HP Universal CMDB and Configuration Manager

Software Version: 10.10

Hardening Guide

Document Release Date: November 2013

Software Release Date: November 2013

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.

Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation.

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

Chapter 1: Introduction to Hardening

This chapter includes:

Hardening Overview

This section introduces the concept of a secure HP Universal CMDB application and discusses the planning and architecture required to implement security. It is highly recommended that you read this section before proceeding to the hardening discussion in the following sections.

HP Universal CMDB is designed so that it can be part of a secure architecture, and can therefore meet the challenge of dealing with the security threats to which it might be exposed.

The hardening guidelines deal with the configuration required to implement a more secure (hardened) HP Universal CMDB.

The hardening information provided is intended primarily for HP Universal CMDB administrators who should familiarize themselves with the hardening settings and recommendations prior to beginning the hardening procedures.

It is highly recommended that you use a reverse proxy with HP Universal CMDB to achieve a secure architecture. For details on configuring a reverse proxy for use with HP Universal CMDB, see "Using a Reverse Proxy" on page 38.

If you must use another type of secure architecture with HP Universal CMDB other than described in this document, contact HP Software Support to determine which architecture is the best one for you to use.

For details on hardening the Data Flow Probe, see "Data Flow Probe Hardening" on page 70.

Note:

- The hardening procedures are based on the assumption that you are implementing only the instructions provided in these chapters, and that you are not performing other hardening

steps documented elsewhere.

- Where the hardening procedures focus on a particular distributed architecture, this does not imply that this is the best architecture to fit your organization's needs.
- It is assumed that the procedures included in the following chaptersare to be performed on machines dedicated to HP Universal CMDB. Using the machines for other purposes in addition to HP Universal CMDB may yield problematic results.
- The hardening information provided in this section is not intended as a guide to making a security risk assessment for your computerized systems.

Hardening Preparations

- Evaluate the security risk/security state for your general network, and use the conclusions when deciding how to best integrate HP Universal CMDB into your network.
- Develop a good understanding of the HP Universal CMDB technical framework and HP Universal CMDB security capabilities.
- Review all the hardening guidelines.
- Verify that HP Universal CMDB is fully functioning before starting the hardening procedures.
- Follow the hardening procedure steps chronologically in each chapter. For example, if you decide to configure the HP Universal CMDB server to support SSL, read "Enabling Secure Sockets Layer (SSL) Communication" on page 17 and then follow all the instructions chronologically.
- HP Universal CMDB does not support basic authentication with blank passwords. Do not use a blank password when setting basic authentication connection parameters.

Tip: Print out the hardening procedures and check them off as you implement them.

Deploying UCMDB in a Secure Architecture

Several measures are recommended to securely deploy your HP Universal CMDB servers:

- **DMZ architecture using a firewall**

  The secure architecture referred to in this document is a typical DMZ architecture using a device as a firewall. The basic concept of such an architecture is to create a complete separation, and to avoid direct access between the HP Universal CMDB clients and the HP Universal CMDB server.

- **Secure browser**

  Internet Explorer and Firefox in a Windows environment must be configured to securely handle Java scripts, applets, and cookies.

- **SSL communication protocol**

  Secure Sockets Layer protocol secures the connection between the client and the server. URLs that require an SSL connection use a secure version (HTTPS) of the Hypertext Transfer Protocol. For details, see "Enabling Secure Sockets Layer (SSL) Communication" on page 17.

- **Reverse proxy architecture**

  One of the more secure and recommended solutions suggests deploying HP Universal CMDB using a reverse proxy. HP Universal CMDB fully supports secure reverse proxy architecture. For details, see "Using a Reverse Proxy" on page 38.

System Access

Java JMX Access Hardening

Note: The procedure described here can also be used for the Data Flow Probe JMX.

In order to ensure that the JMX RMI port is accessible only when providing user credentials, perform the following procedure:

1. In the **wrapper.conf** file on the server, located at **C:\hp\UCMDB\UCMDBServer\bin**, set the following:

   wrapper.java.additional.16=-Dcom.sun.management.jmxremote.authenticate=true

   This setting requires the JMX to ask for authentication.

   - **For the Data Flow Probe JMX**, perform the following:

     In the files **WrapperGateway.conf** and **WrapperManager.conf**, located at **C:\hp\UCMDB\DataFlowProbe\bin**, set the following:

     wrapper.java.additional.17=-Dcom.sun.management.jmxremote.authenticate=true

2. Rename the file **jmxremote.password.template** (located at: **C:\hp\UCMDB\UCMDBServer\bin\jre\lib\management**) to **jmxremote.password**.

   Note: For the Data Flow Probe JMX, this file is located at: **C:\hp\UCMDB\DataFlowProbe\bin\jre\lib\management**.

3. In **jmxremote.password**, add passwords for the roles **monitorRole** and **controlRole**.

   For example:

   monitorRole QED

   controlRole R&D

   would assign the password **QED** to **monitorRole** and the password **R&D** to **controlRole**.

   Note: Ensure that only the owner has read and write permissions on **jmxremote.password** because it contains the passwords in clear text. The file owner must be the same user under which UCMDB Server is running.

4. In the file **jmxremote.access** (located at **C:\hp\UCMDB\UCMDBServer\bin\jre\lib\management**), assign access to **monitorRole** and **controlRole**.

   For example:

   monitorRole readonly

   controlRole readwrite

   would assign read-only access to **monitorRole** and read-write access to **controlRole**.

   Note: For the Data Flow Probe JMX, this file is located at: **C:\hp\UCMDB\DataFlowProbe\bin\jre\lib\management**.

5. Secure files as follows:

   - **For Windows only:** Run the following commands from the command line to secure files:

     cacls jmxremote.password /P <username>:F

     cacls jmxremote.access /P <username>:R

     where **<username>** is the file owner visible in the properties of both files. Open properties of these files and ensure that they are correct and have only one owner.

   - **For Solaris and Linux operating systems:** Set the file permissions for the password file by running:

     chmod 600 jmxremote.password

6. **For Service Pack upgrades, Server migrations and Disaster Recovery:** Change ownership of the file **jmxremote.access** (located at **C:\hp\UCMDB\UCMDBServer\bin\jre\lib\management**) to the operating system user

running the upgrade or migration installation.

Note:

- For the Data Flow Probe JMX, this file is located at: **C:\hp\UCMDB\DataFlowProbe\bin\jre\lib\management**.
- Before uninstalling the product, edit the file permissions for **<UMCDB installation folder>\bin\jre\lib\management\jmxremote.password** so the user you are logged in with can edit it.

Changing System User Name or Password for the JMX Console

The JMX console uses system users, that is, cross-customer users in a multi-customer environment. You can log in to the JMX console with any system user name. The default name and password is **sysadmin/sysadmin**.

You change the password either through the JMX console or through the Server Management tool.

To change the default system user name or password through the JMX console:

1. Launch a Web browser and enter the following address: **http://localhost.<domain_name>:8080/jmx-console**.
2. Enter the JMX console authentication credentials.
3. Locate **UCMDB:service=Authorization Services** and click the link to open the Operations page.
4. Locate the **resetPassword** operation.
   - In the **userName** field, enter **sysadmin**.
   - In the **password** field, enter a new password.
5. Click **Invoke** to save the change.

To change the default system user name or password through the Server Management tool:

1. **For Windows:** run the following file: **C:\hp\UCMDB\UCMDBServer\tools\server_management.bat.**

   For Linux: Run **server_management.sh** located in the following folder: **/opt/hp/UCMDB/UCMDBServer/tools/**.
2. Log in to the tool with the authentication credentials: **sysadmin/sysadmin**.

3. Click the Users link.
4. Select the system user and click **Change password for logged-on user**.
5. Enter the old and new passwords and click **OK**.

Changing the HP Universal CMDB Server Service User

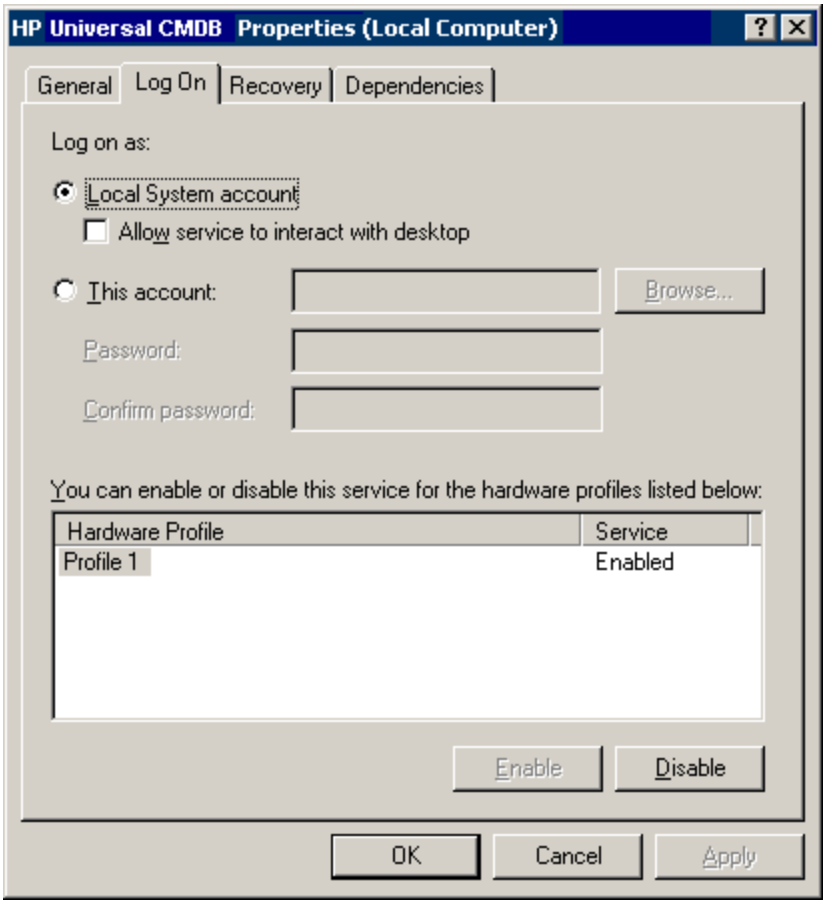
On a Windows platform, the HP Universal CMDB service, which runs all HP Universal CMDB services and processes, is installed when you run the Server and Database Configuration utility. By default, this service runs under the local system user. However, you may need to assign a different user to run the service (for example, if you are using NTLM authentication).

The user you assign to run the service must have the following permissions:

- sufficient database permissions (as defined by the database administrator)
- sufficient network permissions
- administrator permissions on the local server

To change the service user:

1. Disable HP Universal CMDB through the Start menu (**Start > All Programs > HP UCMDB > Stop HP Universal CMDB Server**) or by stopping the HP Universal CMDB Server service. For details, see the section describing how to start and stop the UCMDB Server Service in the *HP Universal CMDB Administration Guide*
2. In the Windows **Services** window, double-click **UCMDB_Server**. The **UCMDB_Server Properties (Local Computer)** dialog box opens.
3. Click the Log On tab.



4. Select **This account** and browse to choose another user from the list of valid users on the machine.
5. Enter the selected user's Windows password and confirm this password.
6. Click **Apply** to save your settings and **OK** to close the dialog box.
7. Enable HP Universal CMDB through the Start menu (**Start > All Programs > HP UCMDB > Start HP Universal CMDB Server**) or by starting the HP Universal CMDB Server service. For details, see the section describing how to start and stop the UCMDB Server Service in the *HP Universal CMDB Administration Guide*.

Encrypt the Database Password for Configuration Manager

The CM database password is stored in the **<*Configuration_Manager_installation_directory*>\conf\database.properties** file. If you want to encrypt the password, our default encryption algorithm complies with the standards of FIPS 140-2.

The encryption is accomplished by means of a key, through which the password is encrypted. The key itself is then encrypted using another key, known as a master key. Both keys are encrypted

using the same algorithm. For details on the parameters used in the encryption process, see "Parameters for Configuration Manager Database Password Encryption" below

Caution: If you change the encryption algorithm, all previously encrypted passwords are no longer usable.

To change the encryption of your database password:

1. Open the **<*Configuration_Manager_installation_directory*>\conf\database.properties** file and edit the following fields:
   - **engineName.** Enter the name of the encryption algorithm.
   - **keySize.** Enter the size of the master key for the selected algorithm.
2. Run the **generate-keys.bat** script, which creates the **<*Configuration_Manager_installation_directory*>\security\encrypt_repository** file and generates the encryption key.
3. Run the **bin\encrypt-password.bat** utility to encrypt the password. Set the **-h** flag to see the available options.
4. Copy the result of the password encryption utility and paste the resulting encryption into the **conf\database.properties** file.

Parameters for Configuration Manager Database Password Encryption

The following table lists the parameters included in the **encryption.properties** file used for CM database password encryption. For details on encrypting the database password, see "Encrypt the Database Password for Configuration Manager" on the previous page.

Parameter	Description
cryptoSource	Indicates the infrastructure implementing the encryption algorithm. The available options are: • **lw.** Uses Bouncy Castle lightweight implementation (Default option) • **jce.** Java Cryptography Enhancement (standard Java cryptography infrastructure)
storageType	Indicates the type of the key storage. Currently, only **binary file** is supported.
binaryFileStorageName	Indicates the place in the file where the master key is stored.

Parameter	Description
cipherType	The type of the cipher. Currently, only **symmetricBlockCipher** is supported.
engineName	The name of the encryption algorithm. The following options are available: • **AES.** American Encryption Standard. This encryption is FIPS 140-2 compliant. (Default option) • **Blowfish** • **DES** • **3DES.** (FIPS 140-2 compliant) • **Null**. No encryption
keySize	The size of the master key. The size is determined by the algorithm: • **AES.** 128, 192, or 256 (Default option is 256) • **Blowfish.** 0-400 • **DES.** 56 • **3DES.** 156
encodingMode	The ASCII encoding of the binary encryption results. The following options are available: • **Base64** (Default option) • **Base64Url** • **Hex**
algorithmModeName	The mode of the algorithm. Currently, only **CBC** is supported.
algorithmPaddingName	The padding algorithm used. The following options are available: • **PKCS7Padding** (Default option) • **PKCS5Padding**

Parameter	Description
jceProviderName	The name of the JCE encryption algorithm. **Note:** Only relevant when crytpSource is jce. For lw, engineName is used.

Chapter 2: Enabling Secure Sockets Layer (SSL) Communication

This chapter includes:

Enable SSL on the Server Machine With a Self-Signed Certificate - UCMDB

These sections explain how to configure HP Universal CMDB to support communication using the Secure Sockets Layer (SSL) channel.

1. **Prerequisites**

   a. Before starting the following procedure, remove the old **server.keystore** located in **C:\hp\UCMDB\UCMDBServer\conf\security\server.keystore**.

   b. Place the HP Universal CMDB keystore (JKS type) in the **C:\hp\UCMDB\UCMDBServer\conf\security** folder.

2. **Generate a Server Keystore**

   a. Create a keystore (JKS type) with a self-signed certificate and matching private key:

      - From **C:\hp\UCMDB\UCMDBServer\bin\jre\bin**, run the following command:

        ```
        keytool -genkey -alias hpcert -keystore
        C:\hp\UCMDB\UCMDBServer\conf\security\server.keystore
        ```

        The console dialog box opens.

      - Enter the keystore password. If the password has changed, run the **changeKeystorePassword** JMX operation, in **UCMDB:service=Security Services**. If the password has not changed, use the default **hppass** password.

      - Answer the question, **What is your first and last name?** Enter the HP Universal CMDB Web server name. Enter the other parameters according to your organization.

      - Enter a key password. The key password MUST be the same as the keystore password.

        A JKS keystore is created named **server.keystore** with a server certificate named **hpcert**.

   b. Export the self-signed certificate to a file:

   From **C:\hp\UCMDB\UCMDBServer\bin\jre\bin**, run the following command:

   ```
   keytool -export -alias hpcert -keystore
   C:\hp\UCMDB\UCMDBServer\conf\security\server.keystore -storepass <your
   password> -file hpcert
   ```

3. **Place the Certificate in the Client's Trusted Store**

   After generating **server.keystore** and exporting the server certificate, for every client that needs to communicate with HP Universal CMDB over SSL using this self-signed certificate, place this certificate in the client's trusted stores.

   Note: There can be one server certificate only in **server.keystore**.

4. **Disable HTTP Port 8080**

   For details, see "Enable or Disable HTTP/HTTPS Ports" on page 31.

   Note: Check that HTTPS communication works before closing the HTTP port.

5. **Restart the Server**

6. **Display HP Universal CMDB**

   To verify that the UCMDB Server is secure, enter the following URL in the Web browser: **https://<UCMDB Server name or IP address>:8443/ucmdb-ui**.

Enable SSL on the Server Machine with a Self-Signed Certificate - Configuration Manager

This section explains how to configure Configuration Manager to support authentication and encryption using the Secure Sockets Layer (SSL) channel.

Configuration Manager uses Tomcat 7.0.19 as the application server.

1. **Prerequisites** (not relevant if installing for the first time)

   Before starting the following procedure, remove the old **tomcat.keystore** file located in the **<*Configuration_Manager_installation_directory*>\java\windows\x86_64\lib\security** folder or the **<*Configuration_Manager_installation_directory*>\java\linux\x86_64\lib\security** folder (whichever is relevant), if it exists.

2. **Generate a Server Keystore**

   Create a keystore (JKS type) with a self-signed certificate and matching private key:

   - From **<*Configuration_Manager_installation_directory*>\java\windows\x86_64\bin** or **<*Configuration_Manager_installation_directory*>\java\linux\x86_64\bin**, run the following command:

     ```
     keytool -genkey -alias tomcat -keyalg RSA -keystore ..\lib\
     security\tomcat.keystore
     ```

     The console dialog box opens.

   - Enter the keystore password. If the password has changed, change it manually in the file.

   - Answer the question, **What is your first and last name?** Enter the Configuration Manager Web server name. Enter the other parameters according to your organization.

   - Enter a key password. The key password MUST be the same as the keystore password.

     A JKS keystore is created named **tomcat.keystore** with a server certificate named **hpcert**.

3. **Place the Certificate in the Client's Trusted Store**

   Add the certificate to the client's trusted stores in Internet Explorer on your computer (**Tools > Internet Options > Content > Certificates**). If not, you will be prompted to do so the first time you attempt to use Configuration Manager.

Limitation: There can be one server certificate only in **tomcat.keystore**.

4. **Modify the server.xml File**

   Open the **server.xml** file, located in **<*Configuration_Manager_installation_directory*>\servers\server-0\conf**. Locate the section beginning with

   ```
   Connector port="8143"
   ```

   which appears in comments. Activate the script by removing the comment character and add the following attributes to the HTTPS connector:

   ```
   keystoreFile="<tomcat.keystore file location>" (see step 2)
   keystorePass="<password>"
   ```

   Comment out the following line:

   ```
   <Listener className="org.apache.catalina.core.AprLifecycleListener"
   SSLEngine="on" />
   ```

   Note: You must not block the HTTP connection port. If you want to block HTTP communication, you can use a firewall for this purpose.

5. **Restart the Server**

   Restart the Configuration Manager server.

6. **Verify the Server Security**

   To verify that the Configuration Manager Server is secure, enter the following URL in the Web browser: **https://<Configuration Manager Server name or IP address>:8143/cnc**.

7. In Configuration Manger, go to **Settings>Application Management>Mail Settings** and change the protocol and port in **Configuration Manager full URL**, according to the values above.

8. In the UCMDB, go to **Infrastructure Settings Manager>General Settings** and change the protocol and port in the **Configuration Manager URL**, according to the values above.

Tip: If you fail to establish a connection, try using a different browser or upgrade to a newer version of the browser.

Enable SSL on the Server Machine With a Certificate from a Certification Authority - UCMDB

To use a certificate issued by a Certification Authority (CA), the keystore must be in Java format. The following example explains how to format the keystore for a Windows machine.

1. **Prerequisites**

   Before starting the following procedure, remove the old **server.keystore** located in **C:\hp\UCMDB\UCMDBServer\conf\security\server.keystore**.

2. **Generate a Server Keystore**

   a. Generate a CA signed certificate and install it on Windows.

   b. Export the certificate into a ***.pfx** file (including private keys) using Microsoft Management Console (**mmc.exe**).

   Enter any string as the password for the **pfx** file. (You are asked for this password when converting the keystore type to a JAVA keystore.) The **.pfx** file now contains a public certificate and a private key and is password protected.

   c. Copy the **.pfx** file you created to the following folder: **C:\hp\UCMDB\UCMDBServer\conf\security**.

   d. Open the command prompt and change the directory to **C:\hp\UCMDB\UCMDBServer\bin\jre\bin.**

   Change the keystore type from **PKCS12** to a **JAVA** keystore by running the following command:

   ```
   keytool -importkeystore -srckeystore c:\hp\UCMDB\UCMDBServer\conf\securit
   y\<pfx file name> -srcstoretype PKCS12 -destkeystore server.keystore
   ```

   You are asked for the source (**.pfx**) keystore password. This is the password you supplied when creating the **pfx** file in step b.)

   e. Enter the destination keystore password. This password must be the same as defined previously in the **changeKeystorePassword** JMX method, in Security Services. If the password was not changed, use the default **hppass** password.

   Note: The source keystore password must be the same as the destination keystore password.

f. After generating the certificate, disable HTTP port 8080. For details, see "Enable or Disable HTTP/HTTPS Ports" on page 31.

g. If you used a password other than **hppass** or the password used for the **.pfx** file, run the **changeKeystorePassword** JMX method and make sure that the key has the same password.

> **Note:** Check that HTTPS communication works before closing the HTTP port.

3. **Restart the Server**

4. **Verify the Server Security**

   To verify that the UCMDB Server is secure, enter the following URL in the Web browser: **https://<UCMDB Server name or IP address>:8443/ucmdb-ui**.

> **Caution:** There can be one server certificate only in **server.keystore**.

Enable SSL on the Server Machine with a Certificate from a Certification Authority - Configuration Manager

For Configuration Manager, in order to use a certificate issued by a Certification Authority (CA) the keystore must be in Java format. The following example explains how to format the keystore for a Windows machine.

1. **Prerequisites**

   Before starting the following procedure, remove the old **tomcat.keystore** file located in the **<*Configuration_Manager_installation_directory*>\java\windows\x86_64\lib\security** folder or the **<Configuration Manager installation directory>\java\linux\x86_64\lib\security** folder (whichever is relevant), if it exists.

2. **Generate a Server Keystore**

   a. Generate a CA signed certificate and install it on Windows.

   b. Export the certificate into a ***.pfx** file (including private keys) using Microsoft Management Console (**mmc.exe**).

      Enter any string as the password for the **pfx** file. (You are asked for this password when converting the keystore type to a JAVA keystore.)
      The **.pfx** file now contains a public certificate and a private key and is password protected.

Copy the **.pfx** file you created to the following folder: **<*Configuration_Manager_installation_directory*>\java\lib\security**.

c. Open the command prompt and change the directory to **<*Configuration_Manager_installation_directory*>\java\bin**.

Change the keystore type from **PKCS12** to a **JAVA** keystore by running the following command:

```
keytool -importkeystore -srckeystore <Configuration_Manager_installation_directory>\conf\security\<pfx file name> -srcstoretype PKCS12 -destkeystore tomcat.keystore
```

You are asked for the source (**.pfx**) keystore password. This is the password you supplied when creating the pfx file in step b.

3. **Modify the server.xml File**

Open the **server.xml** file, located in **<*Configuration_Manager_installation_directory*>\servers\server-0\conf**. Locate the section beginning with

```
Connector port="8143"
```

which appears in comments. Activate the script by removing the comment character and add the following two lines:

```
keystoreFile="../../java/lib/security/tomcat.keystore"
keystorePass="password" />
```

Comment out the following line:

```
<Listener className="org.apache.catalina.core.AprLifecycleListener"
SSLEngine="on" />
```

Note: You must not block the HTTP connection port. If you want to block HTTP communication, you can use a firewall for this purpose.

4. **Restart the Server**

Restart the Configuration Manager server.

5. **Verify the Server Security**

To verify that the Configuration Manager server is secure, enter the following URL in the Web browser: **https://<Configuration Manager Server name or IP address>:8143/cnc**.

6. In Configuration Manger, go to **Settings>Application Management> Mail Settings** and change the protocol and port in **Configuration Manager full URL**, according to the values

above.

7. In the UCMDB, go to **Infrastructure Settings Manager>General Settings** and change the protocol and port in **Configuration Manager URL**, according to the values above.

Limitation: There can be one server certificate only in **tomcat.keystore**.

Enable SSL on the Client Machines - UCMDB

If the certificate used by the HP Universal CMDB Web server is issued by a well-known Certificate Authority (CA), it is most likely that your Web browser can validate the certificate without any further action.

If the CA is not trusted by the Web browser, you should either import the entire certificate trust path or import the certificate used by HP Universal CMDB explicitly into the browser's truststore.

The following example demonstrates how to import the self-signed **hpcert** certificate into the Windows truststore to be used by Internet Explorer.

To import a certificate into the Windows truststore:

1. Locate and rename the **hpcert** certificate to **hpcert.cer**.

   In Windows Explorer, the icon shows that the file is a security certificate.

2. Double-click **hpcert.cer** to open the Internet Explorer Certificate dialog box.

3. Follow the instructions for enabling trust by installing the certificate with the Certificate Import Wizard.

Note: Another method of importing the certificate issued by the UCMDB Server to the Web browser is by logging in to UCMDB, and installing the certificate when the untrusted certificate warning is displayed.

Enable SSL with a Client Certificate - Configuration Manager

If the certificate used by the Configuration Manager Web server is issued by a well-known Certificate Authority (CA), it is most likely that your Web browser can validate the certificate without any further action.

If the CA is not trusted by the server trust store, import the CA certificate into the server trust store.

The following example demonstrates how to import the self-signed **hpcert** certificate into the server trust store (cacerts).

To import a certificate into the Server trust store:

1. On the client machine, locate and rename the **hpcert** certificate to **hpcert.cer**.
2. Copy **hpcert.cer** to the server machine in the **<*Configuration_Manager_installation_directory*>\java\windows\x86_64bin** folder.
3. On the server machine, import the CA certificate into the trust store (cacerts) using the keytool utility with the following command:

   ```
   <Configuration_Manager_installation_directory>\java\bin\keytool.exe -import
   -alias hp -file hpcert.cer -keystore ..\lib\security\cacerts
   ```

4. Modify the **server.xml** file (located in the **<*Configuration_Manager_installation_directory*>\servers\server-0\conf** folder) as follows:

   a. Make the changes described in "Modify the server.xml File" on page 23.

   b. Right after those changes, add the following attributes to the HTTPS connector:

   ```
   truststoreFile="../../java/lib/security/cacerts"
   truststorePass="changeit" />
   ```

   c. Set `clientAuth="true"`.

5. Verify the server security as described in "Verify the Server Security" on page 23.

Enable SSL on the Client SDK

You can utilize HTTPS transportation between the client SDK and the server SDK:

1. On the client machine, in the product that embeds the client SDK, locate the transportation setting and make sure it is configured to HTTPS, and not HTTP.
2. Download the CA certificate/self-signed public certificate to the client machine, and import it into the **cacerts** truststore on the JRE that is going to connect to the server.

   Use the following command:

   ```
   Keytool -import -alias <CA name> -trustcacerts -file <server public certific
   ate path> -keystore <path to client jre trusted cacerts store (e.g. x:\progr
   am files\java\jre\lib\security\cacerts)>
   ```

Enable Mutual Certificate Authentication for SDK

This mode uses SSL and enables both server authentication by the UCMDB and client authentication by the UCMDB-API client. Both the server and the UCMDB-API client send their certificates to the other entity for authentication.

Note: The following method of enabling SSL on the SDK with mutual authentication is the most secure of the methods and is therefore the recommended communication mode.

1. Harden the UCMDB-API client connector in UCMDB:

   a. Access the UCMDB JMX console: Launch a Web browser and enter the following address: **http://<UCMDB machine name or IP address>:8080/jmx-console**. You may have to log in with a user name and password (default is sysadmin/sysadmin).

   b. Locate **UCMDB:service=Ports Management Services** and click the link to open the Operations page.

   c. Locate the **PortsDetails** operation and click **Invoke**. Make a note of the HTTPS with client authentication port number. The default is 8444 and it should be enabled.

   d. Return to the Operations page.

   e. To map the ucmdb-api connector to the mutual authentication mode, invoke the **mapComponentToConnectors** method with the following parameters:

      - **componentName**: ucmdb-api
      - **isHTTPSWithClientAuth**: true
      - All other flags: false

      The following message is displayed:

      Operation succeeded. Component ucmdb-api is now mapped to: HTTPS_CLIENT_AUTH ports.

   f. Return to the Operations page.

2. Make sure the JRE that runs the UCMDB-API client has a keystore containing a client certificate.

3. Export the UCMDB-API client certificate from its keystore.

4. Import the exported UCMDB-API client certificate to the UCMDB Server Truststore.

   a. On the UCMDB machine, copy the created UCMDB-API client certificate file to the following directory on UCMDB:

      C:\HP\UCMDB\UCMDBServer\conf\security

   b. Run the following command:

```
C:\HP\UCMDB\UCMDBServer\bin\jre\bin\keytool.exe -import -v -keystore
C:\HP\UCMDB\UCMDBServer\conf\security\server.truststore -file <exported
UCMDB-api client certificate> - alias ucmdb-api
```

c. Enter the UCMDB Server Truststore password (default **hppass**).

d. When asked, **Trust this certificate?**, press **y** and then **Enter**.

e. Make sure the output **Certificate** was added to the keystore.

5. Export the UCMDB server certificate from the server keystore.

   a. On the UCMDB machine, run the following command:

```
C:\HP\UCMDB\UCMDBServer\bin\jre\bin\keytool.exe -export -alias hpcert
-keystore
C:\HP\UCMDB\UCMDBServer\conf\security\server.keystore
-file C:\HP\UCMDB\conf\security\server.cert
```

   b. Enter the UCMDB Server Truststore password (default **hppass**).

   c. Verify that the certificate is created in the following directory:

   C:\HP\UCMDB\UCMDBServer\conf\security\server.cert

6. Import the exported UCMDB certificate to the JRE of the UCMDB-API client truststore.

7. Restart the UCMDB Server and the UCMDB-API client.

8. To connect from the UCMDB-API client to UCMDB-API server, use the following code:

```
UcmdbServiceProvider provider = UcmdbServiceFactory.getServiceProvider
("https", <SOME_HOST_NAME>, <HTTPS_WITH_CLIENT_AUTH_PORT_NUMBER
(default:8444>));
UcmdbService ucmdbService = provider.connect(provider.createCertificateCrede
ntials(<TheClientKeystore.
e.g: "c:\\client.keystore">, <KeystorePassword>), provider.createClientConte
xt(<ClientIdentification>));
```

Configure CAC Support on UCMDB

This section describes how to configure Common Access Card (CAC) support on UCMDB.

Note: CAC support is only available when using Internet Explorer 8, 9, or 10.

1. Import the root CA and any intermediate certificates into the UCMDB Server Truststore as follows:

   a. On the UCMDB machine, copy the certificate files to the following directory on UCMDB:

      C:\HP\UCMDB\UCMDBServer\conf\security

      Note: If your certificate is in Microsoft p7b format, you may need to convert it to PEM format.

   b. For each certificate, run the following command:

      ```
      C:\HP\UCMDB\UCMDBServer\bin\jre\bin\keytool.exe -import -v -keystore
      C:\HP\UCMDB\UCMDBServer\conf\security\server.truststore -file
      <certificate> - alias <certificate alias>
      ```

   c. Enter the UCMDB Server Truststore password (default **hppass**).

   d. When asked, **Trust this certificate?**, press **y** and then **Enter**.

   e. Make sure the output **Certificate** was added to the keystore.

2. Open the JMX console by launching the Web browser and entering the Server address, as follows: http://<UCMDB Server Host Name or IP>:8080/jmx-console.

   You may have to log in with a user name and password.

3. Under UCMDB, click **UCMDB:service=Ports Management Services** to open the Operations page.

   - (optional) Click **ComponentsConfigurations**. Do the following:
     - Set **HTTPSSetPort** to **8444** and click **Invoke**.
     - Click **Back to MBean**.
   - Click **mapComponentToConnectors**. Do the following:
     - In the mapComponentToConnectors service, set **componentName** to **ucmdb-ui**.
     - Set only **isHTTPSWithClientAuth** to **true**, and click **Invoke**.
     - Click **Back to MBean**.
     - In the mapComponentToConnectors service, set **componentName** to **root**.
     - Set only **isHTTPSWithClientAuth** to **true**, and click **Invoke**.

4. Under UCMDB, click **UCMDB:service=Security Services** to open the Operations page. In the **loginWithCAC** service, do the following:

   - Set **loginWithCAC** to **true**, and click **Invoke**.
   - Click **Back to MBean**.
   - (optional) Click **usernameField** to specify the field from the certificate that will be used by UCMDB to extract a username, and click **Invoke**.

     Note: If you do not specify a field, the default of PRINCIPAL_NAME_FROM_SAN_FIELD is used.

   - Click **Back to MBean**.
   - Click **pathToCRL** to set a path to an to an offline Certificate Revocation List (CRL) to be used if the online list (from the certificate) is not available, and click **Invoke**.

     Note: When you are working with a local CRL and there is a working Internet connection to the UCMDB server, the local CRL is used. The validation of any certificate (even if it is not revoked) fails in the following situations:

     - if the CRL path is set but the CRL file itself is missing
     - if the CRL is expired
     - if the CRL has an incorrect signature

     If you do not set the path to an offline CRL and the UCMDB server cannot access the online CRL, all certificates that contain a CRL or OCSP URL are rejected (since the URL cannot be accessed, the revocation check fails). To give the UCMDB server access to the Internet, uncomment the following lines in the **wrapper.conf** file and provide a valid proxy and port:

     ```
     #wrapper.java.additional.40=-Dhttp.proxyHost=<PROXY_ADDR>
     #wrapper.java.additional.41=-Dhttp.proxyPort=<PORT>
     #wrapper.java.additional.42=-Dhttps.proxyHost=<PROXY_ADDR>
     #wrapper.java.additional.43=-Dhttps.proxyPort=<PORT>
     ```

   - Click **Back to MBean**.
   - (optional) Set **onlyCACCerts** to **true**, and click **Invoke**.

     Set this operation to **true** to accept only certificates that come from a physical CAC device.

You should now be able to log into UCMDB with https://<UCMDB Server Host Name or IP>.<domainname>:8444.

5. Configure UCMDB to use LW-SSO authentication and restart the UCMDB Server.

For details on LW-SSO authentication, see"Enabling Login to HP Universal CMDB with LW-SSO" on page 95.

Change the Server Keystore Password

After installing the Server, the HTTPS port is open and the store is secured with a weak password (the default **hppass**). If you intend to work with SSL only, you must change the password.

The following procedure explains how to change the **server.keystore** password only. However, you should perform the same procedure for changing the **server.truststore** password.

Note: You must perform every step in this procedure.

1. Start the UCMDB Server.
2. Execute the password change in the JMX console:
   a. Launch the Web browser and enter the Server address, as follows: **http://<UCMDB Server Host Name or IP>:8080/jmx-console**.

      You may have to log in with a user name and password.

   b. Under UCMDB, click **UCMDB:service=Security Services** to open the Operations page.

   c. Locate and execute the **changeKeystorePassword** operation.

      This field must not be empty and must be at least six characters long. The password is changed in the database only.
3. Stop the UCMDB Server.
4. Run commands.

   From **C:\hp\UCMDB\UCMBServer\bin\jre\bin**, run the following commands:

   a. Change the store password:

      keytool -storepasswd -new <new_keystore_pass> -keystore C:\hp\UCMDB\UCMDBServer\conf\security\server.keystore -storepass <current_keystore_pass>

   b. The following command displays the inner key of the keystore. The first parameter is the alias. Save this parameter for the next command:

keytool -list -keystore C:\hp\UCMDB\UCMDBServer\conf\security\server.keystore

c. Change the key password (if the store is not empty):

keytool -keypasswd -alias <alias> -keypass <currentPass> -new <newPass> -keystore C:\hp\UCMDB\UCMDBServer\conf\security\server.keystore

d. Enter the new password.

5. Start the UCMDB Server.
6. Repeat the procedure for the Server truststore.

Enable or Disable HTTP/HTTPS Ports

You can enable or disable the HTTP and HTTPS ports from within the user interface or from the JMX console.

To enable or disable the HTTP/HTTPS ports from within the user interface:

1. Log on to HP Universal CMDB.
2. Select **Administration > Infrastructure Settings**.
3. Enter either **http** or **https** in the **Filter** (by Name) box to display the HTTP settings.
   - **Enable HTTP(S) connections**. **True**: the port is enabled. **False**: the port is disabled.
4. Restart the server to apply the change.

> **Caution:** The HTTPS port is open by default; closing this port prevents **Server_ Management.bat** from functioning.

To enable or disable the HTTP/HTTPS ports from the JMX console:

1. Launch a Web browser and enter the following address: http://localhost.<domain_ name>:8080/jmx-console.
2. Enter the JMX console authentication credentials. The default credentials are:
   - Login name = **sysadmin**
   - Password = **sysadmin**
3. Locate **UCMDB:service=Ports Management Services** and click the link to open the Operations page.

4. To enable or disable the HTTP port, locate the **HTTPSetEnable** operation and set the value.
   - **True:** the port is enabled.
   - **False:** the port is disabled.
5. To enable or disable the HTTPS port, locate the **HTTPSSetEnable** operation and set the value.
   - **True:** the port is enabled.
   - **False:** the port is disabled.
6. To enable or disable the HTTPS port with client authentication, locate the **HTTPSClientAuthSetEnable** operation and set the value.
   - **True:** the port is enabled.
   - **False:** the port is disabled.

Map the UCMDB Web Components to Ports

You can configure the mapping of each UCMDB component to the available ports from the JMX console.

To view the current component configurations:

1. Launch a Web browser and enter the following address: **http://localhost.<domain_name>:8080/jmx-console**.
2. Enter the JMX console authentication credentials. The default credentials are:

   Login name = **sysadmin**

   Password = **sysadmin**
3. Locate **UCMDB:service=Ports Management Services** and click the link to open the Operations page.
4. Locate the **ComponentsConfigurations** method and click **Invoke**.
5. For each component, the valid ports and current mapped ports are displayed.

To map the components:

1. Locate **UCMDB:service=Ports Management Services** and click the link to open the Operations page.
2. Locate the **mapComponentToConnectors** method.

3. Enter a component name in the Value box. Select **True** or **False** for each of the ports corresponding to your selection. Click **Invoke**. The selected component is mapped to the selected ports. You can find the component names by invoking the **serverComponentsNames** method.

4. Repeat the process for each relevant component.

Note:

- Every component must be mapped to at least one port. If you do not map a component to any port, it is mapped by default to the HTTP port.
- If you map a component to both the HTTPS port and the HTTPS port with client authentication, only the client authentication option is mapped (the other option is redundant in this case).
- If you set **isHTTPSWithClientAuth** to **True** for the UCMDB UI component, you must also set it to **True** for the root component.

You can also change the value assigned to each of the ports.

To set values for the ports:

1. Locate **UCMDB:service=Ports Management Services** and click the link to open the Operations page.

2. To set a value for the HTTP port, locate the **HTTPSetPort** method and enter a value in the **Value** box. Click **Invoke**.

3. To set a value for the HTTPS port, locate the **HTTPSSetPort** method and enter a value in the **Value** box. Click **Invoke**.

4. To set a value for the HTTPS port with client authentication, locate the **HTTPSClientAuthSetPort** method and enter a value in the **Value** box. Click **Invoke**.

Configure Configuration Manager to Work with UCMDB Using SSL

You can configure Configuration Manager to work with UCMDB using Secure Sockets Layer (SSL). The SSL connector on port 8443 is enabled by default in UCMDB.

1. Go to **<UCMDB installation directory>\bin\jre\bin** and run the following command:

```
keytool -export -alias hpcert -keystore <UCMDB_server_directory>
\conf\security\server.keystore -storepass hppass -file <certificatefile>
```

2. Copy the certificate file to a temporary location on the local Configuration Manager machine.

3. Perform a new installation or reconfigure an existing installation of Configuration Manager. For instructions, see the relevant sections in the interactive *HP Universal CMDB Deployment Guide*.

   In the UCMDB configuration screen, set the protocol to HTTPS, and choose the certificate file that you copied in step 2.

4. Copy **hpcert.cer** to the server machine in the **<*Configuration_Manager_installation_directory*>\java\windows\x86_64\bin** folder.

5. On the server machine, import the certificate into the trust store (cacerts) using the keytool utility with the following command:

   ```
   <Configuration_Manager_installation_directory>\java\bin\keytool.exe -import -alias hp -file hpcert.cer -keystore <Configuration_Manager_installation_directory>\java\windows\x86_64\lib\security\cacerts
   ```

6. Copy **hpcert.cer** to the server machine in the **<*Configuration_Manager_installation_directory*>\java\ windows\x86_64\lib\security** folder.

7. Create a server keystore (JKS type) with a self-signed certificate and matching private key. **From the <*Configuration_Manager_installation_directory*>\java\windows\x86_64\bin** folder, run the following command:

   ```
   keytool –genkey –alias tomcat –keyalg RSA –keystore <Configuration_Manager_installation_directory>\java\windows\x86_64\lib\security\tomcat.keystore
   ```

   a. Enter a keystore password.

   b. For the question: `What is your first and last name?`, enter the Configuration Manager Web server name and enter the other parameters according to your organization.

   c. Enter a key password. The key password MUST be the same as the keystore password. A JKS keystore is created named **tomcat.keystore**, with a server certificate named **hpcert**.

8. Modify the **server.xml** file as follows:

   a. Open the server.xml file, located in **<*Configuration_Manager_installation_directory*>\servers\server-0\conf** folder. Locate the section beginning with:

   ```
   Connector port="8143"
   ```

   which appears as a comment. Activate the script by removing the comment character and add the following lines:

   ```
   keystoreFile="<Configuration_Manager_installation_directory>\java\windows\x86_64\lib\security\tomcat.keystore"
   keystorePass="password"
   truststoreFile="<Configuration_Manager_installation_
   ```

```
directory>\java\windows\x86_64\lib\security\cacerts"
truststorePass="changeit" />
```

b. Comment out the following line:

```
<Listener className="org.apache.catalina.core.AprLifecycleListener" SSLEn
gine="on" />
```

9. Restart the server.

To configure Configuration Manager to work with other products (such as load balancers) using SSL, import the security certificate of the product to the Configuration Manager truststore (default jre truststore) by running the following command:

```
<CM_JAVA_HOME>\bin\keytool -import -trustcacerts -alias <alias> -keystore
<CM_JAVA_HOME>\lib\security\cacerts -storepass changeit -file <certificatefile>
```

Enable the UCMDB KPI Adapter to be used with SSL

You can configure the UCMDB KPI adapter information to be sent using Secure Sockets Layer (SSL).

1. Export the Configuration Manager certificate:

```
<CM_JAVA_HOME>\bin\keytool -export -alias tomcat -keystore
<CM_JAVA_HOME>\lib\security\tomcat.keystore -storepass
<keystore pass> -file <certificate file name>
```

2. Import the certificate that you exported from Configuration Manager into the UCMDB truststore as follows:

```
<UCMDB server dir>\bin\jre\bin keytool -import -trustcacerts
-alias tomcat -keystore <UCMDB server dir>\bin\jre\lib
\security\cacerts -storepass changeit -file <certificatefile>
```

3. Import the certificate that you exported from Configuration Manager into the Probe's truststore as follows:

a. Open the command prompt and run the command:

```
<DataFlowProbe dir>\bin\jre\bin\keytool.exe -import -v -keystore
<DataFlowProbe dir>\conf\security\hpprobeTrustStore.jks -file
<certificatefile> -alias tomcat
```

b. Enter the keystore password: logomania

c. When asked **Trust this certificate?**, press **y** and then **Enter**.

The following message is displayed:

Certificate was added to keystore.

For additional details about hardening the Data Flow Probe, see "Data Flow Probe Hardening" on page 70

4. Restart UCMDB, the Data Flow Probe, and Configuration Manager.

Configure SSL Support for the UCMDB Browser

Note: The instructions provided here are relevant to UCMDB Browser version 1.95. If you are using a later version of the UCMDB Browser that has been upgraded separately from the rest of the UCMDB product suite, see the section on configuring SSL support in the *HP Universal CMDB Browser Installation and Configuration Guide* for that version.

To install and configure SSL support on Tomcat:

1. Create a keystore file to store the server's private key and self-signed certificate by executing one of the following commands:
   - For Windows: **%JAVA_HOME%\bin\keytool -genkey -alias tomcat -keyalg RSA**
   - For Unix: **$JAVA_HOME/bin/keytool -genkey -alias tomcat -keyalg RSA**

   For both commands, use the password value **changeit** (for all other fields in the console dialog that opens, you can use any value).

2. Remove comments from the entry **SSL HTTP/1.1 Connector** in **$CATALINA_BASE/conf/server.xml**, where **$CATALINA_BASE** is the directory in which you installed Tomcat.

   Note: For a full description on how to configure **server.xml** to use SSL, see the Apache Tomcat official site: http://tomcat.apache.org/tomcat-7.0-doc/ssl-howto.html

3. Restart the Tomcat server.

To use the HTTPS protocol for connection to the UCMDB server:

1. In **ucmdb_browser_config.xml**, assign the value **https** to the tag**<protocol>** and assign the UCMDB server HTTPS port value (8443 by default) to the tag **<port>.**

2. Download the UCMDB Server public certificate to the UCMDB Browser machine (if you use

SSL on the UCMDB-Server, the UCMDB administrator can provide you with this certificate), and import it into the **cacerts** trust store on the JRE that is going to connect to the server by executing the following command:

"%JAVA_HOME%\bin\keytool" -import -alias ucmdb -trustcacerts -file <UCMDB-Server-certificate-file> -keystore "%JAVA_HOME%\jre\lib\security\cacerts"

where **<UCMDB-Server-certificate-file>** is the full path to the UCMDB Server public certificate file.

3. Restart the Tomcat server.

Chapter 3: Using a Reverse Proxy

This section describes the security ramifications of reverse proxies and contains instructions for using a reverse proxy with HP Universal CMDB and Configuration Manager. Security aspects of a reverse proxy are discussed but not other aspects such as caching and load balancing.

This chapter includes:

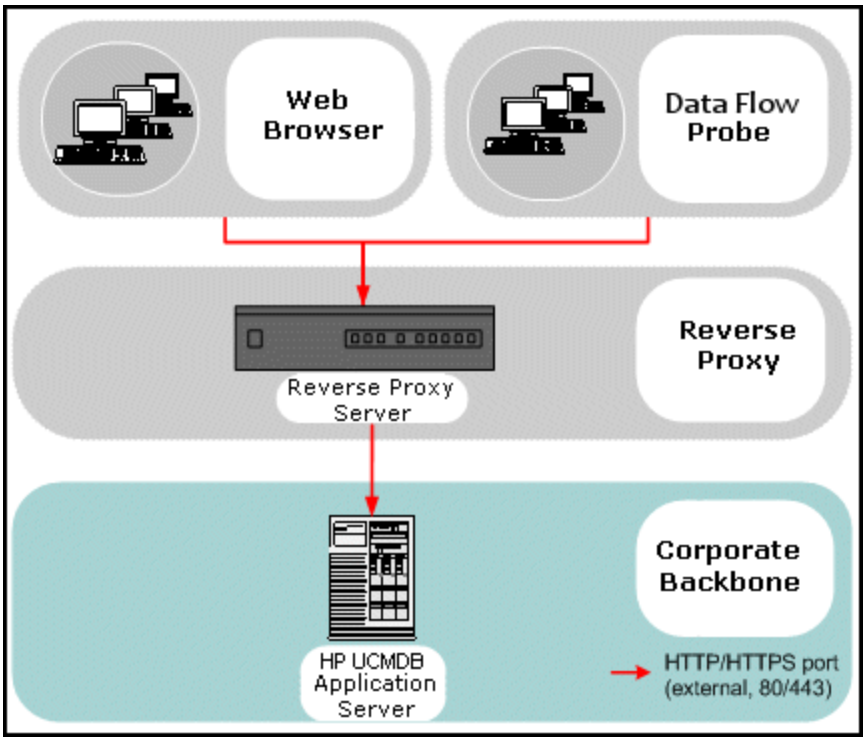
Reverse Proxy Overview

A reverse proxy is an intermediate server that is positioned between the client machine and the Web servers. To the client machine, the reverse proxy appears to be a standard Web server that serves the client machine's HTTP protocol requests.

The client machine sends ordinary requests for Web content, using the name of the reverse proxy instead of the name of a Web server. The reverse proxy sends the request to one of the Web servers. Although the response is sent back to the client machine by the reverse proxy, it appears to the client machine as if it is being sent by the Web server.

It is possible to have multiple reverse proxies, with different URLs, representing the same UCMDB/CM instance. Alternatively, a single reverse proxy server can be used to access several UCMDB/CM servers, by setting different root contexts for each UCMDB/CM server.

HP Universal CMDB and Configuration Manager support a reverse proxy in a DMZ architecture. The reverse proxy is an HTTP mediator between the Data Flow Probe and the Web client and the HP Universal CMDB/CM server.



Note:

- Different types of reverse proxies require different configuration syntaxes. For an example of an Apache 2.0.x reverse proxy configuration, see "Example: Apache 2.0.x Configuration " on page 41.
- It is only necessary to configure the front-end URL setting when creating a direct link to a report using the Scheduler.

Security Aspects of Using a Reverse Proxy Server

A reverse proxy server functions as a bastion host. The proxy is configured to be the only machine addressed directly by external clients, and thus obscures the rest of the internal network. Using a reverse proxy enables the application server to be placed on a separate machine in the internal network.

This section discusses the use of a DMZ and reverse proxy in a back-to-back topology environment.

The following are the main security advantages of using a reverse proxy in such an environment:

- No DMZ protocol translation occurs. The incoming protocol and outgoing protocol are identical (only a header change occurs).

- Only HTTP access to the reverse proxy is allowed, which means that stateful packet inspection firewalls can better protect the communication.
- A static, restricted set of redirect requests can be defined on the reverse proxy.
- Most of the Web server security features are available on the reverse proxy (authentication methods, encryption, and so on).
- The reverse proxy screens the IP addresses of the real servers as well as the architecture of the internal network.
- The only accessible client of the Web server is the reverse proxy.
- This configuration supports NAT firewalls (as opposed to other solutions).
- The reverse proxy requires a minimal number of open ports in the firewall.
- The reverse proxy provides good performance compared to other bastion solutions.

Configure a Reverse Proxy

This section describes how to configure a reverse proxy. As of UCMDB version 10.01, no configuration is necessary in UCMDB. On the reverse proxy side, edit the configuration file according to the reverse proxy's documentation. For an example, see "Example: Apache 2.0.x Configuration " on the next page.

For scheduled jobs created prior to UCMDB version 10.01, you also need to set the configuration in UCMDB as follows:

Configure a Reverse Proxy Using Infrastructure Settings

The following procedure explains how to access Infrastructure Settings to configure a reverse proxy. This configuration is only necessary when creating a direct link to a report using the Scheduler.

To configure a reverse proxy:

1. Select **Administration > Infrastructure Settings > General Settings** category.
2. Change the **Frontend URL** setting. Enter the address, for example, **https://my_proxy_server:443/**.

Note: After making this change, you cannot access the HP Universal CMDB server directly through a client. To change the reverse proxy configuration, use the JMX console on the server machine. For details, see "Configure a Reverse Proxy Using the JMX Console" below.

Configure a Reverse Proxy Using the JMX Console

You can make changes to the reverse proxy configuration by using the JMX console on the HP Universal CMDB server machine. This configuration is only necessary when creating a direct link

to a report using the Scheduler.

To change a reverse proxy configuration:

1. On the HP Universal CMDB server machine, launch the Web browser and enter the following address:

   http://<machine name or IP address>.<domain_name>:8080/jmx-console

   where **<machine name or IP address>** is the machine on which HP Universal CMDB is installed. You may have to log in with the user name and password.

2. Click the **UCMDB-UI > UCMDB-UI:name=UI Server frontend settings** link.

   In the **setUseFrontendURLBySettings** field, enter the server proxy URL, for example, `https://my_proxy_server:443/`.

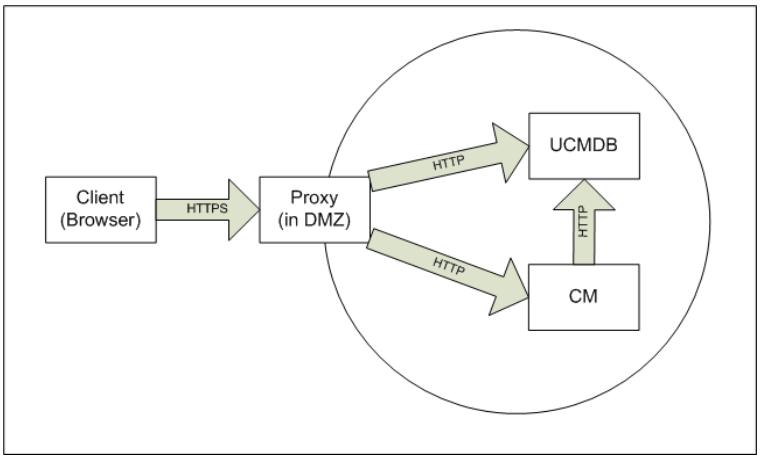
3. Click **Invoke**.

4. To see the value of this setting, use the **showFrontendURLInSettings** method.

Example: Apache 2.0.x Configuration

This section describes a sample configuration file that supports the use of an Apache 2.0.x reverse proxy in a case where both Data Flow Probes and application users connect to HP Universal CMDB.

The following diagram illustrates the configuration process for a reverse proxy for Configuration Manager and the UCMDB.



Note:

- In this example, the HP Universal CMDB machine's DNS name and port is UCMDB_ server.
- In this example, the HP Configuration Manger's DNS name and port is UCMDB_CM_

server.

- Only users with a knowledge of Apache administration should make this change.

1. Open the **<Apache machine root directory>\Webserver\conf\httpd.conf** file.
2. Enable the following modules:
   - **LoadModule proxy_module modules/mod_proxy.so**
   - **LoadModule proxy_http_module modules/mod_proxy_http.so**
3. Add the following lines to the **httpd.conf** file:

```
ProxyRequests off
<Proxy *>
     Order deny,allow
     Deny from all
     Allow from all
</Proxy>

ProxyPass /mam http://UCMDB_server/mam
ProxyPassReverse /mam http://UCMDB_server/mam
ProxyPass /mam_images http://UCMDB_server/mam_images
ProxyPassReverse /mam_images http://UCMDB_server/mam_images
ProxyPass /mam-collectors http://UCMDB_server/mam-collectors
ProxyPassReverse /mam-collectors http://UCMDB_server/mam-collectors
ProxyPass /ucmdb http://UCMDB_server/ucmdb
ProxyPassReverse /ucmdb http://UCMDB_server/ucmdb
ProxyPass /site http://UCMDB_server/site
ProxyPassReverse /site http://UCMDB_server/site
ProxyPass /ucmdb-ui http://UCMDB_server/ucmdb-ui
ProxyPassReverse /ucmdb-ui http://UCMDB_server/ucmdb-ui
ProxyPass /status http://UCMDB_server/status
ProxyPassReverse /status http://UCMDB_server/status
ProxyPass /jmx-console http://UCMDB_server/jmx-console
ProxyPassReverse /jmx-console http://UCMDB_server/jmx-console
ProxyPass /axis2 http://UCMDB_server/axis2
ProxyPassReverse /axis2 http://UCMDB_server/axis2
ProxyPass /icons http://UCMDB_server/icons
ProxyPassReverse /icons http://UCMDB_server/icons
ProxyPass /ucmdb-api http://UCMDB_server/ucmdb-api
ProxyPassReverse /ucmdb-api http://UCMDB_server/ucmdb-api
ProxyPass /ucmdb-docs http://UCMDB_server/ucmdb-docs
ProxyPassReverse /ucmdb-docs http://UCMDB_server/ucmdb-docs
ProxyPass /ucmdb-api/8.0 http://UCMDB_server/ucmdb-api/8.0
ProxyPassReverse /ucmdb-api/8.0 http://UCMDB_server/ucmdb-api/8.0
ProxyPass /cm http://UCMDB_Server/cm
ProxyPassReverse /cm http://UCMDB_Server /cm
```

```
ProxyPass /cnc http://UCMDB_CM_server/cnc
ProxyPassReverse /cnc http://UCMDB_CM_server/cnc
ProxyPass /docs http://UCMDB_CM_server/docs
ProxyPassReverse /docs http://UCMDB_CM_server/docs
ProxyPass /ucmdb-browser http://UCMDB_CM_server/ucmdb-browser
ProxyPassReverse /ucmdb-browser http://UCMDB_CM_server/ucmdb-browser
```

4. Save your changes.

Connect the Data Flow Probe by Reverse Proxy or Load Balancer Using Mutual Authentication

Perform the following procedure to connect the Data Flow Probe through a reverse proxy or load balancer using mutual authentication. This procedure applies to the following configuration:

- Mutual SSL authentication between the probe and a reverse proxy or load balancer based on a client certificate provided by the probe and required by the reverse proxy or load balancer.
- A regular SSL connection between the reverse proxy or load balancer and the UCMDB server.

Note: The following instructions use the **cKeyStoreFile** keystore as the Probe keystore. This is a predefined client keystore that is part of the Data Flow Probe installation and contains a self-signed certificates. For details, see "Server and Data Flow Probe Default Keystore and Truststore" on page 86.

It is recommended to create a new, unique keystore containing a newly generated private key. For details, see "Create a Keystore for the Data Flow Probe" on page 85.

Obtain a Certificate from a Certification Authority

Obtain the CA root certificate and import it into the following locations:

- the Data Flow Probe truststore
- the Data Flow Probe JVM cacerts
- the UCMDB server truststore
- the reverse proxy truststore

1. Import the CA root certificate into the Data Flow Probe truststore.

   a. Place the CA root certificate in the following directory: <Data Flow Probe installation directory>\conf\security\<certificate file name>.

   b. Import the CA root certificate into the Data Flow truststore by running the following script:

```
<Data Flow Probe installation directory>\bin\jre\bin\keytool.exe -import
-trustcacerts -alias <YourAlias> -file C:\hp\UCMDB\DataFlowProbe\conf\sec
urity\<certificate file name> -keystore <Data Flow Probe installation dir
ectory>\conf\security\hpprobeTrustStore.jks
```

The default password is: **logomania**.

2. Import the CA root certificate into the Data Flow Probe JVM cacerts by running the following script:

```
<Data Flow Probe installation directory>\bin\jre\bin\keytool.exe -import -tr
ustcacerts -alias <YourAlias> -file <Data Flow Probe installation directory>
\conf\security\<certificate file name> -keystore <Data Flow Probe installati
on directory>\bin\jre\lib\security\cacerts
```

The default password is: **changeit**.

3. Import the CA root certificate into the UCMDB truststore.

   a. Place the CA root certificate in the following directory: <UCMDB installation directory>\conf\security\<certificate file name>.

   b. Import the CA root certificate into the UCMDB truststore by running the following script:

```
<UCMDB installation directory>\bin\jre\bin\keytool.exe -import -trustcace
rts -alias <YourAlias> -file <UCMDB installation directory>\conf\security
\<ceritificate file name> -keystore <UCMDB installation directory>\conf\s
ecurity\sever.truststore
```

   The default password is: **hppass**.

4. Import the CA root certificate into the reverse proxy truststore. This is step is vendor dependent.

Convert the Certificate to a Java Keystore

Obtain the client certificate (and private key) for the Data Flow Probe from your Certificate Authority (CA) in the PFX/PKCS12 format and convert it to a Java keystore by running the following script:

```
<Data Flow Probe installation directory>\bin\jre\bin\keytool.exe -importkeystore
-srckeystore <PFX keystore full path> -destkeystore <new destination keystore fu
ll path> -srcstoretype PKCS12
```

You will be prompted for the source and destination keystore passwords.

For the source keystore password, use the same password that was used when exporting the PFX keystore.

The default destination keystore password for the Data Flow Probe keystore is: **logomania**.

Note: If you entered a different destination keystore password from the default Data Flow Probe keystore password (logomania), you will need to supply the new password in encrypted

format in the **<Data Flow Probe installation directory>\conf\ssl.properties** file (javax.net.ssl.keyStorePassword). For details, see "Encrypt the Probe Keystore and Truststore Passwords" on page 85.

Place new keystore in the following directory: **<Data Flow Probe installation directory>\conf\security**.

Caution: Do not overwrite the **hpprobeKeyStore.jks** file.

Change the SSL Properties File to Use the Newly Created Keystore

Set the keystore containing the client certificate in the **<Data Flow Probe installation directory>\conf\ssl.properties** file to **javax.net.ssl.keyStore**.

If the password to your keystore is not the default Data Flow Probe keystore password (logomania), then update the **javax.net.ssl.keyStorePassword** after encrypting it. For detail on encrypting the password, see "Encrypt the Probe Keystore and Truststore Passwords" on page 85.

Review the Data Flow Probe Configuration

Edit the **<Data Flow Probe installation directory>\conf\DataFlowProbe.properties** file as follows:

```
appilog.agent.probe.protocol = HTTPS

serverName = <reverse proxy server address>

serverPortHttps = <the HTTPS port that the reverse proxy listens to in order to redirect requests to the UCMDB>
```

Configure UCMDB to Work Using SSL

For details, see "Enabling Secure Sockets Layer (SSL) Communication" on page 17.

If the UCMDB server certificate is created by the same CA that created the rest of the certificates in this procedure, the reverse proxy or load balancer trusts the UCMDB certificate.

Configure CAC Support on UCMDB by Reverse Proxy

This section describes how to configure Common Access Card (CAC) support on UCMDB using a reverse proxy.

1. Open the JMX console by launching the Web browser and entering the Server address, as follows: http://<UCMDB Server Host Name or IP>:8080/jmx-console.

   You may have to log in with a user name and password.

2. Under UCMDB, click **UCMDB:service=Ports Management Services** to open the Operations

page.

- (optional) Click **ComponentsConfigurations**. Do the following:
  - Set **HTTPSetPort** to **8080** and click **Invoke**.
  - Click **Back to MBean**.
- Click **mapComponentToConnectors**. Do the following:
  - In the mapComponentToConnectors service, set **componentName** to **ucmdb-ui**.
  - Set only **isHTTP** to **true**, and click **Invoke**.
  - Click **Back to MBean**.
  - In the mapComponentToConnectors service, set **componentName** to **root**.
  - Set only **isHTTP** to **true**, and click **Invoke**.

3. Under UCMDB, click **UCMDB:service=Security Services** to open the Operations page.
   - Set **loginWithCAC** to **true**, and click **Invoke**.
   - Click **Back to MBean**.
   - Set **withReverseProxy** to **true**, and click **Invoke**.

     This setting tells the UCMDB server to extract from the UCMDB_SSL_CLIENT_CERT header the user name to be used in UCMDB and the certificate to be used for authentication.
   - Click **Back to MBean**.
   - (optional) Set **onlyCACCerts** to **true**, and click **Invoke**.

     Set this operation to **true** to accept only certificates that come from a physical CAC device.
4. Restart the UCMDB Server.

Example: Apache 2.4.4 Configuration

This section describes a sample configuration file for Apache 2.4.4 (in the **<Apache machine root directory>\Webserver\conf\httpd.conf** file:

```
LoadModule headers_module modules/mod_headers.so
LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_html_module modules/mod_proxy_html.so
LoadModule proxy_http_module modules/mod_proxy_http.so
LoadModule socache_shmcb_module modules/mod_socache_shmcb.so
LoadModule ssl_module modules/mod_ssl.so
ServerName Apache_Server_Name:80
```

```
Include conf/extra/httpd-ssl.conf
```

This section describes a sample configuration file for Apache 2.4.4 with SSL (in the **<Apache machine root directory>\Webserver\conf\extra\httpd-ssl.conf** file:

```
Listen 8443<VirtualHost _default_:8443>
ServerName Apache_Server_Name:8443
SSLCACertificateFile "c:/Apache24/conf/ssl.crt"
SSLCARevocationFile "c:/Apache24/conf/ssl.crl"
#SSLCARevocationCheck chain|leaf|none
SSLCARevocationCheck leaf
RequestHeader set UCMDB_SSL_CLIENT_CERT %{SSL_CLIENT_CERT}e


ProxyRequests off
<Proxy *>
     Order deny,allow
     Deny from all
     Allow from all
</Proxy>

ProxyPass /mam http://UCMDB_server/mam
ProxyPassReverse /mam http://UCMDB_server/mam
ProxyPass /mam_images http://UCMDB_server/mam_images
ProxyPassReverse /mam_images http://UCMDB_server/mam_images
ProxyPass /mam-collectors http://UCMDB_server/mam-collectors
ProxyPassReverse /mam-collectors http://UCMDB_server/mam-collectors
ProxyPass /ucmdb http://UCMDB_server/ucmdb
ProxyPassReverse /ucmdb http://UCMDB_server/ucmdb
ProxyPass /site http://UCMDB_server/site
ProxyPassReverse /site http://UCMDB_server/site
ProxyPass /ucmdb-ui http://UCMDB_server/ucmdb-ui
ProxyPassReverse /ucmdb-ui http://UCMDB_server/ucmdb-ui
ProxyPass /status http://UCMDB_server/status
ProxyPassReverse /status http://UCMDB_server/status
ProxyPass /jmx-console http://UCMDB_server/jmx-console
ProxyPassReverse /jmx-console http://UCMDB_server/jmx-console
ProxyPass /axis2 http://UCMDB_server/axis2
ProxyPassReverse /axis2 http://UCMDB_server/axis2
ProxyPass /icons http://UCMDB_server/icons
ProxyPassReverse /icons http://UCMDB_server/icons
ProxyPass /ucmdb-api http://UCMDB_server/ucmdb-api
ProxyPassReverse /ucmdb-api http://UCMDB_server/ucmdb-api
ProxyPass /ucmdb-docs http://UCMDB_server/ucmdb-docs
ProxyPassReverse /ucmdb-docs http://UCMDB_server/ucmdb-docs
ProxyPass /ucmdb-api/8.0 http://UCMDB_server/ucmdb-api/8.0
ProxyPassReverse /ucmdb-api/8.0 http://UCMDB_server/ucmdb-api/8.0
ProxyPass /cm http://UCMDB_Server/cm
ProxyPassReverse /cm http://UCMDB_Server /cm
ProxyPass /cnc http://UCMDB_CM_server/cnc
ProxyPassReverse /cnc http://UCMDB_CM_server/cnc
```

```
ProxyPass /docs http://UCMDB_CM_server/docs
ProxyPassReverse /docs http://UCMDB_CM_server/docs
ProxyPass /ucmdb-browser http://UCMDB_CM_server/ucmdb-browser
ProxyPassReverse /ucmdb-browser http://UCMDB_CM_server/ucmdb-browser
SSLVerifyClient require
SSLVerifyDepth  10
SSLOptions+ExportCertData
```

Chapter 4: Data Flow Credentials Management

This chapter includes:

Data Flow Credentials Management Overview

To perform discovery or run integration, you must set up the credentials to access the remote system. Credentials are configured in the Data Flow Probe Setup window and saved in the UCMDB Server. For details, see the section describing the Data Flow Probe setup in the *HP Universal CMDB Data Flow Management Guide*.

Credentials storage is managed by the Confidential Manager component. For details, see "Confidential Manager" on page 103.

The Data Flow Probe can access the credentials using the Confidential Manager client. The Confidential Manager client resides on the Data Flow Probe and communicates with the Confidential Manager server, which resides on the UCMDB Server. Communication between the Confidential Manager client and the Confidential Manager server is encrypted, and authentication is required by the Confidential Manager client when it connects to the Confidential Manager server.

The Confidential Manager client's authentication on the Confidential Manager server is based on a LW-SSO component. Before connecting to the Confidential Manager server, the Confidential Manager client first sends an LW-SSO cookie. The Confidential Manager server verifies the cookie and upon successful verification, communication with the Confidential Manager client begins. For details about LW-SSO, see "Configure LW-SSO Settings" on page 54.

The communication between the Confidential Manager client and the Confidential Manager server is encrypted. For details about updating the encryption configuration, see "Configure Confidential Manager Communication Encryption " on page 54.

Caution: The Confidential Manager authentication uses the universal time defined on the computer (UTC). In order for the authentication to succeed, ensure that the universal time on the Data Flow probe and the UCMDB Server are the same. The server and probe may be located in different time zones, as UTC is independent of time zone or daylight savings time.

The Confidential Manager client maintains a local cache of the credentials. The Confidential Manager client is configured to download all credentials from the Confidential Manager server and store them in a cache. The credentials changes are automatically synchronized from Confidential Manager server on a continuous basis. The cache can be a file-system or in-memory cache, depending on the preconfigured settings. In addition, the cache is encrypted and cannot be accessed externally. For details about updating the cache settings, see "Configure the Confidential Manager Client's Cache Mode on the Probe" on page 59. For details about updating the cache encryption, see "Configure the Confidential Manager Client's Cache Encryption Settings on the Probe" on page 59.

For details on troubleshooting, see "Change Confidential Manager Client Log File Message Level" on page 62.

You can copy credentials information from one UCMDB server to another. For details, see "Export and Import Credential and Range Information in Encrypted Format" on page 61.

Note: The **DomainScopeDocument** (DSD) that was used for credentials storage on the Probe (in UCMDB version 9.01 or earlier) no longer contains any credentials-sensitive information. The file now contains a list of Probes and network range information. It also contains a list of credential entries for each domain, where each entry includes the credential ID and a network range (defined for this credential entry) only.

This section includes the following topics:

Basic Security Assumptions

Note the following security assumption:

You have secured the UCMDB Server and Probe JMX console to enable access to UCMDB system administrators only, preferably through localhost access only.

Data Flow Probe Running in Separate Mode

When the Probe Gateway and Manager run as separate processes, the Confidential Manager client component becomes part of the Manager process. Credentials information is cached and used by the Probe Manager only. To access the Confidential Manager server on the UCMDB system, the Confidential Manager client request is handled by the Gateway process and from there is forwarded to the UCMDB system.

This configuration is automatic when the Probe is configured in separate mode.

Keeping the Credentials Cache Updated

On its first successful connection to the Confidential Manager server, the Confidential Manager client downloads all relevant credentials (all credentials that are configured in the probe's domain). After the first successful communication, the Confidential Manager client retains continuous synchronization with the Confidential Manager server. Differential synchronization is performed at one-minute intervals, during which only differences between the Confidential Manager server and the Confidential Manager client are synchronized. If the credentials are changed on the UCMDB server side (such as new credentials being added, or existing credentials being updated or deleted),

the Confidential Manager client receives immediate notification from the UCMDB server and performs additional synchronization.

Synchronizing All Probes with Configuration Changes

For successful communication, the Confidential Manager client must be updated with the Confidential Manager server authentication configuration (LW-SSO init string) and encryption configuration (Confidential Manager communication encryption). For example, when the init string is changed on the server, the probe must know the new init string in order to authenticate.

The UCMDB server constantly monitors for changes in the Confidential Manager communication encryption configuration and Confidential Manager authentication configuration. This monitoring is done every 15 seconds; in case a change has occurred, the updated configuration is sent to the probes. The configuration is passed to the probes in encrypted form and stored on the probe side in secured storage. The encryption of configuration being sent is done using a symmetric encryption key. By default, the UCMDB server and Data Flow Probe are installed with same default symmetric encryption key. For optimal security, it is highly recommended to change this key before adding credentials to the system. For details, see "Generate or Update the Encryption Key" on page 63.

Note: Due to the 15 second monitoring interval, it is possible that the Confidential Manager client, on the Probe side, may not be updated with the latest configuration for a period of 15 seconds.

If you choose to disable the automatic synchronization of Confidential Manager communication and authentication configuration between the UCMDB server and the Data Flow Probe, each time you update the Confidential Manager communication and authentication configuration on the UCMDB server side, you should update all Probes with the new configuration as well. For details, see "Disable Automatic Synchronization of the Confidential Manager Client Authentication and Encryption Settings Between the Server and Probes" on page 56.

Secured Storage on the Probe

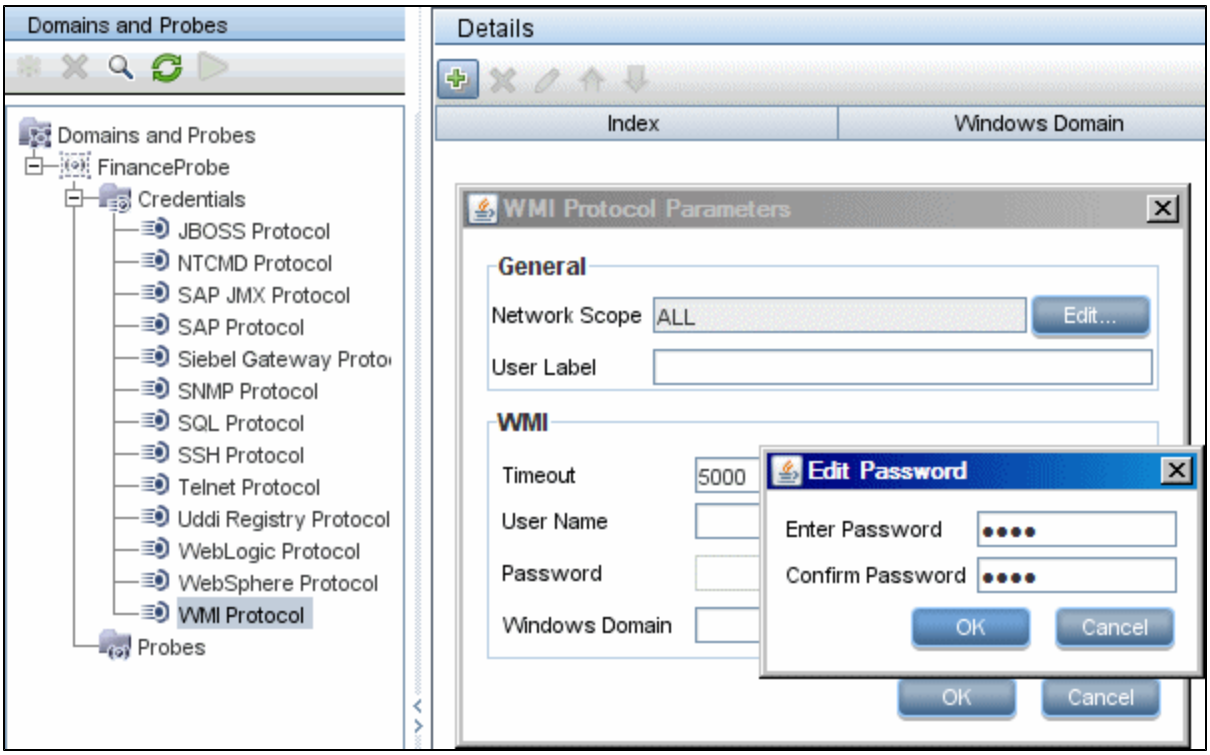
All sensitive information (such as the Confidential Manager communication and authentication configuration and the encryption key) is stored on the Probe in secure storage in the **secured_storage.bin** file, located in **C:\hp\UCMDB\DataFlowProbe\conf\security**. This secured storage is encrypted using DPAPI, which relies on the Windows user password in the encryption process. DPAPI is a standard method used to protect confidential data—such as certificates and private keys—on Windows systems. The Probe should always run under the same Windows user, so that even if the password is changed, the Probe can still read the information stored in secure storage.

Viewing Credentials Information

Note: This section deals with viewing credential information when the data direction is from the

CMDB to HP Universal CMDB

Passwords are not sent from the CMDB to the application. That is, HP Universal CMDB displays asterisks (*) in the password field, regardless of content:



Updating Credentials

Note: This section deals with updating credentials when the data direction is from HP Universal CMDB to the CMDB.

- The communication in this direction is not encrypted, therefore you should connect to the UCMDB Server using https\SSL, or ensure connection through a trusted network.

  Although the communication is not encrypted, passwords are not being sent as clear text on the network. They are encrypted using a default key and, therefore, it is highly recommended to use SSL for effective confidentiality in transit.

- You can use special characters and non-English characters as passwords.

Configure Confidential Manager Client Authentication and Encryption Settings

This task describes configuring the Confidential Manager Client Authentication and Encryption Settings on the UCMDB Server, and includes the following steps:

- "Configure LW-SSO Settings" below
- "Configure Confidential Manager Communication Encryption " below

Configure LW-SSO Settings

This procedure describes how to change the LW-SSO init string on the UCMDB server. This change is automatically sent to Probes (as an encrypted string), unless the UCMDB server is configured to not automatically do this. For details, see "Disable Automatic Synchronization of the Confidential Manager Client Authentication and Encryption Settings Between the Server and Probes" on page 56.

1. On the UCMDB server, launch the Web browser and enter the following address: **http://localhost:8080/jmx-console**.
2. Click **UCMDB-UI:name=LW-SSO Configuration** to open the JMX MBEAN View page.
3. Locate the **setInitString** method.
4. Enter a new LW-SSO init string.
5. Click Invoke.

Configure Confidential Manager Communication Encryption

This procedure describes how to change the Confidential Manager communication encryption settings on the UCMDB Server. These settings specify how the communication between the Confidential Manager client and the Confidential Manager server is encrypted. This change is automatically sent to Probes (as an encrypted string), unless the UCMDB server is configured to not automatically do this. For details, see "Disable Automatic Synchronization of the Confidential Manager Client Authentication and Encryption Settings Between the Server and Probes" on page 56.

1. On the UCMDB server, launch the Web browser and enter the following address: **http://localhost:8080/jmx-console.**
2. Click **UCMDB:service=Security Services** to open the JMX MBEAN View page.
3. Click the **CMGetConfiguration** method.

4. Click **Invoke**.

   The XML of the current Confidential Manager configuration is displayed.

5. Copy the contents of the displayed XML.

6. Navigate back to the **Security Services** JMX MBean View page.

7. Click the **CMSetConfiguration** method.

8. Paste the copied XML into the **Value** field.

9. Update the relevant transport-related settings and click **Invoke**.

   Example:

```
<transport>
      <encryptTransportMode>true</encryptTransportMode>
      <CMEncryptionDecryption>
            <encryptDecryptInitString>radiohead</encryptDecryptInitStrin
            g>
            <cryptoSource>lw</cryptoSource>
            <lwJCEPBECompatibilityMode>true</lwJCEPBECompatibilityMode>
            <cipherType>symmetricBlockCipher</cipherType>
            <engineName>AES</engineName>
            <algorithmModeName>CBC</algorithmModeName>
            <algorithmPaddingName>PKCS7Padding</algorithmPaddingName>
            <keySize>256</keySize>
            <pbeCount>20</pbeCount>
            <pbeDigestAlgorithm>SHA1</pbeDigestAlgorithm>
            <encodingMode>Base64Url</encodingMode>
            <useMacWithCrypto>false</useMacWithCrypto>
            <macType>hmac</macType>
            <macKeySize>256</macKeySize>
            <macHashName>SHA256</macHashName>
      </CMEncryptionDecryption>
</transport>
```

For details about the values that can be updated, see "Confidential Manager Encryption Settings" on page 67.

Configure Confidential Manager Client Authentication and Encryption Settings Manually on the Probe

This task includes the following steps:

- "Disable Automatic Synchronization of the Confidential Manager Client Authentication and Encryption Settings Between the Server and Probes" below
- "Configure Confidential Manager Client Authentication and Encryption Settings on the Probe" below
- "Configure Confidential Manager Communication Encryption on the Probe" on the next page

Disable Automatic Synchronization of the Confidential Manager Client Authentication and Encryption Settings Between the Server and Probes

By default, the UCMDB Server is configured to automatically send the Confidential Manager/LW-SSO settings to all Probes. This information is sent as an encrypted string to the Probes, which decrypt the information upon retrieval. You can configure the UCMDB Server to not send the Confidential Manager/LW-SSO configuration files automatically to all Probes. In this case, it is your responsibility to manually update all Probes with the new Confidential Manager/LW-SSO settings.

To disable automatic synchronization of Confidential Manager/LW-SSO settings:

1. In UCMDB, click **Administration > Infrastructure Settings Manager > General Settings**.
2. Select **Enable automatic synchronization of CM/LW-SSO configuration and init string with probe.**
3. Click the **Value** field and change **True** to **False**.
4. Click the **Save** button.
5. Restart the UCMDB server.

Configure Confidential Manager Client Authentication and Encryption Settings on the Probe

This procedure is relevant if the UCMDB Server has been configured to not send LW-SSO/Confidential Manager configuration and settings automatically to Probes. For details, see

"Disable Automatic Synchronization of the Confidential Manager Client Authentication and Encryption Settings Between the Server and Probes" on the previous page.

1. On the Probe machine, launch the Web browser and enter the following address: **http://localhost:1977**.

   Note: If the Probe Manager and the Probe Gateway are running as separate processes, the address should be entered on the machine that is running the Probe Manager as follows: **http://localhost:1978**.

2. Click **type=CMClient** to open the JMX MBEAN View page.

3. Locate the **setLWSSOInitString** method and provide the same init string that was provided for UCMDB's LW-SSO configuration.

4. Click the **setLWSSOInitString** button.

Configure Confidential Manager Communication Encryption on the Probe

This procedure is relevant if the UCMDB Server has been configured to not send LW-SSO/Confidential Manager configuration and settings automatically to Probes. For details, see "Disable Automatic Synchronization of the Confidential Manager Client Authentication and Encryption Settings Between the Server and Probes" on the previous page.

1. On the Probe machine, launch the Web browser and enter the following address: **http://localhost:1977**.

   Note: If the Probe Manager and the Probe Gateway are running as separate processes, the address should be entered on the machine that is running the Probe Manager as follows: **http://localhost:1978**.

2. Click **type=CMClient** to open the JMX MBEAN View page.

3. Update the following transport-related settings:

   Note: You must update the same settings that you updated on the UCMDB server. To do this, some of the methods that you update on the Probe may require more than one parameter. To see the current probe configuration, click **displayTransportConfiguration** in the JMX MBEAN View page. For details, see "Configure Confidential Manager Communication Encryption " on page 54. For details about the values that can be updated, see "Confidential Manager Encryption Settings" on page 67.

a. **setTransportInitString** changes the **encryptDecryptInitString** setting.

b. **setTransportEncryptionAlgorithm** changes Confidential Manager settings on the Probe according to the following map:

- **Engine name** refers to the <engineName> entry
- **Key size** refers to the <keySize> entry
- **Algorithm padding name** refers to the <algorithmPaddingName> entry
- **PBE count** refers to the <pbeCount> entry
- **PBE digest algorithm** refers to the <pbeDigestAlgorithm> entry

c. **setTransportEncryptionLibrary** changes Confidential Manager settings on the Probe according to the following map:

- **Encryption Library name** refers to the <cryptoSource> entry
- **Support previous lightweight cryptography versions** refers to the <lwJCEPBECompatibilityMode> entry

d. **setTransportMacDetails** change Confidential Manager settings on the Probe according to the following map:

- **Use MAC with cryptography** refers to the <useMacWithCrypto> entry
- **MAC key size** refers to the <macKeySize> entry

4. Click the **reloadTransportConfiguration** button to make the changes effective on the Probe.

For details about the different settings and their possible values, see "Confidential Manager Encryption Settings" on page 67.

Configure the Confidential Manager Client Cache

This task includes the following steps:

- "Configure the Confidential Manager Client's Cache Mode on the Probe" on the next page
- "Configure the Confidential Manager Client's Cache Encryption Settings on the Probe" on the next page

Configure the Confidential Manager Client's Cache Mode on the Probe

The Confidential Manager client stores credentials information in the cache and updates it when the information changes on the Server. The cache can be stored on the file system or in memory:

- **When stored on the file system**, even if the Probe is restarted and cannot connect to the Server, the credentials information is still available.
- **When stored in memory**, if the Probe is restarted, the cache is cleared and all information is retrieved again from the Server. If the Server is not available, the Probe does not include any credentials, so no discovery or integration can run.

To change this setting:

1. Open the **DataFlowProbe.properties** file in a text editor. This file is located in the **c:\hp\UCMDB\DataFlowProbe\conf** folder.
2. Locate the following attribute:
com.hp.ucmdb.discovery.common.security.storeCMData=true
   - To store the information on the file system, leave the default (**true**).
   - To store the information in memory, enter **false**.
3. Save the **DataFlowProbe.properties** file.
4. Restart the Probe.

Configure the Confidential Manager Client's Cache Encryption Settings on the Probe

This procedure describes how to change the encryption settings of the Confidential Manager client's file system cache file. Note that changing the encryption settings for the Confidential Manager client's file system cache causes the file system cache file to be recreated. This recreation process requires restarting the Probe and full synchronization with the UCMDB Server.

1. On the Probe machine, launch the Web browser and enter the following address:
http://localhost:1977.

   Note: If the Probe Manager and the Probe Gateway are running as separate processes, the address should be entered on the machine that is running the Probe Manager as follows: **http://localhost:1978**.

2. Click **type=CMClient** to open the JMX MBEAN View page.

3. Update the following cache-related settings:

   Note: Some of the methods that you update on the Probe may require more than one parameter. To see the current probe configuration, click **displayCacheConfiguration** in the JMX MBEAN View page.

   a. **setCacheInitString** changes the file system cache <encryptDecryptInitString> setting.

   b. **setCacheEncryptionAlgorithm** changes the file system cache settings according to the following map:

      - **Engine name** refers to the <engineName> entry
      - **Key size** refers to the <keySize> entry
      - **Algorithm padding name** refers to the <algorithmPaddingName> entry
      - **PBE count** refers to the <pbeCount> entry
      - **PBE digest algorithm** refers to the <pbeDigestAlgorithm> entry

   c. **setCacheEncryptionLibrary** changes the cache file system settings according to the following map:

      - **Encryption Library name** refers to the <cryptoSource> entry
      - **Support previous lightweight cryptography versions** refers to the <lwJCEPBECompatibilityMode> entry

   d. **setCacheMacDetails** changes the cache file system settings according to the following map:

      - **Use MAC with cryptography** refers to the <useMacWithCrypto> entry
      - **MAC key size** refers to the <macKeySize> entry

4. Click the **reloadCacheConfiguration** button to make the changes effective on the Probe. This causes the Probe to restart.

   Note: Make sure that no job is running on the Probe during this action.

For details about the different settings and their possible values, see "Confidential Manager Encryption Settings" on page 67.

Export and Import Credential and Range Information in Encrypted Format

You can export and import credentials and network range information in encrypted format in order to copy the credentials information from one UCMDB Server to another. For example, you might perform this operation during recovery following a system crash or during upgrade.

- **When exporting credentials information,** you must enter a password (of your choosing). The information is encrypted with this password.
- **When importing credentials information,** you must use the same password that was defined when the DSD file was exported.

Note: The exported credentials document also contains ranges information that is defined on the system from which the document was exported. During the import of the credentials document, ranges information is imported as well.

To export credentials information from the UCMDB Server:

1. On the UCMDB Server, launch the Web browser and enter the following address: **http://localhost:8080/jmx-console**. You may have to log in with a user name and password.
2. Click **UCMDB:service=DiscoveryManager** to open the JMX MBEAN View page.
3. Locate the **exportCredentialsAndRangesInformation** operation. Do the following:
   - Enter your customer ID (the default is 1).
   - Enter a name for the exported file.
   - Enter your password.
   - Set **isEncrypted=True** if you want the exported file to be encrypted with the provided password, or **isEncrypted=False** if you want the exported file to not be encrypted (in which case passwords and other sensitive information are not exported).
4. Click **Invoke** to export.

   When the export process completes successfully, the file is saved to the following location: **c:\hp\UCMDB\UCMDBServer\conf\discovery\<customer_dir>**.

To import credentials information from the UCMDB Server:

1. On the UCMDB Server, launch the Web browser and enter the following address: **http://localhost:8080/jmx-console**.

   You may have to log in with a user name and password.

2. Click **UCMDB:service=DiscoveryManager** to open the JMX MBEAN View page.
3. Locate the **importCredentialsAndRangesInformation** operation.
4. Enter your customer ID (the default is 1).
5. Enter the name of the file to import. This file must be located in **c:\hp\UCMDB\UCMDBServer\conf\discovery\<customer_dir>**.
6. Enter the password. This must be the same password that was used when the file was exported.
7. Click **Invoke** to import the credentials.

Change Confidential Manager Client Log File Message Level

The Probe provides two log files that contain information regarding Confidential Manager-related communication between the Confidential Manager server and the Confidential Manager client. The files are:

- "Confidential Manager Client Log File" below
- "LW-SSO Log File" on the next page

Confidential Manager Client Log File

The **security.cm.log** file is located in the **c:\hp\UCMDB\DataFlowProbe\runtime\log** folder.

The log contains information messages exchanged between the Confidential Manager server and the Confidential Manager client. By default, the log level of these messages is set to INFO.

To change the log level of the messages to DEBUG level:

1. On the Data Flow Probe Manager server, navigate to **c:\hp\UCMDB\DataFlowProbe\conf\log**.
2. Open the **security.properties** file in a text editor.
3. Change the line:

   ```
   loglevel.cm=INFO
   ```

   to:

   ```
   loglevel.cm=DEBUG
   ```

4. Save the file.

LW-SSO Log File

The **security.lwsso.log** file is located in the **c:\hp\UCMDB\DataFlowProbe\runtime\log** folder.

The log contains information messages related to LW-SSO. By default, the log level of these messages is set to INFO.

To change the log level of the messages to DEBUG level:

1. On the Data Flow Probe Manager server, navigate to **c:\hp\UCMDB\DataFlowProbe\conf\log**.
2. Open the security.properties file in a text editor.
3. Change the line:

   ```
   loglevel.lwsso=INFO
   ```

   to:

   ```
   loglevel.lwsso=DEBUG
   ```

4. Save the file.

Generate or Update the Encryption Key

You can generate or update an encryption key to be used for encryption or decryption of Confidential Manager communication and authentication configurations exchanged between the UCMDB Server and the Data Flow Probe. In each case (generate or update), the UCMDB Server creates a new encryption key based on parameters that you supply (for example, key length, extra PBE cycles, JCE provider) and distributes it to the Probes.

The result of running the **generateEncryptionKey** method is a new generated encryption key. This key is stored only in secured storage and its name and details are not known. If you reinstall an existing Data Flow Probe, or connect a new Probe to the UCMDB Server, this new generated key is not recognized by the new Probe. In these cases, it is preferable to use the **changeEncryptionKey** method to change encryption keys. This way, when you reinstall a Probe or install a new Probe, you can import the existing key (whose name and location you know) by running the **importEncryptionKey** method on the Probe JMX console.

Note:

- The difference between the methods used to create a key (**generateEncryptionKey**) and update a key (**changeEncryptionKey**) is that **generateEncryptionKey** creates a new, random encryption key, while **changeEncryptionKey** imports an encryption key whose name you provide.
- Only one encryption key can exist on a system, no matter how many Probes are installed.

This task includes the following steps:

Generate a New Encryption Key

You can generate a new key to be used by the UCMDB Server and Data Flow Probe for encryption or decryption. The UCMDB Server replaces the old key with the new generated key, and distributes this key among the Probes.

To generate a new encryption key through the JMX console:

1. On the UCMDB server, launch the Web browser and enter the following address: **http://localhost:8080/jmx-console**.

   You may have to log in with a user name and password.

2. Click **UCMDB:service=DiscoveryManager** to open the JMX MBEAN View page.

3. Locate the generateEncryptionKey operation.

   a. In the **customerId** parameter box, enter 1 (the default).

   b. For **keySize**, specify the length of the encryption key. Valid values are 128, 192, or 256.

   c. For **usePBE**, specify **True** or **False**:

      - **True:** use additional PBE hash cycles.
      - **False:** do not use additional PBE hash cycles.

   d. For **jceVendor**, you can choose to use a non-default JCE provider. If the box is empty, the default provider is used.

   e. For **autoUpdateProbe**, specify **True** or **False**:

      - **True:** the server distributes the new key to the Probes automatically.
      - **False:** the new key should be placed on the Probes manually.

   f. For **exportEncryptionKey**, specify **True** or **False**.

- **True:** In addition to creating the new password and storing it in secured storage, the Server exports the new password to the file system (**c:\hp\UCMDB\UCMDBServer\conf\discovery\key.bin**). This option enables you to update Probes manually with the new password.
- **False:** The new password is not exported to the file system. To update Probes manually, set **autoUpdateProbe** to False and **exportEncryptionKey** to True.

> **Note:** Make sure that the Probe is up and connected to the server. If the Probe goes down, the key cannot reach the Probe. If you change the key before the Probe goes down, once the Probe is up again, the key is sent again to the Probe. However, if you have changed the key more than once before the Probe goes down, you must change the key manually through the JMX console. (Select **False** for **exportEncryptionKey**).

4. Click **Invoke** to generate the encryption key.

Update an Encryption Key on a UCMDB Server

You use the **changeEncryptionKey** method to import your own encryption key to the UCMDB server and distribute it among all Probes.

To update an encryption key through the JMX Console:

1. On the UCMDB Server, launch the Web browser and enter the following address: **http://localhost:8080/jmx-console.**

   You may have to log in with a user name and password.

2. Click **UCMDB:service=DiscoveryManager** to open the JMX MBEAN View page.

3. Locate the **changeEncryptionKey** operation.

   a. In the **customerId** parameter box, enter **1** (the default).

   b. For **newKeyFileName**, enter the name of the new key.

   c. For **keySizeInBits**, specify the length of the encryption key. Valid values are 128, 192, or 256.

   d. For **usePBE**, specify **True** or **False**:

   - **True:** use additional PBE hash cycles.
   - **False:** do not use additional PBE hash cycles.

e. For **jceVendor**, you can choose to use a non-default JCE provider. If the box is empty, the default provider is used.

f. For **autoUpdateProbe**, specify **True** or **False**:

- **True:** the server distributes the new key to the Probes automatically.
- **False:** the new key should be distributed manually using the Probe JMX console.

> **Note:** Make sure that the Probe is up and connected to the server. If the Probe goes down, the key cannot reach the Probe. If you change the key before the Probe goes down, once the Probe is up again, the key is sent again to the Probe. However, if you have changed the key more than once before the Probe goes down, you must change the key manually through the JMX console. (Select **False** for **autoUpdateProbe**).

4. Click **Invoke** to generate and update the encryption key.

Update an Encryption Key on a Probe

If you choose not to distribute an encryption key from the UCMDB Server to all Probes automatically (because of security concerns), you should download the new encryption key to all Probes and run the **importEncryptionKey** method on the Probe:

1. Place the encryption key file in **C:\hp\UCMDB\DataFlowProbe\conf\security**.
2. On the Probe machine, launch the Web browser and enter the following address: **http://localhost:1977**.

   You may have to log in with a user name and password.

   > **Note:** If the Probe Manager and the Probe Gateway are running as separate processes, the address should be entered on the machine that is running the Probe Manager as follows: **http://localhost:1978**.

3. On the Probe domain, click **type=SecurityManagerService**.
4. Locate the **importEncryptionKey** method.
5. Enter the name of the encryption key file that resides in **C:\hp\UCMDB\DataFlowProbe\conf\security**. This file contains the key to be imported.
6. Click the **importEncryptionKey** button.
7. Perform a restart of the probe.

Manually Change the Encryption Key when the Probe Manager and Probe Gateway are Installed on Separate Machines

1. On the Probe Manager machine, start the Probe Manager service (**Start > Programs > HP UCMDB > Probe Manager**).
2. Import the key from the server, using the Probe Manager JMX. For details, see "Generate a New Encryption Key" on page 64.
3. After the encryption key is imported successfully, restart the Probe Manager and Probe Gateway services.

Define Several JCE Providers

When you generate an encryption key through the JMX Console, you can define several JCE providers, using the **changeEncryptionKey** and **generateEncryptionKey** methods.

To change the default JCE provider:

1. Register the JCE provider jar files in **$JRE_HOME/lib/ext**.
2. Copy the jar files to the $JRE_HOME folder:
   - For the UCMDB Server: $JRE_HOME resides at: **c:\hp\UCMDB\UCMDBServer\bin\jre**
   - For the Data Flow Probe: $JRE_HOME resides at: **c:\hp\UCMDB\DataFlowProbe\bin\jre**
3. Add the provider class at the end of the provider list in the **$JRE_HOME\lib\security\java.security** file.
4. Update the **local_policy.jar** and **US_export_policy.jar** files to include unlimited JCE policies. You can download these jar files from the Sun Web site.
5. Restart the UCMDB Server and the Data Flow Probe.
6. Locate the JCE vendor field for the **changeEncryptionKey** or **generateEncryptionKey** method, and add the name of the JCE provider.

Confidential Manager Encryption Settings

This table lists the encryption settings that can be changed using various JMX methods. These encryption settings are relevant for encryption of communications between the Confidential Manager client and the Confidential Manager server, as well as for encryption of the Confidential Manager client's cache.

Confidential Manager Setting Name	Probe Confidential Manager Setting Name	Setting Description	Possible Values	Default Value
cryptoSource	Encryption Library name	This setting defines which encryption library to use.	lw, jce, windowsDPAPI, lwJCECompatible	lw
lwJCEPBE Compatibility Mode	Support previous lightweight cryptography versions	This setting defines whether to support previous lightweight cryptography or not.	true, false	true
engineName	Engine name	Encryption mechanism name	AES, DES, 3DES, Blowfish	AES
keySize	Key size	encryption key length in bits	For AES - 128, 192 or 256; For DES - 64; For 3DES - 192; For Blowfish - any number between 32 and 448	256
algorithm Padding Name	Algorithm padding name	Padding standards	PKCS7Padding, PKCS5Padding	PKCS7Padding
pbeCount	PBE count	The number of times to run the hash to create the key from password (init string)	Any positive number	20
pbeDigest Algorithm	PBE digest algorithm	Hashing type	SHA1, SHA256, MD5	SHA1
useMacWith Crypto	Use MAC with cryptography	Indication if to use MAC with the cryptography	true, false	false
macKeySize	MAC key size	Depends on MAC algorithm	256	256

Troubleshooting and Limitations

If you change the default domain name on the UCMDB server, you must first verify that the Data Flow Probe is not running. After the default domain name is applied, you must execute the **DataFlowProbe\tools\clearProbeData.bat** script on the Data Flow Probe side.

Note: Execution of the clearProbeData.bat script will cause a discovery cycle on the Probe side once the Probe is up.

Chapter 5: Data Flow Probe Hardening

This chapter includes:

Modify the PostgreSQL Database Encrypted Password

This section explains how to modify the encrypted password for the PostgreSQL database user.

1. Create the Encrypted Form of a Password (AES, 192-bit key)

   a. Access the Data Flow Probe JMX console. Launch a Web browser and enter the following address: **http://<Data Flow Probe machine name or IP address>:1977**. If you are running the Data Flow Probe locally, enter **http://localhost:1977**.

      You may have to log in with a user name and password.

      Note: If you have not created a user, use the default user name sysadmin and the password sysadmin to log in.

b. Locate the **Type=MainProbe** service and click the link to open the Operations page.

c. Locate the **getEncryptedDBPassword** operation.

d. In the **DB Password** field, enter the password to be encrypted.

e. Invoke the operation by clicking the **getEncryptedDBPassword** button.

The result of the invocation is an encrypted password string, for example:

```
66,85,54,78,69,117,56,65,99,90,86,117,97,75,50,112,65,53,67,114,112,65,61
,61
```

2. Stop the Data Flow Probe

Start > All Programs > HP UCMDB > Stop Data Flow Probe

3. Run the set_dbuser_password.cmd Script

This script is located in the following folder:
C:\hp\UCMDB\DataFlowProbe\tools\dbscripts\set_dbuser_password.cmd

Run the **set_dbuser_password.cmd** script with the new password as the first argument,and the PostgreSQL Root Account password as the second argument.

For example:

set_dbuser_password <my_password><root_password>.

The password must be entered in its unencrypted form (as plain text).

4. Update the Password in the Data Flow Probe Configuration Files

a. The password must reside encrypted in the configuration files. To retrieve the password's encrypted form, use the **getEncryptedDBPassword** JMX method, as explained in step 1.

b. Add the encrypted password to the following properties in the **C:\hp\UCMDB\DataFlowProbe\conf\DataFlowProbe.properties** file.

- **appilog.agent.probe.jdbc.pwd**

For example:

```
appilog.agent.probe.jdbc.user = mamprobe
appilog.agent.probe.jdbc.pwd =
66,85,54,78,69,117,56,65,99,90,86,117,97,75,50,112,65,53,67,114,112,65,
61,61
```

- **appilog.agent.local.jdbc.pwd**
- **appilog.agent.normalization.jdbc.pwd**

5. Start the Data Flow Probe

   Start > All Programs > HP UCMDB > Start Data Flow Probe

The clearProbeData Script: Usage

To recreate the database user without altering its current password, run the **clearProbeData.bat** script for Windows or the **clearProbeData.sh** script for Linux.

After running the script:

- Review the following file for errors:
  C:\hp\UCMDB\DataFlowProbe\runtime\log\probe_setup.log for Windows,
  /opt/hp/UCMDB/DataFlowProbe/runtime/log/probe_setup.log for Linux.
- Delete the file, as it contains the database password.

Note: Do not run this script unless instructed to do so by HP Software Support.

Set the JMX Console Encrypted Password

This section explains how to encrypt the password for the JMX user. The encrypted password is stored in the DataFlowProbe.properties file. Users must log in to access the JMX console.

1. ### Create the Encrypted Form of a Password (AES, 192-bit key)

   a. Access the Data Flow Probe JMX console. Launch a Web browser and enter the following address: **http://<Data Flow Probe machine name or IP address>:1977**. If you are running the Data Flow Probe locally, enter **http://localhost:1977**.

   You may have to log in with a user name and password.

   Note: If you have not created a user, use the default user name sysadmin and the password sysadmin to log in.

   b. Locate the **Type=MainProbe** service and click the link to open the Operations page.

   c. Locate the **getEncryptedKeyPassword** operation.

   d. In the **Key Password** field, enter the password to be encrypted.

   e. Invoke the operation by clicking the **getEncryptedKeyPassword** button.

The result of the invocation is an encrypted password string, for example:

```
85,-9,-61,11,105,-93,-81,118
```

2. **Stop the Data Flow Probe**

   Start > All Programs > HP UCMDB > Stop Data Flow Probe

3. **Add the Encrypted Password**

   Add the encrypted password to the following property in the **C:\hp\UCMDB\DataFlowProbe\conf\DataFlowProbe.properties** file.

   appilog.agent.Probe.JMX.BasicAuth.Pwd

   For example:

   ```
   appilog.agent.Probe.JMX.BasicAuth.User=sysadmin
   appilog.agent.Probe.JMX.BasicAuth.Pwd=12,-35,-37,82,-2,20,57,-40,38,80,-111,-99,-64,-5,35,-122
   ```

   Note: To disable authentication, leaves these fields empty. If you do so, users can open the main page of the Probe's JMX console without entering authentication.

4. **Start the Data Flow Probe**

   Start > All Programs > HP UCMDB > Start Data Flow Probe

   Test the result in a Web browser.

Set the UpLoadScanFile Password

This section explains how to set the password for **UpLoadScanFile**, used for off-site scan saving. The encrypted password is stored in the **DataFlowProbe.properties** file. Users must log in to access the JMX console.

1. **Create the Encrypted Form of a Password (AES, 192-bit key)**

   a. Access the Data Flow Probe JMX console. Launch a Web browser and enter the following address: **http://<Data Flow Probe machine name or IP address>:1977**. If you are running the Data Flow Probe locally, enter **http://localhost:1977**.

   You may have to log in with a user name and password.

   Note: If you have not created a user, use the default user name sysadmin and the password sysadmin to log in.

b. Locate the **Type=MainProbe** service and click the link to open the Operations page.

c. Locate the **getEncryptedKeyPassword** operation.

d. In the **Key Password** field, enter the password to be encrypted.

e. Invoke the operation by clicking the **getEncryptedKeyPassword** button.

The result of the invocation is an encrypted password string, for example:

```
85,-9,-61,11,105,-93,-81,118
```

2. **Stop the Data Flow Probe**

Start > All Programs > HP UCMDB > Stop Data Flow Probe

3. **Add the Encrypted Password**

Add the encrypted password to the following property in the **C:\hp\UCMDB\DataFlowProbe\conf\DataFlowProbe.properties** file.

com.hp.ucmdb.discovery.Probe.JMX.UploadAuth.Pwd

For example:

```
com.hp.ucmdb.discovery.Probe.JMX.UploadAuth.User=UploadScanFile
com.hp.ucmdb.discovery.Probe.JMX.UploadAuth.Pwd=116,116,21,34,-59,77,-
108,14,127,4,-89,101,-33,-31,116,53
```

4. **Start the Data Flow Probe**

Start > All Programs > HP UCMDB > Start Data Flow Probe

Test the result in a Web browser.

Remote Access to the PostgreSQL Server

This section explains how to permit/restrict access to the PostgreSQL Data Flow Probe Account from remote machines.

Note:

- By default, access is restricted.
- You cannot access the PostgreSQL Root Account from remote machines.

To permit PostgreSQL access:

- Run the following script in a command prompt window:

  C:\hp\UCMDB\DataFlowProbe\tools\dbscripts\enable_remote_user_access.cmd

To restrict PostgreSQL access:

- Run the following script in a command prompt window:

  C:\hp\UCMDB\DataFlowProbe\tools\dbscripts\remove_remote_user_access.cmd

Enable SSL between UCMDB Server and Data Flow Probe

You can set up authentication for both the Data Flow Probe and the UCMDB Server with certificates. The certificate for each component is sent and authenticated before the connection is established.

Note: The following method of enabling SSL on the Data Flow Probe is the most secure of the methods and is therefore the recommended communication mode. This method replaces the procedure for basic authentication.

This section includes the following topics:

Overview

UCMDB supports the following modes of communication between the UCMDB Server and the Data Flow Probe:

- **Server Authentication.** This mode uses SSL, and the Probe authenticates the UCMDB Server certificate. For details, see "Enable SSL with Server (One-Way) Authentication" on the next page.
- **Mutual Authentication.** This mode uses SSL and enables both Server authentication by the Probe and client authentication by the Server. For details, see "Enable Mutual (Two-Way) Certificate Authentication" on page 79.

- **Standard HTTP.** No SSL communication. This is the default mode, and the Data Flow Probe component in UCMDB does not require any certificates. The Data Flow Probe communicates with the server through the standard HTTP protocol.

Note: Discovery cannot use certificate chains when working with SSL. Therefore, if you are using certificate chains, you should generate a self-signed certificate for the Data Flow Probe to be able to communicate with the UCMDB Server.

Keystores and Truststores

The UCMDB Server and the Data Flow Probe work with keystores and truststores:

- **Keystore.** A file holding key entries (a certificate and a matching private key).
- **Truststore.** A file holding certificates that are used to verify a remote host (for example, when using server authentication, the Data Flow Probe's truststore should include the UCMDB Server certificate).

Mutual Authentication Limitation

The Data Flow Probe keystore (as defined in **C:\HP\UCMDB\DataFlowProbe\conf\security\ssl.properties**) must contain only 1 (one) key entry.

Enable SSL with Server (One-Way) Authentication

This uses SSL, and the Probe authenticates the Server's certificate.

This task includes:

- "Prerequisites" below
- "UCMDB Server Configuration" on the next page
- "Data Flow Probe Configuration" on page 78
- "Restart the Machines" on page 79

Prerequisites

1. Verify that both UCMDB and the Data Flow Probe are running.

   Note: If the Probe is installed in separate mode, these instructions refer to the Probe Gateway.

2. If UCMDB or the Data Flow Probe are not installed in the default folders, note the correct location, and change the commands accordingly.

UCMDB Server Configuration

1. **Export the UCMDB Certificate**

   a. Open the command prompt and run the command:

   ```
   C:\HP\UCMDB\UCMDBServer\bin\jre\bin\keytool.exe -export -alias <keystore alias> -keystore <Keystore file path> -file C:\HP\UCMDB\UCMDBServer\conf\security\server.cert
   ```

   where:

   - **keystore alias** is the name given to the keystore.
   - **Keystore file path** is the full path of the location of the keystore file.

   For example, for the out-of-the-box server.keystore use the following command:

   ```
   C:\HP\UCMDB\UCMDBServer\bin\jre\bin\keytool.exe -export -alias hpcert -keystore C:\hp\ucmdb\ucmdbserver\conf\security\server.keystore -file C:\HP\UCMDB\UCMDBServer\conf\security\server.cert
   ```

   b. Enter the keystore password. For example, the out-of-the-box keystore password is **hppass**.

   c. Verify that the certificate was created in the following directory: **C:\HP\UCMDB\UCMDBServer\conf\security\server.cert**

2. **Harden the Data Flow Probe connector in UCMDB**

   a. Access the UCMDB JMX console: In your Web browser, enter the following URL: **http://<ucmdb machine name or IP address>:8080/jmx-console**. You may have to log in with a user name and password.

   b. Select the service: **Ports Management Services**.

   c. Invoke the **PortsDetails** method, and note the port number for HTTPS. (Default: 8443) Ensure that the value in the **Is Enabled** column is **True**.

   d. Return to **Ports Management Services**.

   e. To map the Data Flow Probe connector to server authentication mode, invoke the **mapComponentToConnectors** method with the following parameters:

   - **componentName**: mam-collectors
   - **isHTTPS**: true
   - **All other flags**: false

The following message is displayed:

Operation succeeded. Component mam-collectors is now mapped to: HTTPS ports.

f. Return to **Ports Management Services**.

g. To map the Confidential Manager connector to server authentication mode, invoke the **mapComponentToConnectors** method with the following parameters:

- **componentName**: cm
- **isHTTPS**: true
- **All other flags**: false

The following message is displayed:

Operation succeeded. Component cm is now mapped to: HTTPS ports.

3. **Copy the UCMDB certificate to each Probe machine**

Copy the certificate file, **C:\HP\UCMDB\UCMDBServer\conf\security\server.cert**, on the UCMDB Server machine to the following folder on each Data Flow Probe machine **C:\HP\UCMDB\DataFlowProbe\conf\security**

Data Flow Probe Configuration

Note: You must configure each Data Flow Probe machine.

1. **Import the server.cert file, created in** "Export the UCMDB Certificate" on the previous page**, to the Probe's Truststore.**

a. Open the command prompt and run the command:

```
C:\HP\UCMDB\DataFlowProbe\bin\jre\bin\keytool.exe -import -v -keystore C:
\HP\UCMDB\DataFlowProbe\conf\security\hpprobeTrustStore.jks -file C:\HP\U
CMDB\DataFlowProbe\conf\security\server.cert -alias hpcert
```

b. Enter the keystore password: logomania

c. When asked **Trust this certificate?**, press **y** and then **Enter**.

The following message is displayed:

Certificate was added to keystore.

2. **Open the DataFlowProbe.properties file located in: C:\HP\UCMDB\DataFlowProbe\conf**

   a. Update the **appilog.agent.probe.protocol** property to **HTTPS**.

   b. Update the **serverPortHttps** property to the relevant port number. (Use the port number from step 2c of "UCMDB Server Configuration" on page 77.)

Restart the Machines

Restart both the UCMDB server and the Probe machines.

Enable Mutual (Two-Way) Certificate Authentication

This mode uses SSL and enables both Server authentication by the Probe and client authentication by the Server. Both the Server and the Probe send their certificates to the other entity for authentication.

This task includes:

Prerequisites

1. Verify that both UCMDB and the Data Flow Probe are running.

   Note: If the Probe is installed in separate mode, these instructions refer to the Probe Gateway.

2. If UCMDB or the Data Flow Probe are not installed in the default folders, note the correct location, and change the commands accordingly.

Initial UCMDB Server Configuration

1. **Export the UCMDB Certificate**

   a. Open the command prompt and run the command:

   ```
   C:\HP\UCMDB\UCMDBServer\bin\jre\bin\keytool.exe -export -alias <keystore alias> -keystore <Keystore file path> -file C:\HP\UCMDB\UCMDBServer\conf\security\server.cert
   ```

   where:

   - **keystore alias** is the name given to the keystore.
   - **Keystore file path** is the full path of the location of the keystore file.

   For example, for the out-of-the-box server.keystore use the following command:

   ```
   C:\HP\UCMDB\UCMDBServer\bin\jre\bin\keytool.exe -export -alias hpcert -keystore C:\hp\ucmdb\ucmdbserver\conf\security\server.keystore -file C:\HP\UCMDB\UCMDBServer\conf\security\server.cert
   ```

   b. Enter the keystore password. For example, the out-of-the-box keystore password is **hppass**.

   c. Verify that the certificate was created in the following directory: **C:\HP\UCMDB\UCMDBServer\conf\security\server.cert**

2. **Harden the Data Flow Probe connector in UCMDB**

   a. Access the UCMDB JMX console: In your Web browser, enter the following URL: **http://<ucmdb machine name or IP address>:8080/jmx-console**. You may have to log in with a user name and password.

   b. Select the service: **Ports Management Services**.

   c. Invoke the **PortsDetails** method, and note the port number for HTTPS with client authentication. (Default: 8444) Ensure that the value in the **Is Enabled** column is **True**.

   d. Return to **Ports Management Services**.

   e. To map the Data Flow Probe connector to mutual authentication mode, invoke the **mapComponentToConnectors** method with the following parameters:

   - **componentName**: mam-collectors
   - **isHTTPSWithClientAuth**: true
   - **All other flags**: false

The following message is displayed:

Operation succeeded. Component mam-collectors is now mapped to: HTTPS_CLIENT_AUTH ports.

f. Return to **Ports Management Services**.

g. To map the Confidential Manager connector to mutual authentication mode, invoke the **mapComponentToConnectors** method with the following parameters:

- **componentName**: cm
- **isHTTPSWithClientAuth**: true
- **All other flags**: false

The following message is displayed:

Operation succeeded. Component cm is now mapped to: HTTPS_CLIENT_AUTH ports.

3. **Copy the UCMDB certificate to each Probe machine**

Copy the certificate file, **C:\HP\UCMDB\UCMDBServer\conf\security\server.cert**, on the UCMDB Server machine to the following folder on each Data Flow Probe machine: **C:\HP\UCMDB\DataFlowProbe\conf\security**

Data Flow Probe Configuration

Note: You must configure each Data Flow Probe machine.

1. **Import the server.cert file, created in** "Export the UCMDB Certificate" on the previous page**, to the Probe's Truststore.**

a. Open the command prompt and run the command:

```
C:\HP\UCMDB\DataFlowProbe\bin\jre\bin\keytool.exe -import -v -keystore C:
\HP\UCMDB\DataFlowProbe\conf\security\hpprobeTrustStore.jks -file C:\HP\U
CMDB\DataFlowProbe\conf\security\server.cert -alias hpcert
```

b. Enter the keystore password: logomania

c. When asked **Trust this certificate?**, press **y** and then **Enter**.

The following message is displayed:

Certificate was added to keystore.

2. **Create a new client.keystore file**

   a. Open the command prompt and run the command:

   ```
   c:\HP\UCMDB\DataFlowProbe\bin\jre\keytool –genkey –alias <ProbeName> -key
   alg RSA –sigalg SHA256withRSA –keysize 2048 –keystore c:\HP\UCMDB\DataFlo
   wProbe\conf\security\client.keystore
   ```

   where **ProbeName** is the unique alias of the Data Flow Probe.

   > **Note:** To ensure that this alias is unique, use the Probe Name identifier that was given to the Probe when defining the Probe.

   b. Enter password for the keystore, of at least 6 characters, and make a note of it.

   c. Enter the password again for confirmation.

   d. Press **Enter** after answering each of the following questions:

   What is your first and last name? [Unknown]:

   What is the name of your organizational unit?[Unknown]:

   What is the name of your organization?[Unknown]:

   What is the name of your City or Locality?[Unknown]:

   What is the name of your State or Province?[Unknown]:

   What is the two-letter country code for this unit?[Unknown]:

   e. Type **yes** when asked **Is CN=Unknown, OU=Unknown, O=Unknown, L=Unknown, ST=Unknown, C=Unknown correct?**

   f. Press **Enter** after answering the following question:

   Enter key password for <probekey> (RETURN if same as keystore password):

   g. Verify the file was created in the following folder, and ensure its file size is greater than 0: **C:\hp\UCMDB\DataFlowProbe\conf\security\client.keystore**

3. **Export the new Client Certificate**

   a. Open the command prompt and run the command:

   ```
   C:\HP\UCMDB\DataFlowProbe\bin\jre\bin\keytool.exe -export -alias <ProbeNa
   me> -keystore C:\hp\UCMDB\DataFlowProbe\conf\security\client.keystore -fi
   le C:\hp\UCMDB\DataFlowProbe\conf\security\<ProbeName>.cert
   ```

b. When asked, enter the keystore password. (The password from Step 2b above.)

The following message is displayed:

Certificate stored in file <C:\hp\UCMDB\DataFlowProbe\conf\security\<ProbeName>.cert>

4. **Open the DataFlowProbe.properties file located in: C:\HP\UCMDB\DataFlowProbe\conf**

a. Update the **appilog.agent.probe.protocol** property to **HTTPS**.

b. Update the **serverPortHttps** property to the relevant port number. (Use the port number from step 2c of "Initial UCMDB Server Configuration" on page 80.)

5. **Open the ssl.properties file located in: C:\HP\UCMDB\DataFlowProbe\conf\security**

a. Update the **javax.net.ssl.keyStore** property to **client.keystore**.

b. Encrypt the password from Step 2b above:

i. Start the Data Flow Probe (or make sure it is already running).

ii. Access the Probe JMX. Browse to: **http://<probe_hostname>:1977**

For example, if running the Probe locally, browse to: **http://localhost:1977**.

iii. Press the **type=MainProbe** link.

iv. Scroll down to the operation **getEncryptedKeyPassword**.

v. Enter the password in the **Key Password** field.

vi. Press the **getEncryptedKeyPassword** button.

c. Copy and paste the encrypted password to update the **javax.net.ssl.keyStorePassword** property.

> **Note:** Numbers are separated by commas. For example: -20,50,34,-40,-50.)

6. **Copy the Probe certificate to the UCMDB machine**

Copy the file **C:\HP\UCMDB\DataFlowProbe\conf\security\client.cert** from the Data Flow Probe machine to the UCMDB machine at **C:\HP\UCMDB\UCMDBServer\conf\security\<ProbeName>.cert**.

Further UCMDB Server Configuration

1. **Add each Probe certificate to the Truststore of UCMDB**

   Note: You must complete the following steps for each Probe certificate.

   a. Open the command prompt and run the command:

   ```
   C:\HP\UCMDB\UCMDBServer\bin\jre\bin\keytool.exe -import -v -keystore C:\hp
   \UCMDB\UCMDBServer\conf\security\server.truststore -file C:\hp\UCMDB\UCMDB
   Server\conf\security\<ProbeName>.cert -alias <ProbeName>
   ```

   b. Enter the keystore password. For example, the out-of-the-box keystore password is **hppass**.

   c. When asked **Trust this certificate?**, press **y** and then **Enter**.

      The following message is displayed:

      Certificate was added to keystore

Restart the Machines

Restart both the UCMDB server and the Probe machines.

Control the Location of the domainScopeDocument File

The Probe's file system holds (by default) both the encryption key and the **domainScopeDocument** file. Each time the Probe is started, the Probe retrieves the **domainScopeDocument** file from the server and stores it on its file system. To prevent unauthorized users from obtaining these credentials, you can configure the Probe so that the **domainScopeDocument** file is held in the Probe's memory and is not stored on the Probe file system.

To control the location of the domainScopeDocument file:

1. Open **C:\hp\UCMDB\DataFlowProbe\conf\DataFlowProbe.properties** and change:

   ```
   appilog.collectors.storeDomainScopeDocument=true
   ```

   to:

   ```
   appilog.collectors.storeDomainScopeDocument=false
   ```

   The Probe Gateway and Probe Manager serverData folders no longer contain the **domainScopeDocument** file.

For details on using the **domainScopeDocument** file to harden DFM, see "Data Flow Credentials Management" on page 49.

2. Restart the Probe.

Create a Keystore for the Data Flow Probe

1. On the Probe machine, run the following command:

```
c:\HP\UCMDB\DataFlowProbe\bin\jre\keytool –genkey –alias <ProbeName> -keyalg
RSA –sigalg SHA256withRSA –keysize 2048 –keystore c:\HP\UCMDB\DataFlowProbe\
conf\security\client.keystore
```

2. Enter a password for the new keystore.
3. Enter your information when asked.
4. When asked **Is CN=… C=… Correct?** enter **yes**, and press **Enter**.
5. Press **Enter** again to accept the keystore password as the key password.
6. Verify that **client.keystore** is created in the following directory: **C:\HP\UCMDB\DataFlowProbe\conf\security**.

Encrypt the Probe Keystore and Truststore Passwords

The Probe keystore and truststore passwords are stored encrypted in **C:\HP\UCMDB\DataFlowProbe\conf\security\ssl.properties**. This procedure explains how to encrypt the password.

1. Start Data Flow Probe (or verify that it is already running).
2. Access the Data Flow Probe JMX console: Launch a Web browser and enter the following address: http://<Data Flow Probe machine name or IP address>:1977. If you are running the Data Flow Probe locally, enter http://localhost:1977.

   Note: You may have to log in with a user name and password. If you have not created a user, use the default user name sysadmin and the password sysadmin to log in.

3. Locate the **Type=MainProbe** service and click the link to open the Operations page.
4. Locate the **getEncryptedKeyPassword** operation.

5. Enter your keystore or truststore password in the **Key Password** field and invoke the operation by clicking **getEncryptedKeyPassword**.
6. The result of the invocation is an encrypted password string, for example:

   ```
   66,85,54,78,69,117,56,65,99,90,86,117,97,75,50,112,65,53,67,114,112,65,61,61
   ```

7. Copy and paste the encrypted password into the line relevant to either the keystore or the truststore in the following file: **C:\HP\UCMDB\DataFlowProbe\conf\security\ssl.properties.**

Server and Data Flow Probe Default Keystore and Truststore

This section includes the following topics:

- "UCMDB Server" below
- "Data Flow Probe" below

UCMDB Server

The files are located in the following directory: **C:\HP\UCMDB\UCMDBServer\conf\security.**

Entity	File Name/Term	Password/Term	Alias
Server keystore	server.keystore (sKeyStoreFile)	hppass (sKeyStorePass)	hpcert
Server truststore	server.truststore (sTrustStoreFile)	hppass (sTrustStorePass)	hpcert (default trusted entry)
Client keystore	client.keystore (cKeyStoreFile)	clientpass (cKeyStorePass)	clientcert

Data Flow Probe

The files are located in the following directory: **C:\HP\UCMDB\DataFlowProbe\conf\security.**

Entity	File Name/Term	Password/Term	Alias
Probe keystore	hpprobeKeyStore.jks (pKeyStoreFile)	logomania (pKeyStorePass)	hpprobe
Data Flow Probe uses the **cKeyStoreFile** keystore as the default keystore during the mutual authentication procedure. This is a client keystore that is part of the UCMDB installation.			

Entity	File Name/Term	Password/Term	Alias
Probe truststore	hpprobeTrustStore.jks (pTrustStoreFile)	logomania (pTrustStorePass)	hpprobe (default trusted entry)
The **cKeyStorePass** password is the default password of **cKeyStoreFile**.			

Chapter 6: Lightweight Single Sign-On (LW-SSO) Authentication – General Reference

This chapter includes:

LW-SSO Authentication Overview

LW-SSO is a method of access control that enables a user to log on once and gain access to the resources of multiple software systems without being prompted to log on again. The applications inside the configured group of software systems trust the authentication, and there is no need for further authentication when moving from one application to another.

The information in this section applies to LW-SSO version 2.2 and 2.3.

- **LW-SSO Token Expiration**

  The LW-SSO Token's expiration value determines the application's session validity. Therefore, its expiration value should be at least the same value as that of the application session expiration value.

- **Recommended Configuration of the LW-SSO Token Expiration**

  Each application using LW-SSO should configure token expiration. The recommended value is 60 minutes. For an application that does not require a high level of security, it is possible to configure a value of 300 minutes.

- **GMT Time**

  All applications participating in an LW-SSO integration must use the same GMT time with a maximum difference of 15 minutes.

- **Multi-domain Functionality**

  Multi-domain functionality requires that all applications participating in LW-SSO integration configure the trustedHosts settings (or the **protectedDomains** settings), if they are required to

integrate with applications in different DNS domains. In addition, they must also add the correct domain in the **lwsso** element of the configuration.

- **Get SecurityToken for URL Functionality**

  To receive information sent as a **SecurityToken for URL** from other applications, the host application should configure the correct domain in the **lwsso** element of the configuration.

LW-SSO System Requirements

Application	Version	Comments
Java	1.5 and later	
HTTP Servlets API	2.1 and later	
Internet Explorer	6.0 and later	Browser should enable HTTP session cookie and HTTP 302 Redirect functionality.
Firefox	2.0 and later	Browser should enable HTTP session cookie and HTTP 302 Redirect functionality.
JBoss Authentications	JBoss 4.0.3 JBoss 4.3.0	
Tomcat Authentications	Standalone Tomcat 5.0.28 Standalone Tomcat 5.5.20	
Acegi Authentications	Acegi 0.9.0 Acegi 1.0.4	
Web Services Engines	Axis 1 - 1.4 Axis 2 - 1.2 JAX-WS-RI 2.1.1	

LW-SSO Security Warnings

This section describes security warnings that are relevant to the LW-SSO configuration:

- **Confidential InitString parameter in LW-SSO.** LW-SSO uses Symmetric Encryption to validate and create a LW-SSO token. The **initString** parameter within the configuration is used for initialization of the secret key. An application creates a token, and each application using the same initString parameter validates the token.

  Caution:

  - It is not possible to use LW-SSO without setting the **initString** parameter.
  - The **initString** parameter is confidential information and should be treated as such in terms of publishing, transporting, and persistency.
  - The **initString** parameter should be shared only between applications integrating with each other using LW-SSO.
  - The **initString** parameter should have a minimum length of 12 characters.

- **Enable LW-SSO only if required.** LW-SSO should be disabled unless it is specifically required.

- **Level of authentication security**. The application that uses the weakest authentication framework and issues a LW-SSO token that is trusted by other integrated applications determines the level of authentication security for all the applications.

  It is recommended that only applications using strong and secure authentication frameworks issue an LW-SSO token.

- **Symmetric encryption implications.** LW-SSO uses symmetric cryptography for issuing and validating LW-SSO tokens. Therefore, any application using LW-SSO can issue a token to be trusted by all other applications sharing the same initString parameter. This potential risk is relevant when an application sharing an initString either resides on, or is accessible from, an untrustworthy location.

- **User mapping (Synchronization).** The LW-SSO framework does not ensure user mapping between the integrated applications. Therefore, the integrated application must monitor user mapping. We recommend that you share the same user registry (as LDAP/AD) among all integrated applications.

  Failure to map users may cause security breaches and negative application behavior. For example, the same user name may be assigned to different real users in the various applications.

  In addition, in cases where a user logs onto an application (AppA) and then accesses a second application (AppB) that uses container or application authentication, the failure to map the user will force the user to manually log on to AppB and enter a user name. If the user enters a different user name than was used to log on to AppA, the followingbehavior can arise: If the user

subsequently accesses a third application (AppC) from AppA or AppB, then they will access it using the user names that were used to log on to AppA or AppB respectively.

- **Identity Manager**. Used for authentication purposes, all unprotected resources in the Identity Manager must be configured with the **nonsecureURLs** setting in the LW-SSO configuration file.
- **LW-SSO Demo mode.**
  - The Demo mode should be used for demonstrative purposes only.
  - The Demo mode should be used in unsecured networks only.
  - The Demo mode must not be used in production. Any combination of the Demo mode with the production mode should not be used.

Troubleshooting and Limitations

This section describes known issues and limitations when working with LW-SSO authentication.

Known Issues

This section describes known issues for LW-SSO authentication.

- **Security context.** The LW-SSO security context supports only one attribute value per attribute name.

  Therefore, when the SAML2 token sends more than one value for the same attribute name, only one value is accepted by the LW-SSO framework.

  Similarly, if the IdM token is configured to send more than one value for the same attribute name, only one value is accepted by the LW-SSO framework.

- **Multi-domain logout functionality when using Internet Explorer 7.** Multi-domain logout functionality may fail under the following conditions:
  - The browser used is Internet Explorer 7 and the application is invoking more than three consecutive HTTP 302 redirect verbs in the logout procedure.

  In this case, Internet Explorer 7 may mishandle the HTTP 302 redirect response and display an **Internet Explorer cannot display the webpage** error page instead.

  As a workaround, it is recommended to reduce, if possible, the number of application redirect commands in the logout sequence.

Limitations

Note the following limitations when working with LW-SSO authentication:

- **Client access to the application.**

  If a domain is defined in the LW-SSO configuration:

  - The application clients must access the application with a Fully Qualified Domain Name (FQDN) in the login URL, for example, http://myserver.**companydomain**.com/WebApp.
  - LW-SSO cannot support URLs with an IP address, for example, http://192.168.12.13/WebApp.
  - LW-SSO cannot support URLs without a domain, for example, http://myserver/WebApp.

  If a domain is not defined in the LW-SSO configuration: The client can access the application without a FQDN in the login URL. In this case, a LW-SSO session cookie is created specifically for a single machine without any domain information. Therefore, the cookie is not delegated by the browser to another, and does not pass to other computers located in the same DNS domain. This means that LW-SSO does not work in the same domain.

- **LW-SSO framework integration.** Applications can leverage and use LW-SSO capabilities only if integrated within the LW-SSO framework in advance.

- **Multi-Domain Support.**

  - Multi-domain functionality is based on the HTTP referrer. Therefore, LW-SSO supports links from one application to another and does not support typing a URL into a browser window, except when both applications are in the same domain.
  - The first cross domain link using **HTTP POST** is not supported.

    Multi domain functionality does not support the first **HTTP POST** request to a second application (only the **HTTP GET** request is supported). For example, if your application has an HTTP link to a second application, an **HTTP GET** request is supported, but an **HTTP FORM** request is not supported. All requests after the first can be either **HTTP POST** or **HTTP GET.**
  - LW-SSO Token size:

    The size of information that LW-SSO can transfer from one application in one domain to another application in another domain is limited to 15 Groups/Roles/Attributes (note that each item may be an average of 15 characters long).
  - Linking from Protected (HTTPS) to non-protected (HTTP) in a multi-domain scenario:

    Multi domain functionality does not work when linking from a protected (HTTPS) to a non-protected (HTTP) page. This is a browser limitation where the referrer header is not sent when linking from a protected to a non-protected resource. For an example, see:
    `http://support.microsoft.com/support/kb/articles/Q178/0/66.ASP`

- Third-Party cookie behavior in Internet Explorer:

  Microsoft Internet Explorer 6 contains a module that supports the "Platform for Privacy Preferences (P3P) Project," meaning that cookies coming from a Third Party domain are blocked by default in the Internet security zone. Session cookies are also considered Third Party cookies by IE, and therefore are blocked, causing LW-SSO to stop working. For details, see: `http://support.microsoft.com/kb/323752/en-us`.

  To solve this issue, add the launched application (or a DNS domain subset as *.mydomain.com) to the Intranet/Trusted zone on your computer (in Microsoft Internet Explorer, select **Menu > Tools > Internet Options > Security > Local intranet > Sites > Advanced)**, which causes the cookies to be accepted.

  Caution: The LW-SSO session cookie is only one of the cookies used by the Third Party application that is blocked.

- **SAML2 token**

  - Logout functionality is not supported when the SAML2 token is used.

    Therefore, if the SAML2 token is used to access a second application, a user who logs out of the first application is not logged out of the second application.

  - **The SAML2 token's expiration is not reflected in the application's session management.**

    Therefore, if the SAML2 token is used to access a second application, each application's session management is handled independently.

- **JAAS Realm.** The JAAS Realm in Tomcat is not supported.

- **Using spaces in Tomcat directories.** Using spaces in Tomcat directories is not supported.

  It is not possible to use LW-SSO when a Tomcat installation path (folders) includes spaces (for example, Program Files) and the LW-SSO configuration file is located in the **common\classes** Tomcat folder.

- **Load balancer configuration.** A load balancer deployed with LW-SSO must be configured to use sticky sessions.

- **Demo mode.** In Demo mode, LW-SSO supports links from one application to another but does not support typing a URL into a browser window, due to an HTTP referrer header absence in this case.

Chapter 7: HP Universal CMDB Login Authentication

This chapter includes:

Setting Up an Authentication Method

To perform authentication, you can work:

- **Against the internal HP Universal CMDB service.**
- **Through the Lightweight Directory Access Protocol (LDAP).** You can use a dedicated, external LDAP server to store the authentication information instead of using the internal HP Universal CMDB service. The LDAP server must reside on the same subnet as all the HP Universal CMDB servers.

  For details on LDAP, see the section about LDAP Mapping in the *HP Universal CMDB Administration Guide*.

  The default authentication method uses the internal HP Universal CMDB service. If you use the default method, you do not have to make any changes to the system.

  These options apply to logins performed through Web services as well as through the user interface.

- **Through LW-SSO.** HP Universal CMDB is configured with LW-SSO. LW-SSO enables you to log in to HP Universal CMDB and automatically have access to other configured applications running on the same domain, without needing to log in to those applications.

  When LW-SSO Authentication Support is enabled (it is disabled by default), you must ensure that the other applications in the Single Sign-On environment have LW-SSO enabled and are working with the same initString parameter.

Enabling Login to HP Universal CMDB with LW-SSO

To enable LW-SSO for HP Universal CMDB, use the following procedure:

1. Access the JMX console by entering the following address into your Web browser: **http://<server_name>:8080/jmx-console**, where **<server_name>** is the name of the machine on which HP Universal CMDB is installed.
2. Under **UCMDB-UI**, click the **name=LW-SSO Configuration** to open the Operations page.
3. Set the init string using the **setInitString** method.
4. Set the domain name of the machine on which UCMDB is installed using the **setDomain** method.
5. Invoke the method **setEnabledForUI** with the parameter set to **True.**
6. **Optional**. If you want to work using multi-domain functionality, select the **addTrustedDomains** method, enter the domain values and click **Invoke**.
7. **Optional.** If you want to work using a reverse proxy, select the **updateReverseProxy** method, set the **Is reverse proxy enabled** parameter to **True**, enter a URL for the **Reverse proxy full server URL** parameter, and click Invoke. If you want to access UCMDB both directly and using a reverse proxy, set the following additional configuration: select the **setReverseProxyIPs** method, enter the IP address for the Reverse proxy ip/s parameter and click **Invoke.**
8. **Optional**. If you want to access UCMDB using an external authentication point, select the **setValidationPointHandlerEnable** method, set the **Is validation point handler enabled** parameter to **True**, enter the URL for the authentication point in the **Authentication point server** parameter, and click **Invoke.**
9. To view the LW-SSO configuration as it is saved in the settings mechanism, invoke the **retrieveConfigurationFromSettings** method.
10. To view the actual loaded LW-SSO configuration, invoke the **retrieveConfiguration** method.

Note: You cannot enable LW-SSO via the user interface.

Setting a Secure Connection with the SSL (Secure Sockets Layer) Protocol

Since the login process involves the passing of confidential information between HP Universal CMDB and the LDAP server, you can apply a certain level of security to the content. You do this by

enabling SSL communication on the LDAP server and configuring HP Universal CMDB to work using SSL.

HP Universal CMDB supports SSL that uses a certificate issued by a trusted Certification Authority (CA).

Most LDAP servers, including Active Directory, can expose a secure port for an SSL based connection. If you are using Active Directory with a private CA, you must add your CA to the trusted CAs in the JRE.

For details on configuring the HP Universal CMDB platform to support communication using SSL, see "Enabling Secure Sockets Layer (SSL) Communication" on page 17 .

To add a CA to trusted CAs to expose a secure port for an SSL based connection:

1. Export a certificate from your CA and import it into the JVM that is used by HP Universal CMDB, using the following steps:

   a. On the UCMDB Server machine, access the **UCMDBServer\bin\JRE\bin** folder.

   b. Run the following command:

   ```
   Keytool -import -file <your certificate file> -keystore C:\hp\UCMDB\UCMDB Server\bin\JRE\lib\security\cacerts
   ```

   For example:

   ```
   Keytool -import -file c:\ca2ss_ie.cer -keystore C:\hp\UCMDB\UCMDBServer\bin\JRE\lib\security\cacerts
   ```

2. Select **Administration > Infrastructure Settings > LDAP General** category.

   Note: It is also possible to configure these settings using the JMX console. For details, see "How to Enable and Define the LDAP Authentication Method Using the JMX Console" on page 99.

3. Locate **LDAP Server URL**, and enter a value, using the format:

   ```
   ldaps://<ldapHost>[:<port>]/[<baseDN>][??scope]
   ```

   For example:

   ```
   ldaps://my.ldap.server:389/ou=People,o=myOrg.com??sub
   ```

   Note the **s** in **ldaps**.

4. Click **Save** to save the new value or **Restore Default** to replace the entry with the default value (a blank URL).

Using the JMX Console to Test LDAP Connections

This section describes a method of testing the LDAP authentication configuration using the JMX console.

1. Launch your Web browser and enter the following address: **http://<server_name>:8080/jmx-console**, where **<server_name>** is the name of the machine on which HP Universal CMDB is installed.

   You may need to log in with a user name and password.

2. Under **UCMDB**, click **UCMDB:service=LDAP Services** to open the Operations page.

3. Locate **testLDAPConnection.**

4. In the **Value** box for the parameter **customer id**, enter the customer ID.

5. Click **Invoke.**

   The JMX MBEAN Operation Result page indicates whether the LDAP connection is successful. If the connection is successful, the page also shows the LDAP root groups.

How to Enable and Define the LDAP Authentication Method

You can enable and define the LDAP authentication method for an HP Universal CMDB system.

Note:

- You can also configure LDAP authentication settings using the JMX console. For details, see "How to Enable and Define the LDAP Authentication Method Using the JMX Console" on page 99.
- For an example of LDAP authentication settings, see "LDAP Authentication Settings - Example" on page 100.

To enable and define the LDAP authentication method in the UCMDB user interface:

1. Select **Administration > Infrastructure Settings > LDAP General** category.

2. Select **LDAP server URL** and enter the LDAP URL value, using the format:

```
ldap://<ldapHost>[:<port>]/[<baseDN>][??scope]
```

For example:

```
ldap://my.ldap.server:389/ou=People,o=myOrg.com??sub
```

3. Select the **LDAP Group Definition** category, locate **Groups base DN**, and enter the distinguished name of the general group.
4. Locate **Root groups base DN** and enter the distinguished name of the root group.
5. Select the **LDAP General** category, locate **Enable User Permissions Synchronization**, and verify that the value is set to **True.**
6. Select the **LDAP General Authentication** category, locate **Password of Search-Entitled User**, and fill in the password.
7. Select the **LDAP Options for Classes and Attributes** category, locate **Group class object**, and fill in the object class name (**group** for Microsoft Active Directory, and **groupOfUniqueNames** for Oracle Directory Server).
8. Locate **Groups member attribute**, and fill in the attribute name (**member** for Microsoft Active Directory, and **uniqueMember** for Oracle Directory Server).
9. Locate **Users object class**, and fill in the object class name (**user** for Microsoft Active Directory, and **inetOrgPerson** for Oracle Directory Server).
10. Locate **UUID attribute**, and fill in the unique identifying attribute for a user in your directory server. Make sure to select an attribute that is unique in your directory server. For example, when using SunOne/Oracle Directory Server, the UID attribute is not unique. In such a case, use either the email address attribute or the distinguished name. Using a non-unique attribute as the unique identifying attribute in the UCMDB may cause inconsistent behavior during login.
11. Save the new values. To replace an entry with the default value, click **Restore Default.**
12. If the infrastructure setting under **LDAP General**, **Is case-sensitivity enforced when authenticating with LDAP**, is set to **True**, then the authentication is case-sensitive.

    Caution: When the value of this infrastructure setting is changed, all external users must be manually deleted by the UCMDB administrator.

13. Map LDAP user groups to UCMDB user groups. For details, see "HP Universal CMDB Login Authentication" on page 94.
14. If you want to define a default set of permissions for users in an LDAP group that does not have

a group mapping, select the **LDAP General** category, locate **Automatically Assigned User Group**, and enter the group name.

15. **Important:** If you are configuring LDAP on a high availability environment, you must restart the cluster for the changes to take effect.

Note: Every LDAP user has a first name, last name, and email address saved in the local repository. If the value of any of these parameters that is stored on the LDAP server differs from the value in the local repository, the LDAP server values will overwrite the local values at each login.

How to Enable and Define the LDAP Authentication Method Using the JMX Console

This task describes how to configure LDAP authentication settings using the JMX console.

Note:

- In a high availability environment, make sure you log in to the JMX console of the Writer server.
- You can also configure LDAP authentication settings in UCMDB. For details, see "How to Enable and Define the LDAP Authentication Method" on page 97.
- For an example of LDAP authentication settings, see "LDAP Authentication Settings - Example" on the next page.

To configure LDAP authentication settings:

1. Launch your Web browser and enter the following address: **http://<server_name>:8080/jmx-console**, where **<server_name>** is the name of the machine on which HP Universal CMDB is installed.

   You may need to log in with a user name and password.

2. Under **UCMDB,** click **UCMDB:service=LDAP Services** to open the Operations page.

3. To view the current LDAP authentication settings, locate the **getLDAPSettings** method. Click **Invoke**. A table displays all the LDAP settings and their values.

4. To change the values of LDAP authentication settings, locate the **configureLDAP** method. Enter the values for the relevant settings and click **Invoke**. The JMX MBEAN Operation Result page indicates whether the LDAP authentication settings were updated successfully.

Note: If you do not enter a value for a setting, the setting retains its current value.

5. After configuring the LDAP settings, you can verify the LDAP user credentials:
   a. Locate the **verifyLDAPCredentials** method.
   b. Enter the customer ID, username, and password.
   c. Click **Invoke**.

      The JMX MBEAN Operation Result page indicates whether the user passes LDAP authentication.
6. **Important:** If you are configuring LDAP on a high availability environment, you must restart the cluster for the changes to take effect.

Note: Every LDAP user has a first name, last name, and email address saved in the local repository. If the value of any of these parameters that is stored on the LDAP server differs from the value in the local repository, the LDAP server values will overwrite the local values at each login.

LDAP Authentication Settings - Example

The following table contains an example of setting values for LDAP authentication:

Setting	Value
Users object class	user
Is case-sensitivity enforced in LDAP authentication	false
Groups member attribute	member
Distinguished Name (DN) Resolution	true
Root Group Filter	(objectCategory=group)
LDAP connection string	ldap://myldap.example.com:389/OU=Users,OU=Dept,OU=US,DC=example,DC=com??sub
LDAP Search User	CN=John Doe,OU=Users,OU=Dept,OU=US,DC=example,DC=com
Group class object	group

Setting	Value
Use bottom up algorithm for find parent groups	true
UUID attribute	sAMAccountName
Groups name attribute	cn
Group Base Filter	(objectclass=group)
Users filter	(&(sAMAccountName=*)(objectclass=user))
Search Retries Count	3
Groups display name attribute	cn
Root groups scope	sub
User display name attribute	cn
Scope for groups search	sub
Enable LDAP authentication	false
Enable LDAP synchronization	true
Root Group	OU=Users,OU=Security Groups,DC=example,DC=com
Group Base	OU=AMRND,OU=Security Groups,DC=example,DC=com
Default Group	AdminsGroup
Groups description attribute	description

Retrieving Current LW-SSO Configuration in Distributed Environment

When UCMDB is embedded in a distributed environment, for example, in a BSM deployment, perform the following procedure to retrieve the current LW-SSO configuration on the processing machine.

To retrieve the current LW-SSO configuration:

1. Launch a Web browser and enter the following address: `http://localhost.<domain_name>:8080/jmx-console.`

   You may be asked for a user name and password.

2. Locate **UCMDB:service=Security Services** and click the link to open the Operations page.

3. Locate the **retrieveLWSSOConfiguration** operation.
4. Click **Invoke** to retrieve the configuration.

Chapter 8: Confidential Manager

This chapter includes:

Confidential Manager Overview

The Confidential Manager framework solves the problem of managing and distributing sensitive data for HP Universal CMDB and other HP Software products.

Confidential Manager consists of two main components: the client and the server. These two components are responsible for transferring data in a secured manner.

- The Confidential Manager client is a library used by applications to access sensitive data.
- The Confidential Manager server receives requests from Confidential Manager clients, or from third party clients, and performs the required tasks. The Confidential Manager server is responsible for saving the data in a secure manner.

Confidential Manager encrypts credentials in transport, in the client cache, in persistency, and in memory. Confidential Manager uses symmetric cryptography for transporting credentials between the Confidential Manager client and the Confidential Manager server by using a shared secret. Confidential Manager uses various secrets for encryption of cache, persistency, and transport according to the configuration.

For detailed guidelines for managing credential encryption on the Data Flow Probe, see "Data Flow Credentials Management" on page 49.

Security Considerations

- You can use the following key sizes for the security algorithm: 128-, 192-, and 256-bits. The algorithm runs faster with the smaller key but it is less secure. The 128-bit size is secure enough in most cases.
- To make the system more secure, use MAC: set **useMacWithCrypto** to **true.** For details, see "Encryption Properties" on page 105.
- To leverage strong customer security providers, you can use the JCE mode.

Configure the HP Universal CMDB Server

When working with HP Universal CMDB, you should configure the secret and crypto-properties of the encryption, using the following JMX methods:

1. On the HP Universal CMDB Server machine, launch the Web browser and enter the Server address, as follows: **http://<UCMDB Server Host Name or IP>:8080/jmx-console.**

   You may have to log in with a user name and password.

2. Under UCMDB, click **UCMDB:service=Security Services** to open the Operations page.

3. To retrieve the current configuration, locate the **CMGetConfiguration** operation.

   Click **Invoke** to display the Confidential Manager server configuration XML file.

4. To make changes to the configuration, copy the XML that you invoked in the previous step to a text editor. Make changes according to the table in "Encryption Properties" on the next page.

   Locate the **CMSetConfiguration** operation. Copy the updated configuration into the **Value** box and click **Invoke.** The new configuration is written to the UCMDB Server.

5. To add users to Confidential Manager for authorization and replication, locate the **CMAddUser** operation. This process is also useful in the replication process. In replication, the server slave should communicate with the server master, using a privileged user.

   - **username**. The user name.
   - **customer.** The default is ALL_CUSTOMERS.
   - **resource.** The resource name. The default is ROOT_FOLDER.
   - **permission.** Choose between ALL_PERMISSIONS, CREATE, READ, UPDATE, and DELETE. The default is ALL_PERMISSIONS.

   Click **Invoke**.

6. If necessary, restart HP Universal CMDB.

   In most cases there is no need to restart the Server. You may need to restart the Server when changing one of the following resources:

   - Storage type
   - Database table name or column names
   - The creator of the database connection

- The connection properties to the database (that is, URL, user, password, driver class name)
- Database type

Note:

- It is important that the UCMDB Server and its clients have the same transport crypto-properties. If these properties are changed on the UCMDB Server, you must change them on all clients. (This is not relevant for the Data Flow Probe because it runs on the same process as the UCMDB Server—that is, there is no need for the Transport crypto-configuration.)
- Confidential Manager Replication is not configured by default, and can be configured if needed.
- If Confidential Manager Replication is enabled, and the Transportation **initString** or any other crypto-property of the master changes, all slaves must adopt the changes.

Definitions

Storage crypto-properties. The configuration that defines how the server holds and encrypts the data (in database or file, which crypto-properties must encrypt or decrypt the data, and so on), how credentials are stored in a secure manner, how encryption is processed, and according to which configuration.

Transport crypto-properties. Transport configuration defines how the server and the clients encrypt the transportation between them, which configuration is used, how credentials are transferred in a secure manner, how encryption is processed, and according to which configuration. You must use the same crypto-properties for transport encryption and decryption, in both server and client.

Replications and replication crypto-properties. Data held securely by Confidential Manager is securely replicated between several servers. These properties define how the data is to be transferred between slave server and master server.

Note:

- The database table that holds the Confidential Manager server configuration is named: **CM_CONFIGURATION.**
- The Confidential Manager Server default configuration file is located in app-infra.jar and is named **defaultCMServerConfig.xml.**

Encryption Properties

The following table describes encryption properties. For details on using these parameters, see "Configure the HP Universal CMDB Server" on the previous page.

Parameter	Description	Recommended value
encryptTransportMode	Encrypt the transported data: true false	true
encryptDecrypt InitString	Password for encryption	Longer than 8 characters
cryptoSource	Encryption implementation library to use: • lw • jce • windowsDPAPI • lwJCECompatible	lw
lwJCEPBE CompatibilityMode	Support previous versions of lightweight cryptography: • true • false	true
cipherType	The type of cipher that Confidential Manager uses. Confidential Manager supports one value only: **symmetricBlockCipher**	symmetric BlockCipher
engineName	• AES • Blowfish • DES • 3DES • Null (no encryption)	AES
algorithmModeName	Mode of block encryption algorithm: • CBC	CBC
algorithmPaddingName	Padding standards: • PKCS7Padding • PKCS5Padding	PKCS7Padding
keySize	Depends on algorithm (what **engineName** supports)	256

Parameter	Description	Recommended value
pbeCount	The number of times to run the hash to create the key from **encryptDecryptInitString.** Any positive number.	1000
pbeDigestAlgorithm	Hashing type: • SHA1 • SHA256 • MD5	SHA256
encodingMode	ASCII representation of the encrypted object: • Base64 • Base64Url	Base64Url
useMacWithCrypto	Defines whether MAC is used with the cryptography: • true • false	false
macType	Type of message authentication code (MAC): • hmac	hmac
macKeySize SHA256	Depends on Mac algorithm	256
macHashName	The Hash Mac algorithm: • SHA256	SHA256

Chapter 9: High Availability Hardening

This chapter includes:

Cluster Authentication

To enable cluster authentication:

1. In the UCMDB, go to **Administration > Infrastructure Settings Manager**.
2. Find the setting **Enable joining High Availability cluster authentication** and set it to **true**.
3. Provide a single server authentication keystore (certificate + private and public keys) in JKS format. This keystore will be placed on all the servers and used for authenticating when connecting to a high availability cluster.

   Place the keystore in the following location: **<UCMDB installation folder>\conf\security** and name it **cluster.authentication.keystore.**

   > **Note:** The UCMDB comes with this keystore pre-configured out-of-the-box. This keystore is the same for all clean UCMDB installations, and thus not secure. If you wish to securely authenticate join requests, delete this file and create a new one.

4. Generate a cluster authentication keystore as follows:

   a. From C:\hp\UCMDB\UCMDBServer\bin\jre\bin, run the following command:

   ```
   keytool -genkey -alias hpcert –keystore <UCMDB installation folder>\conf\
   security\cluster.authentication.keystore -keyalg RSA
   ```

   The console dialog box opens and asks you for a new keystore password.

   b. The default password is **hppass**. If you want to use a different password, update the server by running the following JMX method: **UCMDB:service=High Availability Services: changeClusterAuthenticationKeystorePassword**

   c. In the console dialog box, answer the question **What is your first and last name?** by entering the name of the cluster.

   d. Enter the other parameters according to your organization's details.

e. Enter a key password. The key password must be the same as the keystore password.

A JKS keystore is created in **<UCMDB installation folder>\conf\security\cluster.authentication.keystore**

5. Replace the old **<UCMDB installation folder>\conf\security\cluster.authentication.keystore** on all the servers in the cluster with the new keystore.

6. Restart all the servers in the cluster.

Cluster Message Encryption

Use cluster message encryption to encrypt all the messages in the cluster.

To enable cluster message encryption:

1. In the UCMDB, go to **Administration>Infrastructure Settings Manager**.

2. Find the setting **Enable High Availability cluster communication encryption** and set it to **true**.

3. Provide a secret key for symmetric encryption on all the servers. The key should be placed in a keystore of type JCEKS in the following location **<UCMDB installation folder>\conf\security\cluster.encryption.keystore**.

> **Note:** The UCMDB comes with this keystore pre-configured out of the box. This keystore is the same for all clean UCMDB installations, and thus not secure. If you wish to securely encrypt cluster messages, please delete this file, and create a new one by following this procedure.

4. From **<UCMDB installation folder>\bin\jre\bin**, run the following command:

Keytool –genseckey –alias hpcert –keystore <UCMDB installation folder>\conf\security\cluster.encryption.keystore –storetype JCEKS

5. You will be asked for the new keystore password. The default password is “hppass”. If you want to use a different password, you need to update the server by running the following JMX method:

UCMDB:service=High Availability Services: changeClusterEncryptionKeystorePassword

6. Replace the old **<UCMDB installation folder>\conf\security\cluster.encryption.keystore** of all the servers in the cluster with this new keystore.

7. Restart the servers.

Troubleshooting

Upon every startup of the server, the server sends a test message to the cluster to verify if it successfully connected to the cluster. If there is a problem with the connection, the message fails and the server is stopped to avoid the whole cluster getting stuck.

Some examples of wrong cluster encryption configuration are:

- Disabled encryption on one node when another node enabled it.
- Wrong or missing cluster.encryption.keystore
- Wrong or missing key in the keystore

If the server gets stuck because of a configuration issue, the error message is:

```
2012-09-11 17:48:23,584 [Thread-14] FATAL - #### Server failed to connect proper
ly to the cluster and its service is stopped! Please fix the problem and start i
t again ####

2012-09-11 17:48:23,586 [Thread-14] FATAL -       Potential problems can be: wro
ng security configuration (wrong or missing cluster.encryption.keystore, wrong k
ey, disabled encryption in a cluster with enabled encryption)
```

Changing the Key in the key.bin

In a High Availability environment with several servers, change the **key** in the **key.bin** as follows:

1. Go to the writer machine in the JMX. You can choose any machine in the cluster and click on the **writer** link on the top of each page.
2. In the UCMDB section of the console, click **UCMDB:service=Discovery Manager**.
3. Change the key in one of the following ways:
   - Click **changeEncryptionKey** (this imports the existing encryption key)
   - Click **generateEncryptionKey** (this generates a random encryption key)
4. On the writer machine, go to the file system and find the **key.bin** at: **C:\hp\UCMDB\UCMDBServer\conf\discovery\key.bin**
5. Copy the **key.bin** from the location on the writer machine to each one of other machines in the cluster to the folder: **C:\hp\UCMDB\UCMDBServer\conf\discovery\customer_1** and rename the destination file (for example, **key_new.bin**).
6. For each of the other servers (readers) do the following:
   a. Switch the reader to be a writer (you can do this from the High Availability JMX) and wait until it changes.

b. Connect to the JMX of the current writer and click **UCMDB:service=Discovery Manager**.

c. Click and invoke **changeEncryptionKey**, use the same details you entered in step 3 (for **newKeyFileName**, use the new name you assigned at step 5).

d. Verify that you get the following message: **Key was created successfully**.

We appreciate your feedback!

If you have comments about this document, you can contact the documentation team by email. If an email client is configured on this system, click the link above and an email window opens with the following information in the subject line:

Feedback on Hardening Guide (Universal CMDB and Configuration Manager 10.10)

Just add your feedback to the email and click send.

If no email client is available, copy the information above to a new message in a web mail client, and send your feedback to SW-Doc@hp.com.