on the next screens until the Configuration Summary screen is reached.

14) On the **Configuration Summary** screen, click **Extend** to extend the Domain.

Create directories for the Application Deployment

The application will be later deployed as an EAR file. This can either be one file accessible on shared storage by all members of the cluster, or it can be copies of the same file distributed across the domain.

In either case, a directory with the same path on each machine should be created in order to hold the application artifacts. Later, we'll deploy the application into this directory.

Preparing and Deploying the Application

The application needs to be prepared before it can be deployed into the Enterprise environment. This includes configuring datasources and security -which must be done before deploying into any environment- as well as configuring the application to run in a clustered environment. Afterwards, the application can be deployed and then provisioned.

Preparing the Application

Prior to deployment, several steps need to be performed to prepare the application. This is covered in more detail in the *Oracle Fusion Middleware Developer's Guide for Oracle WebCenter*. Here we briefly cover those steps, pointing out the relevant decisions to be made when deploying into an EDG environment.

Packaging the Application Datasources

First, the application should use JDBC Data Sources – such as those already configured in the Oracle WebLogic Server in previous steps – and not JDBC URLs. These are also referred to as Global data sources

To configure a JDBC Datasource, create a new application module configuration in JDeveloper and choose to connect to a JDBC Datasource. The name of the datasource should be of the form:

```
java:comp/env/jdbc/MyDataSourceDS
```

The name used, MyDataSourceDS will match the name of the datasource that was created on the Oracle WebLogic Server.

Packaging Application Identity and Security

Generally, application credentials for access to external applications are not migrated from the application to the server. When deploying to production environments, this migration is not allowed. Any external connections should use connections (and connection credentials) managed by Fusion Middleware Control.

Likewise, identity should use users and groups already defined in the external LDAP. Note that an external LDAP is required for distributed environments.

For the WebCenter Portal application, in JDeveloper:

1) Open the Application Properties Dialog
2) In the **Security Deployment Options** section:
3) Uncheck the **Credentials** checkbox
4) Uncheck the **Users and Groups** checkbox

Migrating Security Policies

If there are application policies that need to be migrated, these should be merged in with the existing application policies. This can be accomplished with settings in the weblogic-application.xml file:

```
<wls:application-param>     <wls:param-
name>jps.policystore.migration</wls:param-name>
<wls:param-value>MERGE</wls:param-value> </wls:application-
param>

  <wls:application-param>     <wls:param-
name>jps.policystore.removal</wls:param-name>     <wls:param-
value>OFF</wls:param-value> </wls:application-param>

 <wls:listener>

<wls:listener-
class>oracle.security.jps.wls.listeners.JpsApplicationLifecycl
eListener</wls:listener-class> </wls:listener>
```

Configuring the Application for Clustering and Replication

Session state replication will occur automatically once the application is deployed to a cluster, as long as the application is configured correctly.

Ensure that the adf-config.xml file contains the following entry in order to set the scope correctly for adf objects:

```
<adfc:adf-controller-config><adfc:adf-scope-ha-
support>true</adfc:adf-scope-ha-support></adfc:adf-controller-
config>
```

The file can be found in the **Application Resources** tab -> **ADF META-INF** ->adf-config.xml

And in order to set the replication policy, the weblogic-application.xml file should also be configured with the following: (in the same location):

```
<session-descriptor><persistent-store-
type>replicated_if_clustered</persistent-store-type></session-
descriptor>
```

This will enable replication when the application is in a clustered environment but disable it when the application is deployed to a single server

Configuring the Context Root

By default, the context root is set using the Project Name and Application Name. This creates a long context root. Change the default in Project Properties. The Context root will be used in routing to this application.

Deploying the Application

Deployment involves creating an EAR file which is then moved to the server and deployed using Fusion Middleware Console.

Generating an EAR file

For WebCenter Portal applications, the EAR and WAR deployment profiles should already be created.

To create the EAR file in JDeveloper:

1. Create a new **Deployment Profile**, choose **EAR File**, and then click **OK**.
2. Enter a name for the deployment profile and click **OK**.
3. In the **Edit EAR Deployment Profile Properties** screen, choose **Application Assembly**, then choose the WAR associated with the project then click **OK**.
4. In the **Application Properties** screen, ensure that **Auto Generate and Synchronize weblogic-jdbc.xml Descriptors During Deployment** is not selected. Click OK to continue.
5. Choose **Deploy** after right-clicking the Application name and then choose to deploy **to EAR file**.

Deploying the EAR file to the WebLogic Cluster

After generating the EAR file, the file should be placed in a directory that is accessible to all machines in the cluster. This can be on a filesystem that is mounted remotely by all servers.

For example, if all machines are mounting `/u02/admin`, then the EAR file can be placed in `/u02/admin/deployments/MyApp.ear` and all servers would access the deployment files at this path.

In the case where shared storage is not available, then the EAR file should be manually copied to all machines. On each machine, the EAR file should be placed at the exact same directory location.

To deploy this file to the cluster:

1) Logon to the Administration Console for the Domain (at http://host:7001/console)
2) Select **Deployments** from the left hand pane

3) Choose to **Install** a new application and locate the EAR file
4) Choose **Install this deployment as an application**
5) On the **Select Deployment Targets** screen, select the Custom Portal Cluster.
6) On the **Optional Settings** page, select **I will make the deployment accessible from the following location**
7) Choose **Finish** to complete the deployment

After this, all servers in the Custom Portal Cluster should pick up the deployment files and deploy the application.

Provisioning the Application

This includes all resources needed by the new Cluster. Configuring a Gridlink DS, connections to Identity and Policy store. Optionally other connections.

Configure Routing

Although the application can be accessed directly using the Context root at any of the servers in the cluster, adding an HTTP server to route to the cluster will improve availability. Routing through an HTTP server means that the application is available as long as at least one server in the cluster is available. Session replication, which should occur automatically, also ensures that in the event of a server failure, user session information is maintained.

To enable access from Oracle HTTP server, first ensure that the weblogic module is installed. Then add the entry for the application to the mod_weblogic.conf file as follows:

```
<Location /MyApp>
     WebLogicCluster HOST1:9010,HOST2:9010
     SetHandler weblogic-handler
</Location>
```

where the host and ports are those where the application is accessed.

Verifying the Identity store connection

After accessing the application either directly, or through the HTTP Server, attempt to logon using the users which should already be seeded in the LDAP directory.

If the users are not available, or have insufficient privileges, they will need to be created in LDAP.

Register Out of the Box Portlet Producers

If the application will be using the Portlet producers, these can be configured rapidly using the WebLogic Scripting Tool (WLST):

1. Start the WebLogic Scripting Tool from *MW_HOME*/oracle_common/common/bin/wlst.sh
2. In WLST, connect as the administrator.

   For example: connect("weblogic","*admin password*","ADMINHOST:7001",server="WC_CustomPortal1")
3. Register all three out-of-the-box WSRP and PDK-Java producers.

   For example:

   registerOOTBProducers(producerHost=LBRHOST',producerPort=80, appName='MyApp', server='WC_CustomPortal1')

Where MyApp is the name of the application. Here LBR:7777 is the address of where the Portlet producers can be found. In an EDG configuration, this will be an address that directs to a cluster of Portlet producers.

Creating other connections

Other connections need to be created for the application. This includes connections to Content Repositories, Worklist services, and other external applications.

For reference, see the chapter *Getting Framework Applications Up and Running* in the *Oracle Fusion Middleware Administrators Guide for Oracle WebCenter Portal.*

For Content Repositories, the Active Connection should be set using Fusion Middleware Control.

If the application contains Worklist task flows, then a connection to a BPEL Server should be established.

For Announcements and Discussions and for Portlet providers, these connections should also be configured for the application.

Note that for the above connections the address (hostname and port) should point to the location that provides the greatest availability. In an Enterprise environment this will be the load balancer.

There are no additional steps required for configuring a clustered application than for configuring a single server. The configuration information is stored in the Metadata Repository and this is shared by all members of the cluster.

Other migration tasks

If the application already has data that needs to be moved into a production environment, then the *Oracle Fusion Middleware Developer's Guide for Oracle WebCenter outlines* the necessary steps and scripts for transferring data, including Content and Spaces data, into the new environment. Specifically, consult the section on "Moving a WebCenter Portal to an existing Environment" in the section on "*Understanding the Portal LifeCycle*."

Maintaining Applications

Once applications have been successfully deployed, they begin their lifecycle. This lifecycle includes being available as part of a cluster, sharing session state with each other, and storing all their state in a manner that makes the application easier to maintain.

Managing Application State/ Redeployment

For Custom Portal applications, state is stored in MDS. This includes user customizations, all application configuration, resource and portlet customizations.

Connection information to external resources is also stored in MDS. This same MDS partition is shared by all members of the application cluster.

After the application has been deployed, all further customizations will be written to MDS. Customizations use the base documents as reference and then create all further customizations in the database. This includes deployment files such as adf-config.xml and connections.xml

Managing Customizations

Application state during deployment and re-deployment should follow this basic workflow:

1) Application is first deployed. Configuration files are written into MDS. These are used as a base layer.
2) Users begin using application. Customizations and personalizations are stored directly into MDS as an additional layer.
3) Application is redeployed. The base MDS is overwritten by new configuration files and customizations are then applied on top of this.

Managing redeployments

During deployment and redeployment to a cluster, the base MDS is only overwritten once when the first member of the cluster deploys the application.

Note that during deployments and redeployments to a cluster, all servers should be up and running.

Redeployment of a versioned application can take place and is the easiest way to update an application with minor changes, such as might occur in a patch.

A cluster re-deployment proceeds as follows:

1) Logon to FMW Control
2) Select the Cluster to which to redeploy the application and select Application Deployment – Redeploy.
3) Select the Application from the Select Application page
4) Select the Archive for redeployment
5) Specify the context root for the application
6) Specify the same MDS repository and partition for redeployment
7) Review the deployment settings. Do not configure ADF Connections as these have already been set.
8) Redeploy the application
9) The first server writes the MAR to MDS
10) The second server compares the checksum of its MAR to the MAR already in MDS and determines that no MAR deployment is required.

This will automatically redeploy the application to each member of the cluster.

Application failover

When running in a cluster environment, the application is able to provide continuous service to a user even if the managed server to which the user is connected should fail. This includes maintaining the users session on the server. The process happens as follows:

1) User is connected to Managed Server 1. Session state is being replicated to Managed Server 2.
2) Managed Server 1 fails
3) On the next request, the user is automatically directed by Oracle HTTP Server (using the weblogic plug-in) to the secondary server, in this case Managed Server 2.
4) User is connected to Managed Server 2 and is able to continue working. Managed Server 2 becomes the new primary. Session state is replicated to another server – Managed Server 3.

Note that Session state includes user login state and user navigational state.

In a failover, there may be a loss of any unsaved state. For example, if a user has a window open and is filling in a form, but has not yet saved or completed their text, this text may be lost in a failover. In all cases, however, all **saved** state will be available to the failed over user.

Application Scale-Out

Expanding the cluster means adding additional servers to the cluster in order to provide greater performance and greater availability.

To add additional servers, more servers should be created within the Application cluster. This can be accomplished by cloning one of the existing managed servers:

1) From the **Environment-Servers** menu, select one of the existing managed servers
2) Choose to **Clone** the server
3) Provide a name for the new Server. Also, provide the **Listen Address** and **Port** of this new Server.

The server will be created and should now appear as a new member of the cluster. All cluster applications and libraries should already be targeted to this server. Start the Server in order to verify that the application is running.

Optionally, add the new server's host and port information to the WebLogicCluster directive of the HTTP Server. Since this list is built dynamically, routing will happen to the new server even without making this change. However, adding the new server provides greater availability for the initial discovery of the Cluster.

Application Patches and Upgrades

If the application has changed and needs to be deployed again to the cluster, in most cases it is sufficient to redeploy the application. This preserves much of the application configuration such as security and connection data.

Redeployment can take place while the application is running. This will destroy existing session state, however, and there will be user disruption although not an extended downtime. Servers will need to be restarted. However, they can be restarted in a rolling manner in order to maintain uptime.

Appendix A: Creating GridLink Datasources

A single data source implementation has been introduced to support an Oracle RAC cluster. It responds to FAN events to provide Fast Connection Failover (FCF), Runtime Connection Load-Balancing (RCLB), and RAC instance graceful shutdown. The new feature is called WebLogic Active GridLink for RAC; which is implemented as the GridLink data source within WebLogic Server.

In the Configure GridLink RAC Component Schema screen:

1. Select first GridLink data source schema, **WebCenterDS Schema**.
2. Enter values for the following fields, specifying the connect information for the GridLink Oracle RAC database that was seeded through RCU.

    - **Driver:** Select **Oracle driver (Thin) for GridLinkConnections,Versions:10 and later**.
    - **Service Name:** Enter the service name of the Oracle RAC database in lowercase letters, followed by the domain name. For example,wcpedg.mycompany.com.
    - **Username:** Enter the complete user name (including the prefix) for the database schema owner. The user names shown in Table 10-2 assume that wcpedg was used as the prefix for schema creation through RCU.
    - **Password:** Enter the password for the database schema owner.
    - **Enable FAN:** Select this option.
    - **Enable SSL:** Deselect this option.

        If you select SSL, to enable Oracle Notification Service (ONS) notification encryption, provide appropriate **Wallet File** and **Wallet Password** details.
    - **Service Listener:** Enter the Oracle Single Client Access Name (SCAN) address and port for the Oracle RAC database being used. The protocol should be TCP.

        Oracle recommends that you use SCAN addresses to specify the Service Listener (and OSN Host) so you do not need to update a GridLink data source containing SCAN addresses if you add or

remove Oracle RAC nodes. To determine the SCAN address, query theremote_listener parameter in the database:

```
SQL>show parameter remote_listener;
```

Note:

For database versions that do not support SCAN:

For Oracle Database 11*g* Release 1 (11.1), enter the Virtual IP and port of each database's instance listener, for example:

custdbhost1-vip.mycompany.com (Port 1521)

and

custdbhost2-vip.mycompany.com (Port 1521)

For Oracle Database 10*g*, use multi data sources to connect to an Oracle RAC database. .

- **ONS Host:** Enter the SCAN address for the Oracle RAC database and the ONS remote port, as reported by the database:

  ```
  [orcl@db-scan1 ~]$ srvctl config nodeapps -s
  ONS exists: Local port 6100, remote port
  6200, EM port 2016
  ```

  For Oracle Database 11g Release 1 (11.1), enter the host name and port for the database's ONS service. For example:

  custdbhost1.mycompany.com (Port 6200)

  and

  custdbhost2.mycompany.com (Port 6200)

3. Select the next schema, for example **ActivitiesDS Schema**, and specify appropriate details.

Ensure that GridLink information is entered for all WebCenter Portal schemas.

References

1. *Oracle Fusion Middleware Enterprise Deployment Guide for Oracle WebCenter Portal*
   http://docs.oracle.com/cd/E23943_01/core.1111/e12037/toc.htm
2. *Oracle Fusion Middleware Administrators Guide for Oracle WebCenter Portal*

   http://docs.oracle.com/cd/E23943_01/webcenter.1111/e12405/toc.htm
3. *Oracle Fusion Middleware Developers Guide for Oracle WebCenter Portal*
   http://docs.oracle.com/cd/E23943_01/webcenter.1111/e10148/toc.htm

Oracle White Paper Title:
September 2012
Author: Richard Delval
Contributing Authors: Jeni Ferns

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

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

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

0109