HP Operations Orchestration

适用于 Windows 和 Linux

软件版本： 10.10

系统配置与强化指南

文档发布日期： 2014 年 5 月

软件发布日期： 2014 年 5 月

法律声明

担保

HP 产品和服务的唯一担保已在此类产品和服务随附的明示担保声明中提出。此处的任何内容均不构成额外担保。HP 不会为此处出现的技术或编辑错误或遗漏承担任何责任。

此处所含信息如有更改，恕不另行通知。

受限权利声明

机密计算机软件。必须拥有 HP 授予的有效许可证，方可拥有、使用或复制本软件。按照 FAR 12.211 和 12.212，并根据供应商的标准商业许可的规定，商业计算机软件、计算机软件文档与商品技术数据授权给美国政府使用。

版权声明

商标声明

Adobe™ 是 Adobe Systems Incorporated 的商标。

此产品包含“zlib”通用压缩库的接口，版权所有© 1995-2002 Jean-loup Gailly and Mark Adler。

AMD 及 AMD 箭头符号是 Advanced Micro Devices, Inc. 的商标

Google™ 和 Google Maps™ 是 Google Inc. 的商标

Intel®、Itanium®、Pentium® 和 Intel® Xeon® 是 Intel Corporation 在美国及其他国家/地区的商标。

Java 是 Oracle 和/或其附属公司的注册商标。

Microsoft®、Windows®、Windows NT®、Windows® XP 和 Windows Vista® 是 Microsoft Corporation 在美国的注册商标。

Oracle 是 Oracle Corporation 和/或其附属公司的注册商标。

UNIX® 是 The Open Group 的注册商标。

文档更新

此文档的标题页包含以下标识信息：

- 软件版本号，用于指示软件版本。
- 文档发布日期，该日期将在每次更新文档时更改。
- 软件发布日期，用于指示该版本软件的发布日期。

要检查是否有最新的更新，或者验证是否正在使用最新版本的文档，请访问：**http://h20230.www2.hp.com/selfsolve/manuals**

需要注册 HP Passport 才能登录此站点。要注册 HP Passport ID，请访问：**http://h20229.www2.hp.com/passport-registration.html**

或单击“HP Passport”登录页面上的“**New users - please register**”链接。

此外，如果订阅了相应的产品支持服务，则还会收到更新的版本或新版本。有关详细信息，请与您的 HP 销售代表联系。

支持

请访问 HP 软件联机支持网站：**http://www.hp.com/go/hpsoftwaresupport**

此网站提供了联系信息，以及有关 HP 软件提供的产品、服务和支持的详细信息。

HP 软件联机支持提供客户自助解决功能。通过该联机支持，可快速高效地访问用于管理业务的各种交互式技术支持工具。作为尊贵的支持客户，您可以通过该支持网站获得下列支持：

- 搜索感兴趣的知识文档
- 提交并跟踪支持案例和改进请求
- 下载软件修补程序
- 管理支持合同
- 查找 HP 支持联系人
- 查看有关可用服务的信息
- 参与其他软件客户的讨论
- 研究和注册软件培训

大多数提供支持的区域都要求您注册为 HP Passport 用户再登录，很多区域还要求用户提供支持合同。要注册 HP Passport ID，请访问：

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

要查找有关访问级别的详细信息，请访问：

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

HP Software Solutions Now 可访问 HPSW 解决方案和集成门户网站。此网站将帮助您寻找可满足您业务需求的 HP 产品解决方案，包括 HP 产品之间的集成的完整列表以及 ITIL 流程的列表。此网站的 URL 为 **http://h20230.www2.hp.com/sc/solutions/index.jsp**

目录

系统配置与强化

本文档介绍如何配置和强化 HP Operations Orchestration。

服务器和客户端证书身份验证

安全套接字层 (SSL)/传输层安全性 (TLS) 证书采用数字绑定方式将加密密钥绑定到组织的详细信息，从而确保 Web 服务器到浏览器的连接安全。

HP OO 使用 Keytool 实用程序管理加密密钥和受信任的证书。HP OO 安装文件夹中包含此实用程序，它位于 **<安装目录>/java/bin/keytool** 中。有关 Keytool 实用程序的详细信息，请访问 http://docs.oracle.com/javase/7/docs/technotes/tools/solaris/keytool.html。

HP OO Central 的安装包含两个证书管理文件：

- **<安装目录>/central/var/security/client.truststore**：包含受信任证书的列表。
- **<安装目录>/central/var/security/key.store**：包含 HP OO 证书。

在安装新的 HP OO 后或者如果当前证书已过期，建议您替换 HP OO 证书。

服务器证书身份验证

替换 Central SSL/TLS 服务器证书

您可以使用由知名企业签名的证书，也可以自定义服务器证书。

替换用 <黄色> 突出显示的参数以在计算机上匹配 **key.store** 文件的位置和其他详细信息。

> **备注：**以下过程使用位于 **<安装目录>/java/bin/keytool** 中的 Keytool 实用程序。

1. 停止 Central 并备份位于 **<安装目录>/central/var/security/key.store** 中的原始 **key.store** 文件。
2. 在 **<安装目录>/central/var/security** 中打开命令行。
3. 使用以下命令从 Central **key.store** 文件中删除现有服务器证书：

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

4. 如果您的证书已经带有 **.pfx** 或 **.p12** 扩展名，则转至下一步。如果没有，则需要将证书与私钥导出为 PKCS12 格式 (.pfx,.p12)。例如，如果证书格式为 PEM：

```
>openssl pkcs12 -export -in <cert.pem> -inkey <key.key> -out <证书名称>.p12 -name <名称>
```

如果证书格式为 DER，请在 pkcs12 后添加 -inform DER 参数。例如：

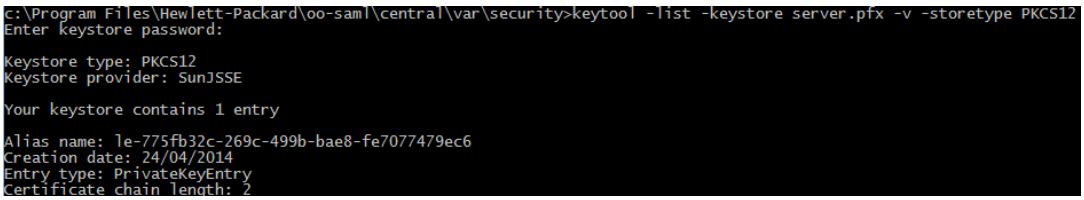
```
>openssl pkcs12 -inform DER -export -in <cert.pem> -inkey <.key> -out <证书名称>.p12 -name <名称>
```

备注：记下您提供的密码。稍后在此过程中输入 keystore 密码时将需要此私钥的密码。

5. 使用以下命令提取证书的别名：

   ```
   keytool -list -keystore <证书名称> -v -storetype PKCS12
   ```

   此时将显示别名。在下例中，别名位于倒数第四行。



6. 将 PKCS12 格式的服务器证书导入 Central **key.store** 文件：

   keytool -importkeystore -srckeystore <PKCS12 格式证书路径> -destkeystore key.store -srcstoretype pkcs12 -deststoretype JKS -alias <证书别名> -destalias tomcat

7. 建议更改 Central 服务器中自动生成的 keystore 的默认“changeit”密码。请参阅更改 Keystore/信任库密码 (第 10 页)。

8. 启动 Central。

使用自签名证书替换 Central SSL/TLS 服务器证书

可以使用 Keytool 实用程序生成自签名证书。

备注：在升级到 HP OO 10.10 之后：

- 如果将新的 Central 安装在与先前安装相同的计算机上，则可使用现有自签名证书。
- 如果将新的 Central 安装在其他计算机上，则需要为每个 Central 生成新的自签名证书(即使先前版本已拥有证书)。

备注：以下过程使用位于 **<安装目录>/java/bin/keytool** 中的 Keytool 实用程序。

替换用 <黄色> 突出显示的参数以在计算机上匹配 **key.store** 文件的位置和其他详细信息。

1. 停止 Central 并备份位于 **<安装目录>/central/var/security/key.store** 中的原始 **key.store** 文件。
2. 在 **<安装目录>/central/var/security** 中打开命令行。
3. 使用以下命令从 Central **key.store** 文件中删除现有服务器证书：

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

4. 生成自签名证书：

   ```
   keytool -genkey -alias tomcat -keyalg RSA -keypass <changeit >-keystore <path/for/new/Keystore> -storepass <changeit>-storetype pkcs12 -dname "CN=<CENTRAL FQDN>, OU=<组织单元>, O=<组织>, L=<位置>, C=<国家/地区>"
   ```

   备注：如果没有输入用于生成新 keystore 的路径，则会在您输入命令的文件夹中创建路径，例如 **<安装目录>/central/var/security**。

5. 将自签名证书导入 Central **key.store** 文件：

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

6. 启动 Central。

将证书导入 RAS 信任库

在安装 RAS 后，如果针对 Central 使用自定义根证书，但是在安装 RAS 期间没有提供此根证书，则需要将受信任的根证书颁发机构 (CA) 导入到 RAS **client.truststore**。如果使用标准签署的根证书，则不需要执行以下过程，因为证书已经在 **client.truststore** 文件中。

默认情况下，HP OO 支持所有自签名证书。但是，在生产环境中，出于安全考虑，建议更改此默认设置。

替换用 <黄色> 突出显示的参数。

备注：以下过程使用位于 **<安装目录>/java/bin/keytool** 中的 Keytool 实用程序。

1. 停止 RAS 并备份位于 **<安装目录>/ras/var/security/client.truststore** 中的原始 **client.truststore** 文件。
2. 在 **<安装目录>/ras/var/security** 中打开命令行。

3. 打开 **<安装目录> ras/conf/ras-wrapper.conf** 文件并将 `-Dssl.support-self-signed` 值设置为 **false**。这将启用受信任的根证书颁发机构 (CA)。

   例如：

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

4. 打开 **<安装目录> ras/conf/ras-wrapper.conf** 文件并将 `-Dssl.verifyHostName` 设置为 **true**。这将验证主机名。

   例如：

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

5. 将受信任的根证书颁发机构 (CA) 导入 RAS **client.truststore** 文件：

   ```
   keytool -importcert -alias <任何别名> -keystore client.truststore -file <证书名称.cer> -storepass <changeit>
   ```

6. 启动 RAS。

将证书导入 OOSH 信任库

如果针对 Central 使用自定义根证书，则需要将受信任的根证书颁发机构 (CA) 导入 OOSH **client.truststore** 中。如果使用标准签署的根证书，则不需要执行以下过程，因为证书已经在 **client.truststore** 文件中。

默认情况下，HP OO 支持所有自签名证书。但是，在生产环境中，出于安全考虑，建议更改此默认设置。

替换用 <黄色> 突出显示的参数。

> **备注：**以下过程使用位于 **<安装目录>/java/bin/keytool** 中的 Keytool 实用程序。

1. 停止 Central 并备份位于 **<安装目录>/central/var/security/client.truststore** 中的原始 **client.truststore** 文件。
2. 编辑 **<安装目录>/central/bin** 中的 **oosh.bat**。
3. 将 `-Dssl.support-self-signed` 值设置为 **false**。这将启用受信任的根证书颁发机构 (CA)。

   例如：

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

4. 将 `-Dssl.verifyHostName` 设置为 **true**。这将验证主机名。

例如：

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

5. 将受信任的根证书颁发机构 (CA) 导入 Central **client.truststore** 文件：

```
keytool -importcert -alias <任何别名> -keystore client.truststore -file <证书名称.cer> -storepass <changeit>
```

6. 运行 OOSH。

将证书导入 Studio Debugger 信任库

在安装 Studio 后，如果针对 Studio 使用自定义根证书，则需要将受信任的根证书颁发机构 (CA) 导入 Studio **client.truststore** 中。如果使用标准签署的根证书，则不需要执行以下过程，因为证书已经在 **client.truststore** 文件中。

默认情况下，HP OO 支持所有自签名证书。但是，在生产环境中，出于安全考虑，建议更改此默认设置。

替换用 **<黄色>** 突出显示的参数。

> **备注：**以下过程使用位于 **<安装目录>/java/bin/keytool** 中的 Keytool 实用程序。

1. 关闭 Studio 并备份位于 **<安装目录>/studio/var/security/client.truststore** 中的原始 **client.truststore** 文件。
2. 编辑 **<安装目录>/studio** 中的 **Studio.l4j.ini** 文件。
3. 将 `-Dssl.support-self-signed` 值设置为 **false**。这将启用受信任的根证书颁发机构 (CA)。

   例如：

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

4. 将 `-Dssl.verifyHostName` 设置为 **true**。这将验证主机名。

   例如：

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

5. 将受信任的根证书颁发机构 (CA) 导入 Studio **client.truststore** 文件：

   ```
   keytool -importcert -alias <任何别名> -keystore client.truststore -file <证书名称.cer> -storepass <changeit>
   ```

6. 启动 Studio。

有关详细信息，请参阅《Studio 创建指南》中的“使用 Studio 调试远程 Central”。

更改 Keystore/信任库密码

- **要更改 Central 密码，请执行以下操作：**

  a. 编辑位于 **<安装目录>/central/tomcat/conf/server.xml** 中的 **server.xml** 文件。

  b. 找到 HTTPS 连接器。例如：

  ```
  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. 更改所需的密码。

  - keyPass - 用于从指定的 keystore 文件访问服务器证书的密码。默认值为“changeit”。
  - keystorePass - 用于访问指定的 keystore 文件的密码。默认值是 keyPass 属性的值。
  - truststorePass - 用于访问信任库的密码。默认值是 **javax.net.ssl.trustStorePassword** 系统属性的值。如果该属性为 null，则不配置信任库密码。如果指定了无效的信任库密码，则将记录警告并尝试不用密码访问信任库，这样将跳过对信任库内容的验证。

  d. 保存文件。

  e. 打开 **central/conf** 中的 **central-wrapper.conf**，并更改：

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

  f. 重新启动 Central。

- **要更改 RAS 信任库密码，请执行以下操作：**编辑 **ras-wrapper.conf** 文件，并更改信任库的 changeit 密码。
- **要更改 OOSH 信任库密码，请执行以下操作：**编辑 **oosh.bat** 文件，并更改信任库的 changeit 参数。
- **要更改 Studio 信任库密码，请执行以下操作：**编辑 **<安装目录>/studio/Studio.l4j.ini** 文件并更改信任库的 changeit 参数。

从支持 SSL 的密码中删除 RC4 密码

远程主机支持使用 RC4 密码。此密码在生成伪随机字节流时有缺陷，因此流中包含各种小偏差，从而降低了随机性。

如果纯文本重复加密(例如 HTTP cookie)，并且攻击者能够获得许多(即数千万)密码文本，则攻击者可能会派生纯文本。

禁用 JRE 级别 RC4 密码(从 Java 7 开始)：

1. 打开 **$JRE_HOME/lib/security/java.security** 文件。
2. 编辑 **jdk.tls.disabledAlgorithms** 属性以禁用 RC4 密码。

有关详细信息，请访问 http://stackoverflow.com/questions/18589761/restict-cipher-suites-on-jre-level。

更改或关闭 HTTP/HTTPS 端口

在 **[OO 主目录]\central\Tomcat\conf** 下的 **server.xml** 文件中，**<服务>** 元素下包含两个名为 **<连接器>** 的元素。这些连接器定义或启用服务器侦听的端口。

通过连接器属性定义每个连接器的配置。第一个连接器定义常规 HTTP 连接器，第二个连接器则定义 HTTPS 连接器。

默认情况下，连接器类似于如下内容。

HTTP 连接器：

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

HTTPS 连接器：

```
<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"/>
```

默认情况下，同时启用这两种连接器。

更改端口值

要更改其中一个端口的值，请执行以下操作：

1. 编辑位于 **<安装目录>/central/tomcat/conf/server.xml** 中的 **server.xml** 文件。
2. 找到 HTTP 或 HTTPS 连接器，并调整行中的 **port** 值。

   备注: 如果将 HTTP 和 HTTPS 保留为活动状态，并且想更改 HTTPS 端口，则将需要更改 HTTP 连接器的 **redirectPort**。

3. 保存文件。
4. 重新启动 Central。

禁用端口

要禁用其中一个端口，请执行以下操作：

1. 编辑位于 **<安装目录>/central/tomcat/conf/server.xml** 中的 **server.xml** 文件。
2. 找到 HTTP 或 HTTPS 连接器，并删除或注释掉行。
3. 保存文件。
4. 重新启动 Central。

客户端证书身份验证(相互身份验证)

X.509 证书身份验证的最常见用途是在使用 SSL/TLS 时验证服务器的标识，通常是在浏览器中使用 HTTPS 时验证服务器的标识。浏览器会自动检查其维护的受信任证书颁发机构列表中的机构是否已经颁发了服务器显示的证书。

您也可以使用 SSL/TLS 进行相互身份验证。服务器将请求来自客户端的有效证书以用作 SSL/TLS 握手协议的一部分。服务器通过检查客户端证书是否由可接受的颁发机构签名来对客户端进行身份验证。如果提供了有效证书，则可以通过应用程序中的 servlet API 获得该证书。

在 Central 中配置客户端证书身份验证

在 Central 中配置客户端证书身份验证之前，请确保按服务器和客户端证书身份验证 (第 5 页)中所述配置了 SSL/TLS 服务器证书。

如果希望 SSL 堆栈在接受连接前要求从客户端获得有效证书链，请将 `clientAuth` 属性设置为 `true`。如果希望 SSL 堆栈请求客户端证书但在未提供证书时不失败，则设置为 `want`。`false` 值(默认值)不要求证书链，除非客户端请求了受使用 CLIENT-CERT 身份验证的安全约束保护的资源。(有关详细信息，请参阅《Apache Tomcat Configuration Reference》。)

设置“证书吊销列表 (CRL)”文件。这包含多个 CRL。在部分加密系统(通常为公钥基础结构 (PKI))的操作中，证书吊销列表 (CRL) 是已吊销的证书列表(更具体地说，是证书序列号的列表)，因此应当不再信任表示这些(已吊销)证书的实体。

备注: 以下过程使用位于 **<安装目录>/java/bin/keytool** 中的 Keytool 实用程序。

1. 停止 Central 服务器。
2. 将相应的根证书 (CA) 导入到 Central **client.truststore**：**<安装目录>/central/var/security/client.truststore**，例如：

   ```
   keytool -importcert -alias <任何别名> -keystore <path>/client.truststore -file <证书路径> -storepass <changeit>
   ```

3. 编辑位于 **<安装目录>/central/tomcat/conf/server.xml** 中的 **server.xml** 文件。
4. 将 `Connector` 标记中的 `clientAuth` 属性设置为 `want` 或 `true`。默认值为 `false`。

   备注: 建议在此过程结束时启动服务器，但是请注意，还可以在此时启动服务器。

5. 添加 `crlFile` 属性以定义用于 SSL/TLS 证书验证的证书吊销列表文件，例如：

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

   文件可以带 `.crl` 扩展名以表示单个证书吊销列表，也可以带 `.pem`(PEM CRL 格式) 扩展名以表示一个或多个证书吊销列表。PEM CRL 格式使用以下页眉和页脚行：

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

   一个 CRL 的 `.pem` 文件结构示例(至于多个 CRL，需连接其他 CRL 块)：

   ```
   -----BEGIN X509 CRL-----
   MIIBbzCB2QIBATANBgkqhkiG9w0BAQUFADBeMQswCQYDVQQGEwJVUzEYMBYGA1UE
   ChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQLEwNEb0QxEDAOBgNVBAsTB1Rlc3Rp
   bmcxFTATBgNVBAMTDFRydXN0IEFuY2hvchcNOTkwMTAxMTIwMTAwWhcNNDgwMTAx
   MTIwMTAwWjAiMCACASCXDTk5MDEwMTEyMDAwMFowDDAKBgNVHRUEAwoBAaAjMCEw
   CgYDVR0UBAMCAQEwEwYDVR0jBAwwCoAIq5rr+cLnVI8wDQYJKoZIhvcNAQEFBQAD
   gYEAC7lqZwejJRW7QvzH11/7cYcL3racgMxH3PSU/ufvyLk7ahR++RtHary/WeCv
   RdyznLiIOA8ZBiguWtVPqsNysNn7WLofQIVa+/TD3T+lece4e1NwGQvj5Q+e2wRt
   GXg+gCuTjTKUFfKRnWz7O7RyiJKKim0jtAF4RkCpLebNChY=
   -----END X509 CRL-----
   ```

6. 启动 Central 服务器。

备注: 对于每个客户端证书，您需要定义用户(内部用户或 LDAP 用户)。用户的名称应该在证书属性中定义。默认为 CN 属性的值。有关更多详细信息，请参阅处理证书主体。

请注意，即使已将 HP OO 设置为具有多个 LDAP 配置，仍然只能使用具有默认 LDAP 的客户端证书属性对用户进行身份验证。Central 将首先尝试使用默认的

LDAP 对用户进行身份验证，如果失败，则将尝试在 HP OO 内部域中进行身份验证。

更新 RAS 中客户端证书的配置

客户端证书是在 RAS 安装期间配置的。但是，如果需要更新客户端证书，则可以在 **ras-wrapper.conf** 文件中手动执行此操作。

先决条件：必须将 Central 的 CA 根证书导入 RAS 信任库。请参阅将证书导入 RAS 信任库 (第 7 页)。

要在外部 RAS 中更新客户端证书的配置，请执行以下操作：

1. 停止 RAS 服务器。
2. 在 **<安装目录>ras/var/conf/ras-wrapper.conf** 中打开 **ras-wrapper.conf** 文件。
3. 根据您的客户端证书更改以下内容：

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

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

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

4. 启动 RAS 服务器。

重要说明！ X.509 客户端证书需要具有 RAS 的主体名称，即 RAS ID(请参阅处理证书主体)。

可以在 Central 的“拓扑”选项卡下找到 RAS ID。请参阅《HP OO Central 用户指南》中的“设置拓扑 – 工作程序”。

在 Studio Remote Debugger 中配置客户端证书

先决条件：必须将 Central 的 CA 根证书导入 Studio Debugger 信任库。请参阅将证书导入 Studio Debugger 信任库 (第 9 页)。

要在 Studio Remote Debugger 中配置客户端证书，请执行以下操作：

1. 关闭 Studio。
2. 编辑 **<安装目录>/studio** 中的 **Studio.l4j.ini**。
3. 根据您的客户端证书更改以下内容：

```
-Djavax.net.ssl.keyStore="<安装目录>/studio/var/security/certificate.p12"
```

```
-Djavax.net.ssl.keyStorePassword=changeit

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

4. 启动 Studio。

备注: 对于客户端证书，您需要定义用户(内部用户或 LDAP 用户)。用户的名称应该在证书属性中定义。默认为 CN 属性的值。有关更多详细信息，请参阅处理证书主体。

请注意，即使已将 HP OO 设置为具有多个 LDAP 配置，仍然只能使用具有默认 LDAP 的客户端证书属性对用户进行身份验证。Central 将首先尝试使用默认的 LDAP 对用户进行身份验证，如果失败，则将尝试在 HP OO 内部域中进行身份验证。

在 OOSH 中配置客户端证书

先决条件：必须将 Central 的 CA 根证书导入 OOSH 信任库。请参阅将证书导入 OOSH 信任库 (第 8 页)。

1. 停止 OOSH。
2. 编辑 **<安装目录>/central/bin** 中的 **oosh.bat**。
3. 根据您的客户端证书更改以下内容：

```
-Djavax.net.ssl.keyStore="<安装目录>/var/security/certificate.p12"

-Djavax.net.ssl.keyStorePassword=changeit

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

4. 启动 OOSH。

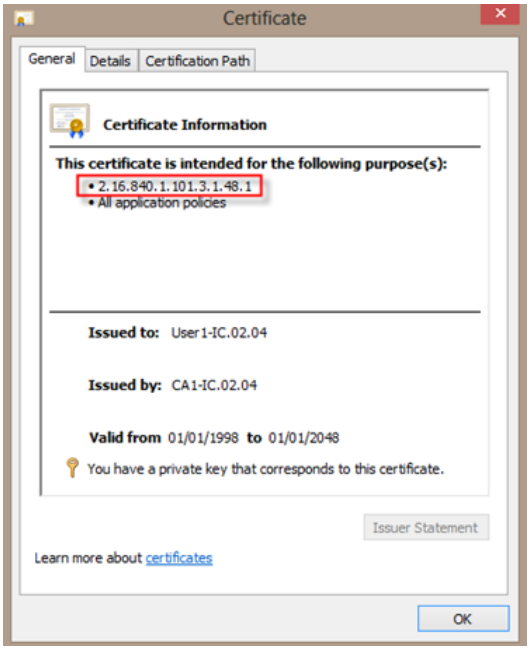
备注: 对于客户端证书，您需要定义用户(内部用户或 LDAP 用户)。用户的名称应该在证书属性中定义。默认为 CN 属性的值。有关更多详细信息，请参阅处理证书主体。

请注意，即使已将 HP OO 设置为具有多个 LDAP 配置，仍然只能使用具有默认 LDAP 的客户端证书属性对用户进行身份验证。Central 将首先尝试使用默认的 LDAP 对用户进行身份验证，如果失败，则将尝试在 HP OO 内部域中进行身份验证。

处理证书策略

HP OO 将处理对端点证书的证书策略所进行的处理。

- 可以在证书中设置目的字符串。
- HP OO 允许您将策略字符串添加为配置项，并检查每个端点证书的策略字符串。如果不匹配，则拒绝证书。
- 通过添加以下配置项启用或禁用证书策略验证：`x509.certificate.policy.enabled=true/false`(默认值为 `false`)。
- 通过添加以下配置项定义策略列表：`x509.certificate.policy.list=<`逗号分隔的列表`>`(默认为空列表)。



处理证书主体

您可以定义如何使用针对 `Subject` 的正则表达式匹配来从证书中获得主体。正则表达式应包含单个组。默认表达式 `CN=(.?)` 与常用名字段匹配。例如，`CN=Jimi Hendrix, OU=` 分配用户名 `Jimi Hendrix`。

- 匹配不区分大小写。
- 证书的主体是 HP OO 中的用户名(LDAP 或内部用户)。
- 要更改正则表达式，请更改配置项：`x509.subject.principal.regex`。

疑难解答

如果服务器没有启动，请打开 **wrapper.log** 文件并在 `ProtocolHandler ["http-nio-8443"]` 中查找错误。

当 Tomcat 正在初始化或正在启动连接器时，会发生这种问题。存在很多变体，但错误消息可提供相关信息。

所有 HTTPS 连接器参数均位于 **C:\HP\oo\central\tomcat\conf\server.xml** 的 Tomcat 配置文件中。

打开此文件并滚动到最后，直到看到 HTTPS 连接器：

```
<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"/>
```

通过将这些参数与在先前步骤中输入的参数进行比较，查看参数是否存在任何不匹配的情况。

联邦信息处理标准 (FIPS)

配置 HP OO 以兼容 FIPS 140-2

此部分说明如何配置 HP Operations Orchestration 以兼容联邦信息处理标准 (FIPS) 140-2。

FIPS 140-2 是由国家标准和技术研究所 (NIST) 定义的用于加密模块的安全要求标准。要查看此标准的出版物，请转至：csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf。

在配置 HP OO 以兼容 FIPS 140-2 后，HP OO 使用以下安全算法：

- 对称密钥算法：AES
- 哈希算法：SHA1

HP OO 使用此安全提供程序：RSA BSAFE Crypto 6.1 版软件。该软件是唯一支持的 FIPS 140-2 安全提供程序。

备注：在配置 HP OO 以兼容 FIPS 140-2 后，除非重新安装 HP OO，否则无法还原到标准配置。

先决条件

备注：如果要从已配置使用 FIPS 的 HP OO 10.10(及更高版本)的安装进行升级，则必须重复下面的步骤 4 和 5，然后重复配置 HP OO 以兼容 FIPS 140-2 (第 20 页)的“配置 Java 安全文件中的属性”部分中的步骤。

在配置 HP OO 以兼容 FIPS 140-2 之前，请执行以下步骤：

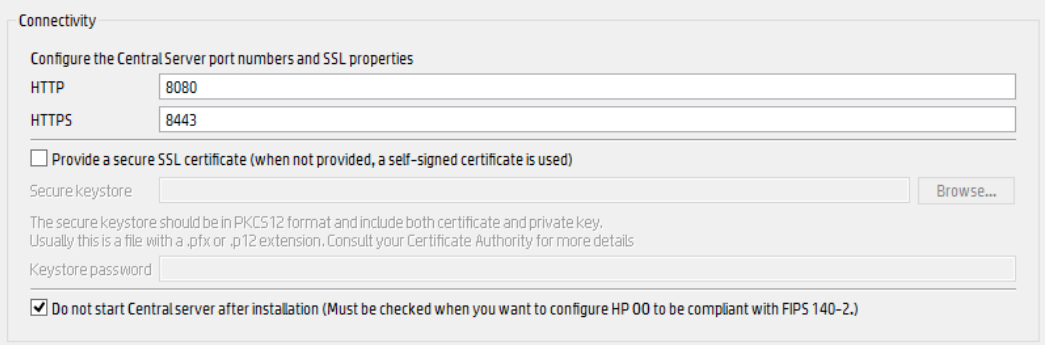
备注：为了兼容 FIPS140-2，您需要关闭 LWSSO。

1. 验证是否正在配置新安装的 HP OO 版本 10.10 或更高版本以与 FIPS 140-2 兼容，以及该版本是否未在使用中。

   您不能配置正在使用的 HP OO 安装版(不管是版本 9.x 还是 10.x)。

2. 验证在安装 HP OO 时未将其配置成安装后启动 Central 服务器：
   - 在静默安装中，`should.start.central` 参数设置为 **no**。
   - 在向导安装中的“Connectivity”步骤中，选中了“Do not start Central server after

复选框。



3. 备份以下目录：
   - **<安装目录>\central\tomcat\webapps\oo.war**
   - **<安装目录>\central\tomcat\webapps\PAS.war**
   - **<安装目录>\central\conf**
   - **<oo_jre>\lib\security**(其中 **<oo_jre>** 是用于安装 HP OO 所使用的 JRE 的目录。默认为 **<安装目录>\java**)
4. 在以下站点上下载并安装 Java Cryptographic Extension (JCE) Unlimited Strength Jurisdiction 策略文件：http://www.oracle.com/technetwork/java/javase/downloads/jce-7-download-432124.html。

   备注：有关如何部署文件和升级 HP OO 所使用的 JRE 的信息，请参阅下载内容中的 **ReadMe.txt** 文件。
5. 安装 RSA BSAFE Crypto 软件文件。在安装了 HP OO 的系统上，将以下内容复制到 **<oo_jre>\lib\ext** 中(其中 **<oo_jre>** 是用于安装 HP OO 所使用的 JRE 的目录。默认为 **<安装目录>\java**)。
   - **<安装目录>\central\lib\cryptojce-6.1.jar**
   - **<安装目录>\central\lib\cryptojcommon-6.1.jar**
   - **<安装目录>\central\lib\jcmFIPS-6.1.jar**

备注：如果要从已配置使用 FIPS 的 HP OO 10.10(及更高版本)的安装进行升级，则必须重复上面的“先决条件”部分中的步骤 4 和 5，然后重复配置 HP OO 以兼容 FIPS 140-2 (第 20 页)的“配置 Java 安全文件中的属性”部分中的步骤。

配置 HP OO 以兼容 FIPS 140-2

以下列表显示为了配置 HP OO 以兼容 FIPS 140-2 而需要执行的过程：

- 配置 Java 安全文件中的属性
- 配置 encryption.properties 文件并启用 FIPS 模式
- 创建兼容 FIPS 的 HP OO 加密
- 替换数据库密码
- 启动 HP OO

配置 Java 安全文件中的属性

编辑 JRE 的 Java 安全文件以添加其他安全提供程序，并为兼容 FIPS 140-2 配置属性。

备注：升级到 HP OO 10.10 会完全替换已安装的 JRE 文件。因此，在升级到 10.10 后必须执行以下步骤。

备注：如果要从已配置使用 FIPS 的 HP OO 10.10(及更高版本)的安装进行升级，则必须重复 联邦信息处理标准 (FIPS) (第 18 页)的“先决条件”部分中的步骤 4 和 5，然后重复此处的步骤。

在编辑器中打开 **<oo_jre>\lib\security\java.security** 文件并执行以下步骤：

1. 对于每个列出的提供程序，以 **security.provider.<nn>=<提供程序名称>** 格式递增首选项顺序编号 <nn>，每次递增 2。

   例如，将提供程序条目从

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

   更改为

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

2. 添加新的默认提供程序 (RSA JCE)。在提供程序列表顶部添加以下提供程序：

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

3. 添加 RSA BSAFE SSL-J Java 安全套接字扩展 (JSSE) 提供程序。

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

4. 将以下行复制并粘贴到 **java.security** 文件中以确保在兼容 FIPS 140-2 模式下使用

RSA BSAFE：

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

可以在 **java.security** 文件中的任意位置粘贴此行。

5. 由于默认的 DRBG 算法 ECDRBG128 并不安全(根据 NIST)，可通过将以下行复制到 **java.security** 文件，将安全属性 **com.rsa.crypto.default** 设置成 **HMACDRBG**：

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

可以在 **java.security** 文件中的任意位置粘贴此行。

6. 保存并退出 **java.security** 文件。

配置 encryption.properties 文件并启用 FIPS 模式

必须更新 HP OO 加密属性文件以便兼容 FIPS 140-2。

1. 备份位于 **<安装目录>\central\var\security** 中的 **encryption.properties** 文件。
2. 在文本编辑器中打开 **encryption.properties** 文件。例如，编辑以下文件：

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

3. 找到 `keySize=128` 并将其替换为 `keySize=256`。
4. 找到 `secureHashAlgorithm=SHA1` 并将其替换为 `secureHashAlgorithm=SHA256`。
5. 找到 `FIPS140ModeEnabled=false` 并将其替换为 `FIPS140ModeEnabled=true`。

   备注：如果 `FIPS140ModeEnabled=false` 不存在，请将 `FIPS140ModeEnabled=true` 作为新行添加到文件末尾。

6. 保存并退出文件。

创建兼容 FIPS 的 HP OO 加密

要创建或替换 HP OO 加密存储文件以便兼容 FIPS，请参阅替换 FIPS 加密 (第 22 页)。

备注：AES 有三种批准的密钥长度：128/192/256(根据 NIST SP800-131A 出版物)。

FIPS 支持以下安全哈希算法：SHA1、SHA256、SHA384、SHA512。

备注：建议更改 key.store(及其私钥条目)和信任库的密码。请参阅更改 Keystore/信任库密码 (第 10 页)。

备注: 建议删除 HP OO 信任库中的所有默认 CA 根证书。(client.truststore 位于 <安装>/central/var/security 中。)

替换数据库密码

按照更改数据库密码 (第 26 页)中所述，替换数据库密码。

启动 HP OO

按照《HP OO 安装指南》中所述，启动 HP OO。

替换 FIPS 加密

HP OO、Central 和 RAS 符合联邦信息处理标准 140-2 (FIPS 140-2)，此标准定义了联邦机构为保护敏感数据或有价值数据而指定基于加密的安全系统时要使用的技术要求。

进行 HP OO 10.10 全新安装后，可选择更改 FIPS 加密算法。

备注: 此过程仅适用于全新安装。升级后无法执行此过程。

更改 Central 上的 FIPS 加密算法

1. 转至 **<Central 安装文件夹>/var/security**。
2. 备份并删除 **encryption_repository** 文件。
3. 转至 **<Central 安装文件夹>/bin/**。
4. 运行 **generate-keys** 脚本。

   此时将在 **<Central 安装文件夹>/var/security/encryption_repository** 中生成新的主密钥。

更改 RAS 加密属性

如果在新位置安装 RAS，则需要完成以下所有步骤。

备注: 在更改 Central 加密属性后，只有当在新 RAS 安装上工作时，这些更改才有效。

要更改 RAS 加密属性，请执行以下操作：

1. 完成 联邦信息处理标准 (FIPS) (第 18 页)的“先决条件”部分中的所有步骤。
2. 完成配置 HP OO 以兼容 FIPS 140-2 (第 20 页)的“配置 Java 安全文件中的属性”中的所有步骤。
3. 将当前 **encryption.properties** 文件从 <安装目录>\ras\var\security 复制到 <安装目录>\ras\bin 文件夹。
4. 使用任意文本编辑器，根据需要编辑和更改 **encryption.properties** 文件。

   有关详细信息，请参阅配置 HP OO 以兼容 FIPS 140-2 (第 20 页)中的“配置 encryption.properties 文件并启用 FIPS 模式”。
5. 保存变更。
6. 在文件夹 <安装目录>\ras\bin 中打开命令行提示符。
7. 运行 **oosh.bat**。
8. 运行 OOShell 命令：`replace-encryption --file encryption.properties`

   备注：如果将 **encryption.properties** 文件复制到其他文件夹，请确保在 OOShell 命令中输入正确的位置。
9. 重新启动 RAS 服务。

配置 LW SSO 设置

安装 HP OO 10.10 时，如果选择从 HP OO 9.x 升级 LW SSO 设置，则这些 LW SSO 设置将迁移，但是在 HP OO 10.10 中将禁用 LW SSO(即使之前已在 HP OO 9.x 中启用)。

如果之后启用 LW SSO，则可能会在某些场景中接收到警告。要从日志中清除警告，请遵循以下步骤，使用完全限定域名设置管理 URL 属性。

- 如果 Central 和 RAS 安装在同一计算机上，且 LW SSO 设置已启用，则必须使用完全限定域名设置管理 URL 属性。

  a. 停止 RAS 进程。

  b. 在 **ras/conf/ras-wrapper.conf** 文件中，将

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

  更改为

  `wrapper.java.additional.<x>=-Dmgmt.url=<协议>://<完全限定域名>:<端口>/oo`

  c. 启动 RAS 进程。

- 如果 RAS 和 Central 安装在不同的计算机上，且 LW SSO 设置已启用，则必须在 RAS 安装期间使用完全限定域名(而不是 IP 地址)指定 Central 的管理 URL。
- 如果通过 LW SSO 将其他应用程序连接到 Central，则必须使用完全限定域名指定 Central 的管理 URL。

  a. 停止 Central 进程。

  b. 在 **central/conf/central-wrapper.conf** 文件中，将

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

  更改为

  ```
  wrapper.java.additional.<x>=-Dmgmt.url=<协议>://<完全限定域名>:<端口>/oo
  ```

  c. 启动 Central 进程。

配置 XSS 策略

HP OO 使用 AntiSamy 提供 XSS 保护。默认保护策略为“antisamy”，它允许使用大部分 HTML 元素，并在用户提交完整的 HTML 页面时非常有用。

可将此策略配置为 AntiSamy 支持的策略之一。有关详细信息，请参阅

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

可通过称为 `xss.policy` 的系统配置属性配置此策略。可能的值包括：antisamy(默认值)、antisamy-slashdot、antisamy-myspace、antisamy-ebay、antisamy-anythinggoes、antisamy-tinymc。

要查看所配置的策略，请转至 **https://host/oo/reports/sysinfo**，然后在“系统配置”部分中查找 `xss.policy` 参数。

更改 Slashdot 默认策略的最简单方法是使用 HP Operations Orchestration Shell 实用程序。

1. 双击批处理文件 `oosh.bat`，可启动 OOSH 实用程序。
2. 在命令行中输入命令，例如：

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

有关 HP Operations Orchestration Shell 实用程序的详细信息，请参阅《HP Operations Orchestration Shell User Guide》。

配置本地化

在 Central-wrapper.conf 中设置系统区域设置

如果 HP OO 系统已本地化，则需要在 **central-wrapper.conf** 文件中设置以下属性以反映系统区域设置：

```
set.LANG=

set.LC_ALL=

set.LANGUAGE=

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

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

例如，对于日语：设置 `set.LANG=ja_JP` 和 `set.LC_ALL=ja_JP`

配置系统

更改数据库密码

1. 如果 Central 已启动并在运行，请停止 Central 服务。
2. 使用 -e -p <密码> 选项运行 encrypt-password 脚本，其中“密码”是数据库的密码。
3. 复制结果，结果应如下所示：

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

4. 转至 **<Central 安装文件夹>/conf** 文件夹，打开 **database.properties** 文件。
5. 将 db.password 值更改为复制的值。

更改数据库 IP

如果需要配置 HP OO 以使用其他数据库实例，则此部分内容与您有关。所有数据库参数(如数据库凭据、架构名称、表等)均应相同。

1. 编辑文件 **\HP Operations Orchestration\central\conf\database.properties**。
2. 查找 jdbc.url 参数。例如：

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

3. 更改数据库服务器的 IP 地址\FQDN。
4. 保存文件。
5. 重新启动 Central。

调整日志记录级别

可以单独针对常规日志记录、部署和执行来调整日志中提供的信息粒度。

粒度选项包括：

- INFO - 默认日志记录信息
- DEBUG - 更多日志记录信息
- ERROR/WARNING - 更少日志记录信息

要调整日志记录中的粒度，请执行以下操作：

1. 打开 **log4j.properties** 文件(在 **/<OO 安装目录>/central/conf/log4j.properties** 下)。
2. 在 **log4j.properties** 文件的以下位置中使用 DEBUG 或 ERROR/WARNING 替换 INFO。

   例如：

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

调整 Quartz 作业的时间安排

在 HP OO 系统中，Quartz 作业会定期运行以维护系统。

每个作业会运行一段设定的时间，并将按设定间隔重复。以下是作业触发器示例：

触发器名称	当前重复间隔	发生的操作
onRolling:OO_EXECUTION_STATES_Trigger	4.5 分钟	滚动状态表以进行清理
queueCleanerTrigger	1 分钟	清理队列表
queueRecoveryTrigger	2 分钟	检查系统是否需要恢复
recoveryVersionTrigger	0.5 分钟	用于恢复的版本计数器
splitJoinTrigger	1 秒	联接完成的分割片段
onRolling:OO_EXECUTION_EVENTS_Trigger **备注：**此触发器已弃用。	12 小时	滚动事件表以进行清理

如果要调整这些作业时间安排以提高性能，请执行以下操作：

备注：时间安排的任何变更都可能对系统造成重大影响。请在对这些触发器执行任何变更之前先咨询您的 HP 服务代表。

1. 使用 URL 输入 Jminix 页面：**{OO_HOST}:{OO_PORT}/oo/jminix/**

   备注：您需要拥有“管理系统设置”权限才可以输入 **jminix**。

2. 打开 OO 选项卡。在 **MBeans** 下有一个名为 **jobTriggersMBean** 的操作。
3. 使用此操作，在右侧选项卡中输入值(使用您要更改的触发器的名称)。使用与表中完全相同的名称，以及新的重复间隔值。

此操作将更改作业的触发时间。

备注: 事件持久性机制已弃用(请参阅 **onRolling:OO_EXECUTION_EVENTS_Trigger**)。如果使用 Remote Debugger 或如果打开了 **events.persistency** 标记，则可以配置此作业。请参阅打开事件日志机制 (第 28 页)。

更改 RAS 端上 Central/负载平衡器的 URL

尽管最佳实践是通过安装程序配置 Central/负载平衡器的 URL，但是如果在已安装 RAS 之后需要更改 URL，则可以通过编辑 **ras-wrapper.conf** 文件实现此目的。

例如，如果针对 Central/负载平衡器安装了 RAS 且 Central/负载平衡器的 FQDN 发生了更改，则需要执行此操作。您将需要更改在 RAS 级别上存储的 Central/负载平衡器的 URL，以便 RAS 能够再次与 Central/负载平衡器通信。

1. 停止 RAS。
2. 打开位于 **<安装文件夹>\ras\conf** 下的 **ras-wrapper.conf** 文件。
3. 编辑以下行中的 URL：

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

4. 重新启动 RAS。

打开事件日志机制

事件日志机制在 HP OO 10.10 中已弃用，将从未来的发行版中删除。

HP OO 在部署时不含事件日志机制。但如果您想使用此机制，可以打开 **events.persistency** 标记。请注意，对于群集场景和希望提高性能的情况，建议关闭此标记。

要打开此标记，请执行以下操作：

1. 停止进程并更新系统中每个节点 (Central/RAS) 上的 **wrapper.conf** 文件。

   在 Central 上，转至 **<安装路径>\central\conf\central-wrapper.conf**

   在 RAS 上，转至 **<安装路径>\ras\conf\ras-wrapper.conf**

2. 在 **wrapper.conf** 文件中，查找参数 **-Devents.persistency** 并将其值更改为 **true**。
3. 重新启动进程。


Java™
COMPATIBLE