# Oracle® Retail Process Orchestration and Monitoring

Security Guide

Release 4.0.1

**F17860-01**

April 2019

Oracle Retail Processing Orchestration and Monitoring Security Guide, Release 4.0.1

F17860-01

Primary Author: Alex Meske

# Contents

# Send Us Your Comments

Oracle Retail Process Orchestration and Monitoring Guide, Release Release 4.0.1

Oracle welcomes customers' comments and suggestions on the quality and usefulness of this document.

Your feedback is important, and helps us to best meet your needs as a user of our products. For example:

- Are the implementation steps correct and complete?
- Did you understand the context of the procedures?
- Did you find any errors in the information?
- Does the structure of the information help you with your tasks?
- Do you need different information or graphics? If so, where, and in what format?
- Are the examples correct? Do you need more examples?

If you find any errors or have any other suggestions for improvement, then please tell us your name, the name of the company who has licensed our products, the title and part number of the documentation and the chapter, section, and page number (if available).

**Note:** Before sending us your comments, you might like to check that you have the latest version of the document and if any concerns are already addressed. To do this, access the Online Documentation available on the Oracle Technology Network Web site. It contains the most current Documentation Library plus all documents revised or released recently.

Send your comments to us using the electronic mail address: retail-doc_us@oracle.com

Please give your name, address, electronic mail address, and telephone number (optional).

If you need assistance with Oracle software, then please contact your support representative or Oracle Support Services.

If you require training or instruction in using Oracle software, then please contact your Oracle local office and inquire about our Oracle University offerings. A list of Oracle offices is available on our Web site at `http://www.oracle.com`.

# Preface

The *Oracle Retail Process Orchestration and Monitoring Guide* describes the tracking and managing of batch jobs.

## Audience

This guide is for system administrators and operations personnel, integrators and implementation staff personnel as well as users of the module.

## Documentation Accessibility

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

### Access to Oracle Support

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

## Customer Support

To contact Oracle Customer Support, access My Oracle Support at the following URL:

`https://support.oracle.com`

When contacting Customer Support, please provide the following:

- Product version and program/module name
- Functional and technical description of the problem (include business impact)
- Detailed step-by-step instructions to re-create
- Exact error message received
- Screen shots of each step you take

## Review Patch Documentation

When you install the application for the first time, you install either a base release (for example, 4.0) or a later patch release (for example, 4.0.1). If you are installing the base release, additional patch, and bundled hot fix releases, read the documentation for all releases that have occurred since the base release before you begin installation. Documentation for patch and bundled hot fix releases can contain critical information related to the base release, as well as information about code changes since the base release.

## Improved Process for Oracle Retail Documentation Corrections

To more quickly address critical corrections to Oracle Retail documentation content, Oracle Retail documentation may be republished whenever a critical correction is needed. For critical corrections, the republication of an Oracle Retail document may at times not be attached to a numbered software release; instead, the Oracle Retail document will simply be replaced on the Oracle Technology Network Web site, or, in the case of Data Models, to the applicable My Oracle Support Documentation container where they reside.

This process will prevent delays in making critical corrections available to customers. For the customer, it means that before you begin installation, you must verify that you have the most recent version of the Oracle Retail documentation set. Oracle Retail documentation is available on the Oracle Technology Network at the following URL:

`http://www.oracle.com/technetwork/documentation/oracle-retail-100266.html`

An updated version of the applicable Oracle Retail document is indicated by Oracle part number, as well as print date (month and year). An updated version uses the same part number, with a higher-numbered suffix. For example, part number E123456-02 is an updated version of a document with part number E123456-01.

If a more recent version of a document is available, that version supersedes all previous versions.

## Oracle Retail Documentation on the Oracle Technology Network

Oracle Retail product documentation is available on the following Web site:

`http://www.oracle.com/technetwork/documentation/oracle-retail-100266.html`

(Data Model documents are not available through Oracle Technology Network. You can obtain them through My Oracle Support.)

## 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 Introduction to Part I: Oracle Retail Applications

The following chapters provide guidance for administrators, developers, and system integrators who securely administer, customize, and integrate the Oracle Retail Applications:

- Pre-installation of Retail Infrastructure in WebLogic
- Post Installation of Retail Infrastructure in Database
- Post Installation of Retail Infrastructure in WebLogic
- Troubleshooting
- Importing Topology Certificate
- Using Self-Signed Certificates
- Functional Security for Applications Using Fusion Middleware
- ReST Services Security Consideration

# 2 Pre-installation of Retail Infrastructure in WebLogic

Oracle Retail applications are primarily deployed in Oracle WebLogic server at the Middleware tier. Java and forms-based applications rely upon Middleware infrastructure for complete security. This is separate from application-specific security features.

This chapter describes the pre-installation steps for secured setup of Oracle Retail infrastructure in WebLogic.

The following topics are covered in this chapter:

- JDK Hardening for Use with Retail Applications
- Pre-installation - Steps for Secured Setup of Oracle Retail Infrastructure in WebLogic
- Certificate Authority
- Obtaining an SSL Certificate and Setting up a Keystore
- Creating a WebLogic Domain
- Configuring the Application Server for SSL
- Configuring WebLogic Scripts if Admin Server is Secured
- Adding Certificate to the JDK Keystore for Installer
- Enforcing Stronger Encryption in WebLogic
- Securing Nodemanager with SSL Certificates
- Using Secured Lightweight Directory Access Protocol (LDAP)
- Webservice Security Policies
- Advanced Infrastructure Security

## JDK Hardening for Use with Retail Applications

See the following sections on JDK hardening for use with Retail applications:

- Upgrading JDK to Use Java Cryptography Extension
- Disabling Weak SSL Protocols and Obsolete Ciphers in JDK7

### Upgrading JDK to Use Java Cryptography Extension

You need to install the unlimited encryption Java Cryptography Extension (JCE) policy, if you want to use the strongest Cipher suites (256 bit_encryption). It is dependent on the Java Development Kit (JDK) version.

Using the following URL, download and install the JCE Unlimited Strength Jurisdiction Policy Files that correspond to the version of your JDK:

```
http://www.oracle.com/technetwork/java/javase/downloads/index.html
```

For JDK 7, download from the following URL, then replace the files in the `JDK/jre/lib/security` directory:

```
http://www.oracle.com/technetwork/java/javase/downloads/jce-7-download-432124.html
```

For JDK8, download the files from URL:

```
http://www.oracle.com/technetwork/java/javase/downloads/jce8-download-2133166.html
```

### Disabling Weak SSL Protocols and Obsolete Ciphers in JDK7

**Note:** This section is applicable only for applications using Java 7.

Edit the following lines in the `/jre/lib/security/java.security` file for the JDK:

```
jdk.certpath.disabledAlgorithms=MD2, RC4, RSA keySize < 1024
```

and

```
jdk.tls.disabledAlgorithms=SSLv3, SSLV2Hello, MD5withRSA, DH keySize < 768
```

**Note:** Restart the entire WebLogic instance using the JDK to enable changes to take effect once the JCE has been installed.

## Pre-installation - Steps for Secured Setup of Oracle Retail Infrastructure in WebLogic

Secured Sockets Layer (SSL) protocol allows client-server applications to communicate across a network in a secured channel. Client and server should both decide to use SSL to communicate secured information, like user credentials or any other secured information.

WebLogic Server supports SSL on a dedicated listen port. Oracle Forms are configured to use SSL as well. To establish an SSL connection, a Web browser connects to a WebLogic Server by supplying the SSL port and the secure Hypertext Transfer Protocol (HTTPS) protocol in the connection URL.

For example: `https://myserver:7002`

**Note:** You need to obtain a separate, signed SSL certificate for each host where the application will be deployed

The Security Guide focuses on securing Oracle Retail Applications in single node setup and not on applications deployed on clusters.

## Certificate Authority

The Certificate Authority or Certification Authority (CA) is an organization that provides digital certificates to entities and acts as a trusted third party. Certificates issued by commercial CAs are automatically trusted by most web browsers, devices, and applications. You should obtain certificates from a trusted CA or commercial CAs to ensure better security.

## Obtaining an SSL Certificate and Setting up a Keystore

**Note:** SSL certificates are used to contain public keys. With each public key there is an associated private key. It is critically important to protect access to the private key. Otherwise, the SSL messages may be decrypted by anyone intercepting the communications.

Perform the following steps to obtain an SSL certificate and setting up a keystore:

1. Obtain an identity (private key and digital certificates) and trust (certificates of trusted certificate authorities) for WebLogic Server.
2. Use the digital certificates, private keys, and trusted CA certificates provided by the WebLogic Server kit, the CertGen utility, Sun Microsystem's keytool utility, or a reputed vendor such as Entrust or Verisign to perform the following steps:
   1. Set appropriate `JAVA_HOME` and `PATH` to `java`.

      For example:

      ```
      export JAVA_HOME=/u00/webadmin/product/jdk export PATH=$JAVA_HOME/bin:$PATH
      ```

   2. Create a new keystore:

      ```
      keytool -genkey -keyalg RSA -keysize 2048 -keystore <keystore> -alias
      <alias>
      ```

      For example:

      ```
      keytool -genkey -keyalg RSA -keysize 2048 -keystore hostname.keystore
      -alias hostname
      ```

   3. Generate the signing request:

      ```
      keytool -certreq -keyalg RSA -file <certificate request file> -keystore
      <keystore> -alias <alias>
      ```

      For example:

      ```
      keytool -certreq -keyalg RSA -file hostname.csr -keystore hostname.keystore
      -alias hostname
      ```

   4. Submit the certificate request to the CA.
3. Store the identity and trust.

   Private keys and trusted CA certificates which specify identity and trust are stored in a keystore.

   The examples in the following steps use the same keystore to store all certificates:

   1. Import the root certificate into the keystore as shown in the following example:

```
keytool -import -trustcacerts -alias  verisignclass3g3ca -file Primary.pem
-keystore  hostname.keystore
```

A root certificate is either an unsigned public key certificate or a self-signed certificate that identifies the Root CA.

2. Import the intermediary certificate (if required) into the keystore as shown in the following example:

```
keytool -import -trustcacerts -alias oracleclass3g3ca -file Secondary.pem
-keystore hostname.keystore
```

3. Import the received signed certificate for this request into the keystore as shown in the following example:

```
keytool -import -trustcacerts -alias hostname -file cert.cer -keystore
hostname.keystore
```

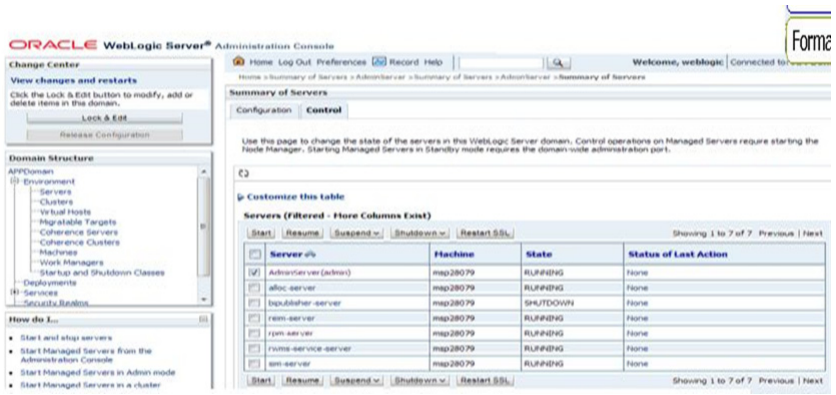
## Creating a WebLogic Domain

A WebLogic domain is created for Oracle Retail Applications as part of the installation. Different domains are created in different hosts for different applications in situations where applications are being managed by different users or deployed on different hosts. Once the domains are created, you need to enable the SSL ports if you have not enabled already.

Perform the following steps to enable the SSL:

1. Log in to WebLogic console using the Administrator user. For example, weblogic.
2. Navigate to <Domain> > Environment > Servers > < Servername> > Configuration > General tab.
3. Click **Lock & Edit**.
4. Select **SSL Listen Port Enabled**.
5. Assign the port number.
6. Click **Save**.
7. Click **Activate Changes**.
8. Restart SSL to enable the changes.

***Figure 2–1 Restarting the Admin Server***


ORACLE WebLogic Server® Administration Console
Home Log Out Preferences Record Help
Welcome, weblogic
Connected to
Forma
Change Center
View changes and restarts
Lock & Edit
Release Configuration
Domain Structure
APPDomain
Environment
Servers
Clusters
Virtual Hosts
Migratable Targets
Coherence Servers
Coherence Clusters
Machines
Work Managers
Startup and Shutdown Classes
Deployments
Services
Security Realms
How do I...
Start and stop servers
Start Managed Servers from the Administration Console
Start Managed Servers in Admin mode
Start Managed Servers in a cluster
Home >Summary of Servers >AdminServer >Summary of Servers >AdminServer >Summary of Servers
Summary of Servers
Configuration
Control
Customize this table
Start
Resume
Suspend
Shutdown
Restart SSL
Showing 1 to 7 of 7 Previous | Next

Click the Lock & Edit button to modify, add or delete items in this domain.

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 Servers in Standby mode requires the domain-wide administration port.

Servers (Filtered - More Columns Exist)

| | Server | Machine | State | Status of Last Action |
|---|---|---|---|---|
| ✓ | AdminServer(admin) | msp28079 | RUNNING | None |
| | alloc-server | msp28079 | RUNNING | None |
| | bipublisher-server | msp28079 | SHUTDOWN | None |
| | reim-server | msp28079 | RUNNING | None |
| | rpm-server | msp28079 | RUNNING | None |
| | rwms-service-server | msp28079 | RUNNING | None |
| | sim-server | msp28079 | RUNNING | None |

## Configuring the Application Server for SSL

Perform the following steps to configure the Application Server for SSL:

1. Configure the identity and trust keystores for WebLogic Server in the WebLogic Server Administration Console:

    1. In the Change Center of the Administration Console, click **Lock & Edit**.
    2. In the left pane of the Console, expand **Environment** and select **Servers**.
    3. Click the name of the server for which you want to configure the identity and trust keystores as shown in the following example:

        ```
        WLS_FORMS is for Forms server
        ```

    4. Select **Configuration**, then select **Keystores**.
    5. In the Keystores field, select the method for storing and managing private keys/digital certificate pairs and trusted CA certificates.

        The following options are available:

        **Demo Identity and Demo Trust** - The demonstration identity and trust keystores, located in the `BEA_HOME\server\lib` directory and the Java Development Kit (JDK) `cacerts` keystore, are configured by default. These are for development purpose only.

        **Custom Identity and Java Standard Trust** - A keystore you create and the trusted CAs defined in the `cacerts` file in the `JAVA_HOME\jre\lib\security` directory.

        **Custom Identity and Custom Trust [Recommended]** - Identity and trust keystores you create.

        **Custom Identity and Command Line Trust** - An identity keystore you create and command-line arguments that specify the location of the trust keystore.

    6. Select **Custom Identity** and **Custom Trust**.
    7. In the Identity section, define the following attributes for the identity keystore:

        **Custom Identity Keystore** - The fully qualified path to the identity keystore.

        **Custom Identity Keystore Type** - The type of the keystore. Generally, this attribute is Java KeyStore (JKS); if it is left blank, it defaults to JKS.

        **Custom Identity Keystore Passphrase** - The password you must enter when reading or writing to the keystore. This attribute is optional or required depending on the type of keystore. All keystores require the passphrase in order to write to the keystore. However, some keystores do not require the passphrase to read from the keystore. WebLogic Server only reads from the keystore, so whether or not you define this property depends on the requirements of the keystore.

    8. In the **Trust** section, define properties for the trust keystore.

        If you choose **Java Standard Trust** as your keystore, specify the password defined when creating the keystore.

    9. Confirm the password.

        If you choose **Custom Trust [Recommended]**, define the following attributes:

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

**Custom Trust Keystore Type** - The type of keystore. Generally, this attribute is JKS; if it is left blank, it defaults to JKS.

**Custom Trust Keystore Passphrase** - The password that you need to enter when reading or writing to the keystore. This attribute is optional or required depending on the type of keystore. All keystores require the passphrase in order to write to the keystore. However, some keystores do not require the passphrase to read from the keystore. WebLogic Server only reads from the keystore, so whether or not you define this property depends on the requirements of the keystore.

10. Click **Save**.

11. Click **Activate Changes** to activate these changes, in the Change Center of the Administration Console.

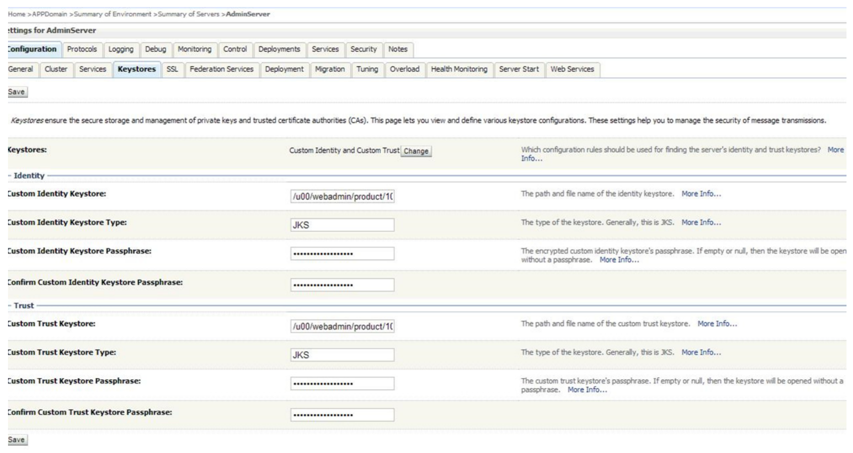
---

**Note:** Not all changes take effect immediately, some require a restart.

---

Figure 2–2 shows how to configure the Application Server for SSL.

***Figure 2–2 Configuring the Identity and Trust Keystores for WebLogic Server***



*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: | /u00/webadmin/product/1( | The path and file name of the identity keystore. More Info... |
| Custom Identity Keystore Type: | JKS | 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 open without a passphrase. More Info... |
| Confirm Custom Identity Keystore Passphrase: | •••••••••••••••••• | |
| – Trust | | |
| Custom Trust Keystore: | /u00/webadmin/product/1( | The path and file name of the custom trust keystore. More Info... |
| Custom Trust Keystore Type: | JKS | 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: | •••••••••••••••••• | |

For more information on configuring Keystores, see the *Administration Console Online Help*.

2. Set SSL configuration options for the private key alias and password in the WebLogic Server Administration Console:

   1. In the Change Center of the Administration Console, click **Lock & Edit**.

   2. In the left pane of the Console, expand **Environment** and select **Servers**.

   3. Click the name of the server for which you want to configure the identity and trust keystores.

4. Select **Configuration**, then select **SSL**.

   In the **Identity and Trust Locations**, the Keystore is displayed by default.

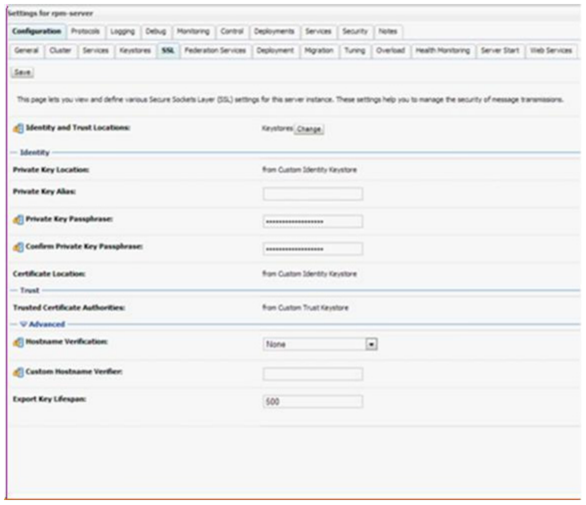
5. In the **Private Key Alias**, type the string alias that is used to store and retrieve the server's private key.
6. In the **Private Key Passphrase**, enter the keystore attribute that defines the passphrase used to retrieve the server's private key.
7. Save the changes.
8. Click the **Advanced** section of SSL tab.
9. In the **Hostname Verification**, select **None**.

   This specifies to ignore the installed implementation of the WebLogic.security.SSL.HostnameVerifier interface (this interface is generally used when this server is acting as a client to another application server).

   For Weblogic Server 10.3.6 and former versions, enable Java Secure Socket Extension by enabling Use JSSE SSL to provide high security.

10. Save the changes.

***Figure 2–3 Configure SSL***



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

Identity

Private Key Location: from Custom Identity Keystore

Private Key Alias:

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

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

Certificate Location: from Custom Identity Keystore

Trust

Trusted Certificate Authorities: from Custom Trust Keystore

Advanced

Hostname Verification: None

Custom Hostname Verifier:

Export Key Lifespan: 500

For more information on configuring SSL, see “Configure SSL” in the *Administration Console Online Help*.

All the server SSL attributes are dynamic; when modified through the Console. They cause the corresponding SSL server or channel SSL server to restart and use the new settings for new connections. Old connections will continue to run with the old configuration. You must reboot WebLogic Server to ensure that all the SSL connections exist according to the specified configuration.

Use the **Restart SSL** button on the Control: Start/Stop page to restart the SSL server when changes are made to the keystore files. You need apply the same for subsequent connections without rebooting WebLogic Server.

Upon restart, you will see entries like the following in the log:

```
<Mar 11, 2019 5:18:27 AM CDT> <Notice> <WebLogicServer> <BEA-000365> <Server
state changed to RESUMING>
<Mar 11, 2019 5:18:27 AM CDT> <Notice> <Server> <BEA-002613> <Channel
```

```
"DefaultSecure" is now ing on 10.141.15.214:57002 for protocols iiops, t3s, ldaps,
https.>
<Mar 11, 2019 5:18:27 AM CDT> <Notice> <Server> <BEA-002613> <Channel
"DefaultSecure[1]" is now ing on 127.0.0.1:57002 for protocols iiops, t3s, ldaps,
https.>
<Mar 11, 2019 5:18:27 AM CDT> <Notice> <WebLogicServer> <BEA-000329> <Started
WebLogic Admin Server "AdminServer" for domain "APPDomain" running in Production
Mode>
<Mar 11, 2019 5:18:27 AM CDT> <Notice> <WebLogicServer> <BEA-000365> <Server state
changed to RUNNING>
<Mar 11, 2019 5:18:27 AM CDT> <Notice> <WebLogicServer> <BEA-000360> <Server
started in RUNNING mode>
```

**Note:** For complete security of the WebLogic Server, it is recommended to secure both Administration as well the Managed Server where application is being deployed. You can choose to disable the non-SSL ports (HTTP). It is recommended to secure the Node Manager.

The steps to secure Node Manager is provided in the following section.

## Configuring WebLogic Scripts if Admin Server is Secured

Perform the following steps to configure the WebLogic scripts if Admin Server is secured:

1. Update the WebLogic startup/shutdown scripts with secured port and protocol to start/stop services.
2. Back up and update the following files in `<DOMAIN_HOME>/bin` with correct Admin server URLs:

   **startManagedWebLogic.sh:** `echo "$1 managedserver1 http://apphost1:7001"`

   **stopManagedWebLogic.sh:** `echo "ADMIN_URL defaults to t3://apphost1:7001 if not set as an environment variable or the second command-line parameter."`

   **stopManagedWebLogic.sh:** `echo "$1 managedserver1 t3://apphost1:7001 WebLogic`

   **stopManagedWebLogic.sh:** `ADMIN_URL="t3://apphost1:7001"`

   **stopWebLogic.sh:** `ADMIN_URL="t3://apphost1:7001"`
3. Change the URLs to:

   `t3s://apphost1:7002`

   `https://apphost1:7002`

## Adding Certificate to the JDK Keystore for Installer

You will need the Oracle Retail Application installer to run Java. In situations where Administration Server is secured using signed certificate, the Java keystore through which the installer is launched must have the certificate installed.

In case the installer is being run using the JDK deployed in `/u00/webadmin/product/jdk`, follow the steps as shown in Example 2–1.

***Example 2–1 Adding Certificate to the JDK Keystore for Installer***

```
apphost1:[10.3.6_apps] /u00/webadmin/ssl> keytool -import -trustcacerts -alias
apphost1 -file /u00/webadmin/ssl/apphost1.cer -keystore
/u00/webadmin/product/jdk/jre/lib/security/cacerts Enter keystore password:
Certificate was added to keystore
apphost1:[10.3.6_apps] /u00/webadmin/ssl>
```

## Enforcing Stronger Encryption in WebLogic

You should use a stronger encryption protocol in your production environment.

See the following sections to enable the latest SSL and cipher suites.

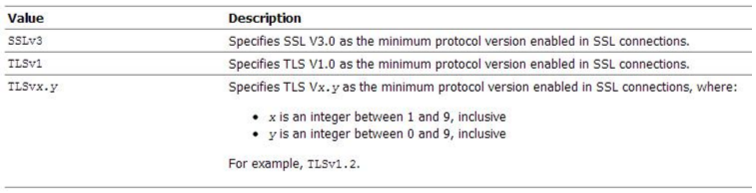
### SSL Protocol Version Configuration

In a production environment, Oracle you should use Transport Layer Security (TLS) Version 1.1 or higher for sending and receiving messages in an SSL connection.

To control the minimum versions of SSL Version 3.0 and TLS Version 1 that are enabled for SSL connections, do the following:

- Set the **WebLogic.security.SSL.minimumProtocolVersion=protocol** system property as an option in the command line that starts WebLogic Server. This system property accepts one of the following values for protocol:

***Figure 2–4 Values for Protocol of System Property***



| Value | Description |
|---|---|
| SSLv3 | Specifies SSL V3.0 as the minimum protocol version enabled in SSL connections. |
| TLSv1 | Specifies TLS V1.0 as the minimum protocol version enabled in SSL connections. |
| TLSvx.y | Specifies TLS Vx.y as the minimum protocol version enabled in SSL connections, where:<br>• x is an integer between 1 and 9, inclusive<br>• y is an integer between 0 and 9, inclusive<br>For example, TLSv1.2. |

- Set the following property in the startup parameters in WebLogic Managed server for enabling the higher protocol:

```
DWebLogic.security.SSL.minimumProtocolVersion=TLSv1.2
-Dhttps.protocols=TLSv1.2
```

**Note:** In case a protocol is set for Managed servers, the same should be set for the Administration server. Ensure that all the managed servers are down when making changes to the Administration server for setting up the protocol. Set the properties in the Administration server, then the Managed server.

#### Enabling Cipher in WebLogic SSL Configuration (For Weblogic 10.3.6 Domains)

Configure the `<ciphersuite>` element in the `<ssl>` element in the `<DOMAIN_ HOME>\server\config\config.xml` file in order to enable the specific Cipher Suite to use as follows:

**Note:** You need to ensure that the tag `<ciphersuite>` is added immediately after tab `<enabled>`.

```
<ssl>
<name>examplesServer</name>
<enabled>true</enabled>
<ciphersuite>TLS_RSA_WITH_AES_256_CBC_SHA</ciphersuite>
<-port>17002</-port>
...
</ssl>
```

---

**Note:** The above can be done using a `wlst` script.

For more information, go to `http://docs.oracle.com/cd/E24329_01/web.1211/e24422/ssl.htm#BABDAJJG`. You should bring down the managed server before making the changes.

---

## Securing Nodemanager with SSL Certificates

Perform the following steps for securing the Nodemanager with SSL certificates:

1. Navigate to `<BEA_HOME>/wlserver_10.3/common/nodemanager` (If you are using the Weblogic 12c domain, the location is `<DOMAIN_HOME>/nodemanager` ) and take a backup of `nodemanager.properties`.
2. Add the following similar entries to `nodemanager.properties`:

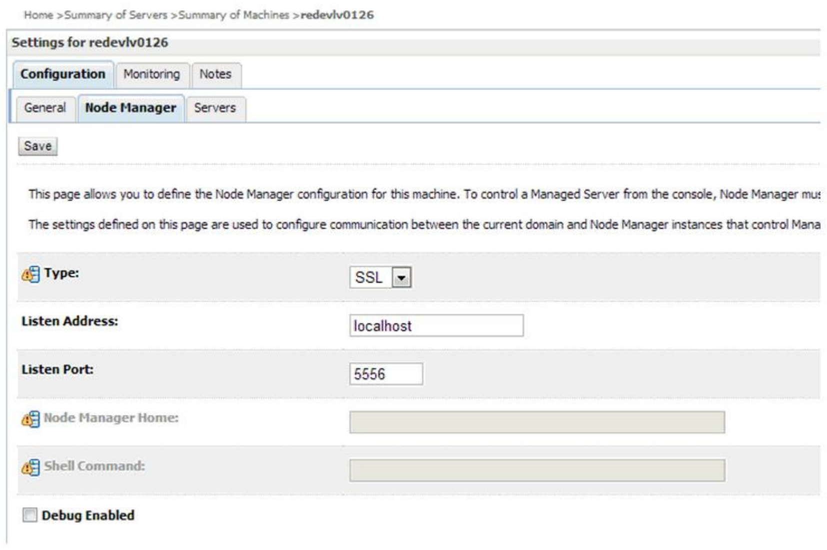
   ```
   KeyStores=CustomIdentityAndCustomTrust
   CustomIdentityKeyStoreFileName=/u00/webadmin/ssl/hostname.keystore
   CustomIdentityKeyStorePassPhrase=[password to keystore, this will get
   encrypted]
   CustomIdentityAlias=hostname
   CustomIdentityPrivateKeyPassPhrase=[password to keystore, this will get
   encrypted]
   CustomTrustKeyStoreFileName=/u00/webadmin/ssl/hostname.keystore
   SecureListener=true
   ```

3. If you are using Weblogic 10.3.6 and the earlier-version servers were enabled with JSSE, then Nodemanager start up script (startNodeManager.sh) should be added as the following parameter:

   ```
   -Dweblogic.security.SSL.enableJSSE=true
   ```

4. Log in to the WebLogic console, navigate to **Environment**, and then **Machines**.
5. Select the nodemanager created earlier and navigate to **Node Manager** tab.
6. In the Change Center, click **Lock & Edit**.
7. In the **Type** field, select **SSL** from the list.
8. Click **Save** and **Activate**.

***Figure 2–5 Securing the Nodemanager***


Home >Summary of Servers >Summary of Machines >redevlv0126

Settings for redevlv0126

Configuration | Monitoring | Notes

General | Node Manager | Servers

Save

This page allows you to define the Node Manager configuration for this machine. To control a Managed Server from the console, Node Manager mus

The settings defined on this page are used to configure communication between the current domain and Node Manager instances that control Mana

Type: SSL

Listen Address: localhost

Listen Port: 5556

Node Manager Home:

Shell Command:

Debug Enabled

9. Restart the WebLogic Domain for changes to take effect, after activating the changes.
10. You need to verify if the nodemanager is reachable in the **Monitoring** tab after the restart.

## Using Secured Lightweight Directory Access Protocol (LDAP)

The Application can communicate with the LDAP server on a secured port. It is recommended to use the secured LDAP server to protect usernames and passwords from being sent in clear text on the network.

For information on Configuring Secure Sockets Layer (SSL), see the *Oracle Fusion Middleware Administration Guide*.

It is important to import the certificates used in the LDAP server into the Java Runtime Environment (JRE) of the WebLogic server for SSL handshake, in case the secure LDAP is used for authentication.

For example:

1. Set `JAVA_HOME` and `PATH` to the JDK being used by WebLogic Domain.
2. Back up the `JAVA_HOME/jre/lib/security/cacerts` directory:

```
/u00/webadmin/product/jdk/jre/lib/security> cp -rp cacerts cacerts_ORIG
```

3. Import the Root and Intermediary (if required) certificates into the java keystore:

```
/u00/webadmin/product/jdk/jre/lib/security> keytool -import -trustcacerts
-alias verisignclass3g3ca -file ~/ssl/Primary.pem -keystore cacerts
/u00/webadmin/product/jdk/jre/lib/security> keytool -import -trustcacerts
-alias oracleclass3g3ca -file ~/ssl/Secondary.pem -keystore cacerts
```

4. Import the User certificate from LDAP server into the java keystore:

```
/u00/webadmin/product/jdk/jre/lib/security> keytool -import -trustcacerts
-alias hostname -file ~/ssl/cert.cer -keystore cacerts
```

**Note:** The default password for the JDK keystore is **changeit**.

The deployed application should be able to communicate with LDAP on the SSL port after successful SSL Handshake.

## Webservice Security Policies

You need to configure the user credentials and other security-related information at the service consumer and the app service provider layers. This provides end-to-end security between Web service consumer and provider.

The security policies certified by Oracle Retail are:

- **Username Token over HTTPS** - This security configuration is referred as Policy A in this document. This policy provides confidentiality due to the use of SSL, however it does not provide non-repudiation as nothing is signed. Wssp1.2-2007-Https-UsernameToken-Plain.xml
- **Message Protection** - This security configuration is referred as Policy B in this document. This policy encrypts the messages itself, so SSL is not used. The sender also signs the messages, which provides non-repudiation of the messages. However, this policy is more complex to implement.
  - Wssp1.2-2007-Wss1.1-UsernameToken-Plain-EncryptedKey-Basic128
  - Wssp1.2-2007-EncryptBody
  - Wssp1.2-2007-SignBody

**Note:**

1. The web services are secured using WebLogic policies (as opposed to OWSM policies).
2. If the application services are secured with any policy other than what is mentioned in this document or custom policies, the instructions in this document will not work.
3. The security setup in this document does not address authorization. Authorization must be taken care of by the individual application hosting the services.
4. Policy B is not supported over HTTPS. So ensure that non-SSL ports are enabled prior to applying Policy B.

## Advanced Infrastructure Security

Depending upon the security needs for your production environment, infrastructure where Oracle Retail applications are deployed can be secured.

Ensure the following to secure complete protection of environment:

- Securing the WebLogic Server Host
- Securing Network Connections
- Securing your Database
- Securing the WebLogic Security Service
- Securing Applications

For more information on Ensuring the Security of Your Production Environment, see Securing a Production Environment for Oracle WebLogic Server 12 C Release 1 (10.3.6) Guide.

# 3 Post Installation of Retail Infrastructure in Database

Oracle Retail applications use the Oracle database as the backend data store for applications. In order to ensure complete environment security the database should be secured.

This chapter describes the post installation steps for secured setup of Retail infrastructure in the Database.

The following topics are covered in this chapter:

- Configuring SSL Connections for Database Communications
- Configuring SSL on the Database Server
- Configuring SSL on an Oracle Database Client
- Configuring SSL on a Java Database Connectivity (JDBC) Thin Client
- Configuring the Password Stores for Database User Accounts
- Configuring the Database Password Policies
- Creating an Encrypted Tablespace in Oracle 12c Container Database
- Additional Information

## Configuring SSL Connections for Database Communications

Secure Sockets Layer (SSL) is the standard protocol for secure communications, providing mechanisms for data integrity and encryption. This can protect the messages sent and received by the database to applications or other clients, supporting secure authentication and messaging. Configuring SSL for databases requires configuration on both the server and clients, which include application servers.

This section covers the steps for securing Oracle Retail Application Clusters (RAC) database. Similar steps can be followed for single node installations also.

## Configuring SSL on the Database Server

The following steps are one way to configure SSL communications on the database server:

1. Obtain an identity (private key and digital certificate) and trust (certificates of trusted certificate authorities) for the database server from a Certificate Authority.

2. Create a folder containing the wallet for storing the certificate information. For Real Application Cluster (RAC) systems, this directory can be shared by all nodes in the cluster for easier maintenance.

   ```
   mkdir-p/oracle/secure_wallet
   ```

3. Create a wallet in the path. For example:

   ```
   orapki wallet create -wallet /oracle/secure_wallet -auto_login
   ```

4. Import each trust chain certificate into the wallet as shown in the following example:

   ```
   orapki wallet add -wallet /oracle/secure_wallet -trusted_cert -cert <trust
   chain certificate>
   ```

5. Import the user certificate into the wallet, as shown in the following example:

   ```
   orapki wallet add -wallet /oracle/secure_wallet -user_cert -cert <certificate
   file location>
   ```

6. Update the listener.ora by adding a TCPS protocol end-point first in the list of end points.

   ```
   LISTENER1=
     (DESCRIPTION=
       (ADDRESS=(PROTOCOL=tcps)(HOST=<dbserver>)(PORT=2484))
       (ADDRESS=(PROTOCOL=tcp)(HOST=<dbserver>)(PORT=1521)))
   ```

7. Update the listener.ora by adding the wallet location and disabling SSL authentication.

   ```
   WALLET_LOCATION = (SOURCE= (METHOD=File) (METHOD_DATA=
     (DIRECTORY=wallet_location)))
   SSL_CLIENT_AUTHENTICATION=FALSE
   ```

8. Update the sqlnet.ora with the same wallet location information and disabling SSL authentication.

   ```
   WALLET_LOCATION = (SOURCE= (METHOD=File) (METHOD_DATA=
     (DIRECTORY=wallet_location)))
   SSL_CLIENT_AUTHENTICATION=FALSE
   ```

9. Update the tnsnames.ora to configure a database alias using TCPS protocol for connections.

   ```
   <dbname>_secure=
     (DESCRIPTION= (ADDRESS_LIST=
       (ADDRESS=(PROTOCOL=TCPS)(HOST=<dbserver>)(PORT=2484)))
       (CONNECT_DATA=(SERVICE_NAME=<dbname>)))
   ```

10. Restart the database listener to pick up listener.ora changes.
11. Verify the connections are successful to the new <dbname>_secure alias.
12. At this point either the new secure alias can be used to connect to the database, or the regular alias can be modified to use TCPS protocol.
13. Export the identity certificate so that it can be imported on the client systems:

    ```
    orapki wallet export -wallet /oracle/secure_wallet -dn <full dn of identity
    certificate> -cert <filename_to_create>
    ```

## Configuring SSL on an Oracle Database Client

The following steps are one way to configure SSL communications on the database client:

1. Create a folder containing the wallet for storing the certificate information.

   ```
   mkdir-p /oracle/secure_wallet
   ```

2. Create a wallet in the path. For example:

   ```
   orapki wallet create -wallet /oracle/secure_wallet -auto_login
   ```

3. Import each trust chain certificate into the wallet as shown in the following example:

   ```
   orapki wallet add -wallet /oracle/secure_wallet -trusted_cert -cert <trust
   chain certificate
   ```

4. Import the identity certificate into the wallet, as shown in the following example:

   ```
   orapki wallet add -wallet /oracle/secure_wallet -trusted_cert -cert
   <certificate file location>
   ```

5. Update the sqlnet.ora with the wallet location information and disabling SSL authentication.

   ```
   WALLET_LOCATION
     = (SOURCE=
     (METHOD=File)
     (METHOD_DATA=
       (DIRECTORY=wallet_location)))
   SSL_CLIENT_AUTHENTICATION=FALSE
   ```

6. Update the tnsnames.ora to configure a database alias using TCPS protocol for connections.

   ```
   <dbname>_secure=
   (DESCRIPTION=
     (ADDRESS_LIST= (ADDRESS=(PROTOCOL=TCPS)(HOST=<dbserver>)(PORT=2484)))
     (CONNECT_DATA=(SERVICE_NAME=<dbname>)))
   ```

7. Verify the connections are successful to the new `<dbname>_secure` alias.

At this point either the new secure alias can be used to connect to the database, or the regular alias can be modified to use TCPS protocol.

## Configuring SSL on a Java Database Connectivity (JDBC) Thin Client

The following steps are one way to configure SSL communications for a Java Database Connectivity (JDBC) thin client:

1. Create a folder containing the keystore with the certificate information.

   ```
   mkdir-p /oracle/secure_jdbc
   ```

2. Create a keystore in the path. For example:

   ```
   keytool -genkey -alias jdbcwallet -keyalg RSA -keystore /oracle/secure_
   jdbc/truststore.jks -keysize 2048
   ```

3. Import the database certificate into the trust store as shown in the following example:

```
keytool -import -alias db_cert -keystore /oracle/secure_jdbc/truststore.jks
-file <db certificate file>
```

4. JDBC clients can use the following URL format for JDBC connections:

```
jdbc:oracle:thin:@(DESCRIPTION= (ADDRESS= (PROTOCOL=tcps) (HOST=<dbserver>)
(PORT=2484)) (CONNECT_DATA= (SERVICE_NAME=<dbname>)))
```

**Note:** The `<dbname>` would be replaced with service name in case of multitenant database (12c).

5. You need to set the properties as shown in Table 2-1, either as system properties or as JDBC connection properties.

***Table 3–1 Setting the Properties***

| Property | Value |
|---|---|
| javax.net.ssl.trustStore | Path and file name of truststore. For example: `/oracle/secure_jdbc/truststore.jks` |
| javax.net.ssl.trustStoreType | JKS |
| javax.net.ssl.trustStorePassword | Password for trust store |

## Configuring the Password Stores for Database User Accounts

Wallets can be used to protect sensitive information, including usernames and passwords for database connections. The Oracle Database client libraries have built-in support for retrieving credential information when connecting to databases. Oracle Retail applications utilize this functionality for non-interactive jobs such as batch programs, so that they are able to connect to the database without exposing user and password information to other users on the same system.

For information on configuring wallets for database access, see "Setting Up Password Stores with Oracle Wallet" in the product installation guide.

## Configuring the Database Password Policies

Oracle Database includes robust functionality to enforce policies related to passwords such as minimum length, complexity, when it expires, number of invalid attempts, and so on. Oracle Retail recommends these policies are used to strengthen passwords and lock out accounts after failed attempts.

For example, to modify the default user profile to lock accounts after five failed login attempts, run the following commands as a database administrator:

1. Query the current settings of the default profile.

```
select resource_name,limit,resource_type from dba_profiles where
profile='DEFAULT';
```

2. Alter the profile, if `failed_login_attempts` is set to unlimited:

```
alter profile default limit FAILED_LOGIN_ATTEMPTS 5;
```

**Note:** Many other profile settings are available for increased security.

For more information, see the *Oracle Database Security Guide*.

## Creating an Encrypted Tablespace in Oracle 12c Container Database

The retail tablespaces can be encrypted in container databases using the following method:

1. Update the `sqlnet.ora` file with the following encryption details:
    1. Configure the `sqlnet.ora` file for the software keystore location.

        ```
        ENCRYPTION_WALLET_LOCATION=
        (SOURCE=
        (METHOD=FILE)
          (METHOD_DATA=
          (DIRECTORY=path_to_keystore)))
        ```

    2. Restart the listener.
2. Set up the Tablespace Encryption in the Container Database.
    1. Create Software Keystores as follows:

        ```
        SQL> ADMINISTER KEY MANAGEMENT CREATE KEYSTORE '/u03/wallet_cdb' IDENTIFIED
        BY "val1ue#";
        ```

        Keystore altered.
    2. Create an Auto-Login Software Keystore as follows:

        ```
        SQL> ADMINISTER KEY MANAGEMENT CREATE AUTO_LOGIN KEYSTORE FROM KEYSTORE
        '/u03/wallet_cdb' identified by "val1ue#';
        ```

        Keystore altered.

        **Note:** The auto-login software keystore can be opened from different computers from the computer where this keystore resides. However, the [local] auto-login software keystore can only be opened from the computer on which it was created.

    3. Open the Software Keystore as follows:

        ```
        SQL> ADMINISTER KEY MANAGEMENT SET KEYSTORE OPEN IDENTIFIED BY "val1ue#"
        Container=ALL;
        ```

        Keystore altered.
    4. Set the Software TDE Master Encryption Key as follows:

        ```
        SQL> ADMINISTER KEY MANAGEMENT SET KEY IDENTIFIED BY "val1ue#" WITH BACKUP
        USING 'TDE_ENCRYPTION' Container=all;
        ```

        Keystore altered.

        **Note:** One can set the Encryption KEY only for particular PDB if required, by specifying the `CONTAINER=<PDB>`.

5. Create the ENCRYPTED TABLESPACE in PDB as follows:

```
SQL> conn sys/D0ccafe1@QOLRP01APP as sysdba
```

Connected.

```
SQL> create tablespace test datafile '+DATA1' size 100m ENCRYPTION DEFAULT
STORAGE (ENCRYPT);
```

Tablespace created.

6. Verify the Encryption:

```
SQL> select * from v$encryption_wallet
```

| WRL_TYPE | WRL_PARAMETER | STATUS | WALLET_TYPE | WALLET OR | FULLY BAC | CON ID |
|---|---|---|---|---|---|---|
| FILE | /u03/wallet_cdb | OPEN | PASSWORD | SINGLE | NO | 0 |

3. For more information on Configuring Transparent Data Encryption (TDE), see:

```
http://docs.oracle.com/database/121/ASOAG/asotrans_config.htm#ASOAG9529
```

4. Other information may be useful during maintenance activity.

   1. Close the Encryption Wallet as follows:

```
SQL> ADMINISTER KEY MANAGEMENT SET KEYSTORE Close IDENTIFIED BY "val1ue#"
Container=ALL;
```

## Additional Information

For more information on the subjects covered in this section, as well as information on other options that are available to strengthen database security, see the *Oracle Database Security Guide 12c Release 1*.

The Oracle Advanced Security Option provides industry-standards-based solutions to solve enterprise computing security problems, including data encryption and strong authentication. Some of the capabilities discussed in this guide require licensing the Advanced Security Option.

For more information, see the *Oracle Database Advanced Security Administrator's Guide 12c Release 1*.

# 4 Post Installation of Retail Infrastructure in WebLogic

This chapter describes the post installation steps for secured setup of Oracle Retail infrastructure in WebLogic.

The following topics are covered in this chapter:

- Retail Application Specific Post installation Steps for Security
- Asynchronous Task JMS Queue Security
- Hardening Use of Headers and Transport Layer Security

## Retail Application Specific Post installation Steps for Security

See the following sections for steps to improve security after an Oracle Retail Application has been installed.

### Batch Set Up for SSL Communication

Java batch programs communicate with Java applications deployed in WebLogic. For example, Oracle Retail Price Management (RPM) and Oracle Store Inventory Management (SIM). The communication needs to have SSL handshake with the deployed application. You need to import the SSL Certificates into the `JAVA_ HOME/jdk/jre/lib/security/cacerts keystore` for successful running of the application batches.

***Example 4–1 Importing Certificates into JDK Keystore***

```
/u00/webadmin/product/jdk/jre/lib/security> cp -rp cacerts cacerts_ORIG
/u00/webadmin/product/jdk/jre/lib/security> keytool -import -trustcacerts -alias
verisignclass3g3ca -file ~/ssl/Primary.pem -keystore cacerts
/u00/webadmin/product/jdk/jre/lib/security> keytool -import -trustcacerts -alias
oracleclass3g3ca -file ~/ssl/Secondary.pem -keystore cacerts
/u00/webadmin/product/jdk/jre/lib/security> keytool -import -trustcacerts -alias
hostname -file ~/ssl/cert.cer -keystore cacerts
```

**Note:** The default password for the JDK keystore is `changeit`.

## Asynchronous Task JMS Queue Security

This section describes the steps for adding security to the asynchronous task JMS queue. Securing the queue will allow only recognized users of the Retail Application to publish tasks to the JMS queue.

### Verifying and Creating Required Async Task Job Role and User

Securing the JMS async task queue requires a special enterprise role and a special user to exist in the retailer's Oracle Internet Directory (OID) instance.

The RETAIL_ASYNC_TASK_JOB is an enterprise role that will be used to group users who will have access to the asynchronous task queue.

The RETAIL _ASYNC_TASK_USER is a special user that Retail Applications can use as a principal for executing their message-driven, bean-based consumer processes. This user is a member of the RETAIL_ASYNC_TASK_JOB.

The RETAIL_ASYNC_TASK_JOB and RETAIL_ASYNC_TASK_USER are included as part of the Retail Default Security Reference Implementation installed as part of the Retail Application.

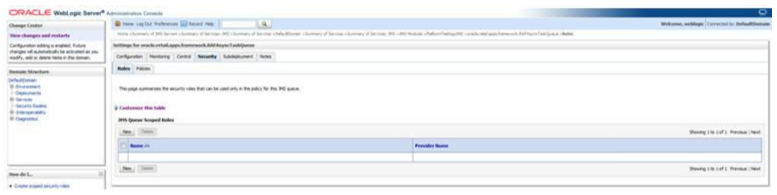
Verify the existence of the job and user in the OID instance. You need to create them if they do not exist.

### Securing the Asynchronous Task JMS Queue

Securing the queue can be done through the Weblogic Administration Console by adding a JMS Queue Scoped role.

1. Log into the WebLogic Administration Console.
2. Navigate to the JMS Module where the asynchronous task queue belongs to and go to the module's Security tab.

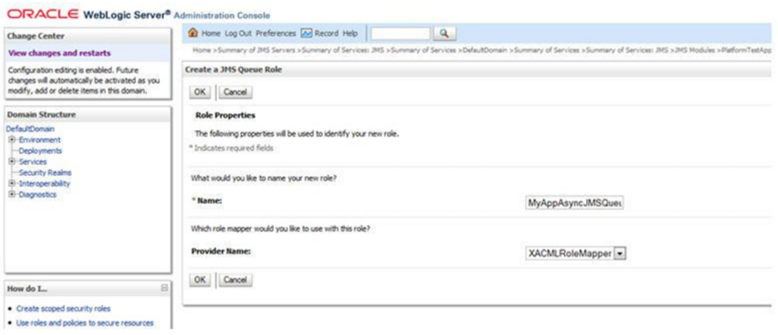
***Figure 4–1   Adding a New JMS Queue Scoped Role***



3. Specify a name for the JMS Queue Scoped Role. The suggested naming convention is [AppCode]AsyncJMSQueueAccessRole]. For example, AllocAsyncJMSQueueAccessRole. The JMS Queue Scoped Role will be created.Under the Roles section, add a new JMS Queue Scoped Roles.
4. Specify a name for the JMS Queue Scoped Role. The suggested naming convention is [AppCode]AsyncJMSQueueAccessRole]. For example, AllocAsyncJMSQueueAccessRole.

   The JMS Queue Scoped Role will be created.

***Figure 4–2 JMS Queue Scoped Role***


ORACLE WebLogic Server® Administration Console
Home Log Out Preferences Record Help
Change Center
View changes and restarts
Domain Structure
DefaultDomain
Environment
Deployments
Services
Security Realms
Interoperability
Diagnostics
How do I...
Create scoped security roles
Use roles and policies to secure resources
OK
Cancel

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

Home >Summary of JMS Servers >Summary of Services: JMS >Summary of Services >DefaultDomain >Summary of Services >Summary of Services: JMS >JMS Modules >PlatformTestApp

Create a JMS Queue Role

Role Properties

The following properties will be used to identify your new role.

* Indicates required fields

What would you like to name your new role?

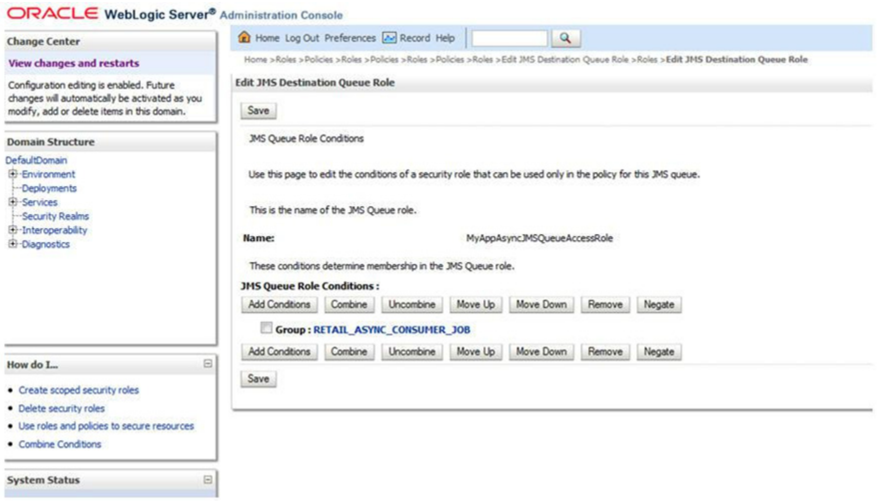
* Name: MyAppAsyncJMSQueu

Which role mapper would you like to use with this role?

Provider Name: XACMLRoleMapper

5. Navigate back to the JMS Module's Security tab.
6. Click the JMS Queue Scoped role that was created and add a Group condition for RETAIL_ASYNC_TASK_JOB.

***Figure 4–3 Adding a Group condition for RETAIL_ASYNC_TASK_JOB***



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

Home >Roles >Policies >Roles >Policies >Roles >Policies >Roles >Edit JMS Destination Queue Role >Roles >Edit JMS Destination Queue Role

Edit JMS Destination Queue Role

JMS Queue Role Conditions

Use this page to edit the conditions of a security role that can be used only in the policy for this JMS queue.

This is the name of the JMS Queue role.

Name: MyAppAsyncJMSQueueAccessRole

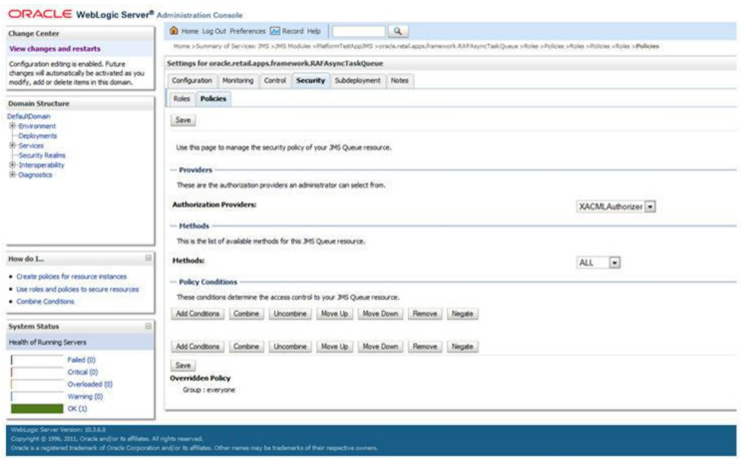
These conditions determine membership in the JMS Queue role.

JMS Queue Role Conditions :

Group : RETAIL_ASYNC_CONSUMER_JOB

7. Navigate back to the JMS Module's Security tab.
8. Go to the Policies section.

***Figure 4–4 Policies Tab***



Home >Summary of Services: JMS >JMS Modules >PlatformTestAppJMS >oracle.retail.apps.framework.RAFAsyncTaskQueue >Roles >Policies >Roles >Policies >Roles >Policies

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

Settings for oracle.retail.apps.framework.RAFAsyncTaskQueue

Use this page to manage the security policy of your JMS Queue resource.

Providers

These are the authorization providers an administrator can select from.

Authorization Providers: XACMLAuthorizer

Methods

This is the list of available methods for this JMS Queue resource.

Methods: ALL

Policy Conditions

These conditions determine the access control to your JMS Queue resource.

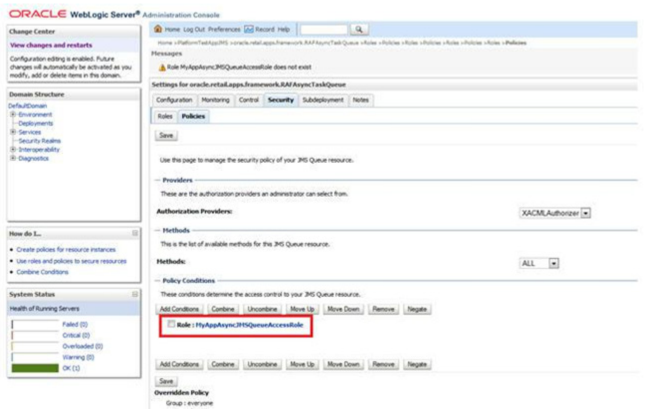
Overridden Policy

Group : everyone

WebLogic Server Version: 10.3.6.0
Copyright © 1996, 2011, 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. Add a new Role based condition specifying the JMS Queue Role created in the previous step.

***Figure 4–5 Adding a New Role***



Home >PlatformTestAppJMS >oracle.retail.apps.framework.RAFAsyncTaskQueue >Roles >Policies >Roles >Policies >Roles >Policies >Roles >Policies

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

Messages

Role MyAppAsyncJMSQueueAccessRole does not exist

Settings for oracle.retail.apps.framework.RAFAsyncTaskQueue

Use this page to manage the security policy of your JMS Queue resource.

Providers

These are the authorization providers an administrator can select from.

Authorization Providers: XACMLAuthorizer

Methods

This is the list of available methods for this JMS Queue resource.

Methods: ALL

Policy Conditions

These conditions determine the access control to your JMS Queue resource.

Role : MyAppAsyncJMSQueueAccessRole

Overridden Policy

Group : everyone

10. Save the changes. The queue is now secured.

11. Proceed to the next section to allow the Retail Web Application to publish tasks to the queue.

## Allowing Publishing to a Secured Asynchronous Task JMS Queue

Once the Asynchronous Task Queue has been secured with a JMS Queue Scoped Role as described in the previous section, further configuration is required to allow users of the Retail web application to publish tasks to the queue.

The JMS Queue Scoped Role was created to include an enterprise role, RETAIL_ASYNC_TASK_JOB. Any users belonging to this enterprise role will be given access to publish tasks to the queue.

Instead of assigning users directly to the RETAIL_ASYNC_TASK_JOB, applications should identify specific enterprise job roles in their system whose users will be allowed to perform asynchronous processing. Those job roles should be configured to extend from the RETAIL_ASYNC_TASK_JOB group.

See the Oracle Internet Directory documentation for details on how to extend one group to another.

## Hardening Use of Headers and Transport Layer Security

This section describes the steps for adding security headers in Web server layer and transport security settings in the Retail Applications `web.xml` file.

### Virtual Host Configuration

Navigate to Web server configuration file and add the following header configurations by restarting Web server:

```
# Header Settings to make sure no-store is active

<IfModule mod_headers.c>
   Header set Cache-Control "private, no-cache, no-store, proxy-revalidate,
no-transform"
   Header set Pragma "no-cache"
</IfModule>

# Guarantee HTTPS for 1 Year including Sub Domains
# Header always set Strict-Transport-Security "max-age=31536000;
includeSubDomains"
```

The X-Frame-Options header is appended automatically by the ADF framework. This header setting specifies whether content can be viewed through a third-party X-Frame. To protect against cross-site scripting, the ADF framework sets this value to SAMEORIGIN, which indicates that only websites within the same domain may frame content.

### Update weblogic.jdbc.remoteEnabled in setStartupEnv.sh

Ensure to update `weblogic.jdbc.remoteEnabled` in `setStartupEnv.sh` to false and restart he domain and node manager.

# 5 Troubleshooting

This chapter covers the common errors, issues, and troubleshooting them.

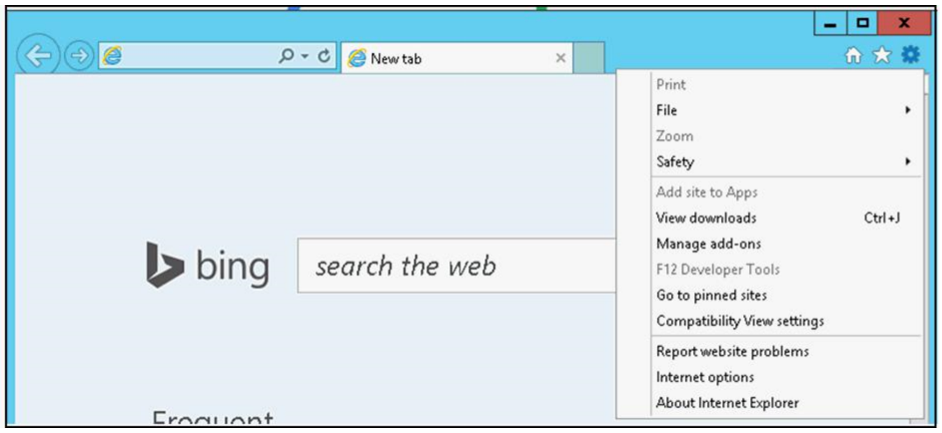
The following topics are covered in this chapter:

## Enabling TLS1.1 and 1.2 Protocols in Internet Explorer 11

To enable TLS 1.1 and 1.2 Internet explorer, do the following:

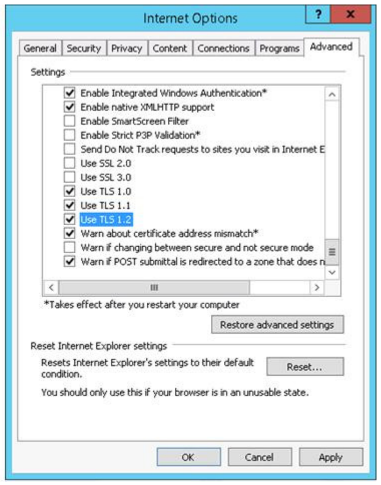
1. Click **Tools** (Alt+X), then **Internet Options**.

*Figure 5–1 Internet Explorer Tools Menu*



2. In **Advanced > Settings**, scroll down and select TLS 1.0, 1.1 and 1.2. Disable SSL 2.0 and SSL 3.0 as follows:

*Figure 5–2 Internet Explorer Internet Options*


Internet Options
General | Security | Privacy | Content | Connections | Programs | Advanced
Restore advanced settings
Reset...
OK
Cancel
Apply

Settings

- ✓ Enable Integrated Windows Authentication*
- ✓ Enable native XMLHTTP support
- ☐ Enable SmartScreen Filter
- ☐ Enable Strict P3P Validation*
- ☐ Send Do Not Track requests to sites you visit in Internet E
- ☐ Use SSL 2.0
- ☐ Use SSL 3.0
- ✓ Use TLS 1.0
- ✓ Use TLS 1.1
- ✓ Use TLS 1.2
- ✓ Warn about certificate address mismatch*
- ☐ Warn if changing between secure and not secure mode
- ✓ Warn if POST submittal is redirected to a zone that does n

*Takes effect after you restart your computer

Reset Internet Explorer settings

Resets Internet Explorer's settings to their default condition.

You should only use this if your browser is in an unusable state.

3. Click **Apply**.

## Java Version 7/8 SSL Handshake Issue while Using Self Signed Certificates

Java Version 7/8 may have issues using self-signed certificates. The self-signed root certificate may not be recognized by Java Version 1.7/1.8 and a certificate validation exception might be thrown during the SSL handshake. You need to create the private key with Subject Key Identifier to fix this problem. You need to include an option `- addext_ski` when the orapki utility is used to create the private key in the root wallet.

## Importing the Root Certificate in Local Client JRE

If customers are using certificates other than those provided by standard certificate authorities like custom CA implementation, then the JRE used for launching the applications from local machines like laptops or desktops might display a different error message.

The most probable cause of this issue could be unavailability of root certificates of the CA within the local JRE being used.

Perform the following steps to import the root certificates:

1. Back up cacert at `<JRE_HOME>/lib/security/cacert`.
2. Import the certificate using the keytool utility as shown in the following example:

```
-trustcacerts -file D:\ADMINISTRATION\SSL\apphost2\Selfsigned\apphost2.root.cer
-alias apphost2 -keystore "C:\Program Files\Java\jre7\lib\security\cacerts"

Enter keystore password: [default is changeit]
Owner: CN=apphost2, OU=<department>, O=<company>,L=<city>,ST=<state or
province>, C=<country>",
Issuer: CN=apphost2, OU=<department>, O=<company>,L=<city>,ST=<state or
province>, C=<country>"
Serial number: 515d4bfb
Valid from: Thu Apr 04 15:16:35 IST 2013 until: Fri Apr 04 15:16:35 IST 2014
Certificate fingerprints:
MD5:  AB:FA:18:2B:BC:FF:1B:67:E7:69:07:2B:DB:E4:C6:D9
SHA1: 2E:98:D4:4B:E0:E7:B6:73:55:4E:5A:BE:C1:9F:EA:9B:71:18:60:BB
SHA256: F3:54:FB:67:80:10:BA:9C:3F:AB:48:0B:27:83:58:BB:3D:22:C5:27:7D:
F4:D1:85:C4:4E:87:57:72:2B:6F:27
Signature algorithm name: SHA1withRSA
Version: 3
Trust this certificate? [no]: (yes) Certificate was added to keystore
C:\Program Files\Java\jre8\lib\security>
```

## Importing the Root Certificate to the Browser

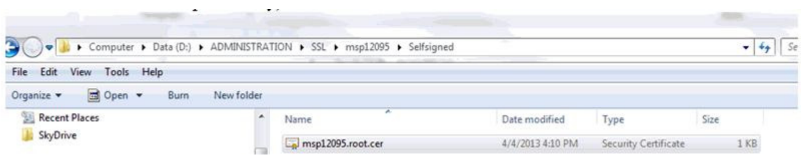
You need to add the signed Weblogic server certificate in the browser to avoid certificate verification error, if the Root Certificate is not in that list of trusted CAs.

### Importing the Root Certificate through Internet Explorer

Perform the following steps to import the Root Certificate through Internet Explorer:

1. Copy the Root Certificate file to the workstation.
2. Rename the file to `fa_root_cert.cer` (this is a quick and easy way to associate the file with the Windows certificate import utility).

***Figure 5–3 Import the Root Certificate File to the Workstation***


Computer ▸ Data (D:) ▸ ADMINISTRATION ▸ SSL ▸ msp12095 ▸ Selfsigned
File Edit View Tools Help
Organize Open Burn New folder
Recent Places
SkyDrive
Name
Date modified
Type
Size
msp12095.root.cer
4/4/2013 4:10 PM
Security Certificate
1 KB

3. Select the file.
4. Click **Install Certificate** and click **Next**.
5. Select **Place all certificates in the following store** and click **Browse**.

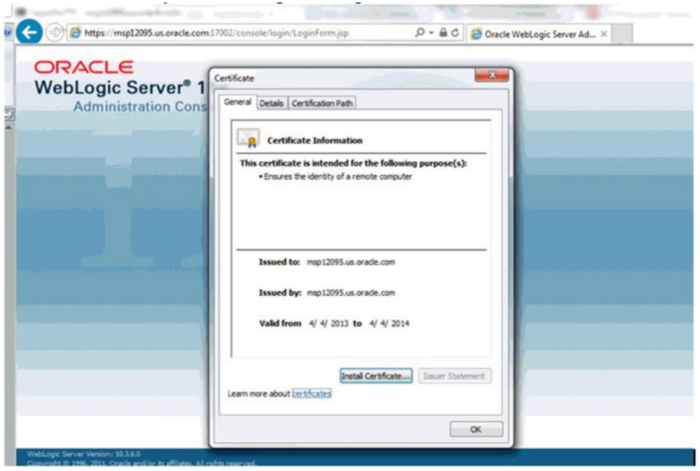
6. Select **Trusted Root Certification Authorities** and click **OK**.
7. Click **Next**.
8. Click **Finish** and then **Yes** at the Security Warning prompt.
9. Click **OK** to close the remaining open dialog boxes.

### Importing the Root Certificate through Mozilla Firefox

Perform the following steps to import the Root Certificate through Mozilla Firefox:

1. Start Mozilla Firefox.
2. Select **Tools > Options** from the main menu.
3. Click **Advanced > Encryption** tab > **View Certificates**.
4. In Certificate Manager, click the **Authorities** tab, then the **Import** button.
5. In the Downloading Certificate dialog, choose **Trust this CA to identify websites** and click **OK**.
6. Click **OK** in Certificate Manager.
7. Open a browser and test the URL using the SSL port.

***Figure 5–4 Importing the Root Certificate File through Mozilla Firefox***



Certificate Information

This certificate is intended for the following purpose(s):
- Ensures the identity of a remote computer

Issued to: msp12095.us.oracle.com

Issued by: msp12095.us.oracle.com

Valid from 4/ 4/ 2013 to 4/ 4/ 2014

Learn more about certificates

WebLogic Server Version: 10.3.6.0
Copyright © 1996, 2011, Oracle and/or its affiliates. All rights reserved.

## Set Up Secure Cookie

To obtain secure cookies, do the following:

1. Enable SSL in the environment.
2. Update the `weblogic.xml` file:
3. Redeploy the `<app>.ear` file.
4. Restart the services.

## Changes to Web Application Descriptor

The following are the changes to make to Web Application Descriptor:

1. Open the deployment descriptor of the Service Workspace, which has the jersey servlet configured.
2. Change transport-guarantee to CONFIDENTIAL.
3. Deploy the Application to secure the ReST Services as a one way SSL as follows:

```
<security-constraint>
   <web-resource-collection>
      <web-resource-name>Workflow Actions</web-resource-name>
      <url-pattern>/services/private/*</url-pattern>
      <http-method>GET</http-method>
      <http-method>POST</http-method>
   </web-resource-collection>
   <auth-constraint>
      ..
   </auth-constraint>
   <user-data-constraint>
      <transport-guarantee>CONFIDENTIAL</transport-guarantee>
   </user-data-constraint>
</security-constraint>
```

**Note:** An SSL connection needs to be used to ensure information being sent is not compromised, especially authentication credentials. If SSL is not used, the user credentials gets passed with BASE-64 encoding which does not encrypt the credentials and would be a hole in security.

## Disabling Hostname Verification

Hostname verification ensures that the hostname in the URL to which the client connects matches the hostname in the digital certificate that the server sends back as part of the SSL connection. However, in case the SSL handshake fails due to inability an to verify hostname, you can use the following this workaround:

**Note:** Disabling hostname verification is not recommended on production environments. This is only recommended for testing purposes. Hostname verification helps to prevent man-in-the-middle attacks.

Perform the following steps to disable the hostname verification for testing purposes:

1. Go to Environment > Domain > Servers > AdminServer.
2. Click the SSL tab.
3. Click Advanced.
4. On Hostname Verification, select NONE.
5. Save and activate changes.
6. On the Node Manager startup script, look for JAVA. Add the following line: Dweblogic.nodemanager.sslHostNameVerificationEnabled=false

   ```
   Dweblogic.nodemanager.sslHostNameVerificationEnabled=false
   ```

   After this change, the script should look as follows:

```
JAVA_OPTIONS="-Dweblogic.nodemanager.sslHostNameVerificationEnabled=false
${JAVA_OPTIONS}"
cd "${NODEMGR_HOME}" set -x
if [ "$LISTEN_PORT" != "" ] then
   if [ "$LISTEN_ADDRESS" != "" ]
```

7. Restart Node Manager.

## Verifying the Certificate Content

In situations where the certificate expires or belongs to a different host, the certificate becomes unusable. You can use the keytool utility to determine the details of the certificate. The certificates should be renewed or new certificates should be obtained from the appropriate certificate authorities, if the certificates expire.

Example:

```
apphost1:[10.3.6_apps] /u00/webadmin/ssl> keytool -printcert -file cert.cer
Certificate[1]:
Owner: CN=apphost1, OU=<department>, O=<company>,L=<city>,ST=<state or province>,
C=<country>"
Issuer: CN=Oracle SSL CA, OU=Class 3 MPKI Secure Server CA, OU=VeriSign Trust
Network, O=Oracle Corporation, C=US
Serial number: 0078dab9f1a5b56e2cd6g92a3987296
Valid from: Thu Oct 11 20:00:00 EDT 2012 until: Sat Oct 12 19:59:59 EDT 2013
Certificate fingerprints:
MD5:2B:71:89:11:01:40:43:FC:6F:D7:FB:24:EB:11:A5:1C
SHA1:
DA:EF:EC:1F:85:A9:DA:0E:E1:1B:50:A6:8B:A8:8A:BA:62:69:35:C1
SHA256:

C6:6F:6B:A7:C5:2C:9C:3C:40:E3:40:9A:67:18:B9:DC:8A:97:52:DB:FD:AB:4B:E5:B2:56:47:E

C:A7:16:DF:B6

Signature algorithm name: SHA1withRSA

Version: 3

Extensions:
```

## Verifying the Keystore Content

Keystores are repositories of certificates. If you face issues related to SSL Certificates, you need to check the certificates in the keystore. You need to import the certificates if they are missing. The keytool command provides the list of the certificates available.

Example:

```
keytool -v -list -keystore /u00/webadmin/product/jdk/jre/lib/security/cacerts
keytool -v -list -keystore /u00/webadmin/product/10.3.X_APPS/WLS/wlserver_
10.3/server/lib/apphost1.keystore
```

## Integration Issues

Oracle Retail applications can be deployed across different hosts and behind network firewalls. Ensure firewalls are configured to allow TCPS connections to enable secure communications among integrated application.

Secured applications using signed certificates need to use the same secured protocols for communication. Ensure that all the communicating applications use the same protocol.

For more information on steps to specify secured protocols, see the Enforcing Stronger Encryption in WebLogic section.

Communicating applications using signed certificates may need to verify the incoming connections. Root certificates should be available in the keystores of the applications to verify the requests from different hosts. It is important to import all the root certificates in the keystores of all communicating applications. For information on steps to import the root certificate in local client JRE, see the Importing the Root Certificate in Local Client JRE section.

## HTTPS Service Encountering Redirect Loop After Applying Policy A

The proxy server access enters into a redirect loop if the services are secured with policy A (username token over SSL) and the deployment is in a cluster. The access to such services does not work.

Perform the following workaround through SB Console for services that are secured with HTTPS:

1. Click **Resource Browser**.
2. Click **Proxy Services under Resource Browser**.
3. Click **Create under Change Center** to start a session.
4. For each of the SSL secured proxy services, perform the following steps:
    1. Click the proxy service you want to change.
    2. Click **Edit** next to **HTTP Transport Configuration**.
    3. Uncheck **HTTPS Required** check box.
    4. Click **Last**.
    5. Click **Save**.
5. Click **Activate** and then **Submit**.

# 6 Importing Topology Certificate

Implementation of SSL into the Oracle Retail deployment is driven by mapping the SSL certificates and wallets to various participating components in the topology.

## Importing Certificates into Middleware and Repository of Oracle Retail Applications

Table 6–1 and Table 6–2 describe the trust stores to be updated while confirming the certificates imported into middleware and repository of Oracle Retail applications. Ensure you have updated the given trust stores with the signed (either self signed or issued by certifying authority) certificates

**Note:** In Table 6–1 and Table 6–2, the *root.cer are the public key certificates and the *server.cer are the private key certificates.

***Table 6–1 Importing Topology Certificate: App Hosts***

| | | Java app-host | | Forms app-host | | RIB app-host | |
|---|---|---|---|---|---|---|---|
| **Component** | **Certificates** | **Java app-Managed server** | **Java app-JAVA cacerts** | **Forms app-Managed server** | **Forms app-JAVA cacerts** | **RIB app-Managed server** | **RIB app-JAVA cacerts** |
| Java.app | appserver.cer | Yes | No | No | No | No | No |
| Java.app | approot.cer | Yes | Yes | No | No | No | Yes |
| Forms.app | fmserver.cer | No | No | Yes | No | No | No |
| Forms.app | fmroot.cer | No | No | No | Yes | No | No |
| RIB.app | ribserver.cer | No | No | No | No | Yes | No |
| RIB.app | ribroot.cer | No | Yes | No | No | Yes | Yes |
| BI Publisher | biserver.cer | No | No | No | No | No | No |
| BI Publisher | biroot.cer | No | Yes | No | Yes | No | No |
| OID | oidcer.cer | No | No | No | No | No | No |
| OID | oidroot.cer | No | Yes | No | No | No | No |

***Table 6–2 Importing Topology Certificate: Other Hosts***

| | | BIPublisher-host | | OID-host | Client-host | |
|---|---|---|---|---|---|---|
| **Component** | **Certificates** | **BIPublisher - Managed server** | **BIPublisher - JAVA cacerts** | **Wallet** | **Browser** | **Client- JAVA cacerts** |
| Java.app | appserver.cer | No | No | No | No | No |
| Java.app | approot.cer | | Yes | Yes | Yes | Yes |
| Forms.app | fmserver.cer | No | No | No | No | No |
| Forms.app | fmroot.cer | | Yes | Yes | Yes | Yes |
| RIB.app | ribserver.cer | No | No | No | No | No |
| RIB.app | ribroot.cer | No | No | No | Yes | Yes |
| BI Publisher | biserver.cer | Yes | No | No | No | No |
| BI Publisher | biroot.cer | Yes | Yes | No | Yes | Yes |
| OID | oidcer.cer | No | No | Yes | No | No |
| OID | oidroot.cer | No | Yes | Yes | Yes | Yes |

# 7 Using Self-Signed Certificates

Self-signed certificates can be used in development environments for securing applications. The generic steps to be followed for creating self-signed certificates and configuring for use for Oracle Retail application deployment are covered in the subsequent sections.

The following topics are covered in this chapter:

-

## Creating a Keystore through the Keytool in Fusion Middleware (FMW) 11g

Perform the following steps to create a keystore through the keytool in Fusion Middleware (FMW) 11g:

1. Create a directory for storing the keystores.

   ```
   mkdir ssl
   ```

2. Run the following to set the environment:

   ```
   cd $MIDDLEWARE_HOME/user_projects/domains/<domain>/bin
   . ./setDomainEnv.sh
   ```

   For example:

   ```
   ../setDomainEnv.sh
   ```

3. Create a keystore and private key, by executing the following command:

   ```
   keytool -genkey -alias <alias> -keyalg RSA -keysize 2048 -dname <dn> -keypass
   <password> -keystore <keystore> -storepass <password> -validity 365
   ```

   Example:

   ```
   keytool -genkey -alias apphost2 -keyalg RSA -keysize 2048 -dname "CN=<Server
   Name>,OU=<Organization Unit>, O=<Organization>,L=<City>,ST=<State>,C=<Country>"
   -keypass <kpass> -keystore /u00/webadmin/ssl/apphost2.keystore -storepass
   <spass> -validity 365
   ```

## Exporting the Certificate from the Identity Keystore into a File

Perform the following steps to export the certificate from the identity keystore into a file (for example, `pubkey.cer`):

1. Run the following command:

```
keytool -export -alias selfsignedcert -file pubkey.cer -keystore identity.jks
-storepass <password>
```

Example:

```
keytool -export -alias apphost2 -file /u00/webadmin/ssl/pubkey.cer -keystore
/u00/webadmin/ssl/apphost2.keystore -storepass <spass>
```

## Importing the Certificate Exported into trust.keystore

Perform the following steps to import the certificate you exported into `trust.keystore`:

1. Run the following command:

   ```
   keytool -import -alias selfsignedcert -trustcacerts -file pubkey.cer -keystore
   trust.keystore -storepass <password>
   ```

   Example:

   ```
   keytool -import -alias apphost2 -trustcacerts -file pubkey.cer -keystore
   trust.keystore -storepass <spass> Owner: CN=apphost2, OU=<Organization Unit>,
   O=<company>,L=<city>,ST=<state or province>, C=<country>
   ```

2. Enter `yes` when prompted whether to trust the certificate.

   Example:

   ```
   Trust this certificate? [no]: yes
   ```

## Configuring WebLogic

You need to enable SSL for WebLogic server's Admin and managed servers by following the steps as provided in the Configuring the Application Server for SSL section.

## Configuring Nodemanager

You need to secure the Node manager by following the steps in Securing Nodemanager with SSL Certificates section.

## Importing Self Signed Root Certificate into Java Virtual Machine (JVM) Trust Store

In order for the Java Virtual Machine (JVM) to trust in your newly created certificate, import your custom certificates into your JVM trust store.

Perform the following steps to import the root certificate into JVM Trust Store:

1. Ensure that `JAVA_HOME` has been configured.
2. Run the following command:

   ```
   $keytool -import -trustcacerts -file rootCer.cer -alias selfsignedcert
   -keystore cacerts
   ```

   Example:

   ```
   keytool -import -trustcacerts -file /u00/webadmin/ssl/root.cer -alias apphost2
   ```

```
-keystore /u00/webadmin/product/jdk1.6.0_30.64bit/jre/lib/security/cacerts
-storepass [spass default is changeit]
```

3. Enter `yes` when prompted whether to trust the certificate.

   Example:

   ```
   Trust this certificate? [no]: yes
   ```

## Disabling Hostname Verification

This section has been covered under Disabling Hostname Verification section.

## Converting PKCS7 Certificate to x.509 Certificate

Certificate authorities provide signed certificates of different formats. However, not all formats of certificates can be imported to Java based keystores. Hence the certificates need to be converted to usable form. Java based Keystores supports x.509 format of certificate.

The following example demonstrates converting certificate PKCS 7 to x.509 format:

1. Copy the PKCS 7 certificate file to a Windows desktop.
2. Rename the file and provide a `.p7b` extension.
3. Open the `.p7b` file.
4. Click the plus (+) symbol.
5. Click the **Certificates** directory.

   An Intermediary certificate if provided by CA for trust.

   > **Note:** If an Extended Validation certificate is being converted you should see three files: the End Entity certificate and the two EV intermediate CA's.

6. Right click on your certificate file.
7. Select **All Tasks > Export**.
8. Click **Next**.
9. Select **Base-64 encoded X.509 (.cer)**.
10. Click **Next**.
11. Browse to a location to store the file.
12. Enter a File name.

    For example, `MyCert`. The `.cer` extension is added automatically.
13. Click **Save**.
14. Click **Next**
15. Click **Save**.

The certificate can be now imported into Java-based keystores.

Example:

```
keytool -import -trustcacerts -alias apphost1 -file
/u00/webadmin/ssl/cert-x509.cer -keystore
/u00/webadmin/product/jdk/jre/lib/security/cacerts Enter keystore password:
[default is changeit]
```

# 8 Functional Security for Applications Using Fusion Middleware

The chapter provides guidance for administrators to understand, configure, and customize functional security for Application Development Framework (ADF) applications or applications using Fusion Middleware platform security infrastructure like OBIEE.

The following topics are covered in this chapter:

## Understanding the Security Model

The Oracle Retail Fusion security model is built upon the Oracle Fusion Middleware platform, which incorporates the Java security model. The Java model is a role-based, declarative model that employs container-managed security where resources are protected by roles that are assigned to users. However, extensive knowledge of the Java-based architecture is unnecessary when using the Oracle Fusion Middleware Security model.

### Key Security Elements

The Oracle Fusion Middleware security model depends upon the following key elements in order to provide uniform security and identity management across the enterprise:

#### Application Policy

Application permissions are granted to members of its application roles. In the default security configuration, each application role conveys a predefined set of permissions. Permission grants are defined and managed in an application policy. After an application role is associated with an application policy, that role becomes a Grantee of the policy. An application policy is specific to a particular application.

### Application Role

After permission grants are defined in an application policy, an application role can be mapped to that policy, and the application role then becomes the mechanism to convey the permissions. In this manner, an application role becomes the container that grants permissions to its members. The permissions become associated with the application role through the relationship between policy and role. After groups are mapped to an application role, the corresponding permissions are granted to all members equally. Membership is defined in the application role definition. Application roles are assigned in accordance with specific conditions and are granted dynamically based on the conditions present at the time authentication occurs. More than one group can be members of the same application role.

### Authentication Provider

An authentication provider is used to access user and group information and is responsible for authenticating users. An Oracle WebLogic Server authentication provider enables you to manage users and groups in one place.

An identity store contains user name, password, and group membership information. An authentication provider accesses the data in the identity store and authenticates against it. For example, when a user name and password combination is entered at login, the authentication provider searches the identity store to verify the credentials provided. The Oracle Retail Fusion application's default authentication provider authenticates against Oracle Internet Directory (OID).

### Users and Groups

A user is an entity that can be authenticated. A user can be a person, such as an application user, or a software entity, such as a client application. Every user is given a unique identifier.

Groups are organized collections of users that have something in common. Users should be organized into groups with similar access needs in order to facilitate efficient security management.

### Security Realm

An Oracle Retail Fusion application is installed on an Oracle WebLogic Server domain, which is created during installation. The Oracle Retail Fusion application security is managed within the security realm for this Oracle WebLogic Server domain. A security realm acts as a scoping mechanism. Each security realm consists of a set of configured security providers, users, groups, security roles, and security policies. Only one security realm can be active for the domain. The Oracle Retail Fusion application's authentication is performed by the authentication provider configured for the default security realm for the WebLogic Server domain in which it is installed. Oracle WebLogic Server Administration Console is the administration tool used for managing an Oracle WebLogic Server domain.

## Permission Grants and Inheritance

Each Oracle Retail Fusion application provides application-specific permissions for accessing different features. Application permissions are typically granted by becoming a member in an application role. Permissions can be granted in two ways - through membership in an application role (direct) and through group and role hierarchies (inheritance). Application role membership can be inherited by nature of the application role hierarchy. In the default security configuration, each application role is pre-configured to grant a predefined set of permissions. Groups are mapped to an application role. The mapping of a group to a role conveys the role's permissions to all members of the group. In short, permissions are granted in Oracle Retail Fusion applications by establishing the following relationships:

- A group defines a set of users having similar system access requirements. Users are added as members to one or more groups according to the level of access required.
- Application roles are defined to represent the role a user typically performs when using the Oracle Retail Fusion application.
- The groups of users are mapped to one or more application roles that match the type of access required by the population.
- An application role is mapped to the application policy that grants the set of permissions required by the role type (for example, an administrator, an author, a consumer).
- Group membership can be inherited by nature of the group hierarchy. Application roles mapped to inherited groups are also inherited, and those permissions are likewise conveyed to the members.

User permissions are determined by the system as follows:

1. A user enters credentials into a Web browser at login. The user credentials are authenticated by the authentication provider against data contained in the identity store.
2. After successful authentication, a Java subject and principal combination is issued, which is populated with the user name and a user's groups.
3. A list of the user's groups is generated and checked against the application roles. A list is created of the application roles that are mapped to each of the user's groups.
4. A user's permission grants are determined from knowing which application roles the user is a member of.

A user can also be granted permissions if they inherit other application roles. Members of application roles can include other groups and application roles. The result is a hierarchical role structure where permissions can be inherited in addition to being explicitly granted. This hierarchy provides that a group is granted the permissions of the application role for which it is a member, and the permissions granted by all roles descended from that role.

For example, the default security configuration includes several predefined groups and application roles. The default BIAdministrator application role includes the BIAdministrators group, the BIAuthor application role includes the BIAuthors group, and the BIConsumer application role includes the BIConsumers group. The default BIAdministrator application role is a member the BIAuthor application role, and the BIAuthor application role is a member of the BIConsumer application role.

The members of these application roles inherit permissions as follows. Members of the BIAdministrators group are granted all the permissions of the BIAdministrator role, the BIAuthor role, and the BIConsumer role. By the nature of this role hierarchy, the user who is a member of a particular group is granted permissions both explicitly and through inheritance.

**Note:** By themselves, groups and group hierarchies do not enable any privilege to access resources controlled by an application. Privileges are conveyed by the permission grants defined in an application policy. A group or application role becomes a Grantee of the application policy. The application policy Grantee conveys the permissions. This is done by becoming a member of the Grantee (application role).

Figure 8–1 shows these relationships between the default groups and application roles.

***Figure 8–1 Relationships between the Default Groups and Application Roles***

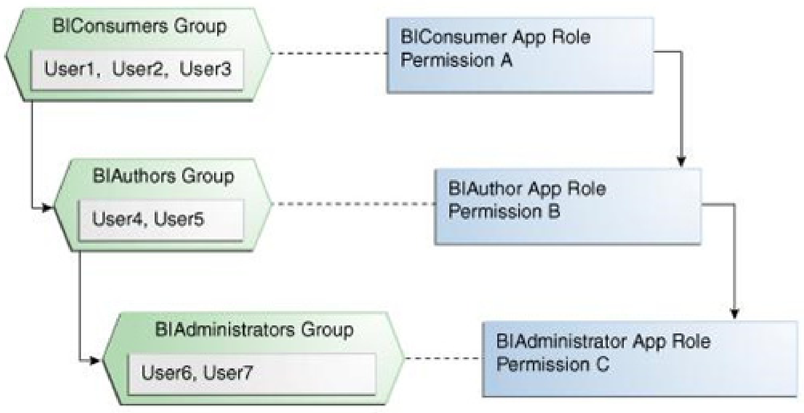


Table 8–1 summarizes how permissions are granted explicitly or are inherited in the previous example and in Figure 8–1.

***Table 8–1 Permissions Granted by the Role Hierarchy Example***

| User Name | Group Membership: Explicit/Inherited | Application Role Membership: Explicit/Inherited | Permission Grants: Explicit/Inherited |
|---|---|---|---|
| User1, User2, User3 | BIConsumers: Explicit | BIConsumer: Explicit | Permission A: Explicit |
| User4, User5 | BIAuthors: Explicit | BIAuthor: Explicit | Permission B: Explicit |
| | BIConsumers: Inherited | BIConsumer: Inherited | Permission A: Inherited |
| User6, User7 | BIAdministrators: Explicit | BIAdministrator: Explicit | Permission C: Explicit |
| | BIAuthors: Inherited | BIAuthor: Inherited | Permission B: Inherited |
| | BIConsumers: Inherited | BIConsumer: Inherited | Permission A: Inherited |

## Managing Authorization

After a user is authenticated, further access to application resources is controlled by the granting of permissions, also known as authorization. The policy store contains the system and application-specific policies and roles required for an application. A policy store can be file-based, LDAP-based, or on an Oracle database. The policy store holds the mapping definitions between the default Oracle Retail Fusion Application's application roles, permissions and groups. Oracle Retail Fusion Application's permissions are granted by mapping groups from the identity store to application roles and permission grants located in the policy store. These mapping definitions between groups (identity store) and the application roles (policy store) are also kept in the policy store.

> **Note:** The best practice is to map groups instead of individual users to application roles. Controlling membership in a group reduces the complexity of tracking access rights for multiple individual users. Group membership is controlled in the identity store.

The `system-jazn-data.xml` file is installed and configured as the default policy store. You can continue to use the default store and modify it as needed for your environment, or you can migrate its data to an Oracle database.

The policy store and credential store must be of the same type in your environment. That is, both must be either file-based, LDAP-based, or on an Oracle database.

Permissions must be defined in a manner that the Oracle Retail Fusion Application understands. All valid Oracle Retail Fusion Application permissions are premapped to application policies, which are in turn premapped to the default application roles.

You cannot create new permissions in the policy store. However, you can customize the default application policy permission grants and application role mappings, as well as create your own.

## Accessing Oracle Enterprise Manager Fusion Middleware Control

Launch Fusion Middleware Control by entering its URL into a Web browser. The URL includes the name of the host and the administration port number assigned during the installation. This URL takes the following form:

```
http://hostname:port_number/em
```

The default port is 7001. For more information about using Fusion Middleware Control, see *Oracle Fusion Middleware Administrator's Guide*.

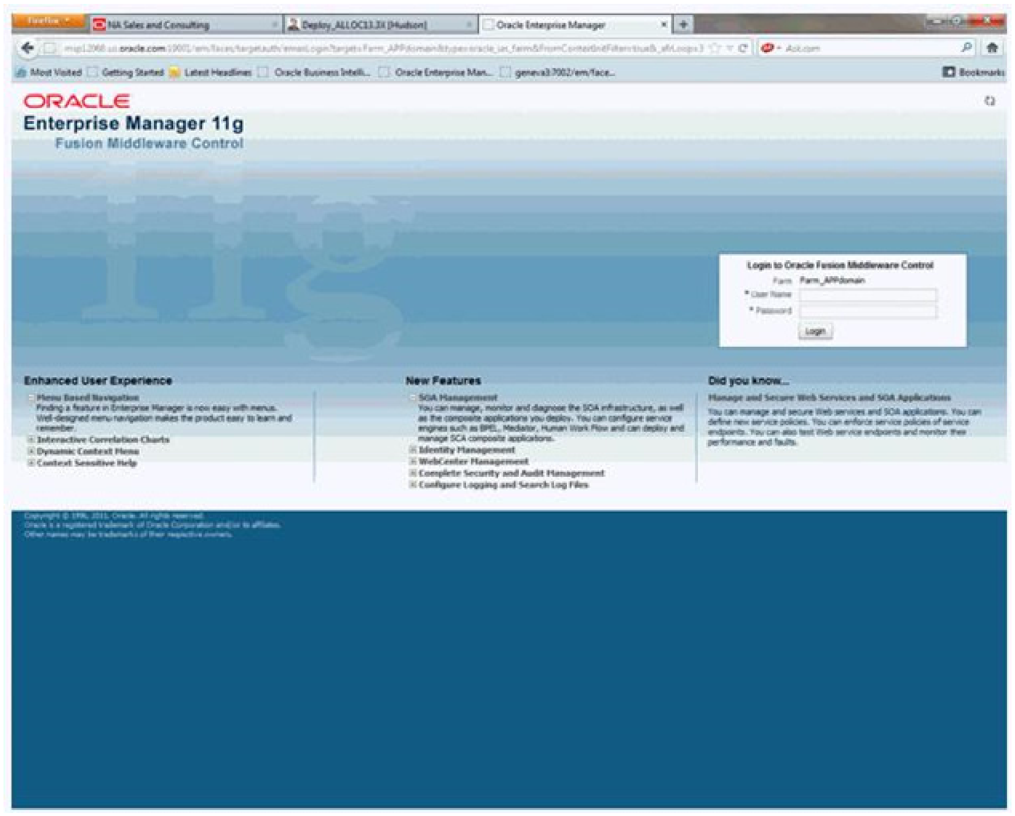
### To display the Security menu in Fusion Middleware Control

1. Log on to Oracle Enterprise Manager Fusion Middleware Control by entering the URL in a Web browser.

   For example, `http://hostname:7001/em`

   The Fusion Middleware Control login page opens.

*Figure 8–2 Fusion Middlware Control Login Page*



ORACLE
Enterprise Manager 11g
Fusion Middleware Control

Login to Oracle Fusion Middleware Control
Farm Farm_APPdomain
* User Name
* Password
Login

Enhanced User Experience
Menu Based Navigation
Finding a feature in Enterprise Manager is now easy with menus. Well-designed menu navigation makes the product easy to learn and remember.
Interactive Correlation Charts
Dynamic Context Menu
Context Sensitive Help

New Features
SOA Management
You can manage, monitor and diagnose the SOA infrastructure, as well as the composite applications you deploy. You can configure service engines such as BPEL, Mediator, Human Work Flow and can deploy and manage SCA composite applications.
Identity Management
WebCenter Management
Complete Security and Audit Management
Configure Logging and Search Log Files

Did you know...
Manage and Secure Web Services and SOA Applications
You can manage and secure Web services and SOA applications. You can define new service policies. You can enforce service policies of service endpoints. You can also test Web service endpoints and monitor their performance and faults.

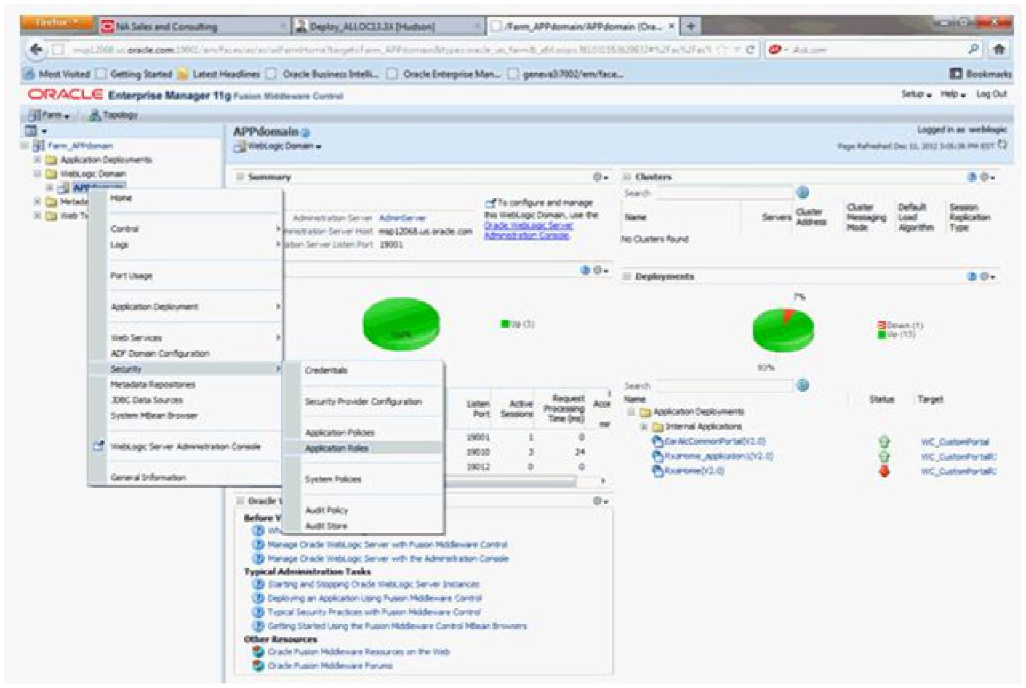
Copyright © 1996, 2011, Oracle. All rights reserved.
Oracle is a registered trademark of Oracle Corporation and/or its affiliates.
Other names may be trademarks of their respective owners.

2. Enter the Oracle Retail Fusion Application's administrative user name and password and click **Login**.

   The password is the one you supplied during the installation of Oracle Retail Fusion Application. If these values have been changed, then use the current administrative user name and password combination.

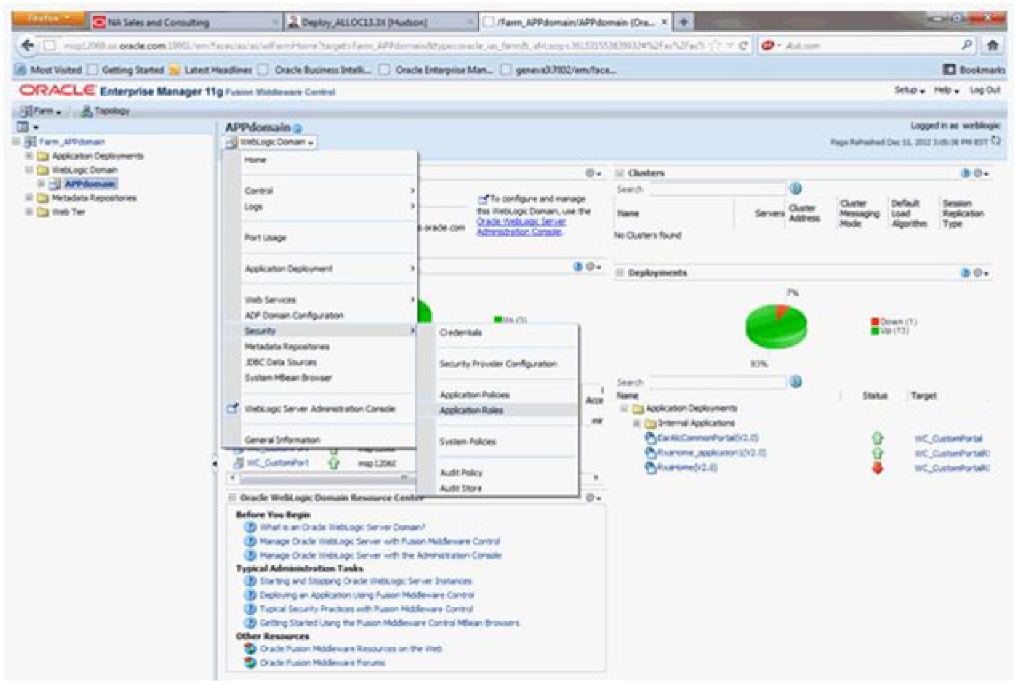
3. From the target navigation pane, click **WebLogic Domain** to display **APPdomain**.

***Figure 8–3 Enterprise Manager AppDomain Security Submenu***



4. Display the Security menu by selecting one of the following methods:
   - Right click **APPdomain** to display the Security menu.
   - Select **Security** to display a submenu.
   - From the content pane, go to **WebLogic Domain** and select **Security**.
   - Select **Security** to display a submenu.

*Figure 8–4 Enterprise Manager WebLogic Domain Security Submenu*



## Managing the Policy Store Using Fusion Middleware Control

Use Fusion Middleware Control to manage the Oracle Retail Fusion Application's application policies and application roles maintained in the policy store whether it is filed-based, LDAP-based, or on an Oracle database.

**Caution:** Oracle recommends you make a copy of the original `system-jazn-data.xml` policy file and place it in a safe location. Use the copy of the original file to restore the default policy store configuration, if needed. Changes to the default security configuration may lead to an unwanted state. The default installation location is `MW_HOME/user_projects/domain/your_ domain/config/fmwconfig`.

The following are common policy store management tasks:

- Modifying the membership of an application role. For more information, see the Modifying Application Roles Using Fusion Middleware Control section.
- Creating a new application role from the beginning. For more information, see the To Create a New Application Role section.
- Creating a new application role based on an existing application role. For more information, see the To Create an Application Role Based on an Existing One section.

## Modifying Application Roles Using Fusion Middleware Control

Members can be added or deleted from an application role using Fusion Middleware Control. You must perform these tasks while in the WebLogic Domain that Oracle Retail Fusion Application is installed in.

**Caution:** You need to be careful when changing the permission grants and membership for the default application roles. Changes could result in an unusable system.

Valid members of an application role are groups, or other application roles. The process of becoming a member of an application role is called mapping. That is, being mapped to an application role is to become a member of an application role. The best practice is to map groups instead of individual users to application roles for easier maintenance.

### To Add or Remove Members from an Application Role

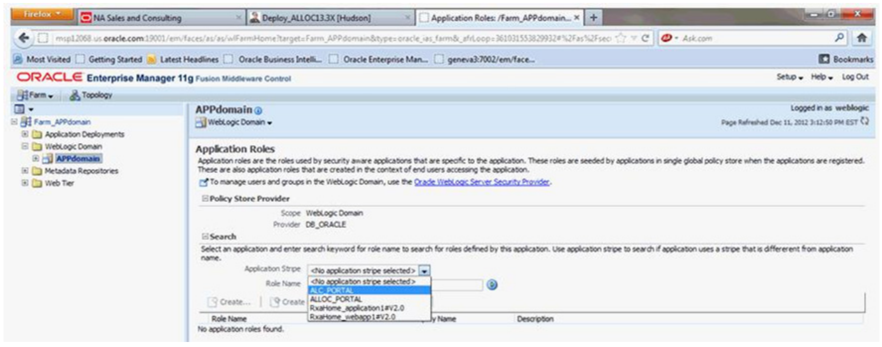
1. Log in to Fusion Middleware Control, navigate to **Security**, then select **Application Roles** to display the **Application Roles** page.

   For information on navigating to the Security menu, see the Accessing Oracle Enterprise Manager Fusion Middleware Control section.

2. Choose **Select Application Stripe to Search**, then select the policy stripe name (for example, ALC_CORE) from the list.

3. Click the search icon next to **Role Name**.

***Figure 8–5 Retail Fusion Application's Application Roles Window***



Application Roles

Application roles are the roles used by security aware applications that are specific to the application. These roles are seeded by applications in single global policy store when the applications are registered. These are also application roles that are created in the context of end users accessing the application.

To manage users and groups in the WebLogic Domain, use the Oracle WebLogic Server Security Provider.

Policy Store Provider

Scope WebLogic Domain
Provider DB_ORACLE

Search

Select an application and enter search keyword for role name to search for roles defined by this application. Use application stripe to search if application uses a stripe that is differerent from application name.

Application Stripe <No application stripe selected>
Role Name

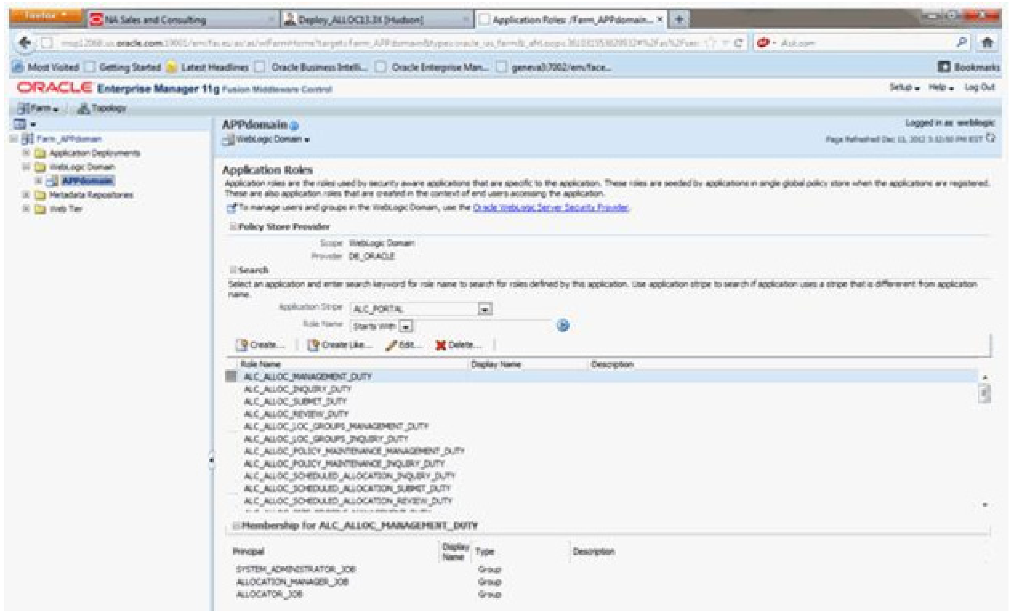
<No application stripe selected>
ALC_PORTAL
ALLOC_PORTAL
RxaHome_application1#V2.0
RxaHome_webapp1#V2.0

No application roles found.

The Oracle Retail Fusion Application's application roles are displayed.

Figure 8–6 displays the default application roles.

*Figure 8–6 Default Application Roles Window*



APPdomain
WebLogic Domain

Application Roles

Application roles are the roles used by security aware applications that are specific to the application. These roles are seeded by applications in single global policy store when the applications are registered. These are also application roles that are created in the context of end users accessing the application.

To manage users and groups in the WebLogic Domain, use the Oracle WebLogic Server Security Provider.

Policy Store Provider

Scope WebLogic Domain
Provider DB_ORACLE

Search

Select an application and enter search keyword for role name to search for roles defined by this application. Use application stripe to search if application uses a stripe that is different from application name.

Application Stripe ALC_PORTAL
Role Name Starts With

| Role Name | Display Name | Description |
|---|---|---|
| ALC_ALLOC_MANAGEMENT_DUTY | | |
| ALC_ALLOC_INQUIRY_DUTY | | |
| ALC_ALLOC_SUBMIT_DUTY | | |
| ALC_ALLOC_REVIEW_DUTY | | |
| ALC_ALLOC_LOC_GROUPS_MANAGEMENT_DUTY | | |
| ALC_ALLOC_LOC_GROUPS_INQUIRY_DUTY | | |
| ALC_ALLOC_POLICY_MAINTENANCE_MANAGEMENT_DUTY | | |
| ALC_ALLOC_POLICY_MAINTENANCE_INQUIRY_DUTY | | |
| ALC_ALLOC_SCHEDULED_ALLOCATION_INQUIRY_DUTY | | |
| ALC_ALLOC_SCHEDULED_ALLOCATION_SUBMIT_DUTY | | |
| ALC_ALLOC_SCHEDULED_ALLOCATION_REVIEW_DUTY | | |

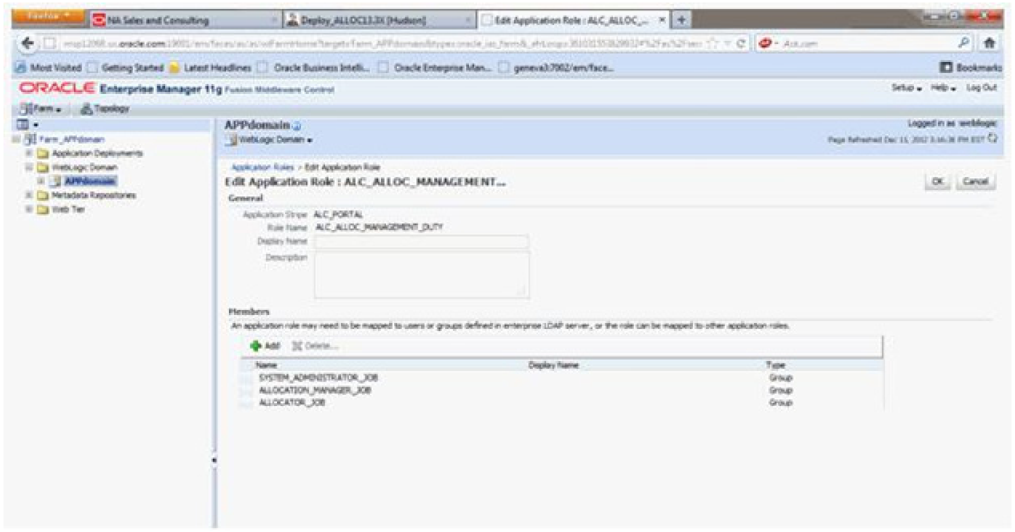
Membership for ALC_ALLOC_MANAGEMENT_DUTY

| Principal | Display Name | Type | Description |
|---|---|---|---|
| SYSTEM_ADMINISTRATOR_JOB | | Group | |
| ALLOCATION_MANAGER_JOB | | Group | |
| ALLOCATOR_JOB | | Group | |

4. Select the cell next to the application role name and click Edit to display the Edit Application Role page.

   Figure 8–7 shows the ALC_ALLOC_MANAGEMENT_DUTY role is selected.

*Figure 8–7 Edit Application Role Window*



APPdomain
WebLogic Domain

Application Roles > Edit Application Role

Edit Application Role : ALC_ALLOC_MANAGEMENT...

General

Application Stripe ALC_PORTAL
Role Name ALC_ALLOC_MANAGEMENT_DUTY
Display Name
Description

Members

An application role may need to be mapped to users or groups defined in enterprise LDAP server, or the role can be mapped to other application roles.

| Name | Display Name | Type |
|---|---|---|
| SYSTEM_ADMINISTRATOR_JOB | | Group |
| ALLOCATION_MANAGER_JOB | | Group |
| ALLOCATOR_JOB | | Group |

---

**Note:** You can add or delete members from the **Edit Application Role** page. The valid members are application roles, and groups.

---

5. Select from the following options:

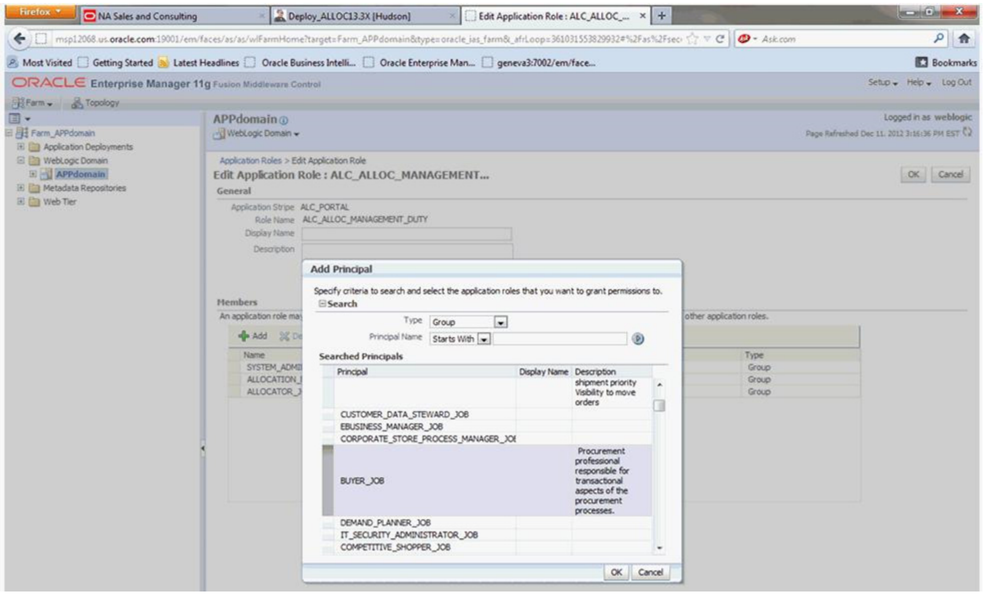
   - **To delete a member:** From **Members**, select from **Name** the member to delete. Then click **Delete**.

- **To add a member:** Click the **Add** button that corresponds to the member to add. Select **Add Application Role**, **Add Group**, and **Add User**.

6. For adding a member, complete Search and select from the available list and click OK.

   Figure 8-8 shows the Add Group dialog after the BUYER_JOB group was selected.

***Figure 8–8 Add Group Dialog Window***


Firefox
NA Sales and Consulting
Deploy_ALLOC13.3X [Hudson]
Edit Application Role : ALC_ALLOC_...
msp12068.us.oracle.com:19001/em/faces/as/as/wlFarmHome?target=Farm_APPdomain&type=oracle_ias_farm&_afrLoop=361031553829932#%2Fas%2Fsec
Ask.com
Most Visited
Getting Started
Latest Headlines
Oracle Business Intelli...
Oracle Enterprise Man...
geneva3:7002/em/face...
Bookmarks
ORACLE Enterprise Manager 11g Fusion Middleware Control
Setup
Help
Log Out
Farm
Topology
Farm_APPdomain
Application Deployments
WebLogic Domain
APPdomain
Metadata Repositories
Web Tier
APPdomain
WebLogic Domain
Logged in as weblogic
Page Refreshed Dec 11, 2012 3:16:36 PM EST
Application Roles > Edit Application Role
OK
Cancel
Members
Add
Search
OK
Cancel

Edit Application Role : ALC_ALLOC_MANAGEMENT...

General

Application Stripe ALC_PORTAL
Role Name ALC_ALLOC_MANAGEMENT_DUTY
Display Name
Description

An application role may ... other application roles.

| Name | Type |
|---|---|
| SYSTEM_ADMI | Group |
| ALLOCATION_ | Group |
| ALLOCATOR_ | Group |

Add Principal

Specify criteria to search and select the application roles that you want to grant permissions to.

Type: Group
Principal Name: Starts With

Searched Principals

| Principal | Display Name | Description |
|---|---|---|
| | | shipment priority Visbility to move orders |
| CUSTOMER_DATA_STEWARD_JOB | | |
| EBUSINESS_MANAGER_JOB | | |
| CORPORATE_STORE_PROCESS_MANAGER_JOI | | |
| BUYER_JOB | | Procurement professional responsible for transactional aspects of the procurement processes. |
| DEMAND_PLANNER_JOB | | |
| IT_SECURITY_ADMINISTRATOR_JOB | | |
| COMPETITIVE_SHOPPER_JOB | | |

The added member displays in the Members column corresponding to the application role modified in the Application Roles page.

## Creating Application Roles Using Fusion Middleware Control

Following are the two methods for creating a new application role:

- **Create New:** A new application role is created. Members can be added at the same time or you can save the new role after naming it and add members later.
- **Copy Existing:** A new application role is created by copying an existing application role. The copy contains the same members as the original, and is made a Grantee of the same application policy. You can modify the copy as needed to finish creating the new role.

### To Create a New Application Role

1. Log in to Fusion Middleware Control, navigate to **Security**, then select **Application Roles** to display the **Application Roles** page.

   For information, see Accessing Oracle Enterprise Manager Fusion Middleware Control section.

2. Choose **Select Application Stripe to Search**, and then click the search icon next to **Role Name**.

   The Oracle Retail Fusion Application's application roles displays.

3. Click **Create** to display the **Create Application Role** page. You can enter all information at once or you can enter a **Role Name**, save it, and complete the remaining fields later.
4. In the **General** section, specify the following:
   - **Role Name** - Enter the name of the application role.
   - **Display Name** - Enter the display name for the application role. This is an optional field.
   - **Description** - Enter a description for the application role. This is an optional field.
5. In the **Members** section, select the groups or application roles to be mapped to the application role.
6. Select **Add Application Role** or **Add Group** accordingly.
7. To search in the dialog box that displays, specify the following:
   - Enter a name in **Name** field and click the blue button to search.
   - Select from the results returned in the **Available** box.
   - Click **OK** to return to the **Create Application Role** page.
   - Repeat the steps until all members are added to the application role.
8. Click **OK** to return to the **Application Roles** page.

### To Create an Application Role Based on an Existing One

1. Log in to Fusion Middleware Control, navigate to **Security**, then select **Application Roles** to display the **Application Roles** page.

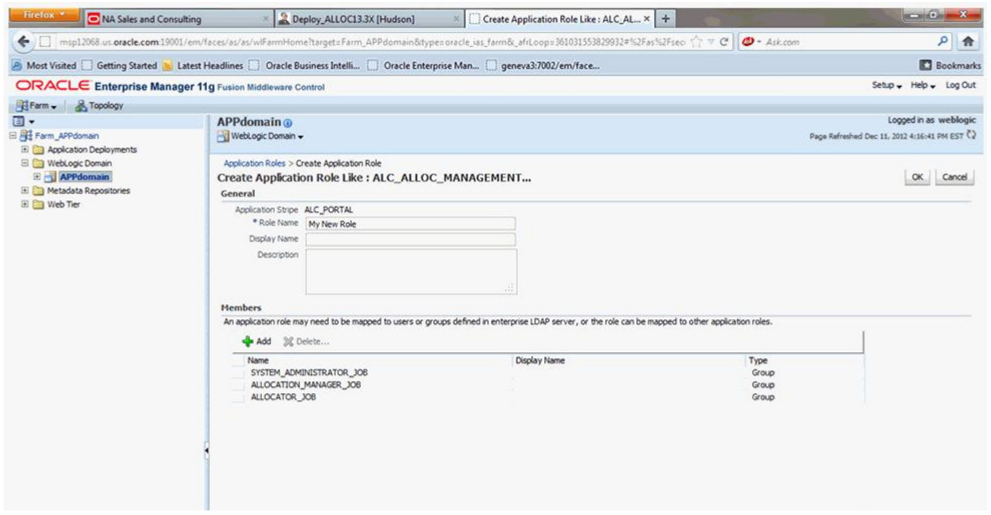
   For more information, see the Accessing Oracle Enterprise Manager Fusion Middleware Control section.
2. Choose **Select Application Stripe to Search**, and then click the search icon next to **Role Name**.

   The Oracle Retail Fusion Application's application roles are displayed.
3. Select an application role from the list to enable the action buttons.
4. Click **Create Like** to display the **Create Application Role Like** page.

   The **Members** section is completed with the same application roles and groups that are mapped to the original role.
5. Complete the **Role Name**, **Display Name**, and **Description** fields.

   Figure 8–9 shows an application role based upon ALC_ALLOC_MANAGEMENT_DUTY after being named MyNewRole, as an example.

***Figure 8–9 Create Application Role Window***



APPdomain
WebLogic Domain

Logged in as weblogic
Page Refreshed Dec 11, 2012 4:16:41 PM EST

Application Roles > Create Application Role

Create Application Role Like : ALC_ALLOC_MANAGEMENT...

General

| | |
|---|---|
| Application Stripe | ALC_PORTAL |
| * Role Name | My New Role |
| Display Name | |
| Description | |

Members

An application role may need to be mapped to users or groups defined in enterprise LDAP server, or the role can be mapped to other application roles.

| Name | Display Name | Type |
|---|---|---|
| SYSTEM_ADMINISTRATOR_JOB | | Group |
| ALLOCATION_MANAGER_JOB | | Group |
| ALLOCATOR_JOB | | Group |

**6.** Use **Add** and **Delete** to modify the members as appropriate and click **OK**.

The just created application role displays in the table at the bottom of the page. The following figure shows the example MyNewRole that is based upon the default ALC_ALLOC_MANAGEMENT_DUTY application role.

## Customizing the Default Security Configuration

You can customize the default security configuration in the following ways:

- Create new application roles. For more information, see the To Create a New Application Role section.
- Modifying membership in an Application Role. For more information, see the Modifying Application Roles Using Fusion Middleware Control section.

## Customizing the Policy Store

The Fusion Middleware Security model can be customized for your environment by creating your own application roles and modifying membership of application roles. Existing application roles can be modified by adding or removing members as needed.

For more information about managing application policies and application roles, see *Oracle Fusion Middleware Application Security Guide*.

---

**Note:** Before creating a new application role and adding it to the default Oracle Retail Fusion Application's security configuration, familiarize yourself with how permission and group inheritance works. When constructing a role hierarchy, it is important that circular dependencies are not introduced. The best practice is to leave the default security configuration in place and first incorporate your customized application roles in a test environment. For more information, see the Permission Grants and Inheritance section.

---

## Session Timeout

Session timeout is defined at the application server level. It is 60 minutes by default, but can be changed through WebLogic configuration.

# 9 ReST Services Security Consideration

The chapter provides the details on how to setup security for Representational State Transfer (ReST) Services.

The following topics are covered in this chapter:

- One Way SSL
- One Way SSL - ReST Services

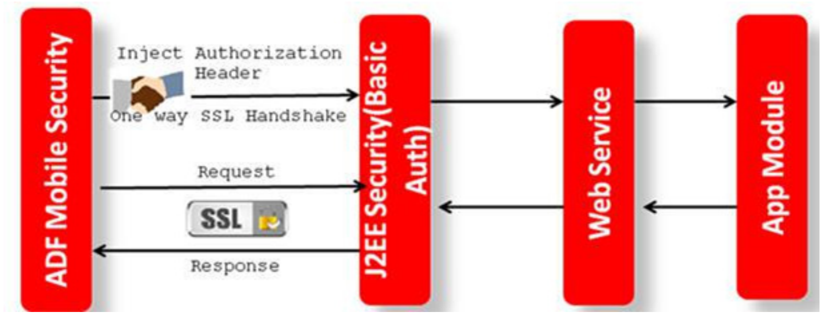
## One Way SSL

With one-way SSL, the server is required to present a certificate to the client, but the client is not required to present a certificate to the server. To successfully negotiate an SSL connection, the client must authenticate the server, but the server will accept a connection from any client. One-way SSL is common on the Internet where customers want to create secure connections before they share personal data. Often, clients will also use SSL to log on for the server to authenticate them.

ReST Committee recommends the use of one way SSL over ReST Services for 16.0.

## One Way SSL - ReST Services

***Figure 9–1 One way SSL: ReST Services***



See the following sequence:

1. ADF Mobile Security Set at the Mobile Client Level to Allow Authentication.
2. ADF Mobile Client Injects Authorization Header for Every Service Call (configuration changes).

3. J2EE-based Basic Authentication (SSL) is configured in the ReST Service Web Application Descriptor to allow secure connectivity to the ReST Service.

# 10

# Introduction to Part II: Oracle Retail Process Orchestration and Monitoring System (POM)

The following chapters provide guidance for administrators, developers, and system integrators who securely administer, customize, and integrate the Oracle Retail Process Orchestration and Monitoring System (POM) application.

- Understanding Security
- Post-Installation Application Administration

# 11 Understanding Security

This chapter contains a technical overview of the authentication process used for POM and how it uses Oracle Access manager and Single Sign-On.

## Security Features of the Application

The security features of the Application are as follows:

- **Access Control** - It is the process of restricting access to a particular entity based upon a broad range of criteria that may or may not include the attributes related to a particular user.
- **Authentication** - It is the process of verifying the identity of a user. The authentication process usually requires a user to provide a user name and password or a combination thereof, upon signing into an application.
- **Authorization** - It is the process of checking to see if an authenticated user has the privilege to access particular system functionality.
- **Data Authorization** - It is the process of determining an authenticated user's rights to act upon a particular set of data. This process typically checks if the authenticated user is linked to a certain level in the organization hierarchy and/or a certain level in the merchandise hierarchy.
- **Role-Based Access** - Within the Oracle Retail's systems, users are assigned to different roles. The role logical grouping has different access rights to specific functions within the various Oracle Retail Systems.
- **User Store** - It is a repository that holds user data required for authentication and authorization processes. Security Features of the Application

## Security Features of the Application

The levels of security offered by POM are as follows:

- **Database-level security** - This is a built in feature of Oracle Database, based on database roles.
- **Application-level security** - This is the screen-level security based on Application User roles.

## Database-level Security

For information on this section, see Pre-installation of Retail Infrastructure in WebLogic.

## Application-level Security

Application-level security requires users to authenticate at login and restricts them to only those resources for which they are authorized.

The user's access to either entire areas of the system (for example, Batch Monitoring) the modes in which users can access areas (for example, viewing Batch Monitoring only) will be restricted through this. The users are associated to groups that are mapped to application roles.The application-specific permissions are granted to these security roles. For more information on the security roles, see Figure 12–1.

# 12 Post-Installation Application Administration

This chapter covers the administration tasks performed during post installation of RMS application.

## Application Security Configuration

Access control of system resources is achieved by requiring users to authenticate at login and by restricting users to only those resources for which they are authorized. A default security configuration is available for immediate use after the Oracle Retail Fusion application is installed and is configured to use the Oracle Fusion Middleware security model. The default configuration includes eleven (11) predefined security roles for application-specific permission grants. Users can be added to predefined groups that are mapped to pre-configured application roles. RMS is pre-configured to grant specific application permissions.

*Table 12–1 Privileges*

| Name | Description |
|---|---|
| Maintain Batch Administration Priv | This privilege provides maintain access to Batch Administration Screen |
| Maintain Batch Monitoring Priv | This privilege provides maintain access to Batch Administration Screen as well as Batch Schedule Viewer Screen.This Privilege also allows to use the Action items in the above mentioned Screen. |
| View Batch Monitoring Priv | A privilege for viewing Batch Monitoring Screen |
| View Application Logs Priv | A privilege for viewing application logs |
| View Historical Batch Logs Priv | This privilege provides view access to the Historic Batch Logs stored. |
| External Configuration Priv | This privilege provides access to the External System Configuration Screen. |

*Table 12–2 Duties*

| Duty | Description | List of Privileges |
|---|---|---|
| Business User Duty | A duty for monitoring batch as a business user. | View Batch Monitoring Privilege |
| | | External Configuration Priv |

*Table 12–2 (Cont.) Duties*

| Duty | Description | List of Privileges |
|---|---|---|
| System Administration Duty | A duty for maintaining application administration information. | View Historical Batch Logs Priv<br>Maintain Batch Monitoring Privilege<br>Maintain Batch Administration Privilege<br>View Application Logs Priv |
| Batch Monitoring User Duty | A duty for maintaining the batch monitoring screen. | View Historical Batch Logs Priv<br>Maintain Batch Monitoring Privilege<br>View Application Logs Priv |
| Batch Administrator Job | System Administrator Duty | View Historical Batch Logs Priv<br>Maintain Batch Monitoring Privilege<br>Maintain Batch Administration Privilege<br>View Application Logs Priv |
| Batch Business Job | Business User Duty | View Batch Monitoring Privilege<br>External Configuration Priv |
| Batch Monitoring Job | Batch Monitoring User Duty | View Historical Batch Logs Priv<br>Maintain Batch Monitoring Privilege<br>View Application Logs Priv |

## Post-Installation Steps for Webservice Security

You need to configure the user credentials and other security-related information at the service consumer and the app service provider layers to provide end-to-end security between web service consumer and the provider.

### Applying Policy A

Applying policy A involves the following:

- Enabling the HTTPS servers
- Creating the Webservice users
- Securing services
- Updating the Webservice deployment
- Webservice Clock Skew setting

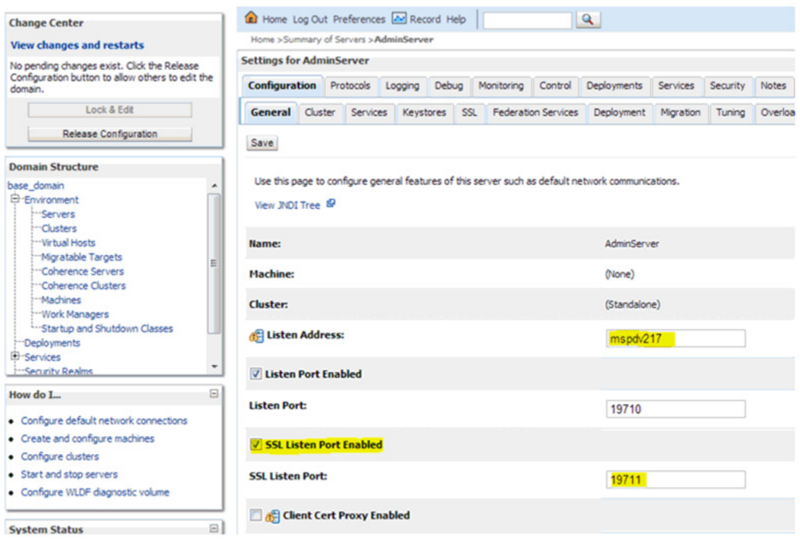
#### Enabling the HTTPS Servers

Perform the following steps to enable HTTPS servers:

1. In WebLogic Admin Console, click **Environment > Servers**.

2. Click the server where the web service has been deployed.
3. Click the **General** tab.
4. Check the **SSL Listen Port Enabled** check box.
5. Enter a port number for the **SSL Listen Port**. This is the port number for service end point.
6. Enter the hostname in **Listen Address** field.
7. Click **Save**.

*Figure 12–1 Enabling the HTTPS Servers*


Change Center
View changes and restarts
Lock & Edit
Release Configuration
Domain Structure
base_domain
Environment
Servers
Clusters
Virtual Hosts
Migratable Targets
Coherence Servers
Coherence Clusters
Machines
Work Managers
Startup and Shutdown Classes
Deployments
Services
Security Realms
How do I...
System Status
Home Log Out Preferences Record Help
Configuration Protocols Logging Debug Monitoring Control Deployments Services Security Notes
General Cluster Services Keystores SSL Federation Services Deployment Migration Tuning Overloa
Save
View JNDI Tree

No pending changes exist. Click the Release Configuration button to allow others to edit the domain.

- Configure default network connections
- Create and configure machines
- Configure clusters
- Start and stop servers
- Configure WLDF diagnostic volume

Home >Summary of Servers >AdminServer

Settings for AdminServer

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

| | |
|---|---|
| Name: | AdminServer |
| Machine: | (None) |
| Cluster: | (Standalone) |
| Listen Address: | mspdv217 |
| ✓ Listen Port Enabled | |
| Listen Port: | 19710 |
| ✓ SSL Listen Port Enabled | |
| SSL Listen Port: | 19711 |
| ☐ Client Cert Proxy Enabled | |

## Creating the Webservice User

Perform the following steps to create roles and users who can access the Web services:

1. In WebLogic Admin Console, click the **Domain Structure** window, and click the **Security Realms** link.

   The default realm appears.
2. Click the link on the realm.
3. Click the **Users and Groups** tab.
4. Click **New**.
5. Enter the user name and password details on the next screen.
6. Leave the default value for **Provider**.
7. Click **OK** to save the changes.

   The new user is shown in the list of users.

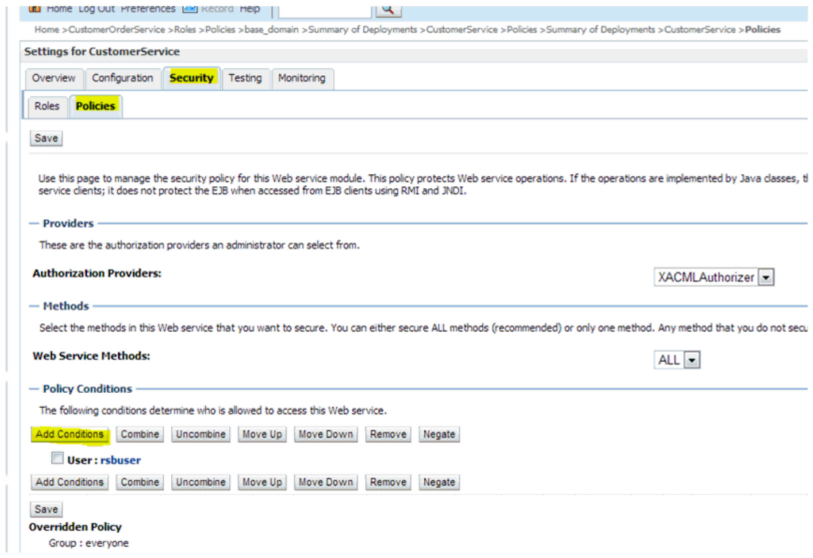
## Securing Services

Perform the following steps in WebLogic Admin Console for each of the services to be secured:

1. Attach the user created in previous step to the service.

2. Click **Deployments**.
3. Click the service you want to secure.
4. Click **Securities** and then **Policies**.

***Figure 12–2 Securing Services***



Home >CustomerOrderService >Roles >Policies >base_domain >Summary of Deployments >CustomerService >Policies >Summary of Deployments >CustomerService >**Policies**

**Settings for CustomerService**

Use this page to manage the security policy for this Web service module. This policy protects Web service operations. If the operations are implemented by Java classes, tl
service clients; it does not protect the EJB when accessed from EJB clients using RMI and JNDI.

**Providers**

These are the authorization providers an administrator can select from.

**Authorization Providers:** XACMLAuthorizer

**Methods**

Select the methods in this Web service that you want to secure. You can either secure ALL methods (recommended) or only one method. Any method that you do not secu

**Web Service Methods:** ALL

**Policy Conditions**

The following conditions determine who is allowed to access this Web service.

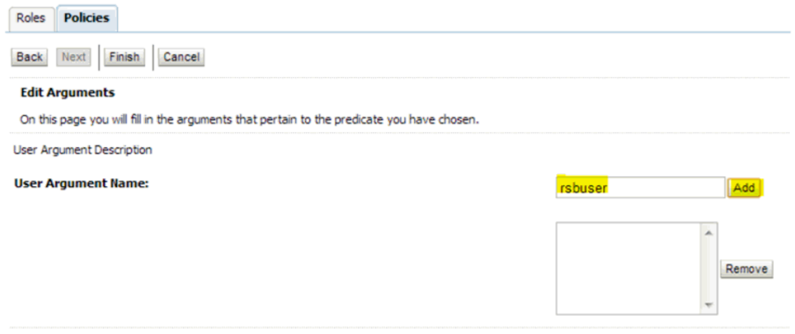
**User : rsbuser**

**Overridden Policy**

Group : everyone

5. Click **Add Conditions**.
6. Click **Predict List: Pick User from dropdown**, then click **Next**.
7. Click **User Argument Name**.
8. Enter the username you created, then click **Add**.
9. Click **Finish**, then **Save**.

***Figure 12–3 Add Conditions Window***



**Edit Arguments**

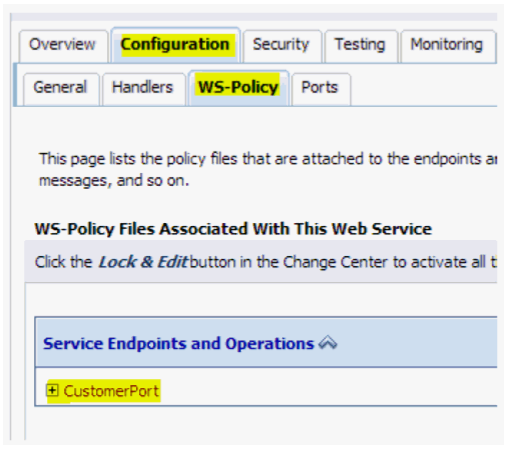
On this page you will fill in the arguments that pertain to the predicate you have chosen.

User Argument Description

**User Argument Name:** rsbuser

10. Attach the policy to the service.
11. Navigate to **Configuration** tab.
12. Click the **WSB Policy** tab and select the service port.

*Figure 12–4 Attaching WS Policy to the Service*



This page lists the policy files that are attached to the endpoints a
messages, and so on.

WS-Policy Files Associated With This Web Service

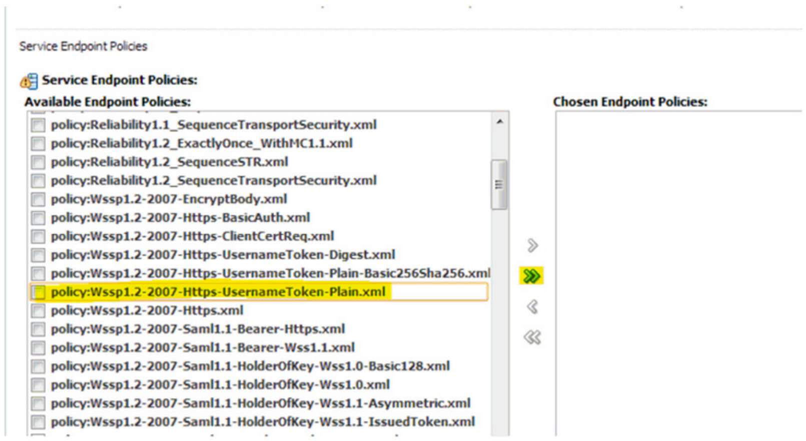
Click the *Lock & Edit* button in the Change Center to activate all t

Service Endpoints and Operations

CustomerPort

13. Click **WebLogic**, then **Next**.
14. Click **Service Endpoint Policies**.
15. Select **policy:Wssp1.22007HttpsUsernameTokenPlain.xml**, then click **Finish**.

*Figure 12–5 Service Endpoint Policies*



Service Endpoint Policies

Service Endpoint Policies:

Available Endpoint Policies:

- policy:Reliability1.1_SequenceTransportSecurity.xml
- policy:Reliability1.2_ExactlyOnce_WithMC1.1.xml
- policy:Reliability1.2_SequenceSTR.xml
- policy:Reliability1.2_SequenceTransportSecurity.xml
- policy:Wssp1.2-2007-EncryptBody.xml
- policy:Wssp1.2-2007-Https-BasicAuth.xml
- policy:Wssp1.2-2007-Https-ClientCertReq.xml
- policy:Wssp1.2-2007-Https-UsernameToken-Digest.xml
- policy:Wssp1.2-2007-Https-UsernameToken-Plain-Basic256Sha256.xml
- policy:Wssp1.2-2007-Https-UsernameToken-Plain.xml
- policy:Wssp1.2-2007-Https.xml
- policy:Wssp1.2-2007-Saml1.1-Bearer-Https.xml
- policy:Wssp1.2-2007-Saml1.1-Bearer-Wss1.1.xml
- policy:Wssp1.2-2007-Saml1.1-HolderOfKey-Wss1.0-Basic128.xml
- policy:Wssp1.2-2007-Saml1.1-HolderOfKey-Wss1.0.xml
- policy:Wssp1.2-2007-Saml1.1-HolderOfKey-Wss1.1-Asymmetric.xml
- policy:Wssp1.2-2007-Saml1.1-HolderOfKey-Wss1.1-IssuedToken.xml

Chosen Endpoint Policies:

16. Click **OK** if WebLogic prompts you to save Plan.xml.

## Updating the Webservice Deployment

Perform the following steps to update the Webservice deployment:

1. In WebLogic Admin Console, click **Deployments**.
2. Click **Lock & Edit** and select the deployed application with the Webservices to be secured.
3. Click **Update** and select the deployment `.ear` file, along with the `Plan.xml` file if it was saved earlier.
4. Click **Finish**.
5. Click **Activate Changes** to reflect the changes.

6. Verify the configuration by checking the WSDL of the service.

   The WSDL must have the policy information in it.

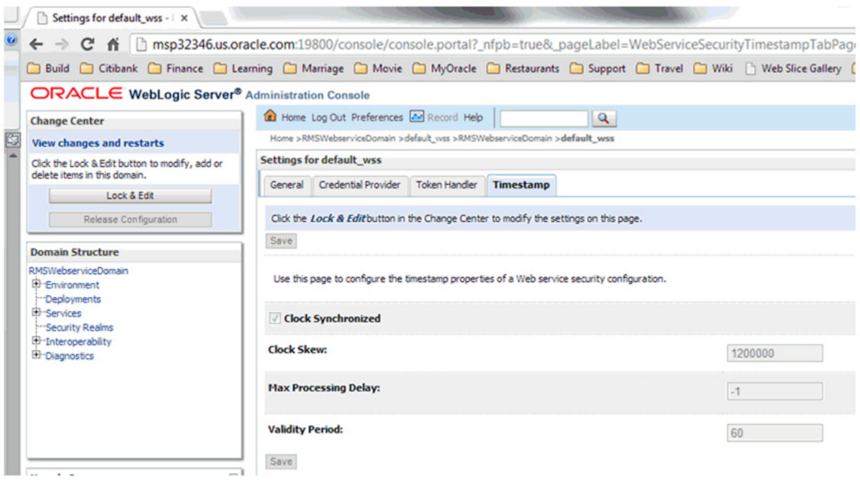
### Webservice Clock Skew Setting

Webservices, when secured, need to be time-synced with providers and consumers. However, for various reasons, the providers and consumers can have different time gaps.

Weblogic can be configured to different tolerance level for webservices to work. Perform the following steps to set the time tolerance level to a different value:

1. Navigate to **WLS Console > Domain > Web Service Security > default_wss > Timestamp**.
2. Click **Lock and Edit**.
3. Update the **Clock Skew** with the new tolerance limit (in milliseconds).
4. Click **Activate Changes**.
5. Restart the managed server hosting Webservice once the changes are implemented.

***Figure 12–6 Set New Time Tolerance***



Settings for default_wss
msp32346.us.oracle.com:19800/console/console.portal?_nfpb=true&_pageLabel=WebServiceSecurityTimestampTabPag
Build Citibank Finance Learning Marriage Movie MyOracle Restaurants Support Travel Wiki Web Slice Gallery
WebLogic Server® Administration Console
Home Log Out Preferences Record Help
Change Center
View changes and restarts
Lock & Edit
Release Configuration
Domain Structure
RMSWebserviceDomain
Environment
Deployments
Services
Security Realms
Interoperability
Diagnostics
General Credential Provider Token Handler Timestamp
Save

Home >RMSWebserviceDomain >default_wss >RMSWebserviceDomain >**default_wss**

Settings for default_wss

Click the Lock & Edit button to modify, add or delete items in this domain.

Click the ***Lock & Edit*** button in the Change Center to modify the settings on this page.

Use this page to configure the timestamp properties of a Web service security configuration.

| | |
|---|---|
| ✓ Clock Synchronized | |
| Clock Skew: | 1200000 |
| Max Processing Delay: | -1 |
| Validity Period: | 60 |

## Applying Policy B

Applying policy B involves the following:

- Creating the Webservice users
- Securing services
- Updating the Webservice deployment

### Creating the Webservice User

Perform the following steps to create roles and users who can access the Web services:

1. In WebLogic Admin Console, click the **Domain Structure** window, then click the **Security Realms** link.

   The default realm appears.

2. Click the link on the realm.
3. Click the **Users and Groups** tab.
4. Click **New**.
5. Enter the user name and password details on the next screen.
6. Leave the default value for **Provider**.
7. Click **OK** to save the changes.

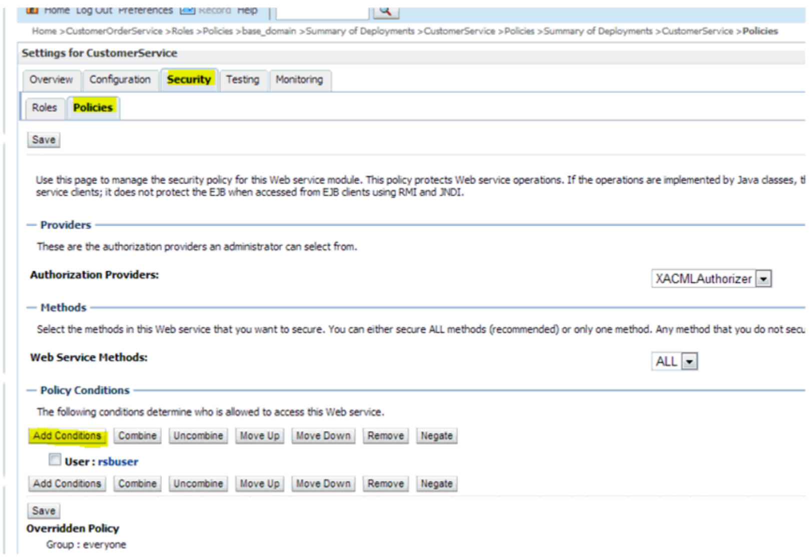
   The new user is shown in the list of users.

### Securing Services

Perform the following steps in WebLogic Admin Console for each of the services to be secured:

1. Attach the user created in previous step to the service.
2. Click **Deployments**.
3. Click the service you want to secure.
4. Click **Securities**, then **Policies**.

***Figure 12–7 Securing Services***


Home Log Out Preferences Record Help
Home >CustomerOrderService >Roles >Policies >base_domain >Summary of Deployments >CustomerService >Policies >Summary of Deployments >CustomerService >Policies
Overview
Configuration
Security
Testing
Monitoring
Roles
Policies
Save
Add Conditions
Combine
Uncombine
Move Up
Move Down
Remove
Negate

Settings for CustomerService

Use this page to manage the security policy for this Web service module. This policy protects Web service operations. If the operations are implemented by Java classes, tl
service clients; it does not protect the EJB when accessed from EJB clients using RMI and JNDI.

**Providers**

These are the authorization providers an administrator can select from.

**Authorization Providers:** XACMLAuthorizer

**Methods**

Select the methods in this Web service that you want to secure. You can either secure ALL methods (recommended) or only one method. Any method that you do not secu

**Web Service Methods:** ALL

**Policy Conditions**

The following conditions determine who is allowed to access this Web service.

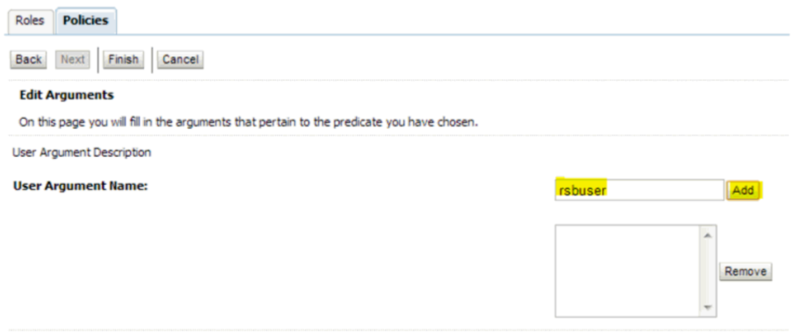
**User :** rsbuser

**Overridden Policy**

Group : everyone

5. Click **Add Conditions > Predict List:**.
6. Pick the **User** from the dropdown, then click **Next**
7. Click **User Argument Name**.
8. Type the username you created, then click **Add**.
9. Click **Finish**, then click **Save**.

***Figure 12–8 Add Condition Window***



**Edit Arguments**

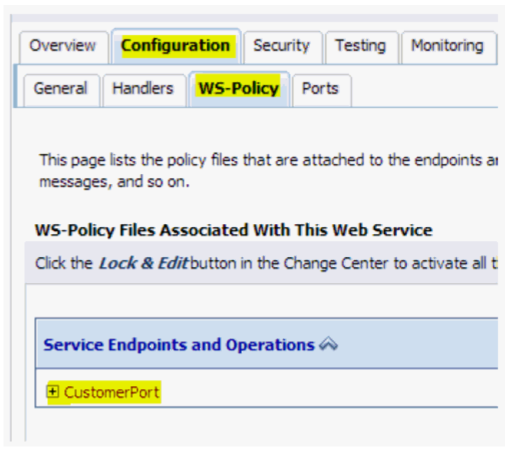
On this page you will fill in the arguments that pertain to the predicate you have chosen.

User Argument Description

**User Argument Name:** rsbuser

10. Attach policy to the service.
11. Navigate to the **Configuration** tab.
12. Click the **WSB Policy** tab and select the service port.

***Figure 12–9 Attaching WS Policy to the Service***



This page lists the policy files that are attached to the endpoints ar messages, and so on.

**WS-Policy Files Associated With This Web Service**

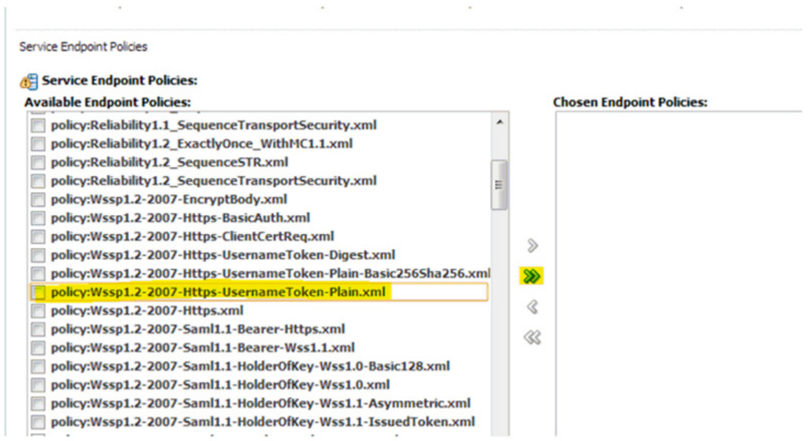
Click the ***Lock & Edit*** button in the Change Center to activate all t

**Service Endpoints and Operations**

CustomerPort

13. Click **WebLogic**, then click **Next**.
14. Click **Service Endpoint Policies**.
15. Select **policy:Wssp1.22007HttpsUsernameTokenPlain.xml**, then click **Finish**.

*Figure 12–10 Service Endpoint Policies*



Service Endpoint Policies

**Service Endpoint Policies:**

**Available Endpoint Policies:**

- [ ] policy:Reliability1.1_SequenceTransportSecurity.xml
- [ ] policy:Reliability1.2_ExactlyOnce_WithMC1.1.xml
- [ ] policy:Reliability1.2_SequenceSTR.xml
- [ ] policy:Reliability1.2_SequenceTransportSecurity.xml
- [ ] policy:Wssp1.2-2007-EncryptBody.xml
- [ ] policy:Wssp1.2-2007-Https-BasicAuth.xml
- [ ] policy:Wssp1.2-2007-Https-ClientCertReq.xml
- [ ] policy:Wssp1.2-2007-Https-UsernameToken-Digest.xml
- [ ] policy:Wssp1.2-2007-Https-UsernameToken-Plain-Basic256Sha256.xml
- [ ] policy:Wssp1.2-2007-Https-UsernameToken-Plain.xml
- [ ] policy:Wssp1.2-2007-Https.xml
- [ ] policy:Wssp1.2-2007-Saml1.1-Bearer-Https.xml
- [ ] policy:Wssp1.2-2007-Saml1.1-Bearer-Wss1.1.xml
- [ ] policy:Wssp1.2-2007-Saml1.1-HolderOfKey-Wss1.0-Basic128.xml
- [ ] policy:Wssp1.2-2007-Saml1.1-HolderOfKey-Wss1.0.xml
- [ ] policy:Wssp1.2-2007-Saml1.1-HolderOfKey-Wss1.1-Asymmetric.xml
- [ ] policy:Wssp1.2-2007-Saml1.1-HolderOfKey-Wss1.1-IssuedToken.xml

**Chosen Endpoint Policies:**

16. Click **OK** if WebLogic prompts you to save Plan.xml.

## Updating the Webservice Deployment

Perform the following steps to update the Webservice deployment:

1. In WebLogic Admin Console, click **Deployments**.
2. Click **Lock & Edit** and select the deployed application with the Webservices to be secured.
3. Click **Update** and select the deployment ear along with the Plan.xml if saved in the previous steps.
4. Click **Finish**.
5. Click **Activate Changes** to reflect the changes.
6. Verify the configuration by checking the WSDL of the service.

   The WSDL must have the policy information in it.