

Oracle® Banking Retail Lending Servicing SSL Configurations Setup Guide



Release 14.9.0.0.0
G58891-01
August 2026

ORACLE®

Copyright © 2021, 2026, Oracle and/or its affiliates.

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

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

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

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

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

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

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>.

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.

Contents

1	Configure SSL on Oracle Weblogic	
1.1	Certificates and Keypairs	1
2	About Choosing the Identity and Trust Stores	
3	Obtaining the Identity Store	
3.1	Create Identity Store with Self-Signed Certificates	1
3.2	Create Keystore	3
3.3	Creating Identity Store with Trusted Certificates Issued by CA	4
3.3.1	Duplicate - Data Source Creation	4
3.3.2	Generate CSR	4
3.4	Export Private Key as Certificate	5
3.4.1	About Obtaining and Importing Trusted Certificate	5
3.4.1.1	Import Intermediate CA Certificate	6
3.4.1.2	Import Identity Certificate	7
3.5	Import as Trusted Certificate	8
4	Organization	
5	Configure Weblogic Console	
6	Configure SSL Mode	
7	Setting SSL Attributes for Managed Servers	
8	Enable SSL in Oracle Banking Retail Lending Servicing	

Preface

- [Purpose](#)
- [Audience](#)
- [Critical Patches](#)
- [Diversity and Inclusion](#)
- [Conventions](#)
- [Acronyms and Abbreviations](#)
- [Screenshot Disclaimer](#)
- [Symbols and Icons](#)
- [Basic Actions](#)
- [Related Resources](#)

Purpose

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

Audience

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

Critical Patches

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

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.

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.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which user supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that user enter.

Acronyms and Abbreviations

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

Table Acronyms and Abbreviations

Abbreviation	Description
SSL	Secure Sockets Layer
JDK	Java Development Kit
JSK	Java KeyStore
CSR	Certificate Signing Request
JRE	Java Runtime Environment
CA	Certificate Authority
HTTPS	Hypertext Transfer Protocol Secure

Screenshot Disclaimer

Not Applicable

Symbols and Icons

Not Applicable

Basic Actions

Not Applicable

Related Resources

The related documents are as follows:

- *Configuration and Deployment Guide*
- Oracle Banking Retail Lending Servicing Installation Guide

1

Configure SSL on Oracle Weblogic

You can configure SSL in the Oracle WebLogic application server using identity and trust.

To setup SSL on the Oracle Weblogic application server:

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 keystores.
3. Configure the identity and trust the keystores for the Oracle Weblogic application server in the administration console.
4. Set SSL attributes for the private key alias and password in the Oracle Weblogic administration console.

Note

For information on certificates and keypairs, refer to [Configure SSL on Oracle Weblogic](#).

- [Certificates and Keypairs](#)
Certificates are used for validating the authenticity of the server, and the keys are used to secure the certificates.

1.1 Certificates and Keypairs

Certificates are used for validating the authenticity of the server, and the keys are used to secure the certificates.

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 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, 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. For information on the types of keystores, refer to the table below:

Table 1-1 Keystores

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

2

About Choosing the Identity and Trust Stores

Oracle Financial Services Software Limited 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 Software 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. For more information on choosing trust stores, refer to the table below:

Table 2-1 Trust Stores

Trust Store	Description
Command Line Trust	If Command-Line Trust is chosen, it 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	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 it is located in the <code>JAVA_HOME/jre/lib/security</code> 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.
Custom Trust	One can also create custom trust stores containing the list of certificates of trusted CAs. For further details on identity and trust stores, refer to the Oracle WebLogic Server documentation on Securing Oracle WebLogic Server.

3

Obtaining the Identity Store

The identity store needs to be obtained as a part of the SSL configuration setup.

This topic contains the following sub-topics:

- [Create Identity Store with Self-Signed Certificates](#)
Self-signed certificates are acceptable for use in a testing or development environment. Oracle Financial Services Software Limited does not recommend the use of self-signed certificates in a production environment.
- [Create Keystore](#)
User can create the keystore using the command to obtain the identity store.
- [Creating Identity Store with Trusted Certificates Issued by CA](#)
User needs to create the key pairs and generate a Certificate Signing Request (CSR) to obtain the identity store.
- [Export Private Key as Certificate](#)
User needs to export the private key as a certificate to obtain the identity store.
- [Import as Trusted Certificate](#)
User can import the obtained certificates as trusted certificates using the command to obtain the identity store.

3.1 Create Identity Store with Self-Signed Certificates

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

User can create the self-signed certificate using the command. For creating a self-signed certificate, the `genkeypair` option provided by the `keytool` utility of Sun Java 6 needs to be utilized.

To create a 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
```

① Note

The items highlighted in bold are placeholders and should be replaced with suitable values when running the command.

In the above command,

Table 3-1 Description of Placeholders and Attributes

Placeholder/Attribute	Description
alias	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.
keystore	keystore is used to specify the location of the JKS file. If no JKS file is present in the path provided, a JKS file will be created. The command will prompt for the following attributes of the certificate and keystore:
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.
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.
First and Last Name (CN)	Specify the domain name of the machine used to access Oracle Banking Branch. For instance, www.example.com.
Name of your Organizational Unit	Specify the name of the department or unit making the request. For example, BPD. Use this field to further identify the SSL Certificate user are creating. For example, by department or by the physical server.
Name of your Organization	Specify the name of the organization making the certificate request, for example, Oracle Financial Services Software Limited. 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	Specify the city in which the organization is physically located. For example, Mumbai.
Name of your State or Province	Specify the state or province in which your organization is physically located. For example, Maharashtra.
Two-Letter Country Code for this Unit	Specify 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, and 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.

For example:

The result of a sample execution of the command is listed below:

```
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\AdminOBREMOKeyStore.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 Software Limited
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.oracle.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 Create Keystore

User can create the keystore using the command to obtain the identity store.

To create the keystore, use the comment given below:

```
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 AdminOBREMOKeyStore.jks -alias OBREMOCert -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
```

① Note

CN=ofss00001.in.example.com is the Host Name of the WebLogic server.

3.3 Creating Identity Store with Trusted Certificates Issued by CA

User needs to create the key pairs and generate a Certificate Signing Request (CSR) to obtain the identity store.

This topic contains the following sub-topics:

- [Duplicate - Data Source Creation](#)
- [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.

3.3.1 Duplicate - Data Source Creation

This image displays the process of Data Source Creation.

3.3.2 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,

Table 3-2 Description of Placeholders

Placeholder	Description
alias	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.
certreq_file	certreq_file is the file in which the CSR will be stored.
keystore	keystore is the location of the keystore containing the public and private key pair.

For example,

The result of a sample execution of the 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\AdminOBREMOKeyStore.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]*

3.4 Export Private Key as Certificate

User needs to export the private key as a certificate to obtain the identity store.

To export the private key, use the command given below:

```
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 OBREMO Cert -file AdminOBREMO Cert.cer -keystore  
AdminOBREMOKeyStore.jks -keypass Oracle123 -storepass Oracle123
```

If successful, the following message will be displayed:

Certificate stored in file < AdminOBREMO Cert.cer>

- [About Obtaining and Importing Trusted Certificate](#)
The Trusted Certificate needs to be obtained from the CA and imported into the identity store.

3.4.1 About Obtaining and Importing Trusted Certificate

The Trusted Certificate needs to be obtained from the CA and imported into the identity store.

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 to submit the CSR and obtain 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. 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.

For information on importing the intermediate CA certificate and identity certificate, refer to the below topics:

- [Import Intermediate CA Certificate](#)
User needs to import the intermediate CA certificate into the identity keystore. 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.
- [Import Identity Certificate](#)
User can import the identity certificate into the keystore using the command.

3.4.1.1 Import Intermediate CA Certificate

User needs to import the intermediate CA certificate into the identity keystore. 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.

Execute the following command to import the intermediate CA certificate into the keystore:

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

In the above command,

Table 3-3 Description of Placeholders

Placeholder	Description
<i>cert_file</i>	<i>alias</i> 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.
<i>cert_file</i>	<i>cert_file</i> is the location of the file containing the intermediate CA certificate in a PKCS#7 format (PEM or DER file).
<i>keystore</i>	<i>keystore</i> 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.

A sample execution of the command is listed below:

```
D:\Oracle\weblogic11g\jrockit_160_05_R27.6.2-20\bin>keytool - importcert -alias  
verisigntrialintermediateca -file
```

```
D:\keystores\VerisignIntermediateCA.cer -trustcacerts -keystore
```

```
D:\keystoreworkarea\AdminOBREM0KeyStore.jks
```

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

Certificate was added to keystore

3.4.1.2 Import Identity Certificate

User can import the identity certificate into the keystore using the command.

Execute the following command to import the identity certificate into the keystore:

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

In the above command,

Table 3-4 Description of Placeholders

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

A sample execution of the 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\Admin0BREM0KeyStore.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, specifically 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.

3.5 Import as Trusted Certificate

User can import the obtained certificates as trusted certificates using the command to obtain the identity store.

To import as trusted certificate, execute the command given below:

```
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 AdminOBREMOCert.cer -  
keystore AdminOBREMOKeyStore.jks -keypass Oracle123 -storepass Oracle123
```

4

Organization

This guide allows you to install the below mentioned Oracle Banking Retail Lending Servicing services, UI, process flow in the same order:

Oracle Banking Retail Lending Servicing Services

1. obrl-ln-maintenance-services
2. obrl-ln-lending-product-services
3. obrl-ln-product-segment-services
4. obrl-ln-account-services
5. obrl-ln-amendment-services
6. obrl-ln-payment-services
7. obrl-ln-restructure-maintenance-services
8. obrl-ln-restructure-services
9. obrl-ln-migration-services
10. obrl-ln-schedule-services
11. slpr-core-services
12. slpr-batchcategory-services
13. slp-setup-services
14. slp-integration-services
15. slpr-file-gen-services

User Interface

Follow the below steps to migrate from the existing app-shell build to the foundation app shell. With the foundation app-shell, UI war is split into individual component server war files. All the component server war files should be deployed in the same managed server

For common core war files, deploy the war files mentioned below:

1. app-shell
2. cmc-component-server
3. moc-component-server
4. sms-component-server
5. obpy-component-server

For domain-specific war files, deploy the individual component server war file mentioned below:

- obrl-component-server

5

Configure Weblogic Console

After the user creates the domain, the SSL needs to be enabled in the WebLogic admin server.

To enable SSL in WebLogic admin server:

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

Figure 5-1 Enabling SSL Options

Summary of Servers

Configuration **Control**

Use this page to change the state of the servers in this WebLogic Server domain. Control operations on Managed Servers require starting the Node Manager. Starting Managed : wide administration port.

[Customize this table](#)

Servers (Filtered - More Columns Exist)

Start Resume Suspend Shutdown Restart SSL

Server	Machine	State	Status of Last Action
☒ AdminServer(admin)		RUNNING	None
☒ ManagedServer_1	new_Machine_1	RUNNING	TASK COMPLETED

Start Resume Suspend Shutdown Restart SSL

2. In the **General** tab, follow the steps below:
 - a. Select **SSL Listen Port Enabled**, **Client Cert Proxy Enabled**, and then **Weblogic Plug-In Enabled**.

Figure 5-2 General Tab

☒ Listen Port Enabled

Listen Port: 9900

☒ SSL Listen Port Enabled

SSL Listen Port: 7002

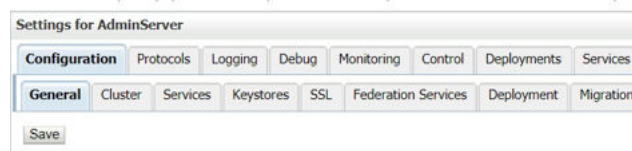
☐ Client Cert Proxy Enabled

Java Compiler: javac

Diagnostic Volume: Low

Default Datasource:

- b. Click **Save**.

Figure 5-3 Settings for Admin Server

3. In the **Keystores** tab, follow the below steps:
 - a. In the **Keystores** tab, specify the details. For more information on fields, refer to the field description table.

Table 5-1 Keystores - Field Description

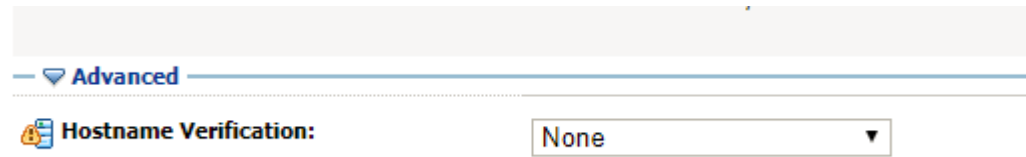
Field	Description
Custom Identity Keystore and Custom Trust Keystore	Specify the value as same as the Keystore Name created in the above steps with full path.
Custom Identity Keystore Type and Custom Trust Keystore Type	Specify the value as jks.
Custom Identity Keystore Passphrase, Confirm Custom Identity Keystore Passphrase, Custom Trust Keystore Passphrase, and Confirm Custom Trust Keystore Passphrase	Specify the value as same as the Store Password entered in the above steps.

- b. Click **Save**.
4. In the **SSL** tab, follow the below steps:
 - a. In the **SSL** tab, specify the details. For more information on fields, refer to the field description table.

Table 5-2 SSL - Field Description

Field	Description
Private Key Alias	Specify the value as same as the alias name entered in the above steps.
Private Key Passphrase and Confirm Private Key Passphrase	Specify the value as same as the Private Key Password entered in the above steps.
Hostname Verification	Change the value to None .

- b. Click **Save**.

Figure 5-4 Hostname Verification

The screen with the Hostname Verification option is displayed.

- c. Repeat steps (a) thru (d) for all the managed servers.

The admin server and managed servers are SSL enabled.

- d. Restart all the servers.

6

Configure SSL Mode

User needs to configure the SSL mode in node manager for the clustered environment.

To configure the SSL mode:

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

Figure 6-1 SSL Mode Configuration

```
LogLimit=0
PropertiesVersion=12.2.1.3.0
AuthenticationEnabled=true
NodeManagerHome=D:\Oracle\Middleware\12cPs3\Oracle_home_new\user_projects\domains\platoinfra_domain\nodemanager
JavaHome=C:\Program Files\Java\jdk1.8.0_1
LogLevel=INFO
DomainsFileEnabled=true
ListenAddress=localhost
NativeVersionEnabled=true
ListenPort=5556
LogToStderr=true
weblogic.StartScriptName=startWebLogic.cmd

SecureListener=true
ListenPort=5557
KeyStores=CustomIdentityAndCustomTrust
CustomIdentityKeystoreType=jks
CustomIdentityKeystoreFileName=C:\AdminOBVAMKeyStore.jks
CustomIdentityKeystorePassPhrase=Oracle123
CustomIdentityPrivateKeyPassPhrase=Oracle123
CustomIdentityAlias=OBVAMCert
CustomTrustKeystoreType=jks
CustomTrustKeystoreFileName=C:\AdminOBVAMKeyStore.jks
CustomTrustKeystorePassPhrase=Oracle123

LogCount=1
QuitEnabled=false
LogAppend=true
weblogic.StopScriptEnabled=false
StateCheckInterval=500
CrashRecoveryEnabled=false
weblogic.StartScriptEnabled=true
LogFile=D:\Oracle\Middleware\12cPs3\Oracle_home_new\user_projects\domains\platoinfra_domain\nodemanager\nodemanager.log
LogFormatter=weblogic.nodemanager.server.LogFormatter
ListenBacklog=50
```

The screen with SSL configurations is displayed.

2. Make sure that the SSL configuration is done in other artifacts, such as `startNodeManager.cmd/.sh`, `startup.properties`, `config.xml`(enable jsse).

7

Setting SSL Attributes for Managed Servers

User needs to configure the SSL attributes of the private key alias and password for the managed servers.

Prerequisite: Log in to the Homepage of the Oracle Weblogic Server Admin Console.

To configure the private key alias and password:

1. On the Homepage, under **Change Center**, click **Lock and Edit**.
2. Expand the **Servers** node.
3. Select the name of the server, to configure keystores.
For example, exampleserver.
4. Navigate to **Configuration** and select the **SSL** tab.
5. From **Identity and Trust Locations**, select **Keystores**.
6. In the **Identity** section, specify the details. For more information on fields, refer to the field description table.

Table 7-1 Identity Section - Field Description

Field	Description
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 is 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. Navigate to the **Controls** tab, check the appropriate server, and click **Restart SSL**.
10. Click **Confirm**, when it prompts.

The screen with settings is displayed as shown in the figures.

Figure 7-1 Settings for New Machine

Settings for new_Machine_1

Configuration Monitoring Notes

General **Node Manager** Servers

Save

This page allows you to define the Node Manager configuration for this machine. To control a are installed.

The settings defined on this page are used to configure communication between the current Manager instances.

Type: SSL ▼

Listen Address: localhost

Listen Port:

Figure 7-2 Settings for Servicing

Settings for SERVICING

Configuration Monitoring Notes

General **Node Manager** Servers

Save

This page allows you to define the Node Manager configuration for this

The settings defined on this page are used to configure communication

Type: SSL ▼

Figure 7-3 Summary of Servers

Summary of Servers

ConfigurationControl

Use this page to change the state of the servers in this WebLogic Server domain. Control operations on Managed Servers require starting the Node Manager. Starting Managed : wide administration port.

[Customize this table](#)

Servers (Filtered - More Columns Exist)

StartResumeSuspendShutdownRestart SSL

☒	Server	Machine	State	Status of Last Action
☒	AdminServer(admin)		RUNNING	None
☒	ManagedServer_1	new_Machine_1	RUNNING	TASK COMPLETED

StartResumeSuspendShutdownRestart SSL

8

Enable SSL in Oracle Banking Retail Lending Servicing

As part of enabling end to end SSL communication between services, following keys are added in almost all the services:

- `eureka.instance.prefer-ip-address`
- `eureka.instance.nonSecurePortEnabled`
- `eureka.instance.securePortEnabled`

To proceed with the current setup, execute the following scripts before deploying services other than `plato-config-service`:

- `update properties set value = 'http://hostname:port/plato-discovery-service/eureka' where value = 'https://hostname:port/plato-discovery-service/eureka'`
- `update properties set value = 'true' where key = 'eureka.instance.prefer-ip-address'`
`update properties set value = 'true' where key = 'eureka.instance.nonSecurePortEnabled'`
- `update properties set value = 'false' where key = 'eureka.instance.securePortEnabled'`
- `update properties set value = 'http' where value = 'https';`

9

Testing Configuration

Once the Oracle WebLogic has been configured for SSL, user can deploy the application in the usual manner and test it in SSL mode.

After deployment, user can test the application in SSL mode.

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 Retail Operations web application be accessed via the HTTPS channel, instead of the HTTP channel.