

Oracle HTTP Server 11g R1 Configuration for FLEXCUBE
Oracle FLEXCUBE
Release 12.4.0.0.0
[May] [2017]



Table of Contents

1. PURPOSE	3
2. INTRODUCTION TO ORACLE HTTP SERVER (OHS)	3
2.1 HTTP LISTENER	3
2.2 MODULES (MODS)	3
3. INSTALLATION OF OHS 11G	4
4. CONFIGURE ORACLE HTTP SERVER INFRONT OF WEBLOGIC SERVER	11
4.1 FOR WEBLOGIC IN SINGLE INSTANCE	11
4.2 FOR WEBLOGIC INSTANCES IN CLUSTER	12
5. ENABLE “WEBLOGIC PLUG-IN ENABLED” FLAG IN WEBLOGIC	13
6. COMPRESSION RULE SETTING	13
6.1 LOADING MOD_DEFLATE	13
6.2 CONFIGURING FILE TYPES	14
6.3 HTTPD.CONF FILE CHANGES	14
7. CONFIGURING SSL FOR ORACLE HTTP SERVER	16
7.1 SSL CONFIGURATION FOR INBOUND REQUEST TO ORACLE HTTP SERVER	16
7.1.1 Create a new Wallet and import Certificate	16
7.1.2 Configuring Wallet in ssl.conf file	20
7.2 CONFIGURING SSL BETWEEN ORACLE HTTP SERVER AND ORACLE WEBLOGIC SERVER	21
7.2.1 Turn off KeepAliveEnabled	21
7.2.2 To enable one-way SSL	21
7.2.3 To enable two-way SSL	22
8. SAMPLE CONFIGURATION FILES	23
9. STARTING, STOPPING, AND RESTARTING ORACLE HTTP SERVER	23
9.1 START	23
9.2 STOP	23
9.3 RESTART	23
10. TEST THE APPLICATION	24
11. SERVER LOGS LOCATION	24
12. REFERENCES	24

1. Purpose

The objective of this document is to explain the installation and configuration of Oracle HTTP Server 11g R1 (11.1.1.6.0). This includes setting up of server details, configuration of compression rules and enabling SSL.

2. Introduction to Oracle HTTP Server (OHS)

Oracle HTTP Server is the Web server component for Oracle Fusion Middleware. It is based on Apache web server, and includes all base Apache modules and modules developed specifically by Oracle. It provides a HTTP listener for Oracle WebLogic Server and the framework for hosting static pages, dynamic pages, and applications over the Web. Key aspects of Oracle HTTP Server are its technology, its serving of both static and dynamic content and its integration with both Oracle and non-Oracle products.

Oracle HTTP Server consists of several components that run within the same process. These components provide the extensive list of features that Oracle HTTP Server offers when handling client requests.

Following are the major components:

2.1 HTTP Listener

Oracle HTTP Server is based on an Apache HTTP listener to serve client requests. An HTTP server listener handles incoming requests and routes them to the appropriate processing utility.

2.2 Modules (mods)

Modules extend the basic functionality of Oracle HTTP Server, and support integration between Oracle HTTP Server and other Oracle Fusion Middleware components. There are modules developed specifically by Oracle for Oracle HTTP Server. Ex: `mod_wl_ohs`, `mod_plsql`

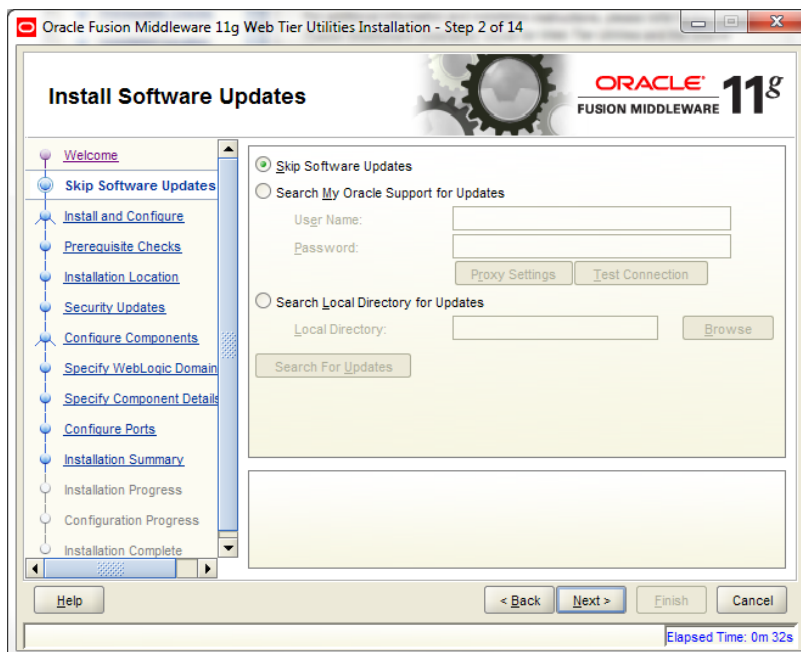
Oracle HTTP Server also includes the base Apache and third-party modules out-of-the-box. These modules are not developed by Oracle. Ex: `mod_proxy`, `mod_perl`

3. Installation of OHS 11g

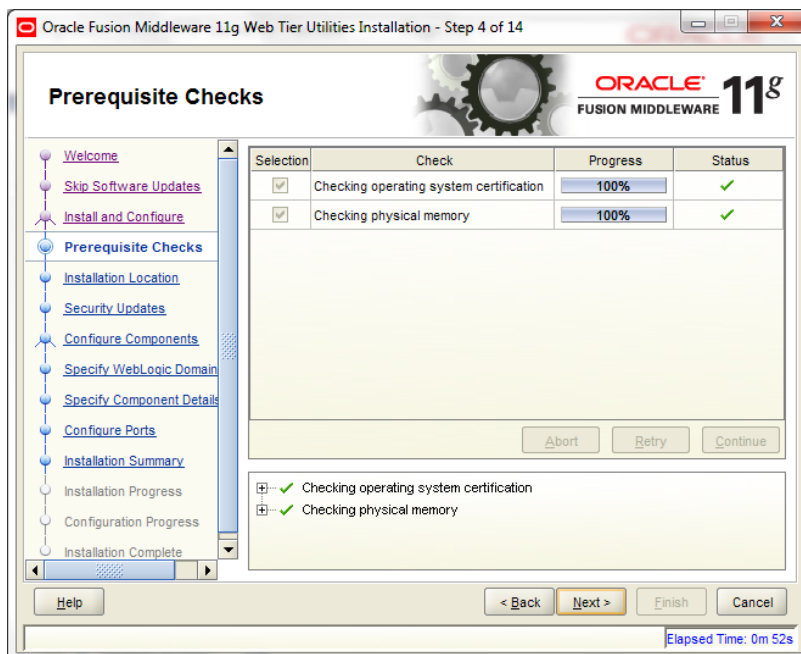
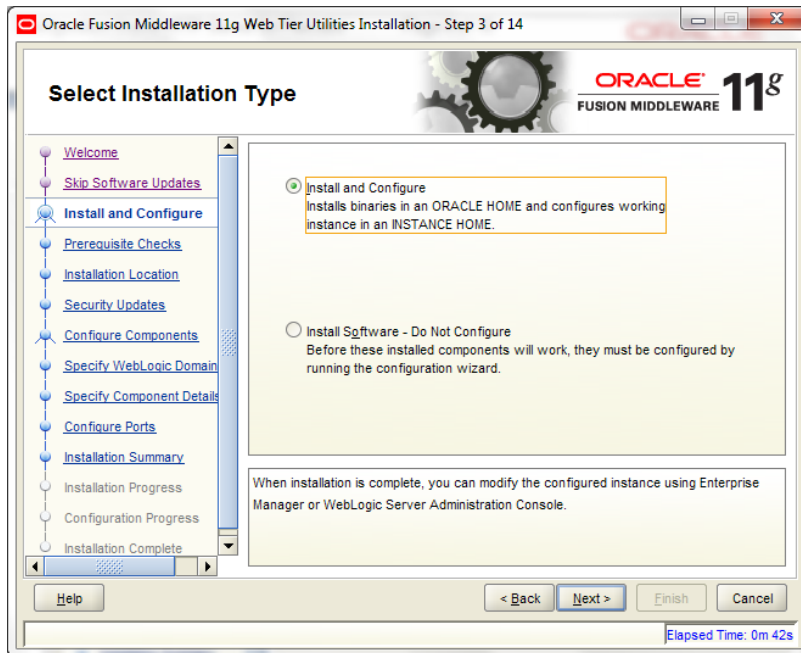
Invoke the setup exe to start the installation

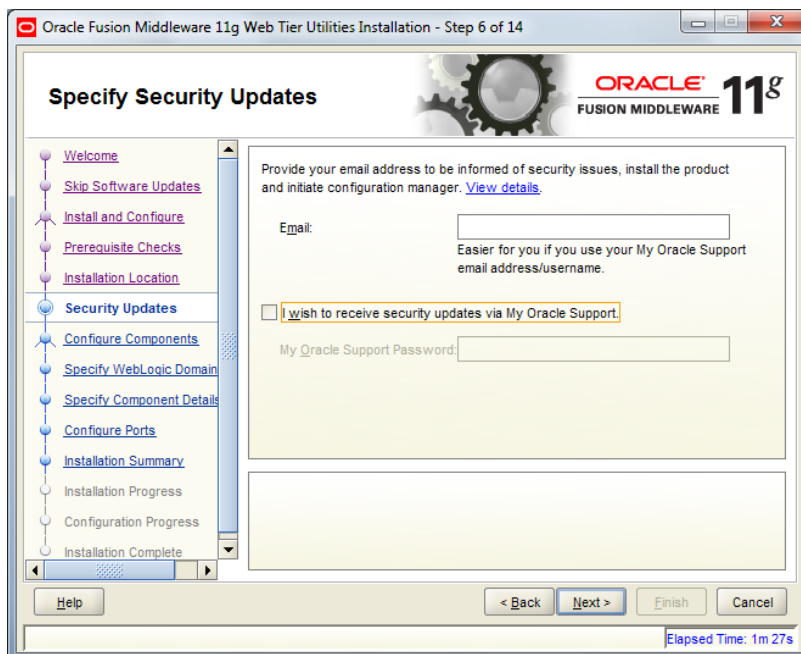
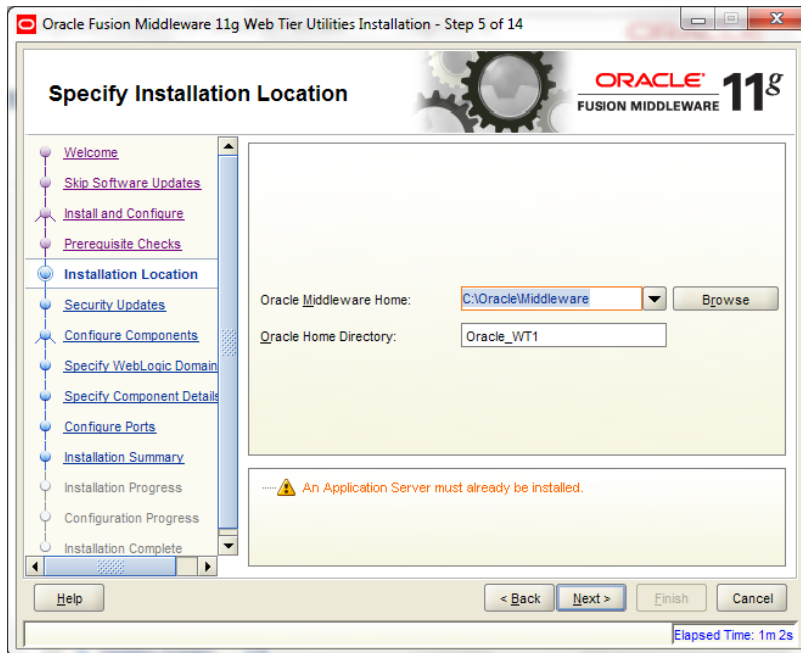


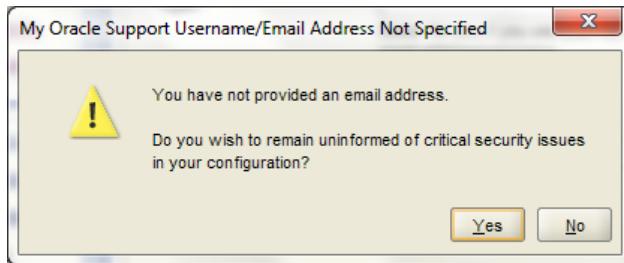
Select Skip Software Updates



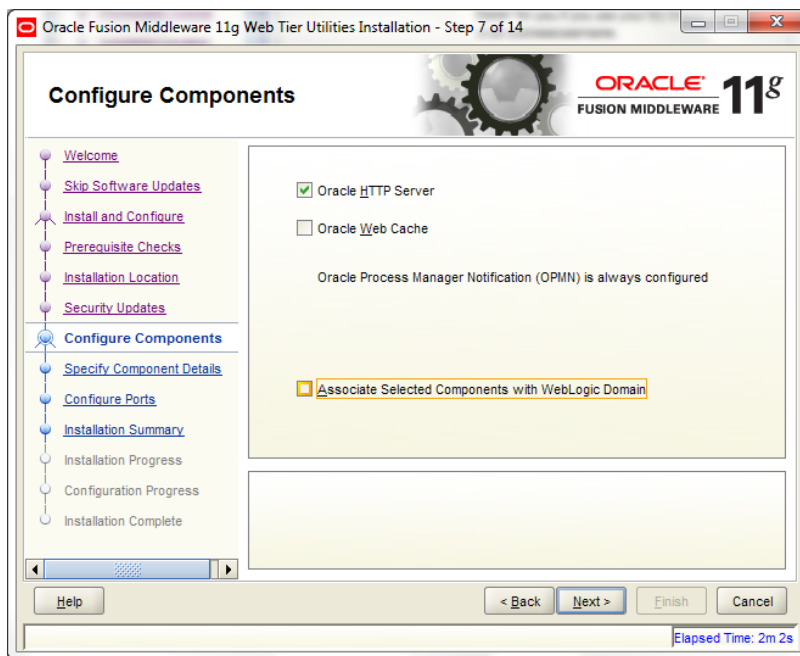
Select Install and Configure



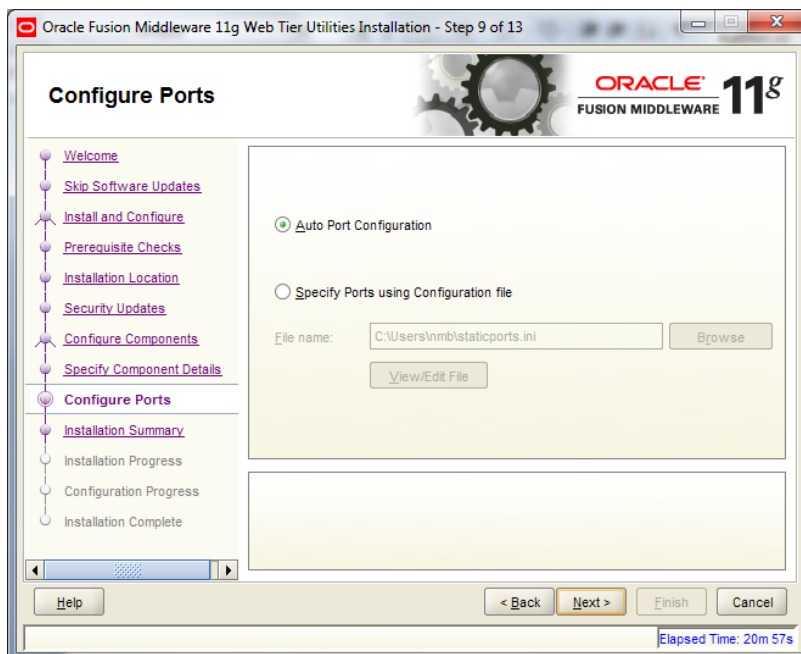
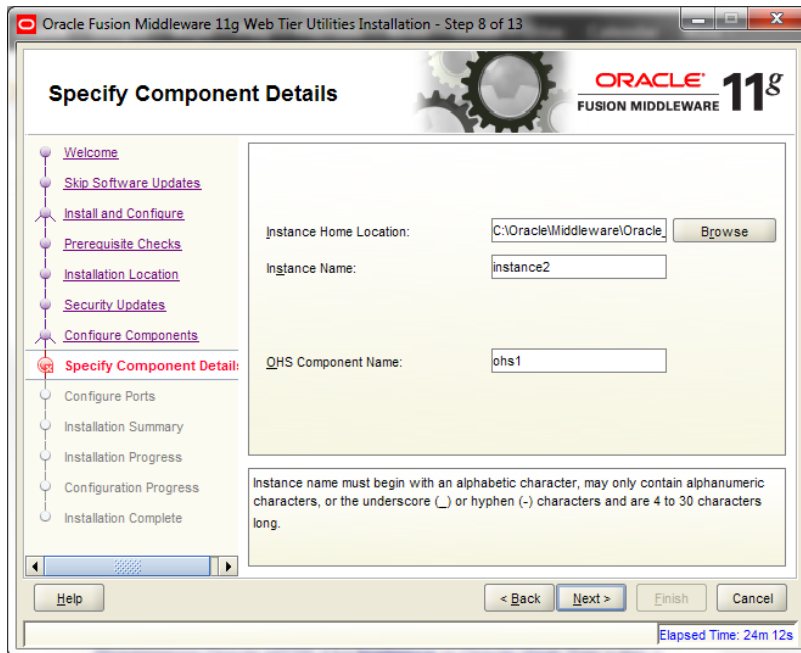


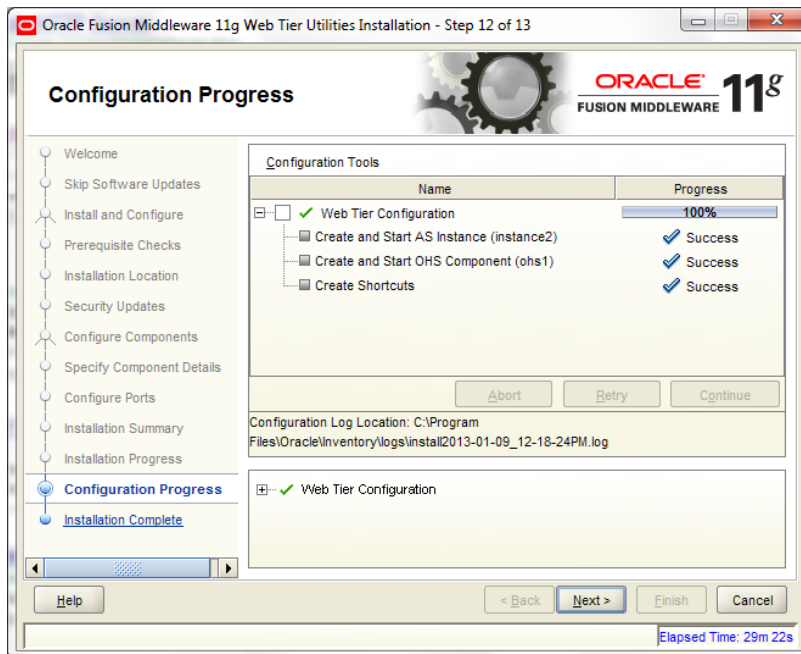
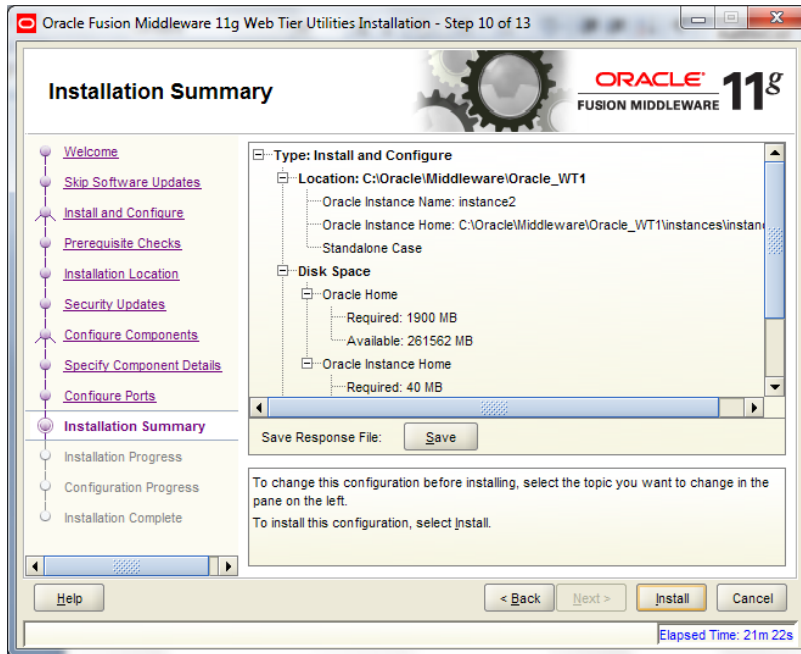


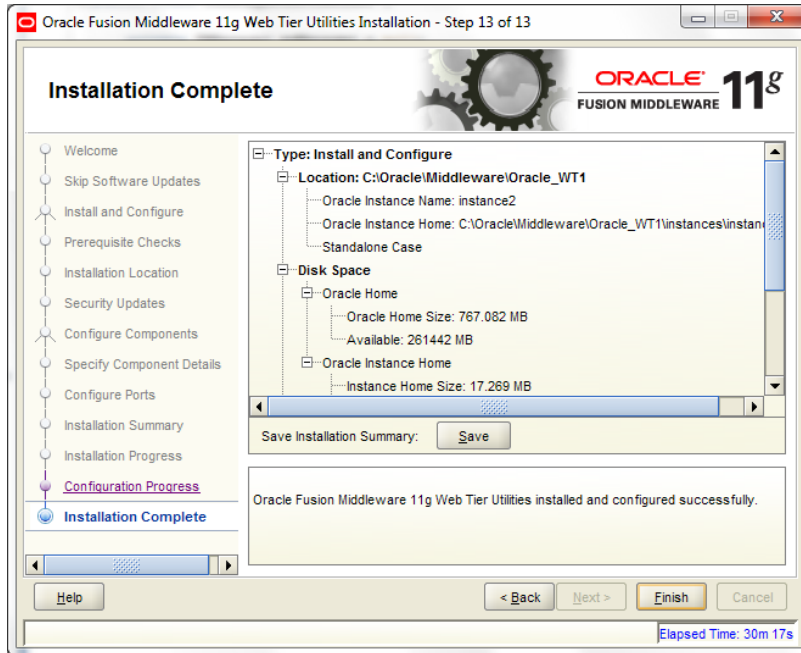
Select only Oracle HTTP Server



Enter the required OHS instance and component names







This completes the installation of Oracle HTTP Server with <Instance> and <component>. Example: Instance is instance1 and component is ohs1.

If you would like to change the port after the installation(OHS Listen Port) edit \$ORACLE_INSTANCE/config/OHS/<component_name>/httpd.conf and change the listen port.
NOTE: This port is for http protocol and not for https.

```

httpd.conf
181
182 #
183 # Listen: Allows you to bind Apache to specific IP addresses and/or
184 # ports, instead of the default. See also the <VirtualHost>
185 # directive.
186 #
187 # Change this to Listen on specific IP addresses as shown below to
188 # prevent Apache from glomming onto all bound IP addresses (0.0.0.0)
189 #
190 # Listen 12.34.56.78:80
191
192 # OHS Listen Port
193 Listen 7777
194
195 #
196 # Dynamic Shared Object (DSO) Support
197 #
198 # To be able to use the functionality of a module which was built as a DSO you
199 # have to place corresponding 'LoadModule' lines at this location so the
200 # directives contained in it are actually available _before_ they are used.
201 # Statically compiled modules (those listed by 'httpd -l') do not need
202 # to be loaded here.
203 #
204 # Example:
205 # LoadModule foo_module "${ORACLE_HOME}/ohs/modules/mod_foo.so"
206

```

4. Configure Oracle HTTP Server in front of Weblogic Server

In Oracle HTTP Server requests from Oracle HTTP Server to Weblogic server are proxied using mod_wl_ohs module. This configuration file needs to be modified to include the Weblogic server and port details.

mod_wl_ohs.conf file is located at

```
#{ORACLE_INSTANCE}/config/OHS/{COMPONENT_NAME}/mod_wl_ohs.conf
```

Add the below directives to mod_wl_ohs.conf file.

4.1 For WebLogic in single instance

```
<Location /<<context/url>> >
```

```
    SetHandler weblogic-handler
```

```
    WebLogicHost <<server name>>
```

```
    WeblogicPort <<port>>
```

```
</Location>
```

Example:

```
<Location /FCJNeoWeb>
```

```
    SetHandler weblogic-handler
```

```
    WebLogicHost wlserver1
```

```
    WeblogicPort 7707
```

```
</Location>
```

This will forward /FCJNeoWeb from HTTP server to /FCJNeoWeb on WebLogic Server wlserver1: 7707

```

1  # NOTE : This is a template to configure mod_weblogic.
2
3  LoadModule weblogic_module    "#{ORACLE_HOME}/ohs/modules/mod_wl_ohs.so"
4  LoadModule deflate_module     "#{ORACLE_HOME}/ohs/modules/mod_deflate.so"
5
6  # This empty block is needed to save mod_wl related configuration from EM to t
7  <IfModule weblogic_module>
8  #     WebLogicHost <WEBLOGIC_HOST>
9  #     WebLogicPort <WEBLOGIC_PORT>
10 #     Debug ON
11 #     WLLogFile /tmp/weblogic.log
12 #     MatchExpression *.jsp
13 </IfModule>
14
15 # <Location /weblogic>
16 #     SetHandler weblogic-handler
17 #     PathTrim /weblogic
18 #     ErrorPage http://WEBLOGIC_HOME:WEBLOGIC_PORT/
19 # </Location>
20
21 <Location /FCJNeoWeb>
22     SetHandler weblogic-handler
23     WebLogicHost wlserver1
24     WeblogicPort 7707
25 </Location>

```

4.2 For Weblogic instances in cluster

```
<Location /<<context/url>> >
```

```
    SetHandler weblogic-handler
```

```
    WebLogicCluster <server1>:<port1>,<server2>:<port2>
```

```
</Location>
```

Example

```
<Location / FCJNeoWeb >
```

```
    SetHandler weblogic-handler
```

```
    WebLogicCluster wlserver1:7010, wlserver2:7010
```

```
</Location>
```

This will forward /FCJNeoWeb from HTTP server to /FCJNeoWeb on WebLogic Cluster wlserver1:7010 and wlserver2:7010

```
mod_wl_ohs.conf
1 # NOTE : This is a template to configure mod_weblogic.
2
3 LoadModule weblogic_module    "${ORACLE_HOME}/ohs/modules/mod_wl_ohs.so"
4 LoadModule deflate_module     "${ORACLE_HOME}/ohs/modules/mod_deflate.so"
5
6 # This empty block is needed to save mod_wl related configuration from EM to this fi
7 <IfModule weblogic_module>
8 #     WebLogicHost <WEBLOGIC_HOST>
9 #     WebLogicPort <WEBLOGIC_PORT>
10 #     Debug ON
11 #     WLLogFile /tmp/weblogic.log
12 #     MatchExpression *.jsp
13 </IfModule>
14
15 # <Location /weblogic>
16 #     SetHandler weblogic-handler
17 #     PathTrim /weblogic
18 #     ErrorPage http://WEBLOGIC_HOME:WEBLOGIC_PORT/
19 # </Location>
20
21 <Location /FCJNeoWeb>
22     <Location / FCJNeoWeb >
23         SetHandler weblogic-handler
24         WebLogicCluster wlserver1:7010,wlserver2:7010
25     </Location>
26
```

5. Enable “WebLogic Plug-In Enabled” flag in weblogic

This flag needs to be enabled in weblogic if it is accessed through proxy plugins. When the WebLogic plugin is enabled, a call to `getRemoteAddr` will return the address of the browser client from the proprietary WL-Proxy-Client-IP header instead of the web server.

- a. Plugin flag at managed server level
 - i. Click on 'Environment' -> 'Servers' -> '<ManagedServer>' -> 'General' -> 'Advanced'
 - ii. Check the 'WebLogic Plug-In Enabled' box.
 - iii. Click 'Save'
 - iv. Restart the Server.
- b. Plugin flag at domain level
 - v. Click on <Domain> -> 'Web Applications'
 - vi. Check the 'WebLogic Plug-In Enabled' box.
 - vii. Click 'Save'
 - viii. Restart the server.

6. Compression rule setting

Content compression in Oracle HTTP Server is done using `mod_deflate`. This can compress HTML, text or XML files to approx. 20 - 30% of their original sizes, thus saving on server traffic. However, compressing files causes a slightly higher load on the server, but clients' connection times to server is reduced.

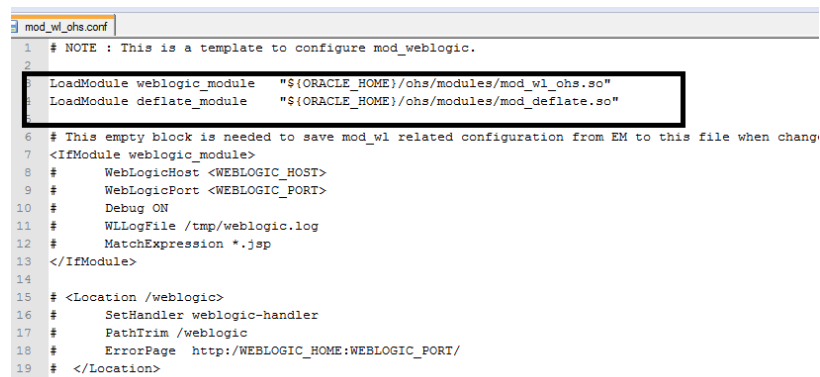
6.1 Loading `mod_deflate`

`mod_deflate` is used for compression in OHS and this is installed in Oracle HTTP Server under location `"${ORACLE_HOME}/OHS/modules/mod_deflate.so"`

But it might not be loaded.

To load the file add the below directive in `mod_wl_ohs.conf` file

```
LoadModule deflate_module      "${ORACLE_HOME}/OHS/modules/mod_deflate.so"
```



```

1 # NOTE : This is a template to configure mod_weblogic.
2
3 LoadModule weblogic_module      "${ORACLE_HOME}/ohs/modules/mod_wl_ohs.so"
4 LoadModule deflate_module      "${ORACLE_HOME}/ohs/modules/mod_deflate.so"
5
6 # This empty block is needed to save mod_wl related configuration from EM to this file when chang
7 <IfModule weblogic_module>
8 #     WebLogicHost <WEBLOGIC_HOST>
9 #     WebLogicPort <WEBLOGIC_PORT>
10 #     Debug ON
11 #     WLLogFile /tmp/weblogic.log
12 #     MatchExpression *.jsp
13 </IfModule>
14
15 # <Location /weblogic>
16 #     SetHandler weblogic-handler
17 #     PathTrim /weblogic
18 #     ErrorPage http://WEBLOGIC_HOME:WEBLOGIC_PORT/
19 # </Location>

```

6.2 Configuring file types

mod_deflate also requires to specify which type files are going to be compressed.

In the LOCATION section of mod_wl_ohs.conf file add the below entries.

```
AddOutputFilterByType DEFLATE text/plain
AddOutputFilterByType DEFLATE text/xml
AddOutputFilterByType DEFLATE application/xhtml+xml
AddOutputFilterByType DEFLATE text/css
AddOutputFilterByType DEFLATE application/xml
AddOutputFilterByType DEFLATE application/x-javascript
AddOutputFilterByType DEFLATE text/html
SetOutputFilter DEFLATE
```

Images are supposed to be in a compressed format, and therefore are bypassed by mod_deflate.

```
21 <Location /FCJNeoWeb>
22     SetHandler weblogic-handler
23     WebLogicHost wlserver1
24     WebLogicPort 7707
25
26     AddOutputFilterByType DEFLATE text/plain
27     AddOutputFilterByType DEFLATE text/xml
28     AddOutputFilterByType DEFLATE application/xhtml+xml
29     AddOutputFilterByType DEFLATE text/css
30     AddOutputFilterByType DEFLATE application/xml
31     AddOutputFilterByType DEFLATE application/x-javascript
32     AddOutputFilterByType DEFLATE text/html
33     SetOutputFilter DEFLATE
```

6.3 httpd.conf file changes

This is a server configuration file which typically contains directives that affect how the server runs, such as user and group IDs it should use, and location of other files. Cross check the existence of mod_wl_ohs.conf include in httpd.conf file.

httpd.conf file is present under location

```
"${ORACLE_INSTANCE}/config/OHS/{COMPONENT_NAME}/httpd.conf"
```

In this file cross check for the below entry

```
include "${ORACLE_INSTANCE}/config/OHS/${COMPONENT_NAME}/mod_wl_ohs.conf"
```

If above include entry is not present, then add the above include section.

```
0013 #Directives to setup logging via ODL
0014 OraLogDir "${ORACLE_INSTANCE}/diagnostics/logs/${COMPONENT_TYPE}/${COMPONENT_NAME}"
0015 OraLogMode odl-text
0016 OraLogSeverity WARNING:32
0017 OraLogRotationParams S 10:70
0018
0019
0020 # Set it to On to enable Audit Logs
0021 OraAuditEnable On
0022
0023 # Include the configuration files needed for mod_weblogic
0024 include "${ORACLE_INSTANCE}/config/${COMPONENT_TYPE}/${COMPONENT_NAME}/mod_wl_ohs.conf"
0025
0026 # Include the SSL definitions and Virtual Host container
0027 include "${ORACLE_INSTANCE}/config/${COMPONENT_TYPE}/${COMPONENT_NAME}/ssl.conf"
0028
0029 # Include the admin virtual host (Proxy Virtual Host) related configuration
0030 include "${ORACLE_INSTANCE}/config/${COMPONENT_TYPE}/${COMPONENT_NAME}/admin.conf"
0031
0032 include "moduleconf/*.conf"
0033
```

7. Configuring SSL for Oracle HTTP Server

Secure Sockets Layer (SSL) is required to run any Web site securely. Secure Sockets Layer (SSL) is an encrypted communication protocol that is designed to securely send messages across the Internet.

Reading of “**SSL_Configuration on Weblogic**” document provided as part of FCUBS installation is recommended before proceeding with further setup.

In Oracle HTTP server, SSL configuration can be done between

1. Browser to Oracle HTTP Server(Mandatory)
2. Oracle HTTP Server to Oracle Weblogic Server(If required)

7.1 SSL configuration for Inbound Request to Oracle HTTP Server

Perform these tasks to enable and configure SSL between browser and Oracle HTTP Server.

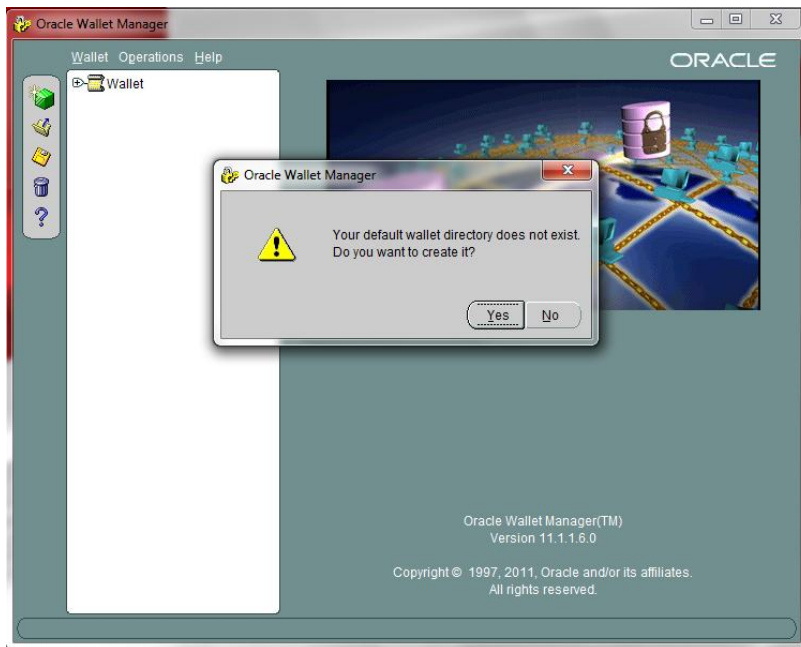
1. Obtain a certificate from CA or create a self signed certificate.
2. Create an Oracle Wallet which contains the above SSL Certificate. The default wallet that is automatically installed with Oracle HTTP Server is for testing purposes only. The default wallet is located in "`${ORACLE_INSTANCE}/config/OHS/${COMPONENT_NAME}/keystores/default`"
3. Configuring Wallet in ssl.conf file

7.1.1 Create a new Wallet and import Certificate

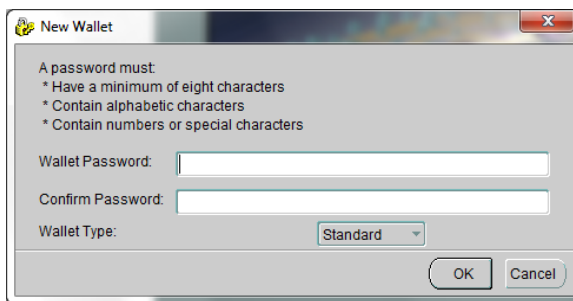
1. Go to the `\Oracle_WT1\bin\launch.exe`, this will launch your wallet manager



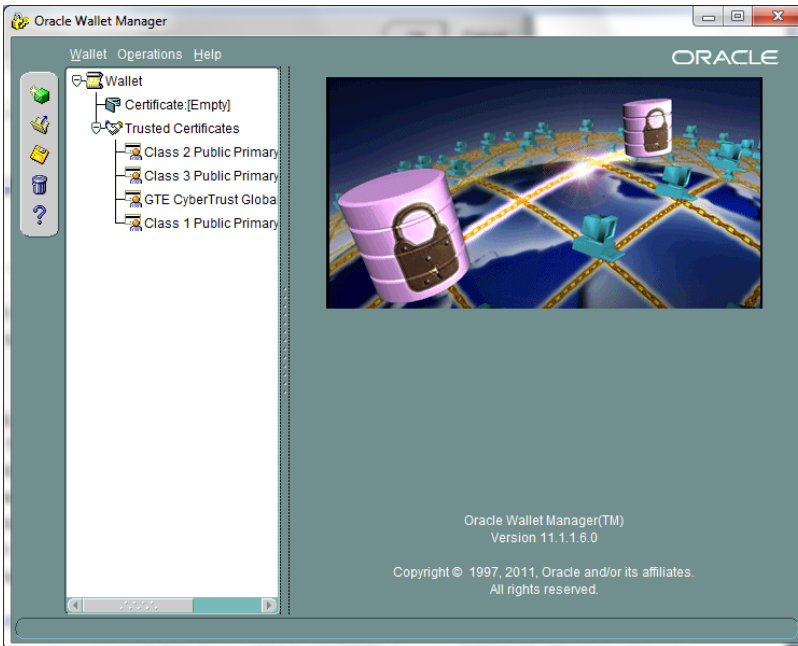
2. Click on Create new and then click no option.



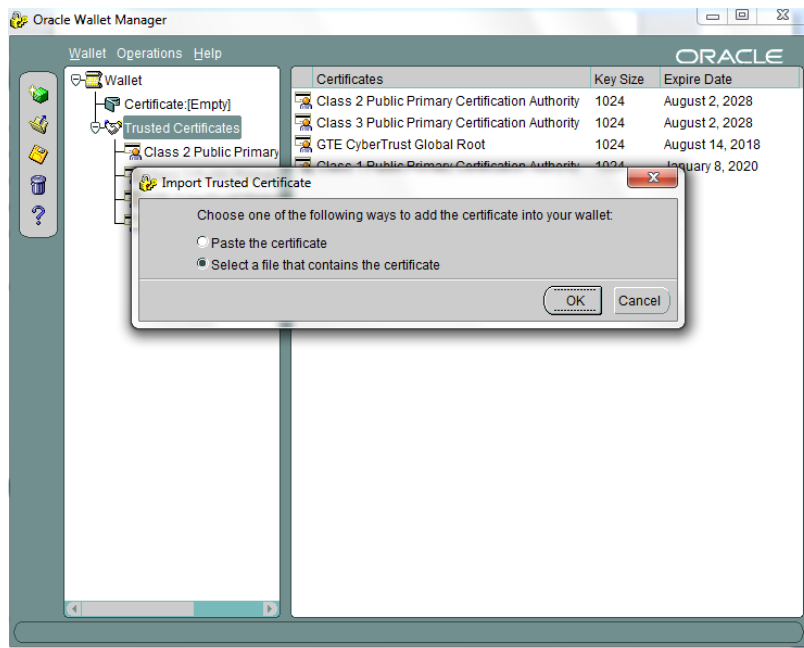
3. Enter the wallet password and click on OK, this will create a new wallet.



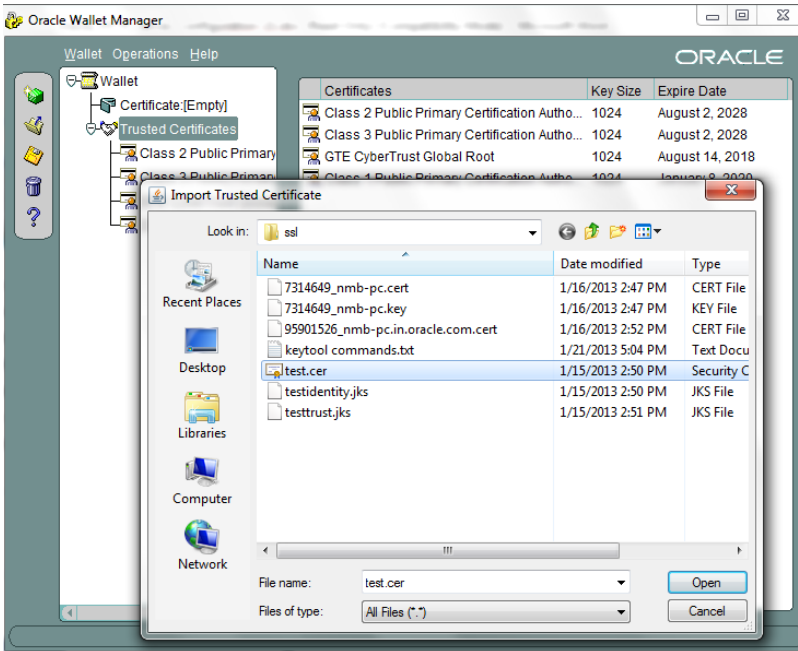
4. Not it will ask for certificate request creation, Click on NO to proceed



5. Right click on trusted certificates and then import trusted certificate.

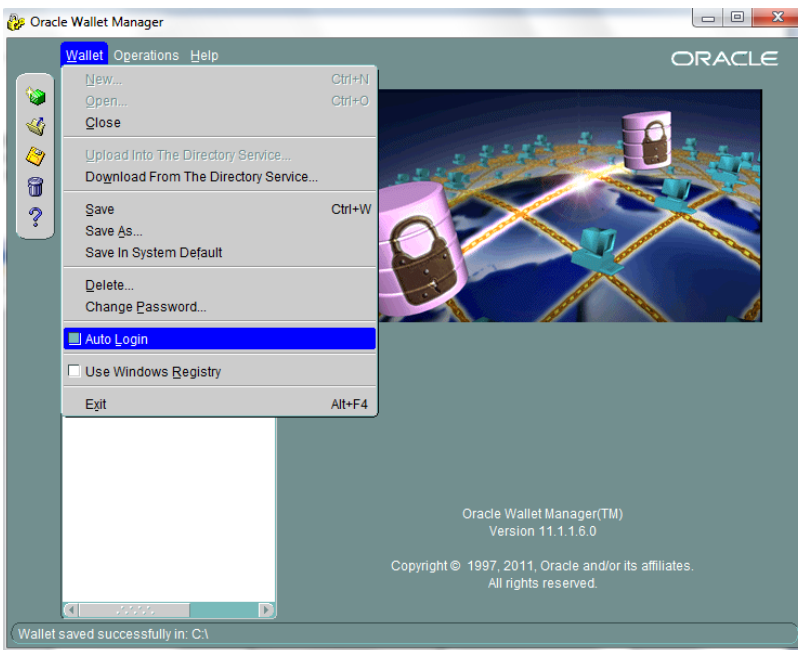


6. Browse to the folder where certificate is stored and click on Open



7. Click on Save Wallet button on the left side navigation and save the wallet either to default location("\${ORACLE_INSTANCE}/config/OHS/\${COMPONENT_NAME}/keystores/default") or folder of your choice.

8. Click on Wallet tab and enable Auto Login



7.1.2 Configuring Wallet in ssl.conf file

In ssl.conf file the newly created wallet need to updated. This file is located under folder

"\${ORACLE_INSTANCE}/config/OHS/\${COMPONENT_NAME}/

1. Change the SSLWallet directive to point to the location of new wallet created.

SSLWallet "\${ORACLE_INSTANCE}/config/\${COMPONENT_TYPE}/\${COMPONENT_NAME}/keystores/"

```

9 # Also the ciphers that the client is permitted to negotiate.
10 SSLCipherSuite
11 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,
12 AES_256_CBC_SHA
13
14 # SSL Certificate Revocation List Check
15 # Valid values are On and Off
16 SSLCRLCheck Off
17
18 #Path to the wallet
19 SSLWallet "${ORACLE_INSTANCE}/config/${COMPONENT_TYPE}/${COMPONENT_NAME}/keystores/"
20
21 <FilesMatch "\.(cgi|shtml|phtml|php)$">
22   SSLOptions +StdEnvVars
23 </FilesMatch>
24
25 <Directory "${ORACLE_INSTANCE}/config/${COMPONENT_TYPE}/${COMPONENT_NAME}/cgi-bin">
26   SSLOptions +StdEnvVars
27 </Directory>
28
29 BrowserMatch ".MSIE.*" \
30   nokeepalive ssl-unclean-shutdown \
31   downgrade-1.0 force-response-1.0
32 </IfModule>
33 </VirtualHost>
34 </IfModule>

```

2. Change the Listen port number in ssl.conf file to the SSL enabled port, by default the value is 4443

```

1 #####
2 # Oracle HTTP Server mod_oss1 configuration file: ssl.conf
3 #####
4
5
6 # OHS Listen Port
7 Listen 4443
8
9 <IfModule oss1_module>
10 ##
11 ## SSL Global Context
12 ##
13 ## All SSL configuration in this context applies both to
14 ## the main server and all SSL-enabled virtual hosts.
15 ##
16
17 #
18 # Some MIME-types for downloading Certificates and CRLs
19 AddType application/x-x509-ca-cert .crt
20 AddType application/x-pkcs7-crl .crl
21
22 # Pass Phrase Dialog:

```

7.2 Configuring SSL between Oracle HTTP Server and Oracle Weblogic Server

SSL for outbound requests from Oracle HTTP Server are configured in mod_wl_ohs.

Refer to “**SSL_Configuration on Weblogic**” document for weblogic server setting mentioned in below section.

7.2.1 Turn off KeepAliveEnabled

The below parameter in mod_wl_ohs should be turned off, by default it is on. Add the below directive under LOCATION section of mod_wl_ohs file

KeepAliveEnabled OFF

```

6      AddOutputFilterByType DEFLATE text/plain
7      AddOutputFilterByType DEFLATE text/xml
8      AddOutputFilterByType DEFLATE application/xhtml+xml
9      AddOutputFilterByType DEFLATE text/css
0      AddOutputFilterByType DEFLATE application/xml
1      AddOutputFilterByType DEFLATE application/x-javascript
2      AddOutputFilterByType DEFLATE text/html
3      SetOutputFilter DEFLATE
4
5      KeepAliveEnabled OFF
6
7      W1SSLWallet "D:\misc\ssl\"
8  </Location>
9

```

7.2.2 To enable one-way SSL

1. Generate a custom keystore identity.jks for Weblogic Server containing a certificate.
2. At Identity section in Keystores tab in weblogic Admin Console for server set
 - a. The custom trust store with the identity.jks file location
 - b. The keystore type as JKS
 - c. The passphrase used to created the keystore

Home > base_domain > Summary of Environment > Summary of Servers > AdminServer

Messages

- ⚠ Changes to your Keystore configuration may require you to update your SSL Configuration. Please review your settings on the SSL tab.
- ✔ All changes have been activated. No restarts are necessary.
- ✔ Settings updated successfully.

Settings for AdminServer

Configuration Protocols Logging Debug Monitoring Control Deployments Services Security Notes

General Cluster Services **Keystores** SSL Federation Services Deployment Migration Tuning Overload Health Monitoring Server Start

Save

Keystores ensure the secure storage and management of private keys and trusted certificate authorities (CAs). This page lets you view and define various keystore configurations to manage the security of message transmissions.

Keystores: Custom Identity and Custom Trust | Change | Which configuration rules should be used for...? More Info...

Identity

Custom Identity Keystore: D:\misc\testidentity.jks The path and file name of the identity keystore.

Custom Identity Keystore Type: JKS The type of the keystore. Generally, this is JKS.

Custom Identity Keystore Passphrase: The encrypted custom identity keystore's passphrase. If the keystore is opened without a passphrase, it will be opened without a passphrase.

Confirm Custom Identity Keystore Passphrase:

Trust

Custom Trust Keystore: D:\misc\testtrust.jks The path and file name of the custom trust keystore.

Custom Trust Keystore Type: JKS The type of the keystore. Generally, this is JKS.

Custom Trust Keystore Passphrase: The custom trust keystore's passphrase. If the keystore is opened without a passphrase, it will be opened without a passphrase. More Info

Confirm Custom Trust Keystore Passphrase:

3. Copy the certificate to Oracle HTTP Server and import the new certificate into OHS wallet as a trusted certificate.
4. Add following new directive in mod_wl_ohs.conf to point to the wallet location
`WISSSLWallet "${ORACLE_INSTANCE}/config/OHS/{COMPONENT_NAME}/keystores/default"`
5. Change the port in mod_wl_ohs file to point to SSL port of Weblogic server.

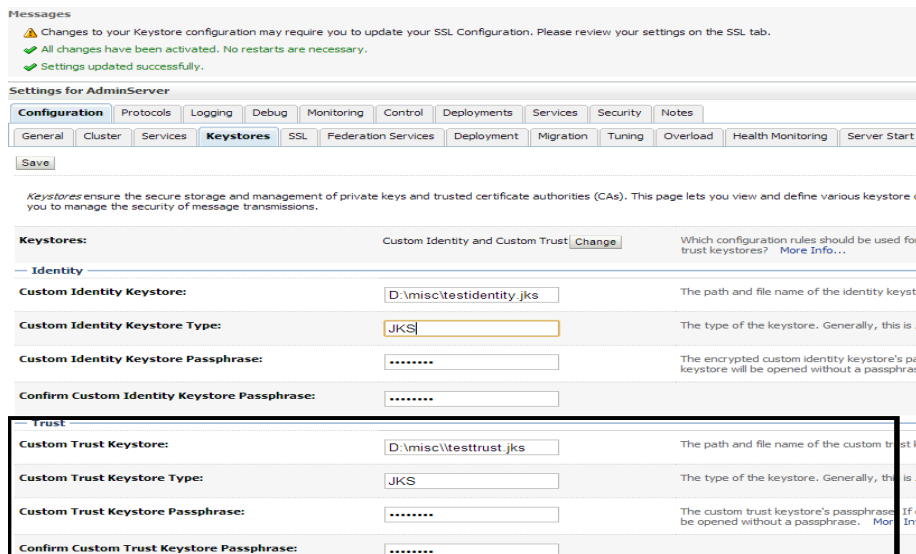
```

20
21 <Location /FCJNeoWeb>
22     CustomIdentityKeyStore
23     WebLogicHost wlserver1
24     WebLogicPort 443
25
26     AddOutputFilterByType DEFLATE text/plain
27     AddOutputFilterByType DEFLATE text/xml
28     AddOutputFilterByType DEFLATE application/xhtml+xml
29     AddOutputFilterByType DEFLATE text/css
30     AddOutputFilterByType DEFLATE application/xml
31     AddOutputFilterByType DEFLATE application/x-javascript
32     AddOutputFilterByType DEFLATE text/html
33     SetOutputFilter DEFLATE
34
35     KeepAliveEnabled OFF
36
37     WISSSLWallet "${ORACLE_INSTANCE}/config/OHS/{COMPONENT_NAME}/keystores/"
38 </Location>
39
40
    
```

6. Restart both Weblogic Server and Oracle HTTP Server

7.2.3 To enable two-way SSL

1. Perform one-way SSL configuration steps
2. Generate a new trust store, trust.jks for Weblogic server
3. Keystore created for one-way SSL could be used, but it is recommended to create a separate truststore
4. Export the user certificate from Oracle HTTP Server wallet, and import it into truststore created above
5. At Trust section in Keystores tab in Weblogic Admin Console for the server set
 - a. The custom trust store with the trust.jks file location
 - b. The keystore type as JKS
 - c. The passphrase used to created the keystore



6. Under the SSL tab

Ensure trusted CA is set as from Custom Trust Keystore.

The screenshot shows the Oracle HTTP Server Administration Console interface. The top navigation bar includes 'Home', 'Log Out', 'Preferences', 'Record', and 'Help'. Below the navigation bar, the breadcrumb trail reads 'Home > base_domain > Summary of Environment > Summary of Servers > AdminServer'. The main heading is 'Settings for AdminServer'. The 'Configuration' tab is selected, and the 'SSL' sub-tab is active. A 'Save' button and a 'Configuration - Services- Tab' button are visible. The main content area is titled 'Identity and Trust Locations' and contains several sections: 'Identity', 'Private Key Location', 'Private Key Alias' (set to 'selfcert'), 'Private Key Passphrase' (masked), 'Confirm Private Key Passphrase' (masked), 'Certificate Location' (set to 'from Custom Identity Keystore'), and 'Trusted Certificate Authorities' (set to 'from Custom Trust Keystore'). A red box highlights the 'Private Key Alias', 'Private Key Passphrase', 'Confirm Private Key Passphrase', and 'Certificate Location' fields.

7. Restart Weblogic Server

8. Sample Configuration Files



9. Starting, Stopping, and Restarting Oracle HTTP Server

Navigate to the below location in command prompt `#{ORACLE_INSTANCE}/bin/` and run below commands

9.1 Start

```
opmnctl startproc ias-component=#{COMPONENT_NAME}
```

Example: `opmnctl startproc ias-component=ohs1`

9.2 Stop

```
opmnctl stopproc ias-component=#{COMPONENT_NAME}
```

Example: `opmnctl stopproc ias-component=ohs1`

9.3 Restart

```
opmnctl restartproc ias-component=#{COMPONENT_NAME}
```

Example: `opmnctl restartproc ias-component=ohs1`

10. Test the application

Test the application deployed on Weblogic using Oracle HTTP Server after restarting both the oracle http server and weblogic server

`https://ohs_servername:ohs_https_port/<<context/url>>`

`http://ohs_servername:ohs_http_port/<<context/url>>`

ohs_servername: server on which OHS is deployed

ohs_https_port: port number mentioned against LISTEN directive in SSL.conf file

ohs_http_port: port number mentioned against LISTEN directive in httpd.conf file

Example:

`https://localhost:4443/FCJNeoWeb/welcome.jsp`

Or

`http://localhost:7777/FCJNeoWeb/welcome.jsp`

11. Server Logs Location

Oracle HTTP Server Logs are generated under folder

`${ORACLE_INSTANCE}/diagnostics/logs/OHS/{COMPONENT_NAME}/`

12. References

SSL_Configuration.doc for Weblogic provided as part of FCUBS installation.

http://docs.oracle.com/cd/E16764_01/web.1111/e10144/under_mods.htm

http://docs.oracle.com/cd/E25054_01/core.1111/e10105/sslconfig.htm



Oracle_HTTP_Server_Configuration
[May] [2017]
Version 12.4.0.0.0

Oracle Financial Services Software Limited
Oracle Park
Off Western Express Highway
Goregaon (East)
Mumbai, Maharashtra 400 063
India

Worldwide Inquiries:
Phone: +91 22 6718 3000
Fax: +91 22 6718 3001
www.oracle.com/financialservices/

Copyright © 2007, 2017, Oracle and/or its affiliates. All rights reserved.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate failsafe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

This software or hardware and documentation may provide access to or information on content, products and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.