HP Client Automation

for Windows operating systems

Software Version: 7.80

SSL Implementation Guide

Manufacturing Part Number: None

Document Release Date: November 2009

Software Release Date: November 2009

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 Notices

Trademark Notices

Documentation Updates

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

- Software Version number, which indicates the software version.
  - The number before the period identifies the major release number.
  - The first number after the period identifies the minor release number.
  - The second number after the period represents the minor-minor release number.
- 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, visit:

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 and new editions if you subscribe to the appropriate product support service. Contact your HP sales representative for details.

Table 1 below lists the changes that were made to this document.

Table 1 Documentation changes

Chapter	Version	Changes
Global	7.8	Rebranded HP Configuration Management to HP Client Automation. Changed text CM to HPCA. Code snippets were only changed as appropriate. Paths on installation media still use \CM\, not changed.
Front Matter	7.8	Title page, rebranded HP Configuration Management to new name HP Client Automation, updated dates.
Chapter 1	5.11	Page 19, listed SSL-CM version-parity requirements and maintenance pack installation information.
Chapter 2	7.8	Page 38, Core and Satellite Overview section added for configuring SSL on a Core and Satellite installation.
Chapter 3	7.8	Page 42, Added note to verify logon to Reporting Server after turning on SSL in the Portal.
Chapter 3	5.11	Page 52, added information about Windows registry key

		changes if using a truststore with a password other than the default.
Chapter 3	7.8	Page 55, added step 9 for enabling inbound SSL.
Chapter 3	7.8	Page 64, HPCA Proxy Server Preload. added note indicating that proxy server preloading using SSL_TCPS does not work in Solaris and AIX UNIX.
Appendix A	5.11	Page 82, added information about Windows registry key changes if migrating from CM version 5.00 to version 5.1*x*.
Appendix A	5.11	Page 84, added information about using an encrypted channel for communications between the CM Enterprise Manager and the job process engine that executes the Notify commands.

Support

You can visit the HP Software support web site at:

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 software 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 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 an active support contract. To find more information about support access levels, go to the following URL:

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

To register for an HP Passport ID, go to the following URL:

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

Contents

1 Introduction to SSL, Certificates, and Private Keys

At the end of this chapter, you will have had the opportunity to:

- Review some of the important *components*, *concepts*, and *terms* that are integral to SSL encryption, including:
  - Private Keys
  - Public Keys
  - Certificates
  - Certificate Authorities
- Review the list of *HP Client Automation (HPCA) products* that can support SSL communication
- Review the various server-client *communications relationships* that are possible in a HPCA environment.

Overview

This chapter starts with the section An Introduction to SSL Encryption, which provides an introduction to some of the important components, concepts, and terms that are relevant to SSL encryption.

Following the introduction is SSL in a Client Automation Environment (starting on page 16), which provides a more specific discussion of SSL in the context of HPCA, including:

- SSL cipher-suite information
- SSL encryption requirements
- A list of the HPCA products that can be configured to use SSL

Using this Guide with Core and Satellite Servers

⚠ If your environment uses Core and Satellite servers, first read the *Core and Satellite Servers Getting Started Guide* as the installation, configuration, and troubleshooting information in that guide may override the information in this guide.

An Introduction to SSL Encryption

Secure Sockets Layer (SSL) is a cryptographic protocol that enables software applications to communicate securely across a network. SSL is designed to prevent eavesdropping, tampering, and message forgery. It is based on a principle called **mutual authentication**, which ensures that both parties to a conversation know precisely with whom they are communicating.

This section describes some of the components, concepts, and terms that are part of SSL encryption.

Public Key Cryptography

SSL implements mutual authentication by using **public key cryptography**. A key is simply a binary code, encoded and served in a text file, and associated with a particular user or software application.

SSL uses two keys—a public key and a private key—to encrypt and decrypt messages that are sent over the network. The public key is given freely to interested parties, but the associated private key remains private and is possessed only by the owner of the certificate. Data encrypted with the public key can be decrypted only with the private key.

In the context of CM, SSL public key cryptography includes:

- A private-public key pair on the server
- A certificate from a trusted **Certificate Authority** (CA). This is typically already present on each system.

Each of these requirements is described below.

The Key Pair

SSL encryption uses a **key pair** to encrypt a transmission. The key pair consists of a **private key** and a **public key**.

Private Key

In the context of HPCA, a key pair must be generated for each server. The server retains the **private key** and must keep it secure.

Public Key

The **public key** is passed to the client by the server. The client must trust that the public key that it receives is truly from the server that it (the client) thinks it is communicating with. Certificate Authorities sign public keys so that the keys can be trusted.

Certificates

A certificate is an electronic document that contains the server's public key, the server name, and a signature from a CA. A certificate authority is a trusted third party who attests that the public key in the certificate belongs to the party – in this case, the server – named in the certificate. It is the responsibility of the CA to verify the credentials of any party that applies for a certificate. This allows others to trust the information in the certificates issued by this CA.

Large companies and public sector organizations typically have their own CAs. There are independent CAs, such as Thawte and Verisign, who charge a fee for their services. There are also free CAs.

A client is configured with certificates from the CAs that it trusts. As long as the server's certificate has been *signed* by a CA that the client trusts (see Obtaining a Certificate from a Certificate Authority, starting on page 31), the server's certificate is considered "trusted," and SSL communications between the server and client can be initiated.

Certificates and Your Environment

This section discusses certificate generation in production and test environments.

Production Situations

It is best to generate a signing request that can be signed by a trusted Certificate Authority.

Test Situations

You can provide either:

- A **self-signed certificate**

  In this case, you must configure the client to trust each server's certificates.

- A **private CA-signed certificate**

  In this case, you can sign each server's certificate quickly—because you are the signing authority—and you only need to configure your clients to trust the private CA certificate that you generated.

Ciphers and Hash Functions

A **cipher** is a method of encrypting information. A **hash function** is a method of compressing information that transforms data into a short, fixed-length string that serves as a digital "fingerprint." Hash functions used in cryptography create a unique fingerprint for every input and work in one direction only; in other words, you cannot derive the original data from the fingerprint. Ciphers and hash functions are used by SSL.

SSL can use a number of ciphers and hash functions to encrypt messages. It uses two ciphers and one hash function for each connection. Together, the two ciphers and hash function are known as the **cipher suite**, and they are used to establish and protect that connection.

Keystores and Truststores

For two-way SSL communication to occur, the server and each client must have a truststore and a keystore.

- A keystore is a database that stores your private keys. It also contains certificates for trusted CAs.
- A truststore stores the public keys that you trust.

The keystore and the truststore are typically implemented as files. A keystore file is protected by a password. A truststore file needs no password because it contains no private information.

Setting up SSL

The following steps represent the generic process for setting up SSL on each machine that will be authenticated:

1 Locate (or create) a keystore.

2 Generate a public-private key pair.

3 If this new key pair is not yet trusted—in other words, if the public key that you generated is not yet in your keystore—follow these steps:

   a Generate a **Certificate Signing Request** (CSR) from the key pair.

   b Send the CSR signed to a trusted CA.

   c When the CA issues a signed certificate in response to your request, import the signed certificate that they send you into the keystore.

4 Configure the client and server to use the public-private key pair certificates.

How SSL Establishes a Secure Connection

A client and a server establish a secure connection by performing a handshake operation. The handshake accomplishes the following:

- The client and server agree on a cipher suite to use for the connection.
- The server sends its certificate—including public key, server name, and CA—to the client. The client can then contact the CA to verify the server's identity. If mutual authentication is required, the server will also request a certificate from the client.
- The client and server generate session keys that will be used for the duration of this connection.
  - The client encrypts a random number with the server's public key, and sends the result to the server.
  - The server decrypts the random number with its private key, which hides the session keys from third parties, since only the server and the client have access to this data.
- The client and server generate session keys that they will use for encryption and decryption.

SSL in a Client Automation Environment

This section presents information you will need to set up and use SSL in your Client Automation environment. It discusses the various HPCA products that can be configured for SSL communication, and it provides an overview of the protocols that are used to secure the various HPCA server-HPCA agent communications.

Supported HPCA Products

This section presents certificate requirements that are specific to HP Client Automation products.

SSL Requirements

To ensure that SSL encryption will work with the HPCA products, the following requirements must be met.

- HPCA servers must have a **public key**, a **private key**, and a **Certificate Authority** public key.
- HPCA agents must have a **Certificate Authority** public key.

SSL Cipher Suite and Encryption Information

HP Client Automation products use the following cipher from the SSL version 3 cipher suite: 168-bit triple DES cipher block chaining mode, 1024-bit RSA asymmetric key exchange, and secure hash algorithm version 1.0.

HPCA Server Components

The following is a list of the HPCA **server** products that can be configured for SSL communication. In the Core and Satellite installation the components are consolidated, but the functionality remains.:

- HP Client Automation Reporting Server (HPCA Reporting Server), see HPCA Reporting Server on page 40
- HP Client Automation Enterprise Manager (Enterprise Manager), see HPCA Enterprise Manager on page 46
- HP Client Automation Messaging Server (HPCA Messaging Server), see HPCA Messaging Server on page 53
- HP Client Automation Configuration Server (Configuration Server), see HPCA Configuration Server on page 55
- HP Client Automation Distributed Configuration Server (Distributed Configuration Server), see HPCA Distributed Configuration Server on page 58
- HP Client Automation Patch Manager (Patch Manager), see HPCA Patch Manager Server on page 60
- HP Client Automation Integration Server (Integration Server), see HPCA Integration Server on page 61
- HP Client Automation Policy Server (HPCA Policy Server), see HPCA Policy Server on page 65

- HP Client Automation Proxy Server (Proxy Server), see HPCA Proxy Server on page 63
- HP Client Automation Portal (Portal), see HPCA Portal on page 66
- HP Client Automation Application Usage Manager (HPCA Application Usage Manager), see HPCA Application Usage Manager on page 73

HPCA Agent Components

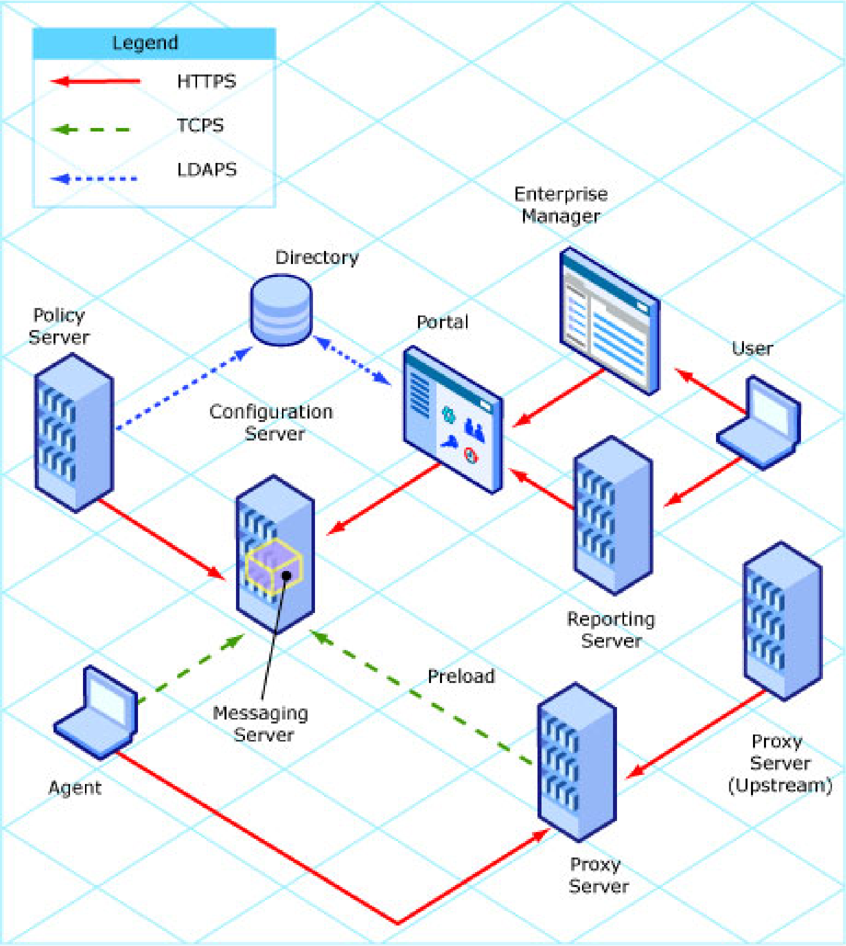
The following is a list of the HPCA **agent** products that can be configured for SSL communication. For information about these products, see the section, HPCA Agents, starting on page 75.

- HP Client Automation Application Manager agent (Application Manager)
- HP Client Automation Application Self-service Manager agent (Application Self-service Manager), see HPCA Application Self-service Manager Agent
- HP Client Automation Inventory Manager agent (Inventory Manager)
- HP Client Automation Patch Manager agent (Patch Manager agent)

Communications in a HPCA Environment

Figure 1 on page 19 presents an overview of the various types of communications and relationships that are possible in a HPCA environment.

Figure 1 Communications Overview



SSL Version Parity

To communicate using SSL, HPCA servers and agents must use the same version of SSL.

- This version of HPCA supports versions 2.0 and 3.0 of the SSL protocol.
- Secure (SSL) communications between version 4.*x* HPCA products and this version of HPCA products is not supported—unless the necessary maintenance has been applied, as described in the following section.

SSL Parity Maintenance

In order to enable SSL communications between a HPCA Configuration Server (version 4.5*x*) and version 5.1*x* HPCA agents, the following maintenance packages must be applied to the HPCA Configuration Server. Doing so will bring it up to the current level, and allow the 4.5*x* code to understand the more recent versions of the SSL protocol.

⚠ These fixes must be applied in the order in which they are listed.

1. Service Pack (SP) 5
2. Fix **6018**
3. Fix **6037**
4. Fix **6038**
5. Fix **6040**

Summary

- Secure Sockets Layer (SSL) is a cryptographic protocol that enables *secure communications* across a network.
- SSL implements mutual authentication by using *public key cryptography*.
- SSL uses a *public key* and a *private key* to encrypt and decrypt messages.
- A certificate must be obtained from a *Certificate Authority*. It contains:
  - The *server's public key*,
  - The *server name*, and
  - A *signature* from a Certificate Authority.
- SSL uses *ciphers* and *hash functions* to encrypt messages.
- Two-way SSL communication requires each server and each client to have a *truststore* and a *keystore*.
- To ensure SSL encryption viability with the HPCA products:
  - HPCA servers must have a *public key*, a *private key*, and a *CA public key*.
  - HPCA agents must have a *CA public key*.

2 Setting up Certificates for SSL

At the end of this chapter, you will have:

- A better understanding of how to set up certificates in your HPCA environment
- An understanding of the *capabilities and limitations* of the Certificate Generation Utility
- Become familiar with the *cipher suite* that is used by the Certificate Generation Utility
- A basis from which to decide whether to use an existing certificate or generate a new certificate

If you are already creating certificates in your environment with existing tools, skip to the next chapter Configuration and Use, starting on page 37.

The Certificate Generation Utility

For testing purposes, HP provides a **Certificate Generation Utility**. This utility makes it easy to create self-signed certificates for testing. It will be used in this chapter to demonstrate the process for setting up SSL for HPCA.

This utility is intended for testing purposes *only* and should **not** be used in a production environment.

Before using this utility, please consider the following:

- The Certificate Generation Utility is **not** a supported HP Client Automation product.
- The Certificate Generation Utility is provided *free of charge*.
- The Certificate Generation Utility is used at *your own discretion*; HP Technical Support **will not** address any issues regarding its use or functionality.

UNIX Note

HP's **Certificate Generation Utility** can generate certificates on Windows platforms only. However, once generated on a Windows system, certificates can be copied over to and used on UNIX platforms.

Locating the Certificate Generation Utility

The Certificate Generation Utility can be found on the HPCA installation media (Classic version) in:

```
INFRASTRUCTURE\extended_infrastructure\certificate_mgmt
```

In order to perform the tasks that are outlined in this chapter, copy the `certificate_mgmt` directory from the HPCA media to a directory on the local machine, such as:

```
C:\temp\certificate_mgmt
```

Be sure to use the Certificate Generation Utility that is provided with HP Client Automation version 5.10 or later. There is a known problem with earlier versions of the Certificate Generation Utility on Windows 2003 Server systems.

Setting up a Certificate

The first step required to set up SSL is to make sure that each system that will be authenticated has a private key and a signed certificate. If you already have a private key for this system, you can use it to generate a certificate. There are no other inputs required.

Using an Existing Private Key

If you already have a private key in `PEM` file format, follow these steps:

1. In the `certificate_mgmt\servers` directory, create a new directory called `hostname`.

   In this case, `hostname` is the name of the server for which a signed certificate is to be created. For example:

   ```
   certificate_mgmt\servers\cmserver1
   ```

2. Copy your private key `PEM` file into the directory that you just created.
3. Rename the `PEM` file that you just copied as follows: *`hostname`*`-prvkey.pem`

   For example:

   ```
   certificate_mgmt\servers\cmserver1\cmserver1-prvkey.pem
   ```

Generating a Signed Certificate

This section provides instructions for creating signed certificates that will be used for SSL configuration. There are three ways that you can generate a signed certificate:

- *Self-signing*

  This is the most convenient but least secure of the three options. Use this strictly for testing.

- Via a *generated CA*

  This option is more secure than self-signing, but not secure enough for a production environment. It creates a new CA with the parameters that you specify, and this new CA signs the certificate.

- Via a *trusted CA*

This is the most secure option of the three. In a production environment, be sure to use certificates that have been signed by a trusted CA.

This task will use the Certificate Generation Utility to demonstrate each of these three options.

Server Names

When you generate a certificate or a certificate request, you must specify the server name. You can use the simple host name (for example, `cmserver1`) or the fully qualified host name (for example, `cmserver1.mycorp.com`).

The name that you specify should match the name that will be used in the URL when this server is accessed.

➤ In the case of Enterprise Manager, the server name used to generate the SSL keys *must* match the server name specified in the `console.properties` file. See HPCA Enterprise Manager on page 46.

Option 1: Generating a Self-Signed Certificate

1 From the `certificate_mgmt` directory, run the following command:

```
cert_mgr create self
```

2 Provide the following information at the prompts:

Parameter	**Example**
Server name (becomes the CN)	**`cmserver1`**
Country name	**`US`**
State or province name	**`California`**
Locality Name	**`Sacramento`**
Organization name	**`Mycompany`**
Organizational unit name	**`IT`**

This information is used to create the **Distinguished Name (DN)** for the certificate. The DN is a unique identifier that is used to provide a name that is unique to the certificate. The DN is derived from the **Common Name (CN)** of the server and the other parameters that you specify.

The components of the DN, including the CN, are visible in the `cert.txt` file, as shown here:

```
Subject: C=US, ST=CA, L=Sacramento, O=MyCompany, OU=IT,
CN=cmserver1
```

➤ It is important that the CN part of the certificate's DN be the same as the server's host name. This is vital to the client trusting that it is communicating with the expected host.

After the utility finishes, the server certificate, private key, and related files are located in the `certificate_mgmt\servers\`*`hostname`* directory. For example, if the server name entered is **`cmserver1`**, the following files are generated:

```
certificate_mgmt\servers\cmserver1\cmserver1-cert.pem

certificate_mgmt\servers\cmserver1\cmserver1-cert.txt

certificate_mgmt\servers\cmserver1\cmserver1-prvkey.pem

certificate_mgmt\servers\cmserver1\cmserver1-signer.pem

certificate_mgmt\servers\cmserver1\cmserver1-signer.txt

certificate_mgmt\servers\cmserver1\cmserver1-cert.rnd

certificate_mgmt\servers\cmserver1\cmserver1-keystore.txt

certificate_mgmt\servers\cmserver1\cmserver1-keystore.jks

certificate_mgmt\servers\cmserver1\cmserver1-truststore.txt

certificate_mgmt\servers\cmserver1\cmserver1-truststore.jks
```

➤ The `keystore` and `truststore` files are generated only when the JAVA_HOME environment variable points to a Java runtime environment (JRE). See Keystore and Truststore Files on page 33 for more information.

After you have verified that your files were correctly generated, proceed to Configuration and Use on page 37.

⚠ Self-signed certificates are adequate for testing purposes *only*; they should **not** be used in a production environment.

Option 2: Generating a Re-usable Certificate Signed by a Generated CA

2 From the `certificate_mgmt` directory, run the following command.

```
cert_mgr create signed
```

3 Provide the following information at the prompts.

Parameter	**Example**
Server name (becomes the CN)	**cmserver1**
Country name	**US**
State or province name	**California**
Locality Name	**Sacramento**
Organization name	**Mycompany**
Organizational unit name	**IT**

➤ The first time you run this command, you will also be prompted for information about the certificate authority (CA) that you are generating. On subsequent runs, it will not prompt you.

The utility generates two sets of files in this case.

— The first set consists of the server certificate and related files, which are located in the `certificate_mgmt\servers\`*`hostname`* directory (see the description of the files under Option 1 on page 26).

— The second set consists of the CA files. These are located in the `certificate_mgmt\ca` directory.

For example, if the server name is specified as **cmserver1**, the following files are generated:

```
certificate_mgmt\servers\cmserver1\cmserver1-cert.pem

certificate_mgmt\servers\cmserver1\cmserver1-cert.txt

certificate_mgmt\servers\cmserver1\cmserver1-prvkey.pem

certificate_mgmt\servers\cmserver1\cmserver1-signer.pem

certificate_mgmt\servers\cmserver1\cmserver1-signer.txt

certificate_mgmt\servers\cmserver1\cmserver1-cert.rnd

certificate_mgmt\servers\cmserver1\cmserver1-keystore.txt
```

```
certificate_mgmt\servers\cmserver1\cmserver1-keystore.jks

certificate_mgmt\servers\cmserver1\cmserver1-truststore.txt

certificate_mgmt\servers\cmserver1\cmserver1-truststore.jks
certificate_mgmt\ca\ca.rnd

certificate_mgmt\ca\ca-cert.pem

certificate_mgmt\ca\ca-prvkey.pem

certificate_mgmt\ca\ca-index.txt

certificate_mgmt\ca\ca-index.txt.attr

certificate_mgmt\ca\ca-index.txt.old

certificate_mgmt\ca\ca-serial

certificate_mgmt\ca\ca-serial.old
```

➤ When you use the **signed** option, the Signing Authority Certificate is copied from the certificate_mgmt\ca directory.

If the certificate_mgmt\ca\ca-cert.pem file already exists, that file will be used. Otherwise, it will be created on the first run and used for generating subsequent certificates.

➤ The keystore and truststore files are generated only when the JAVA_HOME environment variable points to a Java runtime environment (JRE). See Keystore and Truststore Files on page 33 for more information.

After you have verified that your files were correctly generated, proceed to Configuration and Use on page 37.

⚠ This method of generating signed certificates is adequate for testing purposes *only* and should **not** be used in a production environment.

Option 3: Generating a Certificate Signed by a Trusted CA

These steps show you how to use the Certificate Generator Utility to generate a private key and certificate request that you can then send to a trusted CA. This might be an external CA, such as Verisign or Thawte, or a CA that your company or institution owns and administers.

1. From the certificate_mgmt directory, run the command

   cert_mgr create request

2. Provide the following information at the prompts:

Parameter	**Example**
Server name (becomes the CN)	cmserver1
Country name	US
State or province name	California
Locality Name	Sacramento
Organization name	Mycompany
Organizational unit name	IT

After the utility finishes, the certificate request, private key, and related files are located in the `certificate_mgmt\servers\`*`hostname`* directory. For example, if the server name is specified as **`cmserver1`**, the following files are generated:

```
certificate_mgmt\servers\cmserver1\cmserver1.rnd

certificate_mgmt\servers\cmserver1\cmserver1-prvkey.pem

certificate_mgmt\servers\cmserver1\cmserver1-request.pem

certificate_mgmt\servers\cmserver1\cmserver1-request.txt
```

3 Request a signed certificate by sending the *`hostname`*`-request.pem` to your signing authority.

Be sure that the server certificate that is purchased is a **base-64 encoded x.509** certificate. This is typical for certificates that are generated for the Apache Freeware (ModSSL or OpenSSL) Server.

For additional information about obtaining and installing a certificate from an external CA, see Obtaining a Certificate from a Certificate Authority on page 31.

4 When you receive this signed certificate from your signing authority, paste it into the *`servers\hostname\hostname`*`-cert.pem` file.

5 Paste the Signing Authority Certificate (must be in PEM format) into the *`servers\hostname\hostname`*`-signer.pem` file.

You now have a private key, a signed certificate, and the signing authority certificate files that are needed for product configuration. See HPCA Reporting Server, starting on page 40, and HPCA Enterprise Manager, starting on page 46.

Additional Information about Certificates and Keys

This section provides more detailed information about obtaining and installing signed certificates from external certificate authorities. It also contains information about keystores and truststores.

Obtaining a Certificate from a Certificate Authority

To get a signed certificate from a certificate authority (CA), you will need a **Server Certificate Request** (SCR) file. The Certificate Generation Utility creates an SCR file with the following name:

hostname-request.pem

To have the SCR file signed and returned, follow the procedure that is required by your trusted CA. Typically, the SCR file must be opened in a text editor, its text copied to a clipboard, and then pasted into a text field on the signing CA's web page.

To issue a signed certificate, the signing CA will also require proof-of-identity and authority—such as your company's DUNS number, Articles of Incorporation, Partnership Papers, or Business License.

If you open the SCR file with a text editor, the contents will look similar to this:

```
-----BEGIN CERTIFICATE REQUEST-----
MIIDZTCCAs6gAwIBAgIBADANBgkqhkiG9w0BADQFADCBhDELMAkGA1UEBhMCVVMx
CzAJBgNVBAgTAkNBMRAwDgYDVQQHEwdGcmVtb250MQ8wDQhDVQQKEwZTeWdhdGUx
EDAOBgNVBAsTB0FsdHZpZXcxEDAOBgNVBAMTB0FsdHZpZXcxITAfBgkqhkiG9w0B
CQEWEmF2LWRpYWdAc3lnYXRlLmNvbTAeFw0wNDExMjQyMzMxNDJaFw0xNDExMjIy
MA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDmqDqSMV+c6wLDZ7joj5+wZcpKA8jC
4tQFlqH/exqGsyKTmIZ2PjLbuPbjVZIVct8SPVUgIlcsvmWcxO3HSXMAKQl+dG89
Mf3XLTukzlz5LqoVJzuDoLQVcm7Ddx0iff+FLwRhsjl53KQqoRYucLOopirXYc6R
8T+XMo3tkd4q=
-----END CERTIFICATE REQUEST-----
```

Installing a Signed Server Certificate

When the signed SCR file is returned from the public CA:

1. In the signed SCR file's name, change the **request** (request) to **cert** (certificate). For example, change

   hostname-request.pem

to

hostname-cert.pem.

The SCR file might have a different name when it is returned from the CA.

2 Place the renamed SCR file (*hostname*-cert.pem) in the appropriate folder. Where this file should be placed will depend on the specific type of HPCA server that you are working with. See Configuration and Use starting on page 37 for detailed instructions for each HPCA server type.

3 Restart the server, and examine its log to verify that the SSL Manager task starts correctly and successfully verifies the CA certificate and server certificate.

Installing the Private Key File

The Certificate Generation Utility also generates a **private key** in the form of the following PEM file:

hostname-prvkey.pem.

To install the private key, place this file in the appropriate directory on the server. See Configuration and Use starting on page 37 for the specific location of this directory for each type of HPCA server.

If you open the private key file with a text editor, the contents will look similar to the following:

```
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: DES-CBC,6EC0947550541AAB

1MV8Y4rkywlYn30yUB5ULtKLfj0YSzX+KZvxCeuw+9x95x1Ikvej4b8iBDuEOaTR
MIIDZTCCAs6gAwIBAgIBADANBgkqhkiG9w0BADQFADCBhDELMAkGA1UEBhMCVVMx
MzMxNDJaMIGEMQswC7YDVQQGEwJVUzELMAkGA1UECBMCQ0ExEDAOBgNVBAcTB0Zy
ZW1vbnQxDzANBgNVBAoTBlN5Z2F0ZTEQMA4GA1UECxMHQWx0dmlldzEQMA4GA1UE
H1OkihMe0Ny94uj8a6ccMJ+1kRj2grVmaw8tJi+6G76NXhvZvwumfHZMtnhKUKth
Mf3XLTukzlz5LqoVJzuDoLQVcm7Ddx0iff+FLwRhsjl53KQqoRYucLOopirXYc6R
8T+XMo3tkd4q=
-----END RSA PRIVATE KEY-----
```

In order to maintain compatibility with industry standards, HP has adopted the RSA crypto-system method of obtaining certificate requests. The RSA crypto-system is a public key crypto-system that offers encryption and digital signatures (authentication). In the private key shown above the key type (**RSA**) is indicated at the beginning and end of the file.

Keystore and Truststore Files

This section offers information about generating keystore and truststore files, as well as password-changing considerations.

This information pertains only to Java applications. At present, this information applies to Enterprise Manager only.

Generating Keystore and Truststore Files

In order to generate keystore and truststore files using the Certificate Generation Utility, the JAVA_HOME environment variable must point to a **Java Runtime Environment** (JRE). If this variable is not set, you set it by modifying the `cert_mgr.cmd` file.

1. In the `cert_mgr.cmd` file locate the line:

   ```
   rem set JAVA_HOME=C:\Program Files\Java\j2re1.5.0_10
   ```

2. Remove `rem` from this line so that it is no longer commented out.
3. Change the value to the desired JRE path.
4. Save and close the `cert_mgr.cmd` file.

If JAVA_HOME does not point to a JRE, the Certificate Generation Utility will not generate the following files:

`certificate_mgmt\servers\`*`hostname`*`\`*`hostname`*-keystore.txt

`certificate_mgmt\servers\`*`hostname`*`\`*`hostname`*-keystore.jks

`certificate_mgmt\servers\`*`hostname`*`\`*`hostname`*`-truststore.txt`

`certificate_mgmt\servers\`*`hostname`*`\`*`hostname`*`-truststore.jks`

Changing the Keystore File Password

The default Java value is used for the keystore password in the `server.xml` file. You can change this password, and update the `server.xml` file accordingly with the attribute **`keypass`**, as shown below.

```
<Connector port="8443" maxHttpHeaderSize="8192" maxThreads="150"
minSpareThreads="25" maxSpareThreads="75" enableLookups="false"
disableUploadTimeout="true" acceptCount="100" scheme="https"
```

```
secure="true" clientAuth="false" sslProtocol="TLS"
keystoreFile="/keystore.key" keypass="your_keystore_password"/>
```

➤ The truststore/keystore password and private key password must match.

If you use the Certificate Generation Utility to generate certificates, you can specify passwords for the keystore and private key by using the **-trustpass** and **-keypass** options, respectively. For example:

```
cert_mgr create signed -hostname cmserver1 -trustpass
myn3wp4ssw0rd -keypass myn3wp4ssw0rd
```

```
cert_mgr create self -hostname cmserver1 -trustpass
myn3wp4ssw0rd -keypass myn3wp4ssw0rd
```

```
cert_mgr create request -hostname cmserver1 -keypass
myn3wp4ssw0rd
```

Note that the **request** option does not generate a truststore or keystore, so only the private key option is pertinent.

Summary

- HP provides a *Certificate Generation Utility* that makes it easy to create self-signed certificates.
- There are three ways to generate a signed certificate:
  - *Self-signing*
  - Via a *generated CA*
  - Via a *trusted CA*
- To get a signed certificate from a CA, you need a *Server Certificate Request (SCR) file*.

3 Configuration and Use

At the end of this chapter, you will know how to:

Configure secure connections for either Core and Satellite version of HPCA or for the Classic version of HPCA

Core and Satellite:

- Core and Satellite via HTTPS

⚠ If your environment uses Core and Satellite servers, first read the *Core and Satellite Servers Getting Started Guide* as the installation, configuration, and troubleshooting information in that guide may override the information in this guide.

Classic:

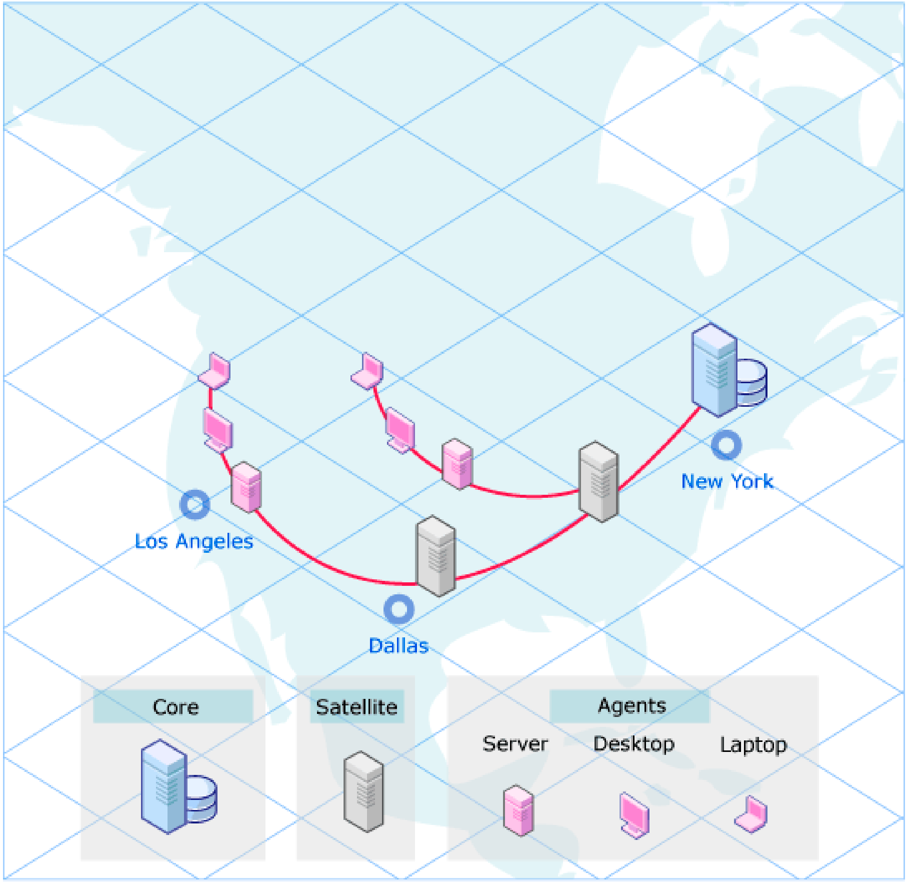
- HPCA Reporting Server via HTTPS
- HPCA Enterprise Manager via HTTPS
- HPCA Messaging Server via HTTPS
- HPCA Configuration Server via TCPS
- HPCA Distributed Configuration Server via HTTPS
- HPCA Patch Manager Server via TCPS
- HPCA Integration Server components:
  - HPCA Proxy Server via HTTPS
  - HPCA Policy Server via HTTPS and LDAPS
- HPCA Portal via HTTPS and LDAPS
- HPCA Application Usage Manager via HTTPS and LDAPS
- Set up HPCA Agents to use SSL

Core and Satellite Overview

This section describes how to implement SSL functionality in your Core and Satellite HPCA environment in order to secure the communications between HPCA servers and HPCA agents.

The consolidated Core and Satellite configuration simplifies configuration of SSL certificates.

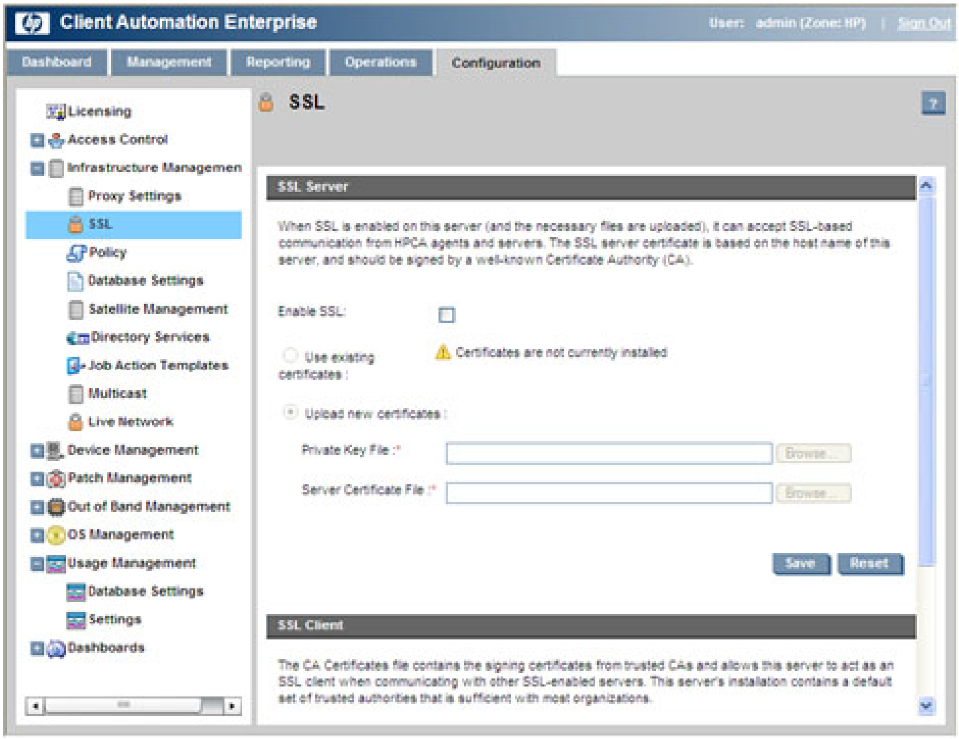
Figure 2 A Typical HPCA Environment



Configuring Core and Satellite with SSL Certificates

A single screen is used to configure SSL certificates in the HPCA Core and Satellite environment.

1. Select the Configuration tab.
2. Navigate to Infrastructure Management → SSL.
3. Enable SSL.
4. Enter the Privacy Key File name.
   (You can use the Browse button to navigate to the file.)
5. Enter the Server Certificate File name.
   (You can use the Browse button to navigate to the file.)
6. Click Save.



SSL Server

When SSL is enabled on this server (and the necessary files are uploaded), it can accept SSL-based communication from HPCA agents and servers. The SSL server certificate is based on the host name of this server, and should be signed by a well-known Certificate Authority (CA).

Enable SSL: □

○ Use existing certificates : ⚠ Certificates are not currently installed

● Upload new certificates :

Private Key File :* Browse...

Server Certificate File :* Browse...

Save Reset

SSL Client

The CA Certificates file contains the signing certificates from trusted CAs and allows this server to act as an SSL client when communicating with other SSL-enabled servers. This server's installation contains a default set of trusted authorities that is sufficient with most organizations.

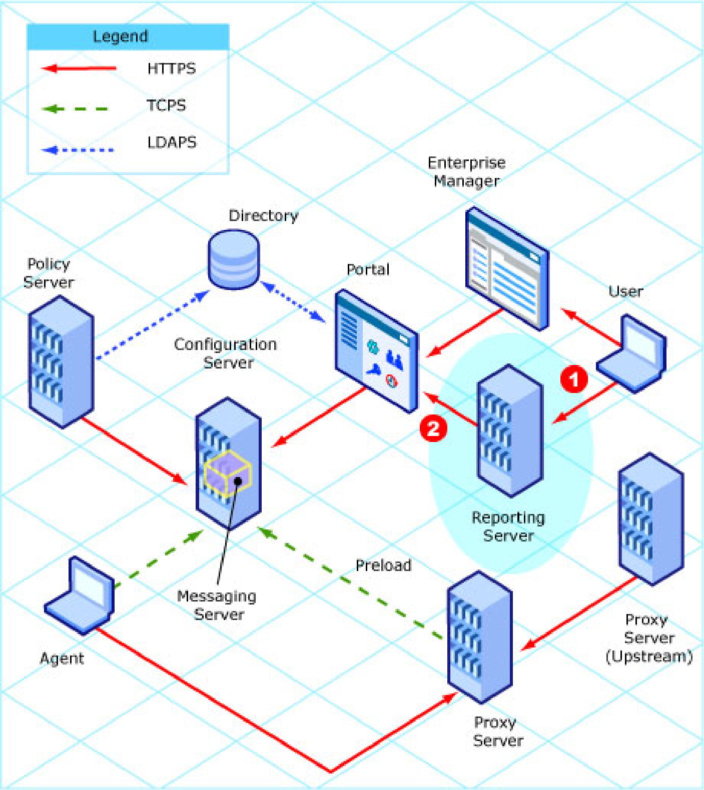
Classic Overview

This chapter describes how to implement SSL functionality in your Classic HPCA environment in order to secure the communications between HPCA servers and HPCA agents.

HPCA Reporting Server

The steps in this section detail how to set up the HPCA Reporting Server to create a secure (**HTTPS**) connection when using web services to connect to the Portal.

Figure 2 HPCA Reporting Server



Legend

1 Connection from the HPCA Reporting Server to the Portal

2 Connection from a user's browser to the HPCA Reporting Server

HPCA Reporting Server to the HPCA Portal

1. Edit the `rrs.cfg` file that is located in the HPCA Reporting Server `etc` folder.

   (Alternatively, you can use the web-based setup for HPCA Reporting Server.)

   Within the `::rrs::packconfig "" {}` section, add or modify the following entries:

   ➤ If you prefer to use the defaults, the following edits are not necessary.

   — `SSL_CADIR`: The CA Certificates directory.

     If left blank, this will default to `etc\CACertificates`.

   — `SSL_CAFILE`: The CA Certificates file.

     If left blank, this will default to `cacert.pem`.

2. Copy the `CA Certificates` file (for example, `cmserver1-signer.pem`) into the directory that is specified for `SSL_CADIR`.

   The default is `etc\CACertificates`.

   ➤ This step is needed only if your certificate is *not* signed by an established and trusted CA.

3. Configure the following parameters in the HPCA Reporting Server configuration file, `rrs.cfg`.

   a. Configuring HPCA Reporting Server to authenticate against the Portal:

      – `RMPLOGON`: Enable/disable Portal logon support

      – `RMPIP`: The fully qualified host name (*localhost* is acceptable) of the Portal server

      – `RMPPORT`: The port of the Portal server (**`443`** if SSL is used)

      – `RMPUSESSL`: Enable/disable use of SSL web services b Configuring HPCA Reporting Server to use web services to populate its Directory Browser:

> All changes here are in the `LDAP` portion of the `rrs.cfg` file.

- `TYPE`: **`rmp-ws`**
- `SERVER`: The fully qualified host name of the Portal server
- `PORT`: The port of the Portal server (**`443`** if SSL is used)
- `USER`: The Portal *service account user ID* (for example, **`admin`**)
- `PASS`: The Portal *service account password* (for example, **`secret`**)
- `USESSL`: **`1`** (to enable SSL support)

4 Verify that you can logon to the HPCA Reporting Server and that the Directory Browser appears properly.

> You will only be able to verify this after completing the steps to turn on SSL in the Portal, HPCA Portal, 66. (These cannot be verified using the Reporting tab in Enterprise Manager.)
>
> Verify SSL is enabled in the Portal by going to http://localhost <apache-port>/<rrs-alias>. Both of these settings are user-specified and may vary, but are typically set as follows:
>
> <apache-port>: 80
>
> <rrs-alias>: reportingserver

On the HPCA Reporting Server logon page, there should be a lock icon; this indicates that SSL web services are enabled.

User's Browser to the HPCA Reporting Server

> The HPCA Reporting Server does not provide an SSL version of the Apache web server. You can either use a current implementation or download the SSL version of Apache from one of the following sites.
>
> - `http://httpd.apache.org/download.cgi`
> - `http://archive.apache.org/dist/httpd/binaries/win32/apache_2.2.6-win32-x86-openssl-0.9.8e.msi`
>
> The second site is a direct link to Apache SSL version 2.2.6 for Windows.

1. Make sure that the following pre-requisites are in place on the HPCA Reporting Server host system:

   — A valid server certificate file and a private key file. If you are using the Certificate Generation Utility to generate certificates, these files are:

   ```
   servername-cert.pem
   ```

   ```
   servername-prvkey.pem
   ```

   where *servername* is the name of the HPCA Reporting Server.

   — Apache is installed.

   — The HPCA Reporting Server is installed. This enables the HPCA Reporting Server to detect the Apache SSL Secured Server at installation time if non-SSL connections are disabled.

2. Stop the Apache 2.2 service.

3. Perform the following steps to enable SSL in `httpd.conf`:

   a. Locate the `httpd.conf` file, and open it in Notepad. By default, this file is located in:

   ```
   C:\Program Files\Apache Software Foundation\
   Apache2.2\conf
   ```

   b. Search for the following string (note that it is preceded by a comment character, **#**):

   ```
   #LoadModule ssl_module modules/mod_ssl.so
   ```

   And uncomment the line by removing the **#** character.

   c. Search for the following two lines:

   ```
   # Secure (SSL/TLS) connections
   #Include conf/extra/httpd-ssl.conf
   ```

   And uncomment the second line by removing the **#** character.

   d. *Optional*: Perform the following two steps only if you want to disable non-SSL connections.

      i. In the `httpd.conf` file, search for the following two lines:

      ```
      #Listen 12.34.56.78:80
      Listen 80
      ```

      Note: `Listen 80` might have been changed to a different port depending on your implementation.

      ii. Comment out the `Listen` line, as shown.

```
#Listen 80
```

e Save the `httpd.conf` file.

4 Perform the following steps to configure the SSL certificates:

a Locate the `extra/httpd-ssl.conf` file, and open it in Notepad. By default, this file is located in:

```
C:\Program Files\Apache Software Foundation\Apache2.2\
conf\extra
```

b Search for the string `Listen 443`.

c Change `443` to an available port (not 443) on which you want this server to listen for SSL connections.

d Search for the string `SSLCertificateFile`

The default value is:

```
SSLCertificateFile "C:/Program Files/Apache Software
Foundation/Apache2.2/conf/server.crt"
```

e Edit the path name in quotation marks to point to your SSL Server Certificate file. Be sure to use forward slashes.

If you are using the Certificate Generation Utility to generate certificates, this path should point to the location of the *`servername`*`-cert.pem` file.

f Search for the string: `SSLCertificateKeyFile`

The default value is:

```
SSLCertificateKeyFile "C:/Program Files/Apache Software
Foundation/Apache2.2/conf/server.key"
```

g Change the path to point to your private key file. Be sure to use forward slashes, not backslashes.

If you are using the Certificate Generation Utility to generate certificates, this value should point to the *`servername`*`-prvkey.pem` file.

h Search for the string `SSLMutex`.

The default value is:

```
SSLMutex "file:C:/Program Files/Apache Software
Foundation/Apache2.2/logs/ssl_mutex"
```

i Change this line to:

```
SSLMutex "default"
```

j Search for the string `<VirtualHost _default_:443>`.

k Change the value `443` to reflect the `Listen` port that is configured in step c above. This will enable SSL on that port.

l Save the `httpd-ssl.conf` file.

5 Restart the Apache2.2 service.

You should now be able to access HPCA Reporting Server via
`https://`*`servername`*`:`*`sslport`***`/`***`rrsurl`*

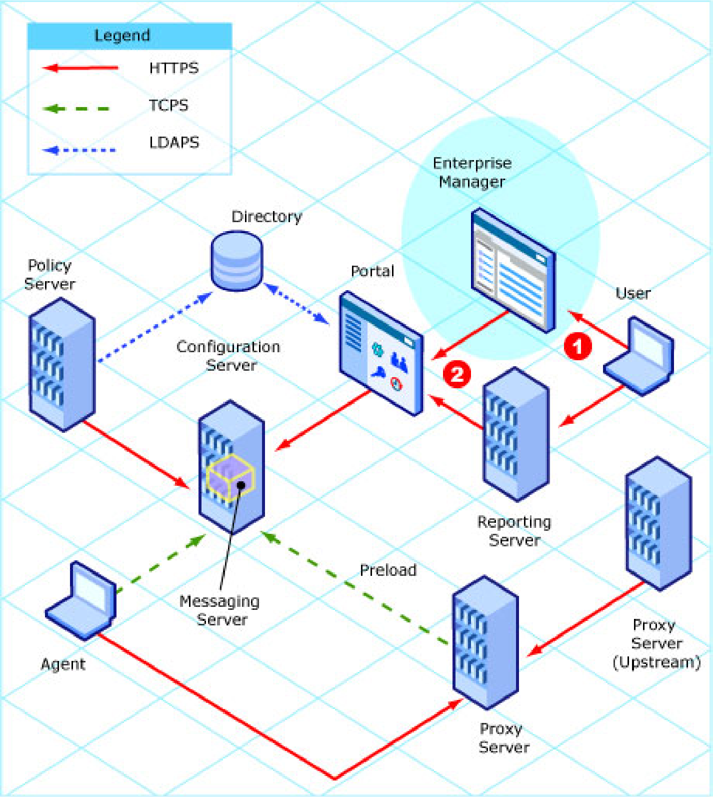
where *`rrsurl`* is the URL suffix that was configured when the HPCA Reporting Server was installed. An example is:

`https://localhost:443/reporting`

HPCA Enterprise Manager

This section details the two secure (**HTTPS**) connections that can be set up for an Enterprise Manager.

Figure 3 HPCA Enterprise Manager



Legend

1 Connection from a user's browser to the Enterprise Manager

2 Connection from the Enterprise Manager to the Portal

The Enterprise Manager can also be used to create a directory service connection for LDAPS.

For more information, refer to the section *To Configure LDAP Directory Services with SSL* in the *HP Client Automation Enterprise Manager User Guide*.

User's Browser to HPCA Enterprise Manager

In this scenario the Enterprise Manager is acting as a *server* so it must have a key pair and a signed certificate for the public key.

The Enterprise Manager is installed with a temporary certificate that will enable a secure connection. In other words, the URL `https://`*`emservername`*`:8443/em/` will work "out of the box."

However, this temporary certificate is not trusted. It should be replaced with a permanent certificate as described in Setting up Certificates for SSL on page 23.

The Enterprise Manager is written in Java, which uses keystores to hold the key pair and signed certificate.

The Enterprise Manager administrator must create the keystore file and can do so using the Certificate Generation Utility, as described in this section.

See Generating Keystore and Truststore Files, on page 33, and Changing the Keystore File Password, on page 33 for more information about keystore files.

If you create a privately signed or self-signed certificate using the Certificate Generation Utility, the keystore file is automatically created (see The Certificate Generation Utility starting on page 24 for more information).

If you generate a request to be signed by a trusted CA, you must import that certificate—after it has been signed—into the proper directory to create the keystore file (step 1 below).

Establishing a secure connection between the Enterprise Manager and a user's browser

Step 1 needs to be performed only if you are using a certificate tool *other than* the HP-provided Certificate Generation Utility; otherwise skip to step 2.

1 Use the following command to import a signed certificate into the proper directory:

```
certificate_mgmt\cert_mgr import signed -hostname xxxxx
-signedcert yyyyyy -signercert zzzzz
```

Parameter	Description	Example
xxxxxx	The fully qualified host name of the system to which the certificate belongs	**`cmserver1.mycorp.com`**
yyyyyy	The fully qualified path and file name of the signed certificate that was returned by the CA	**`C:\certs\cmserver1.mycorp.com-cert.pem`**
zzzzz	The fully qualified path and file name to the certificate of the signing CA	**`C:\certs\cmserver1.mycorp.com-signer.pem`**

2 If you used the Certificate Generation Utility with the **`self`** or **`signed`** option to generate your certificate—or you used a different method to obtain your certificate, and you imported it by completing step 1 above—the Java keystore file is located in:

`certificate_mgmt\servers\emsvrname\emsvrname-keystore.jks`

In this case, *`emsvrname`* is the host name of the Enterprise Manager server.

a Copy the Java keystore file, *`emsvrname`*`-keystore.jks`, to:

`EMInstallDir``\nonOV\jre\b\lib\security`

In this case, *`EMInstallDir`* is the directory in which the Enterprise Manager is installed—by default, `C:\Program Files\HP OpenView`.

b In the *`EMInstallDir`*`\nonOV\jre\b\lib\security` directory, rename the Java keystore file that you just copied.

Old name: *`emsvrname`*`-keystore.jks`

New name: `cm-ec.keystore`

3 To access the Enterprise Manager using a secure connection, use the following URL:

`https://`*`emservername`***`:8443/em`**

For example:

`https://cmserver1.mycorp.com:8443/em`

Disabling Non-SSL Access

After you configure SSL communication on the Enterprise Manager, you should explicitly disable non-SSL access.

To disable non-SSL access to the Enterprise Manager

1 In the *`EMInstallDir`*`\CM-EC\tomcat\conf\server.xml` file, locate the lines:

```
<!-- Define a non-SSL HTTP/1.1 Connector on port 8080 -->
<Connector port="8080" maxHttpHeaderSize="8192"
   compression="on"
compressableMimeTypes="text/html,text/xml"
   URIEncoding="UTF-8"
   keystoreFile="../../nonOV/jre/b/lib/security/cm-
ec.keystore"
   maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
   enableLookups="false" redirectPort="8443" acceptCount="100"
   connectionTimeout="20000" disableUploadTimeout="true" />
```

2 Comment out the connector block for `port=8080`.

```
<!-- Define a non-SSL HTTP/1.1 Connector on port 8080 -->
<!--
<Connector port="8080" maxHttpHeaderSize="8192"
   compression="on"
compressableMimeTypes="text/html,text/xml"
   URIEncoding="UTF-8"
   keystoreFile="../../nonOV/jre/b/lib/security/cm-
ec.keystore"
   maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
   enableLookups="false" redirectPort="8443" acceptCount="100"
   connectionTimeout="20000" disableUploadTimeout="true" />
-->
```

3 Save the file.

4 In the *EMInstallDir*\CM-EC\tomcat\webapps\em\WEB-INF\console.properties file, find the following line:

```
opeurl=http\://localhost\:8080/ope/resources
```

5 Change the protocol from `http` to **`https`** and the port from `8080` to **`8443`**:

```
opeurl=https\://localhost\:8443/ope/resources
```

6 Save the file.

7 Restart the `HPCA Enterprise Manager` service.

HPCA Enterprise Manager to HPCA Portal

In this case, the Enterprise Manager is the client and the Portal is the server.

The Enterprise Manager is written in Java so it uses a truststore file to store the certificates of trusted CAs.

Make sure that the Enterprise Manager truststore file contains the certificate for the CA that signed the Portal signed certificate.

See the sections, Generating Keystore and Truststore Files, on page 33, and Changing the Keystore File Password, on page 33 for more information on keystore files.

Establishing a secure connection between the Enterprise Manager and the Portal

1 Ensure that the Portal server is configured for SSL before continuing on to step 2. See To enable SSL so that the Portal can be accessed in a browser using HTTPS on page 67 for details.

The next step needs to be performed only if you are using a certificate tool *other than* the HP-provided Certificate Generation Utility; otherwise skip to step 3.

2 Use the following command to generate the truststore:

`certificate_mgmt\cert_mgr import signed -hostname` *`xxxxx`* **`-signedcert`** *`yyyyyy`* **`-signercert`** *`zzzzz`*

Parameter	Description	Example
xxxxxx	The host name of the Portal server Note: The host name must be the same as that used when configuring the Enterprise Manager for SSL.	**`cmserver1.mycorp.com`**
yyyyyy	The fully qualified path and file name of the signed certificate that was returned by the CA	**`C:\certs\cmserver1.mycorp.com-cert.pem`**
zzzzz	The fully qualified path and file name of the certificate of the signing CA	**`C:\certs\cmserver1.mycorp.com-signer.pem`**

This process imports the certificate files into the `servers\`*`cmsvrname`* directory that will be used in step 3.

3 If you used the Certificate Generation Utility with the **`self`** or **`signed`** option to generate your certificate—or you used a different method to obtain your certificate, and you imported it by completing step 2 above—the Java truststore file is located in:

`certificate_mgmt\servers\`*`cmsvrname`*`\`*`cmsvrname`*`-truststore.jks`

In this case, *`cmsvrname`* is the host name of the Portal server.

a Copy the Java truststore file, *`cmsvrname`*`-truststore.jks`, to:

`EMInstallDir``\nonOV\jre\b\lib\security`

In this case, *`EMInstallDir`* is the directory in which the Enterprise Manager is installed. In a typical installation, *`EMInstallDir`* is `C:\Program Files\HP OpenView`.

b In the *`EMInstallDir`*`\nonOV\jre\b\lib\security` directory, rename the Java truststore file that you just copied as follows.

Old name: *`cmsvrname`*`-truststore.jks`

New name: `cm-ec.truststore`

4. Go to *`EMInstallDir`*`\CM-EC\tomcat\webapps\em\WEB-INF`, and edit the `console.properties` file as indicated below:

   — Change protocol from `protocol=http\://` to: **`protocol=https\://`**

   — Change the value of `port` to the value that was used to configure the Portal SSL port. By default, this is port 443.

   — Change host=localhost to the fully qualified hostname, such as host=cmserver.hp.com.

   ➤ The host value is used to verify the certificate, and the certificate is issued for the fully-qualified hostname. If the host value does not match the certificate is deemed invalid and loading of the properties file stops, and the entire process stops

5. Restart the `HPCA Enterprise Manager` service to begin using the new truststore.

   ➤ If you are using a truststore with a password other than the default (**changeit**), you must change the following Windows registry key.

   ```
   HKEY_LOCAL_MACHINE\SOFTWARE\Apache Software
   Foundation\Procrun 2.0\HPCMEnterpriseManager\
   Parameters\Java
   ```

   Locate this key and modify the value of the following parameter.

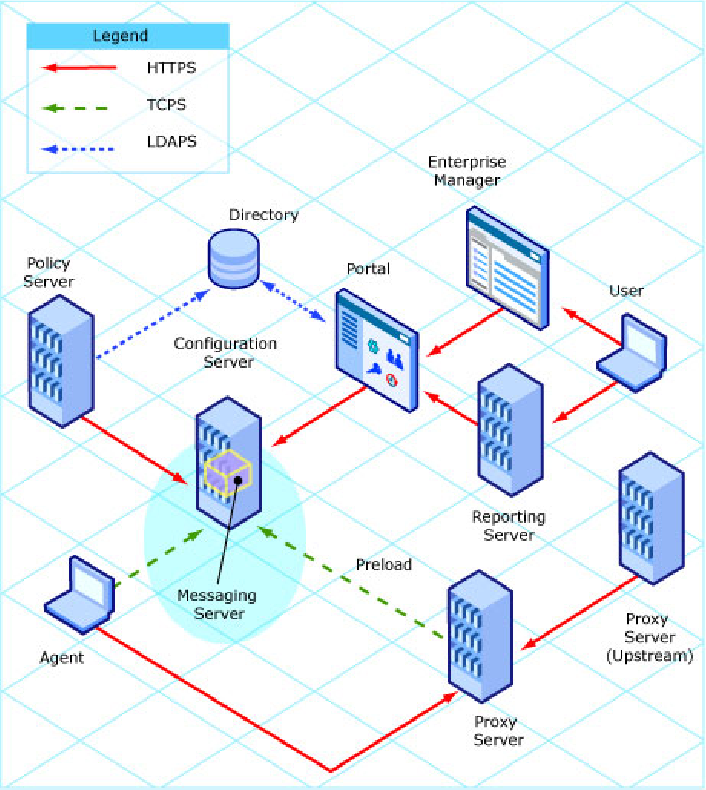
   `Djavax.net.ssl.trustStorePassword=`*`yourpassword`*

   Be sure to restart the `HPCA Enterprise Manager` service.

HPCA Messaging Server

The steps in this section detail how to set up the HPCA Messaging Server for secure (**HTTPS**) connections.

Figure 4 HPCA Messaging Server



The `Overrides Config` section of the HPCA Messaging Server configuration file, `rms.cfg`, has to be populated with the *certificate path*, *private key path*, and *secure port* values. The HPCA Messaging Server installation puts `cacert.pem` in the `/etc/CACertificates` directory.

Establishing a secure connection on the HPCA Messaging Server

1 Stop the HPCA Messaging Server service (**rms**).

2. Copy the private key and signed certificate into the HPCA Messaging Server `Certificates` directory.

   The default is `C:\Program Files\Hewlett-Packard\CM\MessagingServer\etc\Certificates`.

3. Navigate to the `MessagingServer\etc` directory, and open `rms.cfg` in a text editor.

   a. Verify that `module load tls` is uncommented.

   b. In the `Overrides Config` section, add the following parameters:

   ```
   Overrides Config {

   SSL_CERTFILE "C:/Program Files/Hewlett-Packard/CM/
   MessagingServer/etc/Certificates/myserver-cert.pem"

   SSL_KEYFILE "C:/Program Files/Hewlett-Packard/CM/
   MessagingServer/etc/Certificates/
   myserver-prvkey.pem"

   HTTPS_PORT "443"

   }

   module load tls
   ```

4. Save your changes and close `rms.cfg`.

5. Restart the HPCA Messaging Server service (rms).

6. Check the `rms.log` to ensure that the secure server has been started; look for the following message.

   ```
   MSG/HTTPD: secure httpd on tcp://0.0.0.0:443 started
   ```

7. To use SSL for outgoing HTTP posts, you must do the following things in the appropriate `cfg` file (or files):

   — Specify **`HTTPS`** as the TYPE

   — Use a URL with **`https`** specified

   — Include the secure port of the server that will be receiving the posts

   This update is required for the `rms.cfg` file or for any data delivery agent (`core.dda.cfg`, `inventory.dda.cfg`, etc.) that is configured in the HPCA Messaging Server environment.

   Example:

   ```
   msg::register secure1 {
      TYPE   HTTPS
   ```

```
   ADDRESS  {
     PRI  10
     URL  https://localhost:443/proc/inventory
   }
}
```

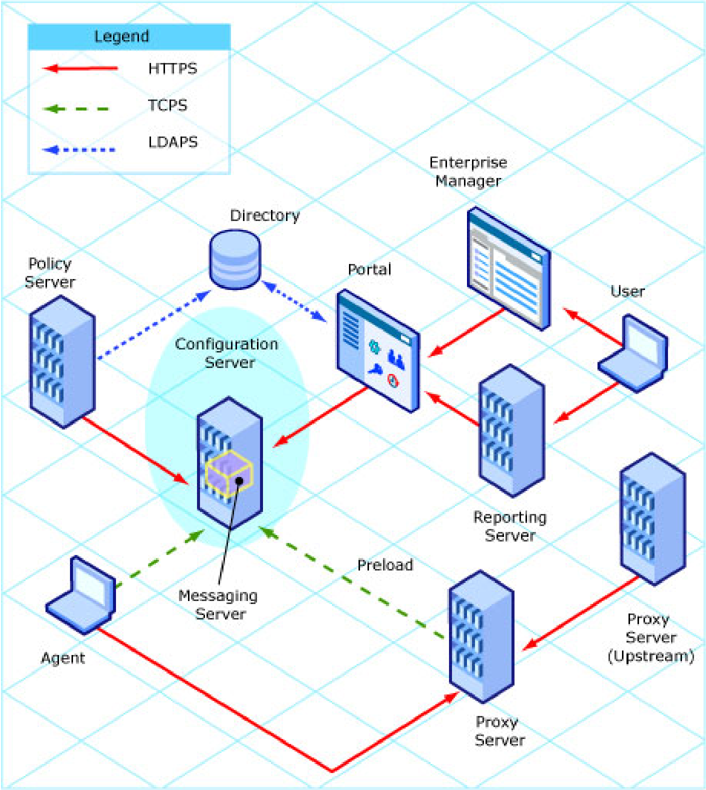
8 To use SSL for inbound HTTP posts, you should include the following changes in rms.cfg:

```
msg::register secure1 {
   TYPE   HTTPS
   PROTO  https

   PORT   3461
   USE    default

   URL    /proc/rim/default
   URL    /proc/xml/obj
```

HPCA Configuration Server

This section details how to set up the Configuration Server for secure (**TCPS**) connections.

Figure 5 HPCA Configuration Server



To confirm that the Configuration Server is configured for SSL support, use a text editor to open the `edmprof` file. This file is located in:

Windows: `<CSInstallDir>\bin`

UNIX: `/$HOME`

In this case, `<CSInstallDir>` is the directory where the Configuration Server is installed.

In the `edmprof` file, verify the following things:

- Verify that the MGR_ATTACH_LIST section contains the `zsslmgr CMD_LINE`, as shown here:

```
[MGR_ATTACH_LIST]
CMD_LINE = (zsslmgr) RESTART = YES
```

➤ You might need to uncomment this line in the `edmprof` file.

- Verify that the MGR_SSL section exists and is populated with the correct locations and file names, as shown here:

```
[MGR_SSL]

CA_FILE = C:\Program Files\Hewlett-Packard\CM\Configuration
Server\bin\CACertificates\

CERTIFICATE_FILE = C:\Program Files\Hewlett-Packard\CM\
ConfigurationServer\bin\Certificates\

KEY_FILE = C:\Program Files\Hewlett-Packard\CM\
ConfigurationServer\bin\Certificates\

SSL_PORT = 443
```

Table 2 below describes these settings.

Table 2 MGR_SSL Settings

Setting	Usage
CA_FILE	This setting is used to identify and locate the Certificate Authority's certificate. The CA certificate is usually stored in a file in **PEM** (Private Enhanced Mail) format. The value for this setting is the full path to a valid and existing certificate file. The SSL Manager task requires a CA certificate to start. An expired or corrupt CA certificate prevents the SSL Manager task from starting.
CERTIFICATE_FILE	This setting is used to identify and locate the server certificate of the HPCA server. The certificate is usually stored in a file in PEM format. The value for this setting is the full path to a valid and existing certificate file. The SSL Manager requires a certificate to start. An expired or corrupt certificate prevents the SSL Manager task from starting.
KEY_FILE	This setting is used to identify and locate the private key. The private key is usually stored in a file in PEM format. The value for this setting is the full path to a valid and existing key file. Usually the private key is stored in the same file as the server certificate, in which case you don't have to include KEY_FILE in the MGR_SSL section.
SSL_PORT	This setting is used to set the port that the SSL Manager should attend for client connections. The SSL protocol default port is **443**.

The Configuration Server looks for your certificates and keys in the directories specified in the MGR_SSL section of the `edmprof` file. Be sure that your files are located in the specified directories.

If you used the Certificate Generation Utility to create your certificates, follow these steps:

1 Copy the certificate_mgmt\CACertificates\cacert.pem file to the location specified by CA_FILE.

2 Copy the certificate_mgmt\servers\*hostname**hostname*-cert.pem file to the location specified by CERTIFICATE_FILE.

3 Copy the certificate_mgmt\servers\*hostname**hostname*-prvkey.pem file to the location specified by KEY_FILE.

   If KEY_FILE is not defined in the MGR_SSL section of the `edmprof` file, copy the *hostname*`-prvkey.pem` file to the location specified by CERTIFICATE_FILE.

In steps 2 and 3 above, *hostname* is the name of the system where the Configuration Server is installed.

HPCA Distributed Configuration Server

This section details the SSL considerations for secure Distributed Configuration Server (DCS) connections via **HTTPS**.

➤ The Distributed Configuration Server configuration file is `dmabatch.rc`. It can be found in the directory into which the DCS was installed.

⚠ If SSL functionality is used during a DCS synchronization, files that exceed 2GB in size will fail.

SSL Considerations

In order to enable SSL functionality, the following conditions must be met.

- The zsslmgr setting (CMD_LINE=(zsslmgr) RESTART=YES) must be present and enabled in the MGR_ATTACH_LIST section of the master and slave (*Source* and *Destination*) Configuration Server edmprof files.
- In the edmprof file of the master Configuration Server, the value of SSL_PORT in the MGR_SSL section must be different than the value of HTTPS_PORT in *risroot*\etc\httpd.rc (the Integration Server's configuration file).
- The port that is set in dmabatch.rc for:

  ```
  -https-port nnn
  ```

  must match the port that is set in the Overrides Config section of httpd.rc:

  ```
  HTTPS_PORT nnnn
  ```

SSL Port Settings

Table 3 below lists the SSL port settings that are in the DCS configuration file, dmabatch.rc. Also listed are the HPCA server configuration settings that they must match.

Table 3 DCS Configuration File Equivalents

dmabatch.rc Setting	Equivalent Setting	Location
-master-ssl-port	SSL_PORT	Source (master) Configuration Server edmprof file
-slave-ssl-port	SSL_PORT	Destination (slave) Configuration Server edmprof file
-https-port	HTTPS_PORT	httpd.rc file of Integration Server

SSL vs. non-SSL Configurations

You can switch between an SSL and a non-SSL configuration by adjusting the -ssl line of the DCS configuration file. Specify:

- 1 for an SSL configuration

- **0** for a non-SSL configuration

For example, if the SSL-enabled Configuration Server ports are **443**, and the SSL-enabled Integration Server port is **444**, the following could be put into `dmabatch.rc`.

```
array set O {
  -ssl               1
  -master-port       3464
  -master-ssl-port   443
  -slave-port        3464
  -slave-ssl-port    443
  -http-port         3466
  -https-port        444
}
```

HPCA Patch Manager Server

This section details how to set up the Patch Manager Server for secure (**TCPS**) connections.

Enable the Integration Server under which the Patch Manager Server is running, as documented in the section HPCA Integration Server starting on page 61.

Post-installation Notes

To establish a secure Security Patch Acquisition session, only the following Patch Manager Server configuration setting needs to be updated.

This can be done via the interface.

- Modify the Configuration Server URL to a *secure connection* value, such as:

  `tcps://Configuration_Server_machine:4430`

  Replacing a standard, non-secure TCP connection value, such as:

  `radia://machine_name:3464`.

HPCA Integration Server

This section details how to set up secure (**TCPS** and **HTTPS**) connections for the HPCA products that run under the Integration Server.

To enable SSL so that the Integration Server can be accessed in a browser using HTTPS

1 Navigate to the location into which the Certificate Generation Utility was copied.

2 Copy the following two files:

   `servers\`*`servername`*`\`*`servername`*`-cert.pem`

   `servers\`*`servername`*`\`*`servername`*`-prvkey.pem`

   In this case, *`servername`* is the name of the system where the Integration Server is installed.

3 Paste these files into:

   `C:\Program Files\Hewlett-Packard\CM\IntegrationServer\etc\Certificates.`

To confirm that the Integration Server is configured for SSL support (via **HTTPS**), use a text editor to open the `httpd.rc` file, which is located in the `IntegrationServer` directory, and confirm that the `Overrides Config` section has been added, as shown below.

```
Overrides Config {

SSL_CERTFILE "C:/Program Files/Hewlett-Packard/CM/
IntegrationServer/etc/Certificates/servername-cert.pem"

SSL_KEYFILE "C:/Program Files/Hewlett-Packard/CM/
IntegrationServer/etc/Certificates/servername-prvkey.pem"

HTTPS_PORT "443"
```

Table 4 on page 62 describes the settings of the `Overrides Config` section.

Table 4 Overrides Config Section Settings

Setting	Usage
SSL_CERTFILE	This setting is used to identify and locate the server certificate of the HPCA server. The certificate is usually stored in a file in **PEM** (Private Enhanced Mail) format. The value for this setting is the full path to a valid and existing certificate file. The SSL Manager requires a certificate to start. An expired or corrupt certificate prevents the SSL Manager task from starting.
SSL_KEYFILE	This setting is used to identify and locate the private key. The private key is usually stored in a file in PEM format. The value for this setting is the full path to a valid and existing key file. Usually the private key is stored in the same file as the server certificate, in which case you don't have to include KEY_FILE in the MGR_SSL section.
HTTPS_PORT	This setting is used to set the port that the SSL Manager should attend for client connections. The SSL protocol default port is **443**.

When the Integration Server is running you can connect to it, via HTTPS, by opening a web browser and typing

```
https://servername:ssl_port
```

To disable standard HTTP (leaving only HTTPS available), open the `httpd.rc` file and in the `Overrides Config` section set PORT to **-1**, as in:

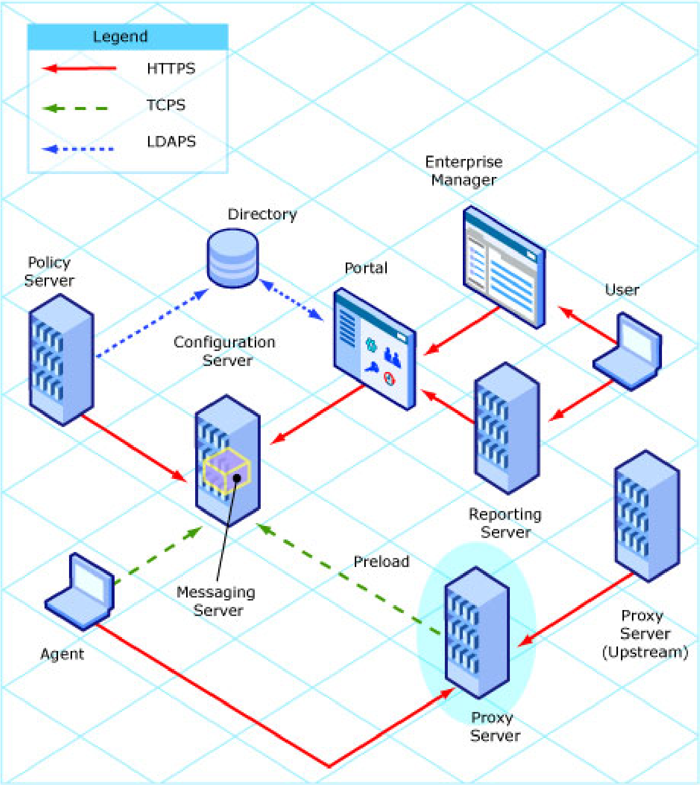
```
Overrides Config {

SSL_CERTFILE "C:/Program Files/Hewlett-Packard/CM/
IntegrationServer/etc/Certificates/servername-cert.pem"

SSL_KEYFILE "C:/Program Files/Hewlett-Packard/CM/
IntegrationServer/etc/Certificates/servername-prvkey.pem"

HTTPS_PORT 443

PORT -1
```

To configure LDAPS and HTTPS support for the Portal, see HPCA Portal on page 66.

HPCA Proxy Server

This section details how to enable SSL communications with a Proxy Server.

Figure 6 HPCA Proxy Server



To enable SSL communications with a Proxy Server, follow the instructions below to set up a **Server Access Profile** (**SAP**) in the Configuration Server Database via the Administrator Configuration Server Database Editor (Admin CSDB Editor]).

1 Log on to the Admin CSDB Editor.

2 Navigate:

File=**PRIMARY**, Domain=**CLIENT**, Class=**Server Access Profile (SAP)**.

3 Set ENABLED=**Y** for individual Instances, or to affect all Instances of the Class, set ENABLED=**Y** in the **_BASE_INSTANCE_**.

HPCA Proxy Server Preload

Proxy server preloading using SSL_TCPS does not work in Solaris and AIX UNIX.

To confirm that the Proxy Server preload Server is configured for SSL support, use a text editor to open the `rps.cfg` file, which is located in the `IntegrationServer` directory, and confirm that it has the following settings.

```
rps::init {
   -static-ssl   1
   -stager       0
```

HPCA Proxy Server Upstream Request

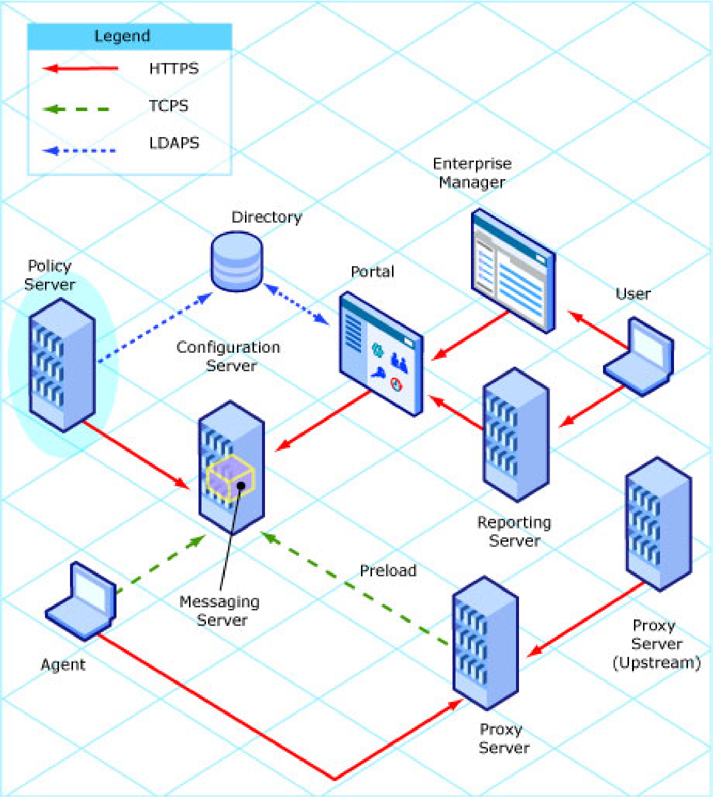
To confirm that the Proxy Server dynamic upstream request is configured for SSL support, use a text editor to open the `rps.cfg` file, which is located in the `IntegrationServer` directory, and confirm that it has the following settings.

```
rps::init {
   ...
   -dynamic-url  https://upstream:3466
```

HPCA Policy Server

This section details how to enable secure communications with a HPCA Policy Server.

Figure 7 HPCA Policy Server



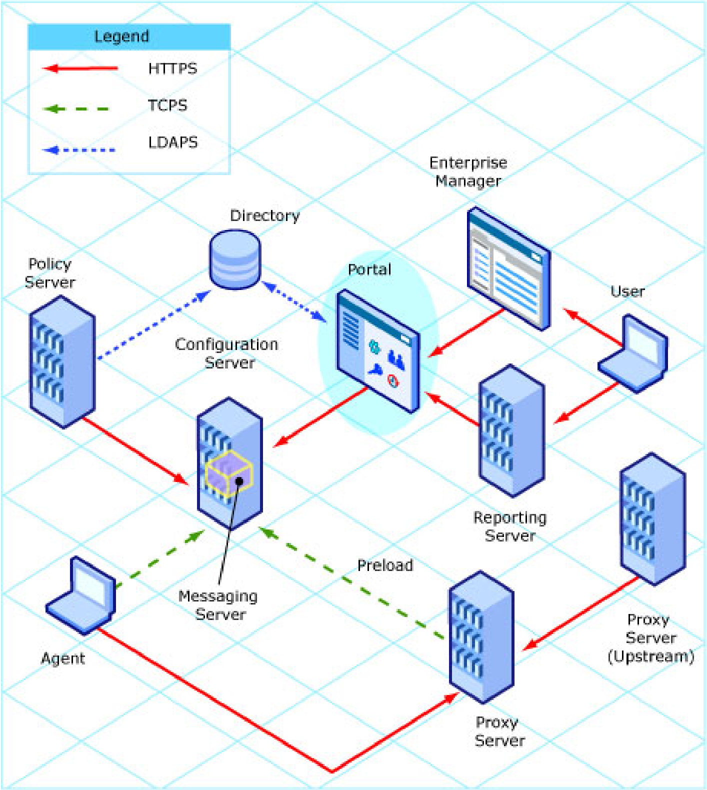
To confirm that HPCA Policy Server LDAP is configured for SSL (**LDAPS**) support, use a text editor to open the `pm.cfg` file, which is located in the `IntegrationServer/etc` directory. Verify that the following settings have been edited for secure LDAP communication. Use the following settings as an example.

```
ldap::init {

    TYPE      ldaps

    LDAP_CACERTDIR          etc/CACertificates
    LDAP_CACERTFILE       etc/CACertificates/cacert.pem
    LDAP_REQUIRE_CERT     demand
    PORT      636
}
```

HPCA Portal

This section details how to enable secure communications for a Portal.

Figure 8 HPCA Portal



This section details how to set up secure (**LDAPS** and **HTTPS**) connections for the Portal.

To confirm that the Portal is configured to connect to a secure LDAP directory using SSL (LDAPS), start the Portal service and check the following:

1. The `ldaps84.dll` and `ldaps82.dll` files are in the root `ManagementPortal` directory.

   The `ldaps84.dll` and `ldaps82.dll` files are unpacked by the `tls.tkd` module when the Portal service starts. If either `ldaps82.dll` or `ldaps84.dll` is missing, follow these steps:

   a. Stop the Portal service.

   b. Delete any existing `lpdaps82.dll` or `ldaps84.dlls` files in the `ManagementPortal` directories or path.

   c. Restart the Portal service.

2. A CA Certificate file containing the LDAP server's CA root certificate (public key) is in a local directory on the Portal.

   A default CA Certificate file, `cacert.pem`, is installed. This includes the public keys for Entrust, VeriSign, Inc., and G.E, and is located in

   ```
   C:\Program Files\Hewlett-Packard\CM\ManagementPortal\
   etc\CACertficates.
   ```

To enable SSL so that the Portal can be accessed in a browser using HTTPS

1. Navigate to the location into which the Certificate Generation Utility was copied.

2. Copy the following two files:

   servers\servername\servername-cert.pem

   servers\servername\servername-prvkey.pem

   In this case, *servername* is the name of the system where the Portal is installed.

3. Paste these files into:

   ```
   C:\Program Files\Hewlett-Packard\CM\ManagementPortal\
   etc\Certificates.
   ```

To confirm that the Portal is configured for SSL support, use a text editor to open the `httpd-managementportal.rc` file, which is located in the `ManagementPortal\etc` directory. Confirm that the `Overrides Config` section has been added, as shown below.

```
Overrides Config {
```

```
SSL_CERTFILE "C:/Program Files/Hewlett-Packard/CM/
ManagementPortal/etc/Certificates/servername-cert.pem"

SSL_KEYFILE "C:/Program Files/Hewlett-Packard/CM/
ManagementPortal/etc/Certificates/servername-prvkey.pem"

HTTPS_PORT "443"
```

Note that the slashes in these paths must be forward slashes on *all* platforms.

4. Confirm that the `tls.tkd` file is in the `modules` directory.
5. In a text editor, open the `http-managementportal.rc` file, and confirm that the `tls.tkd` module is loaded *before* the `rmp.tkd` module, as shown here:

   ```
   module load tls.tkd

   module load rmp.tkd
   ```

6. Restart the Portal.

When the Portal is running, you can connect to it—via HTTPS—by opening a web browser and typing

```
https://servername:ssl_port
```

For example:

```
https://cmserver1:443
```

To disable standard HTTP (leaving only HTTPS available), open the `httpd-managementportal.rc` file and in the `Overrides Config` section set PORT to **-1**, as in:

```
Overrides Config {

SSL_CERTFILE "C:/Program Files/Hewlett-Packard/CM/
ManagementPortal/etc/Certificates/servername-cert.pem"

SSL_KEYFILE "C:/Program Files/Hewlett-Packard/CM/
ManagementPortal/etc/Certificates/servername-prvkey.pem"

HTTPS_PORT 443

PORT -1
```

Note that the slashes in these paths must be forward slashes on *all* platforms.

To add a CA Root Certificate (Public Key) for the LDAPS Server

If the server that is hosting the LDAP directory is using a CA other than Entrust, VeriSign, Inc., or G.E., obtain and place the CA root certificate on a local directory of the Portal host machine. Then:

- Add the contents of the public key to the top of the default `cacert.pem` file,

  ➤ In order to allow for multiple LDAPS connections, add the contents of multiple public keys to the `cacert.pem` file.

  or

- Copy the CA root certificate file to a local directory on the Portal host machine.

To add a directory service connection for LDAPS

1 To enable the Portal to connect to an LDAP directory using SSL, log on to the Portal and navigate to **Zone** → **Configuration** → **Directory Services**.

2 In the Model Administration task group, click **Add Directory Service** and complete the entries that are needed for a directory service **type** of **ds-ldaps**.

3 Complete the Directory Service Properties for LDAPS by specifying the following.

  ➤ The URL, CA Certificate Directory, and CA Certificate File options require specific entries for an LDAPS connection.

  — Specify a Common Name.

  — Optionally, specify a Display Name and Description.

  — Optionally, specify a Startup type.

  — Select **ds-ldaps** as the Type.

  — Type the URL as shown below, substituting the items in < > with your specific values.

```
ldaps://<LDAP_hostname_in_certificate>:
<LDAP_secure_port>/<bind_User>@<domain>
```

  If this value does not match the server's common name, as specified in the LDAP server's certificate, the connection will fail. Therefore, if the subject line of the certificate specifies the CN= value using the fully-qualified DNS hostname, the URL must specify the fully-qualified DNS hostname.

<LDAP secure port>
specifies the LDAP secure port; the default port for LDAPS is **636**.

<bind User>@<domain>
defines the user and domain that will bind to the directory service.

— Specify the Password for the bind User that is specified in the URL.

— Optionally, type a Use to specify a fully-qualified domain at which to mount the directory service. If left blank, the common name will be used to mount the directory service at the highest level.

— In the CA Certificate Directory and CA Certificate File fields, specify the local directory and the file that contain the public key for the LDAP server. The default CA Certificate file that is installed by the Certificate Generation Utility is `cacert.pem`.

— Optionally, increase the LDAP Debug Level to 5 to create an LDAP Debug Log for troubleshooting the LDAP connection. If left at the default value of **0**, the LDAP Debug Log is suppressed.

For detailed information on specifying these properties, refer to the *HP Client Automation Portal Installation and Configuration Guide.* Review the section, Specifying LDAP or LDAPS Directory Service Properties.

4 Click **Submit**.

You will be redirected to the root of your LDAP directory at the base domain that was specified in the Use field.

Securing HPCA Portal-to-HPCA Portal Communications

After a secure Portal is established, the next step is to secure the client end of the connection. To do this, the public keys and the signed certificates that were previously created (for example, `ManagementPortal\etc\Certificates\`*`fully qualified DNS Hostname`*`-cert.pem`) must be shared.

The following instructions will use the references of a *master* Portal which will mount a *subordinate* Portal.

1 Make a new file, `cacert.pem`, in the `CACertificates` directory of the subordinate Portal (`ManagementPortal\etc\CACertificates\cacert.pem`).

2 Open `ManagementPortal\etc\Certificates\Master-`*`fully qualified DNS Hostname`*`-cert.pem` and from it, copy all the lines starting from (and including):

```
-----BEGIN CERTIFICATE-----
```

to

```
-----END CERTIFICATE-------
```

3 Paste these lines into the `cacert.pem` file that was created in Step 1.

4 Repeat steps 1 – 3, but copy the contents of the certificate on the subordinate Portal to a `cacert.pem` file on the master Portal.

The following file locations are for the certificate files for the master and subordinate Portals.

Master HPCA Portal

```
ManagementPortal\etc\CACertificates\cacert.pem
```

```
ManagementPortal\etc\Certificates\
Master-fully qualified DNS Hostname-cert.pem
```

```
ManagementPortal\etc\Certificates\
Master-fully qualified DNS Hostname-prvkey.pem
```

```
ManagementPortal\etc\Certificates\
Master-fully qualified DNS Hostname-request.pem
```

Subordinate HPCA Portal

```
ManagementPortal\etc\CACertificates\cacert.pem
```

```
ManagementPortal\etc\Certificates\
Subordinate-fully qualified DNS Hostname-cert.pem
```

```
ManagementPortal\etc\Certificates\
Subordinate-fully qualified DNS Hostname-prvkey.pem
```

```
ManagementPortal\etc\Certificates\
Subordinate-fully qualified DNS Hostname-request.pem
```

5 Add the references to the newly created `cacert.pem` file to the `httpd-managementportal.rc` file.

The revised configuration section will resemble the following.

```
Overrides Config {

SSL_CERTFILE "ManagementPortal/etc/Certificates/Subordinate-
fully qualified DNS Hostname-cert.pem"
SSL_KEYFILE "ManagementPortal/etc/Certificates/Subordinate-
fully qualified DNS Hostname-prvkey.pem"

HTTPS_PORT 4433

PORT       3466

LOG_LEVEL  3

SSL_CADIR  "ManagementPortal/etc/CACertificates"

SSL_CAFILE "cacert.pem"

}
```

By setting `PORT` to **`-1`** the non-secure port will be disabled.

This will lock down the Portal non-secure port and prevent it from accepting any RMA registrations.

6. Add the setting **`RMP_SECURE_RMP 1`** to the `etc/rmp.cfg` file, as shown in the following example.

   This will enable all Portal-to-Portal communications as *secure*.

   ```
   rmp::init {

       URL            /

       RMP_SECURE_RMP 1

   }
   ```

7. Confirm that the `tls.tkd` file is in the `modules` directory.

Closing Steps

- After completing all of the SSL configurations, start the Portal.
- Add the Directory Service in the master Portal—specifying the subordinate Portal—according to the instructions in the *HPCA Portal Guide*.
- The information that is needed for mounting the subordinate Portal using a secure DSML connection differs from that for mounting a Portal with a non-secure connection in that the URL that is specified must use the HTTPS protocol and the port that is specified must be the secure port of the subordinate Portal.

  The following is an example of an acceptable URL.

  https://subrmp:4443/proc/dsml

  where...

- **subrmp** is the *subordinate Portal hostname*
- **4443** is the *secure port*

HPCA Application Usage Manager

This section details **HTTPS** configuration procedures for HPCA Application Usage Manager.

This section details how to configure the HPCA Application Usage Manager Agent to use an SSL-secured Integration Server as its **collection point**. See HPCA Application Usage Manager Agent in a HPCA Environment, starting below.

> The collection point is a share point—created by the Integration Server—from which the HPCA Application Usage Manager transfers usage data.

Additionally, HPCA Application Usage Manager Agent can run in a non-HPCA environment, as detailed in HPCA Application Usage Manager Agent in a non-HPCA Environment, starting below.

HPCA Application Usage Manager Agent in a HPCA Environment

The collection point for the HPCA Application Usage Manager Agent to use an SSL-secured Integration Server in a HPCA environment is:

```
https://xxx.xxx.xxx.xxx:443/KB_Mgr1_Usage/
```

This can be set in the Admin CSDB Editor.

HPCA Application Usage Manager Agent in a non-HPCA Environment

1. Stop the `HP HPCA Application Usage Manager Agent` service.
2. In *`SystemDrive`*`:\ProgramFiles\Hewlett-Packard\CM` create a new directory called `Agent`.

3. From the `Usage Manager\Agent Install\Setup\CACertificates` directory on the HPCA media, copy the `CACerificates` folder and paste it in `*SystemDrive*:\ProgramFiles\Hewlett-Packard\CM\Agent`.

4. From `IntegrationServer\etc\CACertificates\Server-*hostname*.netcert.pem`, copy the lines

   ```
   -----BEGIN CERTIFICATE-----
   ```

   Thru

   ```
   ------END CERTIFICATE------
   ```

   > If you are using the Portal, modify the path in step 4 as follows. Replace
   >
   > `IntegrationServer\etc\CACertificates`
   >
   > with
   >
   > `ManagementPortal\etc\CACertificates`.

5. On the HPCA agent machine, open the `cacert.pem` file that is in the `CACerificates` directory and, at the end of it, paste the lines that were copied in step 3.

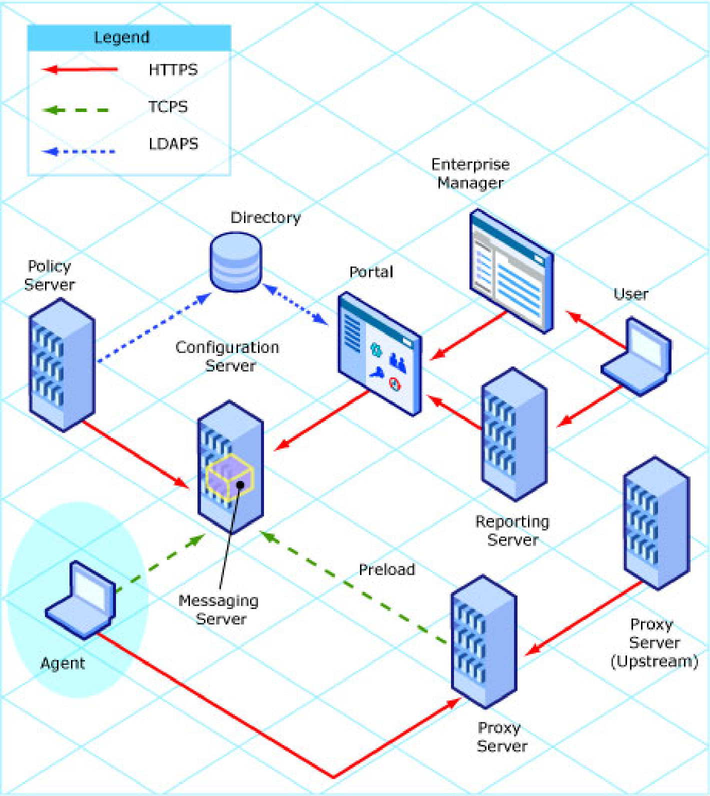
6. In the Registry, change the collection point to **`https://`***`xxx.xxx.xxx.xxx`***`:443/KB_Mgr1_Usage/`**.

7. Start the `HP HPCA Application Usage Manager Agent` service.

HPCA Agents

This section details how to enable secure communications on HPCA agents.

Figure 9 HPCA agents



Secure (SSL) communications are supported on the following HPCA agents.

- Application Manager
- Application Self-service Manager, see HPCA Application Self-service Manager Agent on page 76
- Inventory Manager
- Patch Manager

To enable SSL communications with a Configuration Server for these HPCA agents, pass **SSLMGR** and **SSLPORT** with the appropriate values on a **RADSKMAN** command line, as in:

```
Radskman sslmgr=host,sslport=443
```

HPCA Application Self-service Manager Agent

For the Application Self-service Manager, setup **sslmanager** and **sslport** tags in the ARGS.XML file, as in:

```
<SSLMANAGER>localhost</SSLMANAGER>
<SSLPORT>443</SSLPORT>
```

Summary

Secure SSL communications can be configured for the following HPCA products.

- HPCA Reporting Server via HTTPS
- HPCA Enterprise Manager via HTTPS
- HPCA Messaging Server via HTTPS
- HPCA Configuration Server via TCPS
- HPCA Distributed Configuration Server via HTTPS
- HPCA Patch Manager Server via TCPS
- HPCA Integration Server components:
  - HPCA Proxy Server via HTTPS
  - HPCA Policy Server via HTTPS and LDAPS
- HPCA Portal via HTTPS and LDAPS
- HPCA Application Usage Manager via HTTPS and LDAPS
- HPCA Agents

A Troubleshooting

⚠	If your environment uses Core and Satellite servers, first read the *Core and Satellite Servers Getting Started Guide* as the installation, configuration, and troubleshooting information in that guide may override the information in this guide.

IMPORTANT NOTE:

Before troubleshooting SSL using the information in this section, HP recommends always checking the HP documentation web site for the latest version of this document and the release-specific Release Notes.

To check for recent updates and to verify that you are using the most recent edition, go to http://ovweb.external.hp.com/lpe/doc_serv/.

- In the **Product** list, click the product name.
- In the **Version** list, click the version number.
- In the **OS** list, click the OS type.
- In the **Document** field, click the document title.
- To retrieve the document, click **Open** or **Download**.

Certificate Authorities

The file, `cacert.pem`, contains the CA root certificate (the public key) for the following Certificate Authorities: Entrust, VeriSign, Inc. and G.E. If you are not using one of these CAs, the CA root certificate must be obtained using one of the following methods.

HPCA Agents

- Obtain the certificate from your CA and substitute it for `cacert.pem` in the `CACertificates` sub-directory of the HPCA agent `IDMSYS` location.
- Use HPCA agent self-maintenance to download the certificate to the HPCA agent.

HPCA Portal (HTTPS and LDAPS)

- Obtain the certificate from your CA and substitute it for `cacert.pem` in the `/etc/CACertificates` sub-directory of the directory in which the Portal is installed. If multiple CA root certificates are required, the contents of the public keys can be added at the beginning of the `cacert.pem` file.

Existing Certificate or Private Key

If the Certificate Generation Utility program is run on a HPCA server that already houses a version of the Certificate Generation Utility, the following message might appear.

> "A certificate or private key already exists for the specified server name. Choose another server name."

Do either of the following:

- In the Review and Password window, change the name in the text box Server to Generate For and try again. (This generates a new server certificate request for the server that is identified in this text box.)

  or

- Cancel the installation (since a server certificate request and private key already exist for this server).

SSL Port is Not Enabled

- Verify that the correct port is specified.
- Be sure that the signed certificate is set. If not, the following message will appear in the `httpd-PORT.log` on the Integration Server.

  ```
  20050621 21:49:11 Warning: TLS startup failed: Certificate
  "D:\Program Files\Hewlett-Packard\CM\IntegrationServer\
  etc\Certificates\server.HP.comcert.pem" not found
  ```

- If the port is already in use by another application, the following message will appear in the `httpd-PORT.log` on the Integration Server.

  ```
  20050621 22:10:08 Warning: TLS startup failed: LAVENEL1:443
  couldn't open socket: address already in use
  ```

Expired Certificates

If one or more of your certificates has expired, you will be unable to create an SSL connection between the Portal and the Enterprise Manager. If you discover that you have an expired certificate, follow these steps:

1. On the Portal system, use the current version of the Certificate Generation Utility to create new certificates.
2. On the Enterprise Manager system, use the current version of the Certificate Generation Utility to create new keystore and truststore files.

Host Name Mismatch

The host name that you use to establish an SSL connection to a particular server must match the host name used to create the certificates on that server. The form of the host name, simple or fully qualified, must also match.

For example, if you use the Certificate Generation Utility to create certificates using this command:

```
cert_mgr create signed -hostname cmserver1.mycorp.com
```

You must use the following URL to create the SSL connection:

```
https://cmserver1.mycorp.com:SSLport/AppName
```

where *SSLport* is the SSL port configured on `cmserver1.mycorp.com`, and *AppName* is the specific server application (such as `em` or `reporting`) that you want to access.

The following URLs will *not* create an SSL connection in this case:

```
https://cmserver1:SSLport/AppName
```

```
https://localhost:SSLport/AppName
```

```
http://cmserver1.mycorp.com:SSLport/AppName
```

In the case of the Portal to Enterprise Manager connection, the information used to build this URL is specified in the following file:

```
EMInstallDir\CM-EC\tomcat\webapps\em\WEB-INF\console.properties
```

For example:

```
protocol=https\://

port=443

hostname=cmserver1.mycorp.com
```

Browser Stops Loading Directories

Several problems can cause your browser to stop loading directories when it reaches 80%. If this happens, check the following things.

- Your certificates are valid and have not expired.
- Your keystore or truststore files are valid.

  These files are replaced when you upgrade to version 5.1*x* from an earlier version of HP Client Automation.
- The Certificate Authority that is specified in the `cacert.pem` file is valid and has not expired.

If you suspect a problem with any of these files, regenerate the certificates using the 5.1*x* version of the `cert_mgr` utility (see The Certificate Generation Utility starting on page 24). Replace the certificates on Portal system, and replace the keystore and truststore files on the Enterprise Manager system.

The default keystore/truststore password is stored in the Windows system registry. If you are migrating from HPCA version 5.00 to version 5.1*x* make sure that the default password matches the password that was used to create the keystore and truststore files.

The default password is stored in:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Apache Software Foundation\ Procrun
2.0\HPCMEnterpriseManager\Parameters\Java
```

…in the `Options` parameter,

```
-Djavax.net.ssl.trustStorePassword
```

If you change the default password, be sure to restart the `HPCA Enterprise Manager` service.

HPCA Enterprise Manager Cannot Communicate with HPCA Portal

The following problems can prevent the Enterprise Manager from connecting to the Portal using SSL.

- *The certificate or private key files on the two systems do not correspond with one another.*

  There is no easy way to determine whether this is the problem. If you suspect that it is, regenerate your certificates.

- *There is an incorrect path to the private key or certificates.*

  Check the following settings in the `httpd-managementportal.cfg` file on the Portal system.

Table 5 HPCA Portal Config File Settings

Variable	Value
SSL_CERTFILE	C:/Program Files/Hewlett-Packard/CM/ManagementPortal/etc/CACertificates/*servername*-cert.pem
SSL_KEYFILE	C:/Program Files/Hewlett-Packard/CM/ManagementPortal/etc/CACertificates/*servername*-prvkey.pem
SSL_CADIR	/etc/CACertificates
SSL_CAFILE	/etc/CACertificates/cacert.pem

You can update `httpd-managementportal.cfg` with a text editor.

The paths shown in Table 5 above assume that the default installation directory was used for the Portal server. Be sure that the values of `SSL_CERTFILE` and `SSL_KEYFILE` reflect the actual installation directory in your environment.

- *The host name in the certificate is incorrect.*

  Use the `keytool` that was deployed with your Java Run-time Environment (JRE) to print the name in the certificate. The Owner's CN field should match the name of the machine, including the domain, as shown below.

```
C:\Program Files\HP OpenView\nonOV\jre\b\bin>keytool -
printcert -v -file demo-cert.pem


Owner: CN=demo.mydomain.com, OU=IT, O=MYCO, L=FTC, ST=CO, C=US

Issuer: CN=demo.mydomain.com, OU=IT, O=MYCO, L=FTC, ST=CO,
C=US

Serial number: 0

Valid from: Wed Oct 24 08:55:08 EDT 2007 until: Sat Mar 10
07:55:08 EST 2035

Certificate fingerprints:
     MD5:  FB:0F:64:C9:E2:37:63:1D:AE:62:87:85:5E:E9:F8:64
     SHA1:
51:2A:6B:1D:42:B4:E3:AD:6E:C2:C7:CE:91:DE:84:0F:C1:11:5C:F4
```

You can determine the machine and domain name using the command `ipconfig -all`.

If you determine that the host name in the certificate does not match the machine and domain name, regenerate your certificate with the correct host name.

- *In the* `httpd-managementportal.cfg` *file, the CAFILE or CADIR field is not set correctly.*

  See Table 5 on page 83 for correct settings.

Notify Fails in HPCA Enterprise Manager after Configuring SSL

When you enable SSL communications on the Enterprise Manager and disable the standard HTTP port (8080), you must also change the `opeurl` parameter in *`EMInstallDir`*`\CM-EC\tomcat\webapps\em\WEB-INF\console.properties`.

The `opeurl` parameter specifies the address that is to be used for communications between the Enterprise Manager and the **Operational Process Engine** (OPE), which runs in the same Tomcat instance.

- The protocol for SSL in the `opeurl` should be changed from `http` to **`https`**.
- The port for the SSL in the `opeurl` should match the SSL port that is configured for the Enterprise Manager server, as in:

  ```
  opeurl=https\://localhost\:8443/ope/resources
  ```

See Disabling Non-SSL Access on page 49 for detailed instructions.

Communication to Job Process Engine not Encrypted

Communication between the Enterprise Manager and the job process engine that executes the Notify commands is not encrypted. This is because the default setting for `opeurl` in the `Console.properties` file—`http\://localhost\:8080/ope/resources`—is not an encrypted channel.

To create a work-around, modify:

EMInstallDir/CM-EC/tomcat/webapps/em/WEB-INF/Console.properties

Set the `opeurl` property to

https\://localhost\:*<port>***/ope/resources**.

In this case, *<port>* is the SSL port that is configured for the Enterprise Manager server, such as:

```
opeurl=https\://localhost\:8443/ope/resources
```

See the previous section, Notify Fails in HPCA Enterprise Manager after Configuring SSL (on page 84), for more information.

B Command Line Options for the Certificate Generation Utility

Table 6 below and Table 7 on page 88 describe the options that can be used with the **cert_mgr create** and **cert_mgr import** commands that are described in Chapter 2, Setting up Certificates for SSL.

Table 6 Options for cert_mgr create

Option	Description	Default
-hostname	Host name of the server for which you will create the certificates.	Simple host name of the system on which you are running **cert_mgr**.
-trustpass	The password for the truststore.	changeit
-rndbytes	Size of the random bytes when creating the random file that will be used to create the private key for the server certificate.	2048 bytes
-keysize	Size of the server's private key in bits.	1024 bits
-keypass	The password for the server's certificate when it is added to the keystore.	secret
-days	The number of days the server's certificate will be valid.	9999 days
-carndbytes	Same as **rndbytes**, but for the CA.	2048 bytes
-cakeysize	Same as **keysize**, but for the CA.	1024 bytes
-cadays	Same as **days**, but for the CA.	9999 days

Table 7 Options for cert_mgr import

Option	Description	Default
`-hostname`	Host name of the server for which you will create the certificates.	Simple host name of the system on which you are running `cert_mgr`.
`-signedcert`	The fully qualified path and file name of the signed certificate that was returned by the CA.	
`-signercert`	The fully qualified path and file name to the certificate of the signing CA. Used when importing a certificate via the Certificate Generation Utility.	

Index