Oracle® Banking Corporate Lending Process Management

SSL Configuration Setup Guide

Release 14.7.4.0.0
G11870-01
June 2024

Oracle Banking Corporate Lending Process Management SSL Configuration Setup Guide, Release 14.7.4.0.0

G11870-01

Contents

Preface

This topic contains following sub-topics:

Purpose

This guide is designed to help acquaint you with the Oracle Banking Corporate Lending Process Management (OBCLPM) application. This guide provide the details of configurations for Secure Sockets Layer (SSL) on Oracle Weblogic application server.

Audience

This document is intended for admin or ops-web team who are responsible for installing the Oracle Financial Services Software Limited banking products.

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 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.

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.

Related Resources

For further information on SSL configuration for Oracle Banking Corporate Lending Process Management application, refer to the following manuals.

- *Configuration and Deployment Guide*
- *Oracle Banking Corporate Lending Process Management Pre-Installation Guide*
- *Oracle Banking Corporate Lending Process Management Services Installation Guide*

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 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.

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.

Acronyms and Abbreviations

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

Table 1 Acronyms and Abbreviations

Abbreviation	Description
API	Application Programming Interface
CA	Certificate Authority
CSR	Certificate Signing Request
JKS	Java keystore
SSL	Secure Sockets Layer

Basic Actions

Table 2 List of Basic Actions

Action	Description
Approve	Click **Approve** to approve the initiated report. This button is displayed, once the user click **Authorize**.
Audit	Click **Audit** to view the maker details, checker details of the particular record, and record status. This button is displayed only for the records that are already created.
Authorize	Click **Authorize** to authorize the record created. A maker of the screen is not allowed to authorize the report. Only a checker can authorize a record. This button is displayed only for the already created records.
Close	Click **Close** to close a record. This action is available only when a record is created.
Confirm	Click **Confirm** to confirm the performed action.
Cancel	Click **Cancel** to cancel the performed action.
Compare	Click **Compare** to view the comparison through the field values of old record and the current record. This button is displayed in the widget, once the user click **Authorize**.
Collapse All	Click **Collapse All** to hide the details in the sections. This button is displayed, once the user click **Compare**.
Expand All	Click **Expand All** to expand and view all the details in the sections. This button is displayed, once the user click **Compare**.
New	Click **New** to add a new record. The system displays a new record to specify the required data. **Note:** The fields which are marked with Required are mandatory.
OK	Click **OK** to confirm the details in the screen.
Save	Click **Save** to save the details entered or selected in the screen.
View	Click **View** to view the report details in a particular modification stage. This button is displayed in the widget, once the user click **Authorize**.
View Difference only	Click **View Difference only** to view a comparison through the field element values of old record and the current record, which has undergone changes. This button is displayed, once the user click **Compare**.

Symbols and Icons

The following symbols and icons are used in the screens.

Table 3 Symbols and Icons - Common

Symbol/Icon	Function
	Minimize
	Maximize
	Close
	Perform Search
	Open a list
	Add a new record
	Navigate to the first record
	Navigate to the last record
	Navigate to the previous record

Table 3 (Cont.) Symbols and Icons - Common

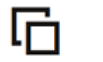
Symbol/Icon	Function
	Navigate to the next record
	Grid view
	List view
	Refresh
	Calender
	Filter
	Copy a record
	Click this icon to add a new row.
	Click this icon to delete an existing row.

Table 3 (Cont.) Symbols and Icons - Common

Symbol/Icon	Function
	Click to view the created record.
	Click to unlock, delete, authorize or view the created record.

Table 4 Symbols and Icons - Audit Details

Symbol/Icon	Function
	A user
	Date and time
	Unauthorized or Closed status
	Authorized or Open status

Table 5 Symbols and Icons - Widget

Symbol/Icon	Function
	Open status

Table 5 (Cont.) Symbols and Icons - Widget

Symbol/Icon	Function
	Unauthorized status
	Closed status
	Authorized status

1

Configuring SSL on Oracle Weblogic

Use this topic to configure SSL on Oracle Weblogic application server.

Setting up SSL on Oracle Weblogic

To setup SSL on Oracle Weblogic application server:

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. 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.

Certificates and Keypairs

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, 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 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 you 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

Choosing the Identity and Trust Stores

Use this topic to choose 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 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 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

Obtaining the Identity Store

Use this topic to obtain identity store.

Creating 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.

In order to create a self-signed certificate, the genkeypair option provided by the keytool utility of Sun Java 8 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 Oracle Weblogic Server.
3. **First and Last Name (CN)**: Enter the domain name of the machine used to access OBCLPM, 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 further identify the SSL Certificate you are creating, for example, by department or by 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, and so on.

> **Note:**
>
> 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

Example

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

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

Keystore Creation

```
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>
```

For example: `keytool -genkeypair -keystore AdminOBCLPMKeyStore.jks -alias OBCLPMCert -dname "CN=ofss00001.oracle.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.oracle.com is the Host Name of the weblogic server

Creating Identity Store with Trusted Certificates Issued by CA

Creation of 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. The **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. The **keyalg** is the key algorithm used to generate the public and private key pair. The RSA key algorithm is recommended.
3. The **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. The **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. The **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 Oracle Weblogic Server
3. **First and Last Name (CN)**: Enter the domain name of the machine used to access OBCLPM, 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 further identify the SSL Certificate you are creating, for example, by department or by 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, and so on.
   Example: Listed below is the result of a sample execution of the command:

```
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\AdminOBCLPMKeyStore.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>
```

Generating CSR

To purchase an SSL certificate, you must 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. The **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. The **certreq_file** is the file in which the CSR will be stored.
3. The **keystore** is the location of the keystore containing the public and private key pair. Example: Listed below is the result of a sample execution of the command:

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

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>
```

For example:`keytool -export -v -alias OBCLPMCert -file AdminOBCLPMCert.cer -keystore AdminOBCLPMKeyStore.jks -keypass Oracle123 -storepass Oracle123If successful, the following message will be displayed: Certificate stored in file <AdminOBCLPMCert.cer>`

Obtaining 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

Importing 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 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.

Importing 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
```

In the above command,

1. The **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. The **cert_file** is the location of the file containing the intermediate CA certificate in a PKCS#7 format (PEM or DER file).
3. The keystore is the location of the keystore containing the public and private key pair.

> **Note:**
>
> 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:

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

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

Certificate was added to keystore
```

Importing 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. The **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. The **cert_file** is the location of the file containing the PKCS#7 formatted reply from the CA, containing the signed certificate.
3. The **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

```
D:\Oracle\weblogic11g\jrockit_160_05_R27.6.2-20\bin>keytool - importcert -
alias cvrhp0729 -file D:\keystores\cvrhp0729.cer - trustcacerts -keystore

D:\keystoreworkarea\AdminOBCLPMKeyStore.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
```

> **Note:**
>
> 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.

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>
```

For example:`keytool -import -v -trustcacerts -alias rootcacert -file AdminOBCLPMCert.cer -keystore AdminOBCLPMKeyStore.jks -keypass Oracle123 -storepass Oracle123`

4

Configuring Identity and Trust Stores for Weblogic

Use this topic to configure identity and trust stores for Weblogic.

Enabling SSL on Oracle Weblogic Server

To configure SSL on Oracle Weblogic server, login in to the Admin Console:

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

Configuring Identity and Trust Stores

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

1. Under **Change Center**, click **Lock & Edit**.
2. Expand **Servers** node.
3. Select the name of the server for which you want 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 Section 2 of this document (Choosing the Identity and Trust Stores).
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 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:

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

5

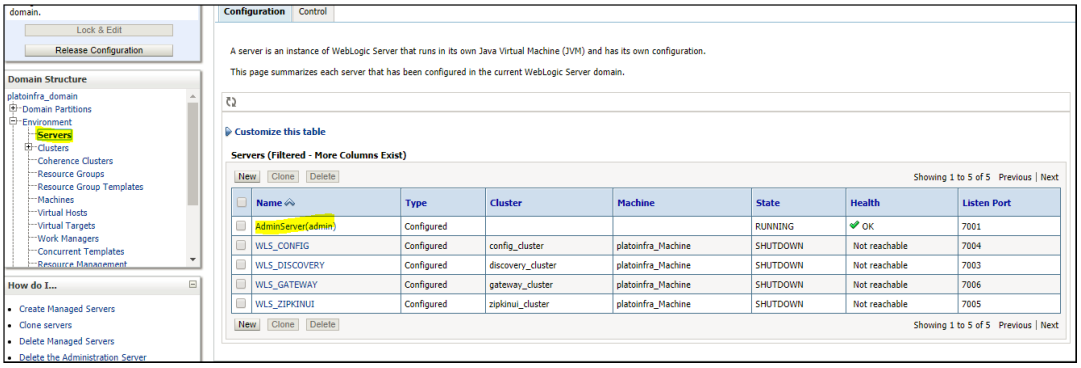
Configuring Weblogic Console (12.2.1.4)

Use this topic to configure Weblogic Console.

After domain creation, follow the below steps to enable SSL in weblogic Admin server:

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

Figure 5-1 Configuration tab



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

Figure 5-2 Listen Port Enabled

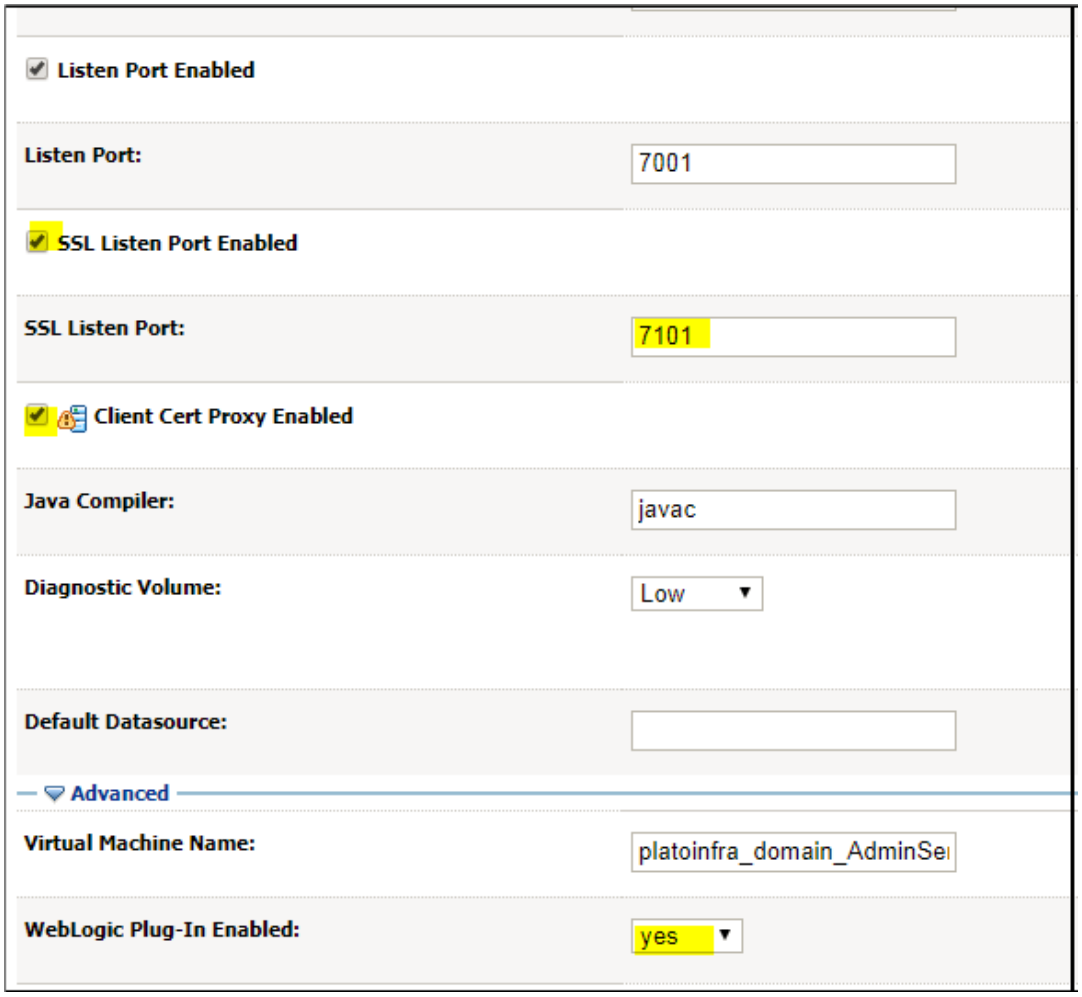
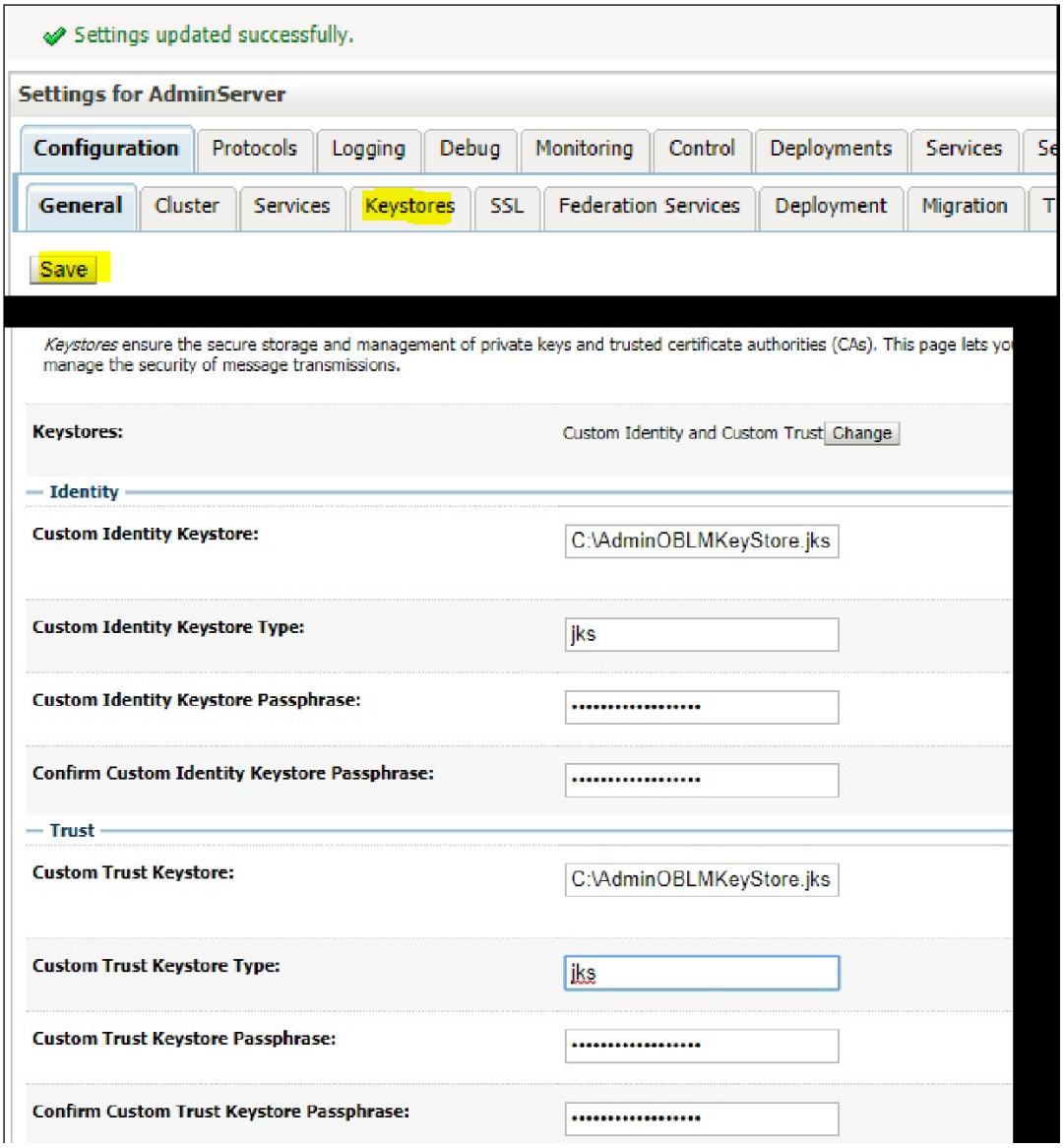


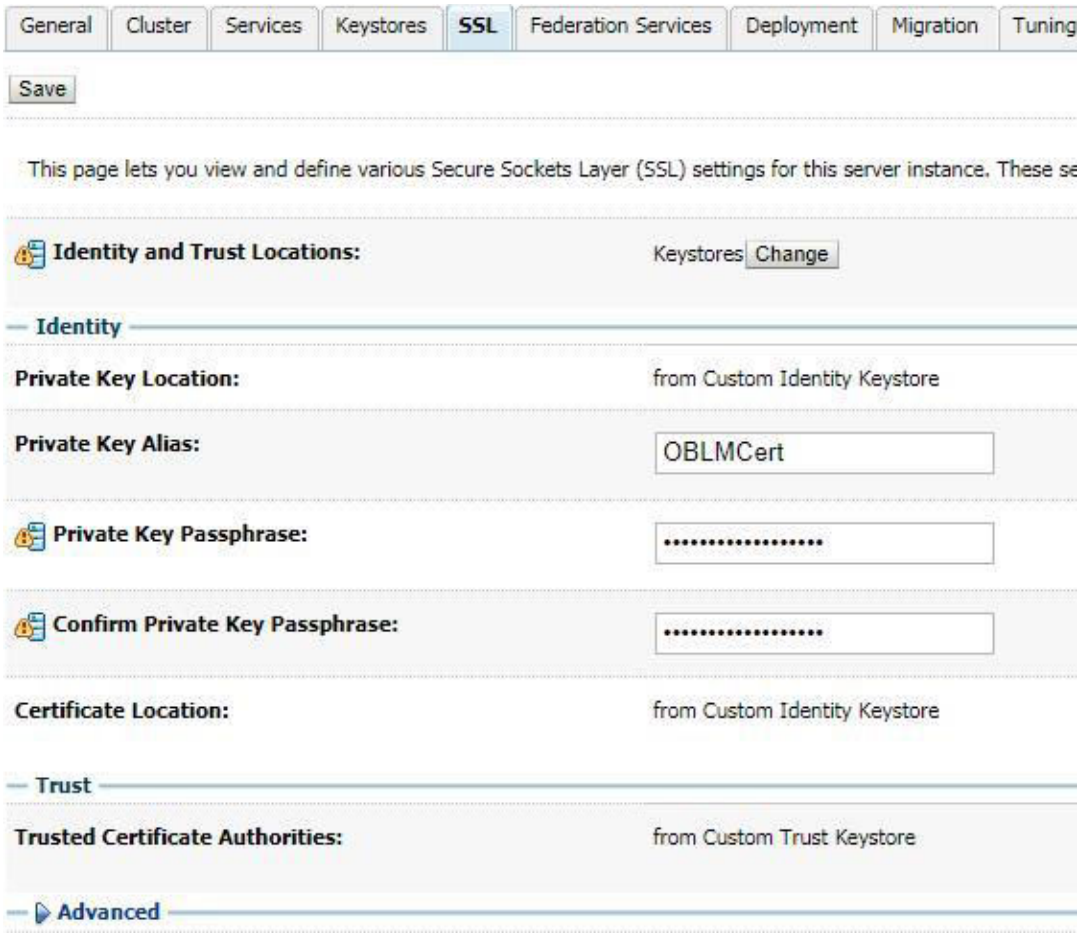
Figure 5-3 Keystores



5. Click **Keystores** tab.
6. Enter Custom Identity Keystore and Custom Trust Keystore same as the Keystore Name created in above steps with full path.
7. Enter Custom Identity Keystore Type and Custom Trust Keystore Type as jks.
8. Enter 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.
11. Enter Private Key Alias as same as the alias name entered in above steps.
12. Enter 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**.

Figure 5-4 SSL Tab



> **Note:**
>
> Repeat the same steps for all the managed servers as well. The admin server and managed servers are SSL enabled. Restart all the servers.

6

Configuring SSL Mode in Node Manager for Clustered Environment

Use this topic to configure SSL mode in node manager for cluster 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

Setting SSL Attributes for Managed Servers

Use this topic to set SSL Attributes for private key alias and password.

Setting SSL Attributes for Private Key Alias and Password

To configure the private key alias and password, log in to the Oracle Weblogic Server Admin Console:

1. Under **Change Center**, click **Lock & Edit**.
2. Expand **Servers** node.
3. Select the name of the server for which you want to configure keystores. Example: exampleserver
4. Go to **Configuration** and select **SSL** tab.
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**.
9. Go to **Controls** tab, check the appropriate server and click **Restart SSL**. Confirm when it prompts.

Figure 7-1 Node Manager tab

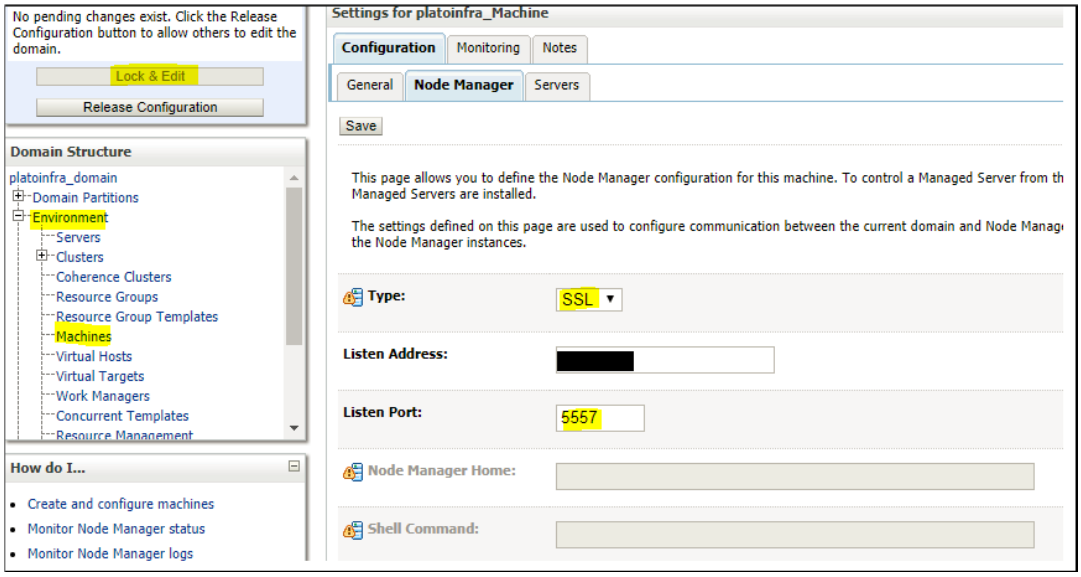
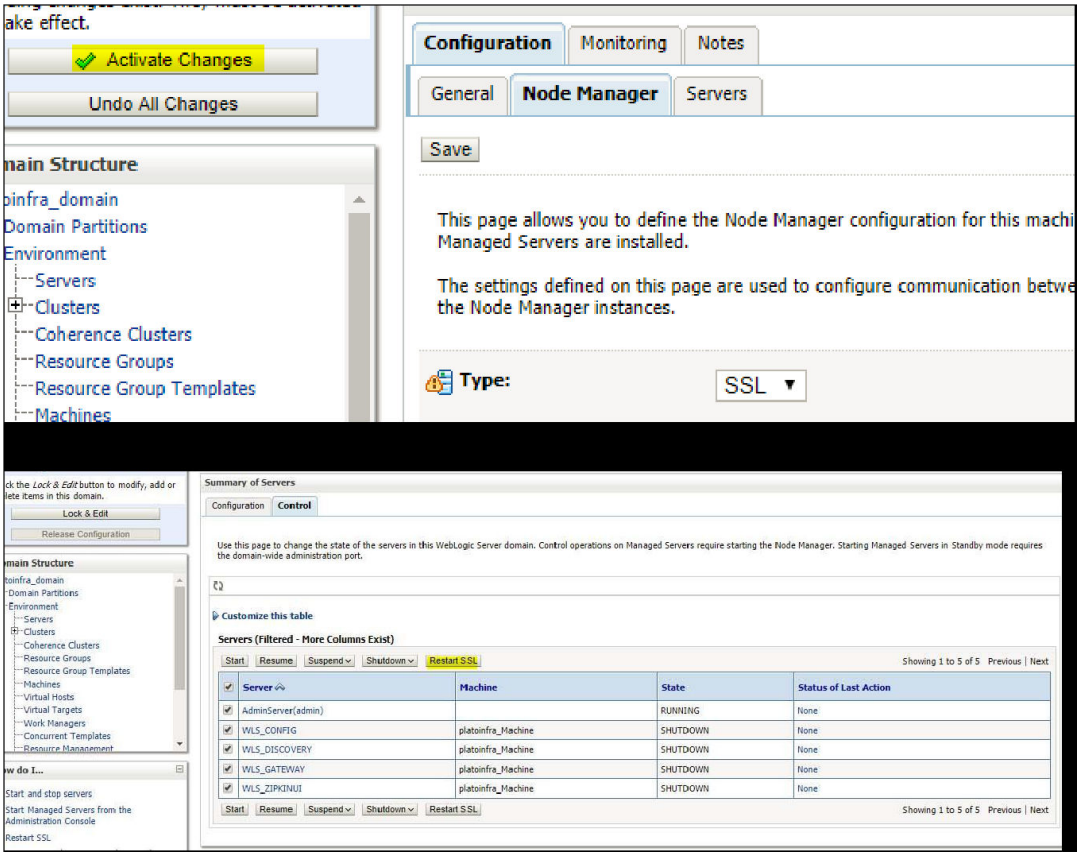


Figure 7-2 Activate Changes and Restart SSL



8
Testing Configuration

Use this topic to test the application in SSL mode.

Once the Oracle Weblogic has been configured for SSL, deploy the application in the usual manner. After deployment, you can test the application in SSL mode. To launch the application in SSL mode you 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 Corporate Lending Process Management web application be accessed through the HTTPS channel, instead of the HTTP channel.

Index