Oracle® Banking Enterprise Default Management

Security Guide

Release 2.11.0.0.0

F36758-01

December 2020

Oracle Banking Enterprise Default Management Security Guide, Release 2.11.0.0.0

F36758-01

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 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" or "commercial computer software documentation" 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 failsafe, 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.

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.

Contents

List of Figures

List of Tables

Preface

This document provides a comprehensive overview of security for Oracle Banking Enterprise Default Management. It includes conceptual information about security principles, descriptions of the product's security features, and procedural information that explains how to use those features to secure Oracle Banking Enterprise Default Management.

This preface contains the following topics:

- Audience
- Documentation Accessibility
- Organization of the Guide
- Related Documents
- Conventions

Audience

This guide is intended for Bank IT Staff responsible for application installation and security configuration.

Documentation Accessibility

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

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/us/corporate/accessibility/support/index.html#info or visit http://www.oracle.com/us/corporate/accessibility/support/index.html#trs if you are hearing impaired.

Organization of the Guide

This document contains:

Chapter 1 About This Guide

This chapter provides details about applicability of this guide.

Chapter 2 Overview

This chapter presents an overview of the application and explains the general principles of application security.

Chapter 3 Secure Installation and Configuration

This chapter provides an overview of secure installation process through recommended deployment topologies and describes the installation and configuration procedure for the infrastructure and product components of the application.

Chapter 4 Security Features

This chapter outlines the specific security mechanisms offered by the application.

Chapter 5 Data Privacy and Security

This chapter explains the data privacy and security features offered by application.

Appendix

This appendix lists the Secure Deployment Checklist which includes guidelines that help secure the application.

Related Documents

For more information, see the following documentation:

- Hardening Tips for Default Installation of Oracle Enterprise Linux 6 at https://docs.oracle.com/cd/E40518_01/server.761/es_security/src/csec_os_harden_linux.html
- Oracle® Fusion Middleware Installation Guide for Oracle WebLogic Server at https://docs.oracle.com/middleware/11119/wls/WLSIG/toc.htm
- Oracle® Collaboration Suite Security Guide at http://docs.oracle.com/cd/B25553_01/collab.1012/b25494/toc.htm
- Oracle® Fusion Middleware Application Security Guide - Configuring and Managing Auditing at http://docs.oracle.com/cd/E23943_01/core.1111/e10043/audpolicy.htm
- For installation and configuration information, see the Oracle Banking Enterprise Default Management Localization Installation Guide - Silent Installation guide.
- For the complete list of licensed products and the third-party licenses included with the license, see the Oracle Banking Enterprise Default Management Licensing Guide.
- For information related to setting up a bank or a branch, and other operational and administrative functions, see the Oracle Banking Enterprise Default Management Administrator Guide.
- For information related to customization and extension, see the Oracle Banking Enterprise Default Management Extensibility Guides for HOST and UI.
- For information on the functionality and features, see the respective Oracle Banking Enterprise Default Management Functional Overview document.
- For recommendations of secure usage of extensible components, see the Oracle Banking Enterprise Default Management Secure Development 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.

1 About This Guide

This guide is applicable for the following products:

- Oracle Banking Platform
- Oracle Banking Enterprise Default Management

References to Oracle Banking Platform or OBP in this guide apply to all the above mentioned products.

2 Overview

This chapter presents an overview of Oracle Banking Platform and explains the general principles of application security.

2.1 Product Overview

Oracle Banking Platform lays the foundation of a single unified Core Banking platform having the following features:

- Amalgamation of Origination, Business Banking, Direct Banking
- Common SMS
- Common Architectural Principles
- Enterprise Ready Business Services

2.2 General Security Principles

The following principles are fundamental for using any application securely.

2.2.1 Restrict Network Access to Critical Services

Keep both the Oracle Banking Platform middle-tier and the database behind a firewall. In addition, place a firewall between the middle-tier and the database. The firewalls provide assurance that access to these systems is restricted to a known network route, which can be monitored and restricted, if necessary. As an alternative, a firewall router substitutes for multiple, independent firewalls.

If firewalls cannot be used, be certain to configure the TNS Listener Valid Node Checking feature which restricts access based upon IP address. Restricting database access by IP address often causes application client or server programs to fail for DHCP clients. To resolve this, consider using static IP addresses, a software or a hardware VPN or Windows Terminal Services or its equivalent.

2.2.2 Follow the Principle of Least Privilege

The principle of least privilege states that users should be given the least amount of privilege to perform their jobs. User privileges should be reviewed periodically to determine relevance to current job responsibilities.

2.2.3 Monitor System Activity

System security stands on three legs:

1. Good security protocols
2. Proper system configuration
3. System monitoring

System needs to be constantly monitored from Oracle Enterprise Manager.

2.2.4 Keep Up To Date on Latest Security Information

Oracle continually improves its software and documentation.

3 Secure Installation and Configuration

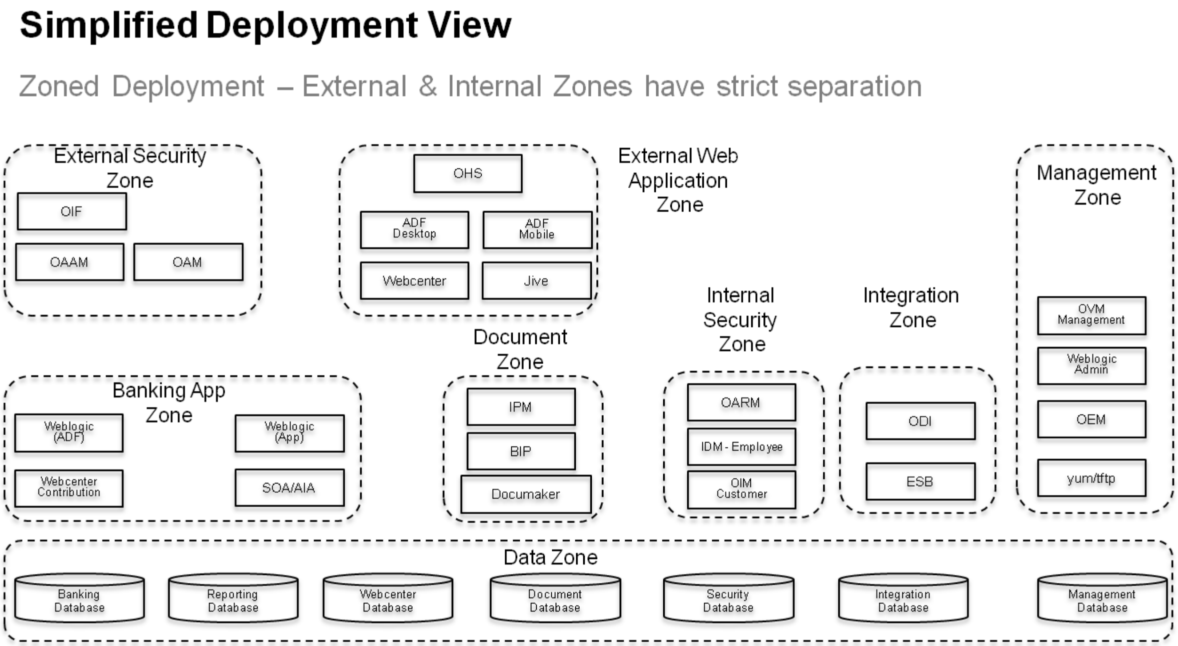
This chapter provides an overview of the recommended deployment topologies and describes the installation and configuration procedure for the infrastructure and product components of Oracle Banking Platform.

3.1 Recommended Deployment Topologies

This section describes the recommended deployment topologies for Oracle Banking Platform.

The simplified deployment view is as shown below:

Figure 3–1 Simplified Deployment View



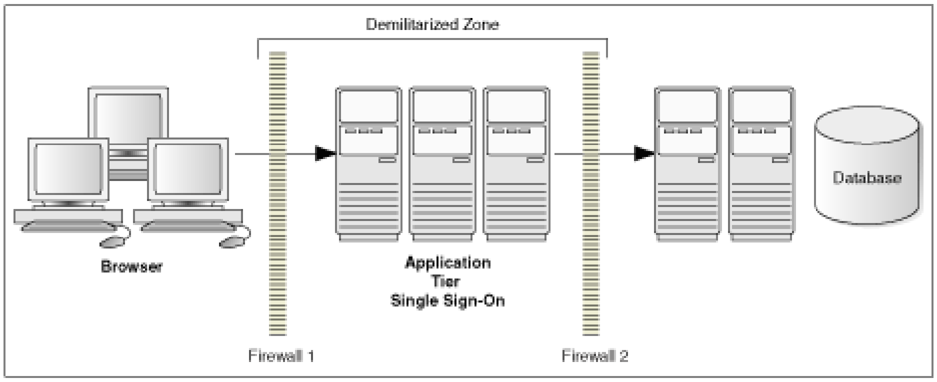
The deployment view for Oracle Banking Enterprise Default Management as shown in Figure 3–1 has the following features:

- Each zone is typically a separate network segment or subnet.
- Firewalls exist between each of these zones.
- The Document Zone and Integration Zones are shown for illustration purposes. Banks choose to typically deploy integration and document zones in the same Banking App Zone.
- Management Zone, Internal Security Zone and Banking Zone are typically an internal zone.
- Data is a separate zone.

- External Tiers have limited access to Data Zone.
  - This is for any personalization information that needs to be stored.
  - Banks may choose to deploy an external data zone which houses the personalization database.
- Access to core banking data (direct database access) is not allowed directly from the External Web Application Zone.
  - This would violate the defence in depth principle.
  - Access to core banking data is through services on HTTP protocol.

The general architectural recommendation is to use the well-known and generally accepted Internet-Firewall-DMZ-Firewall-Intranet architecture shown in Figure 3–2.

Figure 3–2 Traditional DMZ View



Note

The term Demilitarized Zone (DMZ) refers to a server that is isolated by firewalls from both the Internet and the intranet, thus forming a buffer between the two.

Firewalls separating DMZ zones provide two essential functions:

- Blocking any traffic types that are known to be illegal
- Providing intrusion containment, should successful intrusions take over processes or processors

3.2 Installing Linux

For installation of Oracle Banking Platform on Oracle Enterprise Linux 6, modify the default configuration following relevant instructions from the guide Hardening Tips for Default Installation of Oracle Enterprise Linux 6 at the following location:

https://docs.oracle.com/cd/E40518_01/server.761/es_security/src/csec_os_harden_linux.html

- Do not disable X Windows. It is needed for local administration and useful for troubleshooting.
- Do not disable SSH.

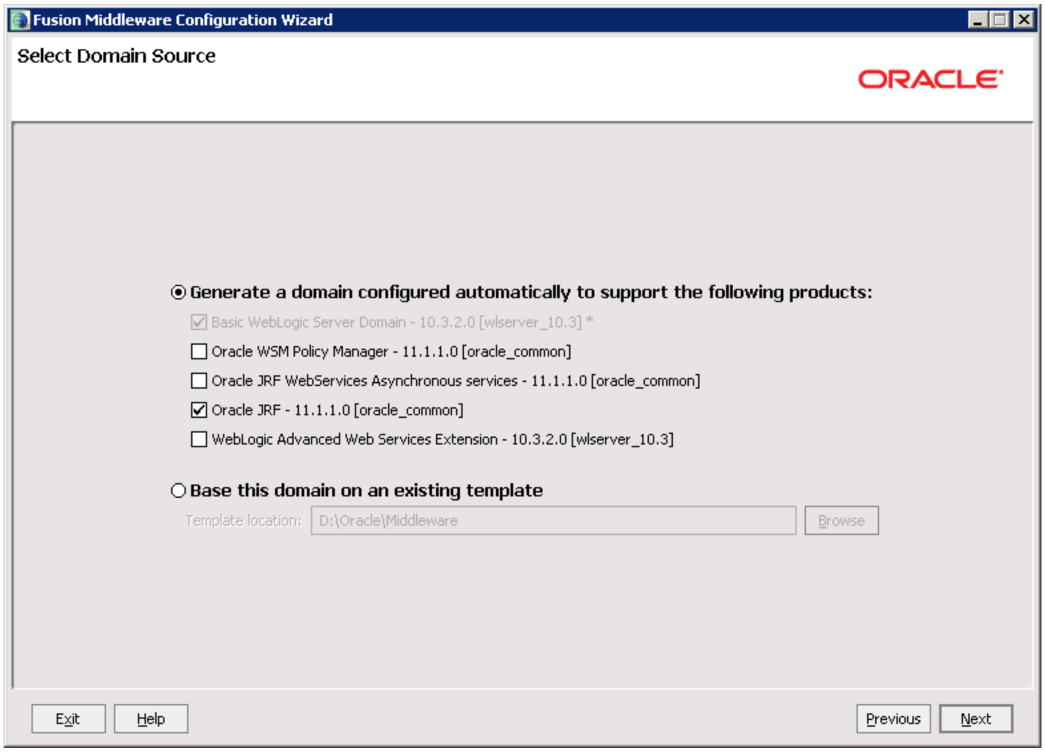
3.3 Installing WebLogic

Installation of WebLogic Server is done using the documentation as mentioned in the installation guide Oracle® Fusion Middleware Installation Guide for Oracle WebLogic Server at https://docs.oracle.com/middleware/11119/wls/WLSIG/toc.htm.

Following options need to be selected during the installation process:

1. Select the option **Generate a domain configured automatically to support the following products:**
2. From the above option, select the **Oracle JRF - 12.2.1.4 [oracle_common]** check box.

Figure 3–3 Select Domain Source



Select Domain Source

● Generate a domain configured automatically to support the following products:

☑ Basic WebLogic Server Domain - 10.3.2.0 [wlserver_10.3] *

□ Oracle WSM Policy Manager - 11.1.1.0 [oracle_common]

□ Oracle JRF WebServices Asynchronous services - 11.1.1.0 [oracle_common]

☑ Oracle JRF - 11.1.1.0 [oracle_common]

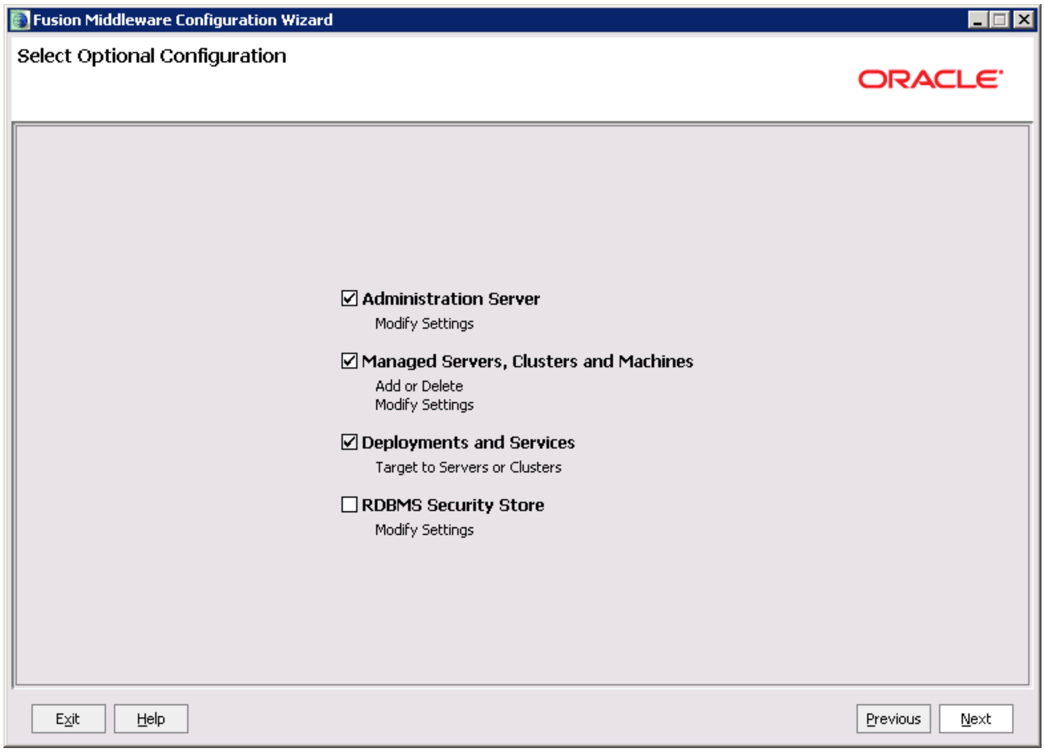
□ WebLogic Advanced Web Services Extension - 10.3.2.0 [wlserver_10.3]

○ Base this domain on an existing template

Template location: D:\Oracle\Middleware

3. Click **Next**.
4. Select the check box against the following options:
   - Administration Server
   - Managed Servers, Clusters and Machines
   - Deployments and Services

Figure 3–4 Select Optional Configuration


Fusion Middleware Configuration Wizard
Select Optional Configuration
ORACLE
Exit
Help
Previous
Next

☑ Administration Server
Modify Settings

☑ Managed Servers, Clusters and Machines
Add or Delete
Modify Settings

☑ Deployments and Services
Target to Servers or Clusters

□ RDBMS Security Store
Modify Settings

3.4 Installing Oracle Banking Enterprise Default Management

The detailed installation steps are present in the Oracle Banking Enterprise Default Management Installation Guide - Silent Installation.

3.5 Configuring SSL

One way SSL between the presentation and application WebLogic server is supported. The detailed configuration is explained below:

> **Note**
>
> Procure an external CA signed certificate before proceeding further. Follow the instructions below to install the certificate once the certificate is available.

Step 1 Import the Certificate into a Java Trust Keystore

Execute the following command:

```
keytool -import -trustcacerts -alias sampletrustself -keystore
SampleTrust.jks -file SampleSelfCA.cer.der -keyalg RSA
```

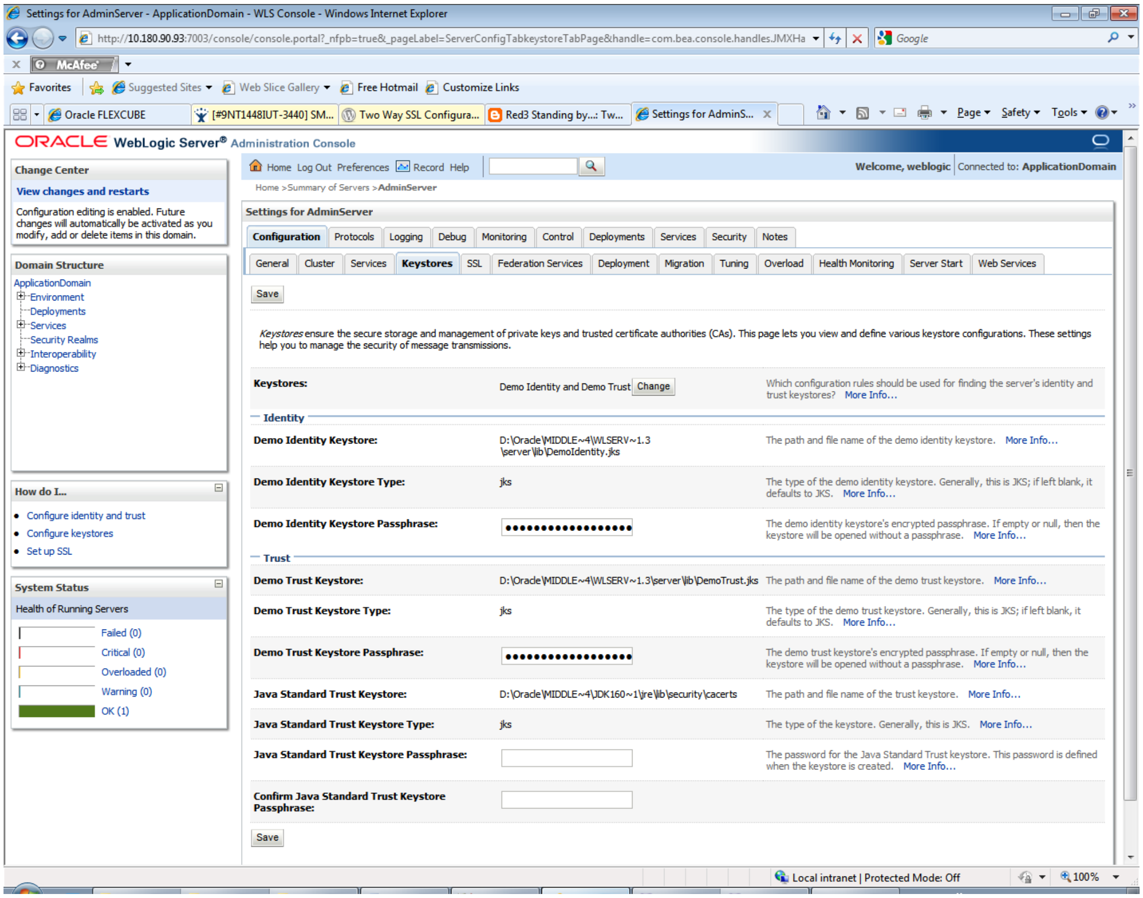
```
keytool -import -alias `hostname -f` -file `hostname -f`.cer -keystore <JAVA_
HOME>/jre/lib/security/cacerts -storepass changeit -noprompt
```

Step 2 Configure Application Domain’s WebLogic with Custom Identity and Trust Keystores

To configure the application domain’s WebLogic:

1. Open WebLogic admin console and navigate to **Home --> Summary of Servers --> AdminServer**. Click the **Keystores** tab.

Figure 3–5 Keystores



Configuration editing is enabled. Future changes will automatically be activated as you modify, add or delete items in this domain.

Settings for AdminServer

Keystores ensure the secure storage and management of private keys and trusted certificate authorities (CAs). This page lets you view and define various keystore configurations. These settings help you to manage the security of message transmissions.

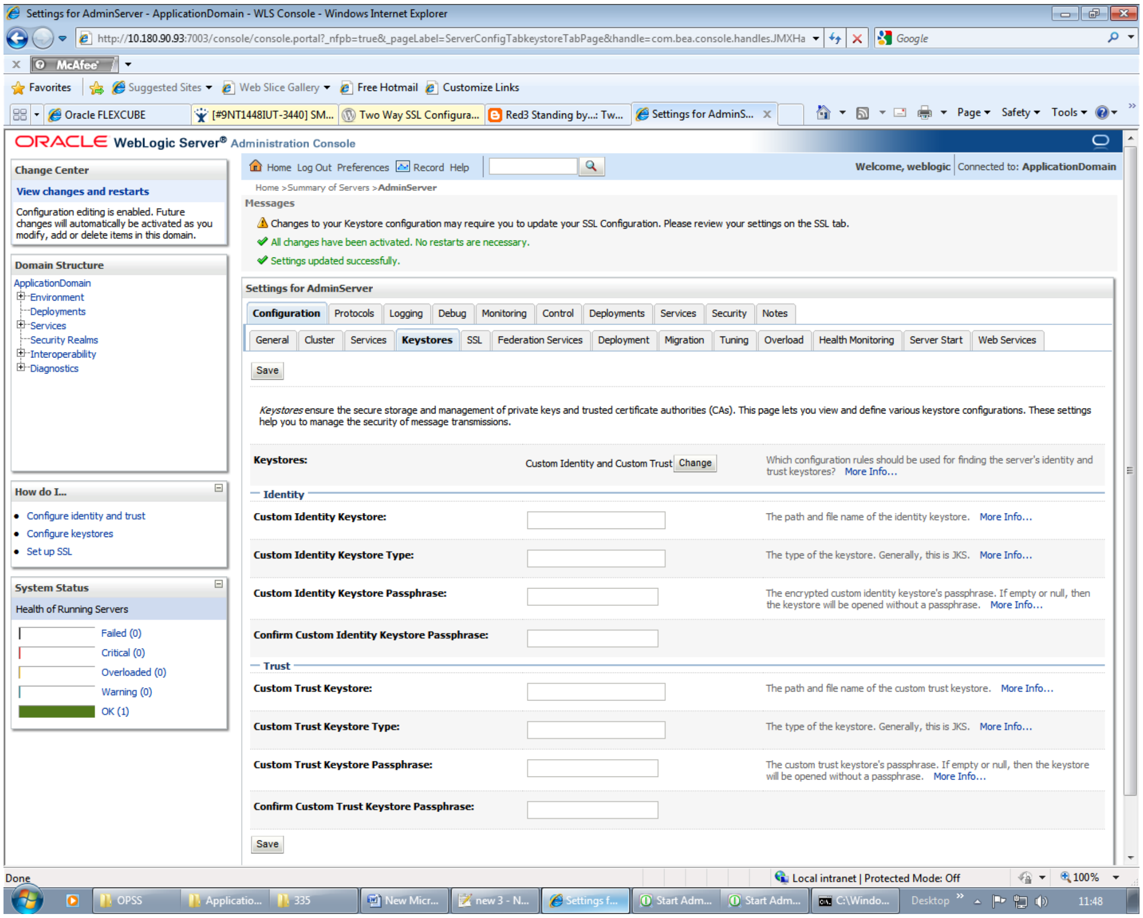
Keystores:	Demo Identity and Demo Trust	Which configuration rules should be used for finding the server's identity and trust keystores? More Info...
Identity		
Demo Identity Keystore:	D:\Oracle\MIDDLE~4\WLSERV~1.3 \server\lib\DemoIdentity.jks	The path and file name of the demo identity keystore. More Info...
Demo Identity Keystore Type:	jks	The type of the demo identity keystore. Generally, this is JKS; if left blank, it defaults to JKS. More Info...
Demo Identity Keystore Passphrase:	••••••••••••••••••	The demo identity keystore's encrypted passphrase. If empty or null, then the keystore will be opened without a passphrase. More Info...
Trust		
Demo Trust Keystore:	D:\Oracle\MIDDLE~4\WLSERV~1.3\server\lib\DemoTrust.jks	The path and file name of the demo trust keystore. More Info...
Demo Trust Keystore Type:	jks	The type of the demo trust keystore. Generally, this is JKS; if left blank, it defaults to JKS. More Info...
Demo Trust Keystore Passphrase:	••••••••••••••••••	The demo trust keystore's encrypted passphrase. If empty or null, then the keystore will be opened without a passphrase. More Info...
Java Standard Trust Keystore:	D:\Oracle\MIDDLE~4\JDK160~1\jre\lib\security\cacerts	The path and file name of the trust keystore. More Info...
Java Standard Trust Keystore Type:	jks	The type of the keystore. Generally, this is JKS. More Info...
Java Standard Trust Keystore Passphrase:		The password for the Java Standard Trust keystore. This password is defined when the keystore is created. More Info...
Confirm Java Standard Trust Keystore Passphrase:		

2. Click the **Change** button.

3. Select **Custom Identity and Java Standard Trust** option from the list.

4. Click the **Save** button.

Figure 3–6 Keystores - Identity and Trust



Configuration editing is enabled. Future changes will automatically be activated as you modify, add or delete items in this domain.

Messages

- Changes to your Keystore configuration may require you to update your SSL Configuration. Please review your settings on the SSL tab.
- All changes have been activated. No restarts are necessary.
- Settings updated successfully.

Settings for AdminServer

Keystores ensure the secure storage and management of private keys and trusted certificate authorities (CAs). This page lets you view and define various keystore configurations. These settings help you to manage the security of message transmissions.

Keystores:	Custom Identity and Custom Trust	Which configuration rules should be used for finding the server's identity and trust keystores? More Info...
Identity		
Custom Identity Keystore:		The path and file name of the identity keystore. More Info...
Custom Identity Keystore Type:		The type of the keystore. Generally, this is JKS. More Info...
Custom Identity Keystore Passphrase:		The encrypted custom identity keystore's passphrase. If empty or null, then the keystore will be opened without a passphrase. More Info...
Confirm Custom Identity Keystore Passphrase:		
Trust		
Custom Trust Keystore:		The path and file name of the custom trust keystore. More Info...
Custom Trust Keystore Type:		The type of the keystore. Generally, this is JKS. More Info...
Custom Trust Keystore Passphrase:		The custom trust keystore's passphrase. If empty or null, then the keystore will be opened without a passphrase. More Info...
Confirm Custom Trust Keystore Passphrase:		

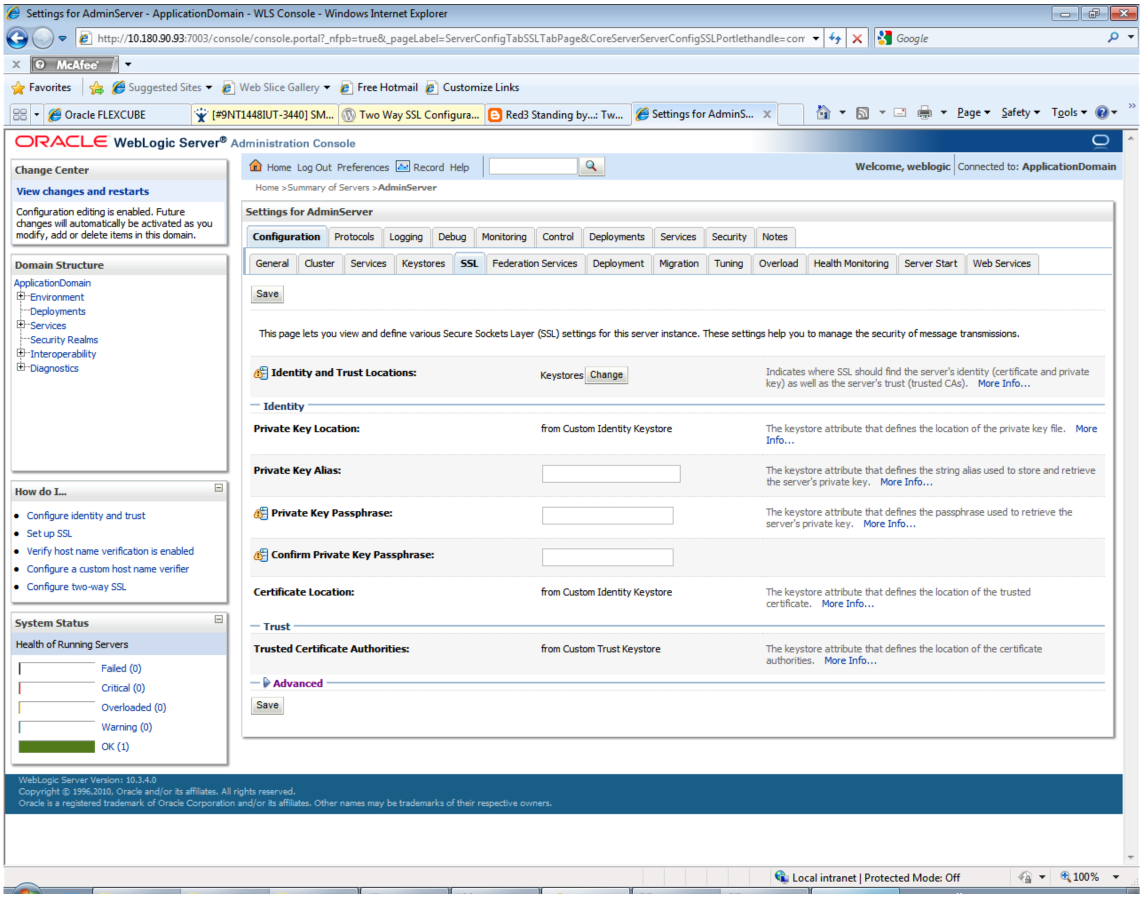
5. Enter the following details in the **Identity** and **Trust** sections:

Table 3–1 Keystore Configuration

Field	Value
Identity	
Custom Identity Keystore	Absolute path of `hostname -f`_identity.jck file
Custom Identity Keystore Type	JCKES
Custom Identity Keystore Passphrase	***
Confirm Custom Identity Keystore Passphrase	***

6. Enter the passphrases that were used while creating Identity Keystore and certificate.
7. Click the **Save** button.
8. Click the **SSL** Tab.

Figure 3–7 SSL



Settings for AdminServer - ApplicationDomain - WLS Console - Windows Internet Explorer

ORACLE WebLogic Server® Administration Console

Change Center

View changes and restarts

Configuration editing is enabled. Future changes will automatically be activated as you modify, add or delete items in this domain.

Home >Summary of Servers >AdminServer

Settings for AdminServer

This page lets you view and define various Secure Sockets Layer (SSL) settings for this server instance. These settings help you to manage the security of message transmissions.

Identity and Trust Locations:	Keystores Change	Indicates where SSL should find the server's identity (certificate and private key) as well as the server's trust (trusted CAs). More Info...
Identity		
Private Key Location:	from Custom Identity Keystore	The keystore attribute that defines the location of the private key file. More Info...
Private Key Alias:		The keystore attribute that defines the string alias used to store and retrieve the server's private key. More Info...
Private Key Passphrase:		The keystore attribute that defines the passphrase used to retrieve the server's private key. More Info...
Confirm Private Key Passphrase:		
Certificate Location:	from Custom Identity Keystore	The keystore attribute that defines the location of the trusted certificate. More Info...
Trust		
Trusted Certificate Authorities:	from Custom Trust Keystore	The keystore attribute that defines the location of the certificate authorities. More Info...

WebLogic Server Version: 10.3.4.0
Copyright © 1996,2010, Oracle and/or its affiliates. All rights reserved.
Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

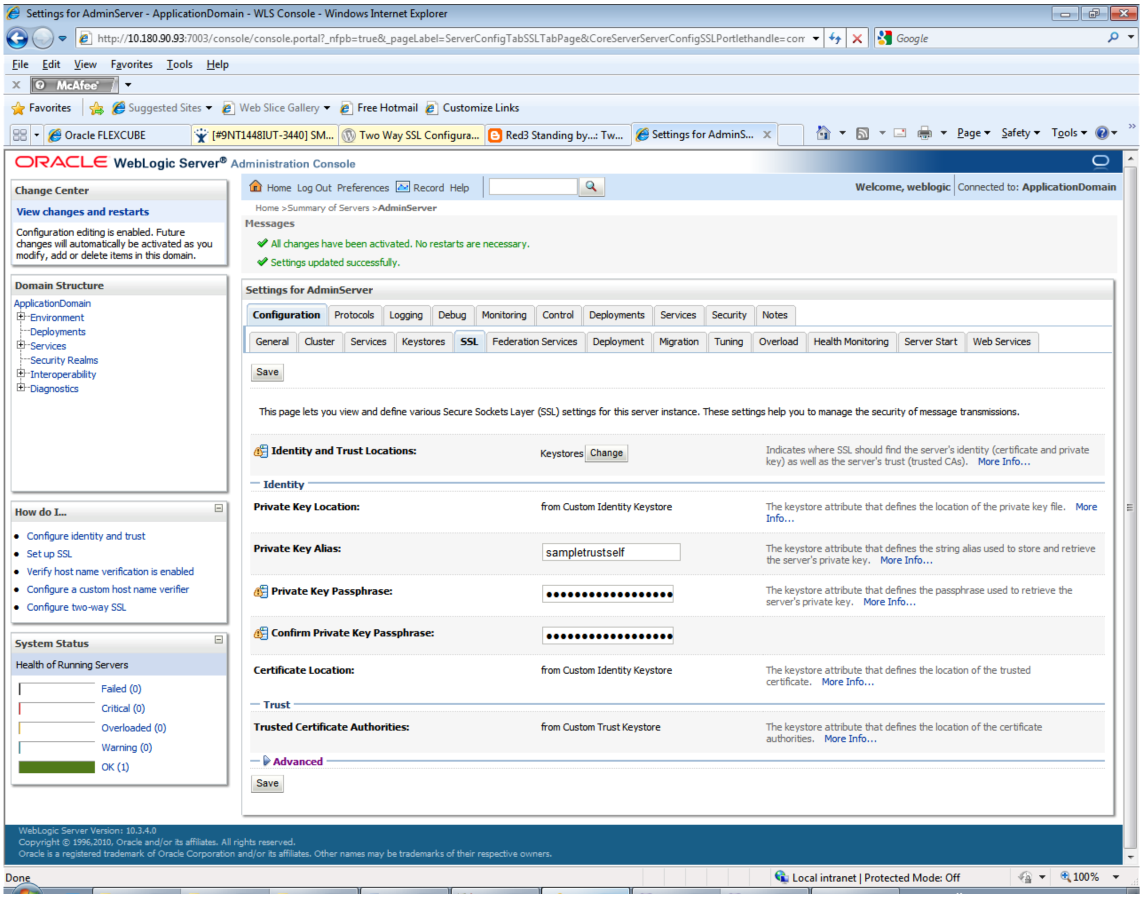
9. Enter the following details in the **Identity** section:

Table 3–2 SSL Configuration

Field	Value
Private Key Alias	`hostname -f`
Private Key Passphrase	***
Confirm Private Key Passphrase	***

10. Enter the passphrases that were used while creating the certificate.

Figure 3–8 SSL Configuration



Configuration editing is enabled. Future changes will automatically be activated as you modify, add or delete items in this domain.

Messages

✓ All changes have been activated. No restarts are necessary.

✓ Settings updated successfully.

Settings for AdminServer

This page lets you view and define various Secure Sockets Layer (SSL) settings for this server instance. These settings help you to manage the security of message transmissions.

Identity and Trust Locations: Keystores — Indicates where SSL should find the server's identity (certificate and private key) as well as the server's trust (trusted CAs). More Info...

Identity

Private Key Location: from Custom Identity Keystore — The keystore attribute that defines the location of the private key file. More Info...

Private Key Alias: sampletrustself — The keystore attribute that defines the string alias used to store and retrieve the server's private key. More Info...

Private Key Passphrase: •••••••••••••••••• — The keystore attribute that defines the passphrase used to retrieve the server's private key. More Info...

Confirm Private Key Passphrase: ••••••••••••••••••

Certificate Location: from Custom Identity Keystore — The keystore attribute that defines the location of the trusted certificate. More Info...

Trust

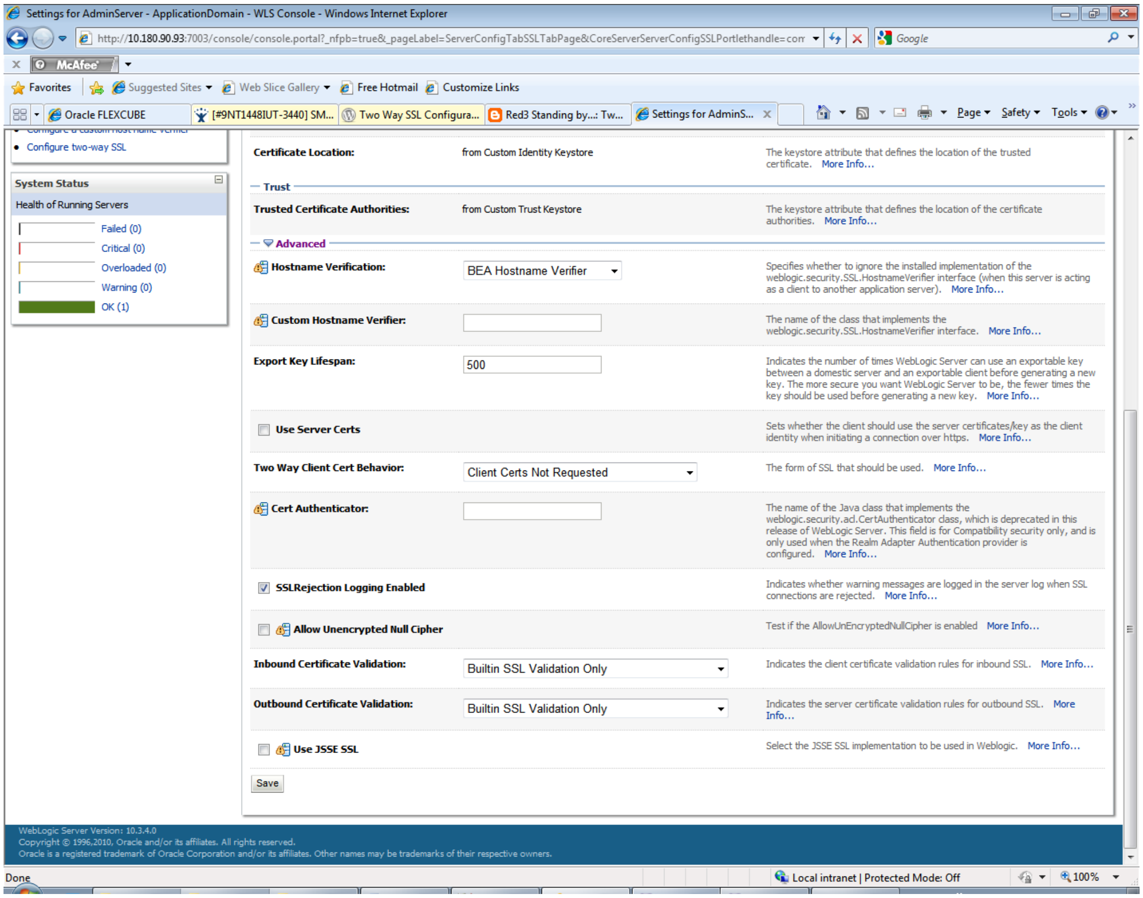
Trusted Certificate Authorities: from Custom Trust Keystore — The keystore attribute that defines the location of the certificate authorities. More Info...

WebLogic Server Version: 10.3.4.0
Copyright © 1996,2010, Oracle and/or its affiliates. All rights reserved.
Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

11. Click the **Save** button.

12. Click the **Advanced** link. Ensure that **Two Way Client Cert Behavior** is set to **Client Certs Not Requested**.

Figure 3–9 SSL - Advanced


Settings for AdminServer - ApplicationDomain - WLS Console - Windows Internet Explorer
http://10.180.90.93:7003/console/console.portal?_nfpb=true&_pageLabel=ServerConfigTabSSLTabPage&CoreServerServerConfigSSLPortlethandle=com
Google
McAfee
Favorites
Suggested Sites
Web Slice Gallery
Free Hotmail
Customize Links
Oracle FLEXCUBE
[#9NT1448IUT-3440] SM...
Two Way SSL Configura...
Red3 Standing by...: Tw...
Settings for AdminS...
Page
Safety
Tools
Configure two-way SSL
System Status
Health of Running Servers
Failed (0)
Critical (0)
Overloaded (0)
Warning (0)
OK (1)
Done
Local intranet | Protected Mode: Off
100%

Certificate Location:	from Custom Identity Keystore	The keystore attribute that defines the location of the trusted certificate. More Info...
Trust		
Trusted Certificate Authorities:	from Custom Trust Keystore	The keystore attribute that defines the location of the certificate authorities. More Info...
Advanced		
Hostname Verification:	BEA Hostname Verifier	Specifies whether to ignore the installed implementation of the weblogic.security.SSL.HostnameVerifier interface (when this server is acting as a client to another application server). More Info...
Custom Hostname Verifier:		The name of the class that implements the weblogic.security.SSL.HostnameVerifier interface. More Info...
Export Key Lifespan:	500	Indicates the number of times WebLogic Server can use an exportable key between a domestic server and an exportable client before generating a new key. The more secure you want WebLogic Server to be, the fewer times the key should be used before generating a new key. More Info...
☐ Use Server Certs		Sets whether the client should use the server certificates/key as the client identity when initiating a connection over https. More Info...
Two Way Client Cert Behavior:	Client Certs Not Requested	The form of SSL that should be used. More Info...
Cert Authenticator:		The name of the Java class that implements the weblogic.security.acl.CertAuthenticator class, which is deprecated in this release of WebLogic Server. This field is for Compatibility security only, and is only used when the Realm Adapter Authentication provider is configured. More Info...
☑ SSLRejection Logging Enabled		Indicates whether warning messages are logged in the server log when SSL connections are rejected. More Info...
☐ Allow Unencrypted Null Cipher		Test if the AllowUnEncryptedNullCipher is enabled More Info...
Inbound Certificate Validation:	Builtin SSL Validation Only	Indicates the client certificate validation rules for inbound SSL. More Info...
Outbound Certificate Validation:	Builtin SSL Validation Only	Indicates the server certificate validation rules for outbound SSL. More Info...
☐ Use JSSE SSL		Select the JSSE SSL implementation to be used in Weblogic. More Info...

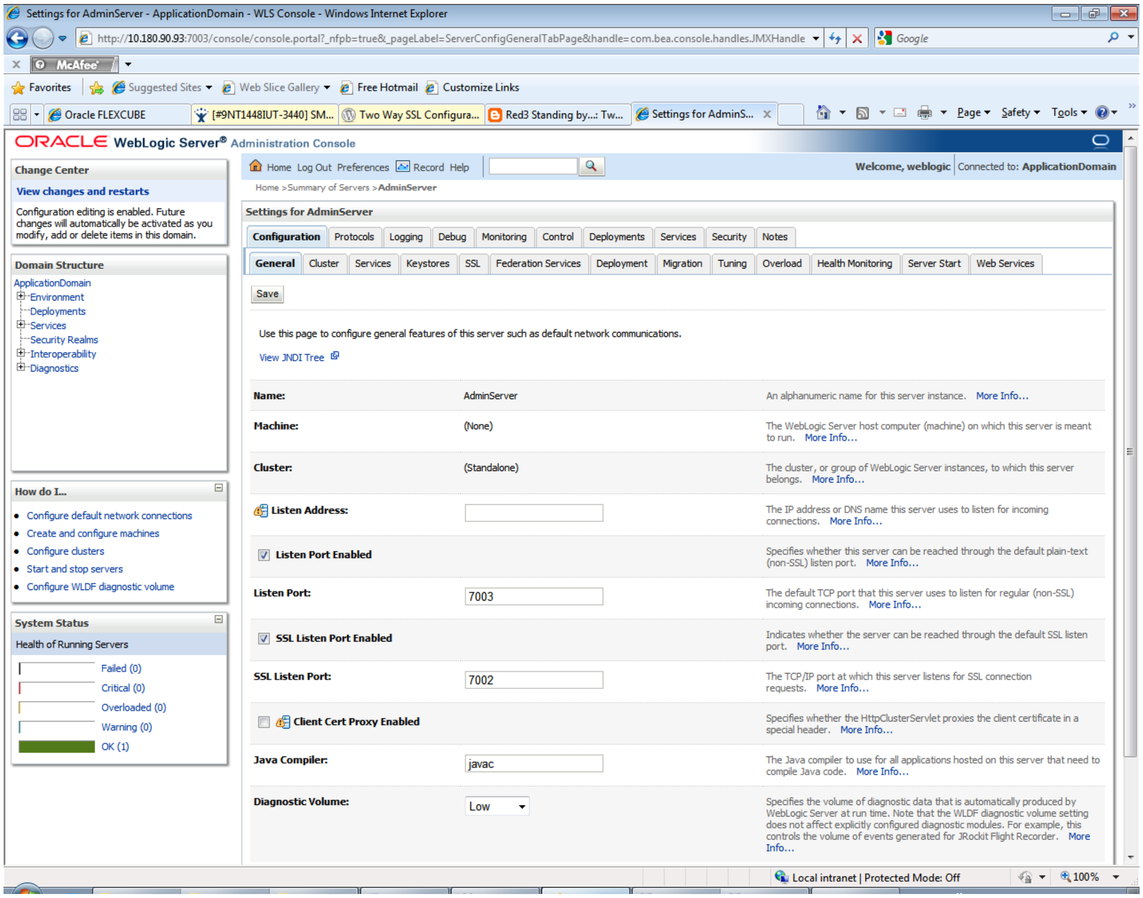
Save

WebLogic Server Version: 10.3.4.0
Copyright © 1996,2010, Oracle and/or its affiliates. All rights reserved.
Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

13. Click the **General** tab. Select the **SSL Listen Port Enabled** check box.

14. Select the **Use JSSE SSL flag**.

Figure 3–10 General



Configuration editing is enabled. Future changes will automatically be activated as you modify, add or delete items in this domain.

Use this page to configure general features of this server such as default network communications.

Name:	AdminServer	An alphanumeric name for this server instance. More Info...
Machine:	(None)	The WebLogic Server host computer (machine) on which this server is meant to run. More Info...
Cluster:	(Standalone)	The cluster, or group of WebLogic Server instances, to which this server belongs. More Info...
Listen Address:		The IP address or DNS name this server uses to listen for incoming connections. More Info...
☑ **Listen Port Enabled**		Specifies whether this server can be reached through the default plain-text (non-SSL) listen port. More Info...
Listen Port:	7003	The default TCP port that this server uses to listen for regular (non-SSL) incoming connections. More Info...
☑ **SSL Listen Port Enabled**		Indicates whether the server can be reached through the default SSL listen port. More Info...
SSL Listen Port:	7002	The TCP/IP port at which this server listens for SSL connection requests. More Info...
□ **Client Cert Proxy Enabled**		Specifies whether the HttpClusterServlet proxies the client certificate in a special header. More Info...
Java Compiler:	javac	The Java compiler to use for all applications hosted on this server that need to compile Java code. More Info...
Diagnostic Volume:	Low	Specifies the volume of diagnostic data that is automatically produced by WebLogic Server at run time. Note that the WLDF diagnostic volume setting does not affect explicitly configured diagnostic modules. For example, this controls the volume of events generated for JRockit Flight Recorder. More Info...

15. Click the **Save** button.

Step 3 Restart Admin Server

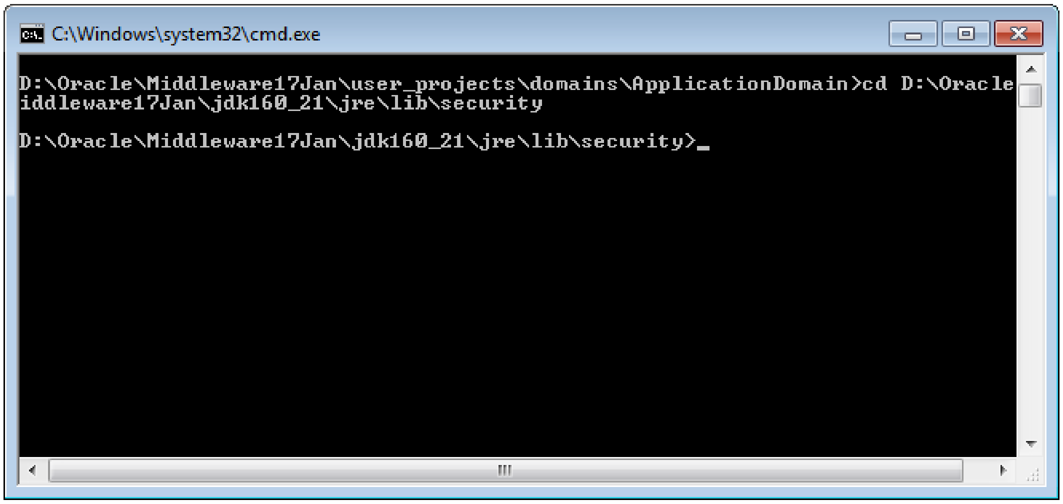
Restart the admin server of the Application Domain. Check the log file of admin server to ensure successful loading of the SSL configuration.

Step 4 Import Certificate in the JRE of Presentation Domain

To import the certificate:

1. Go to <MIDDLEWARE_HOME>\<JDK_HOME>\jre\lib\security

Figure 3–11 Presentation Domain Path



```
D:\Oracle\Middleware17Jan\user_projects\domains\ApplicationDomain>cd D:\Oracle
iddleware17Jan\jdk160_21\jre\lib\security

D:\Oracle\Middleware17Jan\jdk160_21\jre\lib\security>_
```

2. Execute the following command:

   ```
   keytool -import -alias sampletrustself -file D:\SampleSelfCA.cer -
   keystore cacerts
   ```

   Enter the keystore password when prompted to import the certificate in the JRE of the presentation domain.

3. Execute the following command:

   ```
   keytool -import -alias sampletrustself -file D:\SampleSelfCA.cer -
   keystore cacerts
   ```

   Enter the keystore password when prompted to import the certificate in the JRE of the presentation domain.

4. Finally, restart the admin server of the Presentation Domain.

Step 5 Web Services Authentication Configuration

All the host application web services are secured using the OWSM security policies.

The policy to be applied to the web service is defined in config/properties/SecurityAnnotations.properties

Sample entries are as follows:

com.ofss.fc.app.party.service.core.MDMPartyApplicationService=policy:oracle/ wss_saml_token_over_ssl_service_policy

- In an SSL enabled environment, oracle/wss_saml_token_over_ssl_service_policy is used.
- @Policy annotation is added at the server startup in BootstrapServlet.

OR

com.ofss.fc.app.party.service.core.MDMPartyApplicationService=policy:oracle/http_saml20_token_bearer_service_policy

SAML Token Strategy for Third Party Applications

The following sample enumerates one of the SAML token specifications that third party applications can use:

Note

The signature, certificate, digest and other encryption related values are changed.

```
<saml:Assertion Version="2.0"
xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion" ID="SAML-
nW7XYMp231fHjvTtM0JxFA22" IssueInstant="2016-08-
25T14:40:34Z"><saml:Issuer>www.oracle.com</saml:Issuer><dsig:Signa
ture
xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"><dsig:SignedInfo><
dsig:CanonicalizationMethod
Algorithm="http://www.w3.org/2001/10/xml-exc-
c14n#"/><dsig:SignatureMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-
sha1"/><dsig:Reference URI="#SAML-
nW7XYMp231fHjvTtM0JxFA22"><dsig:Transforms><dsig:Transform
Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-
signature"/><dsig:Transform
Algorithm="http://www.w3.org/2001/10/xml-exc-
c14n#"/></dsig:Transforms><dsig:DigestMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/><dsig:DigestVa
lue>vKTL+kQYWTadssdsdxl4dt6kXvc=</dsig:DigestValue></dsig:Referenc
e></dsig:SignedInfo><dsig:SignatureValue>Wdsdadasdsa8addasdaadasda
daddasdasdasdadasdadasdadsdsdds1hmglF0s98kfLtfrEOpRRGn4xNO2z/Ju+KC
TtA5Y4E0ZuHZN5DF2no2mXwTOVZRo0moTRlT5woUFi62iXnLLky+UTpVW5boi3QXdt
qsMI6oscbkgbrrigx5SMbJiR+kNni7vpg7UB2EBI5nLTGsRu4+383zggK5ETWRCAV9
O7Zp/iT5m0KuY0XctLEDAlSuM4069xrJgviMvuH9F3dgMjN/Dwy2pMr3VRsQ5gkMyY
IRNJOvr4DzilckTSORU3chXja7CQDxjGm44mX84yL7OuRaRWfOql8HaA==</dsig:S
ignatureValue><dsig:KeyInfo><dsig:X509Data><dsig:X509Certificate>M
IIC7TCCAdWgAwIBdasdasdasdasdasdadadasdasdasdasdadSHxUe75mI51BSbDim
hMz4TprGhxG7jKDsthcnlWqxlCtJPgZeSR76HI/JGYIqozccKk303Dnc9y1YfqV73v
A/o2opXjzNSBC33ruovq9SiZz4F7v8clmp9wChI6V4AcC0Ojp8</dsig:X509Certi
ficate><dsig:X509IssuerSerial><dsig:X509IssuerName>CN=orakey,
O="oracle
C=us"</dsig:X509IssuerName><dsig:X509SerialNumber>473970469</dsig:
X509SerialNumber></dsig:X509IssuerSerial><dsig:X509SubjectName>CN=
orakey, O="oracle
C=us"</dsig:X509SubjectName><dsig:X509SKI>SG/lnWm3TKwkxoW6KmkBPUyE
0C4=</dsig:X509SKI></dsig:X509Data></dsig:KeyInfo></dsig:Signature
><saml:Subject><saml:NameID
Format="urn:oasis:names:tc:SAML:1.1:nameid-
format:unspecified">nikhilt</saml:NameID><saml:SubjectConfirmation
Method="urn:oasis:names:tc:SAML:2.0:cm:bearer"/></saml:Subject><sa
ml:Conditions NotBefore="2016-08-25T14:40:34Z" NotOnOrAfter="2016-
```

```
08-29T02:00:34Z"/><saml:AuthnStatement AuthnInstant="2016-08-
25T14:40:34Z"><saml:AuthnContext><saml:AuthnContextClassRef>urn:oa
sis:names:tc:SAML:2.0:ac:classes:Password</saml:AuthnContextClassR
ef></saml:AuthnContext></saml:AuthnStatement></saml:Assertion>
```

This XML token is required to be compressed and Base64 encoded and pre-pended with the following string "oit ". This token is added to the HTTP header attribute 'Authorization'. The final token looks as follows:

Note

The actual string so generated is shortened.

```
oit
H4sIAAAAAAAAAJ1XWZOiyhr8Kx2eR28Pu4gxdgSrIIvNIqJviMUim0KByK+/2O30sW
d6JubeJ6is/LKyFsr0e+3n2Yyta1DBpCyeXFDVw3M+wr+ho6cuz4p6dqPMR01VzEq/
TupZ4eegnsFgZrO6NhuIM/9H/ehJEeajG/5cbGhvqHyLLyx/TezALbrwBfh0el7I63
Of+pdQXfZ+wj2l8oD/+Hbz8FwaCLGtuDAAA
```

Some of the key fields in the XML token are enumerated below:

Table 3–3 Key fields in the XML token

XML-tag / attribute	Description
<saml:Issuer>	Default value 'www.oracle.com'
<dsig:DigestValue>	Digest is computed for the entire token, minus the Signature node.
<dsig:SignatureValue>	Signature is calculated for the entire token, with the digest value also being signed.
<dsig:X509Certificate>	The public key to be used in signature verification.

Step 6 Web Service SSL configuration

By default, SSLv3 should be disabled. The steps to disable SSLv3 protocol on Weblogic are as follows:

1. The weblogic.security.SSL.protocolVersion command-line argument lets you specify which protocol is used for SSL connections.
2. After enabling/configuring the SSL for weblogic server, append the following option to the JAVA_ OPTIONS variable.

   -Dweblogic.security.SSL.protocolVersion=TLS1

   -Dweblogic.security.SSL.minimumProtocolVersion=TLSv1.1

Note

If you do not specify the above property, it takes SSLv3 by default.

3.6 Post Installation Configuration

The security practices that should always be followed are listed below:

- Set the proper permissions for users accessing databases. You could also implement roles to manage privileges. Check whether permissions are correctly set in operating system. If these are not correctly set, there may be a security loophole.
- Implement TDE column encryption on the sensitive data.

4 Security Features

This chapter outlines the specific security mechanisms offered by Oracle Banking Enterprise Default Management.

4.1 Security Model

Application security requirements arise from the need to protect data, first, from accidental loss and corruption, and second, from deliberate unauthorized attempts to access or alter that data.

Secondary concerns include protecting against undue delays in accessing or using data, or even against interference to the point of denial of service.

The global costs of such security breaches run up to billions of dollars annually, and the cost to individual companies can be severe, sometimes catastrophic.

The critical security features that provide these protections are:

- **Authentication**: Ensures that only authorized individuals get access to the system and data.
- **Authorization**: Ensures access control to system privileges and data. This builds on authentication to ensure that individuals only get appropriate access. Oracle Database Vault will be used for this purpose.
- **Audit**: Allows administrators to detect attempted breaches of the authentication mechanism and attempted or successful breaches of access control.

The Oracle Banking Enterprise Default Management Security Architecture is explained in detail in the next section.

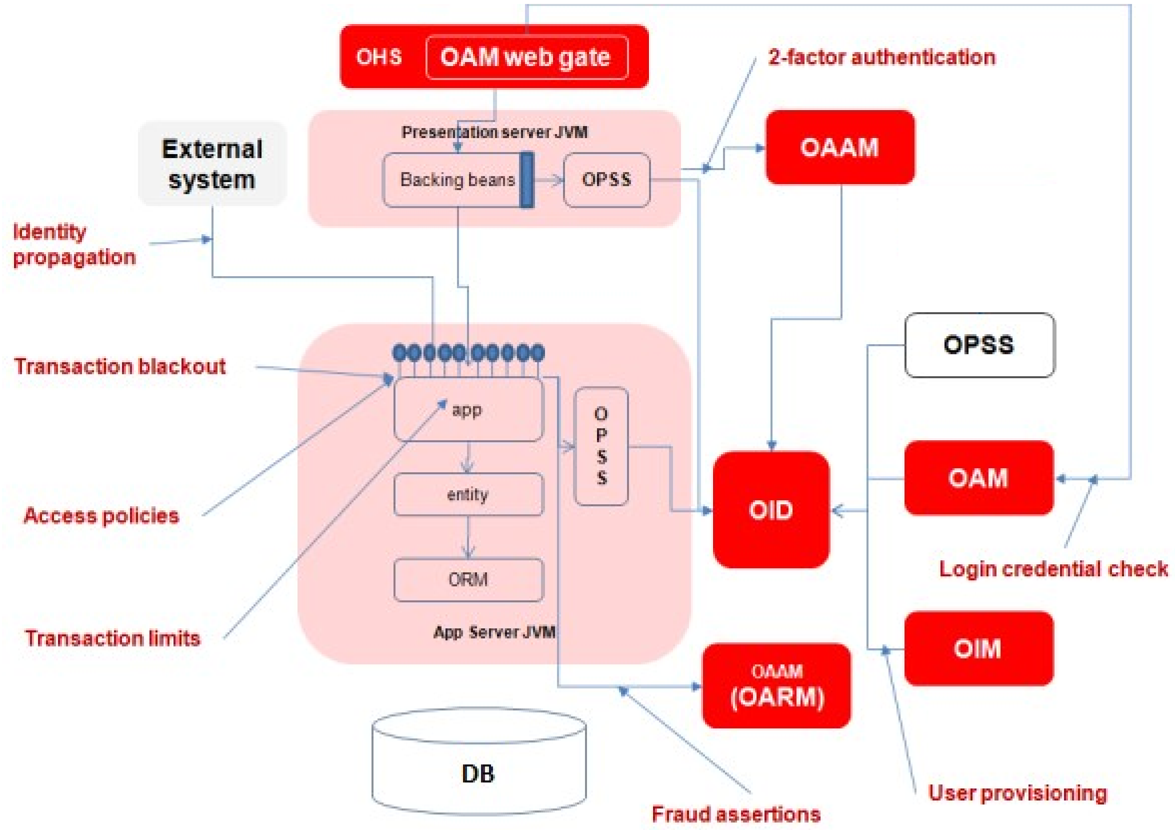
4.2 Security Architecture

Oracle Banking Enterprise Default Management comprises of several modules that interface with various systems in an enterprise to transfer or share data. This data is generated during business activity that takes place during teller operations or processing. While managing the transactions that are within OBEDM’s domain, it also needs to consider security and identity management, and the uniform way in which these services need to be consumed by all applications in the enterprise. This is possible if these capabilities can be externalized from the application itself and are implemented within products that are specialized to handle such services. Examples of these services include authentication against an enterprise identity-store, creating permissions and role-based authorization model that controls access to not only the components of the application, but also the data that is visible to the user based on fine-grained entitlements.

The security is provided in the product with the use of OIM or Local Security option. Below diagrams show the high-level architecture of OIM Based Security and Local Security.

4.2.1 OIM Based Security

Figure 4–1 Security - Participating Systems



The participating systems are as follows:

- Oracle Identity Manager (OIM) to be used for managing user provisioning.
- Oracle Access Manager (OAM) to be used for managing declarative authentication and SSO.
- Oracle Platform Security Services (OPSS) to be used for runtime evaluation of authentication/authorization.
- Oracle Adaptive Access Manager (OAAM)/Oracle Adaptive Risk Manager (OARM) to be used for step-up authentication and fraud management.
- Oracle Internet Directory (OID) is used as the identity/policy store.

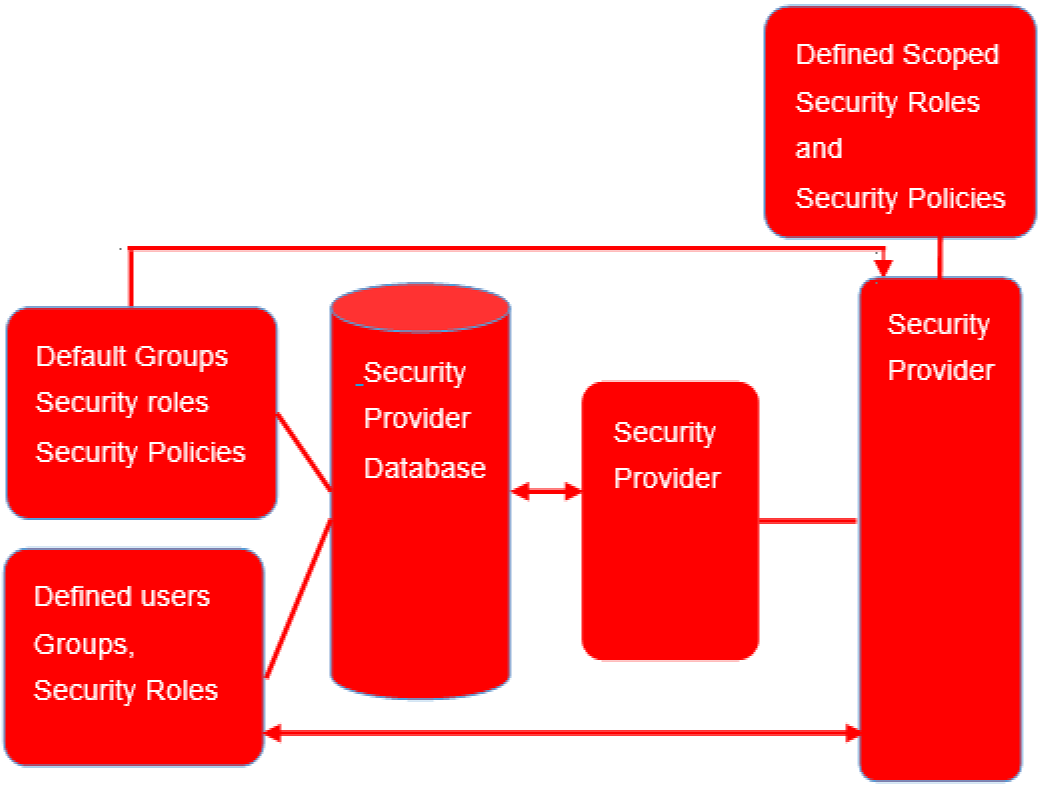
See the document Oracle® Collaboration Suite Security Guide at http://docs.oracle.com/cd/B25553_01/collab.1012/b25494/toc.htm for configuration details of the mentioned applications.

4.2.2 Local Security

This is applicable only for Oracle Banking Enterprise Collections and not for Oracle Banking Enterprise Recovery.

- Local Security option is provided using the Oracle database.
- SQLAuthenticator is configured in weblogic security realm.
- Users, groups, and roles are stored in the database.
- User, Role, and Entitlement management are done using the internal screens developed as a part of local security module.

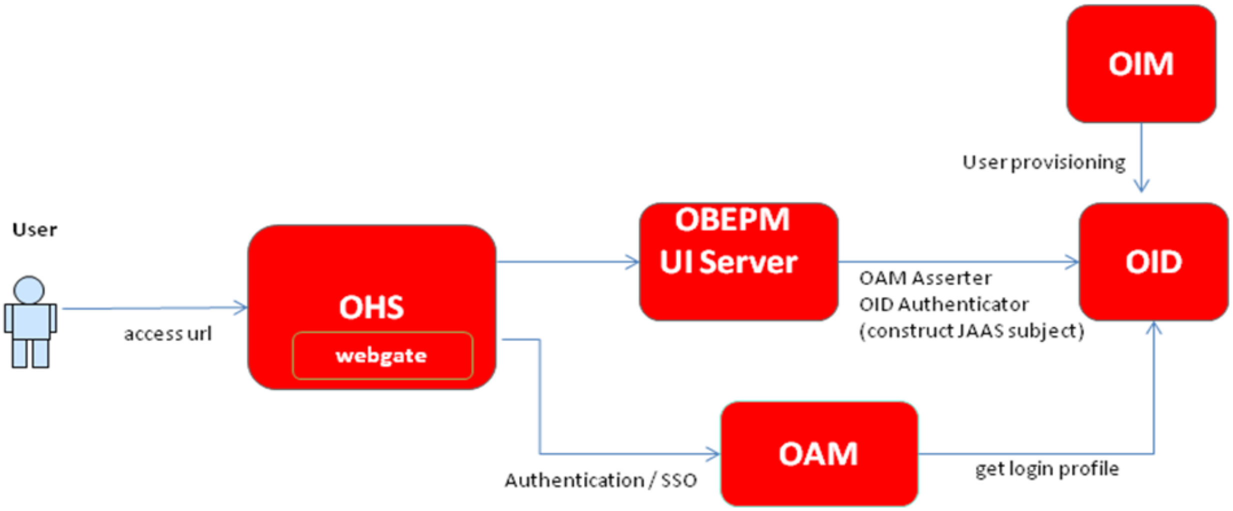
Figure 4–2 Local Security – High Level Architecture


Defined Scoped Security Roles and Security Policies
Default Groups Security roles Security Policies
Security Provider Database
Security Provider
Security Provider
Defined users Groups, Security Roles

4.3 Configuring and Using Authentication

Oracle Banking Enterprise Default Management uses OAM to authenticate users.

Figure 4–3 Authentication and Single Sign On



Data flow is as follows:

- OAM gets login profile from OID.
- OAM intercepts access call to Oracle Banking Enterprise Default Management and authenticates user.
- OAM ensures single sign-on across participating applications (configurable).
- SSO across various enterprise applications for internal users.

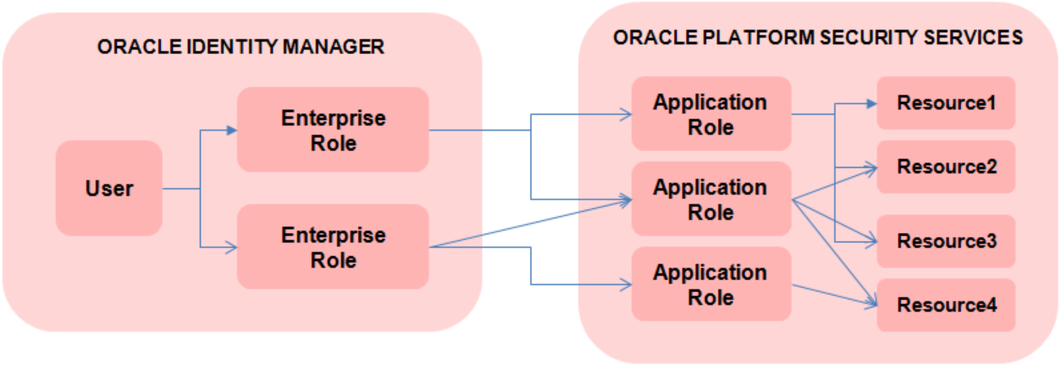
4.4 Configuring and Using Access Control

Authorization includes primarily two processes:

- Permitting only certain users to access, process, or alter transactions
- Applying varying limitations on user access or actions. The limitations placed on (or removed from) users can apply to transactions

Oracle Banking Enterprise Default Management uses OPSS Entitlements for authorization.

Figure 4–4 OPSS Entitlements - Users / Roles / Services



The features are:

- User belongs to the enterprise.
- Users mapped to enterprise roles (used organization-wide).
- Enterprise roles mapped to application roles (application roles used within the application).
- Access policies defined for services defined on application roles.

The application roles are mapped to the enterprise roles (or OID groups) that are managed within the identity store. The association between enterprise role and application role is many-to-many.

Oracle Banking Enterprise Default Management' security solution implements the following factory-shipped roles:

Table 4–1 Factory-Shipped Roles

Role	Description
OBP - read only	This is a read-only role that has inquiry privileges for a number of screens. For screens that have capabilities to change data, these users will have access only to inquiry services.
System Administrator	This role has privileges to perform and approve configuration parameters. The configuration is mostly done during implementation such as technical configuration and server configurations such as managing resource adapters.
Loans Officer	This role has privileges to evaluate, authorize, or recommend approval of commercial, real estate, or credit loans. This user advises borrowers on financial status and methods of payments. Entitlements include loans servicing.
Broker	This role is a customer-facing role involved in selling products to customer. All party-related inquiries and origination capabilities are given to these set of users. Entitlements include party identification, account details, party relationship inquiries, and so on.
Customer	This role provides entitlements for internet banking capabilities. Entitlements include funds transfer, installment payment, partial payoff, and so on.
Product Manager	This role provides entitlements for defining products and offers. Entitlements include offers and products creation.
Due Diligence Officer	This role provides entitlements for performing Know Your Customer (KYC), due diligence, onboarding or recertification process. Entitlements include collecting, analyzing, verifying and archiving legal and KYC documentation.

Industry experience feeds the OBP security model and the entitlements thereof. The latest set of factory-shipped access policies is available in the host mediapack that can be downloaded from e-delivery. These policies are seeded by the Policy-Store setup utility (PIT) during installation. At run time, these are managed using Authorization Policy Manager (APM), which is the GUI of the Oracle Entitlements Server (OES).

4.5 Configuring and Using Security Audit

Oracle Banking Enterprise Default Management relies on the Oracle Fusion Middleware Audit Framework for security audits.

The configuration and usage is explained in detail in the document Oracle® Fusion Middleware Application Security Guide - Configuring and Managing Auditing at http://docs.oracle.com/cd/E23943_01/core.1111/e10043/audpolicy.htm.

4.6 Configuring and Using TDE

Oracle Banking Enterprise Default Management relies on Oracle® Database Advanced Security for encrypting sensitive data.

The configuration is explained in detail in Oracle® Database Advanced Security Administrator's Guide.

OBEDM supports both TDE Tablespace Encryption as well as TDE Column Encryption.

Steps to perform TDE, with sample commands, as shown below:

1. Create Directories in all respective node servers.

```
mkdir -p -m 0700 /oracle/app/admin/IN5FMT/wallet
ssh orkxintdb10 "mkdir -p -m 0700
/oracle/app/admin/IN5FMT/wallet"
ssh orkxintdb11 "mkdir -p -m 0700
/oracle/app/admin/IN5FMT/wallet"
ssh orkxintdb12 "mkdir -p -m 0700
/oracle/app/admin/IN5FMT/wallet"

ssh orkxintdb10 "mkdir -p -m 0700
/oracle/app/database/11.2.0.2/dbhome_1/admin/IN5FMT/wallet"
ssh orkxintdb11 "mkdir -p -m 0700
/oracle/app/database/11.2.0.2/dbhome_1/admin/IN5FMT/wallet"
ssh orkxintdb12 "mkdir -p -m 0700
/oracle/app/database/11.2.0.2/dbhome_1/admin/IN5FMT/wallet"
```

2. Create wallet on all nodes of server.

   orapki wallet create -wallet /oracle/app/admin/IN5FMT/wallet -pwd 'iQlpcQZunsEMUU5dsfzLxoFKnOQ2bcpdp' -auto_login

3. Restart database.
4. Set Master Key from sqlplus.

```
orapki wallet display -wallet /oracle/app/admin/IN5FMT/wallet
-pwd 'iQlpcQZunsEMUU5dsfzLxoFKnOQ2bcpdp'
ALTER SYSTEM SET ENCRYPTION KEY AUTHENTICATED BY
"iQlpcQZunsEMUU5dsfzLxoFKnOQ2bcpdp";
```

5. Shutdown database.
6. Copy wallets into directories of all servers.

```
cd /oracle/app/admin/IN5FMT/wallet
scp -p * oracle@orkxintdb10:/oracle/app/admin/IN5FMT/wallet
scp -p * oracle@orkxintdb11:/oracle/app/admin/IN5FMT/wallet
scp -p * oracle@orkxintdb12:/oracle/app/admin/IN5FMT/wallet
```

```
cp -p * /oracle/app/database/11.2.0.2/dbhome_
1/admin/IN5FMT/wallet/
scp -p *
oracle@orkxintdb10:/oracle/app/database/11.2.0.2/dbhome_
1/admin/IN5FMT/wallet
scp -p *
oracle@orkxintdb11:/oracle/app/database/11.2.0.2/dbhome_
1/admin/IN5FMT/wallet
scp -p *
oracle@orkxintdb12:/oracle/app/database/11.2.0.2/dbhome_
1/admin/IN5FMT/wallet
```

7. Startup database.

8. For TDE Tablespace encryption, create tablespace as <Original>_Encrypted and give quota to owner.

```
CREATE TABLESPACE "FMTAPP_ENCRYPTED" DATAFILE SIZE 512M
AUTOEXTEND ON NEXT 104857600 MAXSIZE UNLIMITED
LOGGING ONLINE PERMANENT BLOCKSIZE 8192
EXTENT MANAGEMENT LOCAL AUTOALLOCATE SEGMENT SPACE MANAGEMENT
AUTO
ENCRYPTION USING 'AES256' DEFAULT STORAGE(ENCRYPT);

alter user FMTAPP quota unlimited on FMTAPP_ENCRYPTED;
```

9. Move the tables with sensitive data in the encrypted tablespace.

```
alter table FMTAPP.SAVING_GOAL move tablespace FMTAPP_
ENCRYPTED;
alter table FMTAPP.TXN_CATEGORY move tablespace FMTAPP_
ENCRYPTED;
alter table FMTAPP.CUST_TXNS move tablespace FMTAPP_ENCRYPTED;
```

10. Rebuild the indexes.

```
alter index FMTAPP.TXN_DATE_AC_INDEX rebuild;
alter index FMTAPP.TXN_UID_IDX rebuild;
alter index FMTAPP.CUST_TXN_ID_IDX rebuild;
alter index FMTAPP.SG_CUSTOMER_NUM_IDX rebuild;
```

11. For TDE column encryption, check for foreign key usage. TDE cannot be used to encrypt columns that are used in a foreign key. Verifying whether a column is used as part of a foreign key can be accomplished by examining the Oracle data dictionary.

12. Encrypt column using TDE.

```
table customers modify (credit_card encrypt);
create table billing_information ( first_name varchar2(40)
,last_name varchar2(40) ,card_number varchar2(19) encrypt
using 'AES256');
```

4.7 Securing Outbound Interactions

Oracle Banking Enterprise Default Management interacts with external systems like Oracle Analytics Publisher (formerly know as Business Intelligence Publisher), Oracle Customer Hub (OCH). These interactions are synchronous and asynchronous in nature.

Synchronous communication is achieved using JAX-WS.

The outbound webservice configurations are present in flx_fw_config_out_ws_cfg_b.

The configurations include URL, Service ID, StubService, and timeout. The IP address and port of the external system is defined in flx_fw_config_var_b.

For example, in case of Oracle Analytics Publisher,

url=http://{servername}:{serverport}/xmlpserver/services/PublicReportService?wsdl

timeOut=10000

stubService=com.oracle.xmlns.oxp.service.publicreportservice.PublicReportServiceService

The security credentials are stored in WebLogic connectors defined during installation.

Asynchronous communication is achieved using remote JMS queue.

The queue configurations are present in flx_fw_config_all_b, where category_id = 'EndpointConfig'. The IP address and port of the external system is defined in flx_fw_config_var_b.

For example, in case of OCH,

OCH.QUEUE_CONNECTION_FACTORY=jms/aia/AIA_CustomerJMSQueueCF

OCH.QUEUE=jms/aia/AIA_CustomerJMSQueue

OCH.PROVIDER.URL=t3:// {servername}:{serverport}/

The security credentials are stored in WebLogic connectors defined during installation.

4.8 Securing Key Store

This section describes the securing key store details.

4.8.1 Generation

The certificate is regenerated during installation, with a default password. Therefore, it needs to be regenerated post installation.

To generate keystore 'cks-keystore.jceks', following command should be used:

```
keytool -genseckey -alias orakey -keypass <Password> -keyalg RSA -keysize
2048 -dname "CN=orakey, O=oracle C=us" -storetype jceks -keystore cks-
keystore.jceks -storepass <Password>
```

The command generates a public/private key pair for the entity. It creates a self-signed certificate that includes the public key and the distinguished name information. The certificate is associated with the private key in a keystore entry.

By default, the keystore files are generated with 2048 bit key. These are required to be packaged as part of the **com.ofss.fc.ixface.sms.jar** file. These certificates are located within encr folder in the **com.ofss.fc.ixface.sms.jar** file.

4.8.2 Certificate Validity and Regeneration

The certificate is valid for 90 days. This is the default validity period, if the validity option is not specified explicitly. On certificate expiry, it has to be regenerated and replaced in the encr folder within the **com.ofss.fc.ixface.sms.jar** file.

4.8.3 Generation with 2048 Bit Key

In order to generate higher than 128 bit key size, **Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy** files are required. These are available at the **Java SE download** page at http://www.oracle.com/technetwork/java/embedded/embedded-se/downloads/jce-7-download-432124.html

The zip file contains policy jars, which you need to copy to overwrite the jars present in the *{java.home}/jre/lib/security* directory. This allows for key strength above 128 bits.

5 Data Privacy and Security

This chapter explains the data privacy and security features offered by Oracle Banking Enterprise Default Management.

5.1 Data Minimization

The primary use cases depicting Personally Identifiable Information (PII) data flows are presented in the following diagrams:

Figure 5–1 Bank Admin as PII Data Originator

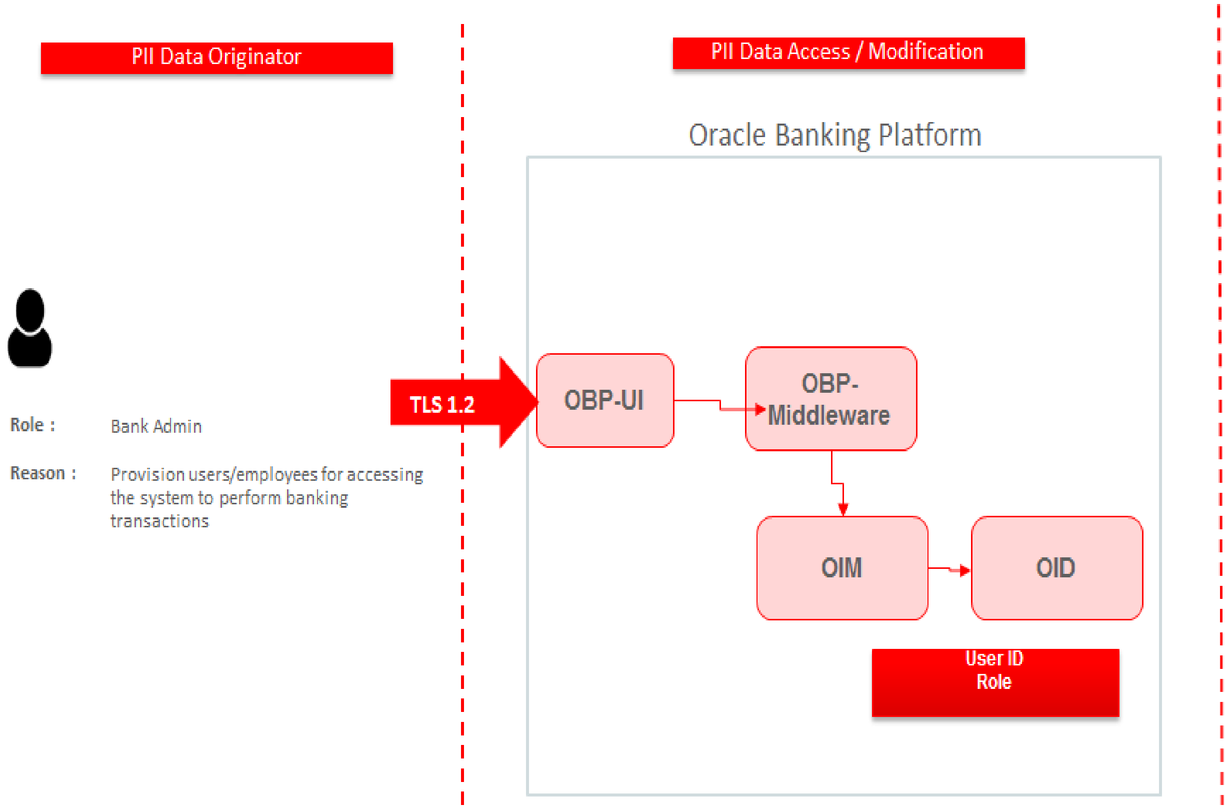

PII Data Originator
PII Data Access / Modification
Oracle Banking Platform
Role :
Bank Admin
Reason :
Provision users/employees for accessing the system to perform banking transactions
TLS 1.2
OBP-UI
OBP-Middleware
OIM
OID
User ID
Role

Figure 5–2 Bank Teller as PII Data Originator from Application Form

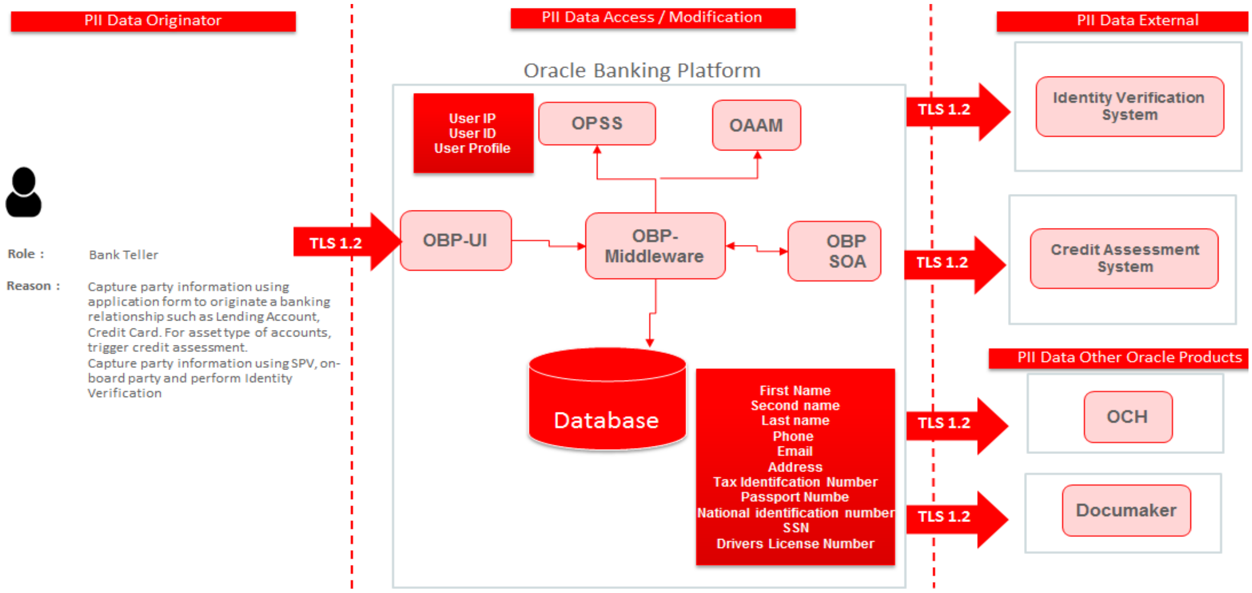
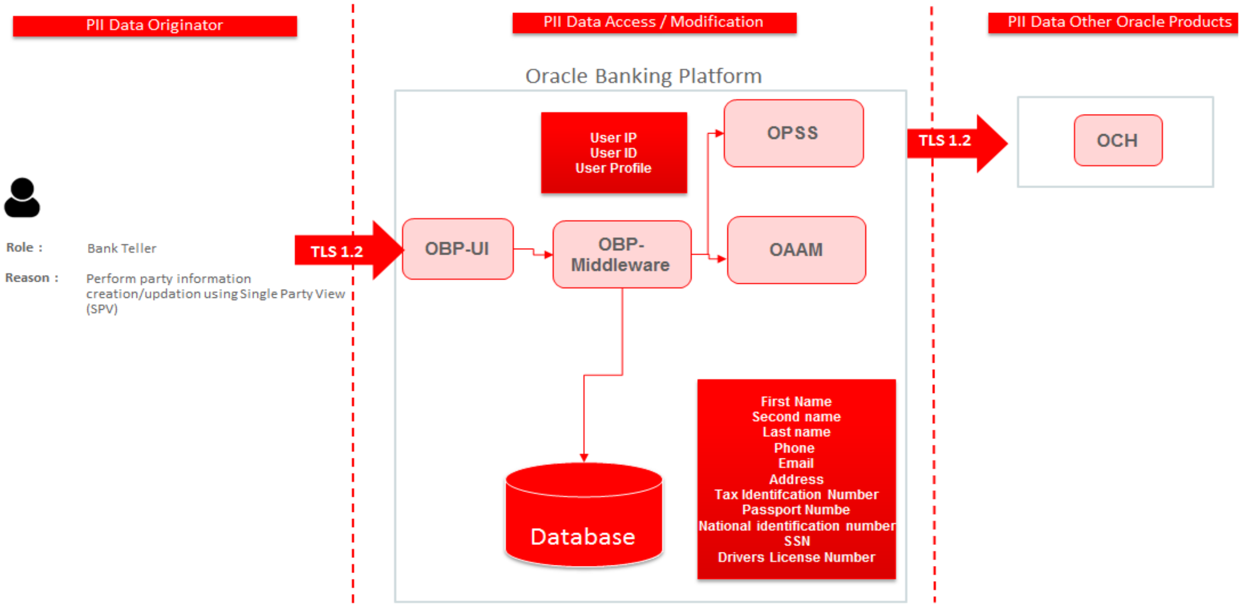

PII Data Originator
PII Data Access / Modification
PII Data External
Oracle Banking Platform
User IP
User ID
User Profile
OPSS
OAAM
TLS 1.2
Identity Verification System
Role :
Bank Teller
Reason :
Capture party information using application form to originate a banking relationship such as Lending Account, Credit Card. For asset type of accounts, trigger credit assessment.
Capture party information using SPV, on-board party and perform Identity Verification
TLS 1.2
OBP-UI
OBP-Middleware
OBP SOA
TLS 1.2
Credit Assessment System
PII Data Other Oracle Products
Database
First Name
Second name
Last name
Phone
Email
Address
Tax Identifcation Number
Passport Numbe
National identification number
SSN
Drivers License Number
TLS 1.2
OCH
TLS 1.2
Documaker

Figure 5–3 Bank Teller as PII Data Originator from Single Party View



5.2 Data Portability

Oracle Banking Enterprise Default Management enables bank users to extract the audit log in an industry standard, so that the file can be provided to a customer or system in a machine-readable format for easy interpretation.

- Interface Logging (Fast Path: OPA008): The user can use this screen for viewing and extracting log of payloads to external interfaces.

- Audit Text Based Search (Fast Path: BAM56): The user can use this screen for viewing and extracting log of entity maintenance in OBEDM.

Both Interface Logging (Fast Path: OPA008) and Audit Text Based Search (Fast Path: BAM56) allow the user to do criteria based search. The search results can be viewed on the screen as well as exported as Excel file format and further saved as a CSV (comma separated values) file format. As PII details are not part of transactions, no enhancements are required in Financial Transactions Log View.

The Party Export Data service helps in extracting data for the parties mentioned in the request. The request is logged and during End of Day batch process, the data is extracted and stored in machine readable formats.

5.3 Encryption

This section explains about encryption of PII data.

5.3.1 Key Management

Oracle Banking Enterprise Default Management encrypts and decrypts PII data using AES. Since it uses symmetric-key algorithm, key management is very critical.

The starting point in any private key management strategy is to create a comprehensive inventory of all keys, their locations and responsible parties. Private keys used must be kept secure as unauthorized individuals can intercept confidential communications or gain unauthorized access to critical systems. Failure to ensure proper segregation of duties means that administrator who generates the encryption keys can use them to access sensitive, regulated data.

Oracle Banking Enterprise Default Management, by default, implements secure storage and access to encryption key.

5.3.2 Secure Storage of Encryption Key

Java Key Store (JKS) is used to hold the encryption key. JKS file is created for each encryption key (For example, for card number encryption, a separate JKS file is created). The key store file, type and corresponding mapping properties are factory shipped with product jar.

Following are the Java key store parameters that are used:

Table 5–1 Java Key Store Parameters

Parameter	Value
Type	Secret Key
Algorithm	AES
Store Type	JCEKS (Triple DES)
Key Size	128
Alias	<<alias>>
Key Password	<<password>>
Store Password	<<password>>
Domain Name	<<domain-name>>
Key Store	<<key store file name>> For example, cks-keystore.jceks for card number

5.3.3 Secure Access of Encryption Key

For accessing the encryption key, the JKS requires valid alias and password. The alias and password are maintained using credential store resource adapter (com.ofss.fc.connector). Connector Instance is created for each encryption key. For example, JNDI Name: ra/FCRJConnectorKEYSTORE_CARD, for card number.

Credential mapping should be done for each JNDI / encryption key as follows:

Table 5–2 Encryption Key Parameters

Property	Value	Mapping / Usage
EIS User	<<alias>>	Alias used for key store
EIS Password	<<password>>	Store / Key password used for the key store

The credential store JNDI name is maintained in the configuration factory (DB based). The property ID has the key lookup name.

Table 5–3 Encryption Key Parameters

Configuration Type	Category	Prop ID	Prop Value
DB Based	CredentialConnector	CKS_RA_JNDIKEY (Format: <<keyLookupName>>_RA_ JNDIKEY)	a/FCRJConnectorKEYSTORE_ CARD

5.4 Tracking Technologies

OBP components have externalized their authentication needs to the Oracle security stack. The applications in the OBP suite do not generate, manipulate, collect or interpret cookies. However, the underlying weblogic and OPSS infrastructure on which OBP is deployed does use cookies for its authentication needs.

5.5 Separate Auditing and Detective Control Privileges

This section explains about auditing and detective control privileges.

5.5.1 Application Logs

Following application logs are supported in OBEDM:

- Financial Transactions
- Entity Maintenance
- Payloads to External Interfaces

PII data for logs are masked prior to logging.

5.6 Logging

This section explains about logging.

5.6.1 Application Logs

Following application logs are supported in OBEDM:

- Financial Transactions
- Entity Maintenance
- Payloads to External Interfaces

PII data for logs are masked prior to logging.

Appendix

This appendix lists the Secure Deployment Checklist which includes guidelines that help secure Oracle Banking Enterprise Default Management.

Secure Deployment Checklist

The following security checklist includes guidelines that help secure your installation:

1. Install only what is required.
2. Lock and expire default user accounts.
3. Enforce password management.
4. Practice the principle of least privilege.
    a. Grant necessary privileges only.
    b. Revoke unnecessary privileges from the PUBLIC user group.
    c. Restrict permissions on run-time facilities.
5. Enforce access controls effectively and authenticate clients stringently.
6. Restrict network access.
    a. Use a firewall.
    b. Never poke a hole through a firewall.
    c. Monitor who accesses your systems.
    d. Check network IP addresses.
    e. Encrypt network traffic.
    f. Harden the operating system.
7. Apply all security patches and workarounds.
8. Contact Oracle Security Products if you come across vulnerability in Oracle Database.