Oracle® Banking Payments

Payments Weblogic Configuration

Release 14.8.0.0.0

G32382-02

May 2025

Oracle Banking Payments Payments Weblogic Configuration, Release 14.8.0.0.0

G32382-02

Contents

8 Create Resources on Weblogic

1
Preface

1.1 Purpose

This guide is designed to help acquaint you with the Oracle Banking Payments application. This guide provides answers to specific features and procedures that the user need to be aware of the module to function successfully.

1.2 Audience

This manual is intended for the following User/User Roles:

Table 1-1 User Roles

Role	Function
Implementation & IT Staff	Implementation & Maintenance of the Software

1.3 Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

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 make sure 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:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text.
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.

2 Configure SSL on Oracle Weblogic

This topic explains to configure SSL on Oracle WebLogic.

This chapter details out the configurations for SSL on the Oracle Weblogic application server.

- Set up SSL on Oracle Weblogic
  This topic explains in setting up the SSL on Oracle Weblogic.
- Certificates and Keypairs
  This topic explains the Certificates and Keypairs used for validating the authenticity of the server.

2.1 Set up SSL on Oracle Weblogic

This topic explains in setting up the SSL on Oracle Weblogic.

To setup SSL on the Oracle Weblogic application server, you need to perform the following tasks:

1. Obtain an identity (private key and digital certificates) and trust (certificates of trusted certificate authorities) for the Oracle Weblogic application server.
2. Store the identity and trust. Private keys and trust CA certificates are stored in key stores.
3. Configure the identity and trust the key stores for the Oracle Weblogic application server in the administration console.
4. Set SSL attributes for the private key alias and password in the Weblogic Remote console.

   Refer Configuring SSL for more details.

2.2 Certificates and Keypairs

This topic explains the Certificates and Keypairs used for validating the authenticity of the server.

Certificates are used for validating the authenticity of the server. Certificates contain the name of the owner, certificate usage, duration of validity, resource location, or distinguished name (DN), which includes the common name (CN - website address or e-mail address depending on the usage) and the certificate ID of the person who certified (signs) this information. It also contains the public key and a hash to ensure that the certificate has not been tampered with. A certificate is insecure until it is signed. Signed certificates cannot be modified.

A certificate can be self-signed or obtained from a reputable certificate authority such as Verisign, Inc., Entrust.net, Thawte, GeoTrust, or InstantSSL.

SSL uses a pair of cryptographic keys - a **public key** and a **private key**. These keys are similar in nature and can be used alternatively. What one key encrypts can be decrypted by the other key of the pair. The private key is kept secret, while the public key is distributed using the certificate.

A key tool stores the keys and certificates in a keystore. The default keystore implementation implements it as a file. It protects private keys with a password. The different entities (key pairs and the certificates) are distinguished by a unique ‘alias’. Through its keystore, the Oracle Weblogic server can authenticate itself to other parties.

In Java, a keystore is a **java.security.KeyStore** instance that you can create and manipulate using the keytool utility provided with the Java Runtime.

There are two keystores to be managed by the Oracle Weblogic server to configure SSL.

- Identity Keystore: This contains the key pairs and the Digital certificate. This can also contain certificates of intermediate CAs.
- Trust Keystore: Contains the trusted CA certificates.

3 Choose the Identity and Trust Stores

This topic explains to choose the identity and trust stores.

Oracle Financial Services Software recommends that the choice of Identity and Trust stores be made upfront. Oracle Weblogic server supports the following combinations of Identity and Trust stores:

- Custom Identity and Command Line Trust
- Custom Identity and Custom Trust
- Custom Identity and Java Standard Trust
- Demo Identity and Demo Trust

Oracle Financial Services does not recommend choosing Demo Identity and Demo Trust for production environments.

It is recommended to separate the identity and trust stores since each Weblogic server tends to have its own identity but might have the same set of trust CA certificates. Trust stores are usually copied across Oracle Weblogic servers to standardize trust rules; it is acceptable to copy trust stores since they contain public keys and certificates of CAs. Unlike trust stores, identity stores contain private keys of the Oracle Weblogic server and hence should be protected against unauthorized access.

Command Line Trust, if chosen requires the trust store to be specified as a command-line argument in the Weblogic Server startup script. No additional configuration of the trust store is required in the Weblogic Server Administration Console.

Java Standard Trust would rely on the cacerts files provided by the Java Runtime. This file contains the list of trust CA certificates that ship with the Java Runtime and are located in the `JAVA_HOME/jre/lib/security` directory. It is highly recommended to change the default Java standard trust store password and the default access permission of the file. Certificates of most commercial CAs are already present in the Java Standard Trust store. Therefore, it is recommended to use the Java Standard Trust store whenever possible. The rest of the document will assume the use of Java Standard Trust since most CA certificates are already present in it.

One can also create custom trust stores containing the list of certificates of trusted CAs.

For further details on identity and trust stores, please refer to the Oracle Weblogic Server documentation on Securing Oracle Weblogic Server.

4

Obtain the Identity Store

This topic explains in obtaining the identity store.

Create Identity Store with Self-Signed Certificates

Self-signed certificates are acceptable for use in a testing or development environment. Oracle Financial Services does not recommend the use of self-signed certificates in a production environment.

To create a self-signed certificate, the genkeypair option provided by the keytool utility of Sun Java 6 needs to be utilized.

Creation of Self-signed Certificate

Browse to the bin folder of JRE from the command prompt and type the following command.

```
keytool -genkeypair -alias alias -keyalg RSA -keysize 1024 -sigalg
SHA1withRSA -validity 365 -keystore keystore
```

In the above command,

1. ***alias*** is used to identify the public and private key pair created. This alias is required later when configuring the SSL attributes for the managed servers in Oracle Weblogic Server.
2. ***keystore*** is used to specify the location of the JKS file. If no JKS file is present in the path provided, one will be created.

The command will prompt for the following attributes of the certificate and keystore:

1. **Keystore Password:** Specify a password that will be used to access the keystore. This password needs to be specified later when configuring the identity store in Oracle Weblogic Server.
2. **Key Password:** Specify a password that will be used to access the private key stored in the keystore. This password needs to be specified later when configuring the SSL attributes of the managed server(s) in the Oracle Weblogic Server.
3. **First and Last Name (CN):** Enter the domain name of the machine used to access FLEXCUBE UBS, for instance, www.example.com
4. **Name of your Organizational Unit:** The name of the department or unit making the request, for example, BPD. Use this field to identify the SSL Certificate you are creating, for example, by department or by the physical server.
5. **Name of your Organization:** The name of the organization making the certificate request, for example, Oracle Financial Services. It is recommended to use the company or organization's formal name, and this name entered here must match the name found in official records.
6. **Name of your City or Locality:** The city in which your organization is physically located, for example, Mumbai.
7. **Name of your State or Province:** The state/province in which your organization is physically located, for example, Maharashtra.
8. **Two-Letter Country Code for this Unit:** The country in which your organization is physically located, for example, US, UK, IN, etc.

Figure 4-1   Stop image

The key generation algorithm has been specified as RSA, the key size as 1024 bits, the signature algorithm as SHA1withRSA, and the validity days as 365. These can be changed to suitable values if the need arises. For further details, please refer to the documentation of the keytool utility in the JDK utilized by the Oracle Weblogic Server.

Listed below is the result of a sample execution of the command:

```
keytool - genkeypair -alias selfcert -keyalg RSA -keysize 1024 -sigalg
SHA1withRSA -validity 365 -keystore D:\keystores\FCUBSKeyStore.jks

Enter keystore password:<Enter a password to protect the keystore>
Re-enter new password:<Confirm the password keyed above>
What is your first and last name?
[Unknown]:  cvrhp0729.i-flex.com
What is the name of your organizational unit?
  [Unknown]: BPD
What is the name of your organization?
  [Unknown]: Oracle Financial Services
What is the name of your City or Locality?
  [Unknown]:  Mumbai
What is the name of your State or Province?
  [Unknown]:  Maharashtra
What is the two-letter country code for this unit?
  [Unknown]:  IN
Is CN=cvrhp0729.i-flex.com, OU=BPD, O=Oracle Financial Services, L=Mumbai,
ST=Maharashtra, C=IN correct?
  [no]: yes
Enter key password for <selfcert>
(RETURN if same as keystore password):<Enter a password to protect the key>
Re-enter new password:<Confirm the password keyed above>
```

- Create Identity Store with Trusted Certificates Issued by CA
  This topic explains to create identity store with trusted certificates issued by CA.

4.1 Create Identity Store with Trusted Certificates Issued by CA

This topic explains to create identity store with trusted certificates issued by CA.

Create Public and Private Key Pair

Browse to the bin folder of JRE from the command prompt and type the following command.

```
keytool -genkeypair -alias alias -keyalg keyalg -keysize keysize -
sigalg sigalg -validity valDays -keystore keystore
```

In the above command,

1. ***alias*** is used to identify the public and private key pair created. This alias is required later when configuring the SSL attributes for the managed servers in Oracle Weblogic Server.
2. ***keyalg*** is the key algorithm used to generate the public and private key pair. The RSA key algorithm is recommended.

3. ***keysize*** is the size of the public and private key pairs generated. A key size of 1024 or more is recommended. Please consult with your CA on the key size support for different types of certificates.
4. ***sigalg*** is the algorithm used to generate the signature. This algorithm should be compatible with the key algorithm and should be one of the values specified in the Java Cryptography API Specification and Reference.
5. ***valdays*** is the number of days for which the certificate is to be considered valid. Please consult with your CA on this period.
6. ***keystore*** is used to specify the location of the JKS file. If no JKS file is present in the path provided, one will be created.

The command will prompt for the following attributes of the certificate and keystore:

1. **Keystore Password:** Specify a password that will be used to access the keystore. This password needs to be specified later when configuring the identity store in Oracle Weblogic Server.
2. **Key Password:** Specify a password that will be used to access the private key stored in the keystore. This password needs to be specified later when configuring the SSL attributes of the managed server(s) in the Oracle Weblogic Server.
3. **First and Last Name (CN):** Enter the domain name of the machine used to access FLEXCUBE UBS, for instance, www.example.com
4. **Name of your Organizational Unit:** The name of the department or unit making the request, for example, BPD. Use this field to identify the SSL Certificate you are creating, for example, by department or by the physical server.
5. **Name of your Organization:** The name of the organization making the certificate request, for example, Oracle Financial Services. It is recommended to use the company or organization's formal name, and this name entered here must match the name found in official records.
6. **Name of your City or Locality:** The city in which your organization is physically located, for example, Mumbai.
7. **Name of your State or Province:** The state/province in which your organization is physically located, for example, Maharashtra.
8. **Two-Letter Country Code for this Unit:** The country in which your organization is physically located, for example, US, UK, IN, etc.

Listed below is the result of a sample execution of the command:

```
bin>keytool - genkeypair -alias cvrhp0729 -keyalg RSA -keysize 1024 -
sigalg SHA1withRSA -validity 365 -keystore
D:\keystores\FCUBSKeyStore.jks

Enter keystore password:<Enter a password to protect the keystore>
Re-enter new password:<Confirm the password keyed above>
What is your first and last name?
[Unknown]:  cvrhp0729.i-flex.com
What is the name of your organizational unit?
  [Unknown]: BPD
What is the name of your organization?
  [Unknown]: Oracle Financial Services
What is the name of your City or Locality?
  [Unknown]:  Mumbai
What is the name of your State or Province?
  [Unknown]:  Maharashtra
```

```
What is the two-letter country code for this unit?
  [Unknown]:  IN
Is CN=cvrhp0729.i-flex.com, OU=BPD, O=Oracle Financial Services, L=Mumbai,
ST=Maharashtra, C=IN correct?
  [no]: yes
Enter key password for <cvrhp0729>
(RETURN if same as keystore password):<Enter a password to protect the key>
Re-enter new password:<Confirm the password keyed above>
```

Generate CSR

To purchase an SSL certificate, one needs to generate a Certificate Signing Request (CSR) for the server where the certificate will be installed.

A CSR is generated from the server and is the server's unique **fingerprint**. The CSR includes the server's public key, which enables server authentication and secure communication.

> **Note:**
>
> If the keystore file or the password is lost and a new one is generated, the SSL certificate and the private key will no longer match. A new SSL Certificate will have to be requested.

The CSR is created by running the following command in the bin directory of the JRE:

```
keytool -certreq -alias alias –file certreq_file -keystore keystore
```

In the above command,

1. ***alias*** is used to identify the public and private key pair. The private key associated with the alias will be utilized to create the CSR. Specify the alias of the key pair created in the previous step.
2. ***certreq_file*** is the file in which the CSR will be stored.
3. ***keystore*** is the location of the keystore containing the public and private key pair.

Listed below is the result of a sample execution of the command.

```
D:\Oracle\Weblogic14c\jrockit_160_05_R27.6.2-20\bin>keytool -certreq-
alias cvrhp0729 -file D:\keystores\certreq.csr –
keystoreD:\keystores\FCUBSKeyStore.jks

Enter keystore password:[Enter the password used to access the keystore]
Enter key password for <cvrhp0729>
(RETURN if same as keystore password):[Enter the password used to access the
key in the keystore]
```

Obtain Trusted Certificate from CA

The processes of obtaining a trusted certificate vary from one CA to another. The CA might perform additional offline verification. Consult the CA issuing the certificate for details on the process to be followed for submission of the CSR and for obtaining the certificate.

Import Certificate into Identity Store

Store the certificate obtained from the CA in the previous step, in a file, preferably in PEM format. Other formats like the p7b file format would require conversion to the PEM format.

Details on performing the conversion are not listed here. Please refer to the Oracle Weblogic Server documentation on Securing Oracle Weblogic Server for details on converting a Microsoft **p7b** file to the **PEM** format.

The command to be executed for importing a certificate into the identity store depend on whether the trust store is chosen (in the earlier step; see section 2 of this document). It is highly recommended to verify the trust path when importing a certificate into the identity store. The commands provided below assume the use of the Java Standard Trust store.

Import the Intermediate CA certificate
Most Certificate Authorities do not use the root CA certificates to issue identity certificates for use by customers. Instead, Intermediate CAs issue identity certificates in response to the submitted CSRs.

If the Intermediate CA certificate is absent in the Java Standard Trust store, the trust path for the certificate will be incomplete for the certificate, resulting in warnings issued by Weblogic Server during runtime. To avoid this, the intermediate CA certificate should be imported into the identity keystore. Although the intermediate CA certificate can be imported into the Java Standard Trust store, this is not recommended unless the intermediate CA can be trusted.

The following command should be executed to import the intermediate CA certificate into the keystore.

```
keytool -importcert -alias alias -file cert_file -trustcacerts -
keystore keystore
```

In the above command,

1. ***alias*** is used to identify the public and private key pair. Specify the alias of the key pair used to create the CSR in the earlier step.
2. ***cert_file*** is the location of the file containing the intermediate CA certificate in a PKCS#7 format (PEM or DER file).
3. ***keystore*** is the location of the keystore containing the public and private key pair.

The trustcacerts flag is used to consider other certificates (higher intermediaries and the root CA) in the chain of trust. If no chain of trust is established during verification, the certificate will be displayed and one would be prompted to verify it. It is recommended that due diligence be observed when the prompt is displayed to verify a certificate when a chain of trust is absent.

Listed below is a sample execution of the command.

```
bin>keytool - importcert -alias verisigntrialintermediateca – file
D:\keystores\VerisignIntermediateCA.cer -trustcacerts -
keystoreD:\keystoreworkarea\FCUBSKeyStore.jks

Enter keystore password:<Enter the password used to access the keystore>
```

Certificate was added to keystore.

Import the Identity Certificate
The following command should be executed to import the identity certificate into the keystore.

```
keytool -importcert -alias alias -file cert_file -trustcacerts -
keystore keystore
```

In the above command,

1. ***alias*** is used to identify the public and private key pair. Specify the alias of the key pair used to create the CSR in the earlier step.

2. ***cert_file*** is the location of the file containing the PKCS#7 formatted reply from the CA, containing the signed certificate.
3. ***keystore*** is the location of the keystore containing the public and private key pair.

The trustcacerts flag is used to consider other certificates (intermediate CAs and the root CA) in the chain of trust. If no chain of trust is established during verification, the certificate will be displayed and one would be prompted to verify it. It is recommended that due diligence be observed when the prompt is displayed to verify a certificate when a chain of trust is absent.

Listed below is a sample execution of the command.

```
\bin>keytool - importcert -alias cvrhp0729 -file
D:\keystores\cvrhp0729.cer - trustcacerts -keystore
D:\keystoreworkarea\FCUBSKeyStore.jks
```

```
Enter keystore password:<Enter the password used to access the keystore>
Enter key password for <cvrhp0729>:<Enter the password used to access the
private key>
```

Certificate was added to keystore.

The previous set of commands assumed the presence of the appropriate root CA certificate (in the chain of trust) in the Java Standard Trust store, i.e. in the cacerts file. If the CA issuing the identity certificate (for the Weblogic Server) does not have the root CA certificate in the Java Standard Trust store, one can opt to import the root CA certificate into cacerts, or the identity store, depending on factors including the trustworthiness of the CA, the necessity of transporting the trust store across the machine, among others.

5 Configure Identity and Trust Stores for Weblogic

This topic explains to configure identity and trust stores for Weblogic.

- Enable SSL on Oracle Weblogic Server
  This topic explains the systematic instructions to enable SSL on Oracle Weblogic Server.
- Configure Identity and Trust Stores
  This topic explains the systematic instructions to configure identity and trust stores.

5.1 Enable SSL on Oracle Weblogic Server

This topic explains the systematic instructions to enable SSL on Oracle Weblogic Server.

To configure SSL on the Oracle Weblogic server, login into the Admin Console and follow the steps given below:

1. In the **Edit Tree**, go to **Environment**, then **Servers**.
2. Select the name of the server for which you want to enable SSL (example - exampleserver).
3. Enable **Listen Port Enabled** and specify a port number.
4. Enable **SSL Listen Port Enabled** and specify a port number.
5. Click **Save**.

5.2 Configure Identity and Trust Stores

This topic explains the systematic instructions to configure identity and trust stores.

To configure the Identity and Trust stores in Oracle Weblogic Server, login to the Admin Console of Weblogic Server.

1. In the **Edit Tree**, go to **Environment**, then **Servers**, then **myServer**.
2. Click the **Security** tab, then the **Keystores** tab.
3. From the Keystores drop-down list, select a method for storing and managing private keys/digital certificate pairs and trusted CA certificates. This choice should match the one made in Configure SSL on Oracle Weblogic section of this document (Choosing the Identity and Trust Stores).
4. Define attributes for the identity and trust keystores. Depending on the keystore type that you selected, different options are available.
5. In the **Identity** section, provide the following details:
   a. **Custom Identity Keystore File Name:** Fully qualified path to the Identity keystore.
   b. **Custom Identity Keystore Type:** Set this attribute to JKS, the type of the keystore. If left blank, it defaults to JKS (Java KeyStore).

   c. **Custom Identity Keystore PassPhrase:** The password you enter when reading or writing to the keystore. This attribute is optional or required depending on the type of keystore. All keystores require the passphrase to write to the keystore. However, some keystores do not require the passphrase to read from the keystore. Oracle Weblogic Server only reads from the keystore. So whether or not you define this property depends on the requirements of the keystore.

6. In the **Trust** section, provide the following details:

   If you choose **Java Standard Trust**, specify the password used to access the trust store.

   If you choose **Custom Trust**, the following attributes have to be provided:

   a. **Custom Trust Keystore**: The fully qualified path to the trust keystore.

   b. **Custom Trust Keystore Type**: Set this attribute to JKS, the type of the keystore. If left blank, it defaults to JKS (Java KeyStore).

   c. **Custom Trust Keystore Passphrase**: The password you enter when reading or writing to the keystore. This attribute is optional or required depending on the type of keystore. All keystores require the passphrase to write to the keystore. However, some keystores do not require the passphrase to read from the keystore. Oracle Weblogic Server only reads from the keystore. So, whether or not you define this property depends on the requirements of the keystore.

   > **Note:**
   >
   > When identity and trust stores are of the JKS format, the passphrases are not required.

7. Click **Save**.

8. Repeat on all applicable servers.

6

Set SSL Attributes for Managed Servers

This topic explains to set SSL attributes for managed servers.

- Set SSL Attributes for Private Key Alias and Password
  This topic explais the systematic instruction to set SSL attributes for private key alias and password.

6.1 Set SSL Attributes for Private Key Alias and Password

This topic explais the systematic instruction to set SSL attributes for private key alias and password.

To configure the private key alias and password, login to the Oracle Weblogic Server Admin Console.

1. In the **Edit Tree**, go to **Environment**, then **Servers**, then **myServer**.
2. Click the **Security** tab, then the **SSL** subtab.
3. Set SSL/TLS attributes for the private key alias and password.
4. In the **Server Private Key Alias** field, enter the keystore attribute that defines the string alias used to store and retrieve the server’s private key. Set this attribute to the alias name defined for the key pair when creating the key pair in the Identity keystore.
   - In the **Server Private Key Pass Phrase** field, enter the keystore attribute that defines the passphrase used to retrieve the server’s private key. The password defined for the key pair (alias_password) at the time of its creation. Confirm the password.
5. If required, click **Show Advanced Fields** to set additional attributes.
6. Click **Save**.
7. Repeat on all applicable servers.
8. Go to the **Controls** tab, check the appropriate server, and click **Restart SSL**. Confirm when it prompts.
9. In the **Monitoring Tree**, go to **Environment**, then **Servers**.
10. Select the checkbox for each server where you want to restart SSL/TLS.
11. Click **Restart SSL** to restart the SSL/TLS listen sockets to apply changes to your keystore.

7

Test Configuration

This topic explains to test the configuration

Once the Oracle Weblogic has been configured for SSL, deploy the application in the usual manner. The application can be tested in SSL mode after deployment. To launch the application in SSL mode, enter the URL in the following format:

https://(Machine Name):(SSL_Listener_port_no)/(Context_root)

> **Note:**
>
> It is recommended that the Oracle Banking payments web application be accessed via the HTTPS channel instead of the HTTP channel.

8

Create Resources on Weblogic

This topic explains the steps to deploy the FC payments application and gateway application in the application server.

- Resource Administration
  This section deals with the process of resource administration on Oracle Weblogic.
- Configure Weblogic for Oracle Banking Payments
  This section explains the systematic instructions to configure the Oracle WebLogic application server for Oracle Banking Payments.
- Setup/Configure Mail Session in Weblogic
  This topic explains to setup/configure mail sessions in Weblogic.

8.1 Resource Administration

This section deals with the process of resource administration on Oracle Weblogic.

All the resources mention in **Resources To be Created** document are need to be created before deployment. One example for each category is explained in the following subsections.

- Create Data Source
  This topic helps to create data source.

8.1.1 Create Data Source

This topic helps to create data source.

The method for creating data sources is explained under the following headings.

Prerequisites

To create the data source, the OCI needs to be enabled. For this, download Oracle Instant Client and install it. The details are given below:

Table 8-1 Oracle Instant Client

Package	Download Location	Remarks
Oracle Instant Client Package	https://www.oracle.com/database/technologies/instant-client/downloads.html	Install Oracle Instant Client in a local directory. While configuring Weblogic for Windows or Unix/ Linux box, the user needs to provide the directory path where Instant Client is installed.

The user needs to do the data source configuration with the OCI driver enabled. The configurations are given below.

- Oracle Weblogic on Windows Box:
  - Set **{ORACLE_HOME}** in the environment variable.

  - Update the Environment Variable Path as `{ORACLE_HOME}/Instance Client`. This is required to load all the **.dll** files.
  - Ensure that the **ojdbc*.jar** file in `{WL_HOME}/server/lib/ojdbc*.jar` is the same as the file `{ORACLE_HOME}/jdbc/lib/ojdbc*.jar`. This is required for ensuring compatibility.
  - Update PATH in **StartWebLogic.bat** or **setDomainEnv.bat**. This must be the directory path where Oracle Instant Client is installed.
- Oracle Weblogic on Unix/Linux Box:
  - Set **{ORACLE_HOME}** in the environment variable.
  - Update the environment variable LD_LIBRARY_PATH as `{ORACLE_HOME}/lib`. This is to load all the **.so** files.
  - Ensure that the **ojdbc*.jar** file in `{WL_HOME}/server/lib/ojdbc*.jar` is the same as the file `{ORACLE_HOME}/jdbc/lib/ojdbc*.jar`. This is to ensure compatibility.
  - Update LD_LIBRARY_PATH in **StartWeblogic.sh** or **setDomainEnv.sh**. This must be the directory path where Oracle Instant Client is installed.
  - If you are still not able to load the **.so** files, then you need to update the EXTRA_JAVA_PROPERTIES by setting Djava.library.path as `{ORACLE_HOME}/lib` in **StartWebLogic.sh** or **setDomainEnv.sh**.

To create the data source, follow the steps given below:

1. Login to the WebLogic Remote Console of the WebLogic application server.
2. In the **Edit Tree**, go to **Services**, then **Data Sources**, Click **New**.
3. Specify a name for the new data source. For Ex: fcjdevDS_XA
4. On the **Create a New JDBC Data Source** screen, specify the fields.
5. In the **JNDI Names** field, enter the JNDI path to the location where this JDBC data source will be bound. Applications look up the data source on the JNDI tree by this name when reserving a connection.
6. Choose the server instances or clusters where you want to deploy the data source.
7. Select **Generic Data Source** from the **Data Source Type** drop-down list.
8. From the **Database Type** drop-down list, select the database management system (DBMS) of the database that you want to connect to. For Ex: jdbc/fcjdevDS_XA
9. From the **Database Driver** drop-down list, select a XA or non-XA JDBC driver
   - For **XA Enabled Data Source**: Oracle's Driver (Thin XA) for Instance Connections, Versions Any (oracle.jdbc.xa.client.OracleXADataSource)
   - For **Non-XA Enabled Data Source**: Oracle's Driver (Thin) for Instance Connections, Versions Any (oracle.jdbc.OracleDriver)
10. Enter the connection details for the database that you want to connect to:
    - **Database Name**: Specify the name of the database you want to connect to.
    - **Host Name**: Specify the DNS name or IP address of the server that hosts the database.
    - **Port**: Specify the port on which the database server listens for connections requests.
    - **Database User Name**: Specify the database user account name that you want to use for connections in the data source.

- **Password**: Specify the password for the database user account.

11. Click **Create**.
12. Follow the above steps to create all the data sources listed in Resources To Be Created

> **Note:**
>
> For *GTXN datasources **- Global Transactions Protocol** should be updated as “LoggingLastResource” under the **Transaction** tab.

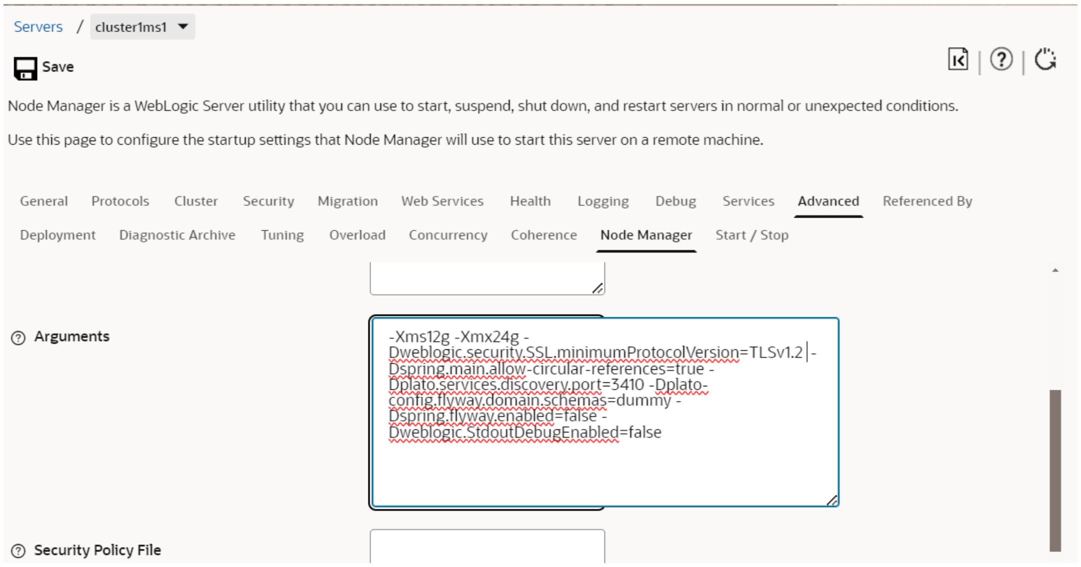
8.2 Configure Weblogic for Oracle Banking Payments

This section explains the systematic instructions to configure the Oracle WebLogic application server for Oracle Banking Payments.

To configure the Oracle WebLogic application server for Oracle Banking Payments, follow the steps given below:

1. Managed Server Startup Arguments
   - In the **Edit Tree**, go to **Environment**, then **Servers**.
   - Select the Managed Server that you want to configure startup arguments.
   - On the **Advanced** tab, select the **Node Manager** subtab. Specify the **Arguments**.

Figure 8-1 Arguments



Node Manager is a WebLogic Server utility that you can use to start, suspend, shut down, and restart servers in normal or unexpected conditions.

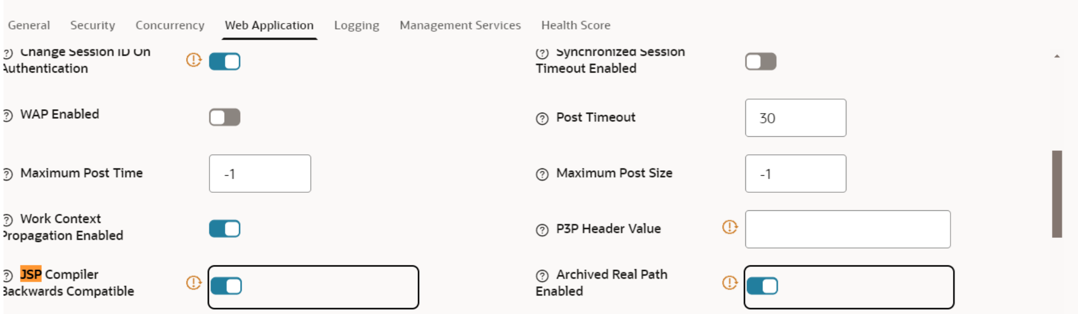
Use this page to configure the startup settings that Node Manager will use to start this server on a remote machine.

Arguments

```
-Xms12g -Xmx24g -
Dweblogic.security.SSL.minimumProtocolVersion=TLSv1.2 -
Dspring.main.allow-circular-references=true -
Dplato.services.discovery.port=3410 -Dplato-
config.flyway.domain.schemas=dummy -
Dspring.flyway.enabled=false -
Dweblogic.StdoutDebugEnabled=false
```

2. Domain Configuration
   - In the **Edit Tree**, go to **Environment**, then **Domain**, under **Web Applications** tab.
   - Check the options **JSP Compiler Backwards Compatible** and **Archived Real Path Enabled**.

Figure 8-2 Options



Change Session ID On Authentication: on

Synchronized Session Timeout Enabled: off

WAP Enabled: off

Post Timeout: 30

Maximum Post Time: -1

Maximum Post Size: -1

Work Context Propagation Enabled: on

P3P Header Value:

JSP Compiler Backwards Compatible: on

Archived Real Path Enabled: on

8.3 Setup/Configure Mail Session in Weblogic

This topic explains to setup/configure mail sessions in Weblogic.

This section describes the set of configurations changes required in the Oracle Weblogic Server when Oracle Banking Payments is configured to generate and send passwords to users via e-mail.

- Create JavaMail Session
  This topic explains the systematic instructions to create JavaMail session.
- Configuration of the TLS/SSL Trust Store for Weblogic Server
  This topic explains the configuration of the TLS/SSL Trust Store for Weblogic Server.

8.3.1 Create JavaMail Session

This topic explains the systematic instructions to create JavaMail session.

To configure the JavaMail session, follow the steps below.

1. Login to WebLogic Remote Console of the WebLogic application server.
2. In the **Edit Tree**, go to **Services**, then **Mail Sessions**.
3. Click **New**.
4. Specify a **Name** and a **JNDI Name** for the mail session.

   ```
   Applications use the JNDI Name to look up the mail session. For
   example, if you enter myMailSession as the JNDI name, applications
   perform the following look up: InitialContext ic = new
   InitialContext();Session session = (Session)
   ic.lookup("myMailSession");
   ```
5. Click **Create**.
6. In the **Session Username** field, specify the user account to use to create an authenticated JavaMail session. Then, in the **Session Password** field, enter he password for the user account.
7. On the **Targets** tab, move the servers or clusters that you want this mail session to target over to **Chosen**.
8. On the **Java Mail Properties** tab, in the **Java Mail Properties** table, click **+** to add a new row. Add the properties listed in below table
   - Double-click the cell under **Properties Name** and specify a name for the property.
   - Double-click the cell under **Properties Value** and specify a value for the property.

9. Click **Save**.

Table 8-2 Create a New Mail Session

Field	Description
Name	Specify the name as **FCUBSMailSession**.
JNDI Name	Specify the JNDI Name as **mail/FCUBSMail**. **Note:** This JNDI name needs to be maintained in fcubs.properties file with encrypted format.
Java Mail Properties	Specify the following mail properties. • mail.host=<HOST_MAIL_SERVER> • mail.smtps.port=<SMTPS_SERVER_PORT> (For example: 1010) • mail.transport.protocol=<MAIL_TRANSFER_PROTOCOL>(For Example: smtps) • mail.smtps.auth=true • mail.smtps.host==<HOST_SMTPS_MAIL_SERVER>

10. fcubs.properties file needs to be updated with the encrypted values of
    - SMTP_HOST
    - SMTP_USER
    - SMTP_PASSWORD
    - SMTP_JNDI

    This can be achieved using the Oracle Banking UBS Installer.

8.3.2 Configuration of the TLS/SSL Trust Store for Weblogic Server

This topic explains the configuration of the TLS/SSL Trust Store for Weblogic Server.

As described in the previous section, Oracle Banking Payments uses SMTPS to send outgoing mails. SMTPS uses SSL to ensure transport-level security of the mail messages and hence the certificate of the mail server needs to be imported into the trust store(s) of the Managed Servers where Oracle Banking Payments is deployed.

The certificate of the mail server needs to be specifically imported into the trust store configured for the Managed Server(s), as configured in the Oracle Banking Payments Installation guide titled SSL Configuration On Weblogic (SSL_Configuration).

For further details on importing the certificate of the mail server into the trust store, refer to the documentation for the Sun Java keytool utility (Key and Certificate Management tool).