Oracle® Banking Enterprise Limits and Collateral Management

Oracle HTTP Server 11g R1 Configuration for FLEXCUBE

Release 14.8.0.0.0
G32493-01
April 2025

Oracle Banking Enterprise Limits and Collateral Management Oracle HTTP Server 11g R1 Configuration for FLEXCUBE, Release 14.8.0.0.0

G32493-01

Contents

1 Preface

1.1 Purpose

This guide helps the user to create the required resources for Oracle Banking Enterprise Limits and Collateral Management.

1.2 Audience

This guide is intended for anyone responsible for installing Oracle Banking Application.

1.3 Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at https://www.oracle.com/corporate/accessibility/.

Access to Oracle Support

Oracle customer access to and use of Oracle support services will be pursuant to the terms and conditions specified in their Oracle order for the applicable services.

1.4 Critical Patches

Oracle advises customers to get all their security vulnerability information from the Oracle Critical Patch Update Advisory, which is available at Critical Patches, Security Alerts and Bulletins. All critical patches should be applied in a timely manner to ensure effective security, as strongly recommended by Oracle Software Security Assurance.

1.5 Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve.

Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

1.6 Conventions

The following text conventions are used in this document:

Table 1-1 Conventions

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1.7 Screenshot Disclaimer

Personal information used in the interface or documents is dummy and does not exist in the real world. It is only for reference purposes.

2 Purpose

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

3

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. The following are the major components -

Table 3-1 Major Component- Oracle HTTP Server

Major Components	Description
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.
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. For example, **mod_wl_ohs**, **mod_plsql** etc. Oracle HTTP Server also includes the base Apache and third-party modules out-of-the-box. These modules are not developed by Oracle. For example, **mod_proxy**, **mod_perl** etc.

4

Install OHS 11g

This topic explains systematic instructions to install Oracle HTTP Server 11g.

1. Launch the setup exe file to start the installation process.

   The **Welcome** screen displays.

   Figure 4-1 Welcome

2. Click **Next**.

   The **Install Software Updates** screen displays.

   Figure 4-2 Install Software Updates

3. Select **Skip Software Updates** option and click **Next**.

   The **Install and Configure** screen displays.

   Figure 4-3 Install and Configure

4. Select **Install and Configure** option and click **Next**.

   The **Prerequisite Checks** screen displays.

   Figure 4-4 Prerequisite Checks

5. Check the **Checking Operating system certification** and **Checking physical memory** boxes and click **Next**.

   The **Installation Location** screen displays.

   Figure 4-5 Installation Location

6. Select **Oracle Middleware Home** path by navigating through **Browse** and then click **Next**.

   The **Security Updates** screen displays.

   Figure 4-6 Security Updates

7. Click **Next**.

   The **My Oracle Support Username/Email Address Not Specified**message displays.

   Figure 4-7 My Oracle Support Username/Email Address Not Specified

8. Click **Yes**.

   The **Configure Components** screen displays.

Figure 4-8   Configure Components

9. Select **Oracle HTTP Server** option and then click **Next**.

   The **Specify Component details** screen displays.

   Figure 4-9   Specify Component details

10. Enter the required **Instance Name** and **OHS Component Name** and click **Next**.

    The **Configure Ports** screen displays.

    Figure 4-10   Configure Ports

11. Select **Auto Port Configuration** option and click **Next**.

    The **Installation Summary** screen displays.

    Figure 4-11   Installation Summary

12. Click **Install**.

    The **Configuration Progress** screen displays.

    Figure 4-12   Configuration Progress

13. Click **Next**.

    The **Installation Complete** screen displays.

    Figure 4-13   Installation Complete

    This completes the installation of Oracle HTTP Server with **Instance** and **Component**. For example, **Instance** is **instance1** and **Component** is **ohs1**.

14. If the user wants to change the OHS Listen Port after the installation, 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.

    Figure 4-14   httpd.conf

5 Configure Oracle HTTP Server in front of Weblogic Server

This topic explains systematic instructions to configure Oracle HTTP Server in front of Weblogic Server.

In Oracle HTTP Server requests from Oracle HTTP Server to the 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-mentioned directives to **mod_wl_ohs.conf** file.

  For WebLogic in single instance -

  ```
  <Location /<<context/url>> >
  SetHandler weblogic-handler
  WebLogicHost <<server name>>
  WeblogicPort <<port>>
  </Location>
  ```

  For 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

  Figure 5-1 mod_wl_ohs.conf

  For Weblogic instances in cluster -

  ```
  <Location /<<context/url>> >
  SetHandler weblogic-handler
  WebLogicCluster <server1>:<port1>,<server2>:<port2>
  </Location>
  ```

  For 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

Figure 5-2   mod_wl_ohs.conf

6 Enable WebLogic Plug-In Enabled flag in Weblogic

This topic explains systematic instructions to enable the 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 webserver.

1. Enable **WebLogic Plug-In Enabled** flag at managed server level as per below steps.
    a. Click **Environment**, and then **Servers**, and then **ManagedServer**, and then **General**, and then **Advanced** tab respectively.

       The **Advanced** tab displays.
    b. On the **Advanced** tab, check the **WebLogic Plug-In Enabled** box.
    c. Click **Save**.
    d. Restart the Server.
2. Enable **WebLogic Plug-In Enabled** flag at domain level as per below steps:
    a. Click **Domain**, and then click **Web Applications**.
    b. Check the **WebLogic Plug-In Enabled** box.
    c. Click **Save**.
    d. Restart the server.

7 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 the server are reduced.

- Load mod_deflate
  This topic explains systematic instructions to load mod_deflate.
- Configure file types
  This topic explains systematic instructions to configure file types.
- Change httpd.conf file
  This topic explains systematic instructions to change the httpd.conf file.

7.1 Load mod_deflate

This topic explains systematic instructions to load 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`

  Figure 7-1 mod_wl_ohs.conf

7.2 Configure file types

This topic explains systematic instructions to configure file types.

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

- In the **LOCATION** section of the **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**.

Figure 7-2 mod_wl_ohs.conf

7.3 Change httpd.conf file

This topic explains systematic instructions to change the httpd.conf file.

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

1. 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`.

2. In this file, cross check for the entry include *${ORACLE_INSTANCE}/config/OHS/${COMPONENT_NAME}/mod_wl_ohs.conf*.
3. If above include entry is not present, then add the *${ORACLE_INSTANCE}/config/OHS/${COMPONENT_NAME}/mod_wl_ohs.conf* in **httpd.conf** file.

Figure 7-3 httpd.conf

8 SSL Configuration 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 Weblogic Configuration document provided as part of FCUBS installation is recommended before proceeding with further setup.

In Oracle HTTP server, SSL configuration can be done between -

- Browser to Oracle HTTP Server (Mandatory)
- Oracle HTTP Server to Oracle Weblogic Server (If required)

- Configure SSL for Inbound Request to Oracle HTTP Server
  This topic explains systematic instructions to enable and configure SSL between browser and Oracle HTTP Server.
- SSL Configuration between Oracle HTTP Server and Oracle Weblogic Server
  This topic describes the SSL configuration process between the Oracle HTTP server and the Oracle Weblogic server.

8.1 Configure SSL for Inbound Request to Oracle HTTP Server

This topic explains systematic instructions 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. Configure Wallet in **ssl.conf** file

- Create a new Wallet and import Certificate
  This topic explains systematic instructions to create a new Wallet and import Certificate.
- Configure Wallet in ssl.conf file
  This topic explains systematic instructions to configure Wallet in the ssl.conf file.

8.1.1 Create a new Wallet and import Certificate

This topic explains systematic instructions to create a new Wallet and import Certificate.

1. Go to the **\Oracle_WT1\bin\launch.exe** to launch wallet manager.

   Oracle Wallet Manager- Home screen displays.

   Figure 8-1 Oracle Wallet Manager- Home
2. Click **Create New**.

   A pop-up message displays on the **Oracle Wallet Manager** screen.

Figure 8-2   Message-Oracle Wallet Manager

3. Click **Yes**.

   The **New Wallet** window displays.

Figure 8-3   New Wallet

4. Enter the **Wallet Password** and **Confirm Password** fields and then click **OK**.

   New Wallet will get created and it will ask for certificate request creation.

5. Click **No** to proceed.

   The **Oracle Wallet Manager** screen displays with **Trusted Certificates**.

Figure 8-4   Oracle Wallet Manager

6. Right-click **Trusted Certificates**.

   The **Import Trusted Certificate** window displays.

Figure 8-5   Import Trusted Certificate

7. Choose **Select a file that contains the certificate** option and click **OK** to import the trusted certificate.

   The **Import Trusted Certificate** window displays to choose certificate location.

Figure 8-6   Import Trusted Certificate

8. Browse to the folder where the certificate is stored and then click **Open**.
9. Click **Save** Wallet on the left side navigation and save the wallet either to the default location (`${ORACLE_INSTANCE}/config/OHS/${COMPONENT_NAME}/keystores/default`) or to desired folder location.
10. Click **Wallet** tab and enable **Auto Login**.

8.1.2 Configure Wallet in ssl.conf file

This topic explains systematic instructions to configure Wallet in the ssl.conf file.

In **ssl.conf** file the newly created wallet need to be updated. This file is located under the folder **${ORACLE_INSTANCE}/config/OHS/${COMPONENT_NAME}/**.

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

   SSLWallet - **${ORACLE_INSTANCE}/config/${COMPONENT_TYPE}/${COMPONENT_NAME}/keystores/**

Figure 8-7   ssl.conf

2. Change the Listen port number in the **ssl.conf** file to the SSL enabled port.

> **Note:**
>
> Listen port number by default: 4443

Figure 8-8 ssl.conf

8.2 SSL Configuration between Oracle HTTP Server and Oracle Weblogic Server

This topic describes the SSL configuration process between the Oracle HTTP server and the Oracle Weblogic server.

SSL for outbound requests from Oracle HTTP Server is configured in **mod_wl_ohs**. Refer to the Weblogic Configuration document for the weblogic server setting mentioned in below topics:

1. #unique_33
2. #unique_34
3. #unique_35

- Turn off KeepAliveEnabled
  This topic explains systematic instructions to turn off KeepAliveEnabled parameter.
- Enable one-way SSL
  This topic explains systematic instructions to enable one-way SSL.
- Enable two-way SSL
  This topic explains systematic instructions to enable two-way SSL.

8.2.1 Turn off KeepAliveEnabled

This topic explains systematic instructions to turn off KeepAliveEnabled parameter.

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

  The KeepAliveEnabled parameter in the **mod_wl_ohs** file is on by default.

Figure 8-9 mod_wl_ohs

8.2.2 Enable one-way SSL

This topic explains systematic instructions to enable one-way SSL.

1. Generate a custom keystore **identity.jks** for Weblogic Server containing a certificate.
2. In the **Settings for AdminServer** screen, set the Keystore details under the **Keystore** tab.

Table 8-1 Keystore- Field Description

Field	Description
Custom Identity Keystore	Specify the **identity.jks** file location
Custom Identity Keystore Type	Specify Keystore type as JKS.
Custom Identity Keystore Passphrase	Specify the passphrase used to create the Keystore.
Confirm Custom Identity Keystore Passphrase	Confirm the passphrase used to create the Keystore.

Figure 8-10 Keystore

3. Copy the certificate to Oracle HTTP Server and import the new certificate into the OHS wallet as a trusted certificate.
4. Add the following new directive in **mod_wl_ohs.conf** to point to the wallet location.

   WlSSLWallet **${ORACLE_INSTANCE}/config/OHS/{COMPONENT_NAME}/keystores/default**
5. Change the port in the **mod_wl_ohs.conf** file to point to the SSL port of the Weblogic server.

Figure 8-11 mod_wl_ohs file

6. Restart both Weblogic Server and Oracle HTTP Server.

8.2.3 Enable two-way SSL

This topic explains systematic instructions to enable two-way SSL.

1. Perform one-way SSL configuration steps as per the topic #unique_34.
2. Generate a new trust store **trust.jks** for the Weblogic server.

   Keystore created for one-way SSL can be used, but it is recommended to create a separate truststore for two-way SSL.
3. Export the user certificate from the Oracle HTTP Server wallet, and import it into the newly created truststore.
4. In the **Settings for AdminServer** screen, set the Keystore details under the **Keystore** tab.

Table 8-2 Keystore - Field Description

Field	Description
Custom Trust Keystore	Specify the **trust.jks** file location.
Custom Trust Keystore Type	Specify the Keystore type as JKS.
Custom Trust Keystore Passphrase	Specify the passphrase used to create the Keystore.
Confirm Custom Trust Keystore Passphrase	Confirm the passphrase used to create the Keystore.

5. Under the **SSL** tab, ensure **Trusted Certificate Authorities** field is set as **from Custom Trust Keystore**.

Figure 8-12 SSL Tab

6. Restart the Weblogic Server.

9
Sample Configuration Files

Refer to the sample configuration files -

- httpd.conf
- mod_wl_ohs.conf
- ssl.conf

10 Start, Stop, and Restart Oracle HTTP Server

This topic explains systematic instructions to Start, Stop, and Restart the Oracle HTTP Server.

- Navigate to the below location in command prompt **${ORACLE_INSTANCE}/bin/** and run the below commands.

Table 10-1 Command- Oracle HTTP Server

Action	Command
Start	**opmnctl startproc ias-component={COMPONENT_NAME}**, For example, **opmnctl startproc ias-component=ohs1**
Stop	**opmnctl stopproc ias-component={COMPONENT_NAME}**, For example, **opmnctl stopproc ias-component=ohs1**
Restart	**opmnctl restartproc ias-component={COMPONENT_NAME}**, For example, **opmnctl restartproc ias-component=ohs1**

11
Test the application

This topic explains systematic instructions to 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>>*

Table 11-1 Server Details

Field	Description
ohs_servername	The server on which OHS is deployed.
ohs_https_port	The port number mentioned against LISTEN directive in **SSL.conf** file.
ohs_http_port	The port number mentioned against LISTEN directive in **httpd.conf** file.

Example - *https://localhost:4443/FCJNeoWeb/welcome.jsp* or *http://localhost:7777/FCJNeoWeb/welcome.jsp*

12 Server Logs Location

This topic describes the location of Oracle HTTP Server Logs.

Oracle HTTP Server Logs are generated under the folder **${ORACLE_INSTANCE}/diagnostics/logs/OHS/{COMPONENT_NAME}/**.

13 References

This topic describes a list of references for Oracle HTTP Server configuration.

The Weblogic Configuration document provided as part of FCUBS installation.

References for Oracle HTTP Server -

- Understanding Oracle HTTP Server Modules
- Configuring SSL in Oracle Fusion Middleware