

# JD Edwards EnterpriseOne 9.1.2.x Configuration with Oracle HTTP Server and a DMZ

*An Oracle JD Edwards EnterpriseOne Red Paper*

*July 2012*

**ORACLE**  
JD EDWARDS ENTERPRISEONE

## **PURPOSE STATEMENT AND DISCLAIMER**

This document is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle. Updates and enhancements are provided in accordance with Oracle's Technical Support Policies at:

[www.oracle.com/support/collateral/oracle-technical-support-policies.pdf](http://www.oracle.com/support/collateral/oracle-technical-support-policies.pdf)

This document may provide considerations when reviewing system performance relative to your JD Edwards EnterpriseOne system. Many factors can influence performance and your results may differ depending on many different variables.

# JD Edwards EnterpriseOne 9.1.2.x Configuration with Oracle HTTP Server and a DMZ

Overview .....	1
Installing Oracle HTTP Server .....	1
Basic configuration .....	16
Environment Setup .....	16
Configuring the OHS Proxy for WebLogic Server .....	16
Configuring SSL .....	18
Inbound SSL to Oracle HTTP Server .....	18
Inbound SSL to Oracle WebLogic Server .....	20
Outbound SSL from Oracle HTTP Server .....	22
Creating an EnterpriseOne Cluster .....	26
Clustering the HTTP Server .....	27
Understanding The DMZ Chapter .....	33
Illustrating the Architecture .....	33
Depicting the Physical Setup .....	34
Configuring the Policies .....	35
Explaining Mapped IP Addresses .....	36
Explaining Open Ports .....	37
Configuring the Host File .....	38
Tracing out Ports .....	38
Appendix A: Configuring Additional Technologies with a DMZ .....	39
Setting Up Mobile Applications .....	39
Setting Up Business Services Server .....	39
Setting Up Transaction Server .....	40
Appendix B: Installing Fusion Middleware Control EM .....	42
Appendix C: Setting up Single Sign-on (SSO) .....	52

# JD Edwards EnterpriseOne 9.1.2.x Configuration with Oracle HTTP Server and a DMZ

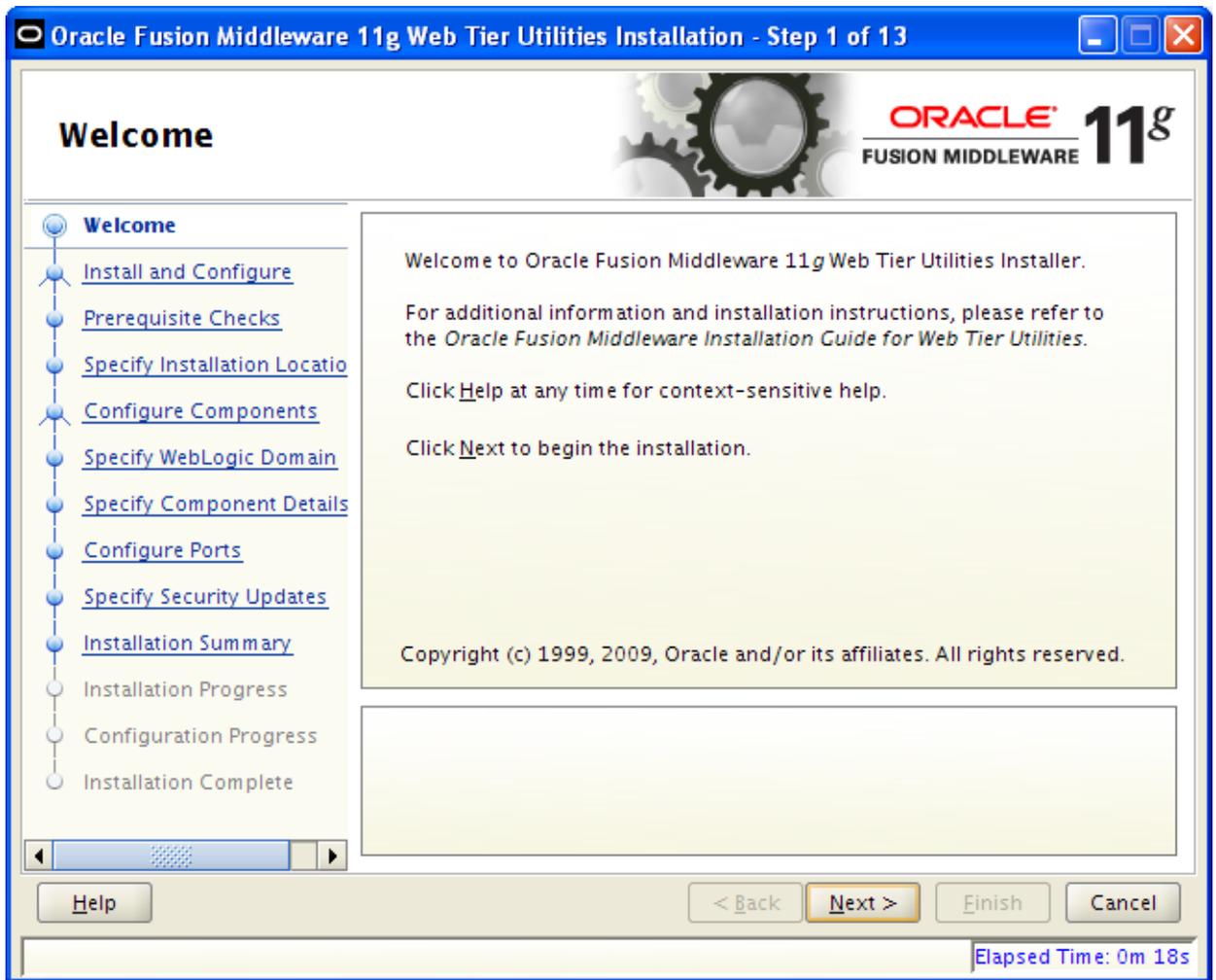
## Overview

The purpose of this document is to provide information about configuring an Oracle HTTP Server to be used with EnterpriseOne software running with WebLogic. Although an Oracle HTTP Server is not necessary in order to run the EnterpriseOne HTML client with WebLogic Server, there may be reasons that one would want to configure the software to make use of a separate HTTP server.

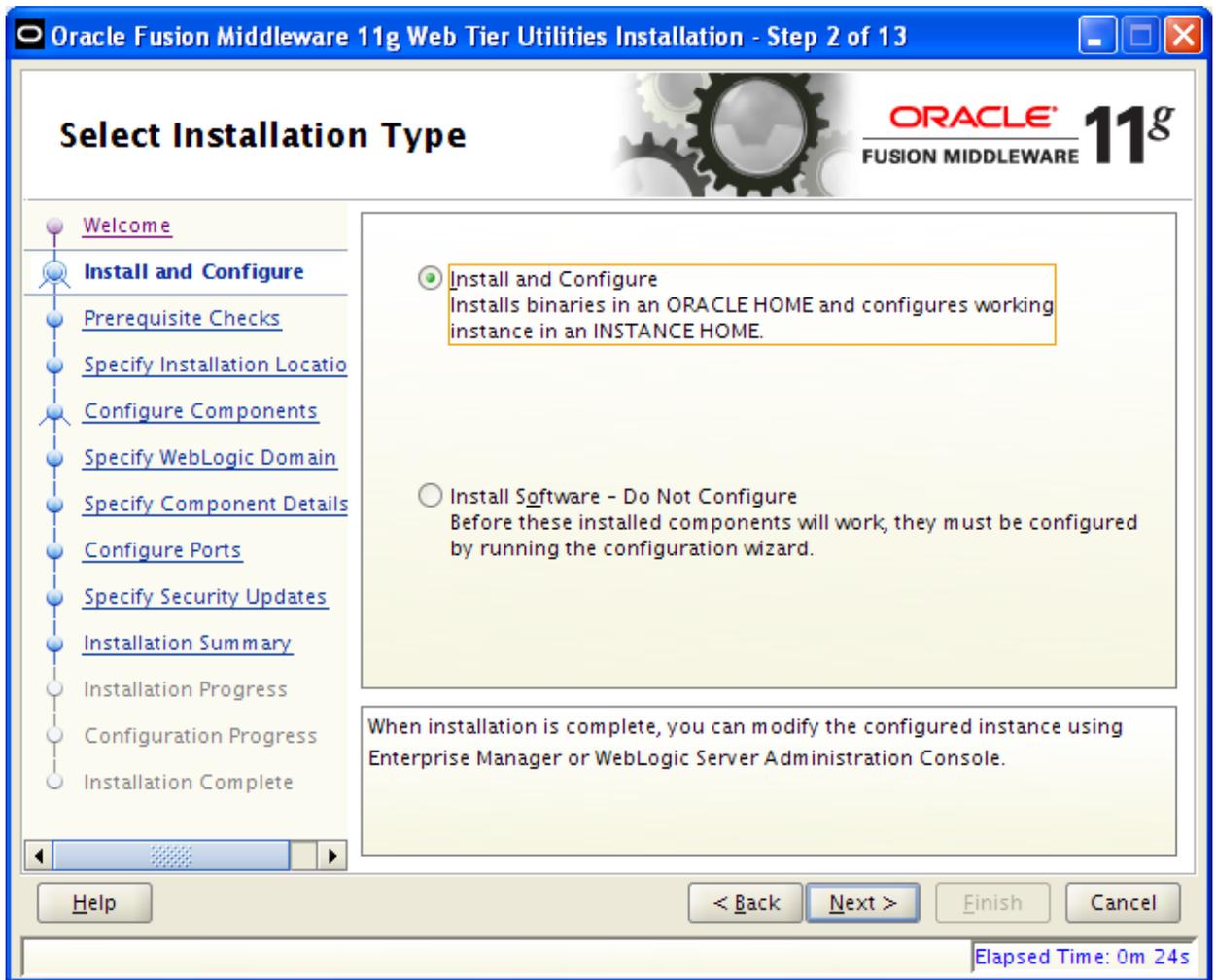
## Installing Oracle HTTP Server

The examples in this document assume you are using a Unix based platform. If you are installing the Oracle HTTP Server on a Windows machine, some of the files names and directories may be slightly different. When installing on Unix, the Oracle web tier components should be installed using a non-root user.

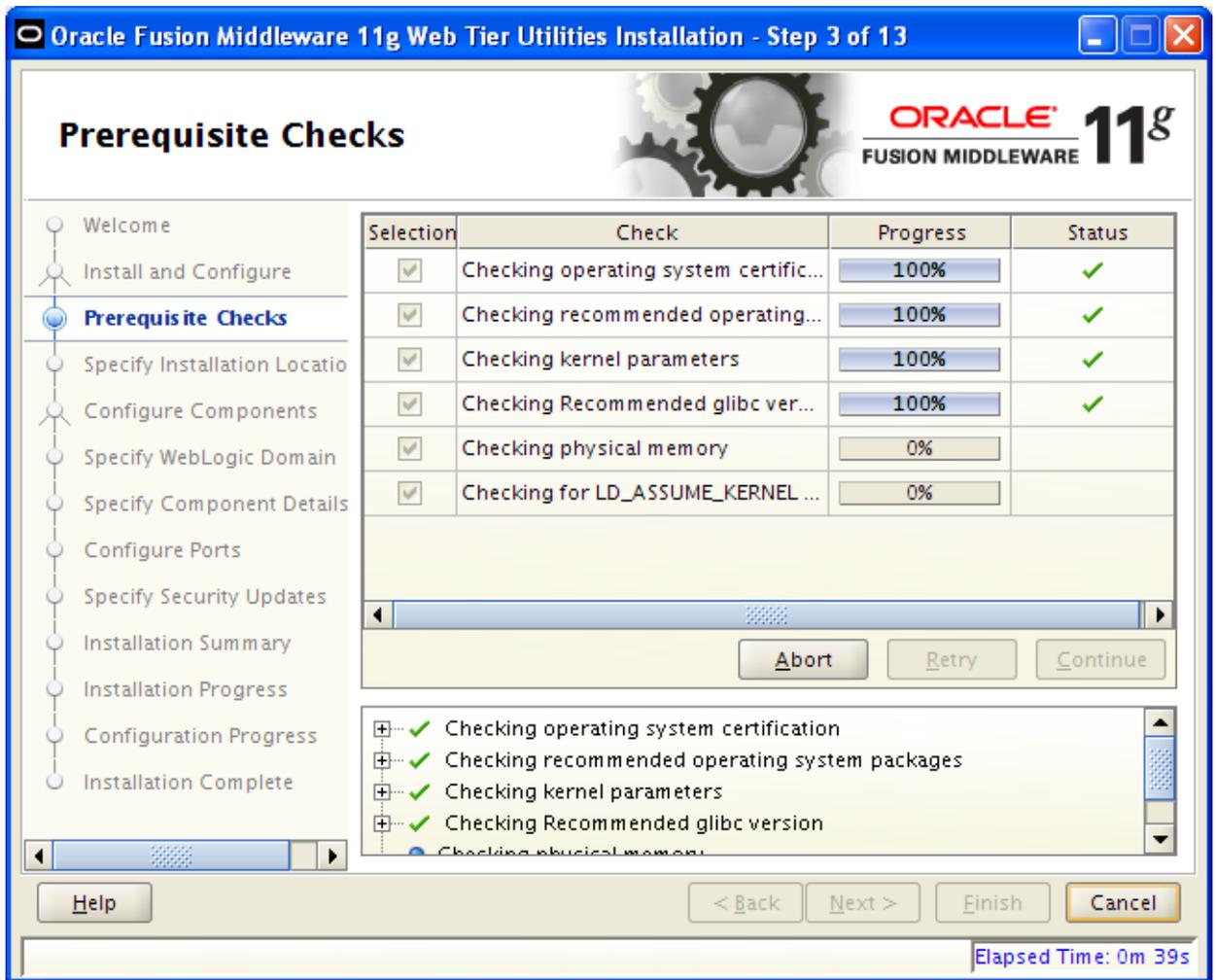
1. Download the “Web Tier Utilities” package that is appropriate for your platform from the edelivery web site (<https://edelivery.oracle.com>). Refer to the JD Edwards EnterpriseOne Current MTR Index (Doc ID 747323.1) for more information.
2. Unzip the downloaded file into a temporary directory on the machine you are targeting for installation. This will create two sub-directories called **Disk1** and **Disk2**.
3. In the **Disk1** sub-directory, you will find the Oracle installer program – **runInstaller**. Run this program to start the installation. The first screen you will see is the welcome screen:



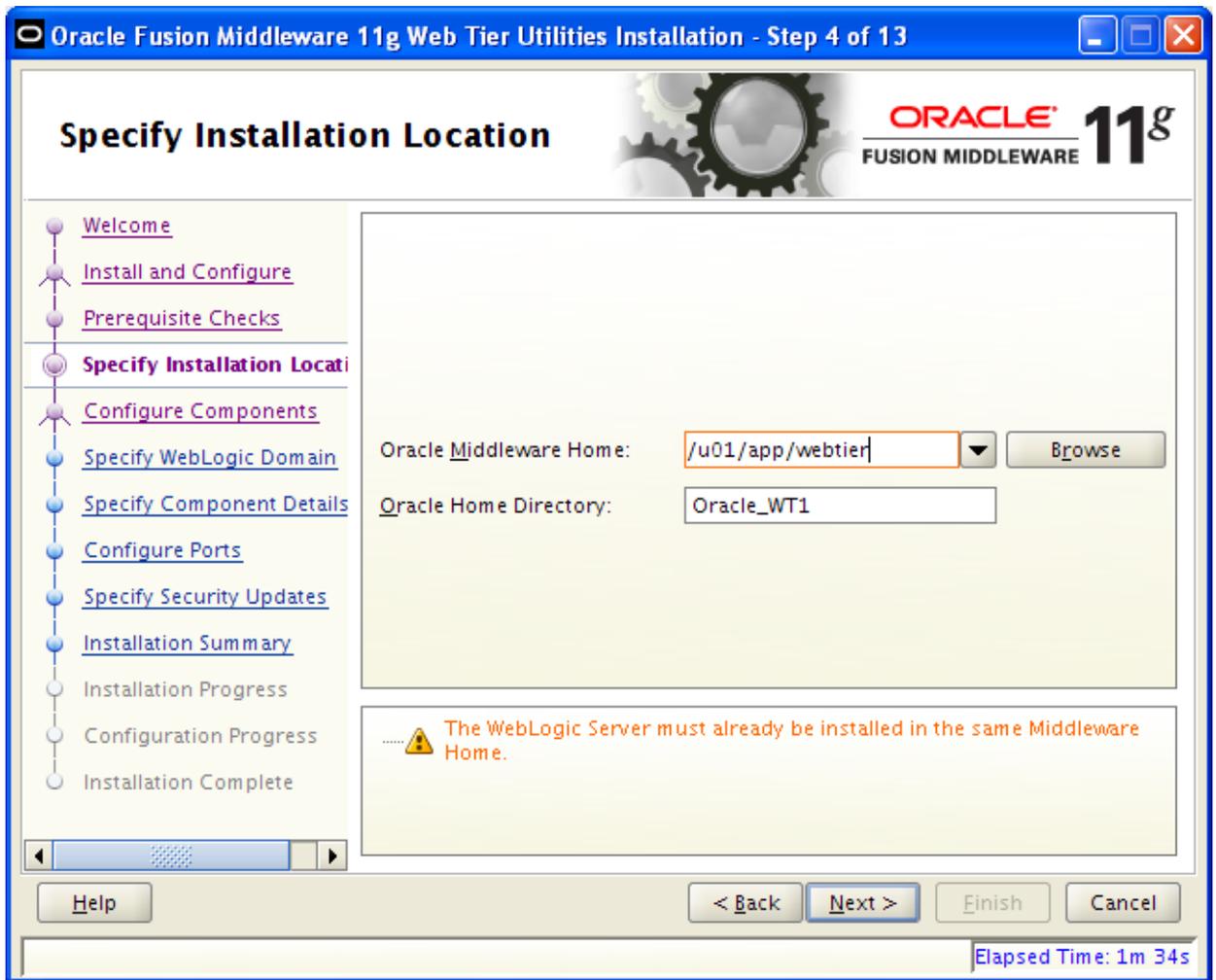
4. Click the "Next" button to begin the installation.



5. The “Install and Configure” option is selected by default. Click “Next”.

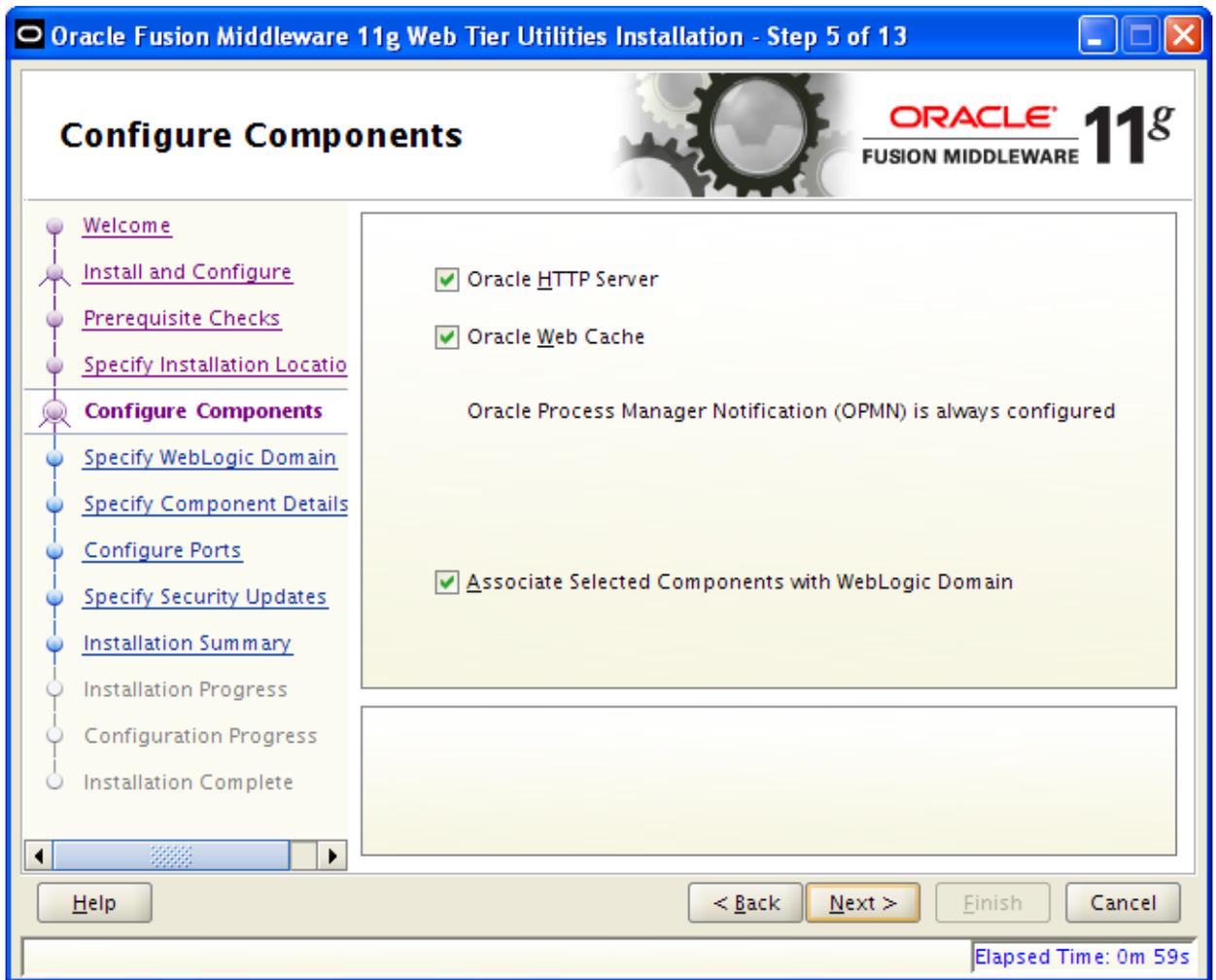


- The system prerequisite checks are run. These will vary slightly depending on which platform you are installing to. You should correct any issues identified by these checks. After these checks complete, click "Next".



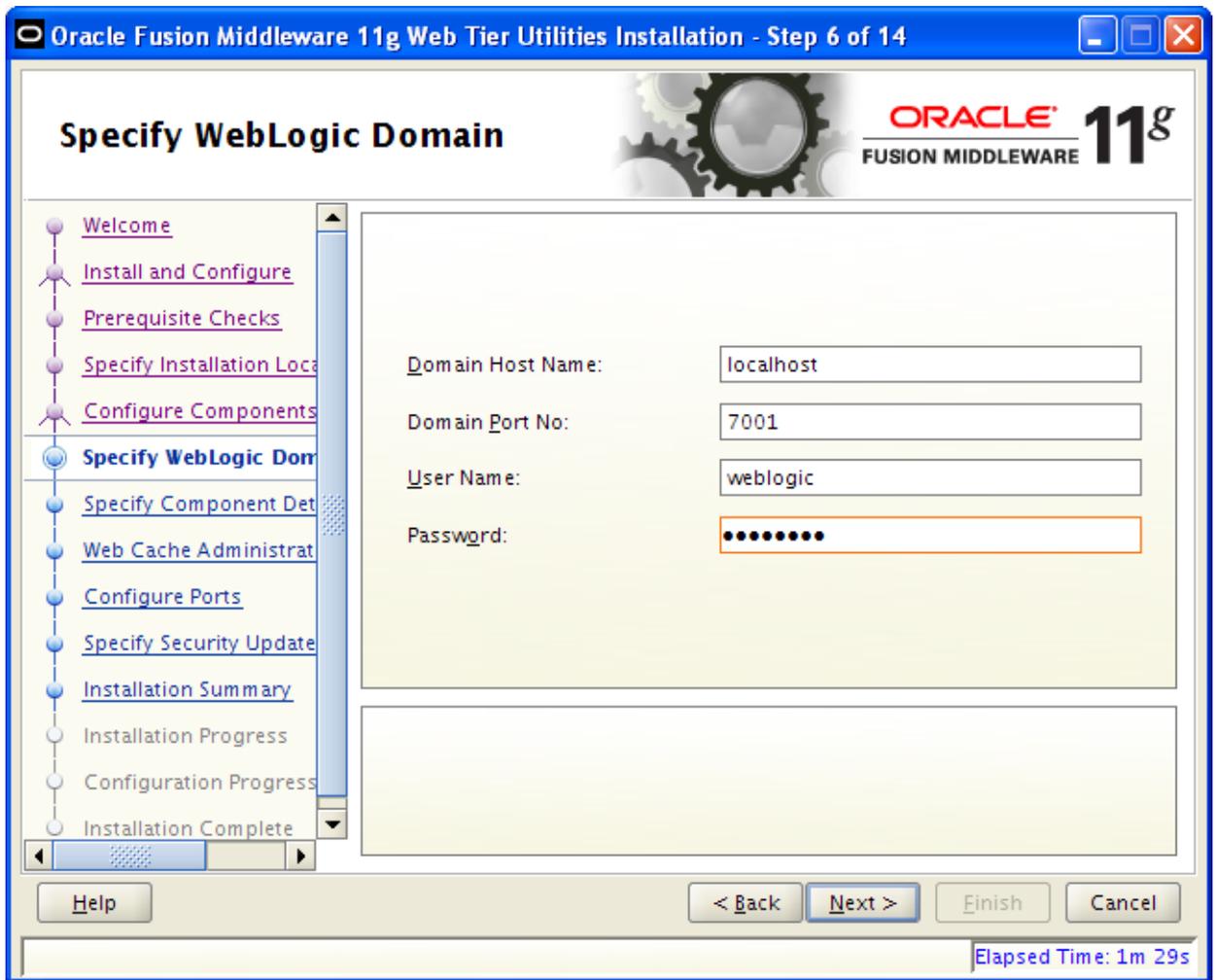
7. Specify the directory where you will install the HTTP Server. Click “Next”. The message stating that the “WebLogic Server must already be installed...” is true only if you plan to use Enterprise Manager to manage your HTTP server. However, most management and configuration can be done without using Enterprise Manager.

**Note:** If you want to configure SSL following the instructions in this document you will need to associate your instance with the WebLogic domain that is running Enterprise Manager (shown on the next screen). See the appendix of this document if you want to install Enterprise Manager to administer your web tier components.



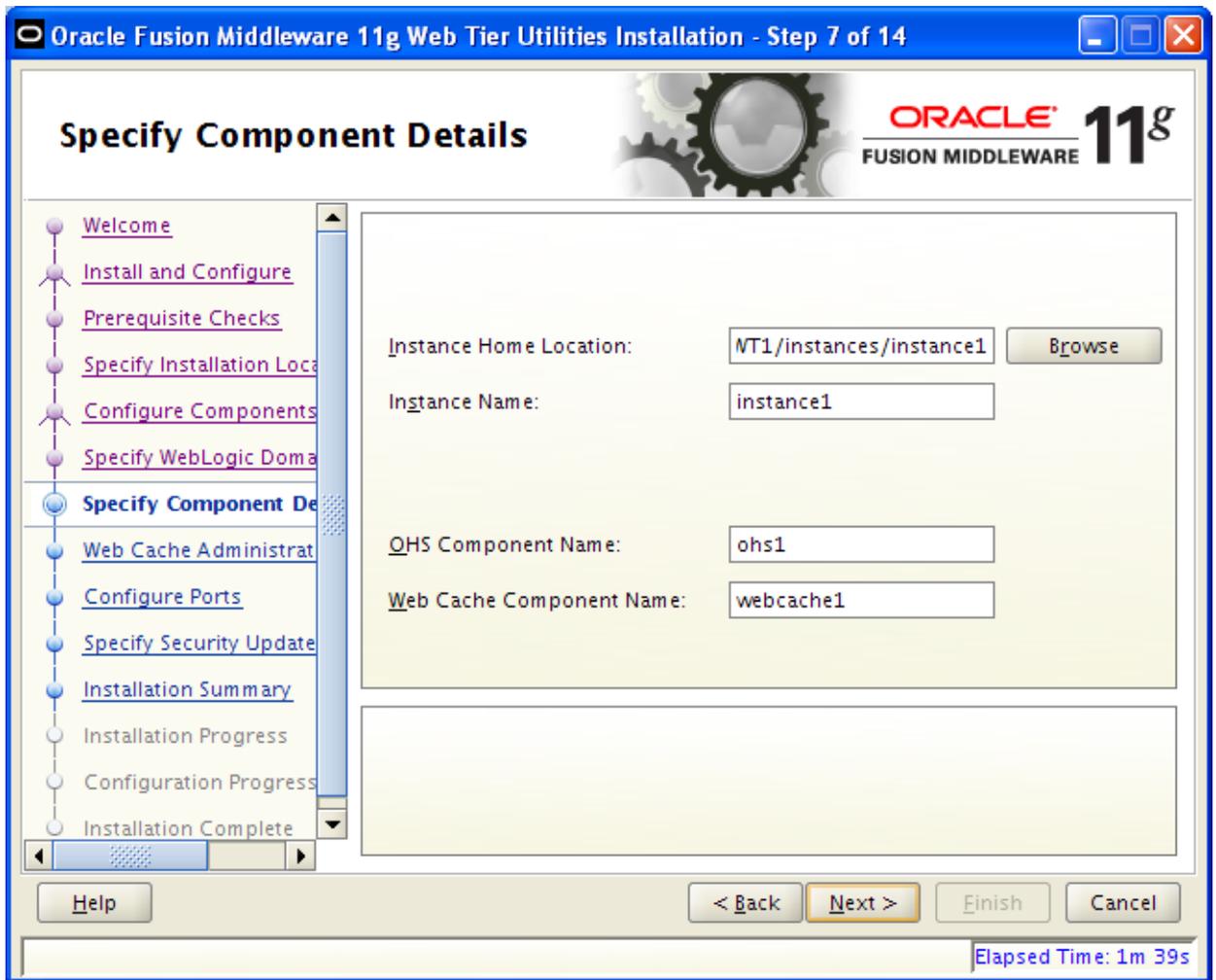
8. For this example, we are installing both Oracle HTTP Server and Oracle Web Cache. If you already have a WebLogic Server installed, you may choose the option to “Associate Selected Components with WebLogic Domain”. This is not necessary, but it will allow you to use the Fusion Middleware Enterprise Manager to manage your HTTP Server. You will have to have previously installed the Fusion Middleware Enterprise Manager for this selection to be valid. After making your selections, click “Next”.

**Note:** If you want to configure SSL following the instructions in this document you will need to associate your instance with the WebLogic domain that is running Enterprise Manager (shown on the next screen). See the appendix of this document if you want to install Enterprise Manager to administer your web tier components.

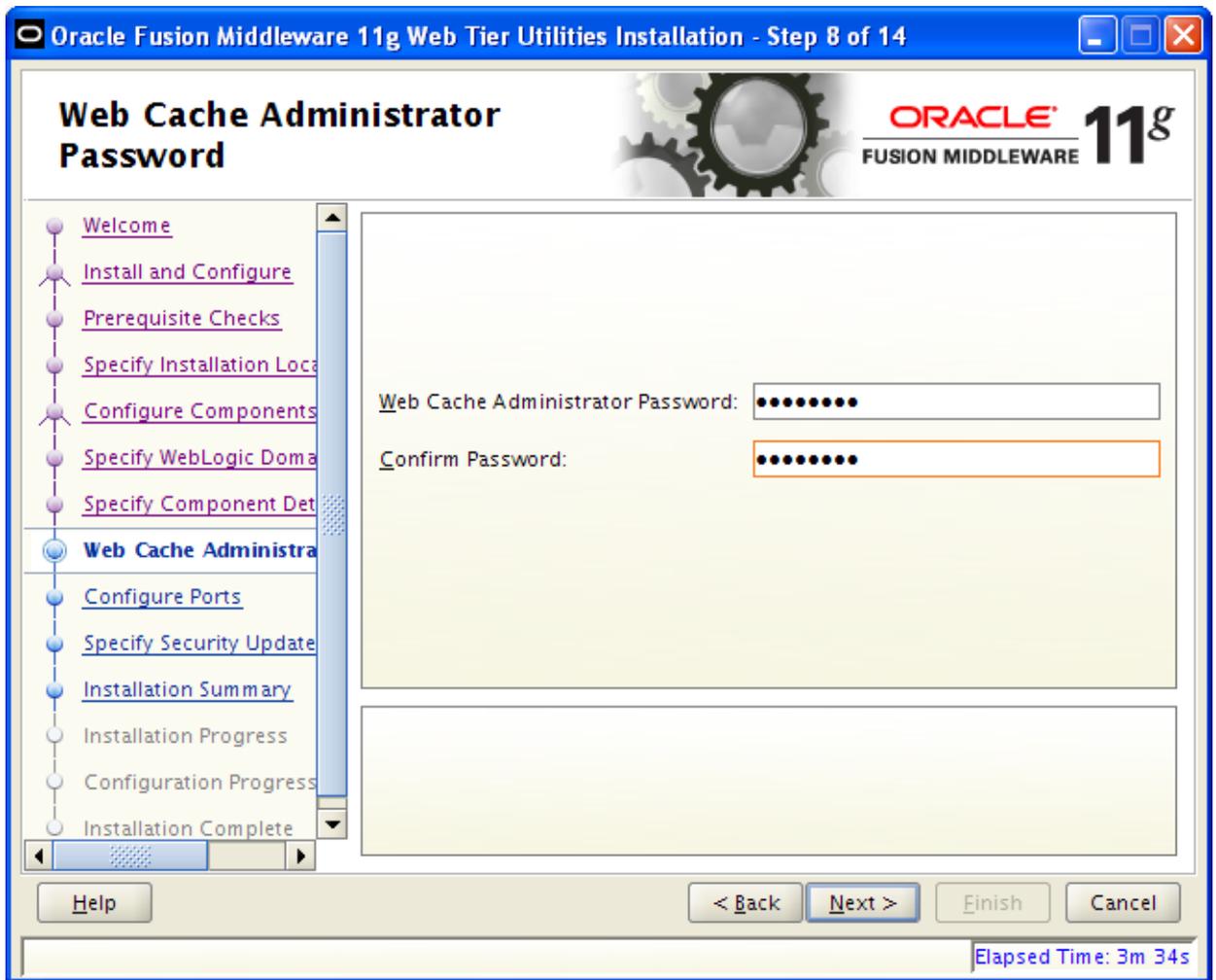


9. This screen will only be displayed if you chose to associate the web tier components with an existing WebLogic Domain. This screen asks for the WebLogic server administration port, user and password. You will only be able to complete this screen if Enterprise Manager is installed and running. Click “Next” after entering the information.

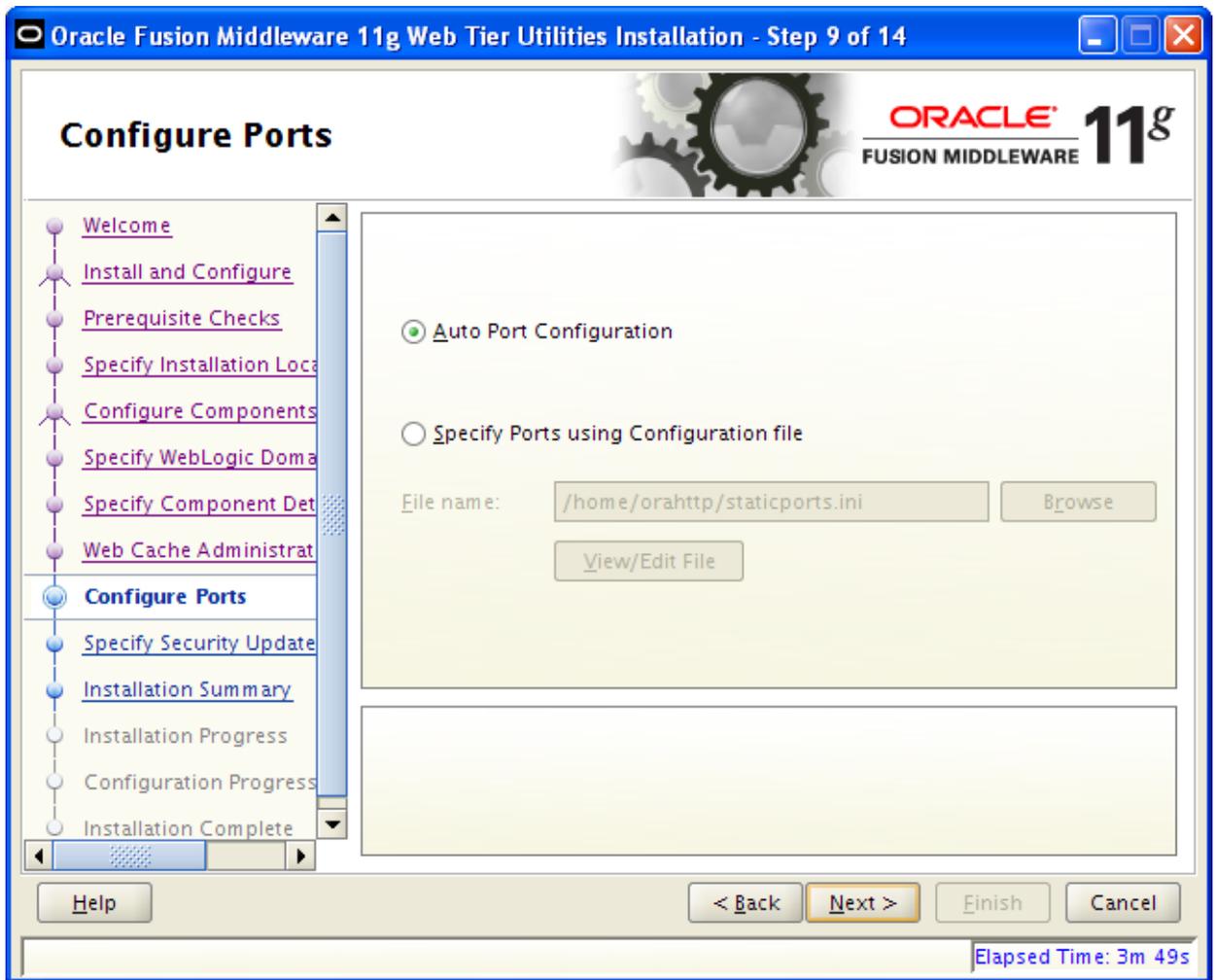
**Note:** Some early versions of the Enterprise Manager application failed to correctly recognize the web tier components unless the default user (“weblogic”) was used. If you find that Enterprise Manager cannot start or stop your web tier components, first check to see whether or not it will work correctly with the default user.



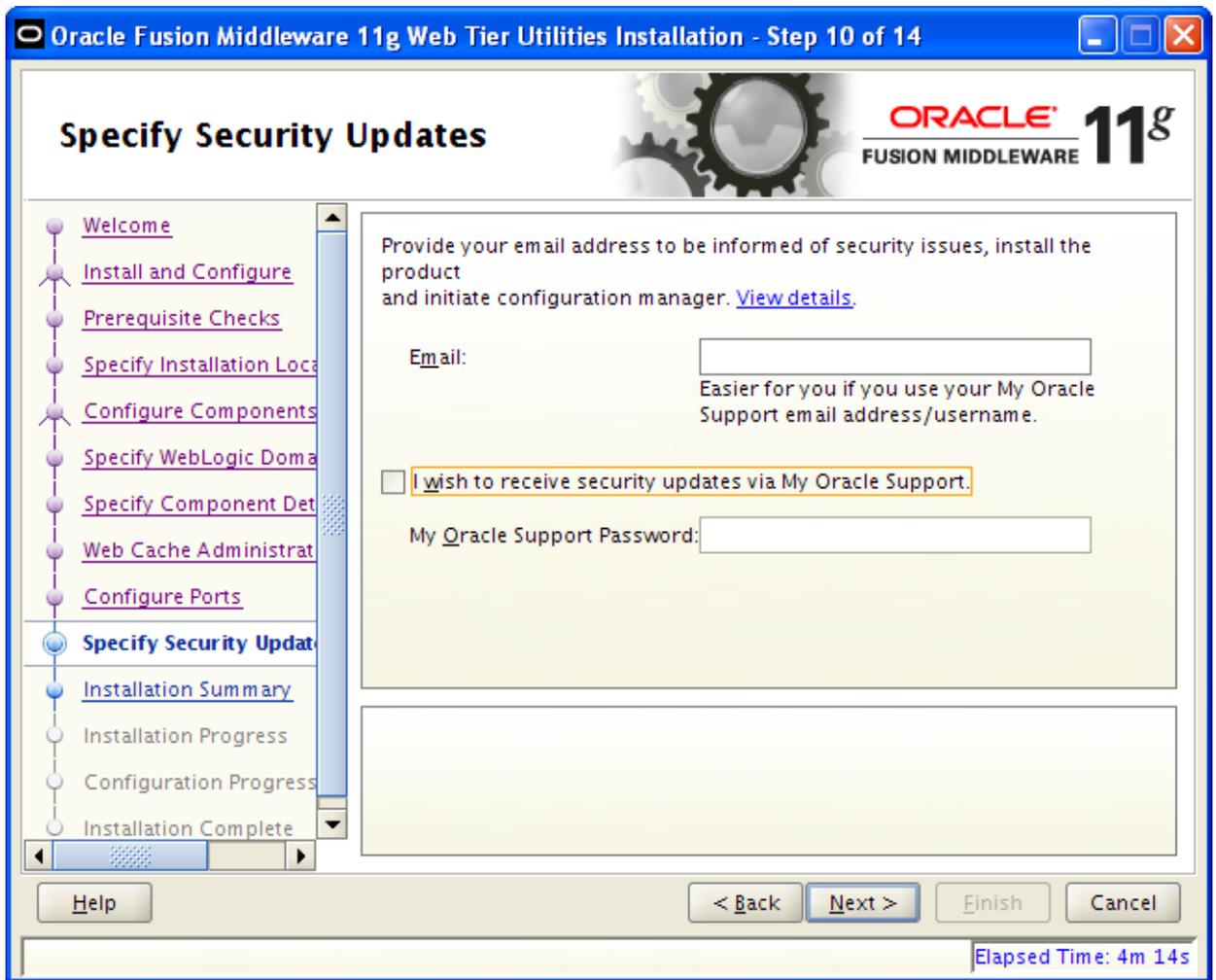
10. This screen allows you to change the directory where the web tier components will be installed. By default, the instance home will be located under the Oracle Middleware Home directory that was previously specified. You may accept the defaults and click “Next”.



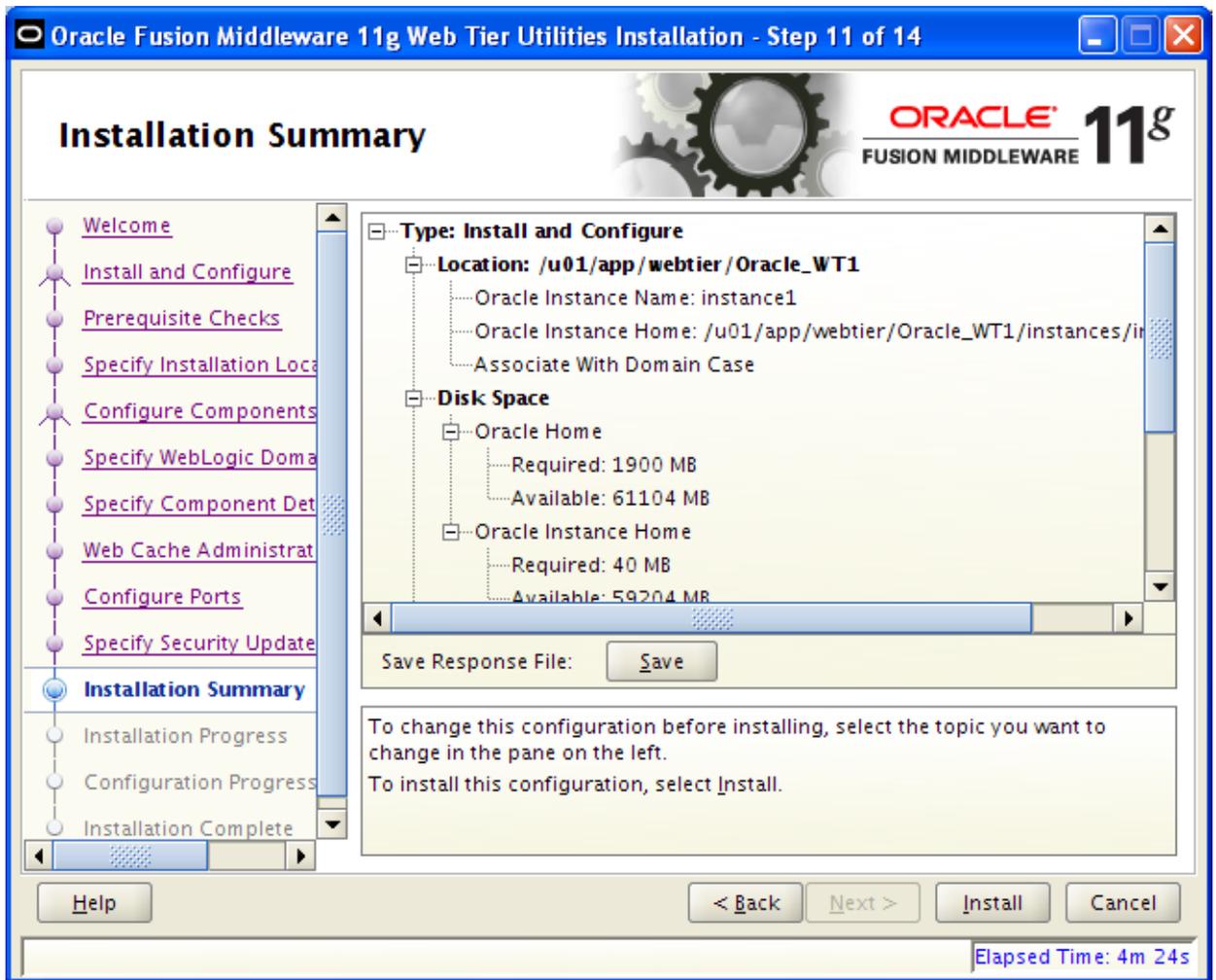
11. This screen is shown if you are installing Web Cache. You must specify a Web Cache administrator password. Click "Next".



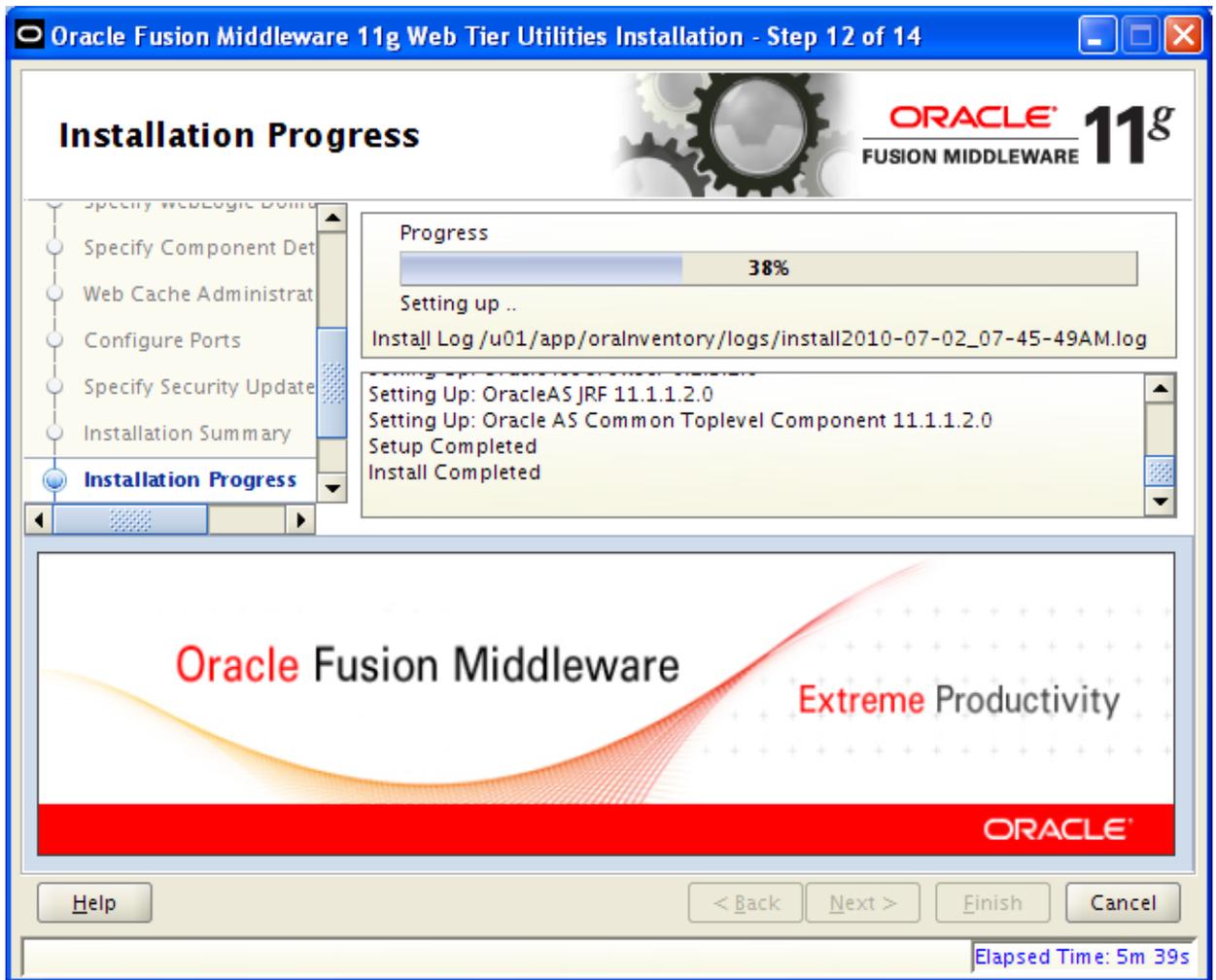
12. This screen allows you to specify the ports that will be used for HTTP configuration. The default is to allow the ports to be automatically configured. Click "Next".



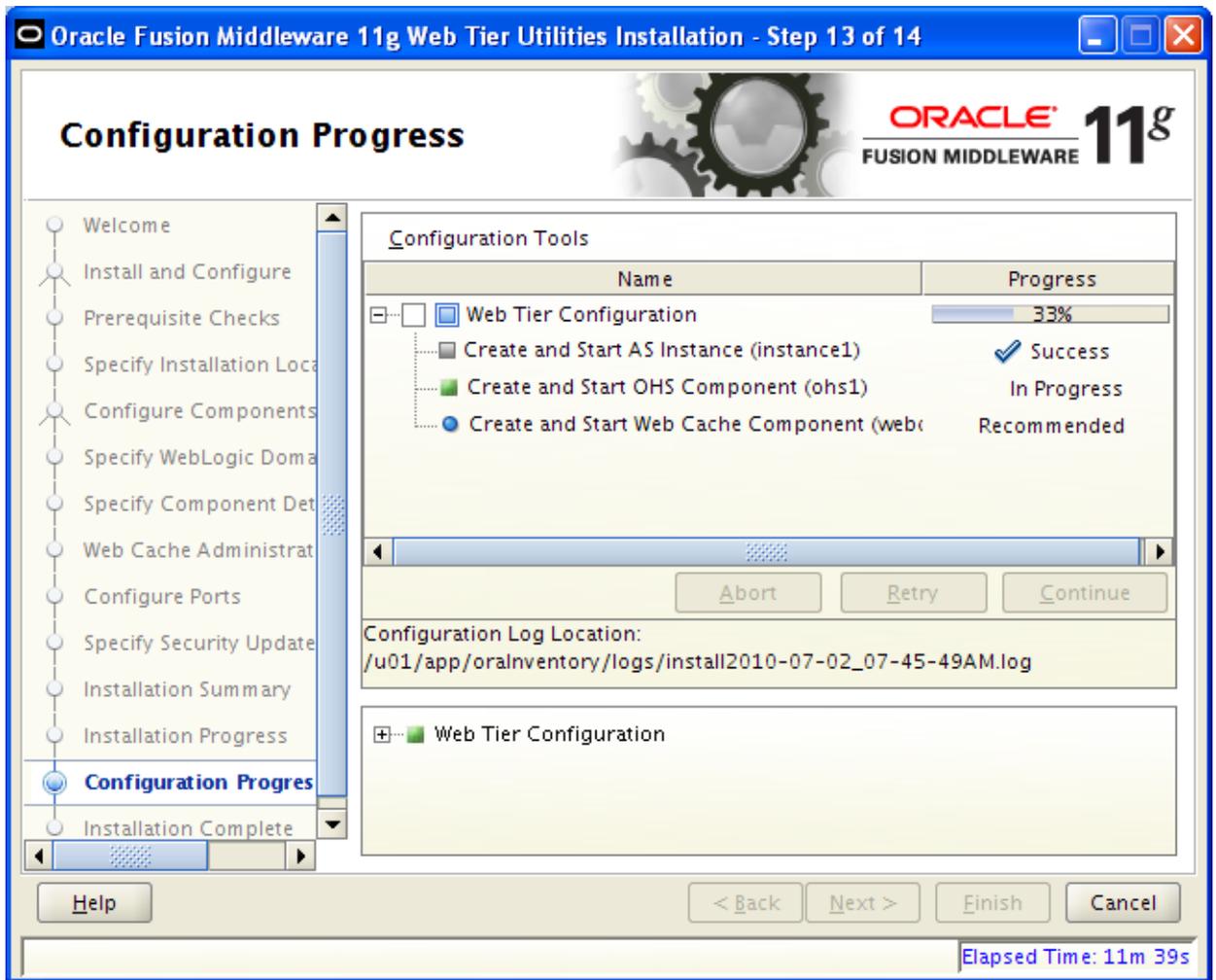
13. This screen allows you to configure an e-mail address to receive updates about the installed product. This is not required. Click "Next".



14. This screen confirms the selections that have been made and checks the available disk space on the system. Click "Install" to proceed with the installation.

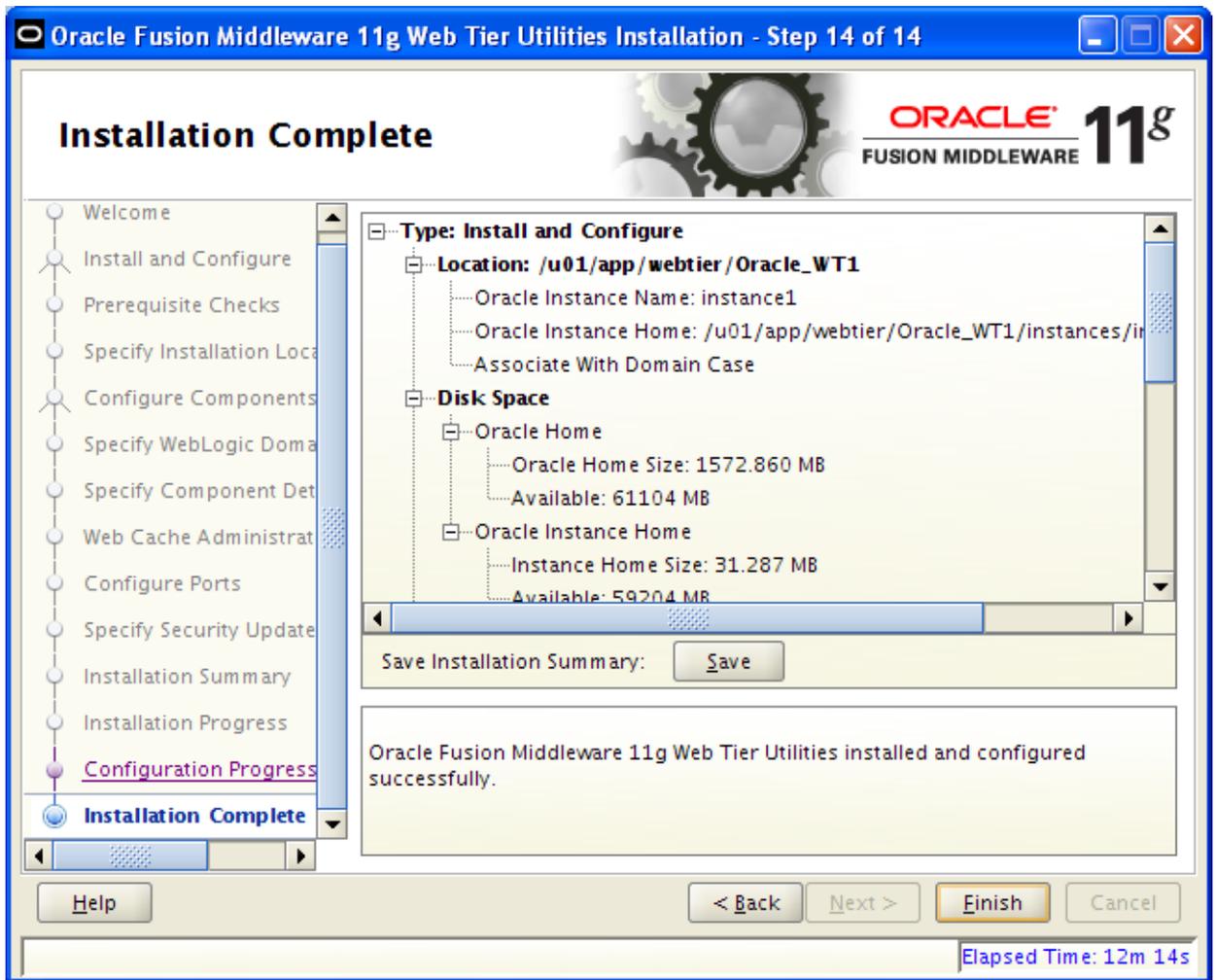


15. Installation progress is shown on this screen.



16. If you have chosen the “Install and Configure” option, the progress of the configuration steps are shown.

**Note:** If this step fails, one possible cause is a failure while deploying the management application to the WebLogic server. You may be able to determine additional failure information by looking at the “AdminServer.log” file in the domain where your Enterprise Manager application is running.



17. Click "Finish" to exit the installer.

## Basic configuration

When going through the following examples, you will notice that some of them use specific machine names, port numbers, and various other component names that will be different in your environment. For reference, below is a list to help you keep track of where the various commands and components are being run, and what you might have to change in your environment:

- dnhpbl04 – Linux server running Oracle web tier components (HTTP Server and Web Cache)
  - 7777 – default HTTP port configured for non-SSL communication
  - 7778 – port configured for SSL communication (HTTPS)
  - 7779 – default HTTP port for second Oracle HTTP instance
  - ohs1 – HTTP Server component name
  - webcache1 – Web Cache component name
- denpbrs1 – AIX server running WebLogic, with EnterpriseOne HTML server already deployed
  - 8081 – port configured for non-SSL connections to EnterpriseOne
  - 8082 – port configured for second (clustered) deployment of EnterpriseOne
  - 8083 – port configured for SSL connections to EnterpriseOne

## Environment Setup

The examples in this document are shown for a Linux machine, and appropriate adjustments should be made if you are using a Windows machine. When using a Unix based machine, there are several environment variables that can be set that will make some of the steps below easier. Below shows the environment variables that have been set in the profile for the Oracle HTTP user for these examples:

```
export ORACLE_BASE=/u01/app
export ORACLE_HOME=$ORACLE_BASE/webtier/Oracle_WT1
export ORACLE_INSTANCE=$ORACLE_HOME/instances/instance1
export CDPATH=.:$ORACLE_HOME
export PATH=$PATH:$ORACLE_INSTANCE/bin:$ORACLE_BASE/webtier/oracle_common/bin
```

Adding the `bin` directory for the instance to the `PATH` will allow you to run the `opmnctl` command without specifying the full directory path. Adding the `bin` directory for the common component directory will allow you to easily run the `orapki` command that is used in the SSL configuration.

One other thing to check at this point is whether or not the java “keytool” utility is in your `PATH`. This utility will be used to generate keys, as well as import and export certificates during the SSL configuration examples.

## Configuring the OHS Proxy for WebLogic Server

At this point you will want to configure the Oracle HTTP Server installation to communicate with an existing EnterpriseOne HTML server running in WebLogic. The following shows how to configure the default HTTP port (7777) in OHS and dispatch a request to the EnterpriseOne HTML client port (8081 in this example) running on a managed WebLogic server.

1. Find the `mod_wl_ohs.conf` to the following directory:

```
<Oracle Middleware Home>/Oracle_WT1/instances/instance1/config/OHS/ohs1
```

2. In `mod_wls_ohs.conf`, add the following lines to the end of the file:

```
<VirtualHost *:7777>
  <Location /jde>
    SetHandler weblogic-handler
    WebLogicHost denpbrs1
    WebLogicPort 8081
  </Location>
</VirtualHost>
```

3. Restart OHS using `opmnctl` (this assumes that `opmnctl` is in your `PATH`):

- `opmnctl stopall`
- `opmnctl startall`

You should now be able to connect to the EnterpriseOne HTML client using your HTTP server host name with port 7777 in place of the EnterpriseOne server host name with port 8081.

**Old:**     `http://denpbrs1:8081/jde/owhtml`

**New:**     `http://dnhpb104:7777/jde/owhtml`

If you use a port other than the default Oracle HTTP port of 7777, you will also have to edit the `httpd.conf` file to add a listener for that port.

# Configuring SSL

SSL secures communication by providing message encryption, integrity, and authentication. The SSL standard allows the involved components (such as browsers and HTTP servers) to negotiate which encryption, authentication, and integrity.

- Encryption provides confidentiality by allowing only the intended recipient to read the message.
- Integrity ensures that a message sent by a client is received intact by the server, unaltered.
- Authentication enables the server and client to check that the other party is who it claims to be.

Oracle Fusion Middleware supports two types of keystores for keys and certificates

- JKS-based keystore and truststore
- Oracle wallet

Java components and applications use the JKS keystore to store keys and certificates. Configuring SSL for these components thus requires setting up and using JKS keystores.

Other components use the Oracle wallet as their storage mechanism. An Oracle wallet is a container that stores your credentials, such as certificates, trusted certificates, certificate requests, and private keys. You can store Oracle wallets on the file system or in LDAP directories such as Oracle Internet Directory. Oracle wallets can be auto-login or password-protected wallets.

Components that use Oracle wallet include:

- Oracle HTTP Server
- Oracle Web Cache
- Oracle Internet Directory

Configuring SSL for these components thus requires setting up and using Oracle wallets.

There are various lines of communication that you may want to secure with SSL. One consideration is that the computation required to encrypt and decrypt HTML pages will affect performance. The examples below will walk through configuring SSL for two different communication lines by building on the basic configuration above, and will also demonstrate the use of both a JKS keystore and an Oracle Wallet.

## ***Inbound SSL to Oracle HTTP Server***

First we'll look at securing the communication between the client browser and Oracle HTTP Server. To start, we'll create a second virtual host by duplicating the section of the `mod_wl_ohs.conf` file that was changed above.

1. In `mod_wls_ohs.conf`, add the following lines to the end of the file:

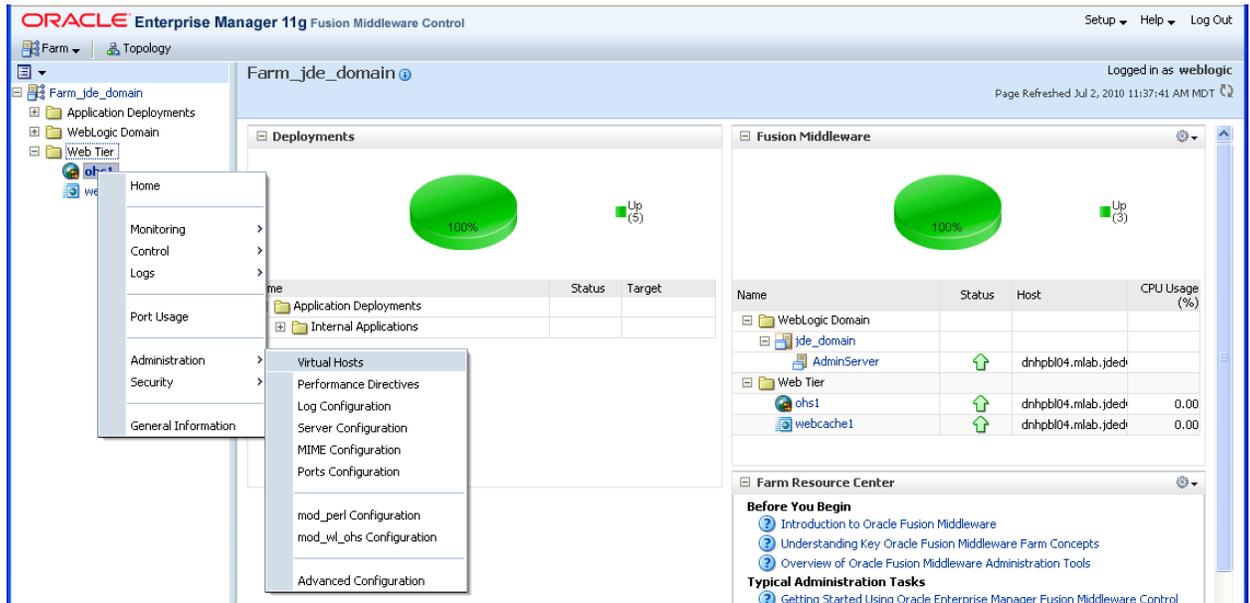
```
<VirtualHost *:7778>
  <Location /jde>
    SetHandler weblogic-handler
    WebLogicHost denpbrsl
    WebLogicPort 8081
  </Location>
</VirtualHost>
```

2. Since we are not using the default Oracle HTTP listen port, we'll also need to add the following line to the `httpd.conf` file (this can also be added just above the `VirtualHost` directive in the `mod_wls_ohs.conf` file):

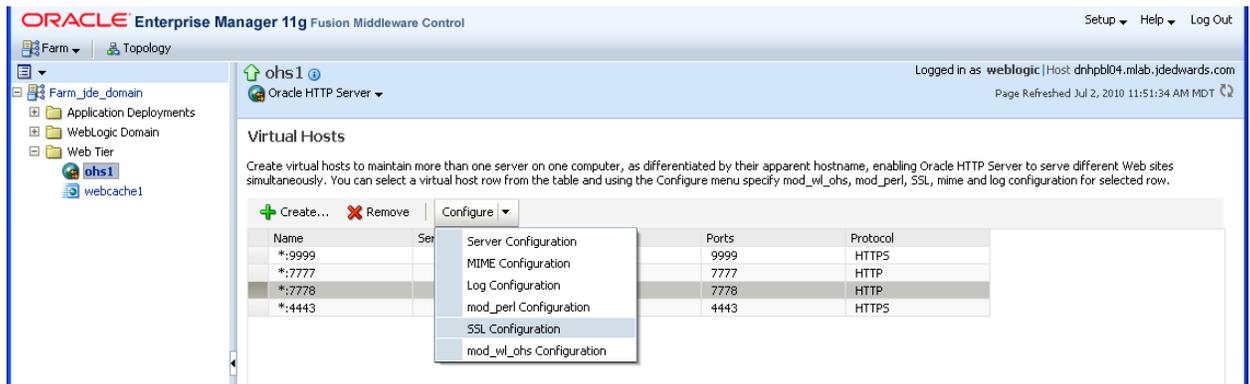
```
Listen 7778
```

- Now we will use Enterprise Manager to configure the inbound communication to the Oracle HTTP Server. This configuration can also be done using the WebLogic scripting tool (WLST), as is documented in the *Oracle Fusion Middleware Administrator's Guide 11g*. You could also modify the `mod_wls_ohs.conf` file manually using the example at the end of this section.

Sign into Enterprise Manager and verify that your Oracle HTTP Server instance is running. Expand the Web Tier section on the left side of the display and right-click on the Oracle HTTP Server (ohs1) component. Select "Administration", then "Virtual Hosts":



- Highlight the virtual host you wish to update, then click "Configure", and select "SSL Configuration".



- The next screen allows you to enable SSL for this virtual host. You can select which type of SSL authentication will be used, the SSL protocols will be accepted, and other attributes. If you have created a custom Oracle wallet, this can also be specified here. For our example, you can just select the "Enable SSL" check box and leave the default values.
- Click **OK** to apply the changes. You will be returned to the virtual hosts screen, and you should now see that the virtual host protocol is set to **HTTPS**.
- Restart the Oracle HTTP Server instance from Enterprise Manager by right-clicking on **ohs1**, and selecting **Control**, followed by **Restart**. You can also restart Oracle HTTP Server by using the `opmnctl` commands given in the previous section of this document.

At this point you have secured the communication channel to the Oracle HTTP Server. You can test this by opening a browser window and trying to connect to the JD Edwards EnterpriseOne HTML application as follows:

```
https://dnhpbl04:7778/jde/E1Menu.maf
```

This will bring up a “Certificate Error” page in Internet Explorer, or an “Untrusted Connection” page in Firefox. This is because we are using the default Oracle Wallet that is delivered with Oracle HTTP Server. This wallet contains a self-signed certificate that is not trusted by your browser. If you were to set up SSL for a production environment, you would most likely be using a different Oracle Wallet that contained your production certificate that was signed by a trusted Certificate Authority. For our purposes, you can select the browser option to allow the untrusted certificate.

If you look at the VirtualHost entry in the mod\_wl\_ohs.conf file, it will now look something like this:

```
<VirtualHost *:7778>
  <Location /jde>
    SetHandler weblogic-handler
    WebLogicHost denpbrsl
    WebLogicPort 8081
  </Location>
  <IfModule ossl_module>
    SSLEngine on
    SSLProtocol nzos_Version_1_0 nzos_Version_3_0_With_2_0_Hello nzos_Version_3_0
    SSLCipherSuite
SSL_RSA_WITH_RC4_128_MD5,SSL_RSA_WITH_RC4_128_SHA,SSL_RSA_WITH_3DES_EDE_CBC_SHA,SSL_RSA_WITH_DES_
CBC_SHA,TLS_RSA_WITH_AES_128_CBC_SHA,TLS_RSA_WITH_AES_256_CBC_SHA
    SSLVerifyClient none
    SSLWallet "${ORACLE_INSTANCE}/config/${COMPONENT_TYPE}/${COMPONENT_NAME}/keystores/default"
    SSLCRLCheck Off
  </IfModule>
</VirtualHost>
```

## Inbound SSL to Oracle WebLogic Server

If you elect to secure the communication channel from the browser to the Oracle HTTP Server, you must also want to secure the communications between the Oracle HTTP Server and the Oracle WebLogic Server applications. This consists of two steps: configuring SSL for your WebLogic managed server, and then configuring Oracle HTTP Server as a client that trusts the WebLogic server.

First, to configure your WebLogic managed server to use SSL (these steps are adapted for this example from the *Oracle Fusion Middleware Administrator's Guide for Oracle WebCenter 11g Release 1*, section 23.6):

1. For this example, we will use the java “keytool” utility to create a custom keystore that will contain our server identity. These commands assume the Java 1.6 version of “keytool” is in your PATH. An example command to generate a custom key pair is below (the values in blue will change based on your site):

```
keytool -genkeypair -v -keyalg RSA -dname "cn=denpbrsl,dc=mlab,dc=jdwards,dc=com" -
alias enterpriseone -keypass admin123 -keystore enterpriseone.jks -storepass admin123 -
validity 365
```

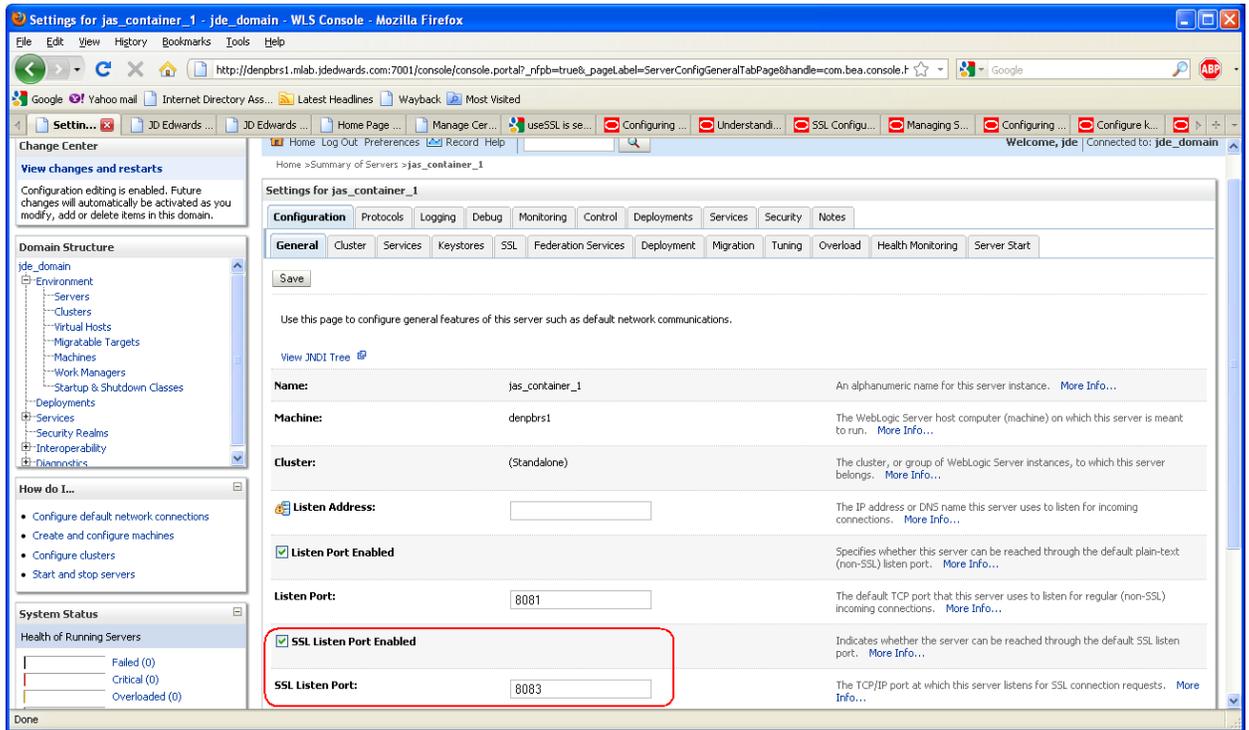
**Note:** After one year this certification will expire and you will need to regenerate the certificate in order for the server to start.

2. Export the certificate containing the public key that will be imported into the client’s trust store. In this case, the client will be the Oracle HTTP Server, with the server being the Oracle WebLogic managed server:

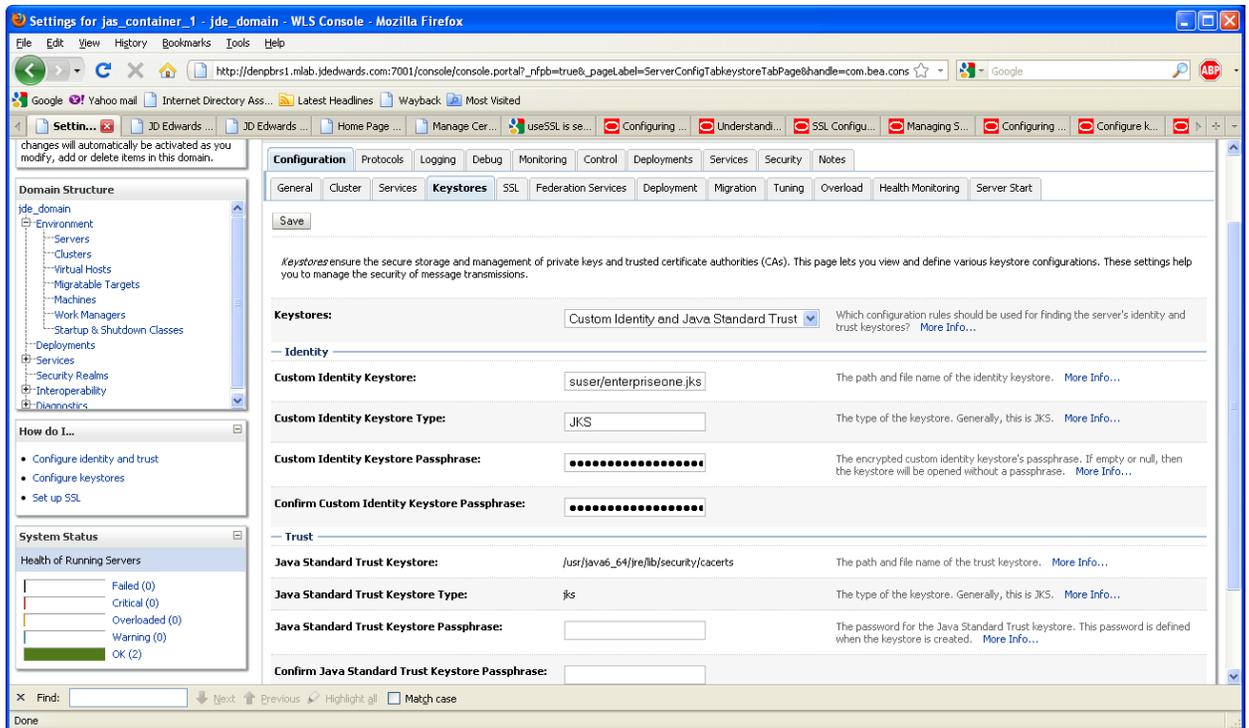
```
keytool -exportcert -v -alias enterpriseone -keystore enterpriseone.jks -storepass
admin123 -rfc -file enterpriseone.cer
```

3. Sign on to the WebLogic administration console on the server that is hosting your EnterpriseOne HTML application. Select the managed server that is running EnterpriseOne from the list of servers. If you are using a production domain, you will need to select the “Lock & Edit” button before you can edit any configuration information. On the “Configuration” tab, and the “General” sub-tab, select the “SSL Listen Port

Enabled” check box and enter the port number to be used in the “SSL Listen Port” field. Click “Save” at the bottom of the screen after these selections have been made. An example is shown below:



4. Next, select the “Keystores” sub-tab.



From the “Keystores” drop-down list, select “Custom Identity and Java Standard Trust”. In the “Identity” section, fill in the “Custom Identity Keystore” field with the path and file name of the identity keystore that was created in step 1. Fill in “JKS” for the “Custom Identity Keystore Type”. And fill in the keystore

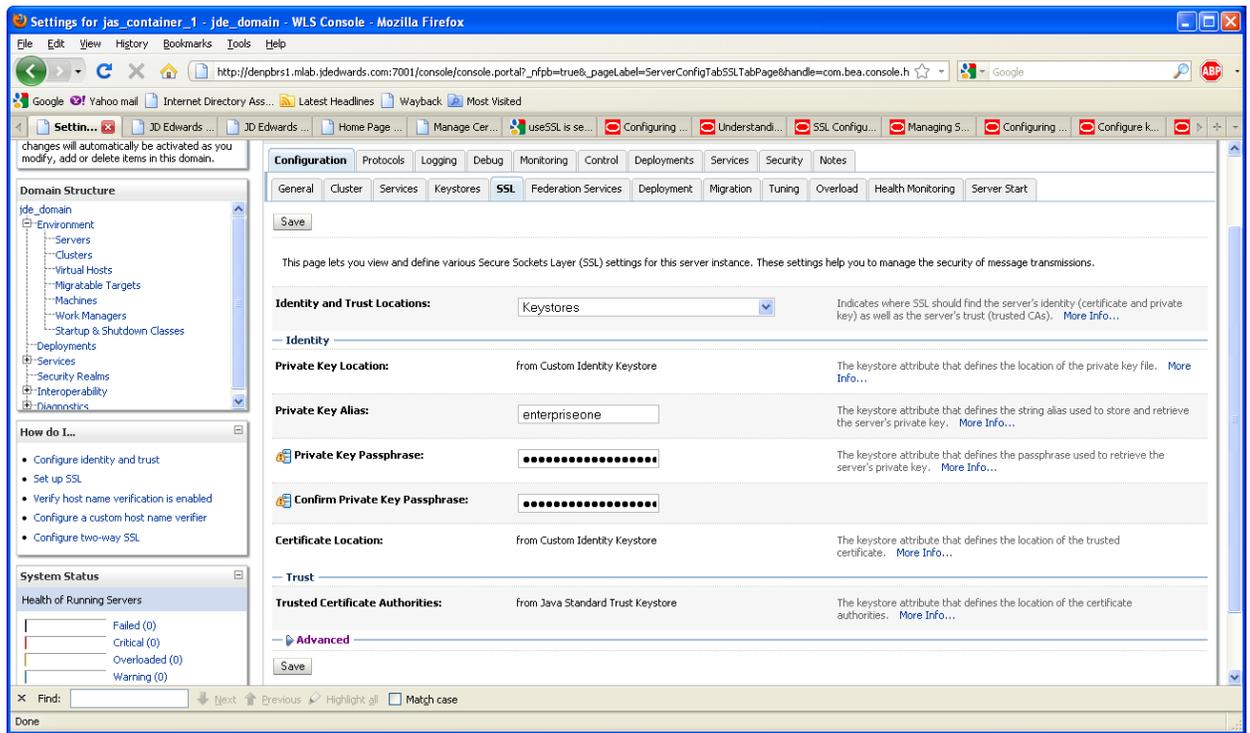
passphrase with the password that was used in step 1. Click “Save” at the bottom of the screen to apply your changes.

While on this screen, notice the path and file name of the “Java Standard Trust Keystore” at the top of the “Trust” section. This is the file containing the recognized Certificate Authorities that WebLogic will recognize. We will use this file in the next step.

- From the command line, run the keytool command to import the self-signed certificate that was created in step 2 above into the Java trust keystore (“changeit” is the default password for the Java trust keystore):

```
keytool -importcert -trustcacerts -alias enterpriseone -file
/home/wlsuser/enterpriseone.cer -keystore /usr/java6_64/jre/lib/security/cacerts
-storepass changeit
```

- Back in the WebLogic console; select the “SSL” sub-tab.



Fill in the “Private Key Alias” and “Private Key Passphrase” for the certificate that was created in step 2 above. Click “Save” to apply your changes.

- Select the “Control” tab for the server and the click “Restart SSL”.

At this point you should be able to connect directly to the EnterpriseOne WebLogic server using SSL. To test this, open a browser window and type in the following URL:

```
https://denbrs1:8083/jde/E1Menu.maf
```

This will again bring up an “Untrusted Connection” page that can be accepted for this example.

## Outbound SSL from Oracle HTTP Server

Now that we have set up the WebLogic Server to use SSL, we need to configure the Oracle HTTP Server to recognize our WebLogic server as a valid Certificate Authority. The steps below can be followed to complete the SSL communication loop (these steps are adapted for this example from the *Oracle Fusion Middleware Administrator's Guide 11g Release 1*, section 6.4):

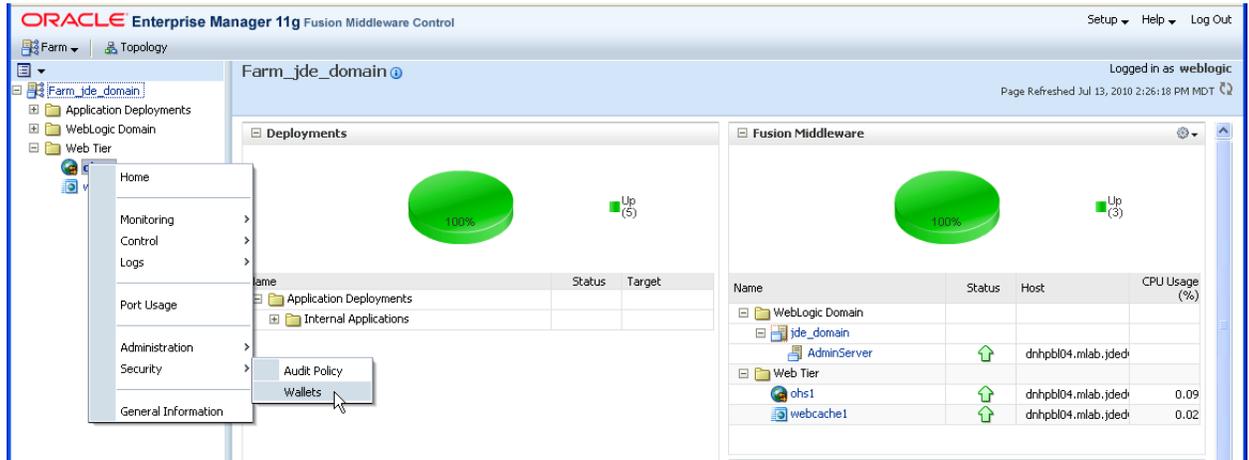
- Copy the file containing the public key certificate to the machine that is running your Oracle HTTP Server. In our example, this is the **enterpriseone.cer** file that was created in step 2 of the previous section. This certificate must be imported into the Oracle wallet that is being used by Oracle HTTP Server.

- There are several ways to import this certificate into the Oracle HTTP Server wallet. One is to use the `orapki` utility, located in the bin directory under your `oracle_common` directory. The example below assumes that the `orapki` utility is already in your PATH, and you have changed the local directory to be the default wallet directory (`$ORACLE_INSTANCE/config/OHS/ohs1/keystores/default`)

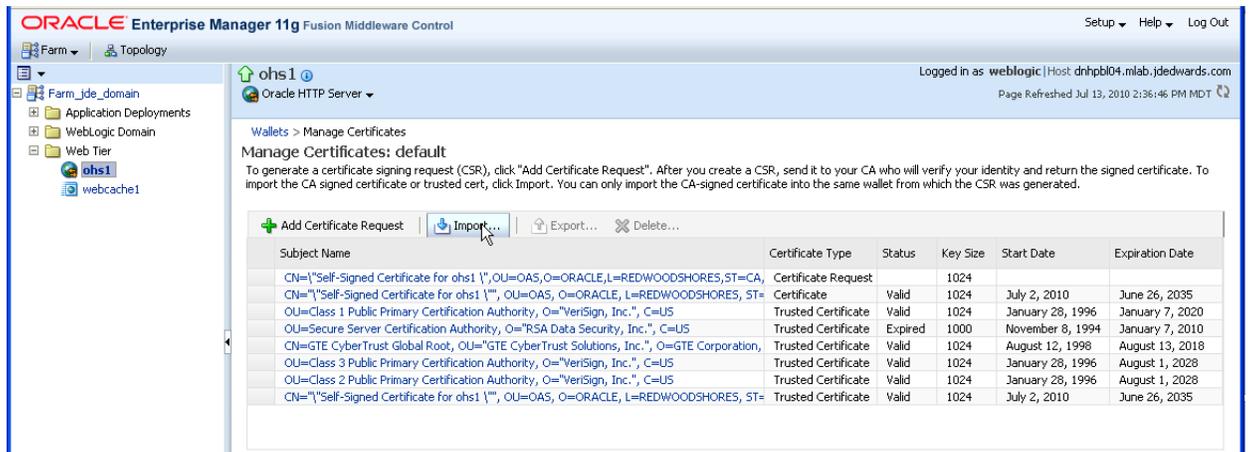
```
orapki wallet add -wallet . -trusted_cert -cert /home/orahttp/enterpriseone.cer -
auto_login_only
```

Alternately, you can use the Enterprise Manager application to import the certificate into your wallet:

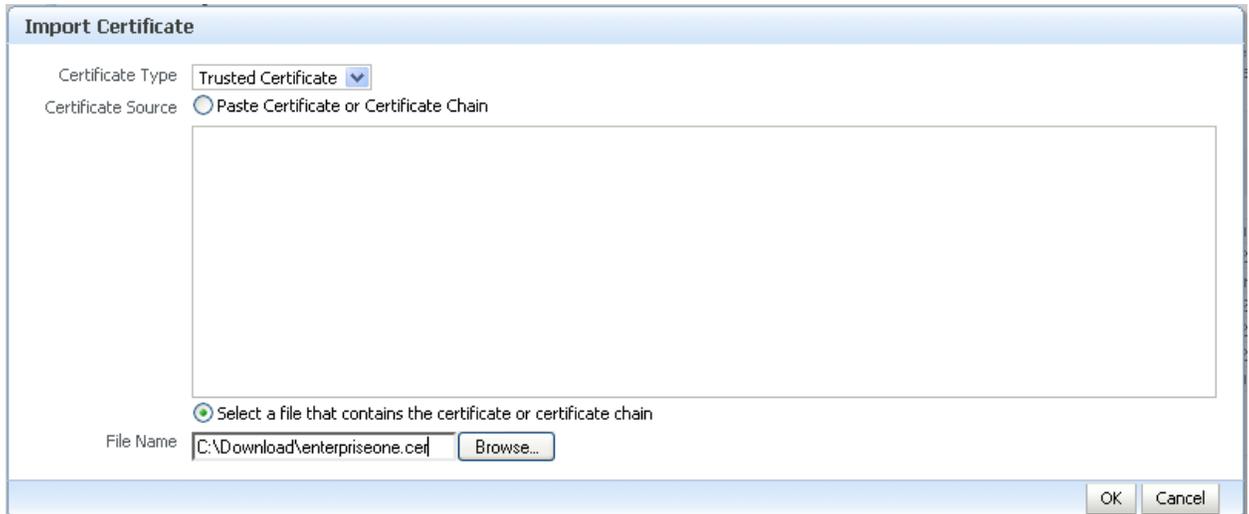
Right-click on the Oracle HTTP Server instance, select “Security”, then “Wallets”:



Select the wallet you wish to update, and you should see a list of the certificates that are currently contained in the wallet. Click on the “Import...” button:



Enter the path and file containing the certificate you wish to import in the field provided. Note that this file will now have to be located on the same machine where you are running your browser, rather than on the machine that is running the Oracle HTTP Server:



Click “OK” to import the certificate. You should see the newly imported certificate in the list.

- After importing the certificate, you will need to edit the virtual host definition in the `mod_wl_ohs.conf` file to enable outbound SSL by adding the following lines *inside* of the `VirtualHost` section. We are also going to change the “WebLogicPort” parameter for this virtual host from 8081 to 8083, since we will now be communicating to the SSL enabled WebLogic port.

```

...
WebLogicPort 8083
...

<IfModule mod_weblogic.c>
  SecureProxy on
  WlSSLWallet "${ORACLE_INSTANCE}/config/OHS/ohs1/keystores/default"
</IfModule>

```

- Restart the Oracle HTTP Server using `opmnctl` or from the Enterprise Manager application.

You should now be able to connect to the EnterpriseOne application through Oracle HTTP Server using SSL. To test this, open a browser window and type in the following URL:

```
https://dnhpb104:7778/jde/E1Menu.maf
```

Looking at the `VirtualHost` entry in the `mod_wl_ohs.conf` file, the final version will now look something like this:

```
<VirtualHost *:7778>
  <Location /jde>
    SetHandler weblogic-handler
    WebLogicHost denpbrsl
    WebLogicPort 8083
  </Location>
  <IfModule ssl_module>
    SSLEngine on
    SSLProtocol nzos_Version_1_0 nzos_Version_3_0_With_2_0_Hello nzos_Version_3_0
    SSLCipherSuite
SSL_RSA_WITH_RC4_128_MD5,SSL_RSA_WITH_RC4_128_SHA,SSL_RSA_WITH_3DES_EDE_CBC_SHA,SSL_RSA_WITH_DES_
CBC_SHA,TLS_RSA_WITH_AES_128_CBC_SHA,TLS_RSA_WITH_AES_256_CBC_SHA
    SSLVerifyClient none
    SSLWallet "${ORACLE_INSTANCE}/config/${COMPONENT_TYPE}/${COMPONENT_NAME}/keystores/default"
    SSLCRLCheck Off
  </IfModule>
  <IfModule mod_weblogic.c>
    SecureProxy on
    WlSSLWallet "${ORACLE_INSTANCE}/config/OHS/ohs1/keystores/default"
  </IfModule>
</VirtualHost>
```

## Creating an EnterpriseOne Cluster

A separate document has been written to discuss clustering the EnterpriseOne HTML server with WebLogic. It is not the intent of this document to duplicate that effort, but rather present a small example to allow us to demonstrate some of the clustering techniques for Oracle HTTP Server.

The general steps are as follows (please refer to the EnterpriseOne WebLogic Clustering document for more detail):

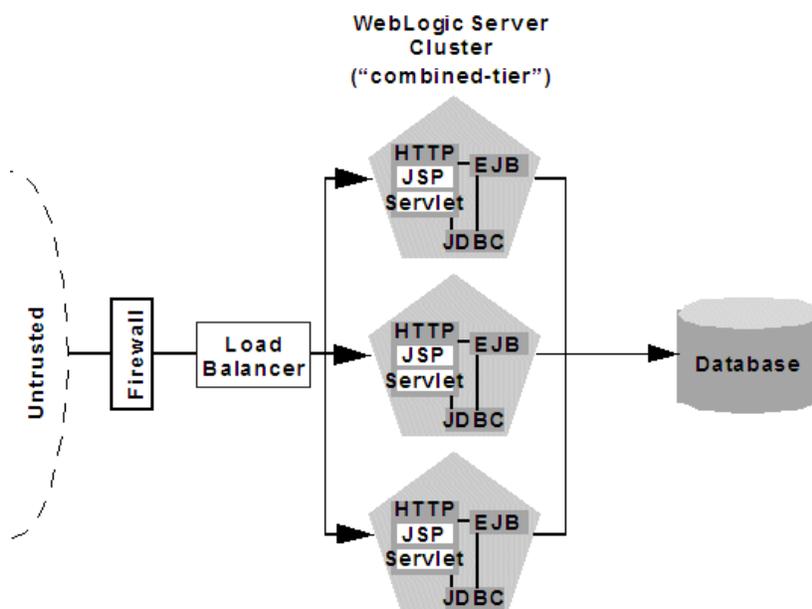
**Note:** An Oracle WebLogic Enterprise license is required for clustering. This is an additional purchase to the standard license.

1. In the WebLogic console application, create a WebLogic cluster and add WebLogic managed servers to the cluster. You can add existing servers to the cluster or create new servers. **Note:** If the EnterpriseOne HTML application is already deployed to a server that you are adding the cluster, you must first undeploy the application and then later re-deploy the EnterpriseOne HTML client after the server has been added.
2. Edit the startup values for the WebLogic managed servers that are in your cluster. In particular, the “-DcloneId” parameter is added with a different value for each server to allow EnterpriseOne to track the runtime metrics for each of these servers independently.

```
-Xms32m -Xmx768m -DcloneId={unique name}
```

3. Deploy the EnterpriseOne HTML client to the cluster. Restart the managed server in the cluster and verify each of the servers independently by signing on to the EnterpriseOne application using each of the specific managed server ports.

You now have a basic clustered architecture that can be represented by the diagram below:



You can use a single Oracle HTTP Server instance as your load balancer by replacing the “WebLogicHost” and “WebLogicPort” parameters in the mod\_wl\_ohs.conf file with the “WebLogicCluster” parameter:

```
<Location /jde>
  SetHandler weblogic-handler
  # WebLogicHost denpbrs1
  # WebLogicPort 8083
  WebLogicCluster denpbrs1:8081,denpbrs1:8082
</Location>
```

## Clustering the HTTP Server

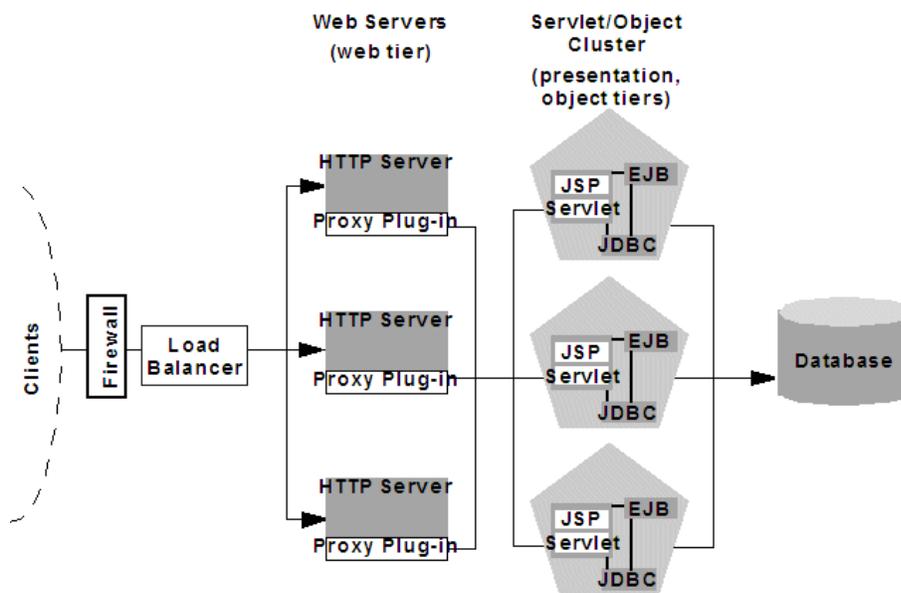
To demonstrate an HTTP server cluster, we will install and configure a second instance of the Oracle HTTP Server on the same Linux machine used above. The installation instructions previously given in this document can be followed, replacing the OHS instance name with “instance2” and the component name with “ohs2”. Once installed, the `mod_wl_ohs.conf` file for the new instance should be edited by adding the following lines, and the instance should be restarted (Note that the default port for this instance is 7779, since we’ve previously used 7777 and 7778):

```
<VirtualHost *:7779>
  <Location /jde>
    SetHandler weblogic-handler
    WebLogicCluster denpbrs1:8081,denpbrs1:8082
  </Location>
</VirtualHost>
```

We now have two separate Oracle HTTP Server instances that can both be used to access the EnterpriseOne HTML application. You can test this by entering either of the following URLs into a browser:

```
http://dnhpb104:7777/jde/owhtml
http://dnhpb104:7779/jde/owhtml
```

At this point we have something similar to the architecture in the diagram below (with two servers instead of three):



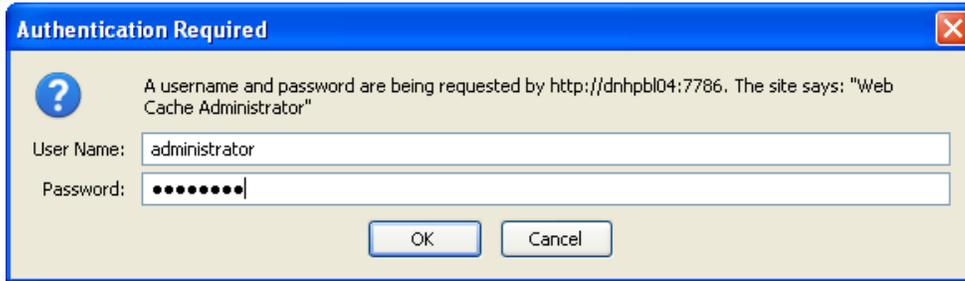
What we need to complete the diagram is a load balancer that will distribute requests among the HTTP server instances. In a production environment you would most likely use a hardware load balancer but for this example we are going to use Oracle Web Cache as our load balancer.

In order to set up Oracle Web Cache as a load balancer we will define both of our HTTP instances (dnhpb104:7777 and dnhpb104:7779) as origin servers. In the example screens below we will be using the Web Cache Manager tool, but the same steps can be performed using the Enterprise Manager application. (More detail related to the following steps can be found in the *Oracle Fusion Middleware Administrator's Guide for Oracle Web Cache 11g Release 1*)

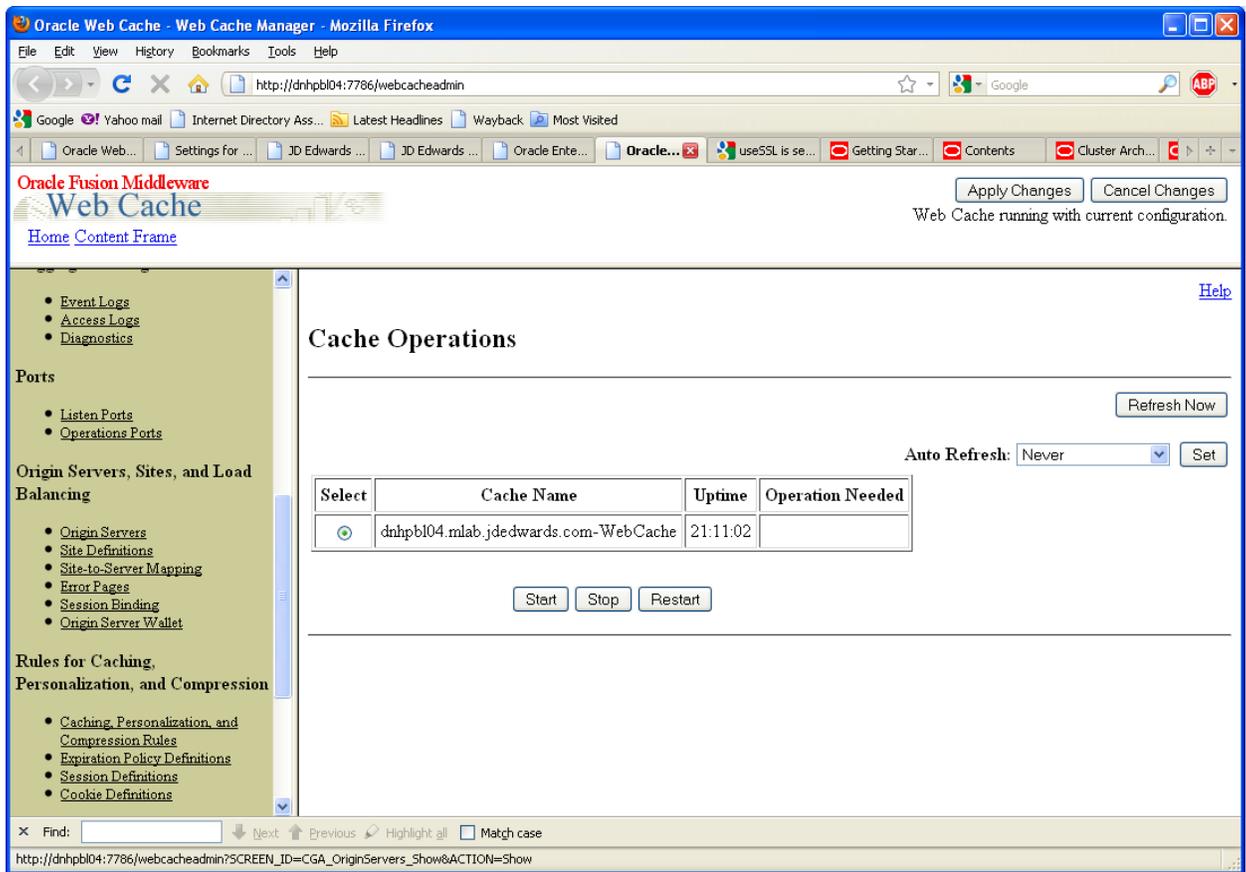
1. Open a browser window and enter the Web Cache Manager URL. You can verify the ports that Web Cache is using by running “`opmnctl status -l`” on the command line, but the default Web Cache administration port is 7786.

```
http://dnhpb104:7786/webcacheadmin
```

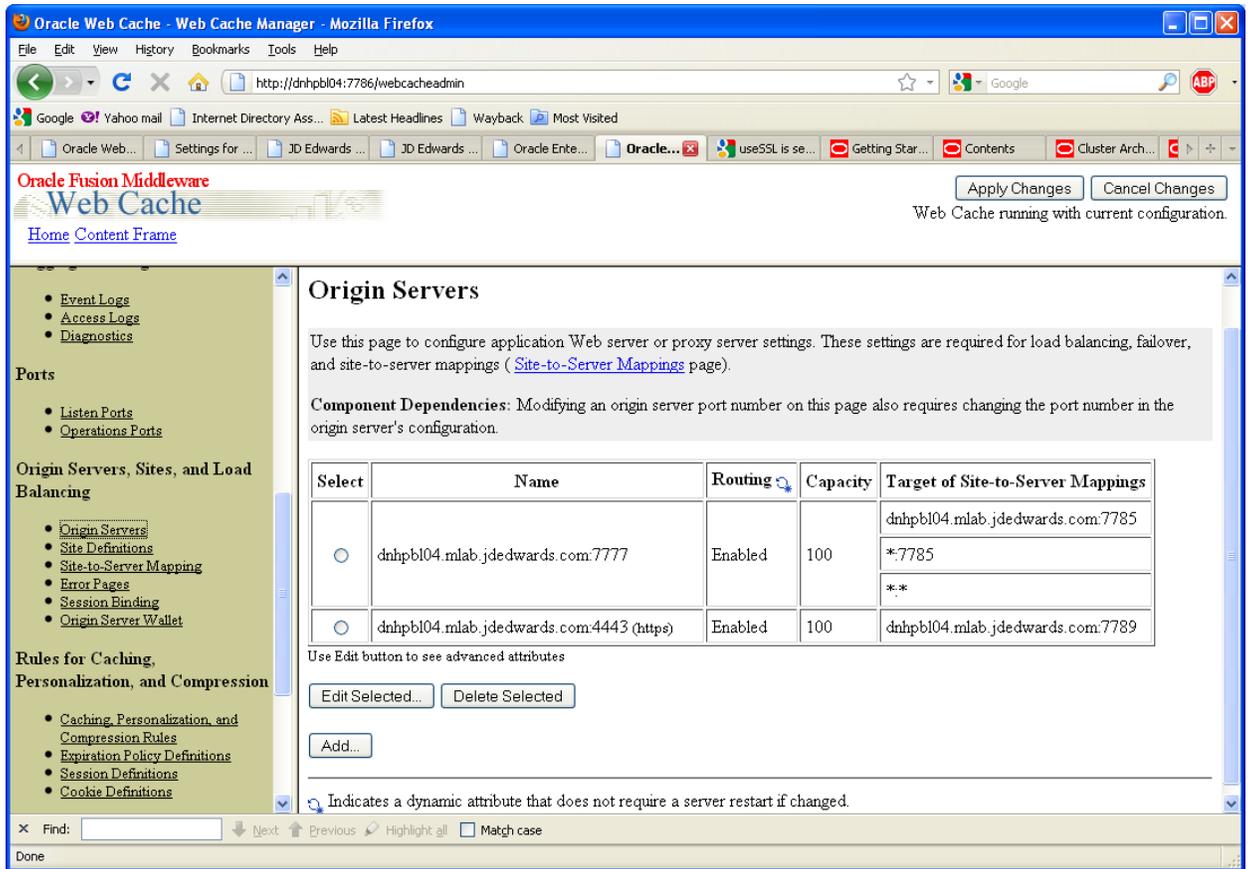
You will be prompted to enter the Web Cache Administrator user and password. The user name is “administrator” and the password is the password you entered when you installed the Oracle Web Cache instance.



2. After logging in you will see the Web Cache administration home page. There is a frame on the left side of the browser window that lists various Web Cache administration functions divided into categories. Use the scroll bar for this frame to scroll down to the section for “Origin Servers, Sites, and Load Balancing”:



3. Select the “Origin Servers” link:



You will see a screen similar to the one above. If you installed Oracle Web Cache at the same time that you installed Oracle HTTP Server, the Oracle HTTP Server default port will already be listed as an origin server in this screen. And the “Site-to-Server Mappings” section of the grid will already have an association of the Oracle HTTP Server default port (7777) to the Oracle Web Cache default port (7785).

4. Click the “Add...” button. You will get a new window that allows you to enter a new origin server:

**Add/Edit Origin Server**

In order for Oracle Web Cache to forward requests to an application Web server, you must map a Web site to the server in the Site-to-Server Mapping page (**Origin Servers, Sites and Load Balancing > Site-to-Server Mapping**).

---

**Identity**

Server Type:  Web Application  Proxy

Host Name:

Port:

Routing:  Enabled  Disabled

---

**Attributes**

Protocol:  HTTP  HTTPS

Capacity:

Failover Threshold:

Ping URL:   
(Examples: /, http://www.company.com:80/index.htm)

Ping Interval (seconds):

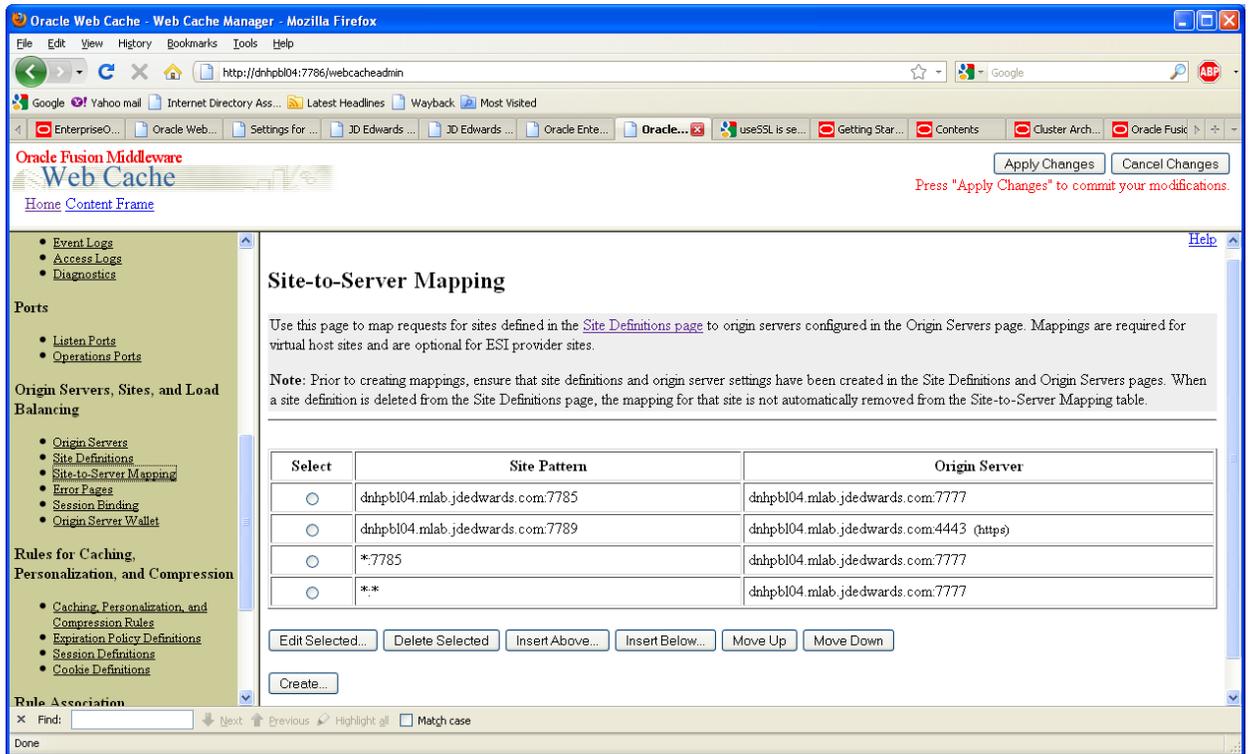
Indicates a dynamic attribute that does not require a server restart if changed.

Done

Enter the name of the server running the second instance of Oracle HTTP Server and the port for that instance. The attributes listed above are default values. Click the “Submit” button after completing this page.

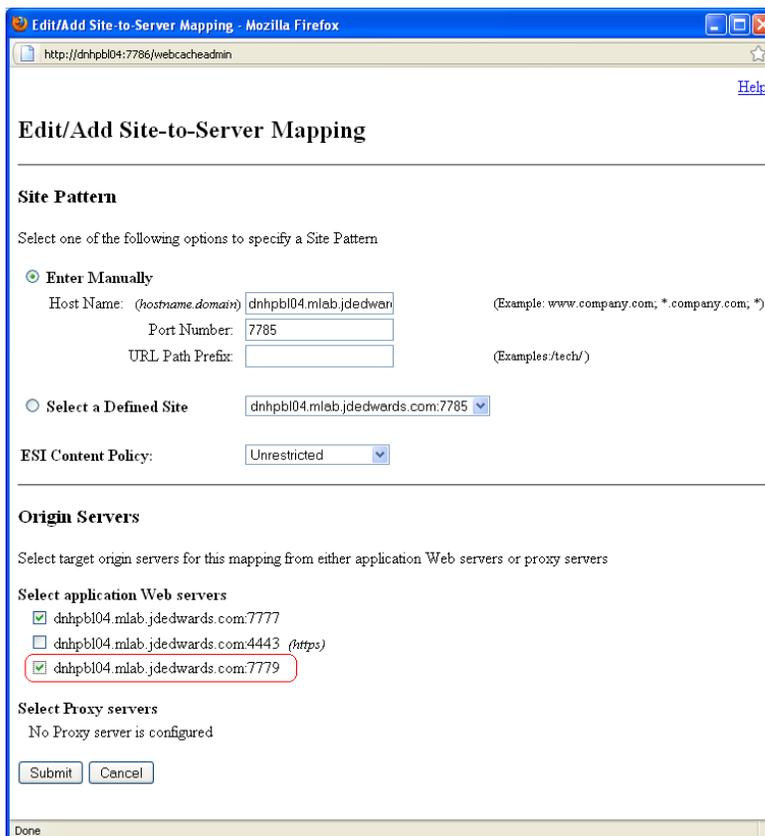
- You will be returned to the “Origin Servers” page and you will see the new entry in the grid. However, there are not yet any “Site-to-Server” mappings listed. “Site-to-Server” mappings are used to tell Web Cache which HTTP URL patterns will be monitored, and which sites those patterns are applicable to.

Select the “Site-to-Server Mapping” link in the navigation pane on the left side of the screen:



You will see the current list of site pattern mappings to origin servers.

- For our example, we want to edit each of the three site patterns that are using the origin server with port 7777. Select each of these in turn and click the "Edit Selected..." button. A new window will be shown that allows you to edit the pattern definition:



Select the second origin server (port 7779) to be included for this pattern. Once you have completed editing the three patterns, your Site-to-Server Mapping list should look something like this:

Select	Site Pattern	Origin Server
<input type="radio"/>	dnhpb104.mlab.jdedwards.com:7785	dnhpb104.mlab.jdedwards.com:7777
		dnhpb104.mlab.jdedwards.com:7779
<input type="radio"/>	dnhpb104.mlab.jdedwards.com:7789	dnhpb104.mlab.jdedwards.com:4443 (https)
<input type="radio"/>	*:7785 (exclude ESI Fragments)	dnhpb104.mlab.jdedwards.com:7777
		dnhpb104.mlab.jdedwards.com:7779
<input type="radio"/>	*:* (exclude ESI Fragments)	dnhpb104.mlab.jdedwards.com:7777
		dnhpb104.mlab.jdedwards.com:7779

- At this point, be sure to click the “Apply Changes” button in the upper right corner of the Web Cache Manager application. After applying changes, you will be returned to the Web Cache administration home page and prompted to restart Web Cache. Click the “Restart” button near the middle of the page.

Once you have completed these steps, you can connect to the EnterpriseOne application by using the following URL:

<http://dnhpb104:7785/jde/E1Menu.maf>

Oracle Web Cache will act as a simple load balancer between the two Oracle HTTP Server instances. Requests from any given client will alternate between the two HTTP servers, but the session will be tracked by using a cookie that is passed back from the EnterpriseOne application. You can test this configuration by logging on to the EnterpriseOne application, and then stopping the second instance of the Oracle HTTP Server. You will see that the session is maintained and that HTTP traffic is now routed to the running HTTP server instance.

## UNDERSTANDING THE DMZ CHAPTER

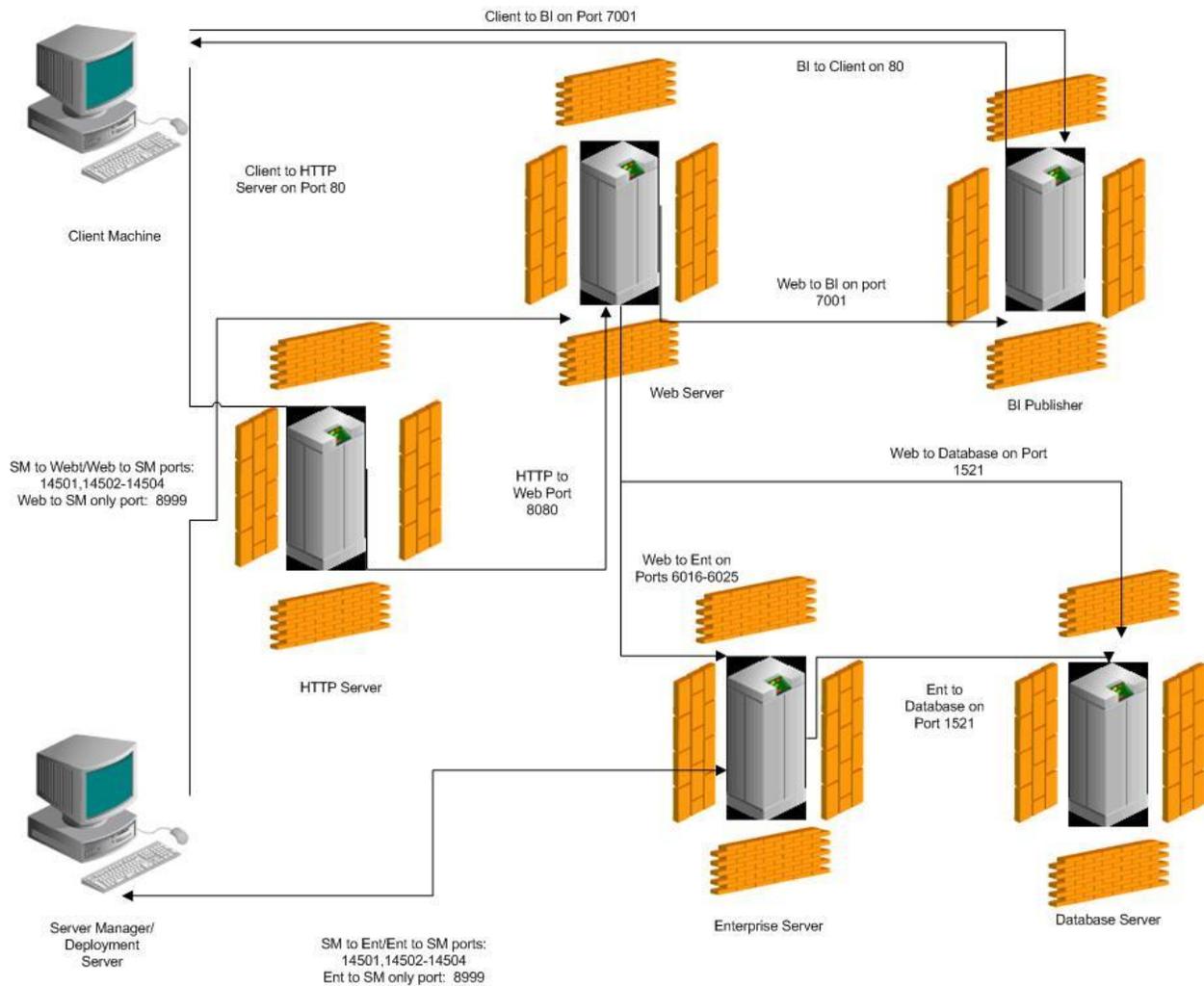
This chapter will cover the set up of a DMZ environment using a hardware firewall. The following components are part of the use-case scenario.

- HTTP Server (Oracle HTTP Server)
- Web Server (Oracle WebLogic Server 10.3.5.0)
- Business Intelligence Publisher (Oracle BI Publisher 11.1.1.5)
- Oracle JD Edwards Deployment Server/Server Manager Console (Tools Release 9.1.0.3, Apps Release 9.1.0)
- Oracle JD Edwards EnterpriseOne Enterprise Server (Tools Release 9.1.0.3, Apps Release 9.1.0)
- Database Server (Oracle Database 11.2.0.1.0)

This chapter assumes that the software components listed above have been installed and are already working properly. Also, knowledge of a hardware firewall and the use of its admin console is also assumed. This chapter does not cover the exact specifics of how to set up zones, interfaces, and policies for a specific brand of firewall.

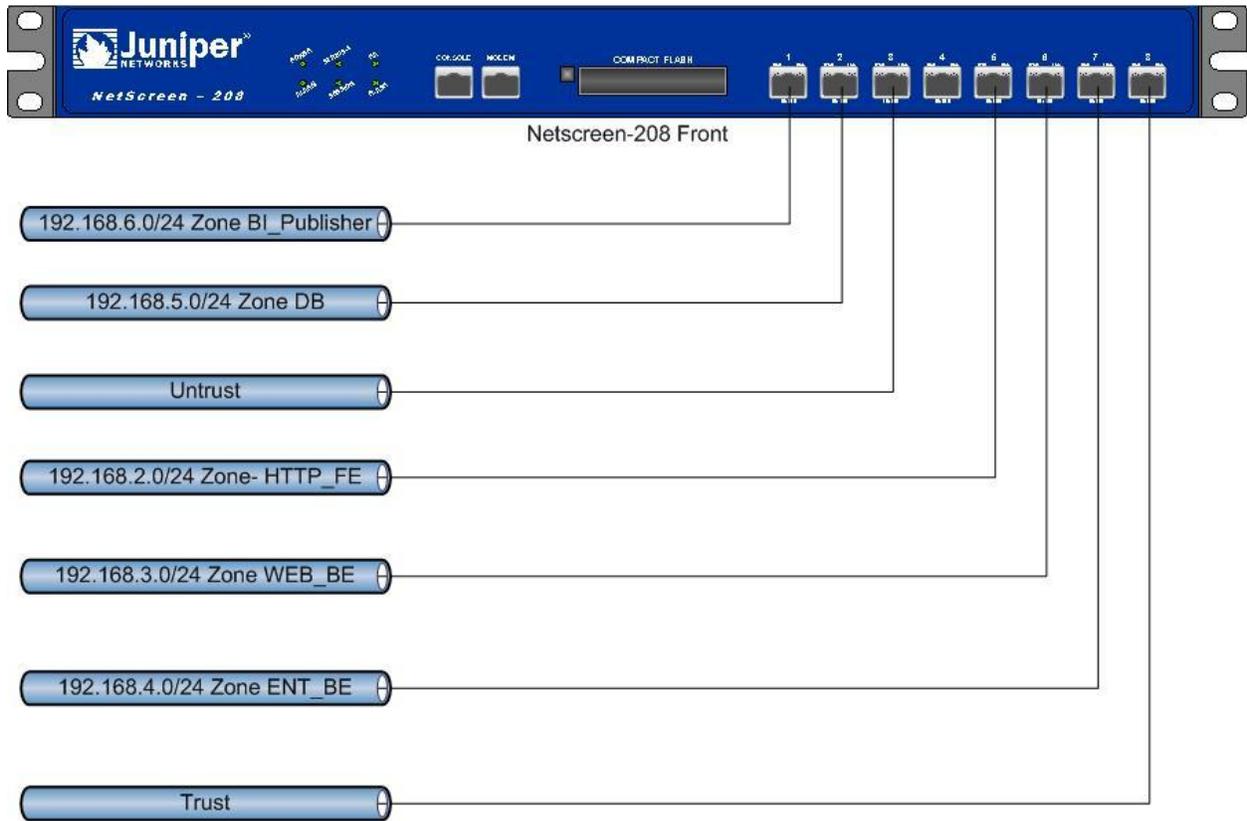
### Illustrating the Architecture

The following figure gives a high-level overview of the architecture implemented in this chapter. Each node is surrounded by a firewall (due to being physically attached to a hardware firewall) with only certain ports being open to certain traffic. In some cases only one-way traffic is allowed, in others two-way traffic is allowed. Refer to the illustration below and the sample policies further in this chapter for more information.



### Depicting the Physical Setup

The following figure shows how the firewall was set up. Each hardware server was connected to a port in the firewall. This effectively closed all ports on each server to incoming and outgoing traffic. The necessary ports and services were then activated through policies set up in the firewall's administrative console.



**Network Descriptions**

192.168.2.0/24 This is the HTTP server Front End Customer facing on port 80  
 192.168.3.0/24 This is the Web server Backend isolated by policies  
 192.168.4.0/24 Enterprise server Backend isolated by policies  
 192.168.5.0/24 This is the Database zone isolated by policies  
 192.168.6.0/24 This is the Business Intelligence Publisher zone isolated by policies

Note: The Server Manager Console machine/deployment server and all client machines lie outside of the firewall.

### Configuring the Policies

Once the machines were connected to the firewall, each was separated from the network and unable to communicate with one another or any machines outside of the firewall. Policies were created to allow the necessary nodes to connect to one another. Below is a chart that lists the policies set up. The policies are categorized by source and destination zones. The exact steps required for setting up zones and policies will differ depending on the firewall you are using.

Source Machine	Destination Machine	Service/Protocol	Source Ports	Destination Ports
Zone: From Untrust to http_fe				
Any	MIP_HTTP_FE	TCP	0-65535	80
Zone: From http_fe To web_be				
192.168.2.200/32	192.168.3.200/32	TCP	0-65535	8080-8080
Zone: From Untrust To web_be				
Any	MIP_WEB_BE	TCP	0-65535	14501-14501
		TCP	0-65535	14502-14504

Zone: From Untrust To ent_be				
Any	MIP_ENT_BE	TCP TCP	0-65535 0-65535	14501-14501 14502-14504
Zone: From web_be To ent_be				
192.168.3.200/32	192.168.4.200/32	TCP UDP	0-65535 0-65535	6016-6025 6016-6025
Zone: From web-be To BI-Publisher				
192.168.3.200/32	192.168.6.200/32	TCP	0-65535	7001-7001
Zone: From ent_be To Untrust				
192.168.4.200/32	Any	TCP TCP	0-65535 0-65535	14501-14501 14502-14504
192.168.4.200/32	Any	TCP	0-65535	8999-8999
Zone: From BI-Publisher to Untrust				
192.168.6.200/32	Any	TCP	0-65535	7001-7001
MIP_BIP	Any	TCP	0-65535	7001-7001
192.168.6.200/32	Any	TCP	0-65535	80
Zone: From db to BI-Publisher				
192.168.5.200/32	192.168.6.200/32	TCP UDP	0-65535 0-65535	1521-1521 1521-1521
Zone: From web_be to Untrust				
192.168.3.200/32	Any	TCP TCP	0-65535 0-65535	14501-14501 14502-14504
192.168.3.200/32	Any	TCP	0-65535	8999-8999
Zone: From BI-Publisher To db				
192.168.6.200/32	192.168.5.200/32	TCP UDP	0-65535 0-65535	1521-1521 1521-1521
Zone: From Untrust to BI-Publisher				
Any	MIP_BIP	TCP	0-65535	7001-7001
Zone: From ent_be To db				
192.168.4.200/32	192.168.5.200/32	TCP UDP	0-65535 0-65535	1521-1521 1521-1521

### Explaining Mapped IP Addresses

The table below explains some of the phrases and terminology used in the table above.

Value	Explanation
-------	-------------

MIP_HTTP_FE	The mapped IP address of the HTTP server. (10.x.x.x)
MIP_WEB_BE	The mapped IP address of the WebLogic server (10.x.x.x)
MIP_ENT_BE	The mapped IP address of the enterprise server (10.x.x.x)
MIP_BIP	The mapped IP address of the Business Intelligence Publisher server (10.x.x.x)

### **Using Mapped IP Addresses**

Mapped IP addresses are used to give a public IP address to an IP address that only exists within the firewall. There are two reasons why we use mapped IP addresses in this scenario.

- The Server Manager Console/deployment server machine (which resides outside the firewall) must connect to both the WebLogic server and Oracle JD Edwards EnterpriseOne enterprise server to manage them.
- Clients outside of the firewall are using One View Reporting and therefore must connect to the Business Intelligence Publisher.

### **Explaining Open Ports**

The following section gives an explanation of why certain ports are open to allow traffic from certain nodes.

#### **Ports 14501, 14502-14504, and 8999**

These are the server manager ports. The port of 14501 is the default port used by the Server Manager Console to initially connect to each Server Manager agent on startup. It can be changed from the server manager console. After startup, the ports 14502-14504 are then used by the server manager agent to connect to the console. Each agent uses only one port, but the exact port is reassigned on the startup of the agent (and is issued in order of free ports). The beginning port range (in this case 14502) can also be changed in the Server Manager Console.

Port 8999 is open because Server Manager uses this port to transfer log information back to the Server Manager Console and the client machine. If you wish to review open log files from a specific server through the Server Manager Console, both the console machine and client must be able to contact the server using port 8999. This port is set during the installation of the Server Manager and can only be changed by reinstalling the Server Manager Console.

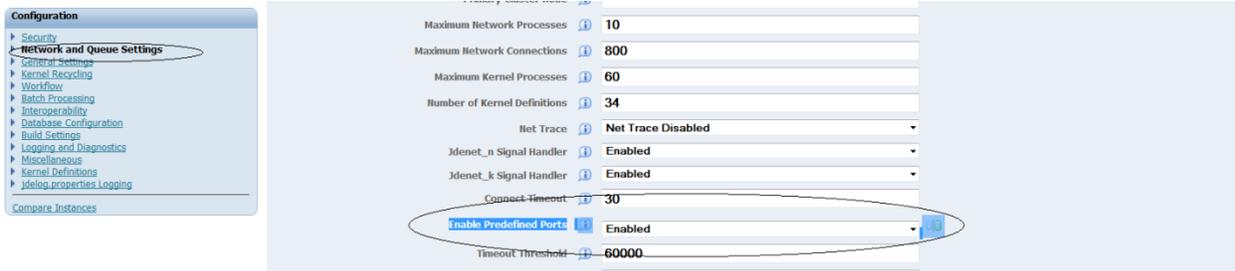
#### **Port 7001**

This port is required by the Business Intelligence Publisher. If you use the BI Publisher installation available from the edelivery web site, you must use this port and it cannot be changed. Both the web server and the each client machine that will use One View Reporting must be able to communicate with the BI Publisher machine on port 7001.

#### **Ports 6016-6025**

These ports are used by the enterprise server to manage various kernels. The ServiceNameListen parameter accessed through Server Manager (or the JDE.INI) provides the lowest end port used by the enterprise server (6016 is the default for Oracle JD Edwards EnterpriseOne Applications Release 9.1) while the higher end can be anywhere within 1000 port numbers of the low end. However this behavior can be modified by changing the setting below in Server Manager for the enterprise server managed instance.

- EnabledPredefinedPorts=Enabled



When EnablePredefinedPorts is set to “Enabled”, the enterprise server code will only use the port ranges equal to ServiceNameListen + (Maximum Network Processes – 1). For a server using default values, this means, ports 6016-6025.

### Configuring the Host File

Due to the configuration of this scenario, each server’s host file must be modified to point to each node in the architecture. In addition, the host file of each client using One View Reporting must include the IP and host name of the BI Publisher machine (this is so the One View Reports can be published to the client).

### Tracing out Ports

Some processes in EnterpriseOne may use ports that aren’t readily obvious. If you have set up your firewall, but EnterpriseOne processes are failing then you can trace which ports are trying to be used. A policy for tracing would look something like this.

Source Machine	Destination Machine	Service/Protocol	Source Ports	Destination Ports	Trace
ANY	192.168.3.200/32 (this will change depending on the machine you are trying to trace out.)	ANY	0-65535	0-65535	ON

In this case, you allow all traffic to the destination machine but you turn on tracing. Repeat the process that is failing or timing out, then review the log (for most firewalls, the trace can be viewed directly from the firewall’s administrative console). After reviewing the log, remove the policy that allows all source machines and protocols and replace it with a policy that only allows the necessary traffic.

## APPENDIX A: CONFIGURING ADDITIONAL TECHNOLOGIES WITH A DMZ

This section gives guidelines on setting up additional technologies with a DMZ and Oracle JD Edwards EnterpriseOne.

### Setting Up Mobile Applications

In order to use the Oracle JD Edwards EnterpriseOne mobile applications, you must add an ADF server to your architecture. The ADF server can be plugged into the hardware firewall, just like the other machines. To use the Oracle JD Edwards EnterpriseOne mobile applications in a DMZ environment, the following additional connections must be allowed:

- Mobile devices must be able to connect to the ADF server.
- The ADF Server must be able to communicate to the business services server.
- The business services server must connect to the database and enterprise server (see the “Setting Up Business Services Server” section below).

Source Machine	Destination Machine	Service/Protocol	Source Ports	Destination Ports
ANY	MIP_ADF	TCP	0-65535	7005
ADF_IP	BSSV_IP	TCP UDP	0-65535 0-65535	6016-6025 6016-6025
BSSV_IP	DBASE_IP	TCP UDP	0-65535 0-65535	1521-1521 1521-1521

### Setting Up Business Services Server

The Business Services Server (BSSV) technology can be used with an HTTP server. The *JD Edwards EnterpriseOne Tools Release 8.98.4 Business Services Server Reference Guide* provides instructions on how to do this. Once the HTTP Server and BSSV have been set up, additional policies need to be set up to allow the BSSV and HTTP Server to communicate. The port needed for BSSV server can be found in a variety of ways.

- Select the BSSV instance in Server Manager, the port number used will be in the upper right hand side.
- Through the WebLogic Administrative Console, view the port number of the server the BSSV deployment is deployed to.

The following additional connections must be allowed in order to use Business Services Server with a DMZ environment:

- The HTTP server must be able to communicate to the business services server.
- The business services server must communicate to the database server.
- The business services server must communicate with the enterprise server.

Source Machine	Destination Machine	Service/Protocol	Source Ports	Destination Ports
HTTP_IP	BSSV_IP	TCP	0-65535	7003

BSSV_IP	ENT_IP	TCP UDP	0-65535 0-65535	6016-6025 6016-6025
BSSV_IP	DBASE_IP	TCP UDP	0-65535 0-65535	1521-1521 1521-1521

The following key explains some of the notation used.

- HTTP\_IP: The IP address of the HTTP server.
- BSSV\_IP: The IP address of the business services server.
- ENT\_IP: The IP address of the enterprise server.
- DBASE\_IP: The IP address of the database server.
- Destination Ports
  - 7003: This document assumes the BSSV server is listening on port 7003. If it is installed to a different port then this policy will need to be changed accordingly.
  - 6016 - 6025: This document assumes the Enterprise Server has the ServiceNameListen parameter set to 6016 and the EnablePredefinedPorts parameter enabled. If it is listening on a different port then the policy will need to be changed.
  - 1521: This document assumes the oracle database is using port 1521. If not, the policy should be changed appropriately.

### Setting Up Transaction Server

The transaction server solution currently does not support an http proxy. External client programs trying to use the transaction server must connect directly to the WebLogic server the transaction server software is deployed on. The HTML server must connect directly to the transaction server. The following additional connections must be allowed.

- All external client programs submitting events must be allowed to connect to the transaction server.
- The HTML server needs to connect to the business services server.
- The business services server needs to connect to the enterprise server.

• Source Machine	• Destination Machine	• Service/Protocol	• Source Ports	• Destination Ports
• CLIENT_MIP	• TS_IP	• TCP	• 0-65535	• 7005
• WEB_IP	• TS_IP	• TCP	• 0-65535	• 7005
• TS_IP	• ENT_IP	• TCP UDP	• 0-65535 0-65535	• 6016-6025 6016-6025
• TS_IP	• DBASE_IP	• TCP UDP	• 0-65535 0-	• 1521-1521 1521-1521

			65535	
--	--	--	-------	--

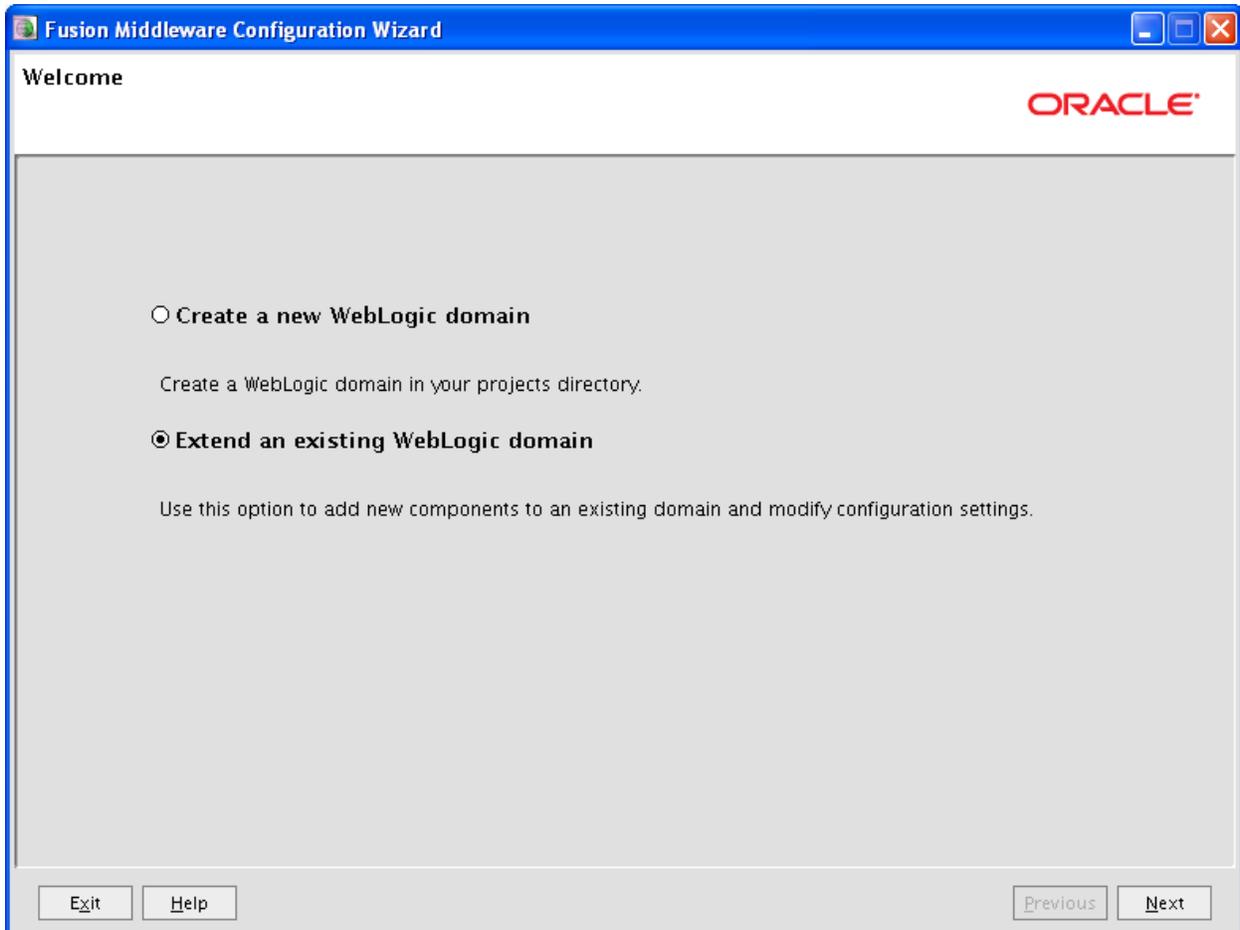
The following key explains some of the notation used.

- CLIENT\_MIP: The mapped IP address of the physical client trying to run an external transaction server client software. Since it is assumed the clients reside outside of the hardware firewall, then each machine trying to run an external transaction server client will need their own mapped IP addresses and policies. An alternative is to allow ANY source machine to connect to the transaction server on port 7005, but this presents a security vulnerability.
- WEB\_IP: The IP address of the WebLogic server that the HTML client resides on. If the HTML client instance and the transaction server instance reside on the same physical machine, this rule is not necessary.
- TS\_IP: The IP address of the transaction server.
- ENT\_IP: The IP address of the enterprise server.
- DBASE\_IP: The IP address of the database server.
- Destination Ports
  - 7005: This document assumes the transaction server is listening on port 7005. If it is installed to a different port then this policy will need to be changed accordingly.
  - 6016 - 6025: This document assumes the Enterprise Server has the ServiceNameListen parameter set to 6016 and the EnablePredefinedPorts parameter enabled. If it is listening on a different port then the policy will need to be changed.
  - 1521: This document assumes the oracle database is using port 1521. If not, the policy should be changed appropriately.

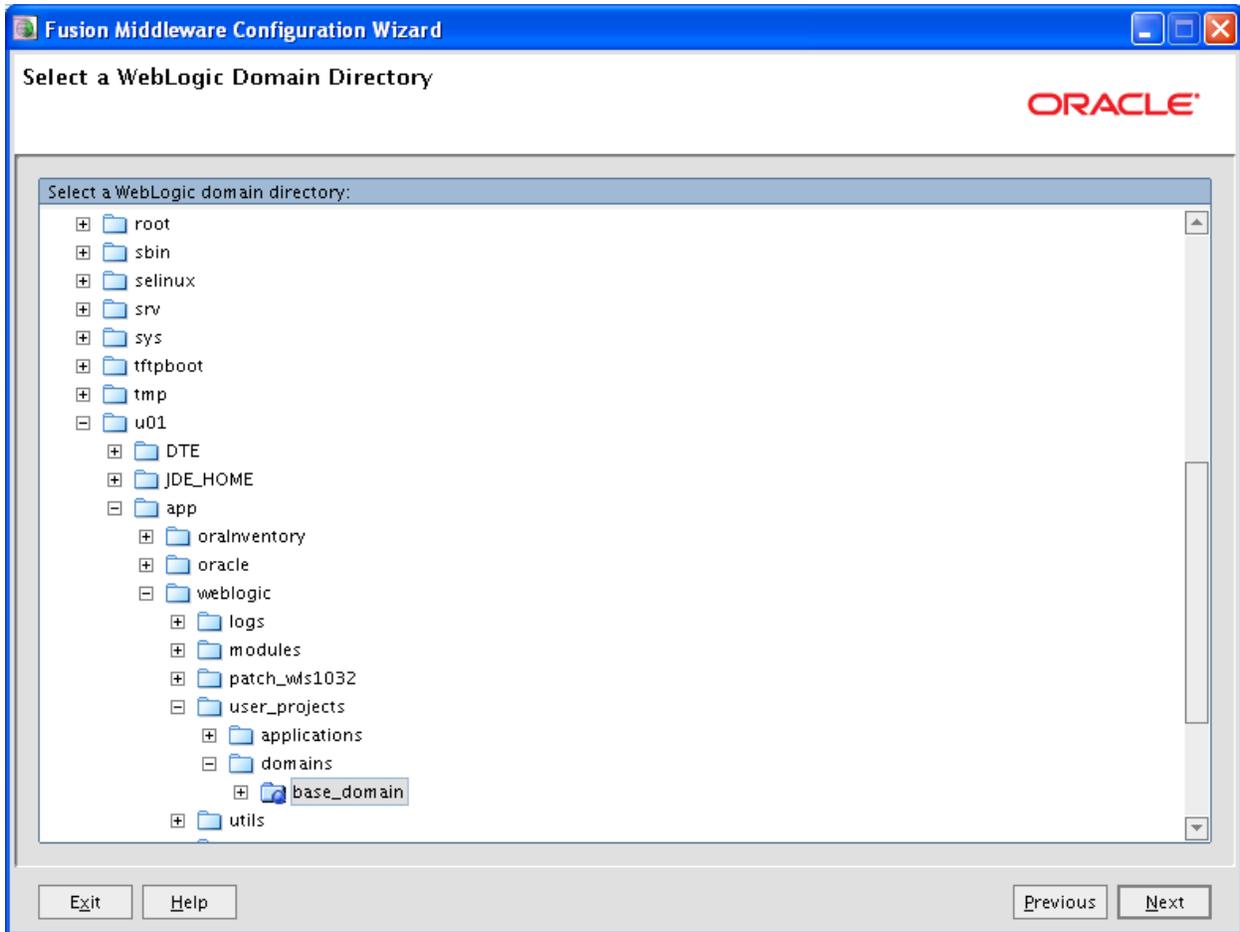
## APPENDIX B: INSTALLING FUSION MIDDLEWARE CONTROL EM

In order to associate the Web Tier components with a WebLogic domain, you must have a previously installed WebLogic Server. In order to add the Fusion Middleware Control Enterprise Manager application, you will also have to install additional components. The Enterprise Manager application is available if you install the WebCenter Suite and also if you install the SOA Suite. In order to get Enterprise Manager by itself, you can install the Application Development Runtime package available from the Oracle Fusion Middleware download site.

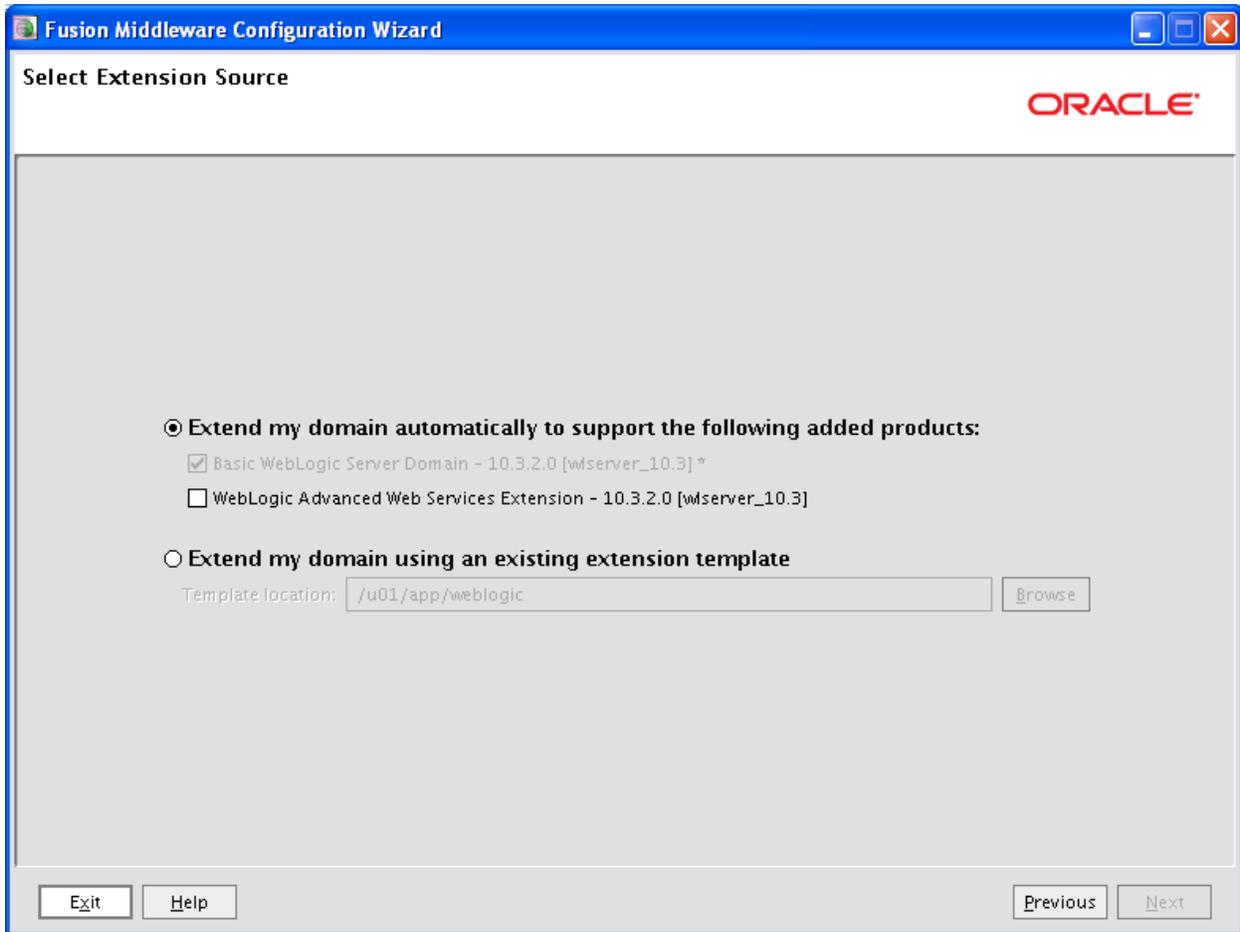
To determine if you have Enterprise Manager available in your current WebLogic installation, run the “`config.sh`” script found in the “`wlserver_10.3/common/bin`” directory under the WebLogic install location. The following screen is displayed:



You can select either of these options, but for this example we've selected "Extend and existing WebLogic Domain". The next screen allows you to choose from any existing WebLogic domains that you have created.



Select a domain and click "Next".

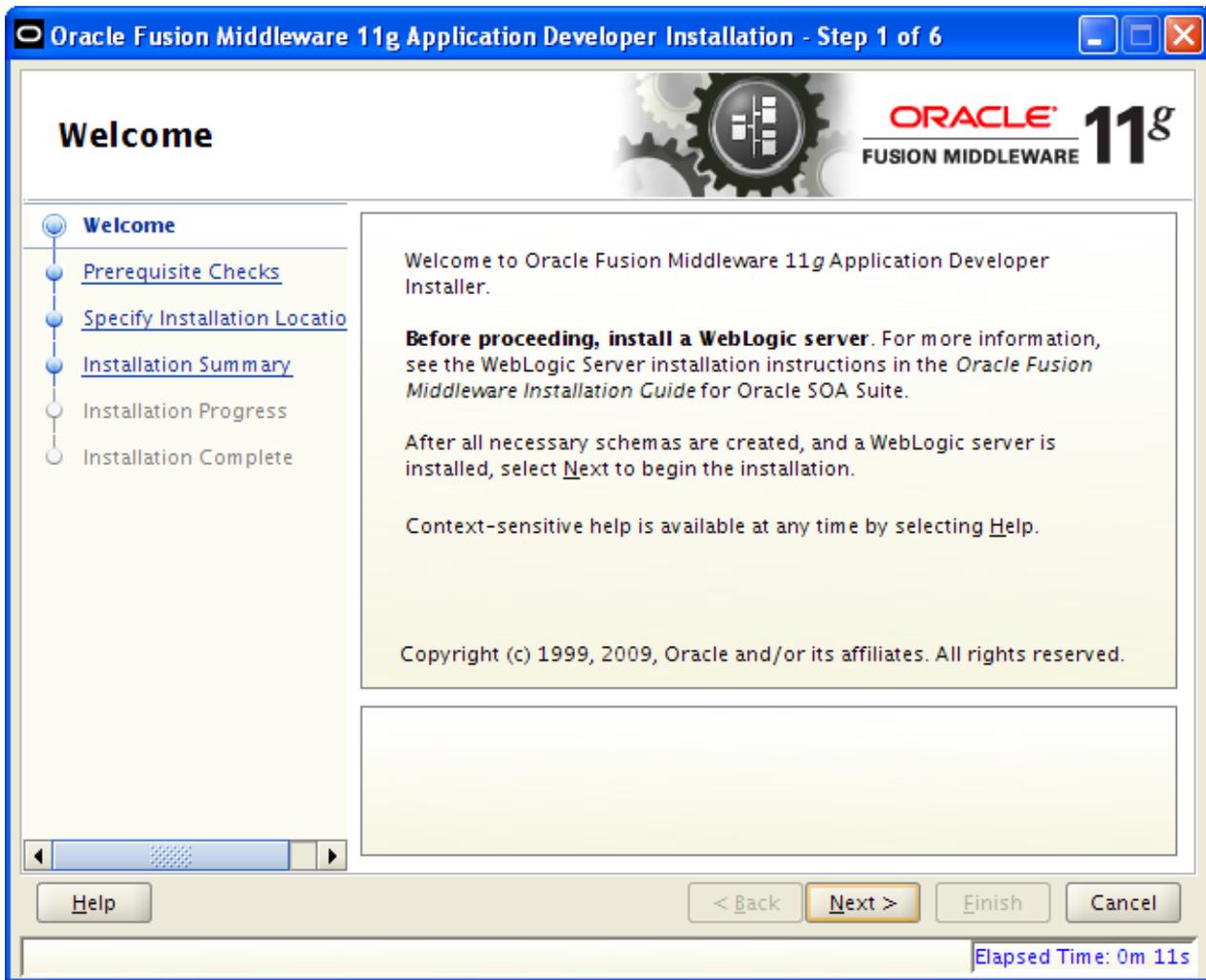


The screen above is what you would see after a typical, basic WebLogic server installation. This installation *does not* have Enterprise Manager available at this point.

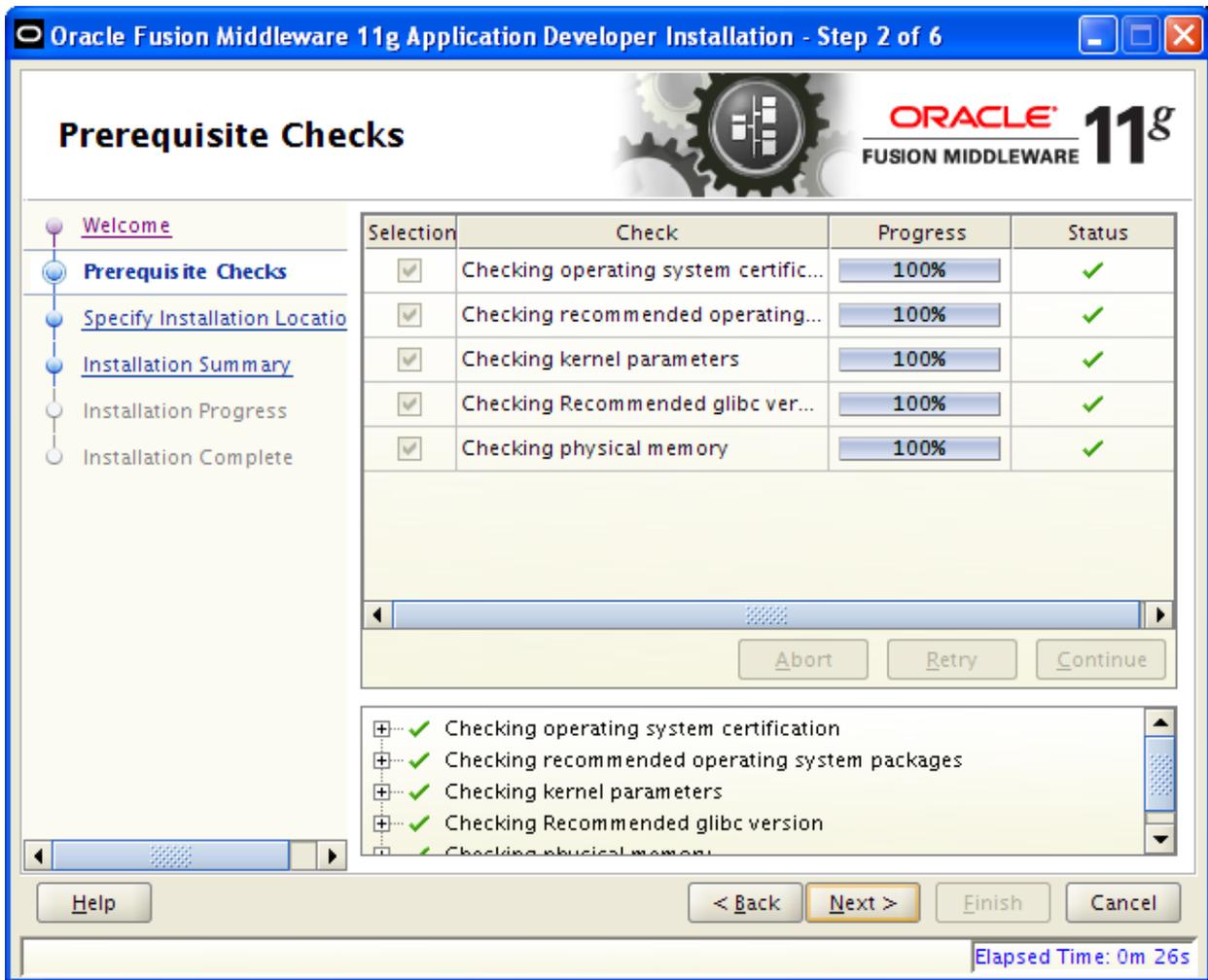
Download the “Application Development Runtime” package from the Oracle Fusion Middleware download site. This file will be called “`ofm_appdev_generic_11.1.1.2.0_disk1_1of1.zip`”, or something similar.

Unzip the downloaded file into a temporary directory.

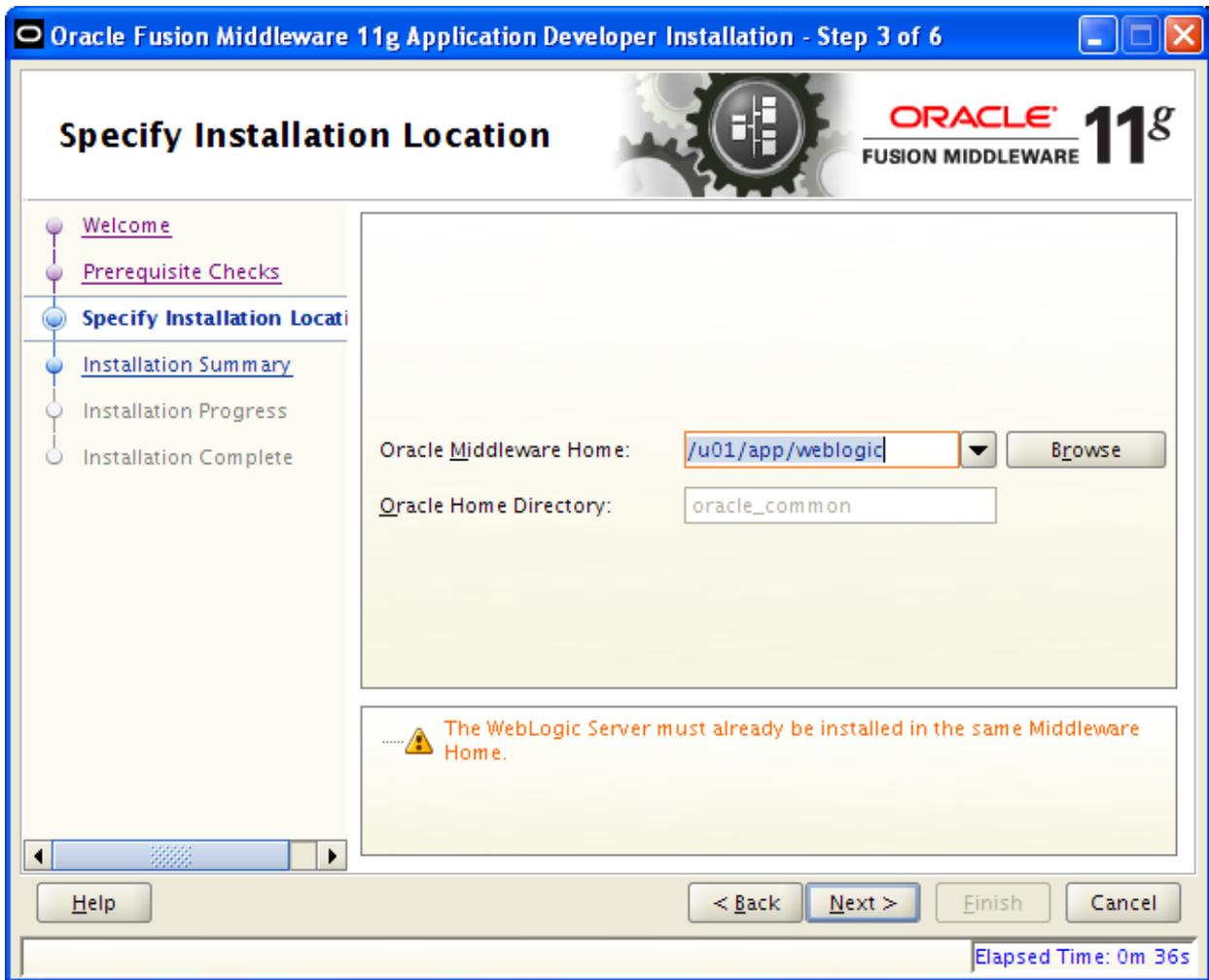
Change to the “`disk1`” directory and run the “`runInstaller`” script. You will need to provide the location of a previously installed JDK or JRE. Best is to provide the location of the JDK that is being used by your WebLogic installation.



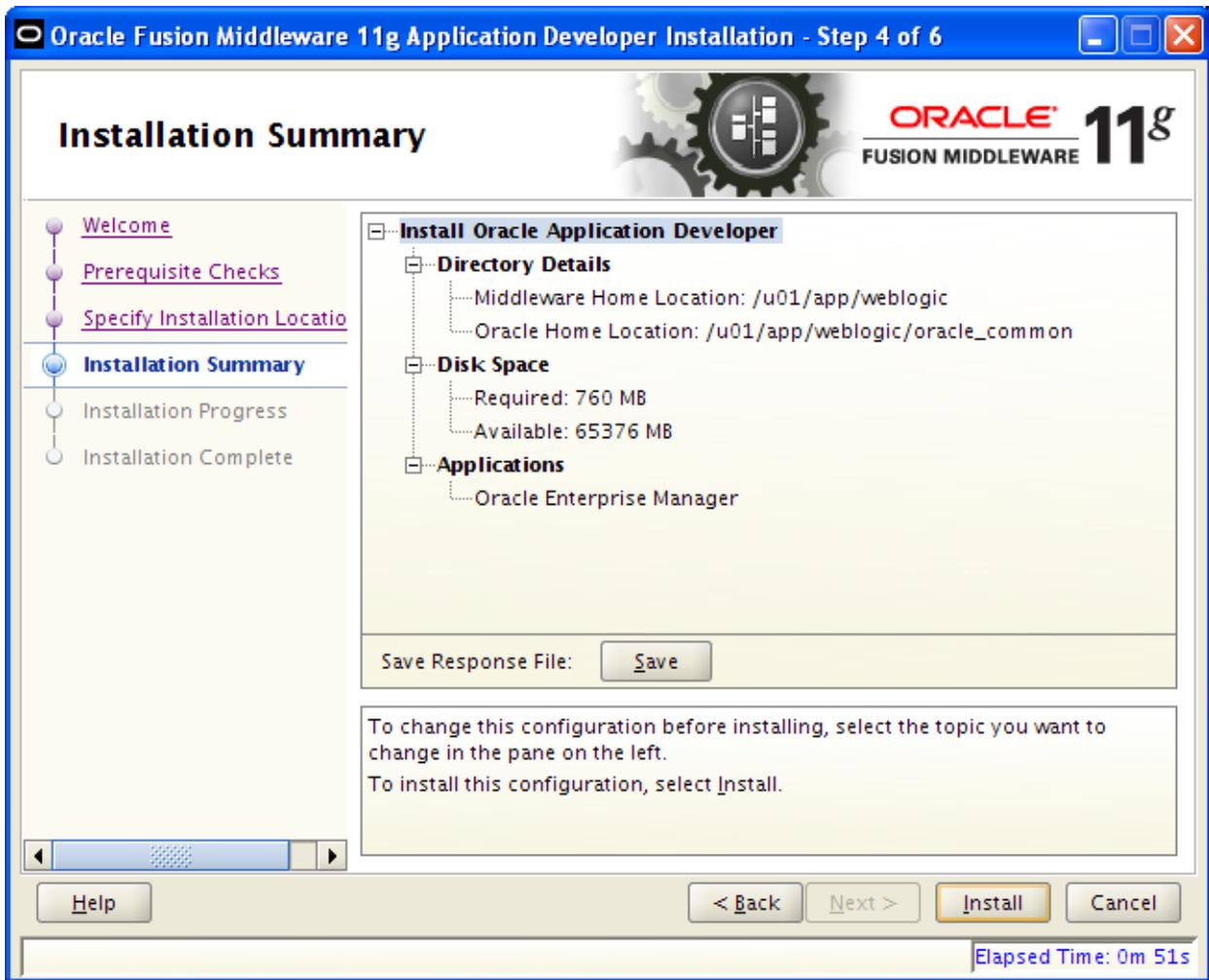
Click "Next" from the welcome screen. As indicated, you must have previously installed a WebLogic server before installing this product.



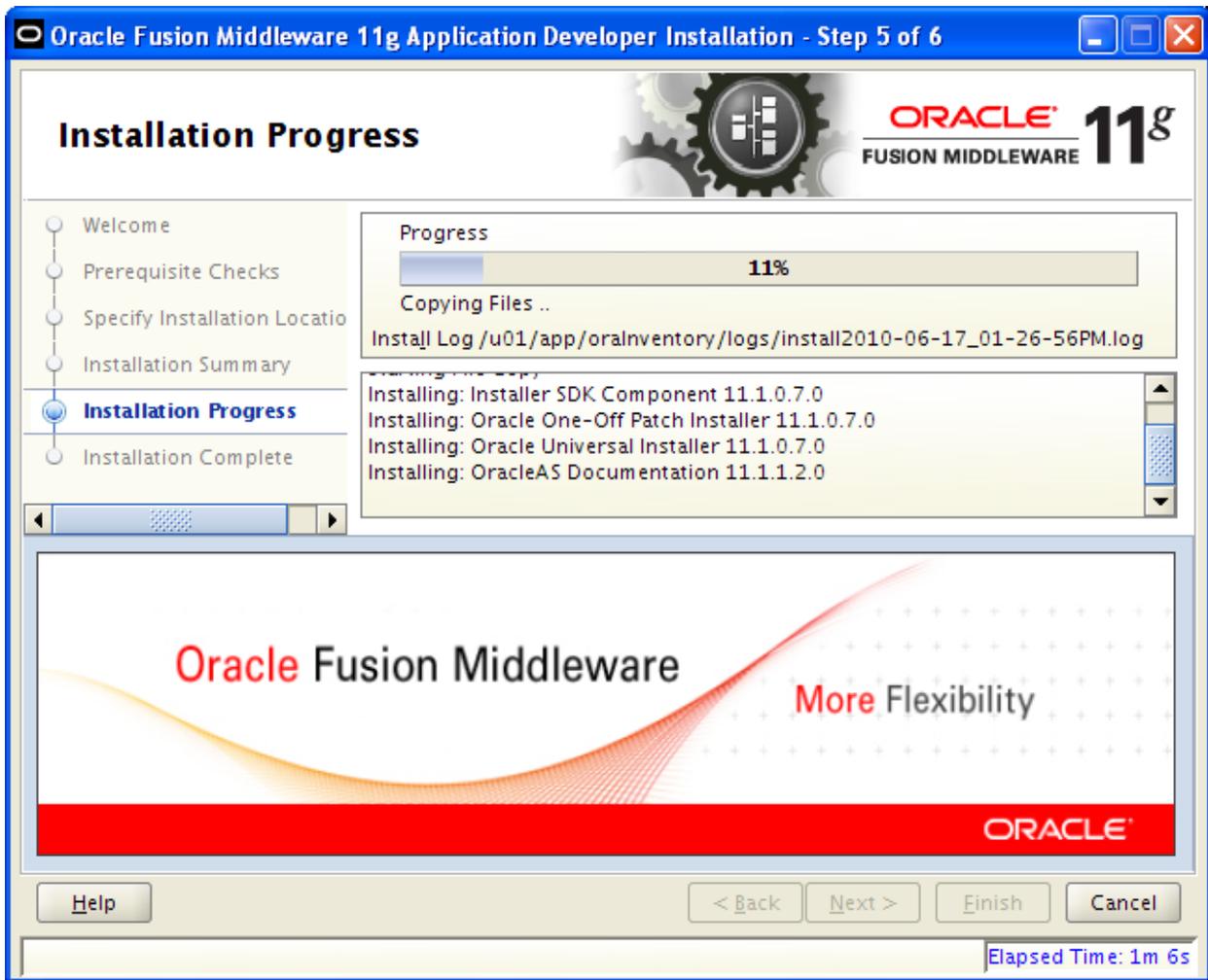
Click "Next" after completing the prerequisite checks.



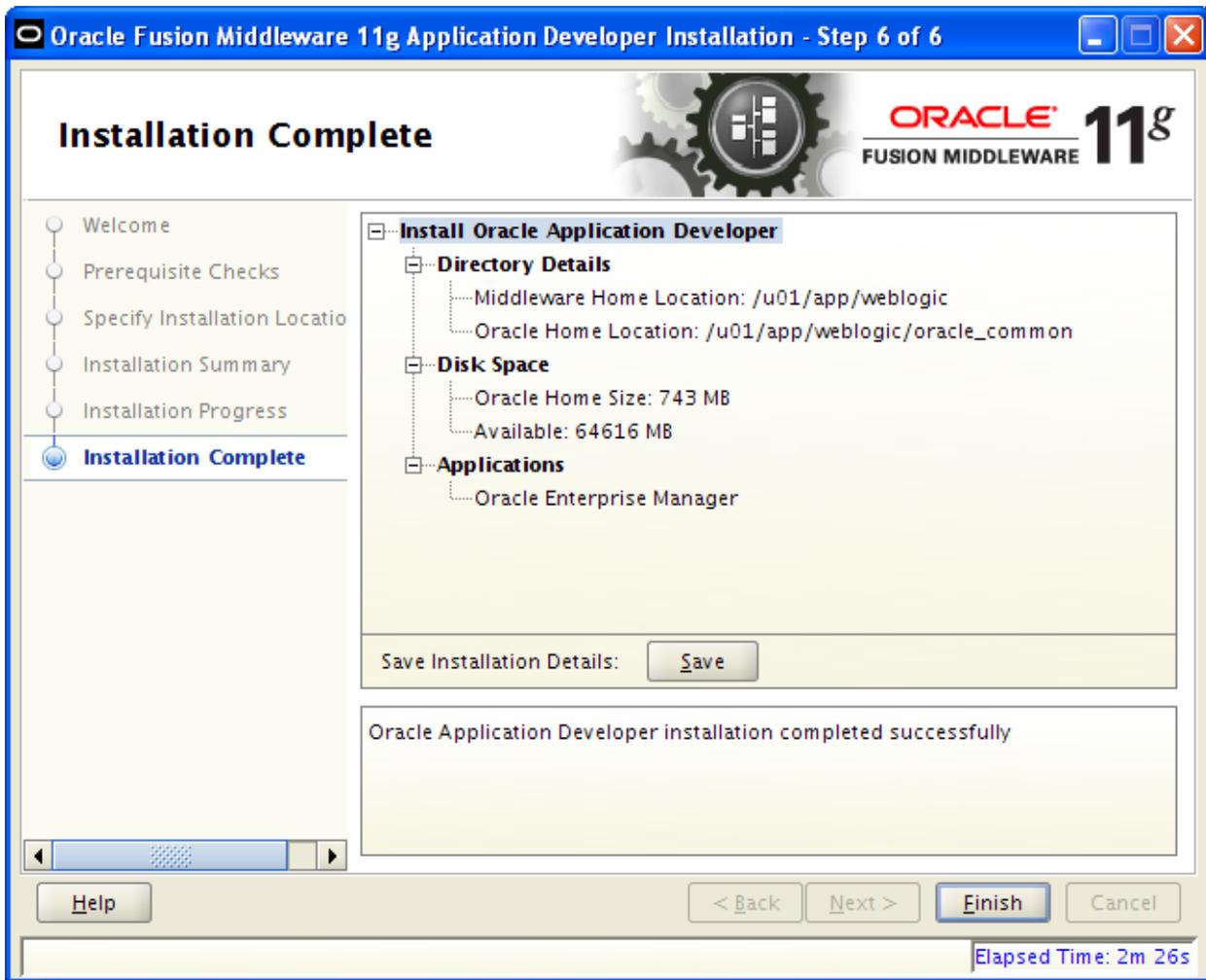
The Oracle Middleware Home directory that you previously used to install the WebLogic server should be displayed. If it is not, change this to the WebLogic server's install location. Click "Next".



Click "Next" at the installation summary confirmation screen.

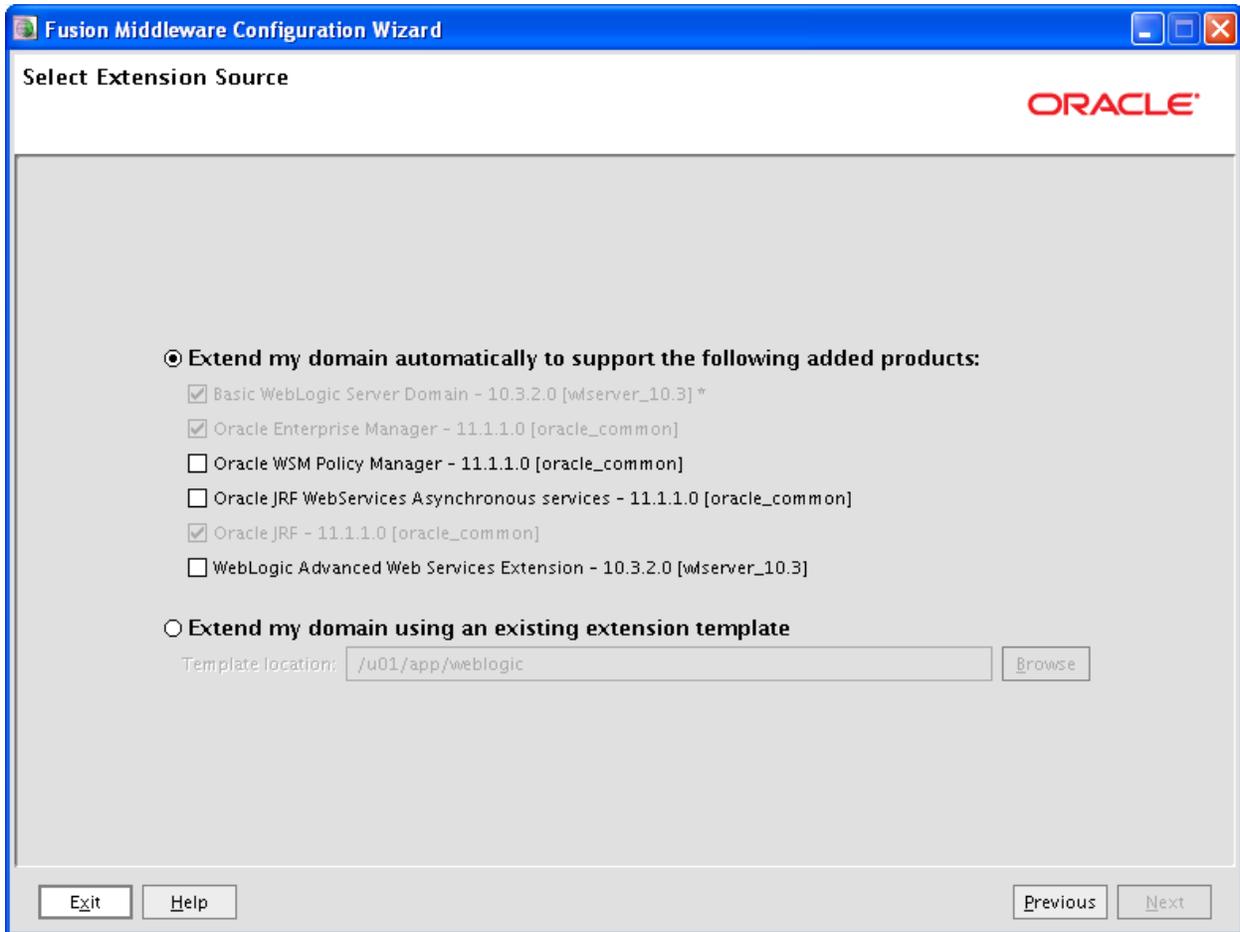


Installation progress is shown.



Click "Finish" when the installation completes.

After exiting the installer, go back and run the "config.sh" script from the WebLogic server's "common/bin" directory again. After selecting an existing domain you will see the following screen:



If you had an existing domain, you will see that Enterprise Manager has been installed. If you create a new domain, you will now have the option of installing the Enterprise Manager application to the domain. If your domain is not running, you can start the Enterprise Manager application by starting the WebLogic administration server with the “startWebLogic.sh” script in the domain home directory. You can now access the Enterprise Manager console by typing the following URL into a browser:

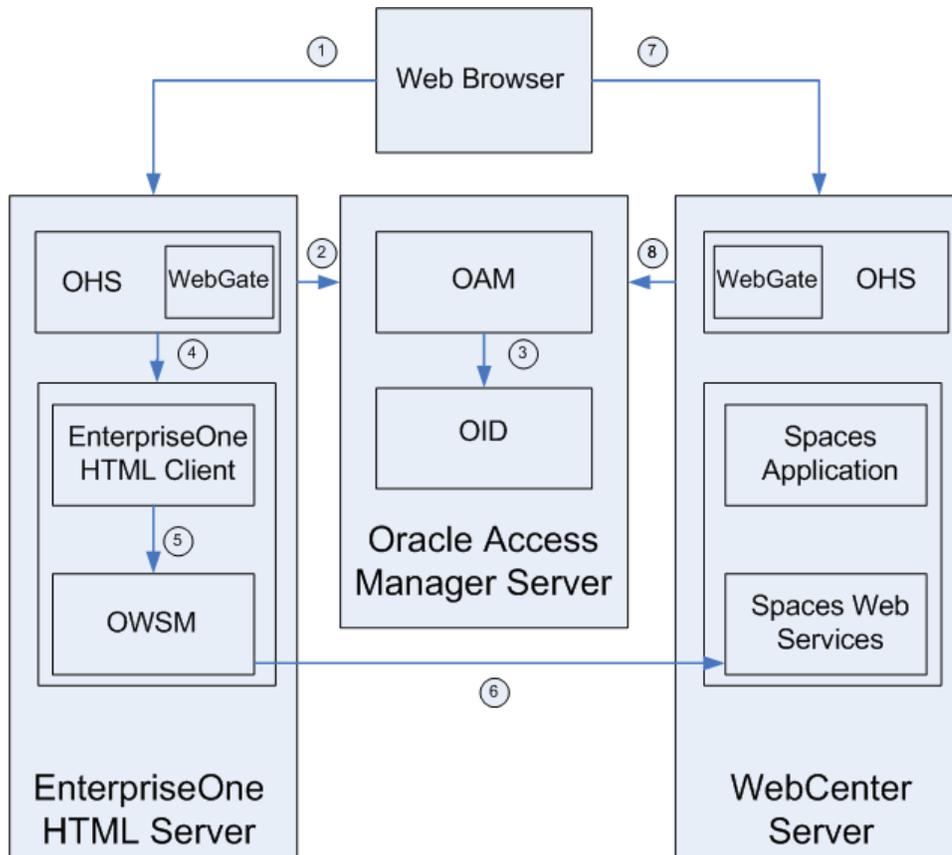
**http://{host}:{port}/em**

Where “{host}” is the machine where you installed WebLogic, and “{port}” is the WebLogic administration port that was used (7001 is the default).

## APPENDIX C: SETTING UP SINGLE SIGN-ON (SSO)

Setting up single sign-on for EnterpriseOne is a complex process that involves several Oracle Fusion Middleware components. A detailed review of this process is beyond the scope of this document, but since Oracle HTTP Server is an integral component of this setup we will provide an overview here. For a detailed example, refer to the document titled *JD Edwards EnterpriseOne WebCenter Spaces Configuration for Related Information Application Framework Guide Release 9.1*.

A sample architecture is shown below. This example is from the document that discusses WebCenter Spaces, and is comprised of a number of components installed over three different servers:



The example above depicts the following work flow:

1. Users access JD Edwards EnterpriseOne by entering their JD Edwards EnterpriseOne URL in a Web browser, and then entering their user ID and password on the sign-on page.
2. The WebGate component on the Oracle HTTP Server (OHS) captures the user credentials and sends them to Oracle Access Manager (OAM) for authentication.
3. OAM compares the user credentials against the Oracle Internet Directory (OID) or a similar LDAP repository.

If the user credentials are not in OID, OAM notifies WebGate and the user is denied access to JD Edwards EnterpriseOne.

If OAM finds the user credentials in OID, OAM authenticates the user credentials.

4. After successful authentication, the user accesses the JD Edwards EnterpriseOne HTML client. If the user is WebCenter enabled, the **My WebCenter Group Spaces** link appears under the **Action** section of the EnterpriseOne menu. Clicking the link opens another browser that opens the WebCenter home page without requiring the user to sign in again.

5. The EnterpriseOne HTML client calls a WebCenter web service through Oracle Web Services Manager (OWSM). For example, the EnterpriseOne HTML client fetches a list of group spaces from WebCenter. This creates group space links in the Related Information Framework. OWSM secures the communication.
6. Securing the communication between JD Edwards EnterpriseOne and WebCenter is accomplished by using keystores on both the EnterpriseOne HTML Client server and the WebCenter server. A keystore is a file that provides information about public and private keys that are used for authentication and data integrity.
7. When the user clicks the group space link in JD Edwards EnterpriseOne, a new Web browser opens and tries to connect to the WebCenter group space.
8. The WebGate component on the WebCenter server determines that the user has been authenticated and allows the user to access the group space without logging in again.

Key components from our point of view are: an LDAP repository such as Oracle Internet Directory; installation and configuration of Oracle Access Manager; installation of Oracle HTTP Server; and installation of the Access Manager WebGate component. WebGate is a plugin that intercepts HTTP requests and forwards them to the access server for authentication and authorization. The WebGate plugin works with Oracle HTTP Server and the Oracle Access Manager components to determine whether or not a user has already been authenticated, and thus is able to allow a user access to an application without requiring an additional login. Additional information can be found in the *Oracle Fusion Middleware Enterprise Deployment Guide for Oracle Identity Management 11g Release 1*.



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

Copyright © 2012, Oracle. All rights reserved.

This document is provided for information purposes only and the contents hereof are subject to change without notice.

This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission. Oracle, JD Edwards, PeopleSoft, and Retek are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

This document is for informational purposes only and may not be incorporated into a contract or agreement.