



# HYPERION® SHARED SERVICES

RELEASE 9.3.1.1

## CONFIGURING OPENLDAP FOR SSL/ TLS COMMUNICATION

ORACLE | Hyperion

### CONTENTS IN BRIEF

About this Document .....	2
About OpenLDAP .....	2
Assumptions .....	2
Required Certificates .....	2
Configuring OpenLDAP for SSL .....	4

## About this Document

This document explains how to secure communication between System 9 and OpenLDAP using Secure Socket Layer (SSL).

This document describes how to deploy Hyperion products to ensure that communication with OpenLDAP uses Secure Socket Layer (SSL). SSL is a cryptographic protocol used to secure data exchange over the network.

## About OpenLDAP

Oracle's Hyperion® Shared Services uses OpenLDAP to support provisioning. OpenLDAP is an open-source Lightweight Directory Access Protocol (LDAP)-enabled user directory that is bundled and configured with Shared Services.

Shared Services uses OpenLDAP to maintain and manage the default user account required by Hyperion security and to store provisioning information; the relationships among users, groups, and roles. In Hyperion parlance, OpenLDAP is referred to as Native Directory.

OpenLDAP is deployed to use port 58089 on the server that hosts Shared Services. By default, the communication between Hyperion products and OpenLDAP does not use SSL.

## Assumptions

This white paper assumes that Shared Services Version 9.3.1.1 is installed and configured.

- Shared Services Version 9.3.1.1 is installed and configured.
- SSL configuration of Hyperion products, including Shared Services is complete. See the [Hyperion SSL Configuration Guide](#).
- You are using OpenSSL to complete the tasks described in this white paper; for example, to generate the Certificate Signing Request (CSR) for OpenLDAP server. If you are using a different tool, follow the instructions in the tool's documentation to complete the tasks explained in this white paper.

## Required Certificates

Hyperion supports one-way SSL communication between Shared Services and OpenLDAP. In this mode, SSL-enabled OpenLDAP presents a certificate to Shared Services.

This scenario requires two certificates:

- A root certificate from the Certificate Authority (CA).
- A signed server certificate from a CA

To establish an SSL connection, Shared Services must trust the CA that issued OpenLDAP's certificate. Java, and by extension, Shared Services, understands trusted third-party CAs because

their root certificates are already available in the Java Virtual Machine (JVM) used by Shared Services.

You must obtain a signed certificate for each OpenLDAP host server. You can reuse an existing certificate; for example one that was used for SSL-enabling Hyperion products, under these conditions:

- You are allowed to use the certificate more than once.
- DN in the certificate exactly matches the host name or IP address of the OpenLDAP host machine.

You may need to export the existing certificate before you can use it to SSL-enable OpenLDAP.

If you use an unknown CA; for example, your own CA or a CA such as OpenSSL, to create and sign certificates, you must import the CA's root (public key) certificate into the keystore used by Shared Services to establish trust. You should also configure OpenLDAP so that it presents the signed certificate to Shared Services to establish trust.

## Obtaining Certificates from Trusted CAs

You require a certificate for each OpenLDAP instance you want to SSL-enable. Oracle recommends that you use certificates from well known third party CA.

Obtaining a certificate from a CA typically involves the following steps:

- Generating a certificate request and sending it to the CA for signing
- Receiving the digitally signed certificate from the CA.

➤ To generate a CSR for OpenLDAP server using OpenSSL:

- 1 From a console, enter the following command to create a CSR (named `openldapserver.csr` in the following command):

```
openssl req -new -out openldapserver.csr
```

- 2 Enter required information. OpenSSL prompts are indicated in bold typeface.

**Enter pass phrase for OpenLDAP\_CA.key:**

You are about to be asked to enter information that will be incorporated into your certificate request.

What you are about to enter is what is called a Distinguished Name or a DN.

There are quite a few fields but you can leave some blank

For some fields there will be a default value,

If you enter '.', the field will be left blank.

-----

**Country Name (2 letter code) [AU]:**US

**State or Province Name (full name) [Some-State]:**California

**Locality Name (eg, city) []:**Santa Clara

**Organization Name (eg, company) [Internet Widgets Pty Ltd]:**Oracle

**Organizational Unit Name (eg, section) []:**Hyperion

**Common Name (eg, YOUR name) []:**10:10:12:85

**Email Address []:**

Please enter the following 'extra' attributes to be sent with your certificate

```
request
A challenge password []:password
An optional company name []:
```

## Removing Password Protection for Private Key

The CSR generation process for OpenLDAP server creates `privkey.pem`, which contains the private key used for the CSR. You must remove the password from `privkey.pem` so that OpenLDAP does not prompt for a password during startup.

➤ To remove password from private key:

- 1 From a console, navigate to the directory where the CSR and `privkey.pem` are stored.
- 2 Enter the following command to create a key file (named `my-server.key` in the following command):

```
openssl rsa -in privkey.pem -out my-server.key
```

## Configuring OpenLDAP for SSL

### Subtopics

- [Add CA Public Key \(root\) Certificate to cacerts Used by Shared Services](#)
- [Copy Certificates and Key File into a Secure Location](#)
- [Update OpenLDAP Configuration](#)
- [Enable OpenLDAP SSL Port](#)
- [Update OpenLDAP Configuration Settings](#)

## Add CA Public Key (root) Certificate to cacerts Used by Shared Services

By default, Java installs `cacerts`, a certificates file, which represents a system-wide keystore with CA certificates. The `cacerts` keystore ships with several root CA certificates. If you are using a certificate from a trusted third-party CA, you do not need to complete this procedure because the `cacerts` file already contains the required CA certificate.

If you are not using a certificate from a trusted third-party CA; for example, if you are using a self-signed certificate, you must add the CA Certificate to the `cacerts` used by Shared Services. You use `keytool` to add CA certificate to `cacerts`.

➤ To add root CA certificate to Shared Services JVM:

- 1 From a console, change directory to `HYPERION_HOME/common/JRE/Sun/1.5.0/lib/security`.
- 2 Execute the following command:

```
HYPERION_HOME/common/JRE/Sun/1.5.0/bin/keytool -import -alias
CERTIFICATE_NAME -keystore cacerts -storepass changeit -file CA_CERTIFICATE
```

For example, use this command to add a certificate with `myopenldapcert` as name in `cacerts` using `C:/certificates/OpenLDAP_CA.crt`.

```
C:/Hyperion/common/JRE/Sun/1.5.0/bin/keytool -import -alias myopenldapcert -keystore cacerts -storepass changeit -file C:/certificates/OpenLDAP_CA.crt
```

On executing this command, `keytool` displays the following:

```
Owner: CN=myCA, OU=Hyperion, O=Oracle Corporation, L=Santa Clara, ST=California, C=US
Issuer: CN=myCA, OU=Hyperion, O=Oracle Corporation, L=Santa Clara, ST=California, C=US
Serial number: elec374dcb50a3b3
Valid from: Thu Nov 05 11:44:46 PST 2009 until: Tue Jan 13 11:44:46 PST 2015
Certificate fingerprints:
    MD5:  E0:ED:6B:CA:0B:26:2D:39:55:26:69:63:FA:9A:8A:C8
    SHA1: 42:81:93:EB:17:DB:14:CF:4F:DD:BE:0B:72:2A:6A:25:61:0C:A2:92
Trust this certificate? [no]:
```

- 3 Enter `yes` to add the certificate to the keystore.
- 4 Restart Shared Services.

## Copy Certificates and Key File into a Secure Location

Copy the following files into a secure directory on the OpenLDAP server.

- CA Certificate
- Signed Certificate for OpenLDAP server

## Update OpenLDAP Configuration

You must update the OpenLDAP configuration file, `slapd.conf` to configure OpenLDAP to use the signed server certificate.

**Note:** In high availability deployments of OpenLDAP, this step must be completed on each OpenLDAP installation. Note that each server requires a separate signed certificate.

➤ To configure OpenLDAP for SSL

- 1 Using a text editor, open `slapd.conf`. Location of `slapd.conf`:

```
HYPERION_HOME/SharedServices/9.3.1/openLDAP/slapd.conf (Windows)
```

```
HYPERION_HOME/SharedServices/9.3.1/openLDAP/usr/local/etc/openldap/slapd.conf (UNIX)
```

- 2 Append entries similar to the following to `slapd.conf`:

**Note:** This example assumes that you are using RSA encryption.

```
TLSCipherSuite HIGH:MEDIUM:+SSLv2:+SSLv3:RSA
TLSCertificateFile absolute_location_of_OpenLDAP_Server_Certificate
TLSCertificateKeyFile absolute_location_of_OpenLDAP_Server_privatekey_file
TLSCACertificateFile absolute_location_of_CA_Certificate_file
TLSVerifyClient allow
```

Your entries could be as follows:

```
TLSCipherSuite HIGH:MEDIUM:+SSLv2:+SSLv3:RSA
TLSCertificateFile C:/OpenLDAP_SSL_files/openldapserver.crt
TLSCertificateKeyFile C:/OpenLDAP_SSL_files/openldapserver.key
TLSCACertificateFile C:/OpenLDAP_SSL_files/OpenLDAP_CA.crt
TLSVerifyClient allow
```

### 3 Save and close `slapd.conf`.

## Enable OpenLDAP SSL Port

### Subtopics

- [Windows](#)
- [UNIX](#)

### Windows

You use the Windows Registry Editor to enable OpenLDAP SSL port.

► To enable OpenLDAP SSL port:

- 1 **Launch Windows Registry Editor.**
  - a. Select **Start** and then **Run**.
  - b. In Run dialog box, enter `regedit`.
  - c. Click **OK**.
- 2 In Windows Registry Editor, select **MY Computer**, then **HKEY\_LOCAL\_MACHINE**, then **SOFTWARE**, then **OpenLDAP**, and then **Parameters**.
- 3 Right-click **Urls** string value and select **Modify**.
- 4 **Edit Value Data**
  - To force OpenLDAP to listen on the secure port only, replace the existing value with `ldaps://LDAP_PORT_NUMBER`; for example, `ldaps://58092`
  - To allow OpenLDAP to listen on secure and nonsecure ports, append `ldaps://LDAP_PORT_NUMBER` to the existing value; for example, `ldap://58089`  
`ldaps://58092`.

**Note:** Use a space to separate the new value from the existing value.

Be sure to substitute `LDAP_PORT_NUMBER` with the secure port number as indicated in the examples.

- 5 Click **OK**.
- 6 Restart OpenLDAP.

## UNIX

Start OpenLDAP with the `-h` option to use both secure and nonsecure ports.

```
slapd -f slapd.conf -h ldaps://server:58092 ldap://:28089 -d
```

## Update OpenLDAP Configuration Settings

Default OpenLDAP settings configure a non-SSL connection between Oracle's Hyperion® Shared Services and OpenLDAP. You must modify the settings in `CSS.xml` to configure a secure connection using the SSL port (see [“Enable OpenLDAP SSL Port” on page 6](#)).

► To update `CSS.xml`:

- 1 Using a text editor, open the `HYPERION_HOME/deployments/APP_SERVER/SharedServices9/config/CSS.xml` file. For example, if you are using WebLogic application server on Windows, `CSS.xml` is generally located in `C:/Hyperion/deployments/WebLogic9/SharedServices9/config`.
- 2 In hub location, change the value of `<dirPort>` attribute to the number of the SSL-enabled OpenLDAP port.

See [“Enable OpenLDAP SSL Port” on page 6](#).

- 3 In `<provider>` definition, add the following new attribute:

```
<authProtocol>ssl</authProtocol>
```

- 4 Your updated settings may be as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Sample XML file generated by XML Spy v4.4 U (http://www.xmlspy.com)-->
<css>
  <hub location="http://myServer:58080">
    <dirPort>58092</dirPort>
  </hub>
  <spi>
    <provider>
      <native name="Native Directory">
        <authProtocol>ssl</authProtocol>
        <password>{CSS}4N6lVcgiE/dRg8rFdvQLcA==</password>
      </native>
    </provider>
  </spi>
</css>
```

- 5 Save and close `CSS.xml`.

## COPYRIGHT NOTICE

Shared Services Configuring OpenLDAP for SSL/TLS Communication, 9.3.1.1

Copyright © 2009, Oracle and/or its affiliates. All rights reserved.

Authors: EPM Information Development Team

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited. The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

### U.S. GOVERNMENT RIGHTS:

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

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

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

This software and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.