Oracle® Banking Cash Management

SSL Setup Guide

Release 14.7.1.0.0
F77022-01
May 2023

ORACLE

Oracle Banking Cash Management SSL Setup Guide, Release 14.7.1.0.0

F77022-01

Contents

Preface

Purpose

This guide provides the information about the configurations of SSL for OracleWebLogic application server.

Audience

This guide is intended for the WebLogic admin or ops-web team who are responsible for installation of the Oracle Financial Services Software banking products.

Acronyms and Abbreviations

The list of the acronyms and abbreviations used in this guide are as follows:

Table 1 List of Acronyms and Abbreviations

Abbreviation	Description
CSR	Certificate Signing Request

List of Topics

This guide is organized as follows:

Table 2 List of Topics

Topics	Description
Configure SSL on Oracle WebLogic	This topic provides the information for configuring SSL on Oracle Weblogic.
Choose the Identity and Trust Stores	This topic provides the information for choosing the identity and trust Stores.
Obtain the Identity Store	This topic provides the information for obtaining the identity store.
Configure Identity and Trust Stores for Weblogic	This topic provides the information for configuring the identity and trust Stores for Weblogic.
Configure Weblogic Console	This topic provides the information for configuring the Weblogic Console.
Configure SSL Mode in Node Manager for Clustered Environment	This topic provides the information for configuring SSL mode in node manager for clustered environment.
Set SSL Attributes for Managed Servers	This topic provides the information for setting the SSL attributes for managed servers.
Test Configuration	This topic provides the information for testing configuration.

1

Configure SSL on Oracle WebLogic

This topic provides the information about the configurations for SSL on Oracle Weblogic application server.

- Setup SSL on Oracle Weblogic
  This topic provides the systematic instructions to setup the SSL on Oracle WebLogic.
- Certificates and Keypairs
  This topic provides the information about the certificates and keypairs.

1.1 Setup SSL on Oracle Weblogic

This topic provides the systematic instructions to setup the SSL on Oracle WebLogic.

1. Obtain an identity (private key and digital certificates) and trust (certificates of trusted certificate authorities) for Oracle WebLogic application server.
2. Store the identity and trust.

   The private keys and trust CA certificates are stored in keystores.
3. Configure the identity and trust the keystores for Oracle WebLogic application server in the administration console.
4. Set SSL attributes for the private key alias and password in Oracle WebLogic administration console.

1.2 Certificates and Keypairs

This topic provides the information about the certificates and keypairs.

The certificates are used for validating the authenticity of the server. Certificates contains the name of the owner, certificate usage, duration of validity, resource location, or distinguished name (DN), which includes the common name (CN - web site address or e-mail address depending of the usage) and the certificate ID of the person who certified (signs) these 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, Digicert Inc., GeoTrust or InstantSSL.

The SSL uses a public key and a private key cryptographickeys. 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 keytool 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, Oracle WebLogic server can authenticate itself to other parties.

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

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

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

2

Choose the Identity and Trust Stores

This topic provides the information for choosing the identity and trust stores.

Oracle Financial Services Software recommends that the choice of Identity and Trust stores be made up front. 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 recommend 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 OracleWebLogic server, and hence should be protected against unauthorized access.

Command Line Trust, if choosen 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 is 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 the Oracle WebLogic Server documentation on Securing Oracle WebLogic Server.

3

Obtain the Identity Store

This topic provides the information for obtaining the identity store.

- Create Identity Store with Self-Signed Certificates
  This topic provides the information to create the identity store with self-signed certificates.
- Keystore Creation
  This topic provides the information to create the keystore.
- Create Identity Store with Trusted Certificates Issued by CA
  This topic provides the information to create Identity Store with Trusted Certificates Issued by CA.
- Export Private Key as Certificate
  This topic provides the information to export private key as certificate.
- Import Trust Certificate
  This topic provides the information to import as trusted certificate.

3.1 Create Identity Store with Self-Signed Certificates

This topic provides the information to create the 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.

Create Self-Signed Certificate

Browse to the bin folder of JRE from the command prompt and type the following command. The items highlighted are placeholders, and should be replaced with suitable values when running the command.

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

Table 3-1 Keyword Description

Keyword	Description
alias	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.
keystore	It 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 prompts for the following attributes of the certificate and keystore:

Table 3-2 Attributes Details

Attributes	Description
Keystore Password	Specify a password used to access the Keystore. This password needs to be specified later when configuring the identity store in Oracle Weblogic Server.
Key Password	Specify a password 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 Oracle Weblogic Server.
First and Last Name (CN)	Enter the domain name of the machine used to access Oracle Banking Virtual Account Management. For example, www.example.com.
Name of your Organizational Unit	The name of the department or unit making the request. For example, BDP. Use this field to further identify the SSL Certificate for creating. For example, by department or by physical server.
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.
Name of your City or Locality	The city in which your organization is physically located. For example, Mumbai.
Name of your State or Province	The state/province in which your organization is physically located. For example, Maharashtra.
Two-letter Country Code for this Unit	The country in which your organization is physically located. For example, US, UK, IN, etc.

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 Oracle WebLogic Server.

The sample execution command is listed as follows:

```
D:\Sample\weblogic11g\jrockit_160_05_R27.6.2-20\bin>keytool -
genkeypair -alias selfcert -keyalg RSA -keysize 1024 -sigalg
SHA1withRSA -validity 365 -keystore D:\keystores\AdminOBVAMKeyStore.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.oracle.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>
```

3.2 Keystore Creation

This topic provides the information to create the keystore.

```
keytool -genkeypair -keystore <keystore_name.jks> -alias <alias_name> -dname
"CN=<hostname>, OU=<Organization Unit>, O=<Organization>, L=<Location>,
ST=<State>,
C=<Country_Code>" -keyalg <Key Algorithm> -sigalg <Signature Algorithm> -
keysize <key size>
-validity <Number of Days> -keypass <Private key Password> -storepass <Store
Password>
```

Example:

```
keytool -genkeypair -keystore AdminOBVAMKeyStore.jks -alias OBVAMCert -dname
"CN=ofss00001.in.example.com, OU=OFSS, O=OFSS, L=Chennai, ST=TN, C=IN" -
keyalg "RSA"
-sigalg "SHA1withRSA" -keysize 2048 -validity 3650 -keypass Password@123 -
storepass Password@123
```

> **Note:**
>
> CN=ofss00001.in.example.com is the Host Name of the weblogic server

3.3 Create Identity Store with Trusted Certificates Issued by CA

This topic provides the information 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. The items highlighted are placeholders, and should be replaced with suitable values when running the command.

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

Table 3-3 Keyword Description

Keyword	Description
`alias`	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.
`keyalg`	It is a key algorithm used to generate the public and private key pair. The RSA key algorithm is recommended.
`keysize`	It 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.
`sigalg`	It 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.
`valdays`	It is the number of days for which the certificate is to be considered valid. Please consult with your CA on this period.
`keystore`	It 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 prompts for the following attributes of the certificate and keystore:

Table 3-4 Attribute Details

Attributes	Description
Keystore Password	Specify a password used to access the Keystore. This password needs to be specified later, when configuring the identity store in Kafka server.
Key Password	Specify a password 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 Oracle Weblogic Server.
First and Last Name (CN)	Enter the domain name of the machine used to access Oracle Banking Virtual Account Management. For example, www.example.com.
Name of your Organizational Unit	The name of the department or unit making the request. For example, BDP. Use this field to further identify the SSL Certificate for creating. For example, by department or by physical server.
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.
Name of your City or Locality	The city in which your organization is physically located. For example, Mumbai.
Name of your State or Province	The state/province in which your organization is physically located. For example, Maharashtra.

Table 3-4 (Cont.) Attribute Details

Attributes	Description
Two-letter Country Code for this Unit	The country in which your organization is physically located. For example, US, UK, IN, etc.

The sample execution of the command is listed below:

```
D:\Oracle\weblogic11g\jrockit_160_05_R27.6.2-20\bin>keytool -genkeypair -
alias cvrhp0729 -keyalg RSA -keysize 1024 -sigalg SHA1withRSA -validity 365 -
keystore
D:\keystores\AdminOBVAMKeyStore.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, the user must generate the 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. 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
```

Table 3-5 Keyword Description

Keyword	Description
alias	Used to identify the public and private key pair created. 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.
certreq_file	It is the file in which the CSR will be stored.
keystore	It is the location of the keystore containing the public and private key pair.

The sample execution command is listed below:

```
D:\Oracle\Weblogic11g\jrockit_160_05_R27.6.2-20\bin>keytool -certreq -
alias cvrhp0729 -file D:\keystores\certreq.csr -keystore
D:\keystores\AdminOBVAMKeyStore.jks
Enter keystore password: <Enter a password to protect the keystore>
Enter key password for <cvrhp0729>[Enter the password used to access
the key in the keystore]
```

3.5 Import Trust Certificate

This topic provides the information to import as trusted certificate.

```
keytool -import -v -trustcacerts -alias rootcacert -file
<export_certificate_file_name_with_location.cer>
-keystore <keystore_name.jks> > -keypass <Private key Password> -
storepass <Store Password>
```

Example:

```
keytool -import -v -trustcacerts -alias rootcacert -file
AdminOBVAMCert.cer
-keystore AdminOBVAMKeyStore.jks -keypass Oracle123 -storepass
Oracle123
```

3.4 Export Private Key as Certificate

This topic provides the information to export private key as certificate.

```
keytool -export -v -alias <alias_name> -file
<export_certificate_file_name_with_location.cer>
-keystore <keystore_name.jks> > -keypass <Private key Password> -
storepass <Store Password>
```

Example:

```
keytool -export -v -alias OBVAMCert -file AdminOBVAMCert.cer -keystore
AdminOBVAMKeyStore.jks
-keypass Oracle123 -storepass Oracle123
```

If successful, the following message is displayed:

Certificate stored in file < AdminOBVAMCert.cer>

- Obtain Trusted Certificate from CA
  This topic provides the information to obtain the trusted certificate from CA.
- Convert .crt and .key file to PKC12 file
  This topic provides the systematic instruction to convert .crt and .key file to PKC12 file.
- Import Certificate into Identity Store
  This topic provides information to import the certificate into identify store.

3.4.1 Obtain Trusted Certificate from CA

This topic provides the information to obtain the 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.

3.4.2 Convert .crt and .key file to PKC12 file

This topic provides the systematic instruction to convert .crt and .key file to PKC12 file.

1. Once CA signed certificate is generated, you will be notified on email to download it from the portal, once zip is downloaded from portal it will contain .crt & .key file for requested server.
2. Convert this .crt and .key file to PKC12 file using openssl command (openssl.exe file could be found under Git installation directory. In my case, “C:\Program Files\Git\usr\bin\openssl.exe”)

   Sample command:

   ```
   openssl pkcs12 -export -in <crt_file> -inkey <key_file> -out <p12_file>
    -name <alias_name>
   ```

   Sample command with values:

   ```
   openssl pkcs12 -export -in whf00pfl.in.example.com.crt -inkey
    whf00pfl.in.example.com.key -out whf00pfl.in.example.com.p12 -name
   whf00pfl.in.example.com
   ```

3. Import PKCS12 file into Java Keystore using the keytool (keytool can be found under $JAVA_HOME/bin)

Sample Command:

```
keytool -importkeystore -deststorepass <password> -destkeystore
 <jks_file> -srckeystore <p12_file> -srcstoretype PKCS12
```

Sample command with values:

```
keytool -importkeystore -deststorepass Oracle@123 -destkeystore
 whf00pfl_keystore.jks -srckeystore whf00pfl.in.example.com.p12 -
 srcstoretype PKCS12
```

3.4.3 Import Certificate into Identity Store

This topic provides information to import the certificate into identify 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.

> **Note:**
>
> 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 chosen (Refer to section **Choose the Identity and Trust Stores**). 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.

1. Import the public certificate into the keystore using the private key alias. Then do the actual import of the certificate:
   Sample command:

   ```
   keytool -importcert -v -alias <alias_name> -file
    <consolidate_certs_file> -keystore <keystore path> -keypass
   <Password>
    -storepass <Password>
   ```

   Sample command with values:

   ```
   keytool -importcert -v -alias whf00pfl.in.example.com -file
    consolidate_cert.pem -keystore whf00pfl_keystore.jks -keypass
    Oracle@123 -storepass Oracle@123
   ```

2. To confirm your keystore is created correctly, you can look at the keystore using the following command:

Sample command:

```
keytool -list -v -keystore <keystore path> -storepass <Password>
```

Sample command with values:

```
keytool -list -v -keystore whf00pfl_keystore.jks -storepass Oracle@123
```

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 must be executed to import the intermediate CA certificate into the keystore.

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

Table 3-6 Keyword Description

Keyword	Description
alias	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.
certreq_file	It is the location of the file containing the intermediate CA certificate in a PKCS#7 format (PEM or DER file).
keystore	It 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.

The sample execution command is listed below:

```
D:\Sample\weblogic11g\jrockit_160_05_R27.6.2-20\bin>keytool - importcert -
alias verisigntrialintermediateca -file
D:\keystores\VerisignIntermediateCA.cer -trustcacerts -keystore
D:\keystoreworkarea\AdminOBVAMKeyStore.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
```

Table 3-7 Keyword Description

Keyword	Description
alias	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.
certreq_file	It is the location of the file containing the PKCS#7 formatted reply from the CA, containing the signed certificate.
keystore	It 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.

The sample execution command is listed below:

```
D:\Oracle\weblogic11g\jrockit_160_05_R27.6.2-20\bin>keytool -
importcert -alias cvrhp0729 -file
D:\keystores\cvrhp0729.cer - trustcacerts -keystore
D:\keystoreworkarea\AdminOBVAMKeyStore.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 reply was installed in 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 into the identity store, depending on factors including trustworthiness of the CA, necessity of transporting the trust store across machine, among others.

4

Configure Identity and Trust Stores for Weblogic

This topic provides the information to configure Identity and Trust Stores for Weblogic.

- Enable SSL on Oracle WebLogic Server
  This topic provides the systematic instructions to enable the SSL on Oracle WebLogic Server.
- Configure Identity and Trust Stores
  This topic provides the systematic instructions to configure the identity and trust store for WebLogic.

4.1 Enable SSL on Oracle WebLogic Server

This topic provides the systematic instructions to enable the SSL on Oracle WebLogic Server.

Login to the **Oracle WebLogic Admin Console** to configure SSL.

1. Under **Change Center**, click **Lock & Edit**.
2. Expand **Servers** node.
3. Select the name of the server for which you want to enable SSL.

   Example: example server
4. Navigate to **Configuration** and select **General** tab.
5. Select the **SSL Listen Port Enabled** option and specify the SSL listen port.
6. In **Listen Address** field, specify the hostname of the machine in which the application server is installed.

4.2 Configure Identity and Trust Stores

This topic provides the systematic instructions to configure the identity and trust store for WebLogic.

Login in to the **Oracle WebLogic Admin Console**.

1. Under **Change Center**, click **Lock & Edit**.
2. Expand **Servers** node.
3. Select the name of the server to configure the keystores.

   Example: exampleserver
4. Go to **Configuration** and select **Keystores** tab.
5. In the filed **Keystores**, select the method for storing and managing private keys/digital certificate pairs and trusted CA certificates.

This choice should match the one made in **Choose the Identity and Trust Stores** section of this document.

6. In the **Identity** section, provide the following details:
   - **Custom Identity Keystore File Name**: Fully qualified path to the Identity keystore.
   - **Custom Identity Keystore Type**: Set this attribute to JKS, the type of the keystore. If left blank, it is defaulted to JKS (Java KeyStore).
   - **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 in order 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.
7. In the **Trust** section, provide the following details:

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

   If the user choose **Custom Trust**, the following attributes have to be provided:
   - **Custom Trust Keystore**: The fully qualified path to the trust keystore.
   - **Custom Trust Keystore Type**: Set this attribute to JKS, the type of the keystore. If left blank, it defaults to JKS (Java KeyStore).
   - **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 in order 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:**
>
> If the identity and trust stores are in the JKS format, the passphrases are not required.

5

Configure Weblogic Console

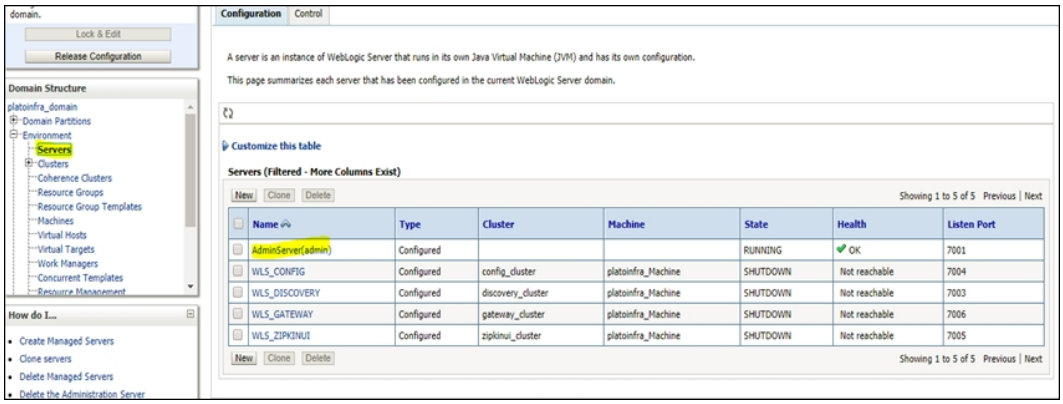
This topic provides the systematic instructions to configure the Weblogic Console.

After domain creation, follow the below steps to enable SSL in Weblogic Admin server.

Login to the **Oracle Weblogic Server Admin Console**.

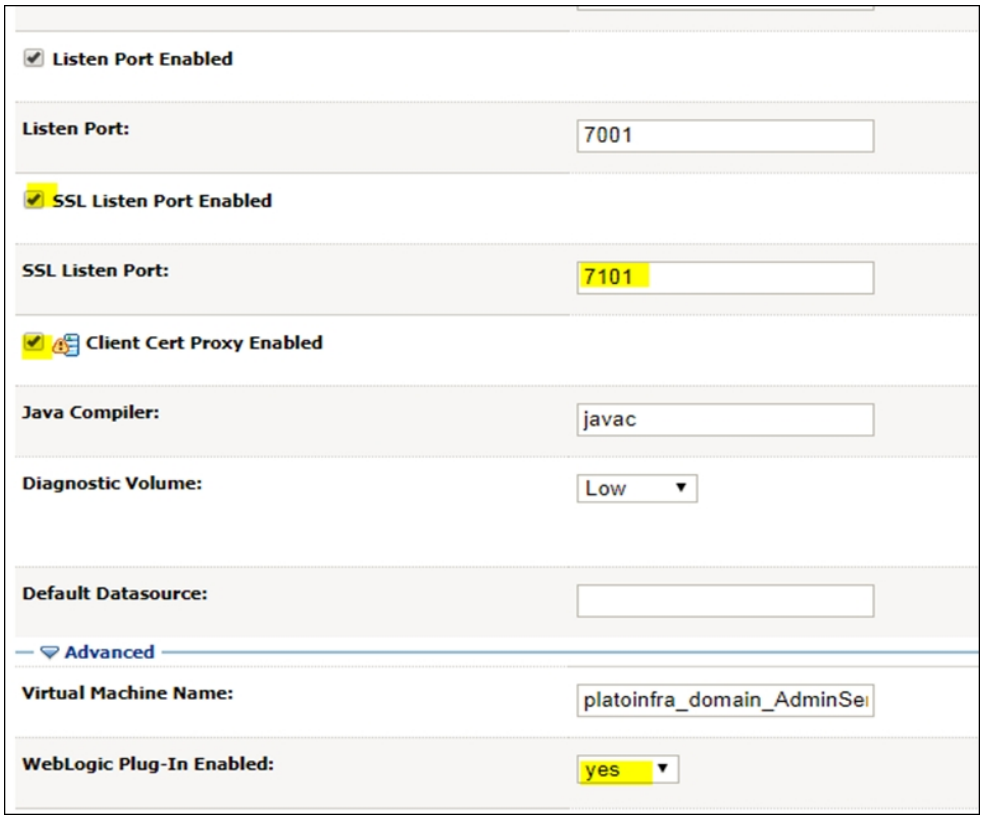
1. Select **Admin Server** to enable SSL Options.

Figure 5-1 Configuration



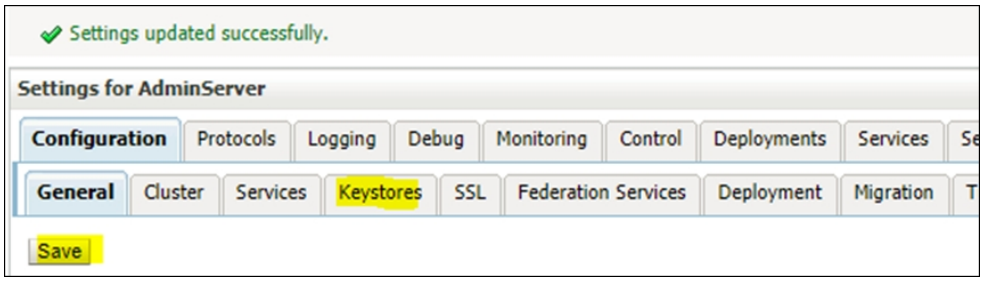
2. Click **General** tab.
3. Select **SSL Listen Port Enabled**, **Client Cert Proxy Enabled**, and **Weblogic Plug-In Enabled**.

Figure 5-2 General



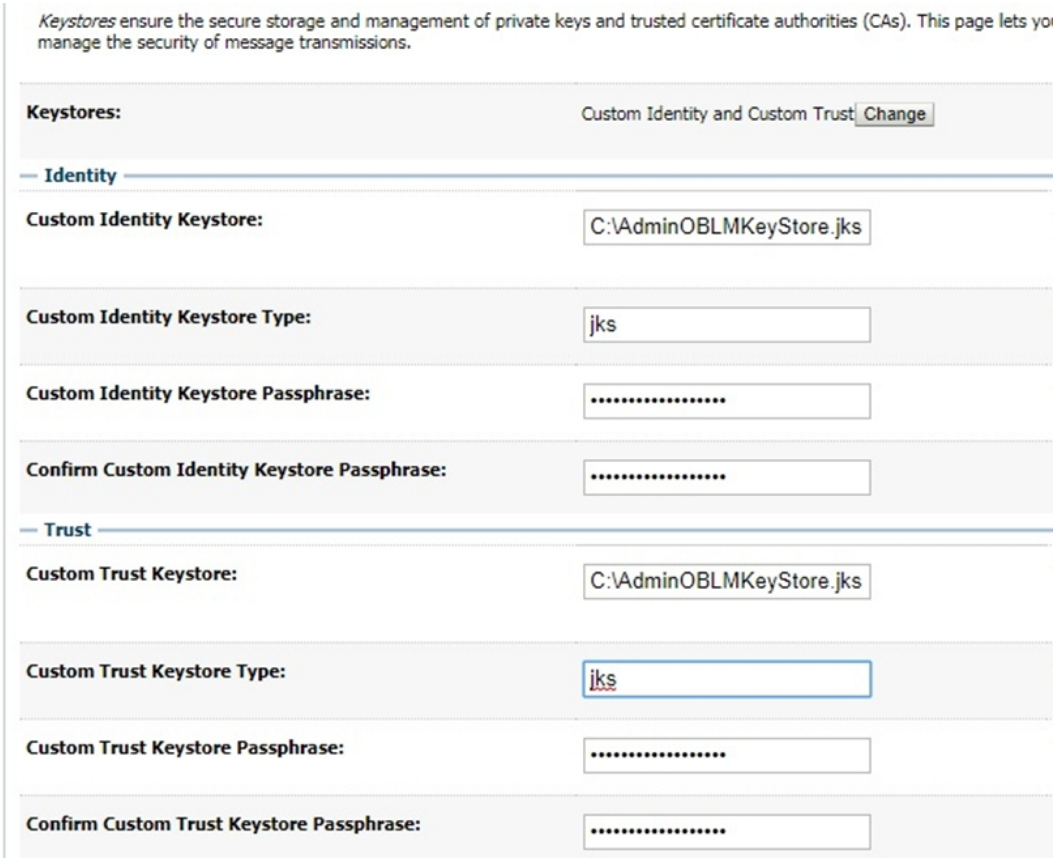
4. Click **Save**.

Figure 5-3 Settings for AdminServer



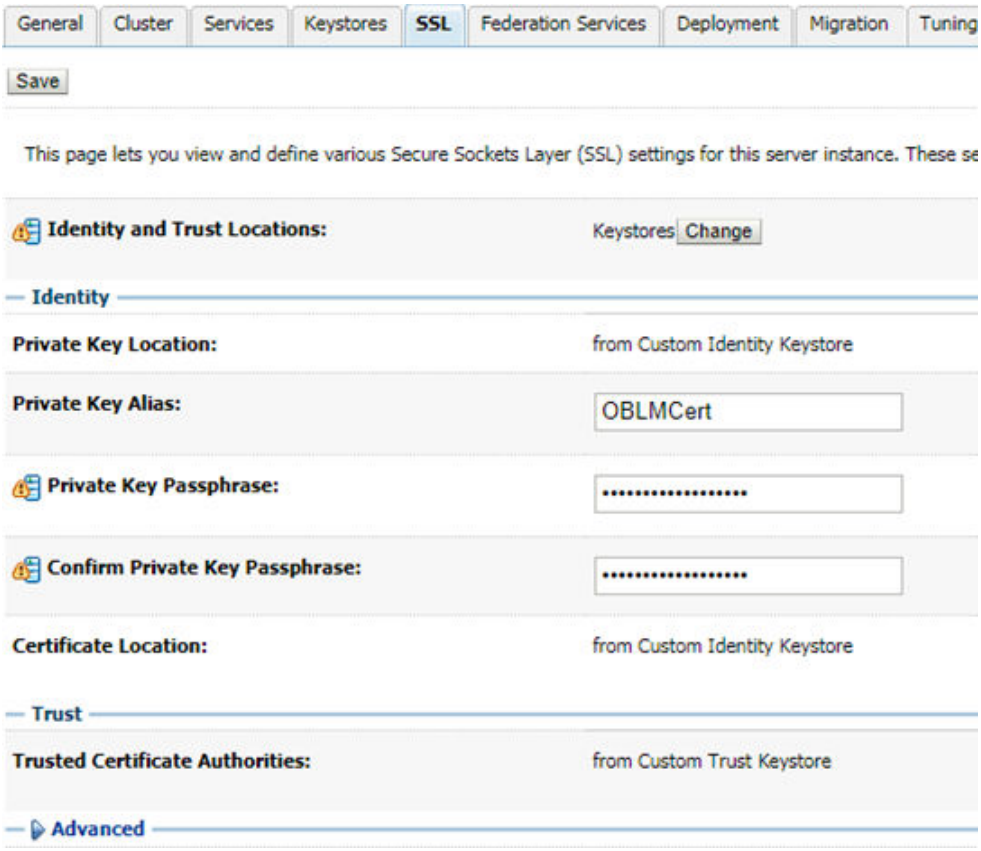
5. Click **Keystores** tab.

Figure 5-4 Keystores



6. Specify **Custom Identity Keystore** and **Custom Trust Keystore** same as the **Keystore Name** created in above steps with full path.
7. Specify **Custom Identity Keystore Type** and **Custom Trust Keystore Type** as jks.
8. Specify **Custom Identity Keystore Passphrase**, **Confirm Custom Identity Keystore Passphrase**, **Custom Trust Keystore Passphrase** and **Confirm Custom Trust Keystore Passphrase** same as the **Store Password** entered in above steps.
9. Click **Save**.
10. Click **SSL** tab.

Figure 5-5 SSL



11. Specify **Private Key Alias** as same as the alias name entered in above steps.
12. Specify **Private Key Passphrase** and **Confirm Private Key Passphrase** as same as the **Private Key Password** entered in above steps.
13. Change the **Hostname Verification** to **None**.
14. Click **Save**.

    Repeat the same steps for all the managed servers as well. The admin server and managed servers are SSL enabled.
15. Restart all the servers.

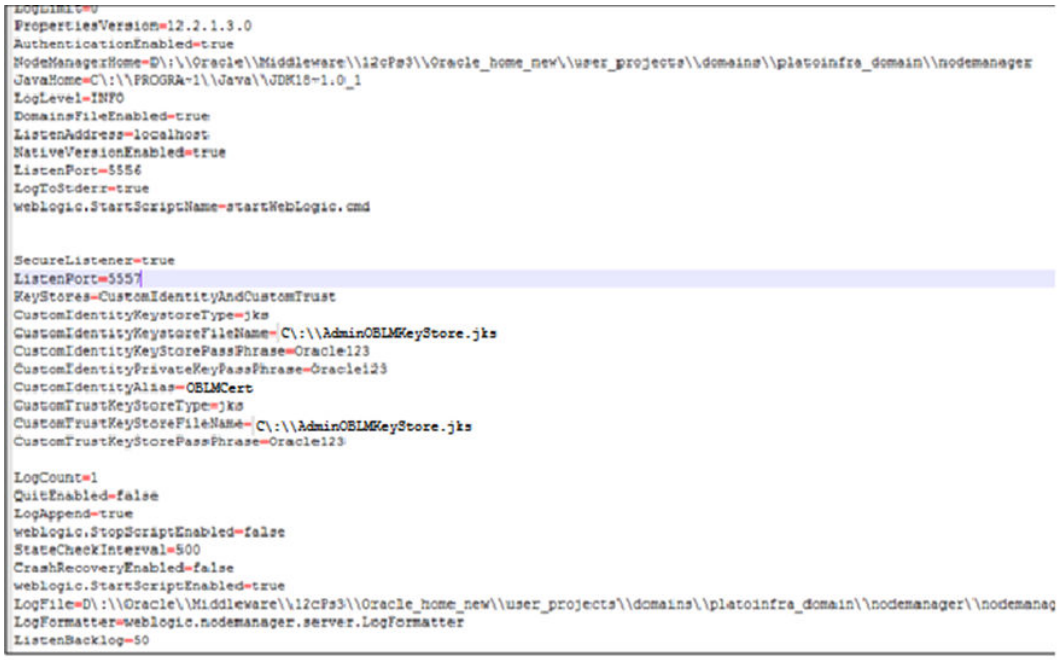
6

Configure SSL Mode in Node Manager for Clustered Environment

This topic provides the systematic instructions to configure the SSL model in node manager for clustered environment.

1. Edit the nodemanager.properties with SSL configurations and restart the node manager.

Figure 6-1 Node Manager Properties



2. Ensure the SSL configuration is performed in other artifacts, such as startNodeManager.cmd/.sh, startup.properties, config.xml(enable jsse).

7

Set SSL Attributes for Managed Servers

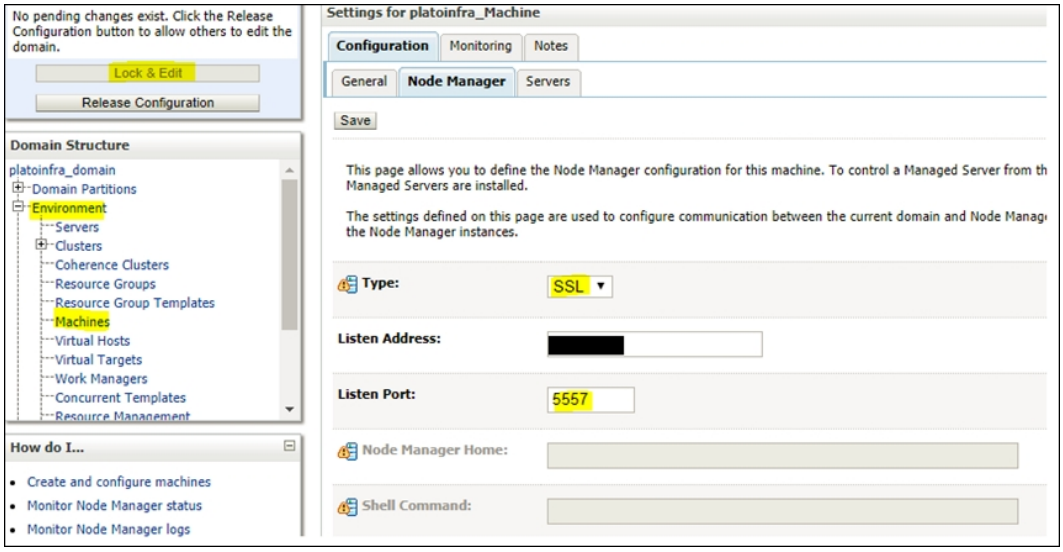
This topic provides the systematic instructions to set the SSL attributes for Managed Servers.

Set SSL Attributes for Private Key Alias and Password
Login to the **Oracle Weblogic Server Admin Console** to configure the private key alias and password.

1. Under **Change Center**, click **Lock and Edit**.
2. Expand **Servers** node.
3. Select the name of the server to configure keystores.

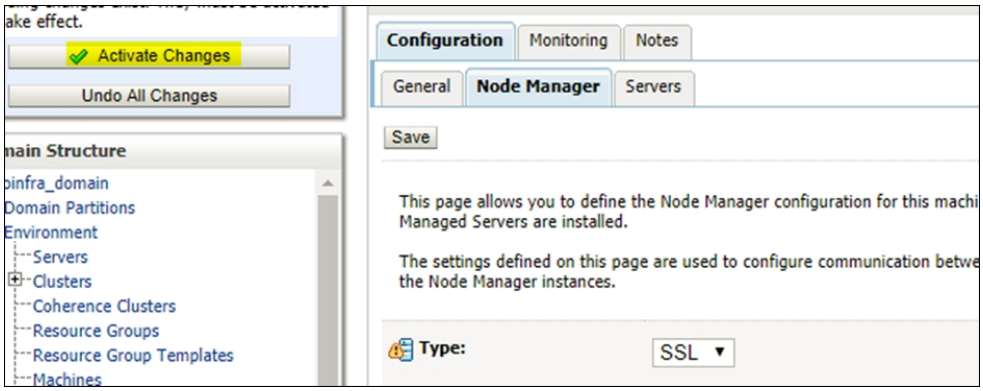
   Example: exampleserver
4. Navigate to **Configuration** and select **SSL** tab.

Figure 7-1 Configuration



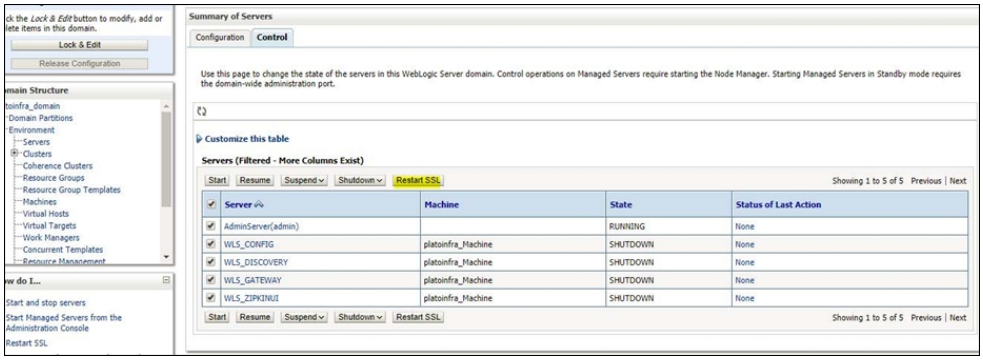
5. Select **Keystores** from **Identity and Trust Locations**.
6. Under **Identity** section, specify the following details:
   - **Private Key Alias**: Set this attribute to the alias name defined for the key pair when creating the key pair in the Identity keystore.
   - **Private Key Passphrase**: The password defined for the key pair (alias_password), at the time of its creation. Confirm the password.
7. Click **Save**.
8. Under **Change Center**, click **Activate Changes**.

Figure 7-2 Configuration - Activate Changes



9. Navigate to **Controls** tab, check the appropriate server.

Figure 7-3 Summary of Servers



10. Click **Restart SSL**, and confirm when it prompts.

8
Test Configuration

This topic provides the information to test the configuration.

Once the Oracle Weblogic has been configured for SSL, deploy the application in the usual manner. After deployment, the user can test the application in SSL mode.

To launch the application in SSL mode, the user need to enter the URL in the following format:

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

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

Index