



RED HAT CERTIFICATE SYSTEM (RHCS)

Integration Guide

Applicable Devices:

KMES Series 3

Applicable Versions:

6.3.1.x



THIS DOCUMENT CONTAINS CONFIDENTIAL INFORMATION PROPRIETARY TO FUTUREX, LP. ANY UNAUTHORIZED USE, DISCLOSURE, OR DUPLICATION OF THIS DOCUMENT OR ANY OF ITS CONTENTS IS EXPRESSLY PROHIBITED.

TABLE OF CONTENTS

[1] DOCUMENT INFORMATION	3
[1.1] DOCUMENT OVERVIEW	3
[1.2] APPLICATION DESCRIPTION	3
[2] PREREQUISITES	4
[3] INSTALL FUTUREX PKCS #11 (FXPKCS11)	5
[3.1] INSTALLING THE FXPKCS11 MODULE IN LINUX	5
[4] KMES SERIES 3 CONFIGURATION	6
[4.1] CREATE A ROLE AND IDENTITY FOR RED HAT CERTIFICATE SYSTEM	6
[4.2] ENABLE THE HOST API COMMANDS REQUIRED FOR THE RED HAT CERTIFICATE SYSTEM OPERATION	7
[4.3] CONFIGURE TLS COMMUNICATION BETWEEN THE KMES SERIES 3 AND THE FUTUREX PKCS #11 (FXPKCS11) LIBRARY	7
[5] EDIT THE FUTUREX PKCS #11 CONFIGURATION FILE	13
[5.1] SPECIAL DEFINES REQUIRED FOR THIS INTEGRATION	14
[6] RED HAT CERTIFICATE SYSTEM INSTALLATION AND SUBSYSTEM DEPLOYMENT	15
[6.1] INSTALL RHCS AND ITS PREREQUISITES	15
[6.2] MODIFY SELINUX TO SUPPORT SUBSYSTEM DEPLOYMENT USING AN HSM	15
[6.3] RUN THE PKISPAWN SCRIPT TO CREATE AND CONFIGURE A SUBSYSTEM INSTANCE	16
[6.4] VIEW THE KEYS AND CERTIFICATES THAT RHCS CREATED ON THE KMES SERIES 3	18
[6.5] IMPORT THE CA ADMINISTRATOR PKCS #12 FILE INTO THE BROWSER	19
[6.6] ACCESSING THE NEW CA SUBSYSTEM IN THE BROWSER	19
APPENDIX A: XCEPTIONAL SUPPORT	20

[1] DOCUMENT INFORMATION

[1.1] DOCUMENT OVERVIEW

The purpose of this document is to provide information regarding the configuration of the Futurex KMES Series 3 with Red Hat Certificate System (RHCS) using Futurex PKCS #11 libraries. For additional questions related to your HSM, see the relevant user guide.

[1.2] APPLICATION DESCRIPTION

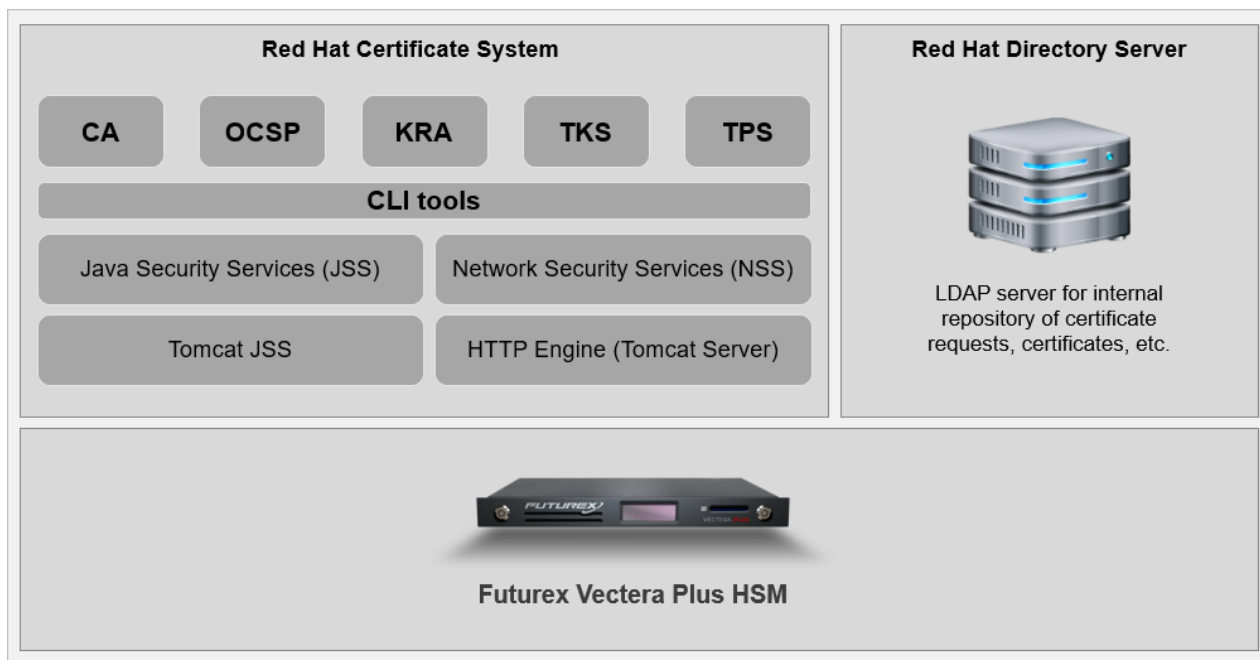
[1.2.1] About Red Hat Certificate System

From Red Hat's knowledge base website: "Red Hat Certificate System provides a powerful security framework to manage user identities and ensure communication privacy. Handling the major functions of the identity life cycle, Red Hat Certificate System simplifies enterprise-wide deployment and adoption of a public key infrastructure (PKI)."

[1.2.2] Basic architecture of an RHCS deployment

Although each RHCS subsystem (CA, KRA, OCSP, TKS, TPS) provides a different service, all share a common architecture. The following architectural diagram displays the common architecture shared by every subsystem type. For more information, please refer to the following Red Hat knowledge base article:

https://access.redhat.com/documentation/en-us/red_hat_certificate_system/9/html/planning_installation_and_deployment_guide/sect-certificate-system-architecture-overview



[2] PREREQUISITES

Supported Hardware:

- KMES Series 3, 6.3.1.x and above

Supported Operating Systems:

- Red Hat Enterprise Linux (RHEL)

Other:

- OpenSSL
- Red Hat Certificate System subscription (purchased through Red Hat)

[3] INSTALL FUTUREX PKCS #11 (FXPKCS11)

In a Linux environment, you must download a tarball of the **Futurex PKCS #11 (FXPKCS11)** binaries from the Futurex Portal and then extract the tar file locally where you want the application to be installed on your system. The following section provides step-by-step installation instructions.

Note: Install FXPKCS11 on the same computer as the application integrating with the KMES Series 3.

[3.1] INSTALLING THE FXPKCS11 MODULE IN LINUX

Extract the tarball file for your Linux distribution in the desired working directory.

Note: To make the Futurex PKCS #11 module accessible system-wide, move it to the `/usr/local/bin` directory as an administrative user. If only the current user needs to use the module, then install it in `$HOME/bin`.

The extracted content of the tar file is a single `fxpkcs11` directory. Inside the `fxpkcs11` directory is the following files and directories:

- **fxpkcs11.cfg:** FXPKCS11 configuration file
- **x86/:** This folder contains the module files for 32-bit architecture
- **x64/:** This folder contains the module files for 64-bit architecture

The x86 and x64 directories each contain two subdirectories, `OpenSSL-1.0.x` and `OpenSSL-1.1.x`. These OpenSSL directories contain the following FXPKCS11 module files built with the respective OpenSSL versions:

- **configTest:** Program to test configuration and connection to the HSM
- **libfxpkcs11.so:** FXPKCS11 Library File
- **libfxpkcs11-Debug.so:** FXPKCS11 Debug Library File
- **PKCS11Manager:** Program to test connection and manage the HSM through the FXPKCS11 library

By default, the FXPKCS11 module looks for the FXPKCS11 configuration file (i.e., `fxpkcs11.cfg`) in the `/etc` directory. Alternatively, a system environment variable can be defined for the location of the FXPKCS11 configuration file. To do so permanently, open the `/etc/profile` file in a text editor as an administrative user, add the following line at the bottom, and save the file.

```
export FXPKCS11_CFG=/usr/local/bin/fxpkcs11/fxpkcs11.cfg
```

Note: The file location specified above must be specific to where the FXPKCS11 configuration file is saved on your system.

[4] KMES SERIES 3 CONFIGURATION

The first half of this section covers general KMES Series 3 configurations that must be made to allow Red Hat Certificate System (RHCS) to integrate with the KMES to store various keys and certificates used in the CA subsystem operation. The second half of this section covers the required steps to configure TLS communication between the KMES Series 3 and the Futurex PKCS #11 (FXPKCS11) library, which RHCS will utilize to communicate with the KMES.

[4.1] CREATE A ROLE AND IDENTITY FOR RED HAT CERTIFICATE SYSTEM

A new role and identity need to be created on the KMES Series 3. This role will be assigned to the identity and subsequently will be used by the FXPKCS11 library to connect to the KMES.

1. Log in to the KMES Series 3 application interface with the default Admin identities.
2. Navigate to the **Identity Management > Roles** menu and click the [**Add...**] button. This will pull up the **Role Editor** dialog.
3. Specify a name for the role, select the **Hardened** checkbox, and set **Logins Required** to **1**.
4. In the **Permissions** tab, select the following permissions:
 - **Certificate Authority > Add | Upload | Export**
 - **Cryptographic Operations > Sign**
 - **Keys > Add**
5. In the **Advanced** tab, set the **Allowed Ports** field to **Host API**.
6. Click the [**OK**] button to finish creating the role.
7. Navigate to the **Identity Management > Identities** menu, then right-click and select **Add > Client Application**.
8. Change the **Storage** to **HSM** and specify a **Name** for the identity.
9. In the **Assigned Roles** tab, select the **Role** that you just created.
10. In the **Authentication** tab, click the [**Add**] button to configure a new credential. This will pull up the **Configure Credential** dialog.
11. Set the credential **Type** to **Password**. The **Provider** field should be set to **Futurex HSM** and the **Mechanism** field should be set to **Hardened Password**. Select the [**Change**] button and set a password for the credential, click [**Save**], then click [**OK**]. The new **Password** credential should be listed now under the **API Key** credential that exists by default.
12. In the main **Identity Editor** dialog, select the **API Key** credential and click [**Remove**], then click [**OK**] to save.

[4.2] ENABLE THE HOST API COMMANDS REQUIRED FOR THE RED HAT CERTIFICATE SYSTEM OPERATION

Because the Futurex PKCS #11 library will connect to the Host API port on the KMES, users must define which Host API commands are enabled for execution by the FXPKCS11 library. To set the enabled commands, complete the following steps:

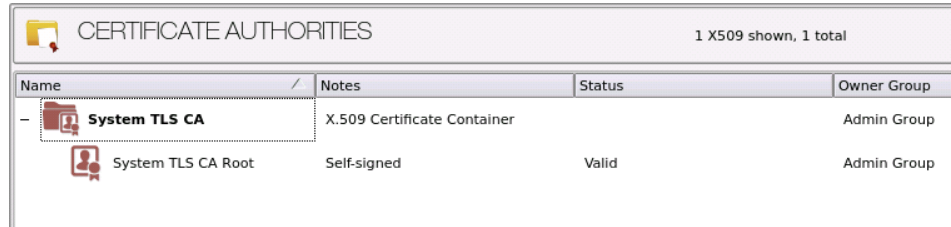
1. Go to **Administration > Configuration > Host API Options**, enable the commands listed below, then click [**Save**].
 - **ATKG**
 - **add** - Add HSM trusted asymmetric key group
 - **get** - Retrieve HSM trusted asymmetric key group
 - **ECHO** - Communication Test/Retrieve Version
 - **RKCP**
 - **get** - Retrieve enabled commands
 - **RKCY** - Create Certificate Authority
 - **RKGP** - Export Asymmetric HSM Trusted Key
 - **RKGS** - Generate Signature
 - **RKIC** - Import Certificate
 - **RKLN** - Lookup Objects
 - **RKLO** - Login User
 - **RKPK** - Pop Generated Key
 - **RKRK** - Retrieve Certificate

[4.3] CONFIGURE TLS COMMUNICATION BETWEEN THE KMES SERIES 3 AND THE FUTUREX PKCS #11 (FXPKCS11) LIBRARY

[4.3.1] Create an X.509 Certificate Container and Root CA Certificate

1. Navigate to the **PKI > Certificate Authorities** menu, then click the [**Add CA...**] button at the bottom of the page.
2. In the **Certificate Authority** dialog, enter a **Name** for the Certificate Container, leave all other fields set to the default values, then click [**OK**].
3. Right-click on the Certificate Container that was created and select **Add Certificate > New Certificate...**
4. In the **Subject DN** tab, select **Classic** in the **Preset** dropdown and set a **Common Name** for the certificate, such as *System TLS CA Root*.
5. In the **Basic Info** tab, leave all settings set to the default values.

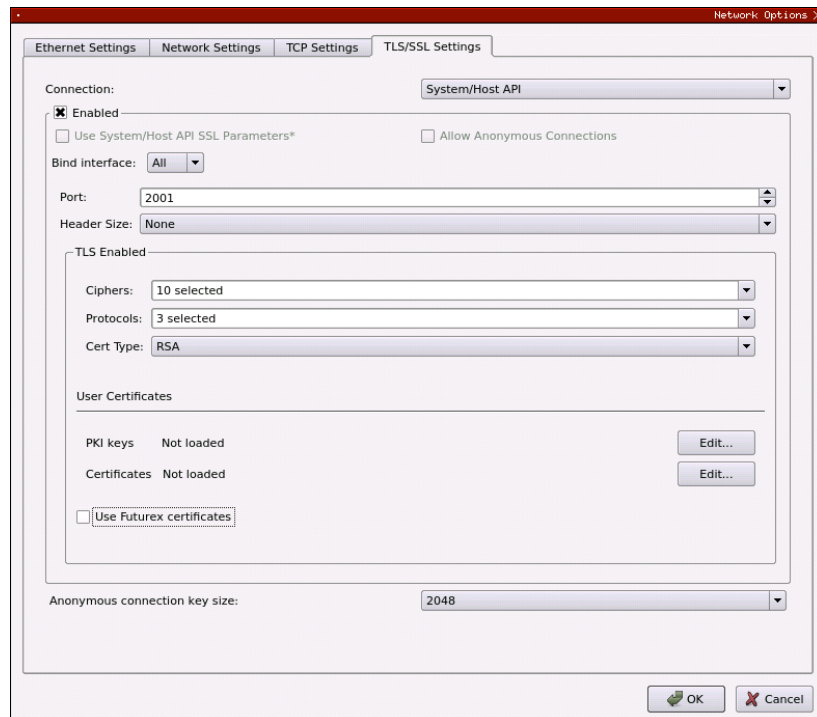
- In the **V3 Extensions** tab, select **Certificate Authority** in the **Profile** dropdown, then click [**OK**]. The *System TLS CA Root* certificate is now listed inside the previously created Certificate Container.



Name	Notes	Status	Owner Group
System TLS CA	X.509 Certificate Container		Admin Group
System TLS CA Root	Self-signed	Valid	Admin Group

[4.3.2] Generate a CSR for the System/Host API connection pair

- Go to **Administration > Configuration > Network Options**.
- In the **Network Options** dialog, select the **TLS/SSL Settings** tab.
- Under the **System/Host API** connection pair, uncheck the **Use Futurex certificates** box, then click [**Edit...**] next to **PKI keys** in the **User Certificates** section.

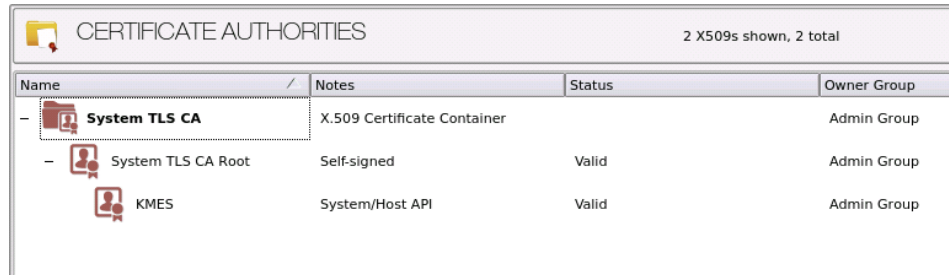


- In the **Application Public Keys** dialog, click [**Generate...**].
- There will be a warning stating that SSL will not be functional until new certificates are imported. Select [**Yes**] if you wish to continue.
- In the **PKI Parameters** dialog, leave the default settings and click [**OK**].
- It should show that a PKI Key Pair is loaded now in the **Application Public Keys** dialog. Now, select [**Request...**].
- In the **Subject DN** tab, set a **Common Name** for the certificate, such as *KMES*.

9. In the **Basic Info** tab, leave the default settings.
10. In the **V3 Extensions** tab, select **TLS Server Certificate** in the **Profile** dropdown.
11. In the **PKCS #10 Info** tab, click [**Browse...**], select a save location for the CSR, specify a name for the file, then click [**Open**].
12. Select [**OK**] to finish generating the CSR. There should be a message stating that the certificate signing request was successfully written to the file location that was selected. Click [**OK**].
13. Click [**OK**] again to save the **Application Public Keys** settings.
14. In the main **Network Options** dialog, it should now say **Loaded** next to **PKI keys**. Select [**OK**].

[4.3.3] Sign the System/Host API CSR

1. Navigate to the **PKI > Certificate Authorities** menu.
2. Right-click on the *System TLS CA Root* certificate, then select **Add Certificate > From Request...**
3. In the file browser, find and select the CSR that was generated for the **System/Host API** connection pair, then click [**Open**].
4. Once loaded, none of the settings need to be modified for the certificate. Click [**OK**].
5. The signed *KMES* TLS certificate will now show under the *System TLS CA Root* certificate in the **Certificate Authorities** menu.



The screenshot shows a window titled "CERTIFICATE AUTHORITIES" with a subtitle "2 X509s shown, 2 total". It contains a table with the following data:

Name	Notes	Status	Owner Group
- System TLS CA	X.509 Certificate Container		Admin Group
- System TLS CA Root	Self-signed	Valid	Admin Group
KMES	System/Host API	Valid	Admin Group

[4.3.4] Export the System TLS CA Root certificate

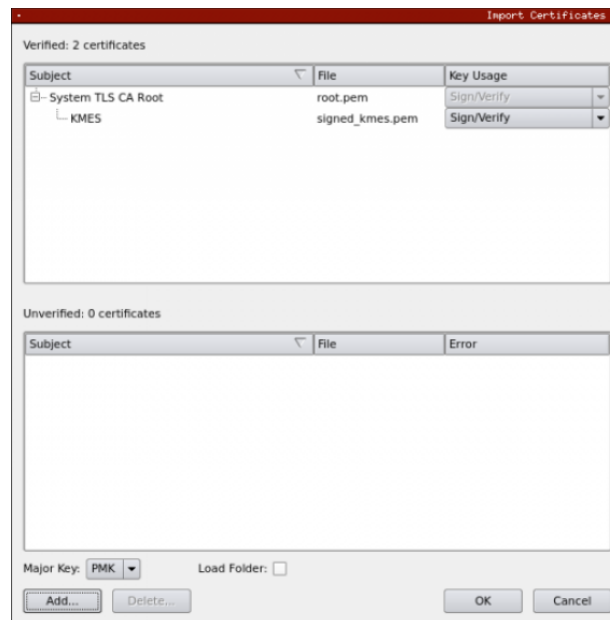
1. Navigate to the **PKI > Certificate Authorities** menu.
2. Right-click on the **System TLS CA Root** certificate, then select **Export > Certificate(s)...**
3. In the **Export Certificate** dialog, select **PEM** in the **Encoding** dropdown, then click [**Browse...**]
4. In the file browser, navigate to the location where you want to save the **System TLS CA Root** certificate, specify a name for the file, then click [**Open**].
5. Click [**OK**]. A message box will pop up stating that the PEM file was successfully written to the location that you specified. Click [**OK**] again to exit the dialog.

[4.3.5] Export the signed System/Host API TLS certificate

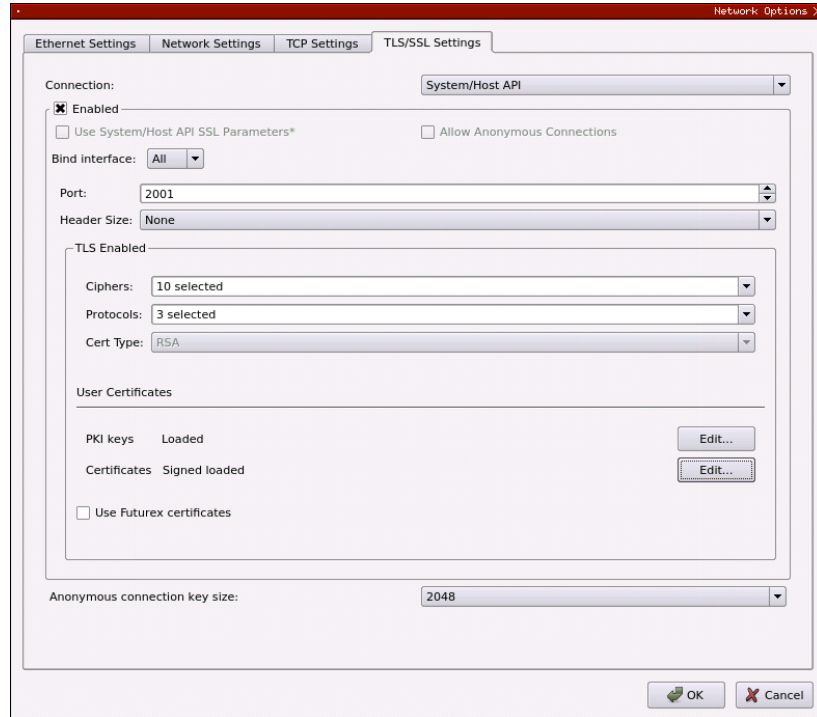
1. Navigate to the **PKI > Certificate Authorities** menu.
2. Right-click on the *KMES* certificate, then select **Export > Certificate(s)...**
3. In the **Export Certificate** dialog, select **PEM** in the **Encoding** dropdown, then click [**Browse...**]
4. In the file browser, navigate to the location where you want to save the signed *KMES* TLS certificate, specify a name for the file, then click [**Open**].
5. Click [**OK**]. A message box will pop up stating that the PEM file was successfully written to the location that you specified. Click [**OK**] again to exit the dialog.

[4.3.6] Load the exported TLS certificates into the System/Host API connection pair

1. Go to **Administration > Configuration > Network Options**.
2. In the **Network Options** dialog, select the **TLS/SSL Settings** tab.
3. Click [**Edit...**] next to **Certificates** in the **User Certificates** section.
4. Right-click on the **System/Host API SSL CA X.509 Certificate Container** and select **Import...**
5. Click [**Add...**] at the bottom of the **Import Certificates** dialog.
6. In the file browser, find and select both the *System TLS CA Root* certificate and the signed *KMES* certificate, then click [**Open**]. The certificate chain should appear in the **Verified** section, as shown below:



- Click [**OK**] to save the changes. In the **Network Options** dialog, the **System/Host API** connection pair should show **Signed loaded** next to **Certificates** in the **User Certificates** section, as shown below:



- Click [**OK**] to save and exit the **Network Options** dialog.

[4.3.7] Generate a TLS private key and CSR for the Futorex PKCS #11 (FXPKCS11) library using OpenSSL

Note: The commands in this section need to be run from a terminal application that has OpenSSL installed.

- Open a terminal and run the following command to generate a TLS private key for the FXPKCS11 library:

```
$ openssl genrsa -out fxpkcs11_tls_privatekey.pem 2048
```

The command will output the private key to a file named `fxpkcs11_tls_privatekey.pem` in the same directory from where the command was run.

- Run the following command to generate a Certificate Signing Request (CSR) for the FXPKCS11 library:

```
$ openssl req -new -key fxpkcs11_tls_privatekey.pem -out fxpkcs11_tls_cert_req.pem -days 365
```

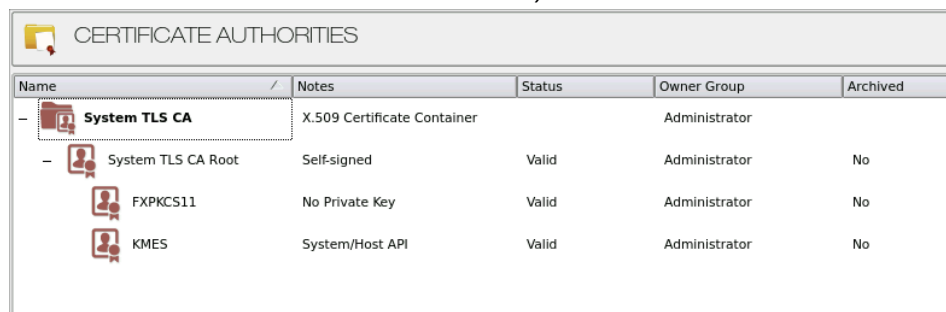
It will prompt you to enter certificate information. Set the default value for every field by pressing the **Enter** key at every prompt.

The command will output the CSR to a file named `fxpkcs11_tls_cert_req.pem` in the same directory from where the command was run.

- Move or copy the CSR file (i.e., `fxpkcs11_tls_cert_req.pem`) to the storage medium configured on the KMES.

[4.3.8] Sign the Certificate Signing Request (CSR) for the FXPKCS11 library

1. Navigate to the **PKI > Certificate Authorities** menu.
2. Right-click on the *System TLS CA Root* certificate and select **Add Certificate > From Request...**
3. In the file browser, find and select the FXPKCS11 CSR (i.e., `tls_cert_req.pem`). Certificate information will populate in the **Create X.509 From CSR** window.
4. In the **Subject DN** tab, select **Classic** in the **Preset** dropdown, then set a **Common Name** for the certificate, such as *FXPKCS11*.
5. In the **Basic Info** tab, leave the default settings.
6. In the **V3 Extensions** tab, select the **TLS Client Certificate** profile, then click **[OK]**.
7. The signed *FXPKCS11* certificate is now listed under the *System TLS CA Root* certificate.



Name	Notes	Status	Owner Group	Archived
System TLS CA	X.509 Certificate Container		Administrator	
System TLS CA Root	Self-signed	Valid	Administrator	No
FXPKCS11	No Private Key	Valid	Administrator	No
KMES	System/Host API	Valid	Administrator	No

[4.3.9] Export the signed FXPKCS11 TLS certificate

1. Navigate to the **PKI > Certificate Authorities** menu.
2. Right-click on the *FXPKCS11* certificate and select **Export > Certificate(s)...**
3. In the **Export Certificate** dialog, select **PEM** in the **Encoding** dropdown, then click **[Browse...]**
4. In the file browser, navigate to the location where you want to save the *FXPKCS11* TLS certificate, specify a name for the file, then click **[Open]**.
5. Click **[OK]**. A message box will pop up stating that the PEM file was successfully written to the location that you specified. Click **[OK]** again to exit the dialog.

Note: The signed *FXPKCS11* TLS certificate and the *System TLS CA Root* certificate need to be copied to the computer that will be running the Red Hat Certificate System instance. In the next section, they will be configured in the FXPKCS11 configuration file and used for TLS communication with the KMES Series 3.

[5] EDIT THE FUTUREX PKCS #11 CONFIGURATION FILE

The Futurex PKCS #11 configuration file (i.e., `fxpkcs11.cfg`) is used by the Futurex PKCS #11 library to connect to the KMES Series 3. It enables the user to modify certain configurations and set connection details. This section covers the **<KMS>** portion of the `FXPKCS11` config file, where the connection details are set.

Note: By default, the `FXPKCS11` library looks for the configuration file at `C:\Program Files\Futurex\fxpkcs11\fxpkcs11.cfg` for Windows and `/etc/fxpkcs11.cfg` for Linux. Alternatively, the `FXPKCS11_CFG` environment variable can be set to the location of the `fxpkcs11.cfg` file.

Open the `fxpkcs11.cfg` file in a text editor as an administrator and edit it accordingly.

```
<KMS>
# Which PKCS11 slot
<SLOT>          0          </SLOT>
<LABEL>        Futurex    </LABEL>

# Login username
<CRYPTO-OPR>    [identity_name] </CRYPTO-OPR>
# Automatically login on session open
#<CRYPTO-OPR-PASS> [identity_password] </CRYPTO-OPR-PASS>

# Key group name
<KEYGROUP-NAME> keygroup1    </KEYGROUP-NAME>

# Asymmetric key group name
<ASYM-KEYGROUP-NAME> ASYM-RHCS </ASYM-KEYGROUP-NAME>

# Connection information
<ADDRESS>      10.0.8.20      </ADDRESS>
<PROD-PORT>    2001          </PROD-PORT>
<PROD-TLS-ENABLED> YES      </PROD-TLS-ENABLED>
<PROD-TLS-ANONYMOUS> NO      </PROD-TLS-ANONYMOUS>
<PROD-TLS-CA>  /connection_certs/root.pem </PROD-TLS-CA>
<PROD-TLS-CERT> /connection_certs/fxpkcs11_tls_cert.pem </PROD-TLS-CERT>
<PROD-TLS-KEY> /connection_certs/fxpkcs11_tls_privatekey.pem </PROD-TLS-KEY>
# <PROD-TLS-KEY-PASS> safest </PROD-TLS-KEY-PASS>

# YES = This is communicating through a Guardian
<FX-LOAD-BALANCE> NO </FX-LOAD-BALANCE>
</KMS>
```

The **<SLOT>** and **<LABEL>** fields specify PKCS11 slot 0 and the label *Futurex*.

The **<CRYPTO-OPR>** field specifies the name of the identity that you created on the KMES.

The **<CRYPTO-OPR-PASS>** field specifies the password of the identity configured in the **<CRYPTO-OPR>** field. This can be used to log the application into the KMES automatically, if required.

If an application creates symmetric keys on the KMES, the value specified in the **<KEYGROUP-NAME>** field is the name of the symmetric key group that the keys will be added to.

If an application creates asymmetric keys on the KMES, the value specified in the **<ASYM-KEYGROUP-NAME>** field is the name of the asymmetric key group that keys will be added to.

The **<ADDRESS>** field specifies the IP address of the KMES that the `FXPKCS11` library should connect to.

The **<PROD-PORT>** field specifies for the FXPKCS11 library to connect to the System/Host API port on the KMES.

The **<PROD-TLS-ANONYMOUS>** field defines whether the FXPKCS11 library authenticates to the server.

The **<PROD-TLS-KEY>** field defines the location of the client private key. Supported formats for the TLS private key are PKCS #1 clear private keys, PKCS #8 encrypted private keys, or a PKCS #12 file that contains the private key and certificates encrypted under the password specified in the **<PROD-TLS-KEY-PASS>** field.

Because a PKCS #12 file is defined in the **<PROD-TLS-KEY>** field in this example, the signed client cert does not need to be defined with the **<PROD-TLS-CERT>** tag, nor do the CA cert/s need to be defined with one or more instances of the **<PROD-TLS-CA>** tag.

If you use a Guardian to manage multiple KMES devices in a cluster, define the **<FX-LOAD-BALANCE>** field as *YES*. Otherwise, set it to *NO*.

After you finish editing the `fxpkcs11.cfg` file, run the `PKCS11Manager` file to test the connection against the KMES and check the `fxpkcs11.log` for errors and information. For more information, refer to the Futurex PKCS #11 technical reference found on the Futurex Portal.

[5.1] SPECIAL DEFINES REQUIRED FOR THIS INTEGRATION

For the Red Hat Certificate System integration, the following two defines must be added to the **<CONFIG>** section of the FXPKCS11 configuration file:

```
<FORCED-ASYMMETRIC-USAGE>  SIGN | VERIFY      </FORCED-ASYMMETRIC-USAGE>
<CHECK-ALREADY-LOGGED-IN>  NO                </CHECK-ALREADY-LOGGED-IN>
```

[6] RED HAT CERTIFICATE SYSTEM INSTALLATION AND SUBSYSTEM DEPLOYMENT

This section outlines the basic installation method for Red Hat Certificate System (RHCS). It is assumed that you already have installed Red Hat Enterprise Linux (RHEL), the system is subscribed to the Red Hat subscription management service, the Red Hat Certificate System subscription is attached, and the required repositories are enabled. Please refer to the RHCS [Get Started](#) article for instructions on how to perform the above actions.

[6.1] INSTALL RHCS AND ITS PREREQUISITES

1. RHCS requires Red Hat Directory Server, which serves as an internal repository for certificate requests, certificates, etc. Install the directory server packages using the following command:

```
# sudo yum install redhat-ds
```

2. Run the directory server installation script, selecting the defaults or customizing as desired:

```
# sudo /usr/sbin/setup-ds-admin.pl
```

3. By default, Red Hat Directory Server does not automatically run on system startup. Run the following command to ensure that the directory server starts automatically if the computer is rebooted.

```
$ sudo systemctl enable dirsrv.target
```

4. Install the certificate system packages:

```
# sudo yum install redhat-pki
```

[6.2] MODIFY SELINUX TO SUPPORT SUBSYSTEM DEPLOYMENT USING AN HSM

If you want to deploy an RHCS subsystem using a Hardware Security Module (HSM) and SELinux is running in enforcing mode, certain SELinux and firewalld settings must be manually updated before deploying the subsystem. The following section describes the required actions.

1. Run the following commands to reset the context of the `fxpkcs11.cfg` file and the main `fxpkcs11` directory:

```
# sudo /sbin/restorecon -v /etc/fxpkcs11.cfg
# sudo /sbin/restorecon -R /usr/local/bin/fxpkcs11/
```

Note: Modify the paths to match the locations of the `fxpkcs11.cfg` file and the main `fxpkcs11` directory on your system.

2. Run the following commands to allow outbound connections to TCP port 2001 (i.e., the System/Host API port on the KMES):

```
# sudo semanage port -m -t http_port_t -p tcp 2001
```

[6.3] RUN THE PKISPAWN SCRIPT TO CREATE AND CONFIGURE A SUBSYSTEM INSTANCE

The `pkispawn` command line tool is used to install and configure a new PKI instance. It eliminates the need for separate installation and configuration steps, and may be run either interactively, as a batch process, or a combination of both (batch process with prompts for passwords). Refer to the `pkispawn` man page for detailed information about all supported options by running "man pkispawn".

The `pkispawn` command reads in its default installation and configuration values from a plain text configuration file (`/etc/pki/default.cfg`). This file consists of name=value pairs divided into [DEFAULT], [Tomcat], [CA], [KRA], [OCSP], [TKS], and [TPS] sections.

Note: It is strongly recommended that you read the [full documentation](#) to understand the purpose of every parameter in the `/etc/pki/default.cfg` file. This will allow you to customize your PKI environment to your specific needs.

Red Hat's recommended procedure for spawning a subsystem that uses an HSM is to create an override configuration file that contains only the parameters necessary for using the HSM as its token. Any parameter settings in this file will override the parameter settings in the `default.cfg` file.

Any of the various RHCS subsystems (CA, KRA, OCSP, TKS, TPS) can be spawned to use the HSM, but this integration guide will focus solely on the Certificate Authority (CA) for brevity.

Prepare an override configuration file with required HSM parameters

1. In a terminal, navigate to the directory where the Futurex PKCS #11 module is installed on your system (e.g., `/usr/local/bin/fixpkcs11`).
2. Run the following Vim command as `sudo`:

```
# sudo vim default_futurex.txt
```

The following is an example override file that can be used for spawning a CA subsystem with the KMES:

Note: All values contained within angle brackets need to be set to a specific value by the user. All other values should be set exactly as shown.

Note: The `pki_ds_password` value must match the password set for the directory manager when Red Hat Directory Server was installed.

```
[DEFAULT]
#####
# Provide HSM parameters #
#####
pki_hsm_enable=True
pki_hsm_libfile=<path_to_fixpkcs11_libfile>
pki_hsm_modulename=FixPKCS11
pki_token_name=Futurex
pki_token_password=<hsm_identity_password>

#####
# Provide PKI-specific HSM token names #
#####
pki_audit_signing_token=Futurex
```



```

pki_ssl_server_token=Futurex
pki_subsystem_token=Futurex

#####
# Provide PKI-specific passwords #
#####
pki_admin_password=<pki_admin_password>
pki_client_pkcs12_password=<pki_client_pkcs12_password>
pki_ds_password=<pki_ds_password>

#####
# Provide non-CA-specific passwords #
#####
pki_client_database_password=<pki_client_database_password>

[CA]
#####
# Provide CA-specific HSM token names #
#####
pki_ca_signing_token=Futurex
pki_ocsp_signing_token=Futurex

```

After you have finished editing, save the file.

Run the pkispawn utility

1. In a terminal, run the following command to deploy a CA subsystem using the KMES Series 3:

Note: The full path to the default_futurex.txt file is required if you are not running the command from the same directory where default_futurex.txt is saved.

```
# sudo pkispawn -s CA -f default_futurex.txt -vvv
```

If the deployment is successful, an installation summary similar to the following will be presented after the command completes:

```

=====
                        INSTALLATION SUMMARY
=====

Administrator's username:          caadmin
Administrator's PKCS #12 file:
    /root/.dogtag/pki-tomcat/ca_admin_cert.p12

To check the status of the subsystem:
    systemctl status pki-tomcatd@pki-tomcat.service

To restart the subsystem:
    systemctl restart pki-tomcatd@pki-tomcat.service

The URL for the subsystem is:
    https://localhost.localdomain:8443/ca

PKI instances will be enabled upon system boot

=====

```

Important: If the `pkispawn` command fails, you must do the following two things before re-attempting to run the `pkispawn` command:

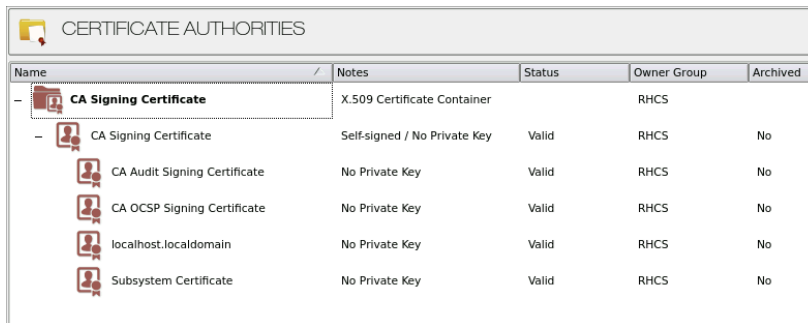
1. Log in to the KMES Series 3 application interface, navigate to **PKI > Certificate Authorities**, and confirm if a Certificate Container was created named *CA Signing Certificate*. If there was, you need to delete the *CA Signing Certificate* Certificate Container (which will also delete all certificates inside it) before running `pkispawn` again, or the command will fail.
2. Delete the partially created CA subsystem instance by running the `pkidestroy` command below:

```
$ sudo pkidestroy -s CA -i pki-tomcat
```

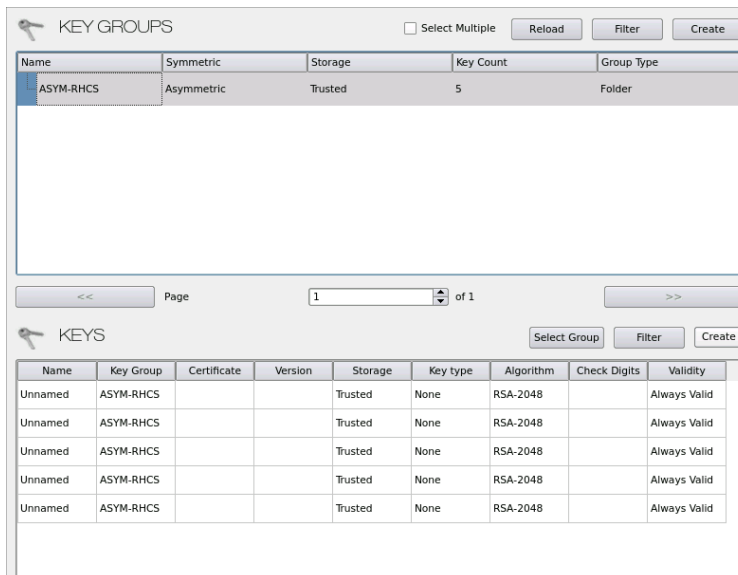
[6.4] VIEW THE KEYS AND CERTIFICATES THAT RHCS CREATED ON THE KMES SERIES 3

To view the keys and certificates that Red Hat Certificate System created on the KMES, perform the following steps:

1. Log in to the KMES Series 3 application interface with the default Admin identities.
2. Navigate to the **PKI > Certificate Authorities** menu. All of the certificates that RHCS created for the CA subsystem instance will be listed inside the *CA Signing Certificate* X.509 Certificate Container.



3. Navigate to the **Key Management > Keys** menu and select the *ASYM-RHCS* asymmetric key group. In the **Keys** section, you will see the private keys of the certificates shown in step 2 above.



[6.5] IMPORT THE CA ADMINISTRATOR PKCS #12 FILE INTO THE BROWSER

NOTE: The following steps were completed using a Firefox web browser. There may be some differences in the actions taken when using a different browser, but the overall intent of the process will be the same.

1. In Firefox, navigate to *Settings* -> *Privacy & Security* -> *Certificates* and click the **View Certificates** button.
2. Under the *Your Certificates* tab, select **Import** to import the CA Administrator PKCS #12 file (i.e., `ca_admin_cert.p12`). When it prompts for a password, enter the value that was configured for the `pki_client_pkcs12_password` define in the `default_futurex.txt` file in section 6.3.

Note: The location of the `ca_admin_cert.p12` file was included in the installation summary for the CA subsystem deployment.

[6.6] ACCESSING THE NEW CA SUBSYSTEM IN THE BROWSER

1. Access the Red Hat Certificate System subsystem console by navigating to the URL below:

<https://<fully qualified domain name>:8443/pki/ui/>



Note: When submitting Certificate Signing Requests (CSRs) in Red Hat Certificate System, the **Common Name** and **UID** fields are both required. If you submit a request with only the **Common Name** field completed, the request will fail, and you will receive an error stating that the **Subject Name** does not match.

This completes the Red Hat Certificate System integration with the Futurex KMES Series 3. All CA subsystem keys are secured within the internal HSM of the KMES and are available to Red Hat Certificate System when required.

APPENDIX A: XCEPTIONAL SUPPORT



In today's high-paced environment, we know you are looking for timely and effective resolutions for your mission-critical needs. That is why our Xceptional Support Team does whatever it takes to ensure you have the best experience and support possible. Every time. Guaranteed.

- 24x7x365 mission critical support
- Level 1 to level 3 support
- Extremely knowledgeable subject matter experts

At Futurex, we strive to supply you with the latest data encryption innovations as well as our best-in-class support services. Our Xceptional Support Team goes above and beyond to meet your needs and provide you with exclusive services that you cannot find anywhere else in the industry.

- Technical Services
- Onsite Training
- Virtual Training
- Customized Consulting
- Customized Software Solutions
- Secure Key Generation, Printing, and Mailing
- Remote Key Injection
- Certificate Authority Services

Toll-Free: 1-800-251-5112

E-mail: support@futurex.com



ENGINEERING CAMPUS

864 Old Boerne Road
Bulverde, Texas, USA 78163
Phone: +1 830-980-9782
+1 830-438-8782
E-mail: info@futurex.com

EXCEPTIONAL SUPPORT

24x7x365
Toll-Free: 1-800-251-5112
E-mail: support@futurex.com

SOLUTIONS ARCHITECT

E-mail: solutions@futurex.com