# HP Operations Orchestration

For the Windows and Linux

Software Version: 10.10

## System Configuration and Hardening Guide

Document Release Date: May 2014

Software Release Date: May 2014

# Legal Notices

## Warranty

The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

## Restricted Rights Legend

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

## Copyright Notice



## Trademark Notices

Adobe™ is a trademark of Adobe Systems Incorporated.

This product includes an interface of the 'zlib' general purpose compression library, which is Copyright © 1995-2002 Jean-loup Gailly and Mark Adler.

AMD and the AMD Arrow symbol are trademarks of Advanced Micro Devices, Inc.

Google™ and Google Maps™ are trademarks of Google Inc.

Intel®, Itanium®, Pentium®, and Intel® Xeon® are trademarks of Intel Corporation in the U.S. and other countries.

Java is a registered trademark of Oracle and/or its affiliates.

Microsoft®, Windows®, Windows NT®, Windows® XP, and Windows Vista® are U.S. registered trademarks of Microsoft Corporation.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates.

UNIX® is a registered trademark of The Open Group.

# Documentation Updates

The title page of this document contains the following identifying information:

- Software Version number, which indicates the software version.
- Document Release Date, which changes each time the document is updated.
- Software Release Date, which indicates the release date of this version of the software.

To check for recent updates or to verify that you are using the most recent edition of a document, go to: **http://h20230.www2.hp.com/selfsolve/manuals**

This site requires that you register for an HP Passport and sign in. To register for an HP Passport ID, go to: **http://h20229.www2.hp.com/passport-registration.html**

Or click the **New users - please register** link on the HP Passport login page.

You will also receive updated or new editions if you subscribe to the appropriate product support service. Contact your HP sales representative for details.

# Support

Visit the HP Software Support Online web site at: **http://www.hp.com/go/hpsoftwaresupport**

This web site provides contact information and details about the products, services, and support that HP Software offers.

HP Software online support provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by using the support web site to:

- Search for knowledge documents of interest
- Submit and track support cases and enhancement requests
- Download software patches
- Manage support contracts
- Look up HP support contacts
- Review information about available services
- Enter into discussions with other software customers
- Research and register for software training

Most of the support areas require that you register as an HP Passport user and sign in. Many also require a support contract. To register for an HP Passport ID, go to:

**http://h20229.www2.hp.com/passport-registration.html**

To find more information about access levels, go to:

**http://h20230.www2.hp.com/new_access_levels.jsp**

**HP Software Solutions Now** accesses the HPSW Solution and Integration Portal Web site. This site enables you to explore HP Product Solutions to meet your business needs, includes a full list of Integrations between HP Products, as well as a listing of ITIL Processes. The URL for this Web site is **http://h20230.www2.hp.com/sc/solutions/index.jsp**

# Contents

# System Configuration and Hardening

This document describes how to configure and harden HP Operations Orchestration.

# Server and Client Certificate Authentication

Secure Socket Layer (SSL)/Transport Layer Security (TLS) certificates digitally bind a cryptographic key to the details of an organization, enabling secure connections from a web server to a browser.

HP OO uses the Keytool utility to manage cryptographic keys and trusted certificates. This utility is included in the HP OO installation folder, in **<Installation dir>/java/bin/keytool**. For more information about the Keytool utility, see http://docs.oracle.com/javase/7/docs/technotes/tools/solaris/keytool.html.

Installations of HP OO Central include two files for the management of certificates:

- **<installation dir>/central/var/security/client.truststore**: Contains the list of trusted certificates.
- **<installation dir>/central/var/security/key.store**: Contains the HP OO certificate.

It is recommended that you replace the HP OO certificate after a new installation of HP OO or if your current certificate is expired.

## *Server Certificate Authentication*

### *Replacing the Central SSL/TLS Server Certificate*

You can use a certificate signed by a well-known company or a custom server certificate.

Replace the parameters that are highlighted in <yellow> to match the location of the **key.store** file and other details on your computer.

> **Note:** The following procedure uses the Keytool utility that is located in **<installation dir>/java/bin/keytool**.

1. Stop Central and back up the original **key.store** file, located in **<installation dir>/central/var/security/key.store**.
2. Open a command line in **<installation dir>/central/var/security**.
3. Delete the existing server certificate from the Central **key.store** file, using the following command:

```
keytool -delete -alias tomcat -keystore key.store -storepass changeit
```

4. If you already have a certificate with **.pfx** or **.p12** extension, then go to the next step. If not, then you need to export the certificate with private key into PKCS12 format (.pfx,.p12). For example, if the certificate format is PEM:

   ```
   >openssl pkcs12 –export –in <cert.pem> -inkey <key.key> -out <certificate name>.p12 –name <name>
   ```

   If the certificate format is DER, add the `–inform DER` parameter after pkcs12.For example:

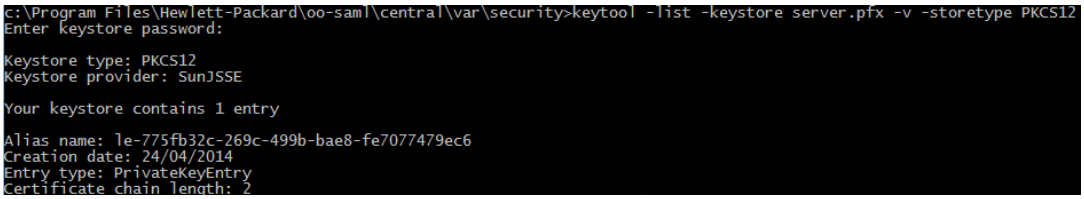
   ```
   >openssl pkcs12 –inform DER –export –in <cert.pem> -inkey <.key> -out <certificate name>.p12 –name <name>
   ```

   > **Note:** Make a note of the password that you provide. You will need this password for the private key when you input the keystore passphrase later in this procedure.

5. Extract the alias for your certificate's alias, using the following command:

   ```
   keytool -list -keystore <certificate_name> -v -storetype PKCS12
   ```

   The alias is displayed. In the example below, it is the fourth line from the bottom.

6. Import the PKCS12 format server certificate to the Central **key.store** file:

   keytool -importkeystore -srckeystore <PKCS12 format certificate path> -destkeystore key.store -srcstoretype pkcs12 -deststoretype JKS -alias <cert alias> -destalias tomcat

7. It is recommended to change the default "changeit" password in the automatically-generated keystore in the Central server. See "Changing the Keystore/Truststore Password" on page 10.

8. Start Central.

## *Replacing the Central SSL/TLS Server Certificate With a Self-Signed Certificate*

You can generate a self-signed certificate using the Keytool utility.

> **Note:** After upgrading to HP OO 10.10:

- If a new Central is installed on the same machine as the previous installation, you can use the existing self-signed certificate.
- If new Centrals are installed on different machines, you need to generate a new self-signed certificate for each one, even if you had a certificate for the previous version.

**Note:** The following procedure uses the Keytool utility that is located in **<installation dir>/java/bin/keytool**.

Replace the parameters that are highlighted in <yellow> to match the location of the **key.store** file and other details on your computer.

1. Stop Central and back up the original **key.store** file, located in **<installation dir>/central/var/security/key.store**.
2. Open a command line in **<installation dir>/central/var/security**.
3. Delete the existing server certificate from the Central **key.store** file, using the following command:

   ```
   keytool -delete -alias tomcat -keystore key.store -storepass <changeit>
   ```

4. Generate a self-signed certificate:

   ```
   keytool -genkey -alias tomcat -keyalg RSA -keypass <changeit >-keystore <pat
   h/for/new/Keystore> -storepass <changeit>-storetype pkcs12 -dname "CN=<CENTR
   AL_FQDN>, OU=<ORGANIZATION_UNIT>, O=<ORGANIZATION>, L=<LOCALITY>,
   C=<COUNTRY>"
   ```

   **Note:** If you do not enter a path for generating the new keystore, it is created in the folder where you entered the command, for example **<installation dir>/central/var/security**.

5. Import the self-signed certificate to the Central **key.store** file:

   ```
   keytool -v -importkeystore -srckeystore <new/path/created/Keystore> -srcstor
   etype PKCS12 -srcstorepass <changeit> -destkeystore key.store -deststoretype
   JKS -deststorepass <changeit>
   ```

6. Start Central.

## *Importing a Certificate to a RAS Truststore*

After installing a RAS, if you are using a custom root certificate for Central and you didn't provide this root certificate during the RAS installation, you will need to import the trusted root certificate authority (CA) to the RAS **client.truststore**. If you are using a standard signed root certificate you

do not have to perform the following procedure as the certificate will already be in the **client.truststore** file.

By default, HP OO supports all self-signed certificates. However, in a production environment, it is recommended to change this default for security reasons.

Replace the parameters that are highlighted in <yellow>.

**Note:** The following procedure uses the Keytool utility that is located in **<installation dir>/java/bin/keytool**.

1. Stop RAS and back up the original **client.truststore** file, located in **<installation dir>/ras/var/security/client.truststore**.
2. Open command line in **<installation dir>/ras/var/security**.
3. Open the **<installation dir> ras/conf/ras-wrapper.conf** file and set the `-Dssl.support-self-signed` value to **false**. This enables the trusted root certificate authority (CA).

   For example:

   ```
   wrapper.java.additional.<x>=-Dssl.support-self-signed=false
   ```

4. Open the **<installation dir> ras/conf/ras-wrapper.conf** file and set the `-Dssl.verifyHostName` to **true**. This verifies the hostname.

   For example:

   ```
   wrapper.java.additional.<x>=-Dssl.verifyHostName=true
   ```

5. Import the trusted root certificate authority (CA) to the RAS **client.truststore** file:

   ```
   keytool -importcert -alias <any_alias> -keystore client.truststore -file <certificate_name.cer> -storepass <changeit>
   ```

6. Start RAS.

## *Importing a Certificate to the OOSH Truststore*

If you are using a custom root certificate for Central, you will need to import the trusted root certificate authority (CA) to the OOSH **client.truststore**. If you are using a standard signed root certificate you do not have to perform the following procedure as the certificate will already be in the **client.truststore** file.

By default, HP OO supports all self-signed certificates. However, in a production environment, it is recommended to change this default for security reasons.

Replace the parameters that are highlighted in <yellow>.

**Note:** The following procedure uses the Keytool utility that is located in **<installation**

**dir>/java/bin/keytool**.

1. Stop Central and back up the original **client.truststore** file, located in **<installation dir>/central/var/security/client.truststore**.
2. Edit the **oosh.bat** from **<installation dir>/central/bin**.
3. Set the `-Dssl.support-self-signed` value to **false**. This enables the trusted root certificate authority (CA).

   For example:

   ```
   -Dssl.support-self-signed=false
   ```

4. Set the `-Dssl.verifyHostName` to **true**. This verifies the hostname.

   For example:

   ```
   -Dssl.verifyHostName=true
   ```

5. Import the trusted root certificate authority (CA) to the Central **client.truststore** file:

   ```
   keytool -importcert -alias <any_alias> -keystore client.truststore -file <certificate_name.cer> -storepass <changeit>
   ```

6. Run OOSH.

## *Importing a Certificate to the Studio Debugger Truststore*

After installing Studio, if you are using a custom root certificate for Studio, you will need to import the trusted root certificate authority (CA) to the Studio **client.truststore**. If you are using a standard signed root certificate you do not have to perform the following procedure as the certificate will already be in the **client.truststore** file.

By default, HP OO supports all self-signed certificates. However, in a production environment, it is recommended to change this default for security reasons.

Replace the parameters that are highlighted in <yellow>.

**Note:** The following procedure uses the Keytool utility that is located in **<installation dir>/java/bin/keytool**.

1. Close Studio and back up the original **client.truststore** file, located in **<installation dir>/studio/var/security/client.truststore**.
2. Edit the **Studio.l4j.ini** file from **<installation dir>/studio**.
3. Set the `-Dssl.support-self-signed` value to **false**. This enables the trusted root certificate

authority (CA).

For example:

```
-Dssl.support-self-signed=false
```

4. Set the `-Dssl.verifyHostName` to **true**. This verifies the hostname.

   For example:

   ```
   -Dssl.verifyHostName=true
   ```

5. Import the trusted root certificate authority (CA) to the Studio **client.truststore** file:

   ```
   keytool -importcert -alias <any_alias> -keystore client.truststore -file <ce
   rtificate_name.cer> -storepass <changeit>
   ```

6. Start Studio.

For more information, see "Debugging a Remote Central with Studio" in the *Studio Authoring Guide*.

## *Changing the Keystore/Truststore Password*

- **To change the Central password**:

  a. Edit the **server.xml** file located in **<Installation dir>/central/tomcat/conf/server.xml.**

  b. Locate the HTTPS connector. For example:

  ```
  keyPass="changeit" keystoreFile="C:/Program Files/Hewlett-Packard/HP Opera
  tions Orchestration/central/var/security/key.store" keystorePass="changeit"
  keystoreType="JKS" maxThreads="200" port="8443" protocol="org.apache.coyot
  e.http11.Http11NioProtocol" scheme="https" secure="true" sslProtocol="TLSv
  1.2" truststoreFile="C:/Program Files/Hewlett-Packard/HP Operations Orches
  tration/central/var/security/client.truststore" truststorePass="changeit"
  truststoreType="JKS"/>
  ```

  c. Change the required password.

  - `keyPass` - the password used to access the server certificate from the specified keystore file. The default value is "`changeit`".
  - `keystorePass` - the password used to access the specified keystore file. The default value is the value of the `keyPass` attribute.
  - `truststorePass` - the password to access the trust store. The default is the value of the **javax.net.ssl.trustStorePassword** system property. If that property is null, no truststore password will be configured. If an invalid truststore password is specified, a warning will

be logged and an attempt will be made to access the truststore without a password, which will skip validation of the truststore contents.

d. Save the file.

e. Open **central-wrapper.conf**, under **central/conf** and change:

```
wrapper.java.additional.<x>=-Djavax.net.ssl.trustStorePassword=changeit
```

f. Restart Central.

- **To change the RAS truststore password**: Edit the **ras-wrapper.conf** file and change the `changeit` parameter of the truststore.
- **To change the OOSH truststore password**: Edit the **oosh.bat** file and change the `changeit` parameter of the truststore.
- **To change the Studio truststore password**: Edit the **<Installation dir>/studio/Studio.l4j.ini** file and change the `changeit` parameter of the truststore.

## *Removing the RC4 Cipher from the SSL-supported Ciphers*

The remote host supports the use of the RC4 cipher. This cipher is flawed in its generation of a pseudo-random stream of bytes so that a wide variety of small biases are introduced into the stream, decreasing its randomness.

If plain text is repeatedly encrypted (for example, HTTP cookies), and an attacker is able to obtain many (i.e., tens of millions of) cipher texts, the attacker may be able to derive the plain text.

Disable the RC4 cipher on the JRE level (starting with Java 7):

1. Open the **$JRE_HOME/lib/security/java.security** file.
2. Edit the **jdk.tls.disabledAlgorithms** property to disable the RC4 cipher.

For more information, see http://stackoverflow.com/questions/18589761/restict-cipher-suites-on-jre-level.

## *Changing or Closing the HTTP/HTTPS Ports*

The file **server.xml** under **[OO_HOME]\central\Tomcat\conf** contains two elements named **<Connector>** under the element **<Service>**. These connectors define or enable the ports that the server are listening to.

Each connector configuration is defined through its attributes. The first connector defines a regular HTTP connector and the second defines an HTTPS connector.

By default, the connectors look as follows.

HTTP connector:

```
<Connector URIEncoding="UTF-8" compression="on" connectionTimeout="20000" po
rt="8080" protocol="org.apache.coyote.http11.Http11NioProtocol" redirectPort
="8443"/>
```

HTTPS connector:

```
<Connector SSLEnabled="true" URIEncoding="UTF-8" clientAuth="false" compress
ion="on" keyAlias="tomcat" keyPass="changeit" keystoreFile="C:/Program Files
/Hewlett-Packard/HP Operations Orchestration/central/var/security/key.store"
keystorePass="changeit" keystoreType="JKS" maxThreads="200" port="8443" prot
ocol="org.apache.coyote.http11.Http11NioProtocol" scheme="https" secure="tru
e" sslProtocol="TLSv1.2" truststoreFile="C:/Program Files/Hewlett-Packard/HP
Operations Orchestration/central/var/security/client.truststore" truststoreP
ass="changeit" truststoreType="JKS"/>
```

By default, both are enabled.

### *Changing Port Values*

To change the values of one of the ports:

1. Edit the **server.xml** file located in **<installation_dir>/central/tomcat/conf/server.xml**.
2. Locate the HTTP or HTTPS connector, and adjust the **port** value in the line.

   **Note:** If you are keeping both HTTP and HTTPS active and you want to change the HTTPS port, you will need to change the **redirectPort** for the HTTP connector.

3. Save the file.
4. Restart Central.

### *Disabling a Port*

To disable one of the ports:

1. Edit the **server.xml** file located in **<installation_dir>/central/tomcat/conf/server.xml**.
2. Locate the HTTP or HTTPS connector, and delete or comment out the line.
3. Save the file.
4. Restart Central.

# Client Certificate Authentication (Mutual Authentication)

The most common use of X.509 certificate authentication is in verifying the identity of a server when using SSL/TLS, most commonly when using HTTPS from a browser. The browser automatically checks that the certificate presented by a server has been issued by one of a list of trusted certificate authorities, which it maintains.

You can also use SSL/TLS with mutual authentication. The server requests a valid certificate from the client as part of the SSL/TLS handshake. The server authenticates the client by checking that its certificate is signed by an acceptable authority. If a valid certificate has been provided, it can be obtained through the servlet API in an application.

## Configuring Client Certificate Authentication in Central

Before you configure the client certificate authentication in Central, make sure you have configured the SSL/TLS server certificate, as described in "Server and Client Certificate Authentication" on page 5.

Set the `clientAuth` attribute to `true` if you want the SSL stack to require a valid certificate chain from the client before accepting a connection. Set to `want` if you want the SSL stack to request a client certificate, but not fail if one is not presented. A `false` value (default) does not require a certificate chain unless the client requests a resource protected by a security constraint that uses CLIENT-CERT authentication. (For more information, see the Apache Tomcat Configuration Reference).

Set the **Certificate Revocation List (CRL)** file. This can contain several CRLs.In the operation of some cryptosystems, usually public key infrastructures (PKIs), a certificate revocation list (CRL) is a list of certificates (or more specifically, a list of serial numbers for certificates) that have been revoked, and therefore, entities presenting those (revoked) certificates should no longer be trusted.

**Note:** The following procedure uses the Keytool utility that is located in **<installation dir>/java/bin/keytool**.

1. Stop the Central server.
2. Import the appropriate root certificate (CA) into Central **client.truststore**: **<installation dir>/central/var/security/client.truststore**, for example:

   ```
   keytool -importcert -alias <any_alias> -keystore <path>/client.truststore -f
   ile <certificate_path> -storepass <changeit>
   ```

3. Edit the **server.xml** file located in **<installation_dir>/central/tomcat/conf/server.xml**.
4. Set the `clientAuth` attribute in the `Connector` tag to `want` or `true`. The default is `false`.

   **Note:** We recommend starting the server at the end of this procedure, but note that it is

also possible to start the server at this point.

5. Add the `crlFile` attribute to define the certificate revocation list file for the SSL/TLS certificate validation, for example:

```
crlFile="<path>/crlname.<crl/pem>"
```

The file can be with the `.crl` extension for a single certificate revocation list or with the `.pem` (PEM CRL format) extension for one or more certificate revocation lists. The PEM CRL format uses the following header and footer lines:

```
-----BEGIN X509 CRL-----
-----END X509 CRL-----
```

Example of the `.pem` file structure for one CRL (for more than one, concatenate another CRL block):

```
-----BEGIN X509 CRL-----
MIIBbzCB2QIBATANBgkqhkiG9w0BAQUFADBeMQswCQYDVQQGEwJVUzEYMBYGA1UE
ChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQLEwNEb0QxEDAOBgNVBAsTB1Rlc3Rp
bmcxFTATBgNVBAMTDFRydXN0IEFuY2hvchcNOTkwMTAxMTIwMTAwWhcNNDgwMTAx
MTIwMTAwWjAiMCACAScXDTk5MDEwMTEyMDAwMFowDDAKBgNVHRUEAwoBAaAjMCEw
CgYDVR0UBAMCAQEwEwYDVR0jBAwwCoAIq5rr+cLnVI8wDQYJKoZIhvcNAQEFBQAD
gYEAC7lqZwejJRW7QvzH11/7cYcL3racgMxH3PSU/ufvyLk7ahR++RtHary/WeCv
RdyznLiIOA8ZBiguWtVPqsNysNn7WLofQIVa+/TD3T+lece4e1NwGQvj5Q+e2wRt
GXg+gCuTjTKUFfKRnWz7O7RyiJKKim0jtAF4RkCpLebNChY=
-----END X509 CRL-----
```

6. Start the Central server.

**Note:** For each client certificate, you need to define a user, either an internal user or LDAP user. The name of the user should be defined in the certificate attributes. The default is value of the CN attribute. See the Processing a Certificate Principal section for more details.

Note that even if HP OO is set up with multiple LDAP configurations, it is only possible to authenticate the user using the client certificate attributes with the default LDAP. Central will first try to authenticate the user with the default LDAP, and if this fails, will try to authenticate within the HP OO internal domain.

## *Updating the Configuration of a Client Certificate in RAS*

The client certificate is configured during the installation of the RAS. However, if you need to update the client certificate, you can do this manually in the **ras-wrapper.conf** file.

**Prerequisite**: You must import the CA root certificate of Central into the RAS truststore.. See "Importing a Certificate to a RAS Truststore" on page 7.

To update the configuration of the client certificate in an external RAS:

1. Stop the RAS server.
2. Open the **ras-wrapper.conf** file from <**installation dir>ras/var/conf/ras-wrapper.conf**.
3. Change the following according to the your client certificate:

```
wrapper.java.additional.<x>=-Djavax.net.ssl.keyStore=<installation dir>/var/
security/certificate.p12"

wrapper.java.additional.<x>=-Djavax.net.ssl.keyStorePassword=changeit

wrapper.java.additional.<x>=-Djavax.net.ssl.keyStoreType=PKCS12
```

4. Start the RAS server.

**Important Note!** The X.509 client certificate needs to have the principal name of the RAS, which is the RAS ID (see Processing a Certificate Principal).

You can find the RAS ID under the **Topology** tab in Central. See "Setting Up Topology – Workers" in the *HP OO Central User Guide*.

## *Configuring a Client Certificate in Studio Remote Debugger*

**Prerequisite**: You must import the CA root certificate of Central into the Studio Debugger truststore.. See "Importing a Certificate to the Studio Debugger Truststore" on page 9.

To configure the client certificate in the Studio Remote Debugger:

1. Close Studio.
2. Edit the **Studio.l4j.ini** from **<installation dir>/studio**.
3. Change the following according to the your client certificate:

```
-Djavax.net.ssl.keyStore="<installation dir>/studio/var/security/certificate
.p12"

-Djavax.net.ssl.keyStorePassword=changeit

-Djavax.net.ssl.keyStoreType=PKCS12
```

4. Start Studio.

**Note:** For the client certificate, you need to define a user, either an internal user or LDAP user. The name of the user should be defined in the certificate attributes. The default is value of the CN attribute. See the Processing a Certificate Principal section for more details.

Note that even if HP OO is set up with multiple LDAP configurations, it is only possible to authenticate the user using the client certificate attributes with the default LDAP. Central will

first try to authenticate the user with the default LDAP, and if this fails, will try to authenticate within the HP OO internal domain.

## *Configuring a Client Certificate in OOSH*

**Prerequisite**: You must import the CA root certificate of Central into the OOSH truststore. See "Importing a Certificate to the OOSH Truststore" on page 8.

1. Stop OOSH.
2. Edit the **oosh.bat** from **<installation dir>/central/bin**.
3. Change the following according to the your client certificate:

```
-Djavax.net.ssl.keyStore="<installation dir>/var/security/certificate.p12"
-Djavax.net.ssl.keyStorePassword=changeit
-Djavax.net.ssl.keyStoreType=PKCS12
```

4. Start OOSH.

**Note:** For the client certificate, you need to define a user, either an internal user or LDAP user. The name of the user should be defined in the certificate attributes. The default is value of the CN attribute. See the Processing a Certificate Principal section for more details.

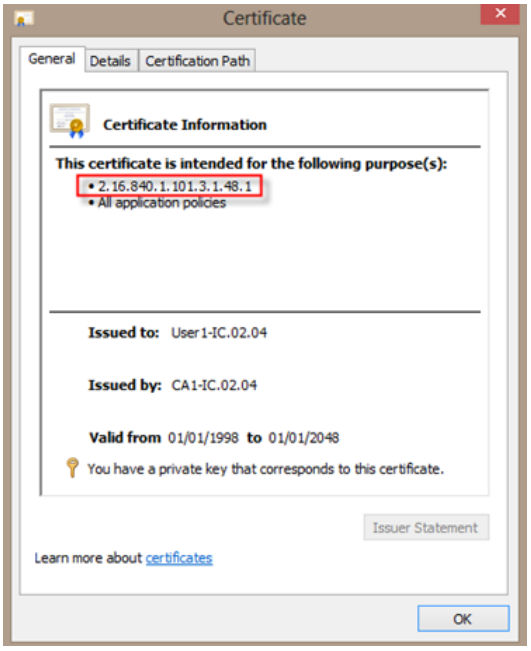
Note that even if HP OO is set up with multiple LDAP configurations, it is only possible to authenticate the user using the client certificate attributes with the default LDAP. Central will first try to authenticate the user with the default LDAP, and if this fails, will try to authenticate within the HP OO internal domain.

## *Processing Certificate Policies*

HP OO handles the processing of certificate polices for the end certificate.

- You can set the purpose string in the certificate.
- HP OO lets you add the policy string(s) as a configuration item and check the policy string of each end certificate. If it does not match, reject the certificate.
- Enable or disable the certificate policy verification by adding the following configuration item: `x509.certificate.policy.enabled=true/false` (default is `false`).
- Define the policy list by adding the following configuration item:

`x509.certificate.policy.list=<comma_separated_list>` (the default is an empty list).




## *Processing a Certificate Principal*

You can define how to get the principal from a certificate using a regular expression match against the `Subject`. The regular expression should contain a single group. The default expression `CN=(.?)` matches the common name field. For example, `CN=Jimi Hendrix, OU=` assigns a user name of `Jimi Hendrix`.

- The matches are case-insensitive.
- The principal of the certificate is the user name in HP OO (LDAP or internal user).
- To change the regular expression, change the configuration item: `x509.subject.principal.regex`.

# *Troubleshooting*

If the server doesn't start, open the **wrapper.log** file and look for an error in `ProtocolHandler ["http-nio-8443"]`.

This can happen when Tomcat is initializing or starting the connector. There are many variations but the error message can provide information.

All the HTTPS connector parameters are in the Tomcat configuration file located at **C:\HP\oo\central\tomcat\conf\server.xml**.

Open the file and scroll to the end, until you see the HTTPS connector:

```
<Connector SSLEnabled="true" clientAuth="false" keyAlias="tomcat" keystoreFile="
C:/HP/oo/central/var/security/keystore.p12" keystorePass="tomcat-keystore-passwo
rd" keystoreType="PKCS12" maxThreads="200" port="8443" protocol="org.apache.coyo
te.http11.Http11NioProtocol" scheme="https" secure="true" sslProtocol="TLSv1.
2"/>
```

See if there is any mismatch in the parameters, by comparing them to the parameters you entered in the previous steps.

# Federal Information Processing Standard (FIPS)

## *Configuring HP OO for FIPS 140-2 Compliance*

This section explains how to configure HP Operations Orchestration to be compliant with Federal Information Processing Standards (FIPS) 140-2.

FIPS 140-2 is a standard for security requirements for cryptographic modules defined by the National Institute of Standards Technology (NIST). To view the publication for this standard, go to: csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf.

After you have configured HP OO for FIPS 140-2 compliance, HP OO uses the following security algorithm:

- Symmetric-key algorithm: AES
- Hashing algorithm: SHA1

HP OO uses the security provider: RSA BSAFE Crypto software version 6.1.This is the only supported security provider for FIPS 140-2.

**Note:** Once you have configured HP OO to be compliant with FIPS 140-2, you cannot revert back to the standard configuration unless you re-install HP OO.

## *Prerequisites*

**Note:** If you are upgrading from an installation of HP OO 10.10 (and later) that was already configured with FIPS, you must repeat steps 4 and 5 below, and then repeat the steps in the "Configure the Properties in the Java Security File" section in "Configuring HP OO to be Compliant with FIPS 140-2" on page 21.

Before configuring HP OO to be compliant with FIPS 140-2, perform the following steps:

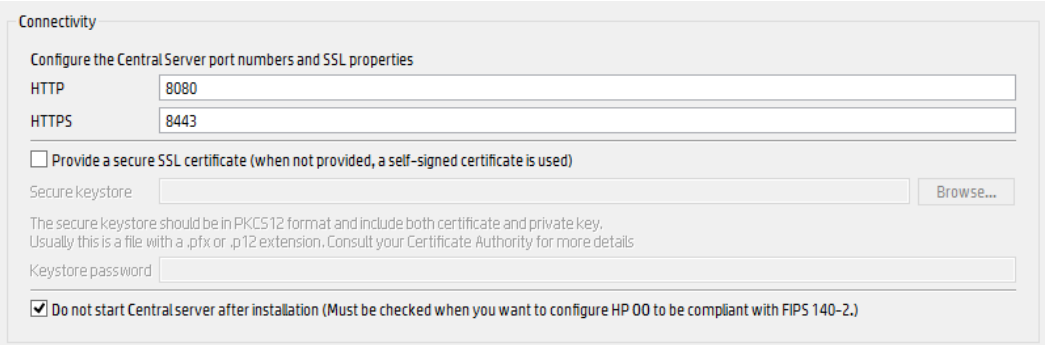
**Note:** In order to be FIPS140-2 compliant, you need to turn off LWSSO.

1. Verify that you are configuring a new installation of HP OO version 10.10 or higher to be compliant with FIPS 140-2, and that it is not in use.

   You cannot configure an installation of HP OO that is in use (whether version 9.x or or 10.x).

2. Verify that when HP OO was installed, it was configured not to start the Central server after installation:

- In a silent installation, the `should.start.central` parameter was set to **no**.
- In a wizard installation, in the **Connectivity** step, the **Do not start Central server after installation** check box was selected.




3. Back up the following directories:
   - **<installation dir>\central\tomcat\webapps\oo.war**
   - **<installation dir>\central\tomcat\webapps\PAS.war**
   - **<installation dir>\central\conf**
   - **<oo_jre>\lib\security** (where **<oo_jre>** is the directory in which the JRE used by HP OO is installed. By default, this is **<installation dir>\java**)
4. Download and install the Java Cryptographic Extension (JCE) Unlimited Strength Jurisdiction Policy Files from the following site: http://www.oracle.com/technetwork/java/javase/downloads/jce-7-download-432124.html.

   **Note:** See the **ReadMe.txt** file from the downloaded content for information on how to deploy the files and upgrade the JRE used by HP OO.
5. Install the RSA BSAFE Crypto software files. On the system on which HP OO is installed, copy the following to **<oo_jre>\lib\ext** (where **<oo_jre>** is the directory in which the JRE that is used by HP OO is installed. By default, this is **<installation dir\java**).
   - **<installation dir>\central\lib\cryptojce-6.1.jar**
   - **<installation dir>\central\lib\cryptojcommon-6.1.jar**
   - **<installation dir>\central\lib\jcmFIPS-6.1.jar**

**Note:** If you are upgrading from an installation of HP OO 10.10 (and later) that was already configured with FIPS, you must repeat steps 4 and 5 from the "Prerequisites" section above, and then repeat the steps in the "Configure the Properties in the Java Security File" section in

"Configuring HP OO to be Compliant with FIPS 140-2" below.

# Configuring HP OO to be Compliant with FIPS 140-2

The following list shows the procedures that you need to perform in order to configure HP OO to be compliant with FIPS 140-2:

- Configure the Properties in the Java Security File
- Configure the encryption.properties File and Enable FIPS Mode
- Create FIPS-Compliant HP OO Encryption
- Replace the Database Password
- Start HP OO

## Configure the Properties in the Java Security File

Edit the Java security file for the JRE to add additional security providers and configure the properties for FIPS 140-2 compliance.

**Note:** The upgrade to HP OO 10.10 completely replaces the installed JRE files. Therefore, the following steps must be done after upgrading to 10.10.

**Note:** If you are upgrading from an installation of HP OO 10.10 (and later) that was already configured with FIPS, you must repeat steps 4 and 5 from the "Prerequisites" section in " Federal Information Processing Standard (FIPS)" on page 19, and then repeat the steps here.

Open the **<oo_jre>\lib\security\java.security** file in an editor and perform the following steps:

1. For every provider listed, in the format **security.provider.<nn>=<provider_name>**, increment the preference order number `<nn>` by two.

   For example, change a provider entry from:

   ```
   security.provider.1=sun.security.provider.Sun
   ```

   to

   ```
   security.provider.3=sun.security.provider.Sun
   ```

2. Add a new default provider (RSA JCE). Add the following provider at the top of the provider list:

   ```
   security.provider.1=com.rsa.jsafe.provider.JsafeJCE
   ```

3. Add the RSA BSAFE SSL-J Java Secure Sockets Extension (JSSE) Provider.

   ```
   security.provider.2=com.rsa.jsse.JsseProvider
   ```

4. Copy and paste the following line into the **java.security** file to ensure **RSA BSAFE** is used in FIPS 140-2 compliant mode:

   ```
   com.rsa.cryptoj.fips140initialmode=FIPS140_SSL_MODE
   ```

   You can paste this line anywhere in the **java.security** file.

5. Because the default DRBG algorithm ECDRBG128 is not safe (according to NIST), set the security property **com.rsa.crypto.default** to **HMACDRBG**, by copying the following line into the **java.security** file:

   ```
   com.rsa.crypto.default.random=HMACDRBG
   ```

   You can paste this line anywhere in the **java.security** file.

6. Save and exit the **java.security** file.

## *Configure the encryption.properties File and Enable FIPS Mode*

The HP OO encryption properties file must be updated to be FIPS 140-2 compliant.

1. Back up the **encryption.properties** file, which is located in **<installation dir>\central\var\security**.

2. Open the **encryption.properties** file in a text editor. For example, edit the following file:

   **C:\Program Files\Hewlett-Packard\HP Operations Orchestration\central\var\security\encryption.properties.**

3. Locate `keySize=128` and replace it with `keySize=256`.

4. Locate `secureHashAlgorithm=SHA1` and replace it with `secureHashAlgorithm=SHA256`.

5. Locate `FIPS140ModeEnabled=false` and replace it with `FIPS140ModeEnabled=true`.

   **Note:** If `FIPS140ModeEnabled=false` does not exist, add `FIPS140ModeEnabled=true` as a new line to the end of the file.

6. Save and close the file.

## *Create FIPS-Compliant HP OO Encryption*

To create or replace the HP OO encryption store file, so that it is FIPS compliant, see "Replacing the FIPS Encryption" on the next page.

**Note:** AES has three approved key lengths: 128/192/256 by NIST SP800-131A publication.

These secure hash algorithms are supported in FIPS: SHA1, SHA256, SHA384, SHA512.

**Note:** It is recommended to change the passwords of the key.store (and its private key entry) and the truststore. See "Changing the Keystore/Truststore Password" on page 10.

**Note:** It is recommended to delete all the default CA root certificates from the HP OO truststore. (The client.truststore is located at <installation>/central/var/security.)

### *Replace the Database Password*

Replace the database password, as described in "Changing the Database Password" on page 27.

### *Start HP OO*

Start HP OO, as described in the *HP OO Installation Guide*.

## *Replacing the FIPS Encryption*

HP OO, Central, and RAS comply with Federal Information Processing Standard 140-2 (FIPS 140-2), which defines the technical requirements to be used by federal agencies when these organizations specify cryptographic-based security systems for protection of sensitive or valuable data.

After a fresh installation of HP OO 10.10, you have the option to change the FIPS encryption algorithm.

**Note:** This procedure is only for fresh installations. You cannot perform it after an upgrade.

### *Changing the FIPS Encryption Algorithm on Central*

1. Go to **<Central installation folder>/var/security**.
2. Back up and delete the **encryption_repository** file.
3. Go to **<Central installation folder>/bin**.
4. Run the **generate-keys** script.

   A new master key is generated in **<Central installation folder>/var/security/encryption_repository**.

### *Changing the RAS Encryption Properties*

If the installation of the RAS is in a new location, you need to complete all the steps below.

**Note:** These changes are only valid if you working on a new RAS installation after you have changed the Central encryption properties.

To change the RAS encryption properties:

1. Complete all the steps in the "Prerequisites" section in " Federal Information Processing Standard (FIPS)" on page 19.
2. Complete all the steps in the "Configure the Properties in the Java Security File" in "Configuring HP OO to be Compliant with FIPS 140-2" on page 21.
3. Copy the current **encryption.properties** file from `<installation dir>\ras\var\security` to `<installation dir>\ras\bin` folder.
4. Using any text editor, edit and change the **encryption.properties** file as required.

   For more information, see "Configure the encryption.properties File and Enable FIPS Mode" in "Configuring HP OO to be Compliant with FIPS 140-2" on page 21.
5. Save the changes.
6. Open a command line prompt in the folder `<installation dir>\ras\bin`.
7. Run **oosh.bat**.
8. Run the OOShell command: `replace-encryption --file encryption.properties`

   **Note:** If you copied the **encryption.properties** file to a different folder, make sure you enter the correct location in the OOShell command.
9. Restart the RAS service.

# Configuring LWSSO Settings

When you install HP OO 10.10, if you choose to upgrade the LWSSO settings from HP OO 9.x, these LWSSO settings will be migrated, but LWSSO will be disabled in HP OO 10.10, even if it was previously enabled in HP OO 9.x.

When you enable LWSSO afterward, you may receive warnings under certain scenarios. To clear the warnings from the log, follow the steps below to set the management URL property using the fully qualified domain name.

- When Central and a RAS are installed on the same machine, and the LWSSO settings are enabled, you must set the management URL property using the fully qualified domain name.

  a. Stop the RAS process.

  b. In the **ras/conf/ras-wrapper.conf** file, change

  ```
  wrapper.java.additional.<x>=-Dmgmt.url=<protocol>://localhost:<port>/oo
  ```

  to

  ```
  wrapper.java.additional.<x>=-Dmgmt.url=<protocol>://<FullyQualifiedDomainName>:<port>/oo
  ```

  c. Start the RAS process.

- When the RAS is installed on a different machine from the Central and LWSSO settings are enabled, you must specify the management URL of the Central with the fully qualified domain name during the RAS installation, instead of the IP address.

- When connecting another application to Central through LWSSO, you must specify the management URL of the Central with the fully qualified domain name.

  a. Stop the Central process.

  b. In the **central/conf/central-wrapper.conf** file, change

  ```
  wrapper.java.additional.<x>=-Dmgmt.url=<protocol>://localhost:<port>/oo
  ```

  to

  ```
  wrapper.java.additional.<x>=-Dmgmt.url=<protocol>://<FullyQualifiedDomainName>:<port>/oo
  ```

  c. Start the Central process.

# Configuring the XSS Policy

HP OO has XSS protection with AntiSamy. The default protection policy is "antisamy", which allows most HTML elements, and may be useful if users are submitting full HTML pages.

This policy is configurable to one of the supported policies by AntiSamy. For more information, see

https://www.owasp.org/index.php/Category:OWASP_AntiSamy_Project#Stage_2_-_Choosing_a_base_policy_file

The policy is configurable via a system configuration property called `xss.policy`. Possible values include: antisamy (the default), antisamy-slashdot, antisamy-myspace, antisamy-ebay, antisamy-anythinggoes, antisamy-tinymc.

To check which policy is configured, go to **https://host/oo/reports/sysinfo** and look for the parameter `xss.policy` in the **system configuration** section.

The simplest way to change the default Slashdot policy is using the HP Operations Orchestration Shell utility.

1. Double-click the `oosh.bat` batch file, to start the OOSH utility.
2. In the command line, type, for example:

```
ssc --url https://host/oo --key  xss.policy --value antisamy-anythinggoes
```

For more information about the HP Operations Orchestration Shell utility, see the *HP Operations Orchestration Shell User Guide*.

# Configuring Localization

## *Setting the System Locale in Central-wrapper.conf*

If your HP OO system is localized , you will need to set the following properties to reflect the system locale, in the **central-wrapper.conf** file:

```
set.LANG=

set.LC_ALL=

set.LANGUAGE=

wrapper.java.additional.<x>=-Duser.language=

wrapper.java.additional.<x>=-Duser.country=
```

For example, for Japanese: `set.LANG=ja_JP` and `set.LC_ALL=ja_JP`

# Configuring the System

## Changing the Database Password

1. If Central is up and running, stop the Central Service.
2. Run the encrypt-password script with the -e -p <password> option, where password is the database password.
3. Copy the result it should appear as follows:

   ```
   ${ENCRYPTED}<some_chars>.
   ```

4. Go to the folder **<Central installation folder>/conf**, and open the **database.properties** file.
5. Change the `db.password` value to the one that you copied.

## Changing the Database IP

This section is relevant when you need to configure HP OO to work with another database instance. All the database parameter such as database credentials, schema name, tables, and so on, should be identical.

1. Edit the file **\HP Operations Orchestration\central\conf\database.properties.**
2. Look for the `jdbc.url` parameter. For example:

   ```
   jdbc.url=jdbc\:jtds\:sqlserver\://16.60.185.109\:1433/schemaName;sendStringP
   arametersAsUnicode\=true
   ```

3. Change the IP address\FQDN of the database server.
4. Save the file.
5. Restart Central.

## Adjusting the Logging Levels

It is possible to adjust the granularity of the information that is provided in the log, separately for regular logging, deployment, and execution.

The granularity options are:

- `INFO` - Default logging information
- `DEBUG` - More logging information

- `ERROR/WARNING` - Less logging information

To adjust the granularity in the logging:

1. Open the **log4j.properties** file (under **/<oo-installation>/central/conf/log4j.properties**).
2. Replace `INFO` with `DEBUG` or `ERROR/WARNING` in the following place in the **log4j.properties** file.

   For example:

```
log.level=INFO
execution.log.level=DEBUG
deployment.log.level=DEBUG
```

## *Adjusting the Timing of Quartz Jobs*

In the HP OO system, quartz jobs run periodically for maintenance of the system.

Each job runs for a set amount of time, and is repeated at set intervals. Following are examples of job triggers:

| Trigger Name | Current Repeat Interval | What Happens |
|---|---|---|
| **onRolling:OO_EXECUTION_ STATES_Trigger** | 4.5 minutes | Rolling the states table for purging |
| **queueCleanerTrigger** | 1 minute | Purging the queue tables |
| **queueRecoveryTrigger** | 2 minutes | Checks if the system needs recovery |
| **recoveryVersionTrigger** | 0.5 minute | Version counter to be used for the recovery |
| **splitJoinTrigger** | 1 second | Joins finished splits |
| **onRolling:OO_EXECUTION_ EVENTS_Trigger**<br>**Note:** This trigger is deprecated. | 12 hours | Rolling the events table for purging |

If you want to tweak these job timings in order to improve performance, perform the following:

**Note:** Any change to the timings can have a major effect on the system. Consult with your HP service representative before making any changes to these triggers.

1. Enter the Jminix page using the URL: **{OO_HOST}:{OO_PORT}/oo/jminix/**

   **Note:** You need **Manage System Settings** permission in order to enter **jminix**.

2. Open the OO tab. Under **MBeans**, there is an operation named **jobTriggersMBean**.

3. Use this operation, and enter the values on the right tab, using the name of the trigger you want to change. Use the exact same name as the table, with a new value for the repeat interval.

This changes the triggering times of the job.

**Note:** The events persistency mechanism is deprecated (see **onRolling:OO_EXECUTION_EVENTS_Trigger**). You can configure this job if you use the Remote Debugger or if you turned on the **events.persistency** flag. See "Turning on the Event Log Mechanism" below.

## *Changing the URL of a Central/Load Balancer on the RAS Side*

While the best practice is to configure the URL of a Central/load balancer via the installer, if you need to make changes to the URL after the RAS was already installed, you can do this by editing the **ras-wrapper.conf** file.

For example, this would be needed if you installed a RAS against a Central/load balancer and the Central/load balancer's FQDN changed. You will need to change Central/load balancer's URL stored at the RAS level, so that the RAS is able to communicate again with the Central/load balancer.

1. Stop the RAS.

2. Open the **ras-wrapper.conf** file, located under **<installation folder>\ras\conf**.

3. Edit the URL in the following line:

   ```
   wrapper.java.additional.<x>=-Dmgmt.url=http://localhost:8080/oo
   ```

4. Restart the RAS.

## *Turning on the Event Log Mechanism*

The event log mechanism was deprecated in HP OO 10.10 and will be removed in a future release.

HP OO is deployed without the event log mechanism. However, if you want to use this mechanism, you can turn on the **events.persistency** flag. Note that for clustering scenarios and performance boost, we recommend leaving this flag off.

To turn on this flag:

1. Stop the process and update the **wrapper.conf** file on every node (Central/RAS) in your system.

   On Central, go to **<installation_path>\central\conf\central-wrapper.conf**

   On RAS, go to **<installation_path>\ras\conf\ras-wrapper.conf**

2. In the **wrapper.conf** file, look for the parameter **-Devents.persistency** and change its value to **true**.

3. Restart the process.


Java™
COMPATIBLE