Sterling Secure Proxy

# Certificates Guide

*Version 34*

Sterling Secure Proxy

# Certificates Guide

*Version 34*

**Note**
Before using this information and the product it supports, read the information in "Notices" on page 41.

This edition applies to version 3.4 of IBM Sterling Secure Proxy and to all subsequent releases and modifications until otherwise indicated in new editions.

# Contents


## Chapter 1. SSL-TLS Certificates . . . . 1

About SSL/TLS Certificates . . . . . 1
Certificate Implementation Models Using Sterling Secure Proxy . . . . . 2
- Implement Certificates that Use a Common Certificate Authority . . . . . 2
- Implement Self-Signed Certificates . . . . . 2
- Implement Self-Signed Certificates with Different Certificates for Inbound and Outbound Connections . . . . . 3
- Configure a Secure Connection to Sterling External Authentication Server Server . . . . . 4
- Use Multiple Key Stores in Sterling Secure Proxy 6

Import a Public Certificate into a Trusted Certificate Store . . . . . 7
Import Private Keys into a System Certificate Store . 7
Create a New Trusted Certificate Store. . . . . 8
Create a New System Certificate Store . . . . . 8

## Chapter 2. Store System Certificates on an HSM . . . . . 9

Store System Certificates on a Hardware Security Module (HSM) . . . . . 9
Enable the HSM Environment . . . . . 9
Disable the HSM Environment . . . . . 11
Manage Key Certificates . . . . . 11
- Create Self-Signed Certificates . . . . . 11
- Import a Certificate. . . . . 13
- Export a Certificate . . . . . 14
- Obtain a Certificate from the HSM Device . . . 15
- Store a Certificate on the HSM Device . . . . 17
- Copy a Certificate . . . . . 18
- Delete a Certificate . . . . . 19
- List Key Certificates on the HSM Device. . . . 20
- Load References to Keys on the HSM into the Sterling Secure Proxy System Certificate Store . . 21
- Update the HSM Password for HSM Key Certificates Stored in the Sterling Secure Proxy System Store . . . . . 23

Manage CSRs. . . . . 23
- Create a CSR . . . . . 24
- Update a CSR . . . . . 25
- Delete a CSR . . . . . 26
- List CSRs on the CM Store . . . . . 27
- Retrieve a CSR to Send to a Certification Authority . . . . . 27
- Retrieve the CA-signed Certificate . . . . . 28

## Chapter 3. Manage CM Certificates . . 29

Manage Certificates Between Sterling Secure Proxy Components . . . . . 29
- Use a Common Certificate for the Engine and CM . . . . . 29
- Use Different Certificates for the Engine and CM 32
- Restore the Factory Certificate on UNIX or Linux 34
- Restore the Factory Certificate on Microsoft Windows . . . . . 35

Change the Password of the CM Key Store and Trust Store on UNIX or Linux . . . . . 35
Change the Password of the CM Key Store and Trust Store on Microsoft Windows. . . . . 36
Change the Password of the Engine Key Store and Trust Store on UNIX or Linux . . . . . 36
Change the Password of the Engine Key Store and Trust Store on Microsoft Windows. . . . . 37
Configuration Utilities. . . . . 37

## Notices . . . . . 41

# Chapter 1. SSL-TLS Certificates

## About SSL/TLS Certificates

Certificates are used in secure communications to encrypt and decrypt data. You create certificates using certificate creation software such as Sterling Certificate Wizard. Each certificate is made up of two components: the public key and the private key. Always keep your private key secret.

As an added measure of security, you can obtain your certificate from a certificate authority (CA). A CA verifies all of the identity information in your certificate, then adds its signature. In an SSL or TLS transaction, your certificate is presented to your trading partner, who can recognize the signature of the CA using the CA root certificate. This assures your trading partner that you are who you say you are. There are many free and commercial certificate authorities. Some companies use an internal certificate authority.

If you use a certificate that is not validated by a CA, it is called a self-signed certificate. Self-signed certificates are used when identity verification is not required, such as internal communications or product testing.

To implement SSL or TLS over FTP or HTTP when using a CA, you need to acquire the CA root certificate from the trading partner, and you must make it available to Sterling Secure Proxy. You must also make your private key and certificate available to Sterling Secure Proxy.

To implement SSL or TLS over FTP or HTTP using self-signed certificates, provide your certificates to your trading partner. Also, acquire your trading partner certificates and make them available to Sterling Secure Proxy. You also make the private key available to Sterling Secure Proxy.

Public certificates and CA root certificates must be in base 64 or DER format. Private keys, accompanied by their matching public certificates, must be contained in a base 64 key certificate or a PKCS12 file.

### Certificate Implementation Models Using Sterling Secure Proxy

The following sections topics present several models for using certificates and shows how to implement the model in Sterling Secure Proxy.

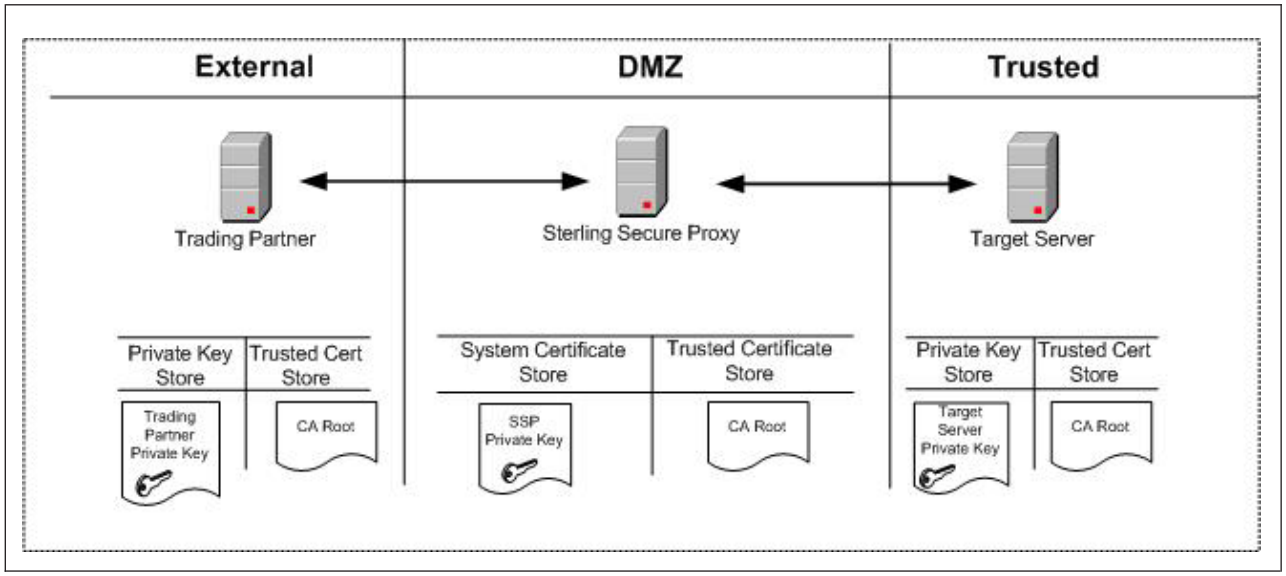
- Implement Certificates that Use a Common Certificate Authority
- Implement Self-Signed Certificates
- Implement Self-Signed Certificates with Different Certificates for Inbound and Outbound Connections
- Configure a Secure Connection to Sterling External Authentication Server Server
- Use Multiple Key Stores in Sterling Secure Proxy

# Certificate Implementation Models Using Sterling Secure Proxy

## Implement Certificates that Use a Common Certificate Authority

### About this task

In this scenario Sterling Secure Proxy, the target server, and the trading partner use the same certificate authority (CA). The certificate distribution looks like this:




Sterling Secure Proxy has its private key and the root certificate from the CA. The trading partner has its private key and the root certificate from the CA. The target server has its private key and the root certificate from the CA.

Use the following procedure to implement this model in Sterling Secure Proxy:

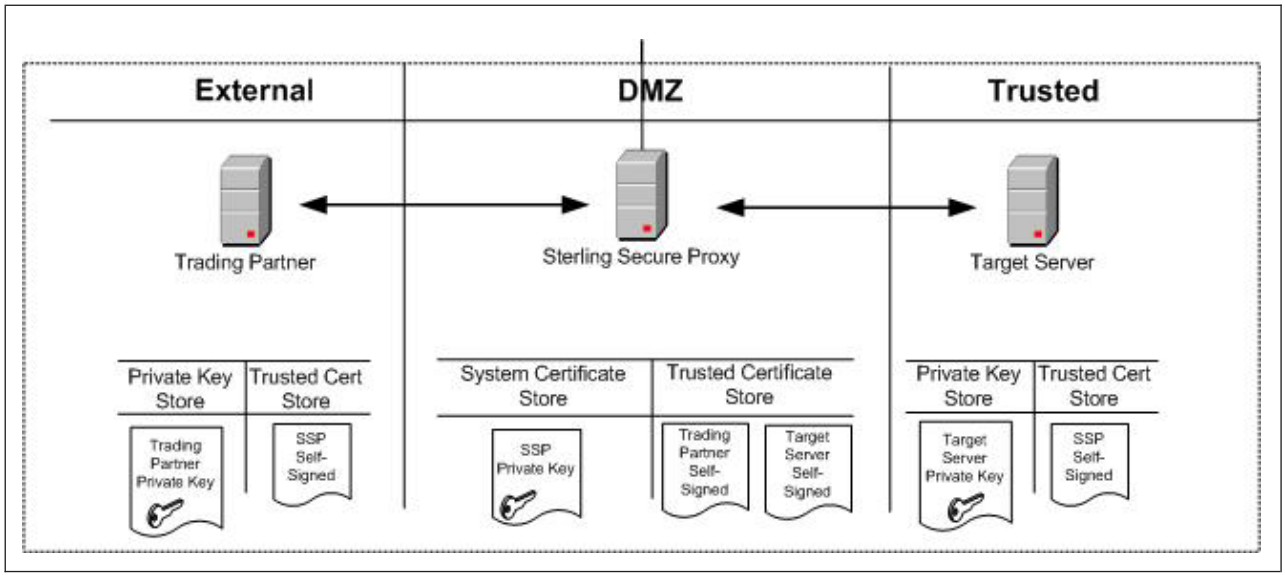
### Procedure

1. Acquire the root certificate from the common CA.
2. Import the CA root certificate into the default store. Refer to *Import a Public Certificate into a Trusted Certificate Store.*
3. Import the Sterling Secure Proxy private key into the default system certificate store. Refer to *Import Private Keys into a System Certificate Store*.

## Implement Self-Signed Certificates

### About this task

In this scenario, there are no CA certificates. Self-signed certificates are used by all entities. The certificate distribution looks like this:




Sterling Secure Proxy has its private key and the self-signed certificates from the trading partner and the target server. The trading partner has its private key and the self-signed certificate of Sterling Secure Proxy. The target server has its private key and the self-signed certificate of Sterling Secure Proxy.

Use the following procedure to implement this model:

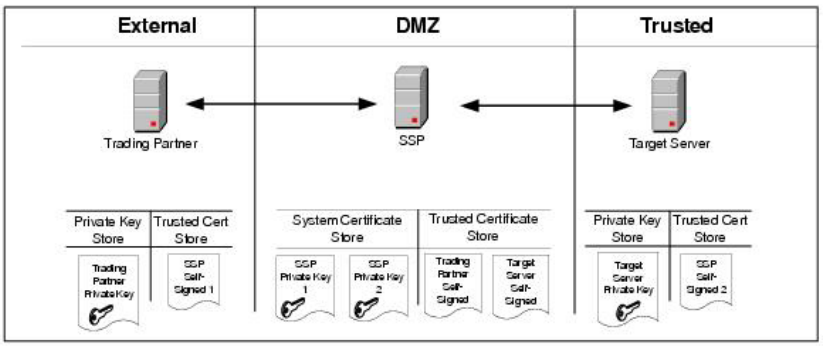
### Procedure

1. Provide the Sterling Secure Proxy self-signed certificate to your trading partner and your target server.
2. Acquire the self-signed certificates from the trading partner and target server.
3. Import the trading partner and target server self-signed certificates into the default Store. Refer to *Import a Public Certificate into a Trusted Certificate Store*.
4. Import the Sterling Secure Proxy private key into the default System Certificate Store. Refer to *Import Private Keys into a System Certificate Store*.

## Implement Self-Signed Certificates with Different Certificates for Inbound and Outbound Connections

### About this task

In this scenario, there are no CA certificates. Separate self-signed certificates are used for the inbound and outbound connections. The certificate distribution looks like this:




Sterling Secure Proxy has two private keys and the self-signed certificates from the trading partner and the target server. The trading partner has its private key and one self-signed certificate from Sterling Secure Proxy. The target server has its private key and the other self-signed certificate from Sterling Secure Proxy.

Use the following procedure to implement this model:

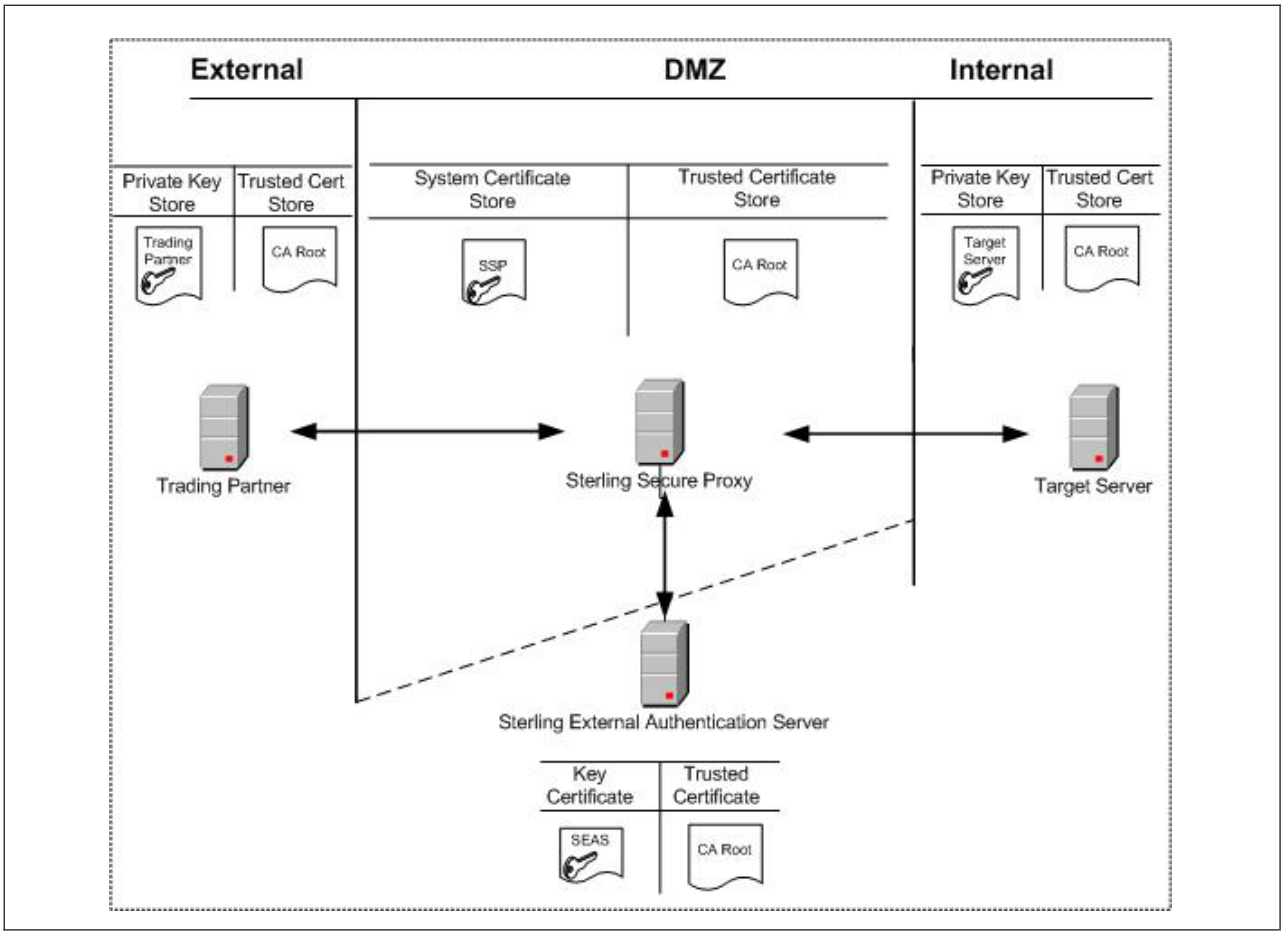
### Procedure

1. Provide the Sterling Secure Proxy self-signed certificate to your trading partner and your target server.
2. Acquire the self-signed certificates from the trading partner and target server.
3. Import the trading partner and target server self-signed certificates into the default Trusted Certificate Store. Refer to *Import a Public Certificate into a Trusted Certificate Store*.
4. Import the Sterling Secure Proxy private keys into the default System Certificate Store. Refer to *Import Private Keys into a System Certificate Store*.

## Configure a Secure Connection to Sterling External Authentication Server Server

### About this task

You can configure a secure connection between Sterling Secure Proxy and Sterling External Authentication Server Server as shown in the following diagram:




In this scenario, Sterling Secure Proxy has the private key in the system certificate store and the CA root certificate in the trusted certificate store. The trading partner has a private key and the CA root certificate. The target server has a private key and the CA root certificate. Sterling External Authentication Server has a private key in its own key certificate store and the CA root certificate. Use the following procedure to implement this model.

This example shows the Sterling External Authentication Server implementation with a single certificate. You can also use a multiple Sterling Secure Proxy certificates model.

Use the following procedure to implement this model:

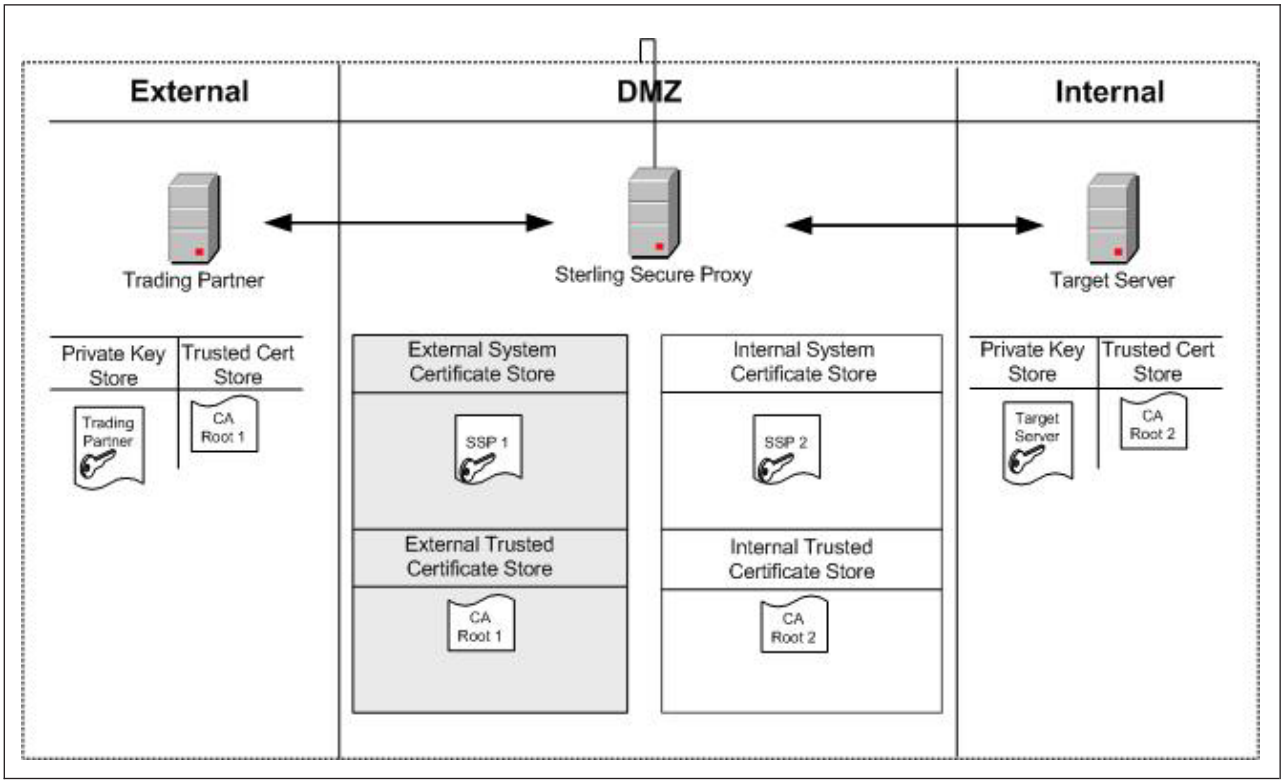
## Procedure

1. Acquire the root certificate from the common CA.
2. Import the CA root certificate into the default Trusted Certificate Store. Refer to *Import a Public Certificate into a Trusted Certificate Store*.
3. Import the Sterling Secure Proxy private key into the default System Certificate Store. Refer to *Import Private Keys into a System Certificate Store*.
4. Configure certificates for use by Sterling External Authentication Server. Refer to the Sterling External Authentication Server documentation library for instructions.

# Use Multiple Key Stores in Sterling Secure Proxy

## About this task

Sterling Secure Proxy gives you the option of having multiple key stores or trust stores. This is useful if you do not want all your keys in a single location. Also, if you are running multiple Sterling Secure Proxy engines, it may be better to have a separate system certificate store or trusted certificate store for each engine. The following diagram shows a very basic model using multiple key stores:




In this scenario, Sterling Secure Proxy has two key certificates: SSP1 in the external system certificate store and SSP2 in the internal system certificate store. Different CAs are used for internal and external communications. The CA root certificate for external communication (CA Root 1) is in the external trusted certificate store. The CA root certificate for internal communication (CA Root 2) is in the internal trusted certificate store. The trading partner has its own private key and the CA Root 1 certificate. The target server has its own private key and the CA Root 2 certificate. Use the following procedure to implement this model:

## Procedure

1. Acquire the external CA root certificate.
2. Acquire the internal CA root certificate.
3. Create a new trusted certificate store for your external communications (External Store in diagram above). Refer to *Create a New Trusted Certificate Store*.
4. Import the external CA root certificate into the External Store. Refer to *Import a Public Certificate into a Trusted Certificate Store*.

5. Create a new trusted certificate store for your internal communications (Internal Store in diagram above). Refer to *Create a New Trusted Certificate Store*.
6. Import the internal CA root certificate into the internal trusted certificate store. Refer to *Import a Public Certificate into a Trusted Certificate*.
7. Create a new system certificate store for external communications (External System Certificate Store in diagram above). Refer to *Create a New System Certificate Store*.
8. Import the SSP1 private key into the new external system certificate store. Refer to *Import Private Keys into a System Certificate Store*.
9. Create a new system certificate store for internal communications (Internal System Certificate Store in diagram above). Refer to *Create a New System Certificate Store*.
10. Import the SSP2 private key into the new internal system certificate store. Refer to *Import Private Keys into a System Certificate Store*.

## Import a Public Certificate into a Trusted Certificate Store

### About this task

Use the following procedure to import the public certificate from your trading partner, target server, or CA into a trusted certificate store:

### Procedure

1. Click **Credentials** from the menu bar.
2. Expand the **Certificate Stores** tree and then the **Trusted Certificates Stores** tree.
3. Select a trust store. The default trust store is dfltTrustStore.
4. Click **New**.
5. Specify a name in the **Trusted Certificate Name** field.
6. Click **Browse** to select the certificate to import.
7. Double-click the certificate to select.
8. Click **OK**.

## Import Private Keys into a System Certificate Store

### About this task

Use the following procedure to import an Sterling Secure Proxy private key into a system certificate store:

### Procedure

1. Click **Credentials** from the menu bar.
2. Expand the **Certificate Stores** tree and then the **System Certificate Stores** tree.
3. Select a key store. The default key store is dfltKeyStore.
4. Click **New**.
5. Specify values for the following:
   - System Certificate Name
   - Password (passphrase associated with the system certificate)
   - Confirm Password
6. Click **Browse** and select the certificate to import.
7. Click **OK**.

## Create a New Trusted Certificate Store

### About this task

Use the following procedure to create a new trusted certificate store:

### Procedure

1. Click **Credentials** from the menu bar.
2. Click **Actions** > **New Certificate Store** > **Trusted Certificate Store**.
3. Specify a name for the certificate store in the **Trusted Certificate Store Name** field.
4. Click **Save**.

   Refer to *Import a Public Certificate into a Trusted Certificate Store* to add certificates.

## Create a New System Certificate Store

### About this task

To create a new system certificate store:

### Procedure

1. Click **Credentials** from the menu bar.
2. Click **Actions** > **New Certificate Store** > **System Certificate Store**.
3. Specify a name for the certificate store in the **System Certificate Store Name** field.
4. Click **Save**. Refer to *Import Private Keys into a System Certificate Store* to add certificates to the certificate store.

# Chapter 2. Store System Certificates on an HSM

## Store System Certificates on a Hardware Security Module (HSM)

A Hardware Security Module (HSM) is a hardware-based security device that generates, stores, and protects cryptographic keys. Sterling Secure Proxy uses keys and certificates stored in its store or on an HSM. Sterling Secure Proxy maintains information in its store about all keys and certificates.

To access keys in an HSM device, a reference to the keys and the passphrase protecting the key must be added to Sterling Secure Proxy. This reference is secure and cannot be used by an intruder to access the certificate information. You can configure keys on the HSM at CM, using command line scripts described in this chapter.

For more security, create the keys on the HSM device and store the HSM private keys on the device. To import externally-created keys into the HSM, first import the external keys into the HSM and then destroy the files containing the external private key.

HSMs implement the Java JCE API. This interface accesses the keys in the device. The JCE implementations for Safenet and Thales have the following differences:

- Safenet uses slots, logical entities defined through the Safenet administration utility. Designate a slot for Sterling Secure Proxy and assign a user PIN. Configure Sterling Secure Proxy and identify the slot to use. Only one slot can be used by Sterling Secure Proxy.
- Safenet uses a single keystore for all keys in a slot. The user PIN protects all the keys in the slot. Each key within a slot must have a unique alias.
- Thales uses a security world that contains one or more HSM modules. The modules can reside on the same or different machines. The keys in the security world are protected by an operator smart card. Create an operator smart card set for Sterling Secure Proxy, identify “1 of N” for the cards, and assign a passphrase to each card. Before Sterling Secure Proxy can start, insert the operator smart card protecting the Sterling Secure Proxy keys into the card reader.
- Thales supports multiple keystores. Each keystore can contain multiple keys, but Sterling Secure Proxy only stores one key per keystore. With Thales, multiple keys can have the same alias. For example, on Sterling B2B Integrator, all keys on an Thales HSM have the alias Key. Each keystore has a unique instance ID defined as a 40-character hexadecimal string. The combination of the instance ID and the key alias makes each key unique.

## Enable the HSM Environment

### About this task

Use the setupHSM command to enable or disable the HSM environment. Run this command on the engine. If you are using a netHSM module and CM has access to the netHSM, you can also run the command on CM. Running the command on CM allows you to configure the HSM keys without requiring a running engine. However, you must stop CM.

Stop the engine or CM before you run this command. Additionally, you must have permission to write files to the Sterling Secure Proxy installation directory. If you reinstall the HSM support software, run the setupHSM -enable command again to make sure that any updated jar files and libraries are copied to the installation directory.

Use the setupHSM -enable command to copy files from the HSM hardware to Sterling Secure Proxy, copy the HSM security providers in the right order, update the security.properties file with the appropriate Certicom TLS security string for the HSM you are using, and add any environment variables to the startup scripts.

To set up the HSM environment for Microsoft Windows, type the following command:

```
setupHSM -enable [parameters]
```

To set up the HSM environment for UNIX or Linux, type the following command:

```
setupHSM.sh -enable [parameters]
```

Following is a description of the enable parameters:

| Parameter | Description |
|---|---|
| hsm | HSM type. Required if you are enabling the HSM. Valid values = nCipher \| Eracom. |
| slot | Slot number assigned to Sterling Secure Proxy. The optional parameter is valid for Safenet only. Default=0. |
| path | Path to the root directory of the HSM runtime support software. Required.<br><br>If the path contains embedded spaces, enclose the whole parameter in double-quotes. For example, “path=C:\Program Files\Safenet”.<br><br>On UNIX, the value is normally /opt/nfast for Thales and /opt/Safenet for Safenet.<br><br>On Microsoft Windows, the value is normally C:\nfast for Thales and C:\Program Files\Safenet for Safenet. |
| netserver | Host name or IP address of the netHSM server. Optional.<br><br>Valid for Safenet on UNIX. It is ignored on Microsoft Windows. |
| systempass | Engine system passphrase or CM passphrase, depending upon where the command is run. Optional. If you do not configure this parameter, the user is prompted for the passphrase. |

Following is a sample script to setup an Safenet HSM on UNIX:

```
setupHSM.sh -enable hsm=eracom slot=1 path=/opt/Eracom
```

Following is a sample script to setup an Thales HSM on UNIX:

```
setupHSM.sh -enable hsm=nCipher path=/opt/nfast
```

# Disable the HSM Environment

## About this task

Use the setupHSM command to enable or disable the HSM environment. Run this command on the engine. If you are using a netHSM module and CM has access to the netHSM, you can also run the command on CM. Running the command on CM allows you to configure the HSM keys without requiring a running engine. However, you must stop CM.

Stop the engine or CM before you run this command. Additionally, you must have permission to write files to the Sterling Secure Proxy installation directory. If you reinstall the HSM support software, run the setupHSM -enable command again to make sure that any updated jar files and libraries are copied to the installation directory.

Use setupHSM -disable to delete the HSM provider files, remove the HSM security providers, restore the default Certicom TLS security provider definitions, and remove the HSM environment variables from the startup scripts.

To disable the HSM environment, type the following command:

Following is a description of the HSM setup disable parameter:

```
setupHSM -disable systempass
```

| Parameter | Description |
|---|---|
| systempass | Engine system passphrase or CM passphrase, depending upon where the command is run. Optional. If you do not define this parameter, you are prompted for the passphrase. |

# Manage Key Certificates

## Create Self-Signed Certificates

### About this task

Use the manageKeyCerts -create command to create a self-signed key certificate. Stop CM before you run this command.

Consider the following before you use this command:

- If the engine parameter is defined, a certificate is created on the HSM configured for that engine. If a netHSM is used and multiple engines access the netHSM, any of the engines can be specified to create the certificate on the HSM.
- If the engine uses a PCI module and it cannot be accessed by other engines, group the key certificates for that engine in a separate system certificate store. Those key certificates cannot be shared with other engines.
- If the engine parameter is not defined, and HSM support is enabled on CM, the key certificate is created on the HSM configured for CM. Make sure that engines that use this key certificate can access the HSM enabled for CM.
- If the engine parameter is not defined and HSM support is not enabled on CM, the key certificate is created on the Sterling Secure Proxy system certificate keystore.

To create a self-signed key certificate, type the following command:

```
manageKeyCerts -create [parameters]
```

Following are the parameters used to create a key certificate:

| Parameter | Description |
|---|---|
| certName | Name of the key certificate on Sterling Secure Proxy. Required. |
| certStore | Name of the system certificate store where the key certificate will be stored. This field is optional. If the store does not exist, it is created. Default=dfltKeyStore. |
| engine | Name of the engine with access to the HSM. Optional. |
| alias | Alias for the key certificate on the HSM. Optional.<br><br>If no value is defined, the alias defaults to certificate name. |
| keySize | Key size of the file to create. Valid values = 1024 \| 2048 \| 4096. Default=1024. |
| CN | Certificate common name. Required.<br><br>If the name contains spaces, enclose the command and string in double quotes, for example "CN=my name". |
| email | E-mail address. Optional. |
| O | Organization. Optional. If the value contains spaces, enclose the command in double quotes, for example, "O=my org". |
| OU | Organization unit. Optional. Repeat this parameter to specify more than one organization unit. If the value contains spaces, enclose the command in double quotes, for example, "OU=my unit". |
| L | Location (city). Optional.<br><br>If the value contains spaces, enclose the command in double quotes, for example, "L=my location". |
| ST | State. Optional.<br><br>If the value contains spaces, enclose the command in double quotes, for example, "ST=my state". |
| C | Two letter country code. Optional. |
| daysValid | How many days the key certificate is valid. Optional. Default=365. |
| serial | Serial number for the key certificate. Optional. Default=1. |
| certSignBit | Whether to set the certificate signing bit on in the key usage flags. Valid values = n \| y\| false \| true. Default=n. |
| replace | Whether to replace a key certificate if a certificate with the same name already exists in the Sterling Secure Proxy system certificate store. Optional. Valid values = false \| true. Default=false. |
| systempass | Passphrase for CM. |
| adminid | Administrator ID. Optional. Prompts if not defined. |

| Parameter | Description |
|---|---|
| adminpass | Administrator password. Optional. Prompts if not defined. |
| keystorepass | Keystore password. Optional. Prompts if not defined.<br><br>For Safenet, the user PIN for the slot used by Sterling Secure Proxy.<br><br>For Thales, the passphrase for the operator smart card that will be used to protect the key. The card must be inserted in the module's card reader. |
| keypass | Passphrase for the key in the keystore. Optional. Prompts if not defined.<br><br>For Safenet, this parameter can be anything and will be ignored.<br><br>For Thales, this must be the same value as the keystore password. |

# Import a Certificate

## About this task

Use the manageKeyCerts -import command to import a certificate into the Sterling Secure Proxy system certificate store and the HSM. Stop CM before you run this command.

Consider the following before you use it:

- If you define the engine parameter, the certificate is imported to the HSM configured for that engine. If a netHSM is used and multiple engines access the netHSM, any of the engines can be specified to handle the request. Configure HSM support on the engine.
- If the engine uses a PCI module and that module cannot be accessed by other engines, group the key certificates for that engine in a separate system certificate store. You cannot share the key certificates on that system certificate store with other engines.
- If you do not define the engine parameter, and HSM support is enabled on CM, the key certificate is imported on the HSM configured for CM. Make sure that engines that use this key certificate can access the HSM enabled for CM.
- If you do not define the engine parameter and HSM support is not enabled on CM, the key certificate is imported to the Sterling Secure Proxy system certificate store only.

To import a key certificate into the Sterling Secure Proxy system certificate store, type the following command:

```
manageKeyCerts -import [parameters]
```

Following is a description of the import parameters:

| Parameter | Description |
|---|---|
| certName | Name of the key certificate on Sterling Secure Proxy. Required. |

| Parameter | Description |
|---|---|
| certStore | Name of the system certificate store where the key certificate will be stored. This field is optional. If the store does not exist, it is created. Default=dfltKeyStore. |
| engine | Name of the engine with access to the HSM. Optional. |
| alias | Alias for the key certificate on the HSM. Optional.<br><br>If no value is defined, the alias defaults to certificate name. |
| file | Fully-qualified path of the key certificate file to import. Required.<br><br>The file must be PEM or PKCS12. The script looks for BEGIN/END PEM markers in the file. If they are not found, the file is assumed to be PKCS12 format. |
| replace | Whether to replace a system certificate if a certificate with the same name already exists in the Sterling Secure Proxy system certificate store. Optional. Valid values = n \| y \| false \| true. Default=n. |
| systempass | CM system passphrase. |
| adminid | Administrator ID. Optional. Prompts if not defined. |
| adminpass | Administrator password. Optional. Prompts if not defined. |
| pemkeypass | Password for the PEM private key, if the file is PEM. Optional. Prompts if not defined. |
| pkcs12storepass | Password of the PKCS12 file, if the file is not PEM. Optional. Prompts if not defined. |
| pkcs12keypass | Passphrase for the key in the PKCS12 file, if the import file is not PEM. Optional. Prompts if not defined. |
| keystorepass | Keystore password. Optional. Prompts if not defined.<br><br>For Safenet, the user PIN for the slot used by Sterling Secure Proxy.<br><br>For Thales, the passphrase for the operator smart card that will be used to protect the key. The card must be inserted in the module's card reader. |
| keypass | Passphrase for the key on the keystore. Optional. Prompts if not defined.<br><br>For Safenet, this parameter is not used.<br><br>For Thales, define this parameter using the same value as the keystore password. |

# Export a Certificate

## About this task

Use the manageKeyCerts -export command to export a certificate from the system store or the HSM. CM can be running when you run this command.

Consider the following before you use this command:

- If you specify the engine parameter and the certificate is stored on the HSM, the certificate is exported from the HSM configured at the engine. You must enable the HSM on the engine. If a netHSM is used and multiple engines can access it, any of the engines can be specified to export the certificate.
- If you do not specify the engine parameter and the key certificate is stored in an HSM, the certificate is exported from the HSM configured for CM. You must enable the HSM on CM to export a certificate from it.
- If the certificate is not stored on an HSM, the engine parameter is ignored and the certificate is exported from the Sterling Secure Proxy system certificate store.
- For key certificates stored on the HSM, only the public certificate in PEM format will be exported. The private key cannot be exported.

To export a key certificate from the Sterling Secure Proxy system certificate store, type the following command:

```
manageKeyCerts -export [parameters]
```

Following is a description of the export parameters:

| Parameter | Description |
|---|---|
| certName | Name of the key certificate on Sterling Secure Proxy. Required. |
| certStore | Name of the system certificate store where the key certificate will be stored. This field is optional. Default=dfltKeyStore. |
| engine | Name of the engine with access to the HSM. Optional. |
| format | Format for the key certificate file. This parameter is required for non-HSM key certificates. Forced to pem for the HSM key certificates. Valid values = pem \| pkcs12. |
| file | Fully-qualified path of the file where the key certificate file will be stored.<br><br>Required. |
| systempass | CM system passphrase. |
| adminid | Administrator ID. Optional. Prompts if not defined. |
| adminpass | Administrator password. Optional. Prompts if not defined. |
| pkcs12storepass | Password of the PKCS12 file, if the format is PKCS12 and the key certificate is not stored on an HSM. Optional. Prompts if not defined. |
| pkcs12keypass | Passphrase for the key in the PKCS12 file, if the format is PKCS12 and the key certificate is not stored on an HSM. Optional. Prompts if not defined. |
| pemkeypass | Passphrase to encrypt the private key if the format is PEM. |

## Obtain a Certificate from the HSM Device

### About this task

Use the manageKeyCerts -getFromHSM command to extract a reference to a key in the HSM and add the entry into the Sterling Secure Proxy keystore. Stop CM before you run this command.

Consider the following before you use this command:

- If you define the engine parameter, certificate information is obtained from the HSM at the engine. You must enable the HSM on the engine.
- If you configure netHSM, and multiple engines access the netHSM, any of the engines can be specified in the command.
- If you do not specify the engine parameter, the key certificate is obtained from the HSM configured at CM. You must enable the HSM on CM to obtain information from the HSM at CM.
- For the Thales HSM, the keystore blob for the key (Key Instance, as displayed by KeySafe) must be provided in the keyStoreData parameter. Obtain this 40-character hexadecimal string by running the -listHSM command.
- After a reference to an HSM key certificate is successfully obtained, the HSM key cannot be obtained again under a different Sterling Secure Proxy system certificate name. This action results in an error.

To obtain a key certificate from the HSM, type the following command:

```
manageKeyCerts -getFromHsm [parameters]
```

Following is a description of the getFromHSM parameters:

| Parameter | Description |
|---|---|
| certStore | Name of the system certificate store where the key certificate will be stored. This field is optional. If the store does not exist, it is created. Default=dfltKeyStore. |
| engine | Name of the engine with access to the HSM. Optional. |
| certName | Name of the key certificate on Sterling Secure Proxy. Required. |
| alias | Alias for the key certificate on the HSM. Required. |
| keyStoreData | HSM keystore blob string. Required with the Thales HSM. This is a 40-character hex string, displayed as Key Instance by the Thales KeySafe utility. If not provided and the HSM key certificate already exists in the system certificate store, the current keystore blob is used to pull the key back into the CM store. Use the -listHsm command to get the blobs for key certificates in the HSM. Alternatively, the blob string can be written to a file. Specify that file name in the keyStoreFile parameter. |
| keyStoreFile | File containing the HSM keystore blob string. If defined, this parameter overrides the keyStoreData parameter. |
| replace | Whether to replace a key certificate if a certificate with the same name already exists in the Sterling Secure Proxy system certificate store. Optional. Valid values = n \| y \| false \| true. Default=n. |
| systempass | CM system passphrase. |
| adminid | Administrator ID. Optional. Prompts if not defined. |
| adminpass | Administrator password. Optional. Prompts if not defined. |

| Parameter | Description |
|---|---|
| keystorepass | Keystore password. Optional. Prompts if not defined.<br><br>For Safenet, use the user PIN for the slot used by Sterling Secure Proxy.<br><br>For Thales, the passphrase for the operator smart card that will be used to protect the key. The card must be inserted in the module's card reader. |
| keypass | Passphrase for the key on the keystore. Optional. Prompts if not defined.<br><br>For Safenet, this parameter is not used.<br><br>For Thales, define this parameter using the same value as the keystore password. |

# Store a Certificate on the HSM Device

## About this task

If you have an existing certificate in the Sterling Secure Proxy certificate store, use the manageKeyCerts -storeOnHsm command to store the key certificate in the HSM. Stop CM before you use this command.

Consider the following before you use this command:

- If you define the engine parameter, the certificate is stored at the HSM for the engine. You must enable HSM on the engine.
- If you configure a netHSM and multiple engines access the netHSM, any of the engines can be specified to run the request.
- If the engine uses a PCI module and the module cannot be accessed by other engines, group the key certificates for the engine into a separate system certificate store. You cannot share the key certificates on that system certificate store with other engines.
- If you do not specify the engine parameter, and HSM support is enabled on CM, the key certificate is stored on the HSM configured at CM.
- If the key certificate is already stored in an HSM, the command fails.
- After a key certificate is stored in an HSM, the key certificate record at CM is updated with a reference to the key in the HSM. If it has a PEM private key, the private key is deleted from the certificate store.

To store a key certificate on the HSM, type the following command:

```
manageKeyCerts -storeOnHsm [parameters]
```

Refer to the following table for a description of the storeOnHsm parameters:

| Parameter | Description |
|---|---|
| certName | Name of the key certificate on Sterling Secure Proxy. Required. |
| certStore | Name of the system certificate store where the key certificate is stored. This field is optional. Default=dfltKeyStore. |
| engine | Name of the engine with access to the HSM. Optional. |

| Parameter | Description |
|---|---|
| alias | Alias for the key certificate on the HSM. Optional.<br><br>If no value is defined, the alias defaults to certificate name. |
| systempass | CM system passphrase. |
| adminid | Administrator ID. Optional. Prompts if not defined. |
| adminpass | Administrator password. Optional. Prompts if not defined. |
| keystorepass | Keystore password. Optional. Prompts if not defined.<br><br>For Safenet, the user PIN for the slot used by Sterling Secure Proxy.<br><br>For Thales, the passphrase for the operator smart card that will be used to protect the key. The card must be inserted in the module's card reader. |
| keypass | Passphrase for the key on the keystore. Optional. Prompts if not defined.<br><br>For Safenet, this parameter is not used.<br><br>For Thales, define this parameter using the same value as the keystore password. |

# Copy a Certificate

## About this task

Use the manageKeyCerts -copy command to copy an existing key certificate and assign it a new name. Stop CM before you run this command.

Consider the following before you use this command:

- If you specify the engine parameter and the key certificate is stored in an HSM, the engine makes a copy of the key certificate on the HSM, using the new alias provided. HSM support must be enabled at the engine to run this command.
- If you configure netHSM and multiple engines access it, specify any of the engines to run the request.
- If you do not specify the engine parameter and the key certificate is stored in an HSM, the command makes a copy of the certificate on the HSM configured at CM, using the new alias provided. You must configure the HSM at CM to use this command.
- If the key certificate is not stored in an HSM, the engine and new alias parameters are ignored.

To copy a key certificate, type the following command:

```
manageKeyCerts -copy [parameters]
```

Following is a description of the copy parameters:

| Parameter | Description |
|---|---|
| certName | Name of the key certificate on Sterling Secure Proxy. Required. |

| Parameter | Description |
|---|---|
| certStore | Name of the system certificate store where the key certificate is stored. This field is optional. Default=dfltKeyStore. |
| newName | Name for the copy of the key certificate. Required. Default=certName. |
| newAlias | Alias for the copy of the key certificate on the HSM. This parameter is required if the key certificate is stored on the HSM. |
| replace | Whether to replace a key certificate if a certificate with the new name already exists in the Sterling Secure Proxy system certificate store. Optional. Valid values = n \| y \| false \| true. Default=n. |
| engine | Name of the engine with access to the HSM. Optional. |
| systempass | CM system passphrase. |
| adminid | Administrator ID. Optional. Prompts if not defined. |
| adminpass | Administrator password. Optional. Prompts if not defined. |

# Delete a Certificate

## About this task

Use the manageKeyCerts -delete command to delete a key certificate from the Sterling Secure Proxy keystore or from the HSM. Stop CM before you use this command.

Consider the following before you use this command:

- If the system certificate is in use, the command fails. The list of netmap nodes using the system certificate is displayed.
- If the key certificate is stored in an HSM, specify the deleteFromHsm parameter to delete the key certificate from the HSM as well.
- If the engine parameter is defined, the key certificate is stored in an HSM, and deleteFromHSM is set to yes, the key certificate is deleted from the HSM at the engine. You must configure HSM support at the engine to use this command.
- If a netHSM is configured and multiple engines access the netHSM, any of the engines can be specified to run the command.
- If the engine parameter is not specified, the key certificate is stored in an HSM, and the deleteFromHSM is set to yes, the command deletes the key certificate from the HSM at CM. HSM support must be enabled at CM to use this command.
- If the key certificate is not stored in an HSM, the deleteFromHSM and engine parameters are ignored.

To delete the key certificate from the Sterling Secure Proxy certificate store, type the following command:

```
manageKeyCerts -delete [parameters]
```

Following is a description of the delete parameters:

| Parameter | Description |
|---|---|
| certName | Name of the key certificate on Sterling Secure Proxy. Required. |
| certStore | Name of the system certificate store where the key certificate will be stored. This field is optional. Default=dfltKeyStore. |
| deleteFromHsm | Determines whether to delete the key certificate from the HSM. This parameter is required if the key certificate is stored on the HSM. Valid values = y \| n \| true \| false. |
| engine | Name of the engine with access to the HSM. Optional. |
| systempass | CM system passphrase. |
| adminid | Administrator ID. Optional. Prompts if not defined. |
| adminpass | Administrator password. Optional. Prompts if not defined. |

# List Key Certificates on the HSM Device

## About this task

Use the manageKeyCerts -listHsm command to list keys on the HSM. This command can be run while CM is running.

Consider the following before you use this command:

- For Thales HSMs, all HSM keys that can be loaded with the provided smart card passphrase are listed, if the keyStoreData parameter is not defined.
- If you define the engine parameter, the keys stored on the HSM at the engine are listed. You must configure HSM support at the engine to use this command.
- If a netHSM is used and multiple engines access it, any of the engines can be specified to run the request.
- If the engine parameter is not defined, the command lists the keys stored on the HSM at CM. HSM support must be enabled at CM.

To list the key certificate on the HSM device, type the following command:

```
manageKeyCerts -listHsm [parameters]
```

Refer to the following table for a description of the list parameters:

| Parameter | Description |
|---|---|
| engine | Name of the engine with access to the HSM. Optional. |
| keyStoreData | HSM keystore blob string. Used with the Thales HSM. This is a 40-character hex string, displayed as "Key Instance" by the Thales KeySafe utility. If it is not provided, all keys that can be loaded with the provided smart card passphrase are listed. Alternatively, the blob string can be written to a file. Specify that file name in the keyStoreFile parameter. |
| keyStoreFile | File containing HSM keystore data. If defined, this parameter overrides the keyStoreData parameter. |
| systempass | CM system passphrase. |
| adminid | Administrator ID. Optional. Prompts if not defined. |

| Parameter | Description |
|---|---|
| adminpass | Administrator password. Optional. Prompts if not defined. |
| keystorepass | Keystore password. Optional. Prompts if not defined.<br><br>For Safenet, the user PIN for the slot used by Sterling Secure Proxy.<br><br>For Thales, the passphrase for the operator smart card that will be used to protect the key. The card must be inserted in the module's card reader. |

# Load References to Keys on the HSM into the Sterling Secure Proxy System Certificate Store

## About this task

Use the manageKeyCerts -loadHsm command to load references to keys on the HSM device into the Sterling Secure Proxy system certificate store. Stop CM before you run this command.

Consider the following before you use this command:

- This command is the same as the -getFromHSM command invoked for a list of HSM keys in a properties file. It facilitates HSM key migration from Sterling Secure Proxy 2.0.02 to Sterling Secure Proxy 3.1.0, and provides an easy way to populate Sterling Secure Proxy with HSM keys.
- If you define the engine parameter, the keys stored on the HSM at the engine are listed. Enable HSM support at the engine to use this command. If a netHSM is used and multiple engines access the netHSM, any of the engines can be specified to handle the request.
- If you do not specify the engine parameter, keys stored on the HSM at CM are listed. HSM support must be enabled at CM to use this command.
- After a reference to an HSM key certificate is imported into Sterling Secure Proxy, that HSM key cannot be referenced again under a different Sterling Secure Proxy system certificate name.
- To override the certificate store for a certificate, use the certStore=<store name> in the input properties file.

To load references to keys on the HSM device into the system certificate store, type the following:

```
manageKeyCerts -loadHsm [parameters]
```

Refer to the following table for a description of the loadHsm parameters:

| Parameter | Description |
|---|---|
| certStore | Name of the system certificate store where the key certificate will be stored. This field is optional. If the store does not exist, it is created. Default=dfltKeyStore. |
| engine | Name of the engine with access to the HSM. Optional. |
| systempass | CM system passphrase. |
| adminid | Administrator ID. Optional. Prompts if not defined. |

| Parameter | Description |
| --- | --- |
| adminpass | Administrator password. Optional. Prompts if not defined. |
| keystorepass | Keystore password. Optional. Prompts if not defined.<br><br>For Safenet, the user PIN for the slot used by Sterling Secure Proxy.<br><br>For Thales, the passphrase for the operator smart card that will be used to protect the key. The card must be inserted in the module's card reader. |
| autoGenName | Whether to auto-generate the name for the key certificate on Sterling Secure Proxy. Optional. If enabled, and the properties for the key certificate do not specify the certName property, a name is generated using the prefix "hsm_" followed by a hash of the key certificate properties (alias, keystore, type, provider, issuer, subject, serial). Default=n. |
| replace | Whether to replace a key certificate if a certificate with the same name already exists in the Sterling Secure Proxy system certificate store. Optional. Valid values = false \| true. Default=false. |
| file | Path to the file containing information about the HSM key certificates to load. Required.<br><br>It refers to the output of the Sterling Secure Proxy 2.0.02 RemoveSystemCert -l script, the -listHSM command, or a text file with lines in the format key=value. To load multiple key certificates, separate the properties for each with a blank line or a line starting with "[".<br><br>All key certificates on the HSM device are listed. The command searches the property file, to find a key certificate on the HSM that matches the specified property. If a match is found, an entry for the matched HSM key certificate is added to the Sterling Secure Proxy system certificate store with the name specified in the properties, or with an auto-generated name if the parameter called autoGenName=y.<br><br>If the file is the output of the Sterling Secure Proxy 2.0.02 RemoveSystemCert -l script, the lines on the file are mapped to properties as follows:<br>• PrivateKeyInfo for ID-alias (for Safenet)<br>• Name-certName<br>• KeyStoreType-type<br>• Issuer-issuer<br>• Subject-subject<br>• Serial-serial<br><br>If the file is the output of the version 2.0.02 RemoveSystemCert -l script, remove all lines up to, but not including, the first "PrivateKeyInfo for ID" at the top of the file.<br><br>If the file is the output of the manageKeyCerts -listHSM script, remove all lines up to, but not including, the first "[1]==========", from the top of the file. |

## Update the HSM Password for HSM Key Certificates Stored in the Sterling Secure Proxy System Store

### About this task

Use the manageKeyCerts -updateHsmPass command after you change the password for the HSM, using the HSM administration utilities. Stop CM before you run this command.

Consider the following before you use this command:

- This command does not change the HSM keystore password. It is changed through the HSM administration utilities. You must stop and restart the engine after you change a key store password through the HSM administration utilities.
- If you define the engine parameter, this command first tries to load the HSM keys with their current passwords. If a key cannot be loaded, it tries to load the HSM keys with the new password. If the key is successfully loaded, the password for the key is updated on Sterling Secure Proxy. HSM support must be enabled at the engine to use this command. If netHSM is used and multiple engines access the netHSM, any of the engines can be specified to handle the request.
- To update the keystore password on all system certificates, define the certStore=* parameter.

To update the password of the HSM on the Sterling Secure Proxy system certificate store, type the following command:

```
manageKeyCerts -updateHsmPass [parameters]
```

Following is a description of the updateHsmPass parameters:

| Parameter | Description |
|---|---|
| certStore | Name of the system certificate store where the key certificates are stored. This field is optional. Default=dfltKeyStore. |
| engine | Name of the engine with access to the HSM. Optional. |
| systempass | CM system passphrase. |
| adminid | Administrator ID. Optional. Prompts if not defined. |
| adminpass | Administrator password. Optional. Prompts if not defined. |
| newKeyStorePass | New HSM keystore password. Optional.<br><br>For Safenet, the new user PIN for the slot used by Sterling Secure Proxy.<br><br>For Thales, the new passphrase for the operator smart card that will be used to protect the key. The card must be inserted in the module's card reader. |

# Manage CSRs

### About this task

Use the manageCSRs command on CM to manage Certificate Signing Requests (CSRs). CSRs created with this command cannot be viewed through the CM GUI.

### Procedure

1. Create a CSR. The script generates a temporary self-signed key certificate in the HSM or an Sterling Secure Proxy system certificate store, if the HSM is not enabled. Then send the CSR to a Certification Authority (CA).
2. When the CA returns the CA-signed certificate, run the manageCSRs command again to replace the self-signed key certificate with the CA-signed certificate. The updated CA-signed certificate is added to the Sterling Secure Proxy system certificate store, and the CSR status is set to complete.

   The key certificate can now be used by Sterling Secure Proxy.
3. Use the manageCSRs command to perform the following tasks:
   - Create a CSR
   - Update a CSR
   - Delete a CSR
   - List CSRs on the CM Store
   - Retrieve a CSR to Send to a Certification Authority
   - Retrieve the CA-signed Certificate

## Create a CSR

### About this task

Use the manageCSRs -create command to create a CSR for a key certificate at either the HSM or the Sterling Secure Proxy system certificate store. You can use this command while CM is running.

Consider the following before you use this command:

- If you define the engine parameter, a key certificate is created on the HSM configured for the engine. You must enable HSM support at the engine in order to run this command. If a netHSM is used and multiple engines access the netHSM, any of the engines can be specified to handle the request.
- If you do not define the engine parameter and HSM support is not enabled on CM, the system certificate store certificate is created on the Sterling Secure Proxy system certificate store.

To create a CSR on Microsoft Windows:

```
manageCSRs -create [parameters]
```

To create a CSR on UNIX or Linux:

```
manageCSRs.sh -create [parameters]
```

Following is a description of the create CSR parameters:

| Parameter | Description |
|---|---|
| csrName | Name for the CSR. Required. |
| engine | Name of the engine with access to the HSM. Optional. |
| alias | Alias for the key certificate on the HSM. Optional.<br><br>If no value is defined, the alias defaults to CSR name. |
| keySize | Key size of the file to create. Valid values = 1024 \| 2048 \| 4096 Default=1024. |

| Parameter | Description |
|---|---|
| CN | Certificate common name. Required.<br><br>If the name contains spaces, enclose the command and string in double quotes, for example, "CN=my name". |
| O | Organization. Optional.<br><br>If the value contains spaces, enclose the command and string in double quotes, for example, "O=my org". |
| OU | Organization unit. Optional. Repeat this parameter to specify more than one organization unit. If the value contains spaces, enclose the command and string in double quotes, for example, “OU=my unit”. |
| L | Location (city). Optional.<br><br>If the value contains spaces, enclose the command in double quotes, for example, "L=my location". |
| ST | State. Optional.<br><br>If the value contains spaces, enclose the command and string in double quotes, for example, "ST=my state". |
| C | Two letter country code. Optional. |
| email | E-mail address. Optional. |
| file | Fully-qualified path to the file where the CSR will be stored. If this parameter is not defined, the output of the CSR is displayed on the monitor.<br><br>To obtain the CSR information later, use the -getpkcs10 command. Optional. |
| systempass | CM system passphrase. Optional. |
| adminid | Administrator ID. Optional. Prompts if not defined. |
| adminpass | Administrator password. Optional. Prompts if not defined. |
| keystorepass | Keystore password. Optional. Prompts if not defined.<br><br>For Safenet, the user PIN for the slot used by Sterling Secure Proxy.<br><br>For Thales, the passphrase for the operator smart card, used to protect the key. Be sure that the card is in the card reader before you run this command. |
| keypass | Passphrase for the key on the keystore. Optional. Prompts if not defined.<br><br>This value is not used by the Safenet HSM. |

# Update a CSR

## About this task

Use the manageCSRs -update command to update a pending CSR with the CA-signed certificate. Stop CM before you run this command.

Consider the following when using this command:

- If the key certificate is created in an HSM and you specify the engine parameter, the command notifies the engine to update the key certificate on the HSM. Configure HSM support at the engine to use this command.
- If a netHSM is used and multiple engines access it, any of the engines can be specified to perform the update.
- If the engine uses a PCI module and that module cannot be accessed by other engines, you must group the key certificates for the engine in a separate system certificate store. You cannot share the key certificates on that system certificate store with other engines.
- If the key certificate was created in an HSM and you do not specify the engine parameter, the command updates the key certificate on the HSM at CM. You must enable HSM support at CM.
- If the key certificate was not created in an HSM, it is updated on the Sterling Secure Proxy system certificate store. The engine parameter is ignored.

To update a pending CSR, type the following command:

```
manageCSRs -update [parameters]
```

Following is a description of the update parameters:

| Parameter | Description |
|---|---|
| csrName | Name for the CSR. Required. |
| engine | Name of the engine with access to the HSM. Optional. |
| file | Fully-qualified path of the CA-signed certificate file. Required. |
| certName | Name of the key certificate on Sterling Secure Proxy. Required. |
| systempass | CM system passphrase. |
| adminid | Administrator ID. Optional. Prompts if not defined. |
| adminpass | Administrator password. Optional. Prompts if not defined. |
| newKeyStorePass | New HSM keystore password. Optional.<br><br>If defined, this value overrides the keystore password used when the CSR was created. This parameter allows you to update a CSR on the HSM after the keystore password for the HSM is changed. |

# Delete a CSR

## About this task

Use the manageCSRs -delete command to delete a CSR from the CM store. This command can be run while CM is running.

Consider the following when using this command:

- If the CSR is pending and its key certificate was generated on an HSM, the temporary key certificate is deleted from the HSM.

- If the CSR is complete, this command deletes the CSR, but does not delete the key certificate. To delete the key certificate, use the manageKeyCerts -delete command.

To delete a CSR from CM, type the following command:

```
manageCSRs -delete [parameters]
```

Following is a description of the delete parameters:

| Parameter | Description |
|---|---|
| csrName | Name for the CSR. Required. |
| engine | Name of the engine with access to the HSM. Optional. |
| systempass | CM system passphrase. |
| adminid | Administrator ID. Optional. Prompts if not defined. |
| adminpass | Administrator password. Optional. Prompts if not defined. |
| newKeyStorePass | New HSM keystore password. Optional.<br><br>If defined, this value overrides the keystore password used when the CSR was created. This parameter allows you to update a CSR on the HSM after the keystore password for the HSM is changed. |

## List CSRs on the CM Store

### About this task

Use the manageCSRs -list command to display a list of CSRs on CM. This command can be run while CM is running.

To list the CSRs in the CM store, type the following command:

```
manageCSRs -list [parameters]
```

Refer to the following table for a description of the list parameters:

| Parameter | Description |
|---|---|
| systempass | CM system passphrase. |
| adminid | Administrator ID. Optional. Prompts if not defined. |
| adminpass | Administrator password. Optional. Prompts if not defined. |

## Retrieve a CSR to Send to a Certification Authority

### About this task

Use the manageCSRs -getpkcs10 command to retrieve a CSR to send to a Certificate Authority (CA). This command can be run while CM is running.

To retrieve a CSR from the HSM that is ready to send to a CA, type the following command:

```
manageCSRs -getpkcs10 [parameters]
```

Refer to the following table for a description of the list parameters:

| Parameter | Description |
|---|---|
| csrName | Name for the CSR. Required. |
| file | Fully-qualified path of the file where the CSR will be stored. |
| systempass | CM system passphrase. Optional. |
| adminid | Administrator ID. Optional. Prompts if not defined. |
| adminpass | Administrator password. Optional. Prompts if not defined. |

## Retrieve the CA-signed Certificate

### About this task

Use the manageCSRs -getcacert command to retrieve the CA-signed certificate received from a CA, after the update command has been run. The certificate is returned in PEM format. This command can be run while CM is running.

To retrieve the CA-signed certificate from the HSM, type the following command:

```
manageCSRs -getcacert [parameter]
```

Refer to the following table for a description of the getcacert parameters:

| Parameter | Description |
|---|---|
| csrName | Name for the CSR. Required. |
| file | Fully qualified path where the CA-signed certificate will be stored. If not specified, the certificate text is written to the display. |
| systempass | CM system passphrase. Optional. Prompts if not defined. |
| adminid | Administrator ID. Optional. Prompts if not defined. |
| adminpass | Administrator password. Optional. Prompts if not defined. |

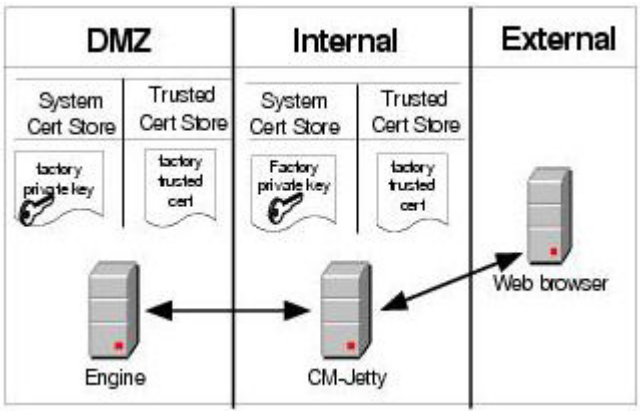
# Chapter 3. Manage CM Certificates

## Manage Certificates Between Sterling Secure Proxy Components

To maintain security in Sterling Secure Proxy, the engine and Configuration Manager (CM) communicate using SSL. Sterling Secure Proxy uses TCP/IP communications links between the web browser and the Jetty web server, the web server and CM, and CM and the engine. The only link that can be unsecure is between the web browser and the Jetty web server.

When you install Sterling Secure Proxy, a default certificate is installed to allow you to communicate. All components of the Sterling Secure Proxy system including CM, engine, and the Jetty web server share the same certificate. This self-signed certificate is called the factory certificate and has a ten year expiration.

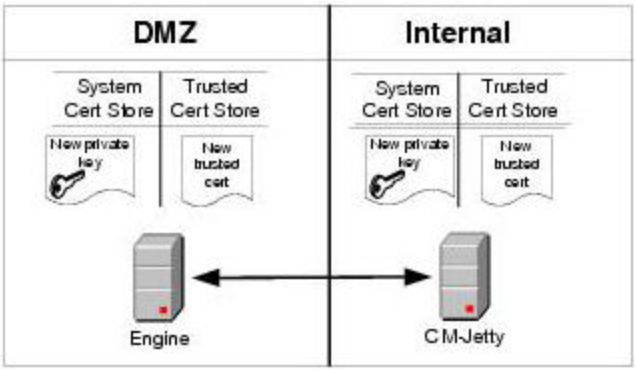
Before you can begin production, you must import a secure certificate. The default configuration uses a single key to secure the connection between the engine and CM. The certificate distribution looks like this:




To secure the communication between these components, replace the factory certificates using one of the models in this chapter.

## Use a Common Certificate for the Engine and CM

The simplest way to update the certificate distribution is to replace the factory certificate with a new certificate and use that certificate for both the engine and CM. The certificate distribution looks like this:




Following are the procedures to replace a factory certificate with a common certificate:

- Replace the Factory Certificate with a Common Certificate on UNIX or Linux
- Replace the Factory Certificate with a Common Certificate on Microsoft Windows

## Replace the Factory Certificate with an Engine Certificate and CM Certificate on UNIX or Linux

### About this task

To replace the factory certificates, with one certificate at the engine and a different certificate at CM on UNIX or Linux:

### Procedure

1. Stop CM.
   a. Navigate to the CM *install_dir*/bin directory, where *install_dir* is the installation directory, and type the following command: `./stopCM.sh`
   b. At the passphrase prompt, type the passphrase defined for the engine and press **Enter**.
   c. Type the user name and password for the administrator.
2. From *install_dir*/bin, type the following command to replace the factory certificate with a CM certificate: `./configureCmSsl.sh -u cmCert=<cert file> cmCertAlias=<alias>`

   where:

   - *<cert file>* is the path and file name to the certificate that replaces the factory certificate for CM.
   - *<alias>* is the alias name for the new CM certificate. It can be any value other than `factory`. If you do not specify an alias, `cm` is assigned as the default.
3. On the CM computer, type the following command to replace the factory certificate with an engine certificate: `./configureCmSsl.sh -u engCert=<cert file> engCertAlias=<alias>`

   where:

   - *<cert file>* is the path and file name to the certificate you want to use to replace the factory certificate for the engine.
   - *<alias>* is the alias name for the new engine certificate. This can be any value other than `factory`. If you do not specify an alias, `engine` is assigned as the default.
4. Type the following command to create an export file of the certificate store:

`configureCmSsl -e file=<export file>`

where <*export file*> is the path and file for the export file.

5. Copy the file you created in step 4 to the engine.
6. Stop the engine.
   a. On the engine, navigate to the *install_dir*/bin directory and type the following command: `./stopEngine.sh`
   b. At the passphrase prompt, type the passphrase defined for the engine and press **Enter**.
7. From the *install_dir*/bin directory, type the following command to import the certificates created in step 2 and step 3. `configureEngineSsl -i file=<export file> engCertAlias=<alias>`

   where:

   - <*export file*> is the path and file for the export file.
   - <*alias*> is an alias name for the engine certificate assigned in step step 3. If an engCertAlias was omitted in step 3, specify `engine` as the alias.
8. Start the engine.
   a. Type the following command: `./startEngine.sh`
   b. At the passphrase prompt, type the passphrase for the engine and press **Enter**.
9. Start CM.
   a. Navigate to the *install_dir*/bin directory and type the following command: `./startCM.sh`
   b. At the passphrase prompt, type the passphrase defined for CM and press **Enter**.

## Replace the Factory Certificate with a Common Certificate on Microsoft Windows

### About this task

To replace the factory certificate used between the engine and CM on Microsoft Windows:

### Procedure

1. Stop CM on Microsoft Windows from Microsoft Windows services.
2. Using a command line interface on CM, navigate to the *install_dir*\bin directory.
3. Type the following command to replace the factory certificate:

   `configureCmSsl -u commonCert=<cert file> commonCertAlias=<alias>`

   where:

   - <*cert file*> is the path and file name to the certificate that replaces the factory certificate.
   - <*alias*> is an alias name for the new certificate. This can be any value other than `factory`. If you do not specify an alias, `common` is assigned as the default.
4. Type the following command to create an export file of the certificate store: `configureCmSsl -e file=<export file>`

   where <*export file*> is the path and file for the export file.
5. Copy the file you created in step 3 to the engine.
6. Stop the engine on Microsoft Windows from Microsoft Windows services.

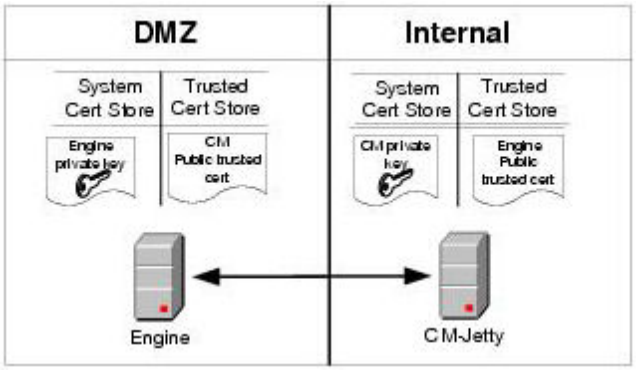
7. Using a command line at the engine, navigate to the *install_dir*\bin directory and type the following command to import the certificate store created in step 4. `configureEngineSsl -i file=<export file> engCertAlias=<alias>`

   where:

   - *<export file>* is the path and file for the export file.
   - *<alias>* is the alias name for the new certificate assigned in step 4.
8. Start the engine on Microsoft Windows from Microsoft Windows services.
9. Start CM on Microsoft Windows from Microsoft Windows services.

## Use Different Certificates for the Engine and CM

You can use different certificates to secure the engine-to-CM connection and to secure the Jetty web server-to-CM connection. This certificate distribution is illustrated below:




Following are the procedures to replace a factory certificate with an engine and a CM certificate:

- Replace the Factory Certificate with an Engine Certificate and CM Certificate on UNIX or Linux
- Replace the Factory Certificate with an Engine and CM Certificate on Microsoft Windows

### Replace the Factory Certificate with a Common Certificate on UNIX or Linux

#### About this task

To replace the factory certificate used between the engine and CM on UNIX or Linux:

#### Procedure

1. Stop CM.
   a. At CM, navigate to the *install_dir*/bin directory, where *install_dir* is the installation directory, and type the following command: `./stopCM.sh`
   b. At the passphrase prompt, type the passphrase defined for the engine and press **Enter**.
   c. Type the user name and password for the administrator.
2. Type the following command to replace the factory certificate:
   `./configureCmSsl.sh -u commonCert=<cert file> commonCertAlias=<alias>`

where:

- <*cert file*> is the path and file name to the certificate that replaces the factory certificate.
- <*alias*> is an alias name for the new certificate. This can be any value other than `factory`. If you do not specify an alias, `common` is assigned as the default.

3. Type the following command to create an export file of the certificate store:
   `./configureCmSsl.sh -e file=<export file>`
   where <*export file*> is the path and file for the export file.
4. Copy the file you created in step 3 to the engine.
5. Stop the engine.
   a. At the engine, navigate to the *install_dir*/bin directory and type the following command:
      `./stopEngine.sh`
   b. Type the passphrase defined for the engine and press **Enter**.
6. At the engine, navigate to the *install_dir*/bin directory and type the following command to import the certificate store created in step 3:
   `./configureEngineSsl.sh -i file=<export file> engCertAlias=<alias>`
   where:
   - <*export file*> is the path and file for the export file.
   - <*alias*> is the alias name for the new certificate assigned in step 2
7. Start the engine.
   a. From *install_dir*/bin, type the following command:
      `./startEngine.sh`
   b. At the passphrase prompt, type the passphrase for the engine and press **Enter**.
8. Start CM.
   a. Navigate to the *install_dir*/bin directory and type the following command:
      `./startCM.sh`
   b. At the passphrase prompt, type the passphrase defined for CM and press **Enter**.

## Replace the Factory Certificate with an Engine and CM Certificate on Microsoft Windows

### About this task

To replace the factory certificate, with a certificate at the engine and a different certificate at CM on Microsoft Windows:

### Procedure

1. Stop CM on Microsoft Windows from Microsoft Windows services.
2. Using a command line interface at CM, navigate to the *install_dir*\bin directory.
3. Type the following command to replace the factory certificate with a CM certificate: `configureCmSsl -u cmCert=<cert file> cmCertAlias=<alias>`
   where:
   - <*cert file*> is the path and file name to the certificate that replaces the factory certificate for CM.

- *<alias>* is the alias name for the new CM certificate. It can be any value other than `factory`. If you do not specify an alias, `cm` is assigned as the default.

4. On the CM computer, type the following command to replace the factory certificate with an engine certificate: `configureCmSsl -u engCert=<cert file> engCertAlias=<aaliaslias>`

   where:

   - *<cert file>* is the path and file name to the certificate you want to use to replace the factory certificate for the engine.
   - *<alias>* is the alias name for the new engine certificate. This can be any value other than `factory`. If you do not specify an alias, `engine` is assigned as the default.

5. Type the following command to create an export file of the certificate store: `configureCmSsl -e file=<export file>`

   where *<export file>* is the path and file for the export file.

6. Copy the file you created in step 5 to the engine.
7. Stop the engine on Microsoft Windows from Microsoft Windows services.
8. Type the following command to import the certificates created in step 3 and step 4. `configureEngineSsl -i file=<export file> engCertAlias=<alias>`

   where:

   - *<export file>* is the path and file for the export file.
   - *<alias>* is an alias name for the engine certificate assigned in step 3. If an engCertAlias was omitted in step 3, specify `engine` as the alias.

9. Start the engine on Microsoft Windows from Microsoft Windows services.
10. Start CM on Microsoft Windows from Microsoft Windows services.

# Restore the Factory Certificate on UNIX or Linux

## About this task

To restore the certificate distribution to the factory settings on UNIX or Linux:

## Procedure

1. Stop CM.
   a. Navigate to the *install_dir*/bin directory, where *install_dir* is the installation directory, and type the following command: `./stopCM.sh`
   b. At the passphrase prompt, type the passphrase defined for the engine and press **Enter**.
   c. Type the user name and password for the administrator.
2. From the *install_dir*/bin directory, type the following command to restore the factory certificate: `./configureCmSsl.sh -r`
3. Type the following command to export the factory-restored certificate store: `./configureCmSsl.sh -e file=<export file>`

   where *<export file>* is the path and file for the *export file*.

4. Copy the *export file* to the engine.
5. Stop the engine.
   a. Navigate to the *install_dir*/bin directory and type the following command: `./stopEngine.sh`
   b. At the passphrase prompt, type the passphrase defined for the engine and press **Enter**.

6. From the *install_dir*/bin directory, type the following command to import the factory-restored certificate store: `configureEngineSsl -i file=<export file> engCertAlias=factory`

   where *<export file>* is the path and file for the certificate store.
7. Start the engine.
   a. Type the following command: `./startEngine.sh`
   b. At the passphrase prompt, type the passphrase for the engine and press **Enter**.
8. Start CM.
   a. On CM, navigate to the *install_dir*/bin directory and type the following command: `./startCM.sh`
   b. At the passphrase prompt, type the passphrase defined for CM and press **Enter**.

   **Note:** Restoring the configuration to use the factory certificate does not delete the certificates that were previously in use.

## Restore the Factory Certificate on Microsoft Windows

### About this task

To restore the certificate distribution to the factory settings on Microsoft Windows:

### Procedure

1. Stop CM on Microsoft Windows from Microsoft Windows services.
2. From the *install_dir*\bin directory, type the following command to restore the factory certificate: `configureCmSsl -r`
3. Type the following command to export the factory-restored certificate store: `configureCmSsl -e file=<export file>`

   where *<export file>* is the path and file for the export file.
4. Copy the export file to the engine.
5. Stop the engine on Microsoft Windows from Microsoft Windows services.
6. From the *install_dir*/bin directory, type the following command to import the factory-restored certificate store:

   `configureEngineSsl -i file=<export file> engCertAlias=factory`

   where *<export file>* is the path and file for the certificate store.
7. Start the engine on Microsoft Windows from Microsoft Windows services.
8. Start CM on Microsoft Windows from Microsoft Windows services.

   **Note:** Restoring the configuration to use the factory certificate does not delete the certificates that were previously in use.

# Change the Password of the CM Key Store and Trust Store on UNIX or Linux

### About this task

To change the password:

### Procedure

1. Stop CM.

a. Navigate to the *install_dir*/bin directory, where *install_dir* is the installation directory, and type the following command: `./stopCM.sh`
b. At the passphrase prompt, type the passphrase defined for the engine and press **Enter**.
c. Type the user name and password for the administrator.

2. From the *install_dir*/bin directory, type the following command: `./configureCmSsl.sh -x`
3. When prompted, type the existing password and press **Enter**.
4. Type the new password and press **Enter**.
5. Start CM.
   a. From the *install_dir*/bin directory, type the following command: `./startCM.sh`
   b. At the passphrase prompt, type the passphrase defined for CM and press **Enter**.

## Change the Password of the CM Key Store and Trust Store on Microsoft Windows

### About this task

To change the password:

### Procedure

1. Stop CM on Microsoft Windows from Microsoft Windows services.
2. From the *install_dir*/bin directory, type the following command: `configureCmSsl -x`
3. When prompted, type the existing password and press **Enter**.
4. Type the new password and press **Enter**.
5. Start CM on Microsoft Windows from Microsoft Windows services.

## Change the Password of the Engine Key Store and Trust Store on UNIX or Linux

### About this task

The password for the key store and the trust store is set to password at installation. To change the password:

### Procedure

1. Stop the engine.
   a. Navigate to the *install_dir*/bin directory, and type the following command: `./stopEngine.sh`
   b. At the passphrase prompt, type the passphrase defined for the engine and press **Enter**.
   c. Type the user ID and password of the administrator.
2. Using a command line interface on CM, navigate to the *install_dir*/bin directory and type the following command:
   `./configureEngineSsl.sh -x`
3. When prompted, type the existing password and press **Enter**.

4. Type the new password and press **Enter**.
5. Retype the password and press **Enter**.
6. Start the engine.
   a. Navigate to the *install_dir*/bin directory on the engine and type the following command:
      ```
      ./startEngine.sh
      ```
   b. At the passphrase prompt, type the passphrase for the engine and press **Enter**.

## Change the Password of the Engine Key Store and Trust Store on Microsoft Windows

### About this task

The password for the key store and the trust store is set to password at installation. To change the password:

### Procedure

1. Stop the engine on Microsoft Windows from Microsoft Windows services.
2. Using a command line interface on CM, navigate to the *install_dir*/bin directory and type the following command:
   ```
   configureEngineSsl -x
   ```
3. When prompted, type the existing password and press **Enter**.
4. Type the new password and press **Enter**.
5. Retype the password and press **Enter**.
6. Start the engine on Microsoft Windows from Microsoft Windows services.

## Configuration Utilities

Two utilities are used in the previous procedures to configure SSL:

- configureCmSsl
- configureEngineSsl

Refer to the tables below to identify the functions that can be performed on the engine and CM. You are prompted for a password when one is required.

Use the following functions to configure CM, using the configureCmSsl utility:

| Parameter | Description |
|---|---|
| -s | Show current configuration. |

| Parameter | Description |
| --- | --- |
| -u | Update configuration. Available options include:<br>• commonCert—fully-qualified location of the common certificate to be shared by the Sterling Secure Proxy components engine, CM, and web server.<br>• commonCertAlias—alias for the common certificate and shared by all Sterling Secure Proxy components. If the certificate file name is omitted, a certificate with this alias must exist in the key store. If no alias is provided, the value defaults to common.<br>• cmCert—the fully-qualified location of CM and jetty web server certificate.<br>• cmCertAlias—alias for the CM/jetty web server certificate. If no file name is provided, a certificate with this alias must exist in the key store. If no alias is provided, the value defaults to cm.<br>• engCert—the fully-qualified location of the engine certificate.<br>• engCertAlias—alias for the engine certificate.<br>• webCert—the fully-qualified location of the jetty web server certificate.<br>• webCertAlias—alias for the jetty web server certificate. If no file name is provided, a certificate with this alias must exist in the key store. If no alias is provided, the value defaults to webserver.<br>• cmClientCert—the fully-qualified location of the CM client certificate.<br>• cmClientCertAlias—alias for the CM client certificate. If no file name is provided, a certificate with this alias must exist in the key store. This certificate is used by CM to communicate with the engine. If no alias is provided, the value defaults to cmServer.<br>• cmServerCert—the fully-qualified location of the CM server certificate.<br>• cmServerCertAlias—alias for the CM server certificate. If no file name is provided, a certificate with this alias must exist in the key store. This certificate is used by CM to communicate with the jetty web server. If no alias is provided, the value defaults to cmClient.<br>• cmSslProt—the SSL or TLS protocol used for the session between CM and the engine. Valid values are: SSLv2, SSLv3, TLSv1, or SSLv2Hello.<br>• cmCiphers—ordered list of cipher suites for communication between CM and the engine. Separate ciphers with a comma, colon, or semicolon.<br>• https—identifies if security is enabled between a web browser and the jetty web server. n = disable security, Y = security enabled. https is enabled by default.<br>• webHost—the IP bind address for the jetty web server. The default value is localhost. If CM has multiple NIC cards, use the field to specify the IP address of the NIC card to use for the jetty web server.<br>• webPort—the listen port for the jetty server. The default value is 8443. |
| | • webSslProt—the SSL or TLS protocol for the link between the web browser and the jetty web server. Valid values include SSLV2, SSLv3, TLSv1, or SSLv2Hello.<br>• webCiphers—an ordered list of cipher suites to use on the connection between the web browser and jetty web server. Separate ciphers with a comma, colon, or semicolon.<br>• clientAuth—enables client authentication for web browser clients. n= disabled. y = enabled. This option is set to n by default. If you enable clientAuth, you must add trusted certificates for the web server clients.<br>• trustedCert—fully-qualified location of the trusted certificate for the web client. |

| Parameter | Description |
|---|---|
| -e | Export configuration. The export option is:<br>• file—to identify the fully-qualified location of the export file. It can be imported into CM or the engine. |
| -i | Import configuration. The import option is:<br>• file—to identify the fully-qualified location of the export file. It can be imported into CM or the engine. |
| -d | Delete a certificate. The delete option is:<br>• alias—the alias of the certificate to delete. This can be specified multiple times. |
| -x | Change key store password. |
| -r | Restore factory settings. |
| -h | List the usage and parameters of the command. |

Use the following functions to configure SSL on the engine, using the configureEngineSsl utility:

| Parameter | Description |
|---|---|
| -s | Show current configuration. |
| -i | Import configuration. Options include:<br>• file—the fully-qualified location of the import file.<br>• engCertAlias—the alias for the engine certificate. |
| -d | Delete a certificate. Options include:<br>• alias—the alias of the certificate to delete. This can be specified multiple times. |
| -x | Change key store password. |
| -h | List the usage and parameters of the command. |

# Notices

This information was developed for products and services offered in the U.S.A.

IBM® may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing*

*IBM Corporation*

*North Castle Drive*

*Armonk, NY 10504-1785*

*U.S.A.*

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

*Intellectual Property Licensing*

*Legal and Intellectual Property Law*

*IBM Japan Ltd.*

*1623-14, Shimotsuruma, Yamato-shi*

*Kanagawa 242-8502 Japan*

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:**
INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be

incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

*IBM Corporation*

*J46A/G4*

*555 Bailey Avenue*

*San Jose, CA 95141-1003*

*U.S.A.*

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© IBM 2012. Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. 2012.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

## Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at “Copyright and trademark information” at http://www.ibm.com/legal/copytrade.shtml.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Linear Tape-Open, LTO, the LTO Logo, Ultrium and the Ultrium Logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

Connect Control Center®, Connect:Direct®, Connect:Enterprise®, Gentran®, Gentran®:Basic®, Gentran:Control®, Gentran:Director®, Gentran:Plus®, Gentran:Realtime®, Gentran:Server®, Gentran:Viewpoint®, Sterling Commerce™, Sterling Information Broker®, and Sterling Integrator® are trademarks or registered trademarks of Sterling Commerce™, Inc., an IBM Company.

Other company, product, and service names may be trademarks or service marks of others.

Printed in USA