AXWAY VA (VALIDATION AUTHORITY) SERVER

Integration Guide

Applicable Devices:
Vectera Plus

TABLE OF CONTENTS

[1] DOCUMENT INFORMATION

[1.1] DOCUMENT OVERVIEW

The purpose of this document is to provide information regarding the configuration of Futurex HSMs with Axway VA (Validation Authority) using PKCS #11 libraries. For additional questions related to your HSM, see the relevant administrator's guide.

[1.2] APPLICATION DESCRIPTION

The Validation Authority Server (VA Server) product ensures the integrity and validity of online transactions by delivering real-time validation of digital certificates issued by any Certification Authority (CA). It is a robust server application used for enabling the most widely used secure Internet applications to validate digital certificates.

VA Server is comprised of a VA validation server acting as either a Repeater or Responder operating on a Windows or Linux platform, and a web-based VA administration server that provides centralized management of your validation processing components through an admin UI.

Based on its server license, VA Server can be set up to operate as either a Responder or a Repeater. As a Responder, VA Server also offers support for hardware security modules (HSM), a critical component that is designed to provide the highest level of security and performance for protected key storage, high-speed signatures and hardware key generation.

The VA Server maintains a store of digital certificate revocation data by obtaining the issuing CA Certificate Revocation List (CRL), a cumulative list of revoked certificates.

The VA Server is CA neutral, and supports multiple CAs, several different trust models, and CA specific validation policies. To validate a digital certificate, a client application can query the VA Server rather than having to perform the cumbersome task of obtaining and processing the entire CRL every time it encounters a digital certificate. Client applications can query the VA Server utilizing open standard protocols, including the Online Certificate Status Protocol (OCSP) defined by RFC 6960 (formerly 2560) and the Server-Based Certificate Validation Protocol (SCVP) defined by RFC 5055. Clients can use SCVP to delegate the entire certificate validation operation, including path construction and intermediate CA validation, to the VA Server. VA Server supports multiple validation policies that clients can reference in a SCVP request to specify authentication and authorization requirements.

[1.3] COPYRIGHT AND TRADEMARK NOTICES

Futurex makes no warranty of any kind with regard to this information, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Futurex shall not be liable for errors contained herein or for incidental or consequential damages concerned with the furnishing, performance, or use of this material.

[1.4] TERMS OF USE

This integration guide, as well as the software and/or products described in it, are furnished under agreement with Futurex and may be used only in accordance with the terms of such agreement. Except as permitted by such agreement, no part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, recording, or otherwise, without prior written permission of Futurex.

[2] OUR STORY

For over 40 years, Futurex has been a globally recognized provider of scalable, versatile, and secure data protection solutions for organizations worldwide. More than 15,000 customers have trusted Futurex's innovative Hardened Enterprise Security Platform to provide market-leading solutions for the secure encryption, storage, transmission, and certification of sensitive data. Futurex maintains an unyielding commitment to offering advanced, standards-compliant solutions, including:

- Hardware security modules for cryptographic data processing
- Enterprise key, certificate, and token lifecycle management
- Remote key management and injection platforms
- Secure, hand-held devices for configuration, management, and compliant key loading
- High availability solutions for centralized configuration, management, monitoring, load balancing, and disaster recovery
- Secure storage and access of sensitive data
- Customizable data encryption solutions that meet users' specific needs

In understanding the diverse needs of our customers, we actively maintain and develop our expertise across multiple disciplines including hardware design and development, software and firmware engineering, regulatory compliance and certification, enterprise architecture design, and technical support. This drives our success and enables us to reach organizations of every size and industry. The cryptographic environments developed by our Solutions Architects incorporate Futurex technology and VirtuCrypt cloud-based services exclusively, with zero reliance on third-party software or hardware. By directly overseeing all aspects of development and production of our technology, we maintain the agility and knowledge necessary to support complex customer environments where solutions grow alongside their business.

Throughout every facet of our organization, we maintain a focus on providing exceptional customer service, best-in-class technology, and effective solutions for our customers. The continuous expansion of our innovative products and services exhibits our dedication to meeting the growing business needs of our global customers and partners. Through our results-oriented engineering culture, we have provided organizations worldwide with custom solutions supporting aggressive times to market.

Our products satisfy the most rigorous security requirements, proving our unyielding dedication to the standards-based security of our enterprise-class solutions. As we move forward, Futurex will continue to be a global leader in the data security and electronic transaction industries by maintaining high performance standards, providing quality service, and expanding our best-in-class product suite.

[3] PREREQUISITES

Supported Futurex Hardware:

- Vectera Plus, 6.7.x.x and above

Supported Operating Systems:

- Microsoft Windows Server 2012 R2, 2016 and 2019 with latest Service Pack applied
- Red Hat Enterprise Linux (RHEL) 7.x

Resource Recommendations:

- Processor: 3.0 GHz, quad core/CPU
- Memory: 16 GB
- Disk Space: 500 GB

Other:

- OpenSSL
- Validation Authority Server Installer

[4] INSTALL FUTUREX PKCS #11 (FXPKCS11)

In a Windows environment, the easiest way to install the PKCS #11 module is by using **FXTools**. FXTools can be downloaded from the Futurex Portal. In a Linux environment, you need to download a tarball of the PKCS #11 binaries from the Futurex Portal. Then, extract the *.tar* file locally where you want the application to be installed in your file system. Step by step installation instructions for both of these scenarios is provided in the following subsections.

[4.1] INSTRUCTIONS FOR INSTALLING THE PKCS #11 MODULE USING FXTOOLS IN WINDOWS

- Run the FXTools installer as an administrator

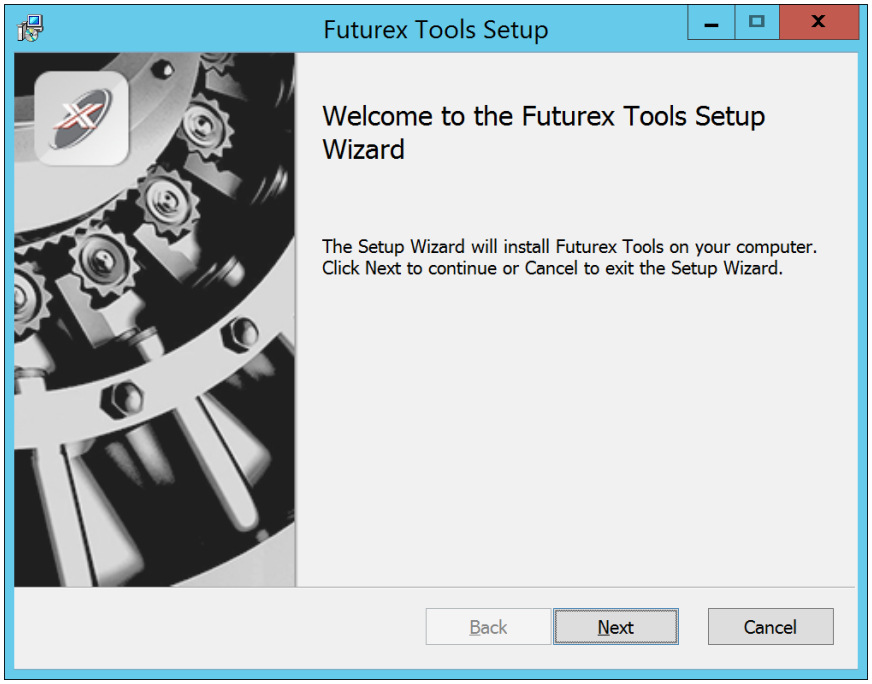


FIGURE: FUTUREX TOOLS SETUP WIZARD

By default, all tools are installed on the system. A user can overwrite and choose not to install certain modules.

- **Futurex Client Tools –** Command Line Interface (CLI) and associated SDK for both Java and C.
- **Futurex CNG Module –** The Microsoft Next Generation Cryptographic Library.
- **Futurex Cryptographic Service Provider (CSP) –** The legacy Microsoft cryptographic library.
- **Futurex EKM Module –** The Microsoft Enterprise Key Management library.
- **Futurex PKCS #11 Module –** The Futurex PKCS #11 library and associated tools.
- **Futurex Secure Access Client –** The client used to connect a Futurex Excrypt Touch to a local laptop, via USB, and a remote Futurex device.

After starting the installation, all noted services are installed. If the Futurex Secure Access Client was selected, the Futurex Excrypt Touch driver will also be installed (Note this sometimes will start minimized or in the background).

After installation is complete, all services are installed in the "*C:\Program Files\Futurex*" directory. The CNG Module, CSP Module, EKM Module, and PKCS #11 Module all require configuration files, located in their corresponding directory with a *.cfg* extension. In addition, the CNG and CSP Modules are registered in the Windows Registry (*HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Cryptography\Defaults\Provider*) and are installed in the "*C:\Windows\System32*" directory.

[4.2] INSTRUCTIONS FOR INSTALLING THE PKCS #11 MODULE IN LINUX

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

NOTE: For the Futurex PKCS #11 module to be accessible system-wide, it would need to be placed into */usr/local/bin* by an administrative user. If the module only needs to be utilized by the current user, then installing into *$HOME/bin* would be the appropriate location.

The extracted content of the *.tar* file is a single *fxpkcs11* directory. Inside of the *fxpkcs11* directory are the following files and directories (Only files/folders that are relevant to the installation process are included below):

- *fxpkcs11.cfg* -> PKCS #11 configuration file
- *x86/* - This folder contains the module files for 32-bit architecture
- *x64/* - This folder contains the module files for 64-bit architecture

Within the *x86* and *x64* directories are two directories. One named *OpenSSL-1.0.x* and the other named *OpenSSL-1.1.x*. Both of these OpenSSL directories contain the PKCS #11 module files, built with the respective OpenSSL versions. These files are listed below, with short descriptions of each:

- *configTest* -> Program to test configuration and connection to the HSM
- *libfxpkcs11.so* -> PKCS #11 Library File
- *PKCS11Manager* -> Program to test connection and manage the HSM through the PKCS #11 library

The *configTest* and *PKCS11Manager* programs look for the *fxpkcs11.cfg* file at the following path:

```
/etc/fxpkcs11.cfg
```

Because of this, it is necessary either to move the *fxpkcs11.cfg* file from the */usr/local/bin/fxpkcs11* directory to the */etc* directory, or to set the FXPKCS11_CFG environment variable to point to the *fxpkcs11.cfg* file.

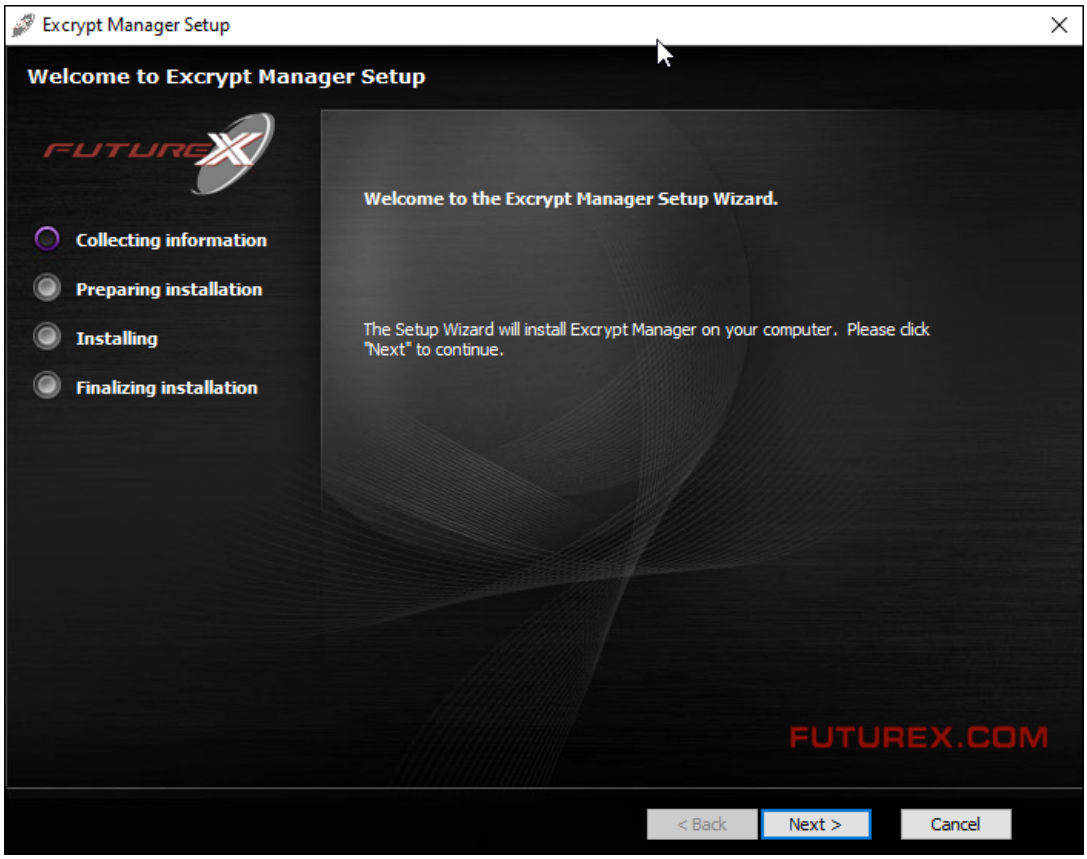
[5] INSTALL EXCRYPT MANAGER (IF USING WINDOWS)

Excrypt Manager is a Windows application that can be used to configure the HSM in subsequent sections. HSM configuration can also be completed using FXCLI, the Excrypt Touch, or the Guardian Series 3. For more information about using these tools/devices to configure the HSM, please see the relevant Administrator's Guide.

NOTE: If you plan to use a Virtual HSM for the integration, all configurations will need to be performed using either FXCLI, the Excrypt Touch, or the Guardian Series 3.

NOTE: The Excrypt Manager version must be from the 4.4.x branch or later to be compatible with the HSM firmware, which must be 6.7.x.x or later.

- Run the Excrypt Manager installer as an administrator.



The installation wizard will ask you to specify where you want Excrypt Manager to be installed. The default location is "*C:\Program Files\Futurex\Excrypt Manager*". Once that is done click "Install".

[6] INSTALL FUTUREX COMMAND LINE INTERFACE (FXCLI)

NOTE: Windows users can skip this step because FXCLI was included with the FXTools installation.

[6.1] INSTRUCTIONS FOR INSTALLING FXCLI IN LINUX

NOTE: These instructions are for Ubuntu-based Linux distributions. For instructions on how to install FXCLI on other Linux distributions, such as Debian or Red Hat, please see the relevant Administrator's guide.

Download the FXCLI module

The user must download the correct *.deb* package files from the Futurex Portal.

Below is the full list of *.deb* files for Ubuntu/Debian-based Linux distributions:

- fxcl-1.4.1-linux-amd64-ssl1.0-cli-fxparse.deb
- fxcl-1.4.1-linux-amd64-ssl1.0-cli-hsm.deb
- fxcl-1.4.1-linux-amd64-ssl1.0-cli-kmes.deb
- fxcl-1.4.1-linux-amd64-ssl1.0-devel.deb
- fxcl-1.4.1-linux-amd64-ssl1.0-java.deb
- fxcl-1.4.1-linux-amd64-ssl1.1-cli-fxparse.deb
- fxcl-1.4.1-linux-amd64-ssl1.1-cli-hsm.deb
- fxcl-1.4.1-linux-amd64-ssl1.1-cli-kmes.deb
- fxcl-1.4.1-linux-amd64-ssl1.1-devel.deb
- fxcl-1.4.1-linux-amd64-ssl1.1-java.deb
- fxcl-1.4.1-linux-i386-ssl1.0-cli-fxparse.deb
- fxcl-1.4.1-linux-i386-ssl1.0-cli-hsm.deb
- fxcl-1.4.1-linux-i386-ssl1.0-cli-kmes.deb
- fxcl-1.4.1-linux-i386-ssl1.0-devel.deb
- fxcl-1.4.1-linux-i386-ssl1.0-java.deb
- fxcl-1.4.1-linux-i386-ssl1.1-cli-fxparse.deb
- fxcl-1.4.1-linux-i386-ssl1.1-cli-hsm.deb
- fxcl-1.4.1-linux-i386-ssl1.1-cli-kmes.deb
- fxcl-1.4.1-linux-i386-ssl1.1-devel.deb
- fxcl-1.4.1-linux-i386-ssl1.1-java.deb

If the system is **64-bit**, users should select from the files marked **amd64**. If the system is **32-bit**, users should select from the files marked **i386**.

If running an OpenSSL version in the **1.0.x** branch, users should select from the files marked **ssl1.0**. If running an OpenSSL version in the **1.1.x** branch, users should select from the files marked **ssl1.1**.

Additionally, users can install the packages based on the desired features they wish to install. For example, if your cryptographic infrastructure does not have a KMES Series 3 device, it would not be necessary to download the files for **cli-kmes**.

Futurex offers the following features for FXCLI:

- Java Software Development Kit (**java**)
- HSM command line interface (**cli-hsm**)
- KMES command line interface (**cli-kmes**)
- Software Development Kit headers (**devel**)
- YAML parser used to parse bash output (**cli-fxparse**)

Install FXCLI

To install *.deb* packages on a Linux system, use the **apt** command. The following example uses the *.deb* package for a computer with a 64-bit processor, running an OpenSSL version in the 1.0.x branch, to install cli-hsm. Once you have downloaded the *.deb* file that you wish to install from the Futurex Portal, run the following command in a terminal:

```
$ sudo dpkg -i fxcl-1.4.1-linux-amd64-ssl1.0-cli-hsm.deb
```

NOTE: After the installation is completed, system environment variables must be defined for the location of the FXCLI binaries. To do so permanently you must add the following two lines to your *.bashrc* file:

```
PATH=$PATH:/usr/bin/fxcli-hsm
PATH=$PATH:/usr/bin/fxcli-kmes
```

[7] CONFIGURE THE FUTUREX HSM

In order to establish a connection between the PKCS #11 library and the Futurex HSM, a few configuration items need to first be performed, which are the following:

NOTE: All of the steps in this section can be completed through either Excrypt Manager or FXCLI (if using a physical HSM rather than a virtual HSM). Optionally, steps 4 through 6 can be completed through the Guardian Series 3, which will be covered in Appendix A.

1. Connect to the HSM via the front USB port (NOTE: If you are using a virtual HSM for the integration you will have to connect to it over the network either via FXCLI, the Excrypt Touch, or the Guardian Series 3)
    a. Connecting via Excrypt Manager
    b. Connecting via FXCLI
2. Validate the correct features are enabled on the HSM
3. Setup the network configuration
4. Load the Futurex FTK
5. Configure a Transaction Processing connection and create a new Application Partition
6. Create a new Identity that has access to the Application Partition created in the previous step
7. Configure TLS Authentication. There are two options for this:
    a. Enabling server-side authentication
    b. Creating client certificates for mutual authentication

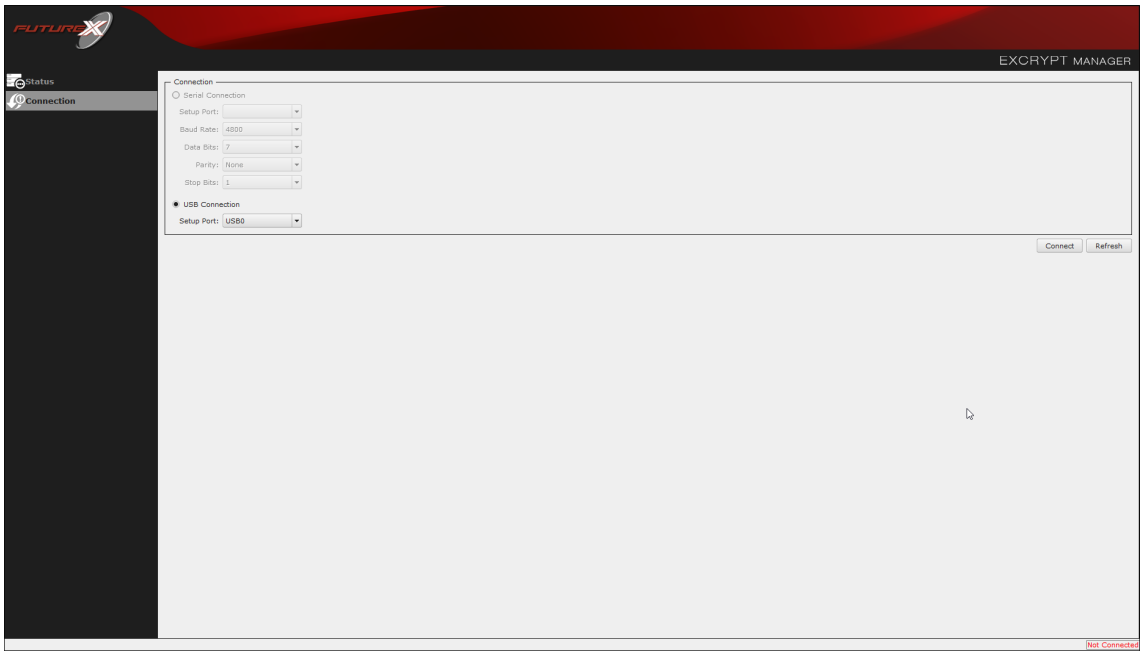
Each of these action items is detailed in the following subsections.

[7.1] CONNECT TO THE HSM VIA THE FRONT USB PORT

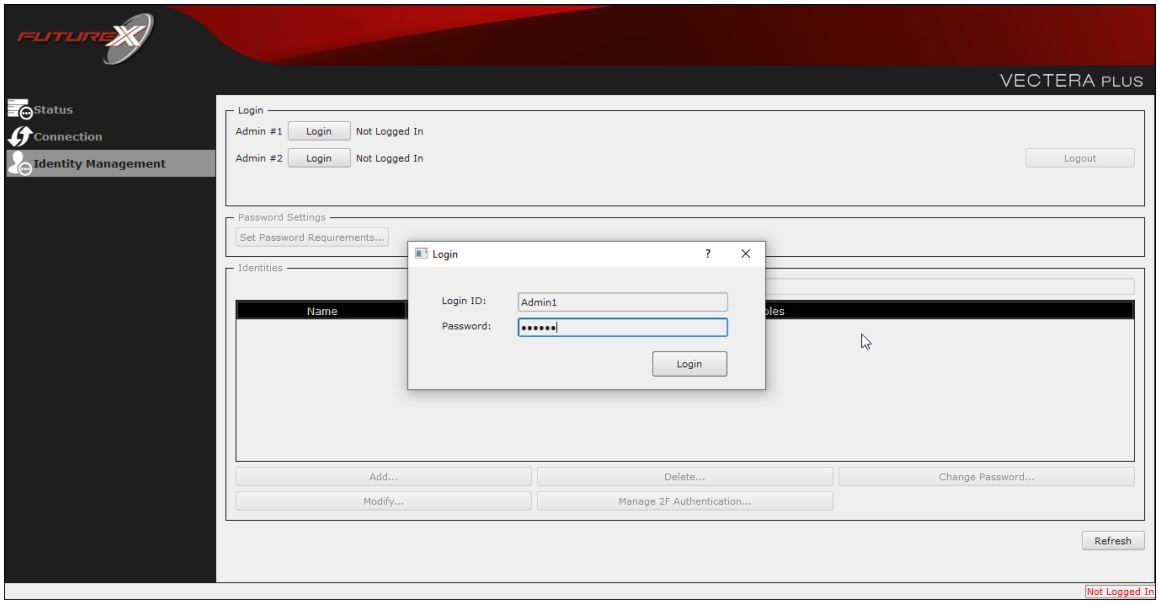
For both Excrypt Manager and FXCLI you need to connect your laptop to the front USB port on the HSM.

Connecting via Excrypt Manager

Open Excrypt Manager, click “Refresh” in the lower right-hand side of the Connection menu. Then select “USB Connection” and click “Connect”.

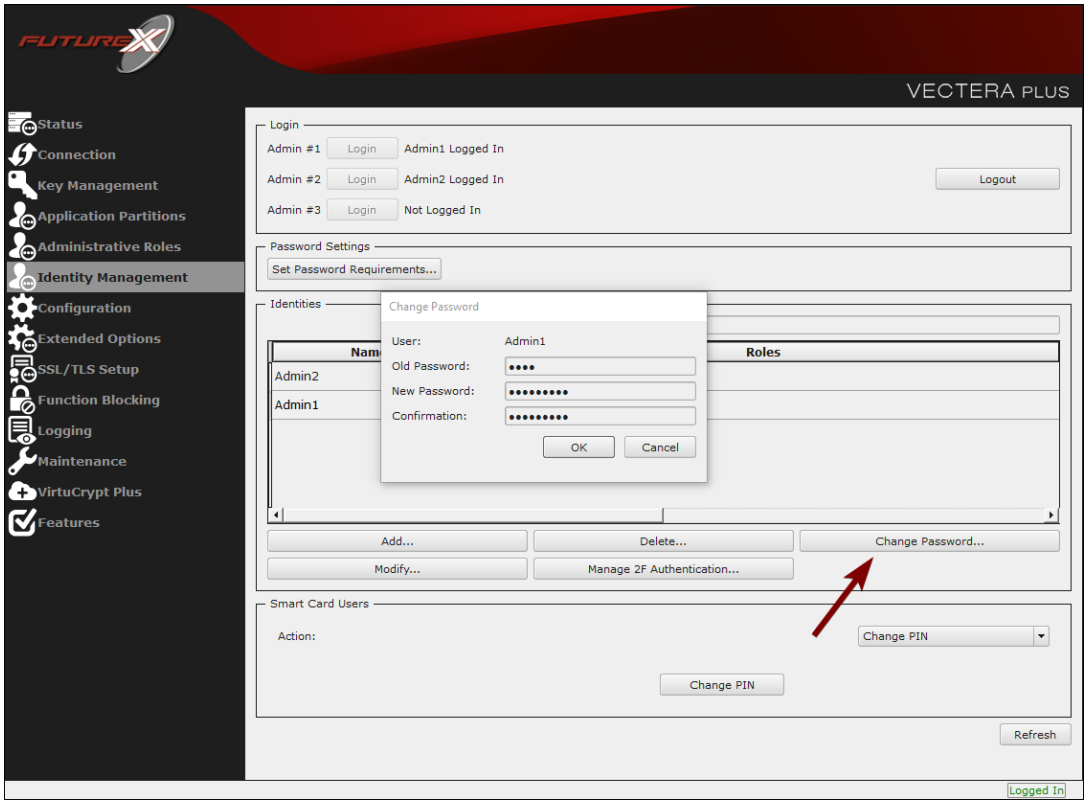


Login with both default Admin identities.



The default Admin passwords (i.e. “safe”) must be changed for both of your default Admin Identities (e.g. “Admin1” and “Admin2”) in order to load the major keys onto the HSM.

To do so via Excrypt Manager navigate to the Identity Management menu, select the first default Admin identity (e.g. “Admin1”), then click the “Change Password…” button. Enter the old password, then enter the new password twice, and click “OK”. Perform the same steps as above for the second default Admin identity (e.g. “Admin2”).



Connecting via FXCLI

Open the FXCLI application and run the following commands:

```
$ connect usb
$ login user
```

NOTE: The "login" command will prompt for the username and password. You will need to run it twice because you must login with both default Admin identities.

The default Admin passwords (i.e. “safe”) must be changed for both of your default Admin Identities (e.g. “Admin1” and “Admin2”) in order to load the major keys onto the HSM.

The following FXCLI commands can be used to change the passwords for each default Admin Identity.

```
$ user change-password -u Admin1
$ user change-password -u Admin2
```

NOTE: The user change-password commands above will prompt you to enter the old and new passwords. It is necessary to run the command twice (as shown above) because the default password must be changed for both default Admin identities.

[7.2] FEATURES REQUIRED IN HSM

In order to establish a connection between the PKCS #11 Library and the Futurex HSM, the HSM must be configured with the following features:

- **PKCS #11** -> Enabled
- **Command Primary Mode** -> General Purpose (GP)

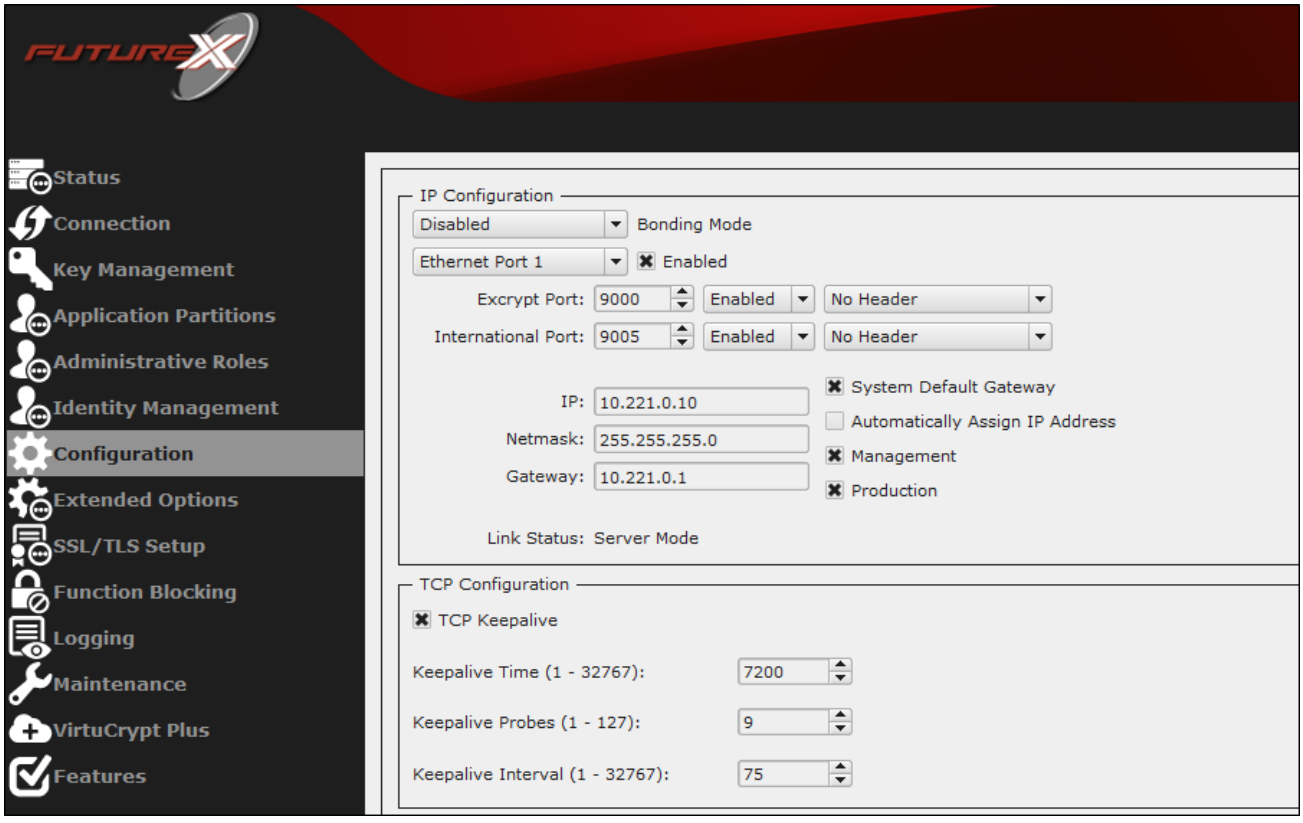
NOTE: For additional information about how to update features on your HSM, please refer to your HSM Administrator's Guide, section **"Download Feature Request File"**.

NOTE: Command Primary Mode = General Purpose, will enable the option to create the FTK major key in the HSM. This key will be required to be able to use the PKCS #11 library to communicate with the HSM. For detailed information about how to load major keys in HSMs please refer to your HSM Administrator's Guide.

[7.3] NETWORK CONFIGURATION (HOW TO SET THE IP OF THE HSM)

For this step you will need to be logged in with an identity that has a role with permissions ***Communication:Network Settings****. The default Administrator role and Admin identities can be used.*

Navigate to the *Configuration* page. There you will see the option to modify the IP configuration, as shown below:



Alternatively, the following **FXCLI** command can be used to set the IP for the HSM:

```
$ network interface modify --interface Ethernet1 --ip 10.221.0.10 --netmask 255.255.255.0 --gateway
10.221.0.1
```

NOTE: The following should be considered at this point:

- All of the remaining HSM configurations in this section can be completed using the Guardian Series 3 (please refer to Appendix A for instructions on how to do so), with the exception of the final subsection that covers how to create connection certificates for mutual authentication.
- If you are performing the configuration on the HSM directly now, but plan to add the HSM to a Guardian later, it may be necessary to synchronize the HSM after it is added to a Device Group on the Guardian.
- If configuration through a CLI is required for your use-case, then you should manage the HSMs directly.

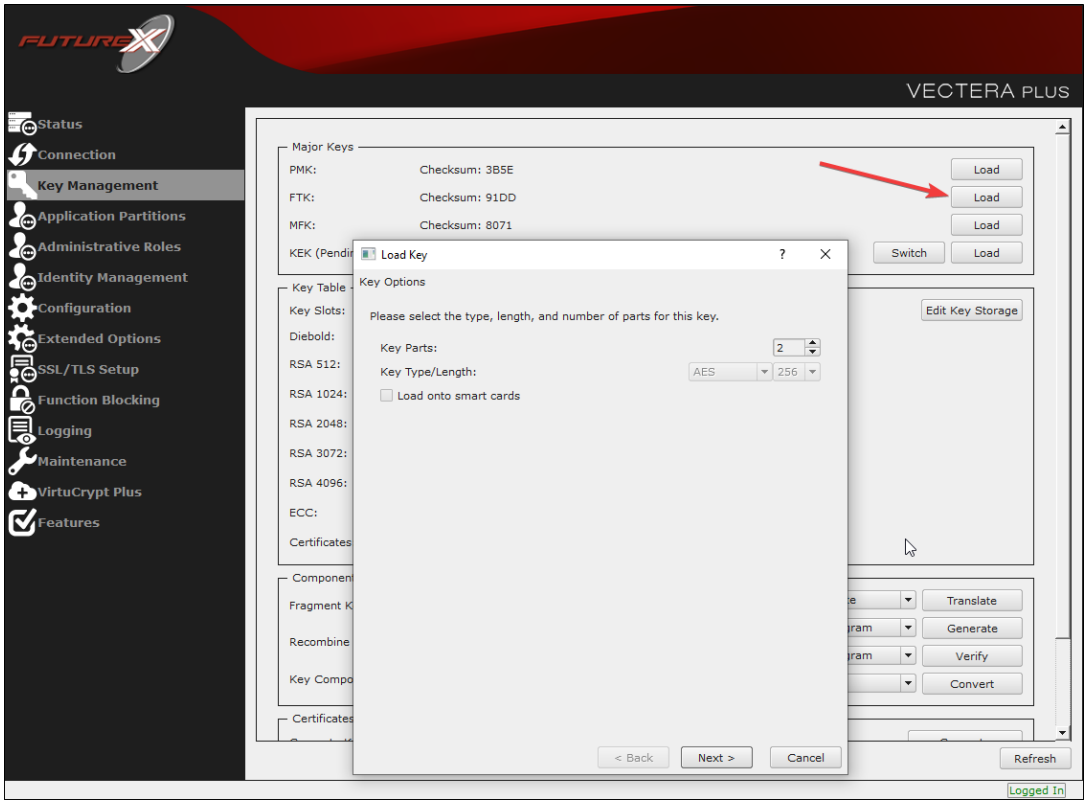
[7.4] LOAD FUTUREX KEY (FTK)

For this step you will need to be logged in with an identity that has a role with permissions ***Major Keys:Load****. The default Administrator role and Admin identities can be used.*

The FTK is used to wrap all keys stored on the HSM used with PKCS #11. If using multiple HSMs in a cluster, the same FTK can be used for syncing HSMs. Before an HSM can be used with PKCS #11, it must have an FTK.

NOTE: This process can also be completed using FXCLI, the Excrypt Touch, or the Guardian Series 3. For more information about how to load the FTK into an HSM using these tools/devices, please see the relevant Administrative Guide.

After logging in, select *Key Management*, then "Load" under FTK. Keys can be loaded as components that are XOR'd together, M-of-N fragments, or generated. If this is the first HSM in a cluster, it is recommended to generate the key and save to smart cards as M-of-N fragments.



Alternatively, the following **FXCLI** commands can be used to load an FTK onto an HSM.

If this is the first HSM you are setting up you will need to generate a random FTK. Optionally, you can also load it onto smart cards simultaneously with the -m and -n flags.

```
$ majorkey random --ftk -m [number_from_2_to_9] -n [number_from_2_to_9]
```

If it's a second HSM that you're setting up in a cluster then you will load the FTK from smart cards with the following command:

```
$ majorkey recombine --key ftk
```

[7.5] CONFIGURE A TRANSACTION PROCESSING CONNECTION AND CREATE AN APPLICATION PARTITION

For this step you will need to be logged in with an identity that has a role with permissions ***Role:Add, Role:Assign All Permissions, Role:Modify, Keys:All Slots****, and* ***Command Settings:Excrypt****. The default Administrator role and Admin identities can be used.*

NOTE: For the purposes of this integration guide you can consider the terms "Application Partition" and "Role" to be synonymous. For more information regarding Application Partitions, Roles, and Identities, please refer to the relevant Administrator's guide.

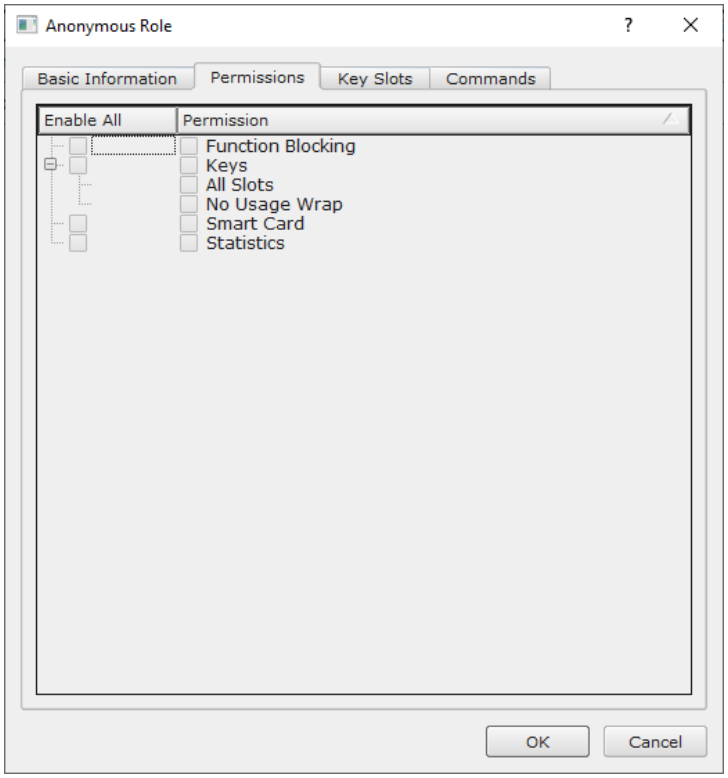
Configure a Transaction Processing Connection

Before an application logs in to the HSM with an authenticated user, it first connects via a “Transaction Processing” connection to the “Anonymous” Application Partition. For this reason, it is necessary to take steps to harden the “Anonymous” Application Partition. These three things need to be configured for the “Anonymous” partition:

1. It should not have access to the “All Slots” permissions
2. It should not have access to any key slots
3. Only the PKCS #11 communication commands should be enabled

Go to *Application Partitions*, select the “Anonymous” Application Partition, and click Modify.

Navigate to the "Permissions" tab and ensure that the “All Slots” key permission is unchecked. None of the other key permissions should be enabled either.



Under the "Key Slots" tab you need to ensure that there are no key ranges specified. By default, the Anonymous Application Partition has access to the entire range of key slots on the HSM.

Lastly, under the “Commands” tab make sure that only the following **PKCS #11 Communication commands** are enabled for the "Anonymous" Application Partition:

- **ECHO**: Communication Test/Retrieve Version
- **PRMD**: Retrieve HSM restrictions
- **RAND**: Generate random data
- **HASH**: Retrieve device serial
- **GPKM**: Retrieve key table information
- **GPKS**: General purpose key settings get/change
- **GPKR**: General purpose key settings get (read-only)

Alternatively, the following **FXCLI** commands can be used to remove all permissions and key ranges that are currently assigned to the “Anonymous” role and enable only the PKCS #11 Communication commands:

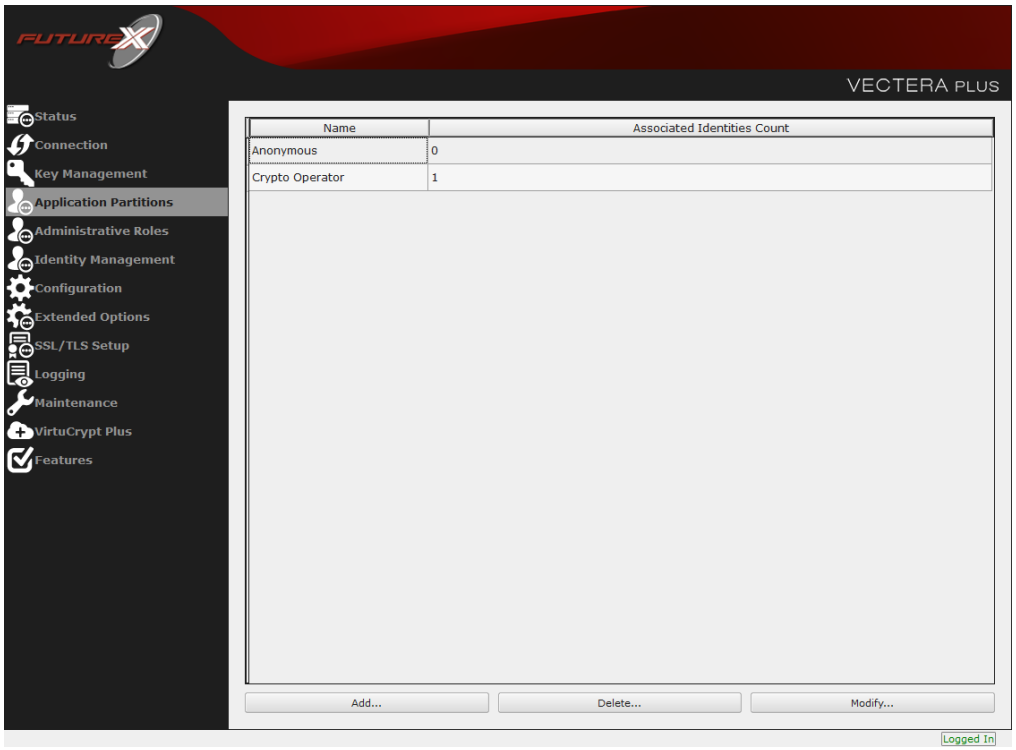
```
$ role modify --name Anonymous --clear-perms --clear-key-ranges
```

```
$ role modify --name Anonymous --add-perm Excrypt:ECHO --add-perm Excrypt:PRMD --add-perm Excrypt:RAND
--add-perm Excrypt:HASH --add-perm Excrypt:GPKM --add-perm Excrypt:GPKS --add-perm Excrypt:GPKR
```

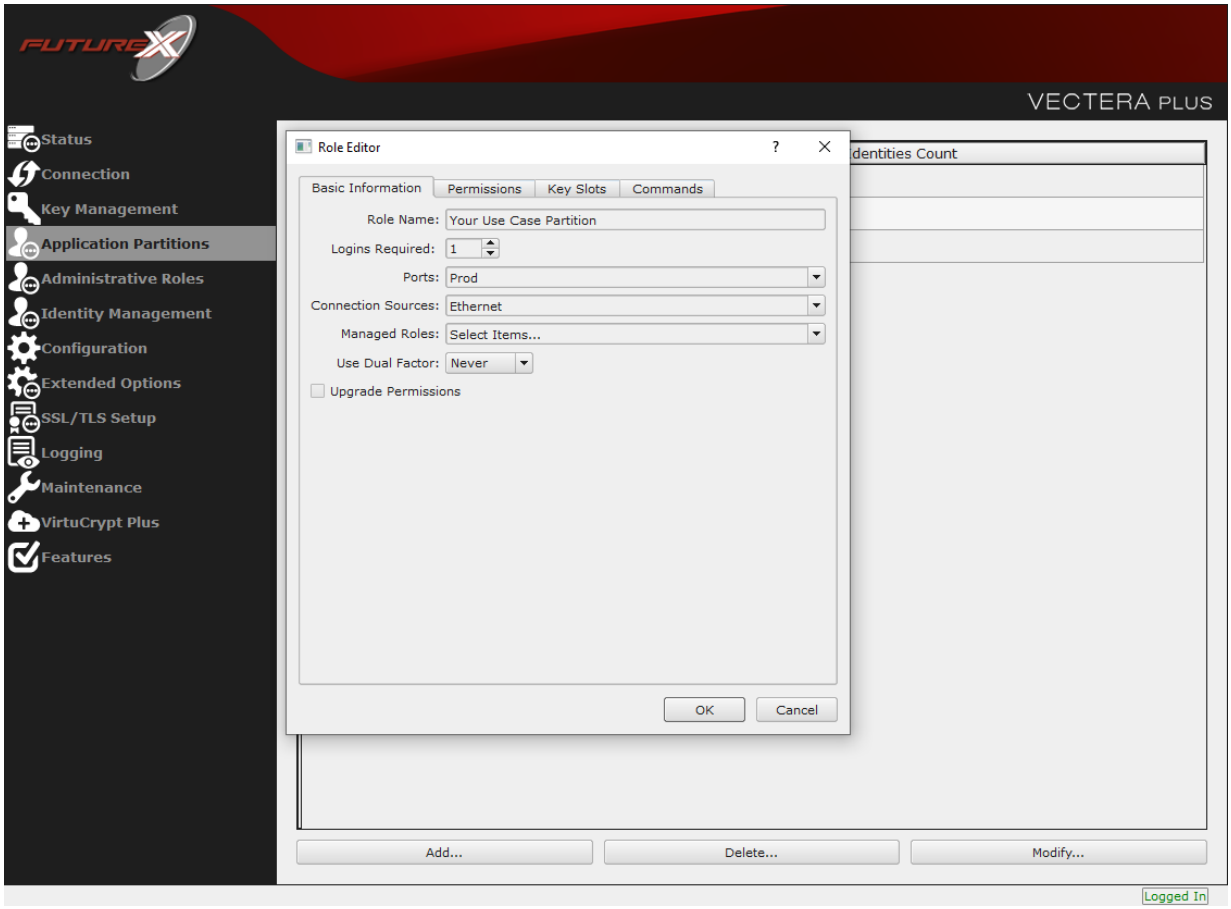
Create an Application Partition

In order for application segregation to occur on the HSM, an Application Partition must be created specifically for your use case. Application partitions are used to segment the permissions and keys on an HSM between applications. The process for configuring a new application partition is outlined in the following steps:

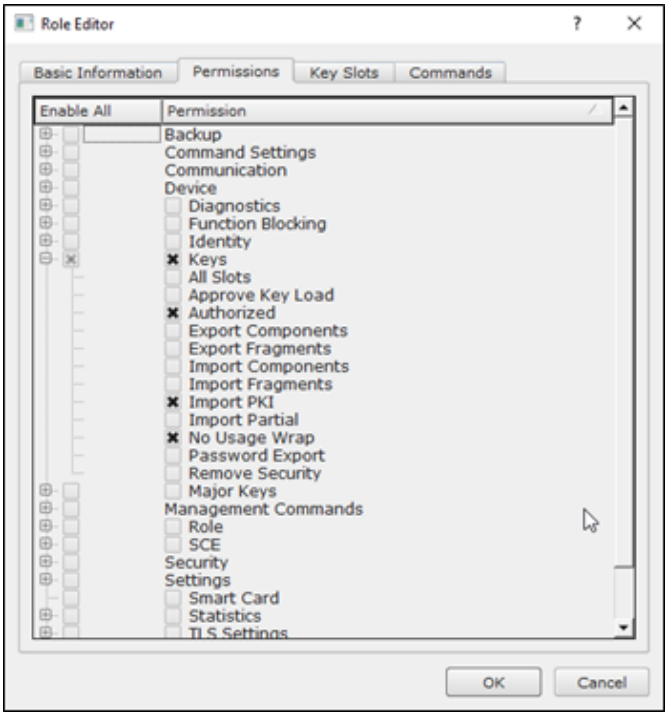
Navigate to the *Application Partitions* page and click the "Add" button at the bottom.



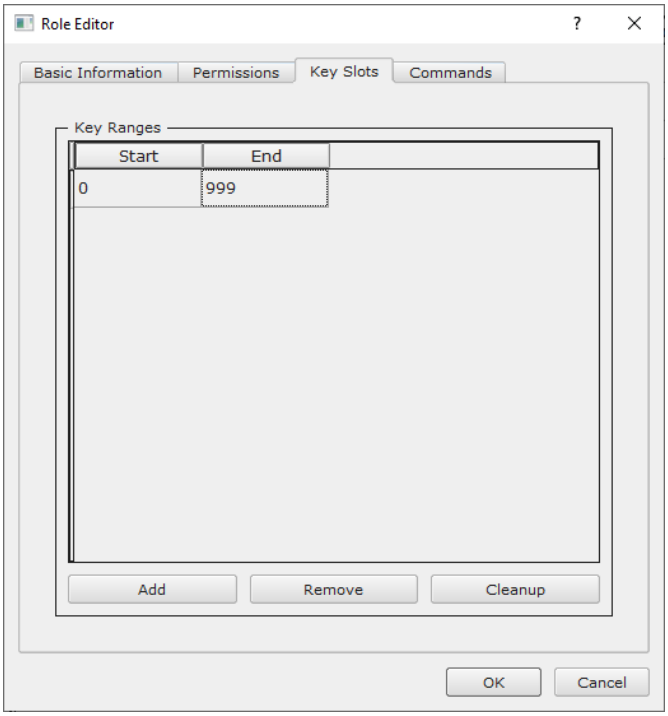
Fill in all of the fields in the *Basic Information* tab exactly how you see below (except for the *Role Name* field). In the *Role Name* field, specify any name that you would like for this new Application Partition. *Logins Required* should be set to "1". *Ports* should be set to "Prod". *Connection Sources* should be configured to "Ethernet". The *Managed Roles* field should be left blank because we'll be specifying the exact Permissions, Key Slots, and Commands that we want this Application Partition/Role to have access to. Lastly, the *Use Dual Factor* field should be set to "Never".



Under the "Permissions" tab, select the key permissions shown in the screenshot below. The **Authorized** permission allows for keys that require login. The **Import PKI** permission allows trusting an external PKI, which is used by some applications to allow for PKI symmetric key wrapping (It is not recommended to enable unless using this use case). The **No Usage Wrap** permission allows for interoperable key wrapping without defining key usage as part of the wrapped key (This is only recommended if exchanging keys with external entities or using the HSM to wrap externally used keys).



Under key slots, it is recommended that you create a range of 1000 total keys (here we've specified the key range 0-999), which do not overlap with another Application Partition. Within this range, there must be ranges for both symmetric and asymmetric keys. If more keys are required by the application, configure accordingly.



Based on application requirements there are particular functions that need to be enabled on the Application Partition in order to utilize the HSMs functionality. The most often used commands are included below. These can be enabled under the "Commands" tab.

PKCS #11 Communication Commands

- ECHO: Communication Test/Retrieve Version
- PRMD: Retrieve HSM restrictions
- RAND: Generate random data
- HASH: Retrieve device serial
- GPKM: Retrieve key table information
- GPKS: General purpose key settings get/change
- GPKR: General purpose key settings get (read-only)

Key Operations Commands

- APFP: Generate PKI Public Key from Private Key
- ASYL: Load asymmetric key into key table
- GECC: Generate an ECC Key Pair
- GPCA: General purpose add certificate to key table
- GPGS: General purpose generate symmetric key
- GPKA: General purpose key add
- GPKD: General purpose key slot delete/clear
- GRSA: Generate RSA Private and Public Key
- LRSA: Load key into RSA Key Table
- RPFP: Get public components from RSA private key

Interoperable Key Wrapping

- GPKU: General purpose key unwrap (unrestricted)
- GPUK: General purpose key unwrap (preserves key usage)
- GPKW: General purpose key wrap (unrestricted)
- GPWK: General purpose key wrap (preserves key usage)

Data Encryption Commands

- ADPK: PKI Decrypt Trusted Public Key
- GHSH: Generate a Hash (Message Digest)
- GPED: General purpose data encrypt and decrypt
- GPGC: General purpose generate cryptogram from key slot
- GPMC: General purpose MAC (Message Authentication Code)
- GPSR: General purpose RSA encrypt/decrypt or sign/verify with recovery
- HMAC: Generate a hash-based message authentication code
- RDPK: Get Clear Public Key from Cryptogram

Signing Commands

- ASYS: Generate a Signature Using a Private Key
- ASYV: Verify a Signature Using a Public Key
- GPSV: General purpose data sign and verify
- RSAS: Generate a Signature Using a Private Key

Alternatively, the following **FXCLI** commands can be used to create the new Application Partition and enable all of the functions that are needed:

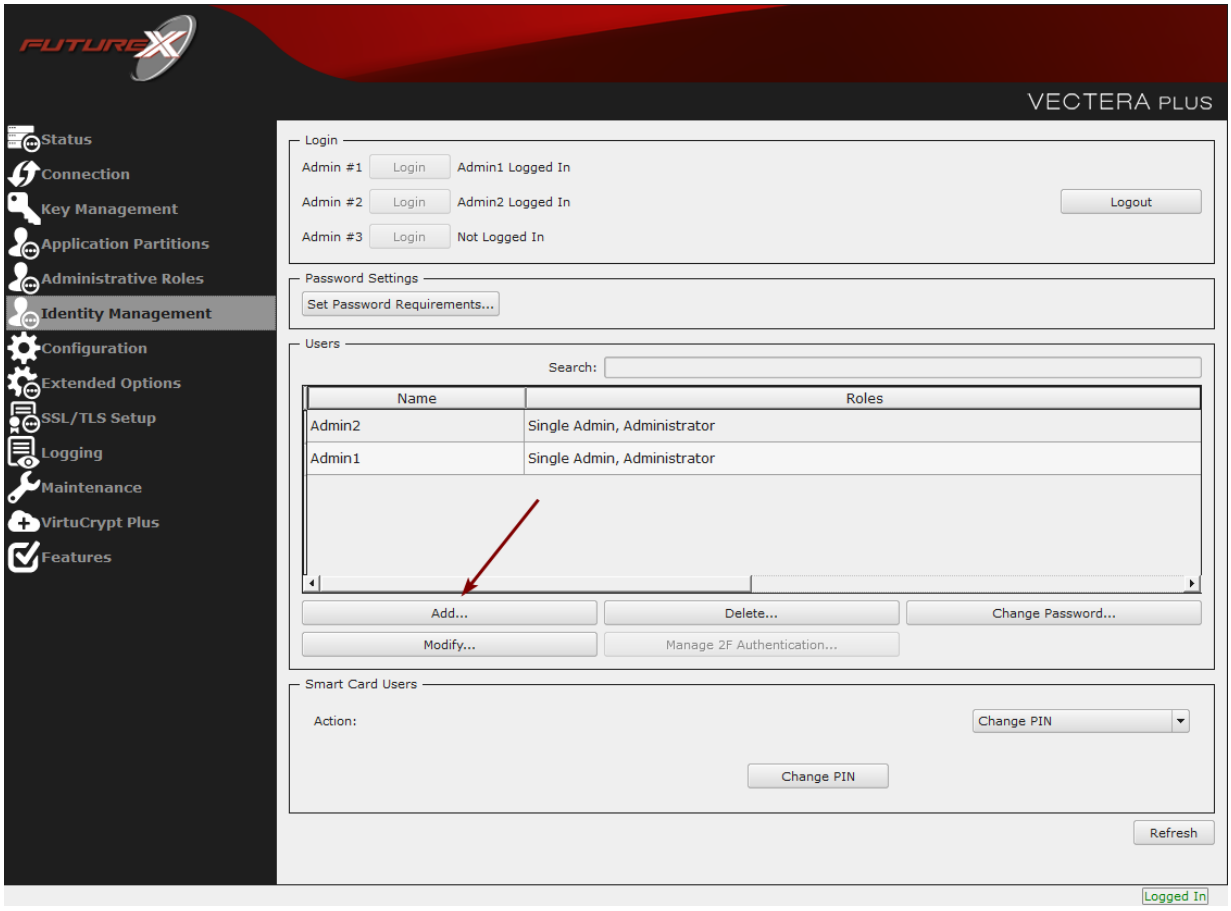
```
$ role add --name Role_Name --application --key-range (0,999) --perm "Keys:Authorized" --perm "Key-
s:Import PKI" --perm "Keys:No Usage Wrap"
```

```
$ role modify --name [role_name] --clear-perms --add-perm Excrypt:ECHO --add-perm Excrypt:PRMD --add-
perm Excrypt:RAND --add-perm Excrypt:HASH --add-perm Excrypt:GPKM --add-perm Excrypt:GPKS --add-perm
Excrypt:GPKR --add-perm Excrypt:APFP --add-perm Excrypt:ASYL --add-perm Excrypt:GECC --add-perm
Excrypt:GPCA --add-perm Excrypt:GPGS --add-perm Excrypt:GPKA --add-perm Excrypt:GPKD --add-perm
Excrypt:GRSA --add-perm Excrypt:LRSA --add-perm Excrypt:RPFP --add-perm Excrypt:GPKU --add-perm
Excrypt:GPUK --add-perm Excrypt:GPKW --add-perm Excrypt:GPWK --add-perm Excrypt:ADPK --add-perm
Excrypt:GHSH --add-perm Excrypt:GPED --add-perm Excrypt:GPGC --add-perm Excrypt:GPMC --add-perm
Excrypt:GPSR --add-perm Excrypt:HMAC --add-perm Excrypt:RDPK --add-perm Excrypt:ASYS --add-perm
Excrypt:ASYV --add-perm Excrypt:GPSV --add-perm Excrypt:RSAS
```

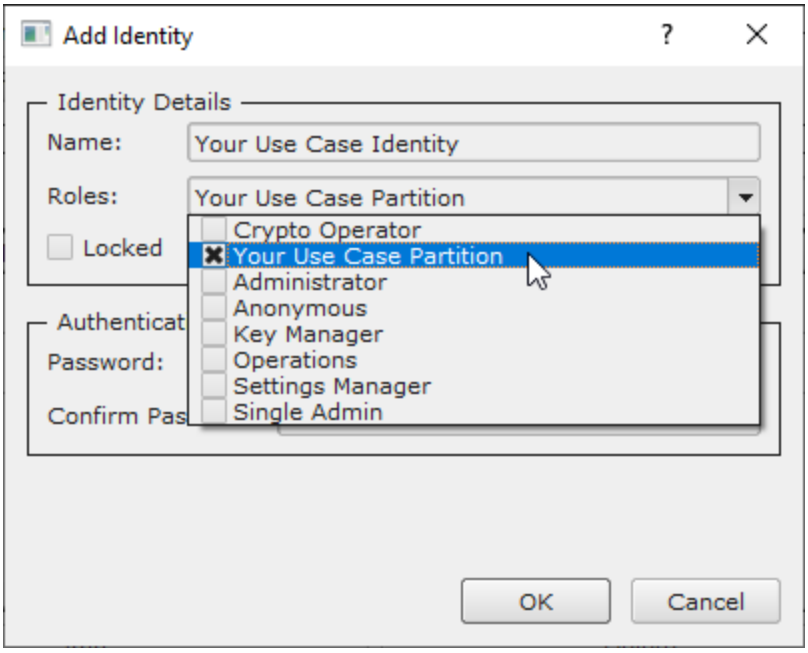
[7.6] CREATE NEW IDENTITY AND ASSOCIATE IT WITH THE NEWLY CREATED APPLICATION PARTITION

*For this step you will need to be logged in with an identity that has a role with permissions **Identity:Add**. The default Administrator role and Admin identities can be used.*

A new identity must be created, which will need to be associated with the Application Partition created in the previous step. To create this new identity, go to *Identity Management*, and click "Add".



Specify a name for the new identity, and in the Roles dropdown select the name of the Application Partition created in the previous step. This will associate the new Identity with the Application Partition that you created.



Alternatively, the following **FXCLI** command can be used to create a new Identity and associate it with the role that was created:

```
$ identity add --name Identity_Name --role Role_Name --password safest
```

This new identity must be set in fxpkcs11.cfg file, in the following section:

```
#HSM crypto operator identity name
<CRYPTO-OPR>     [insert name of identity that you created]     </CRYPTO-OPR>

# Production connection
<PROD-ENABLED>    YES        </PROD-ENABLED>
<PROD-PORT>        9100         </PROD-PORT>
```

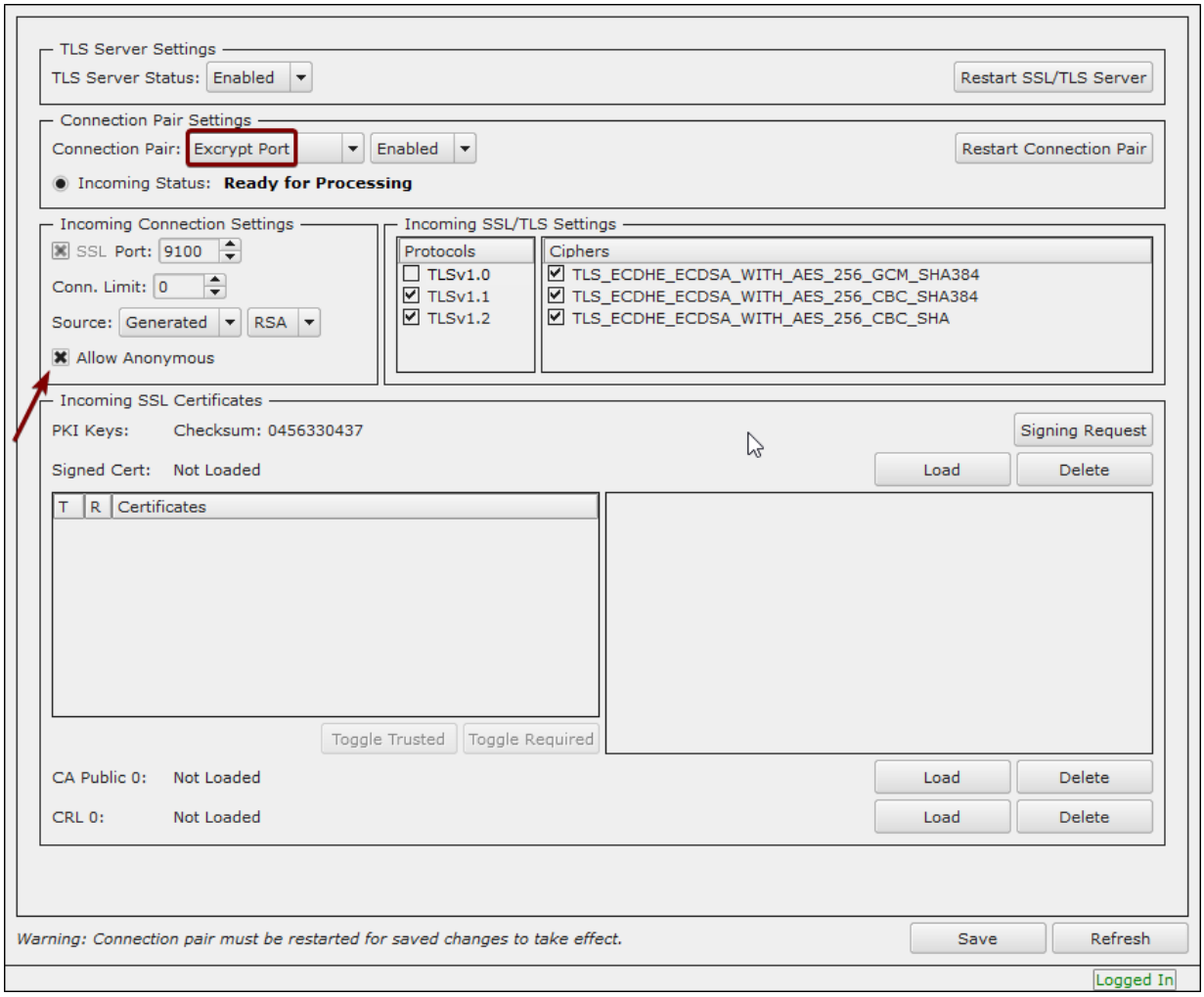
NOTE: Crypto Operator in the fxpkcs11.cfg file must match exactly the name of the identity created in the HSM.

[7.7] CONFIGURE TLS AUTHENTICATION

For this step you will need to be logged in with an identity that has a role with permissions ***Keys:All Slots, Management Commands:Certificates, Management Commands:Keys, Security:TLS Sign****, and* ***TLS Settings:Upload Key****. The default Administrator role and Admin identities can be used.*

Enable Server-Side Authentication (Option 1)

Mutually authenticating to the HSM using client certificates is recommended, but server-side authentication is also supported. To enable server-side authentication go to *SSL/TLS Setup*, then select the Excrypt Port and enable the "Allow Anonymous" setting.



Alternatively, the following **FXCLI** command can be used to enable server-side authentication with the "Allow Anonymous" SSL/TLS setting:

```
$ tls-ports set -p "Excrypt Port" --anon
```

Create Connection Certificates for Mutual Authentication (Option 2)

Mutually authenticating to the HSM using client certificates is recommended, and enforced by default. In the example below, FXCLI is utilized to generate a CA that then signs the HSM server certificate and a client certificate. The client keys and CSR are generated in Windows PowerShell with OpenSSL. For other options for managing certificates required for mutual authentication with the HSM, please review the relevant Administrator's guide.

Find the **FXCLI** program that was installed with FXTools, and run it as an administrator.

Things to note:

- For this example, the computer running FXCLI is connected to the front port of the HSM. Remote management is possible however, using the HSMs Web Portal, or the Excrypt Touch.
- For commands that create an output file, if you do not specify a file path (as is the case here) it will save the file to the directory from which the FXCLI program is executed.
- Using user-generated certificates requires a PMK to be loaded on the HSM.
- If you run **help** by itself it will show a full list of available commands. You can see all of the available options for any given command by running the command name followed by **help**.

```
# Connect your laptop to the HSM via the USB port on the front, then run this command.
$ connect usb
```

```
# Log in with both default Admin identities. This command will prompt for the username and password.
You will need to run this command twice.
$ login user
```

```
# Generate TLS CA and store it in an available key slot on the HSM
$ generate --algo RSA --bits 2048 --usage mak --name TlsCaKeyPair --slot next
```

```
# Create root certificate
$ x509 sign \
    --private-slot TlsCaKeyPair \
    --key-usage DigitalSignature --key-usage KeyCertSign \
    --ca true --pathlen 0 \
    --dn 'O=Futurex\CN=Root' \
    --out TlsCa.pem
```

```
# Generate the server keys for the HSM
$ tls-ports request --pair "Excrypt Port" --file production.csr --pki-algo RSA
```

```
# Sign the server CSR with the newly created TLS CA
$ x509 sign \
    --private-slot TlsCaKeyPair \
    --issuer TlsCa.pem \
    --csr production.csr \
    --eku Server --key-usage DigitalSignature --key-usage KeyAgreement \
    --ca false \
    --dn 'O=Futurex\CN=Production' \
    --out TlsProduction.pem
```

```
# Push the signed server PKI to the production port on the HSM
$ tls-ports set --pair "Excrypt Port" \
    --enable \
    --pki-source Generated \
    --clear-pki \
    --ca TlsCa.pem \
```

```
    --cert TlsProduction.pem \
    --no-anon
```

***NOTE**: The following OpenSSL commands will need to be run from Windows PowerShell, rather than from the FXCLI program.*

```
# Generate the client keys
$ openssl genrsa -out privatekey.pem 2048
```

```
# Generate client CSR
$ openssl req -new -key privatekey.pem -out ClientPki.csr -days 365
```

Using FXCLI, sign the CSR that was just generated using OpenSSL.

```
# Sign the client CSR under the root certificate that was created
$ x509 sign  \
 --private-slot TlsCaKeyPair \
 --issuer TlsCa.pem \
 --csr ClientPki.csr \
 --eku Client --key-usage DigitalSignature --key-usage KeyAgreement \
 --dn 'O=Futurex\CN=Client' \
 --out SignedPki.pem
```

Switch back to Windows PowerShell for the remaining commands.

```
## Make PKCS12 file
# Concatenate the signed client cert and private key into one pem file
$ cat SignedPki.pem >> Tree.pem
```

```
$ cat privatekey.pem >> Tree.pem
```

```
# Use OpenSSL to create a PKCS#12 file that can be used to authenticate, as a client, using our PKCS
#11 library
$ openssl pkcs12 -export -in Tree.pem -out PKI.p12 -name "ClientPki" -password pass:safest
```

[8] EDIT THE FXPKCS11 CONFIGURATION FILE

The *fxpkcs11.cfg* file allows the user to set the PKCS #11 library to connect to the HSM. To edit, run a text editor as an Administrator and edit the configuration file accordingly. Most notably, the fields shown below must be set inside the **<HSM>** section (note that the full *fxpkcs11.cfg* file is not included).

NOTE: Our PKCS #11 library expects the PKCS #11 config file to be in a certain location (*C:\Program Files\Futurex\fxpkcs11\fxpkcs11.cfg* for Windows and */etc/fxpkcs11.cfg* for Linux), but that location can be overwritten using an environment variable (FXPKCS11_CFG).

```
# Connection information
<ADDRESS>              10.0.5.58           </ADDRESS>

# Load balancing
<FX-LOAD-BALANCE>           YES               </FX-LOAD-BALANCE>

# Log configuration
<LOG-FILE> C:\Program Files\Futurex\fxpkcs11\fxpkcs11.log </LOG-FILE>

# HSM crypto operator identity name
<CRYPTO-OPR>          [identity_name]     </CRYPTO-OPR>

# Production connection
<PROD-ENABLED>         YES                </PROD-ENABLED>
<PROD-PORT>            9100               </PROD-PORT>

# Production SSL information
<PROD-TLS-ANONYMOUS>    NO                </PROD-TLS-ANONYMOUS>
<PROD-TLS-CA>     C:\Program Files\Futurex\fxpkcs11\TlsCa.pem    </PROD-TLS-CA>
<PROD-TLS-CA>     C:\Program Files\Futurex\fxpkcs11\TlsProduction.pem    </PROD-TLS-CA>
<PROD-TLS-KEY>     C:\Program Files\Futurex\fxpkcs11\PKI.p12     </PROD-TLS-KEY>
<PROD-TLS-KEY-PASS>      safest          </PROD-TLS-KEY-PASS>
```

In the **<ADDRESS>** field, the IP of the HSM that the PKCS #11 library will connect to is specified.

If a Guardian is being used to manage HSMs in a cluster, the **<FX-LOAD-BALANCE>** field must be defined as "YES". If a Guardian is not being used it should be set to "NO".

In the **<LOG-FILE>** field, set the path to the PKCS #11 log file.

In the **<CRYPTO-OPR>** field, the name of the identity created in step 7.6 needs to be specified.

The **<PROD-ENABLED>** and **<PROD-PORT>** fields declare that the PKCS #11 library will connect to Production port 9100.

The **<PROD-TLS-ANONYMOUS>** field defines whether the PKCS #11 library will be authenticating to the server or not.

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, it is not necessary to define the signed client cert with the **<PROD-TLS-CERT>** tag, or the CA cert/s with one or more instances of the **<PROD-TLS-CA>** tag.

For additional details reference the Futurex PKCS #11 technical reference found on the Futurex Portal.

Once the *fxpkcs11.cfg* is edited, run the *PKCS11Manager* file to test the connection against the HSM, and check the *fxpkcs11.log* for errors and information. For more information, see our Administrator's Guide.

[9] STEPS TO CONFIGURE THE FUTUREX PKCS #11 LIBRARY WITH AXWAY VA (VALIDATION AUTHORITY) SERVER

[9.1] INSTALL VA SERVER

[9.1.1] Install VA Server on Windows

NOTE: VA Server is no longer installed as an interactive service on Windows. This applies to both the Admin UI service and the Validation Authority Service that is installed as part of VA Server.

1. Using an account in the Administrators group, log on to the computer on which you will install the VA Server.
2. Copy the `Validation_Authority_Server_<Release Version>win-x86-64_BNXXX.exe` file that you received from Axway Global Support to the Windows system.
   Where:
   Release Version = 5.1_Install for 5.1 GA Release
   Release Version = 5.1_SP1 for Service Pack 1

   NOTE: The distributed installation file is digitally signed and checked by the Windows platform prior to installation.
3. Double-click `Validation_Authority_Server_<Release Version>win-x86-64_BNXXX.exe`.
   The *Welcome* page displays.
   Follow the on-screen instructions as you proceed through the installation.
   - Click **Next** to move forward to the next installation window.
   - Click **Back** to return to the previous installation window.
   - Click **Cancel** to close the installation program without installing any component of the VA Server. To install VA Server, re-run the installation program.
   - Click **Next**.
     The *License Agreement* page displays.
4. Click **Accept** to accept the license agreement and go to the next page in the installer. Click **No** to cancel the installation.
   The *Customer Information* page displays.
5. Type your **User Name**, **Company Name**, and **Email Address** in the text fields provided. These are required fields except for the Email Address. However, you should provide an email address because it is used by the VA administration server to perform email notifications.
6. Click **Next**.
   The *Choose Destination Location* page displays, showing the default destination folder where VA Server components are installed.
7. To select a different destination folder, click **Browse** and enter the folder location.

8. Click **Next**.
   The *VA Server Information* page displays.
9. Enter the requested information on the host name, port number, and user for the VA administration server.
   a. Type the VA Server host name.
      The host name identifies the computer. The default host name is the name of the computer on which you are installing the VA Server.
   b. Type the VA administration server port number.
      This port number identifies the port at which the VA administration server listens for HTTPS requests from the browser. If you use a port other than the default (13333), make a note of it for future reference.
   c. Type the VA administration server user and password.
      This user is the initial user who can log in to the VA administration server. The default user name is "admin". If you type a different name, make a note of it. After completing the installation, you will log in to the VA administration server using this user name.
      The password must be at least eight characters long, contain at least one alphabetic character, one digit, one special character, one upper case character, one lower case character, and meet the requirements in the *Manage VA administration server users* section on page 77 of the Axway Validation Authority Administrator Guide. Re-type the password to confirm it. Click **Next** to continue.
10. Select either the option to generate a self-signed certificate or import a PFX / P12 file. If you select **Generate a Self-Signed Certificate**, click Next and continue to step 11. If you select **Import PFX / P12 from file**:
    a. Select the file to import from the file selection dialog box and then click **Open**.
    b. Enter a password to decrypt the file. This password was originally used to protect the PFX file.
    c. By default, the **Encrypt Admin UI Private Key** option is selected. If you do not want this option, uncheck the box to disable the password field. Enter a password to encrypt the admin server key for the VA Server. The password must be at least eight characters long, contain at least one alphabetic character, one digit, one special character, one upper case character, one lower case character, and meet the requirements in the *Manage VA administration server users* section on page 77 of the Axway Validation Authority Administrator Guide. This encryption option, along with the provided password, will automatically call apachepassphrase for unattended startup.
    d. Click **Next** to continue. *The Start Copying Files* page displays.
11. Check the current settings to ensure they are as desired. If you need to make any changes to the settings, use the **Back** button. Otherwise, click **Next** to continue.
12. Files are installed to the specified destination location.
    After the installation finishes, the *InstallShield Wizard Complete* page displays.
    The VA Server is successfully installed. This can be later verified using the Admin Server **User Interface >**

Help > About page, which displays the current version.

13. Clear the **Launch Administrative Server User Interface** check box to start the VA administration server at a later time.
14. Click **Finish**.
    The installation program adds the VA Server to your **Start** menu. If you access **Control Panel > Administrative Tools > Services** you will see Axway Validation Authority and Axway VA Admin included in the list of services. You can access the VA Server admin UI and this document from the **Start** menu.
    The installation also automatically creates an VA administrative server private key (`adminserver.key`) and SSL certificate (`adminserver.crt`) in the `<VADataDir>\entserv` directory. (Example: `C:\ProgramData\Axway\VA\entserv` in Windows.)
    You are now ready to use the VA administration server to configure, start, and manage the VA Server.

[9.1.2] Install VA Server on Linux

You do not have to be root in order to install the VA Server, but non-root users will not be able to configure the installation to use a port lower than 1024. When installing as root on a port lower than 1024, you will be asked whether to run the server in"setuid root" mode. This mode is required to start the server using the admin UI. In this case, the server will run as root, but only during initialization. Once the listening sockets have been established, the process will "step down" to that of the invoking user (for example, `nobody`).

NOTE: The distributed installation file is digitally signed by the Axway generated GPG key and can be verified using the shipped GPG public key prior to installation.

1. Copy the `Validation_Authority_Server_<Release Version>_linux-x86-64_BN<build number>.rpm` file that you received from Axway Global Support to the Linux system.
   Where:
   Release Version = 5.1_Install for 5.1 GA release
   Release Version = 5.1_SP1 for Service Pack 1
   NOTE: This RPM package is dependent on other RPM packages that are generally available from RHEL RPM repository(s). If these packages are not already installed on the system, the installation will report necessary missing package(s) and fail. If this happens, install the missing packages and install this RPM package again.
2. Extract the files with the following command:
   ```
   rpm -U Validation_Authority_Server_<Release Version>_linux-x86-64_BN<build number>.rpm
   ```
   NOTE: If a previous version of the RPM is installed on the system, this command will remove the previous version and install the new version to `/opt/va_install/<Version><SPnumber>/VCeva` where Version = 5.1. SPnumber is only applicable for Service Pack releases (example: SP1).
3. Change directories to the `Validation Authority Server` directory:
   ```
   cd /opt/va_install/<Version><SPnumber>/VCeva
   ```
   IMPORTANT: Do not install under the `va_install` directory when running the install script. The `rpm`

`uninstall` will erase the va_install directory.

4. Enter the following at the command line prompt to run the installation script:
```
./install_eva
```

   The installation script then prompts whether to install using ports 1024 and above, assuming you are not installing as root.
   You must install as root to select a port under 1024. You must also answer "yes" when prompted to run `setuid root` to start the server through the admin UI.

5. Enter `y` (yes) or `n` (no).
   The installation script displays the Axway software licensing agreement and prompts you with the following:
```
Do you agree to the above terms? [y/n]
Default: [y]
```

6. Press **[Enter]** to accept the software licensing agreement.
   The installation script next prompts you for a location to install the VA Server.
```
Enter the Validation Authority install directory
Default: [/opt/axway]
```

7. Press **[Enter]** to accept the default, or enter a location to install the VA Server, then press **[Enter]**.
   The installation script next prompts you to enter a port number for the VA administration server:
```
Enter the port number for the Validation Authority Administration Server [1-65535].
Default: [13333]
```

8. The VA administration server is the administration component of the *Validation Authority*. This server, which is installed during the installation process, provides and administration user interface (admin UI) through which you configure and operate the VA validation server. If you choose to use a port other than the default, make a note of it for future reference. This port number identifies the port at which the VA administration server listens and exchanges information to perform configuration operations with the browser using HTTPS requests.

9. Press **[Enter]** to accept the default port number for the VA administration server, or enter a different number and press **[Enter]**.
   The script prompts you for the email address of the server administrator. It displays:
```
Enter the email address of the server administrator:
Default: [sysadmin]
```

   The VA administration server uses this email address to send informational messages to the server administrator during configuration and administration performed at the VA dialog boxes.

10. Press **[Enter]** to accept this email address, or enter a different address and then press **[Enter]**.
    The script prompts you for the server host name:
```
Enter the server's hostname (either a DNS name or IP address):
Default: [computer_name.yourdomain.com]
```

    Where `computer_name` is the name of your host computer, and `yourdomain` is the domain name for your host computer.
    The host name identifies the computer on which you have installed the *Validation Authority*.

11. Press **[Enter]** to accept the default server host name, or enter a different name and press **[Enter]**.
    The script prompts you for a user name to run the VA administration server. It displays:
    ```
    Enter the username to run the VA and Administration Servers as:
    Default: []
    ```
    If you are not installing as root, the default username displayed will be the user ID.
12. Press **[Enter]** to use the default username, or enter a different name and press **[Enter]**.
13. The following message displays:
    ```
    In order to start the VA via the web interface on a port less than 1024 ves must executre as
    setuid root. Do you wish to set this bit?
    Default: [y]
    ```
    The name of the VA Server process is `ves`.
14. If you plan to use a validation port number of 1024 or greater, type `n`, otherwise accept the default and press **[Enter]**.
    The script prompts you to identify the VA administration server user. This user is the initial user who can log in to the VA administration server. The default user name is "admin".
    ```
    Please enter the Administration server user id
    [admin]:
    ```
15. Press **[Enter]** to use the default VA administration server user name, or enter a different name and press **[Enter]**.
    If you type a different name, make a note of it. After completing the installation, you will log on to the VA administration server using this user name.
    The system configures the VA administration server user and then prompts for the VA administration server user password. Next confirm the password entry.
16. Enter and confirm the VA administration server user password.
    ```
    Please enter the Administration Server user password:
    Please confirm the Administration Server user password:
    ```
    The password must be at least 8 characters long and contain one uppercase, one lowercase, one digit, one special character.
17. The following message displays:
    ```
    Would you like to use an imported certificate, rather than generating a self-signed one, for the
    admin server's SSL certificate? [y|n]:
    Default: [n]
    ```
    Either enter **[n]** to generate a self-signed certificate, or **[y]** to import a PFX / P12 certificate. If **[n]**, continue to Step 18. If **[y]**:
    a. Optionally enter **[y]** to protect the private key. If you select **[y]**, a password prompt is displayed when the admin server starts.
    b. Enter the path to the certificate you are importing.

c. Enter the password to decrypt the file. This password was originally used to protect the PFX file.

d. At the PEM pass phrase prompt, enter a password to encrypt the admin server key for the admin UI. You will be prompted for this password when the admin server starts.
The installation automatically creates a VA administration server private key (`adminserver.key`) and SSL certificate (`adminserver.crt`) in the `/var/lib/va/entserv` directory.

18. The installation process completes and you are prompted to start the admin server.

```
Would you like to start the EVA Administration Server [y/n]?
Default: [y]
```

NOTE: VA Server is successfully installed. This can be verified using the Admin Server **User Interface > Help > About** page which displays the current version.

19. Press **[Enter]** to start the VA administration server.

[9.2] CONFIGURE VA SERVER

[9.2.1] Access the VA administration server UI

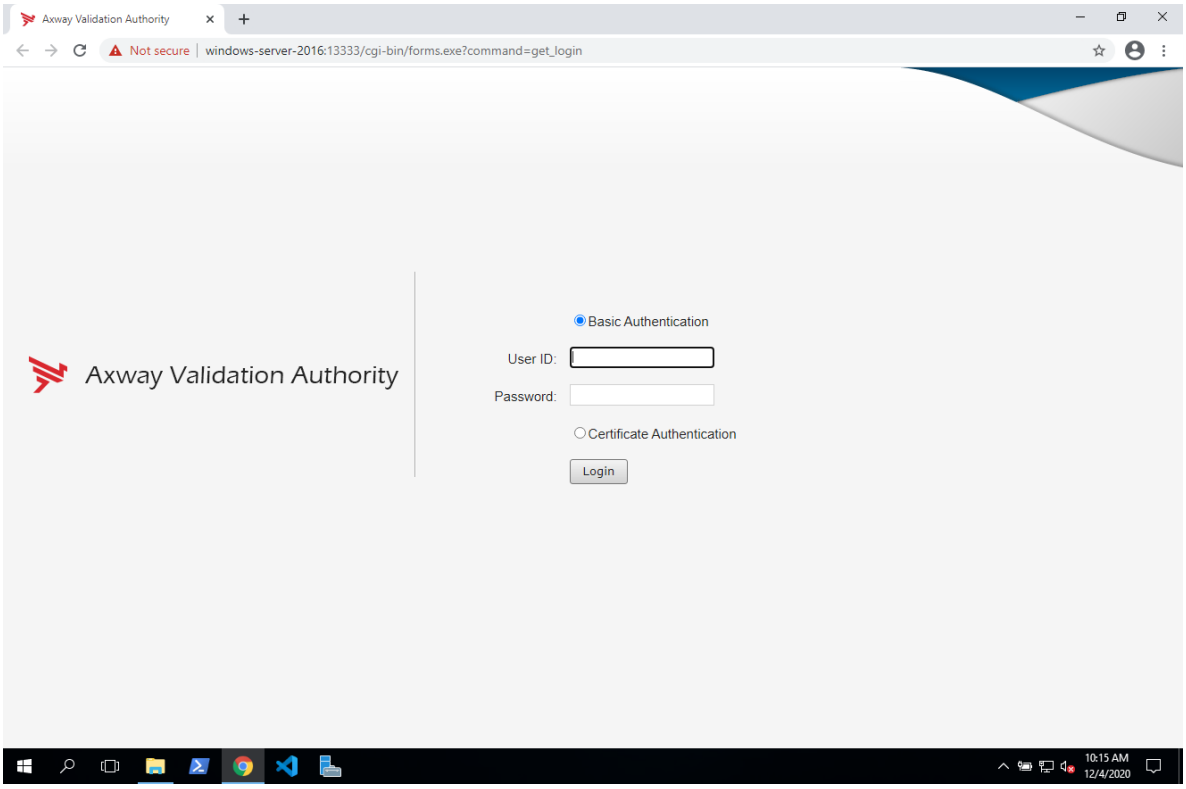
The admin UI requires an HTTPS server. This server is automatically installed and configured during VA Server installation. You can launch the admin UI automatically as the final step of installation, from the desktop icon created during the installation, or by accessing it directly from a browser using the VA administration server URL. For a standard connection, the URL is:

`https://<hostname>:<port>`.

Where *`<hostname>`* and *`<port>`* are the VA Server host name and VA administration server port number you provided during installation (13333 by default).

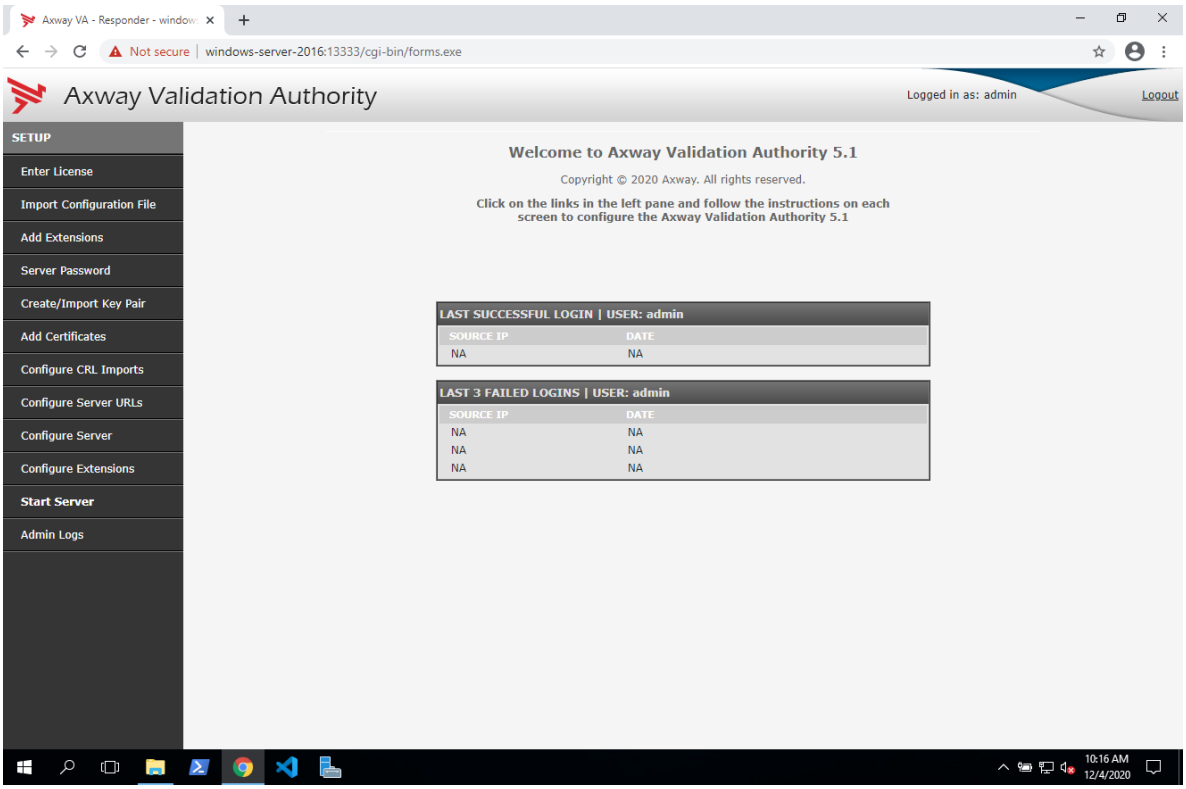
NOTE: The VA administration server is, by default, only available using SSL (https). Operating the VA administration server using non-SSL (http) disables certificate-based authentication for users.

When the web interface opens for the first time, you will receive an SSL certificate warning. Bypass this warning and proceed to the login page.



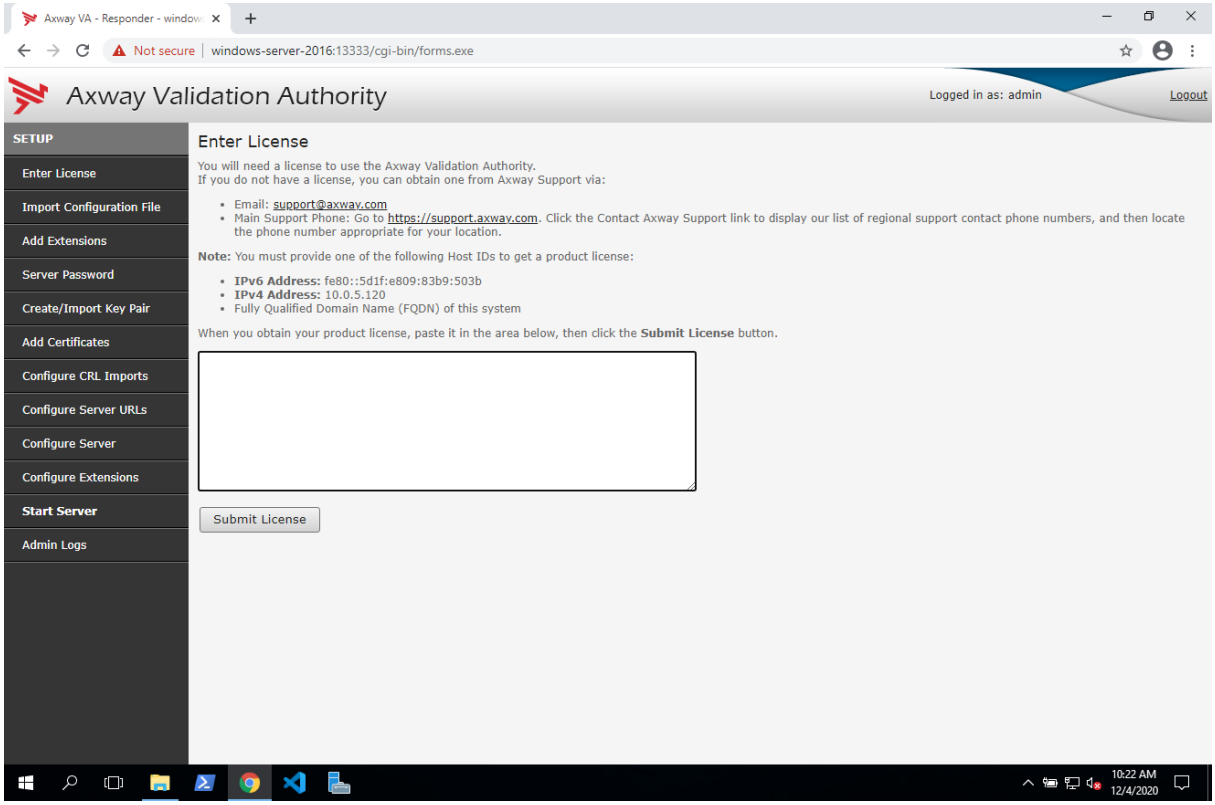
At the Administrative Login prompt, log in with Basic Authentication using the credentials set during installation.

After successful login it will load the home page of the admin UI.

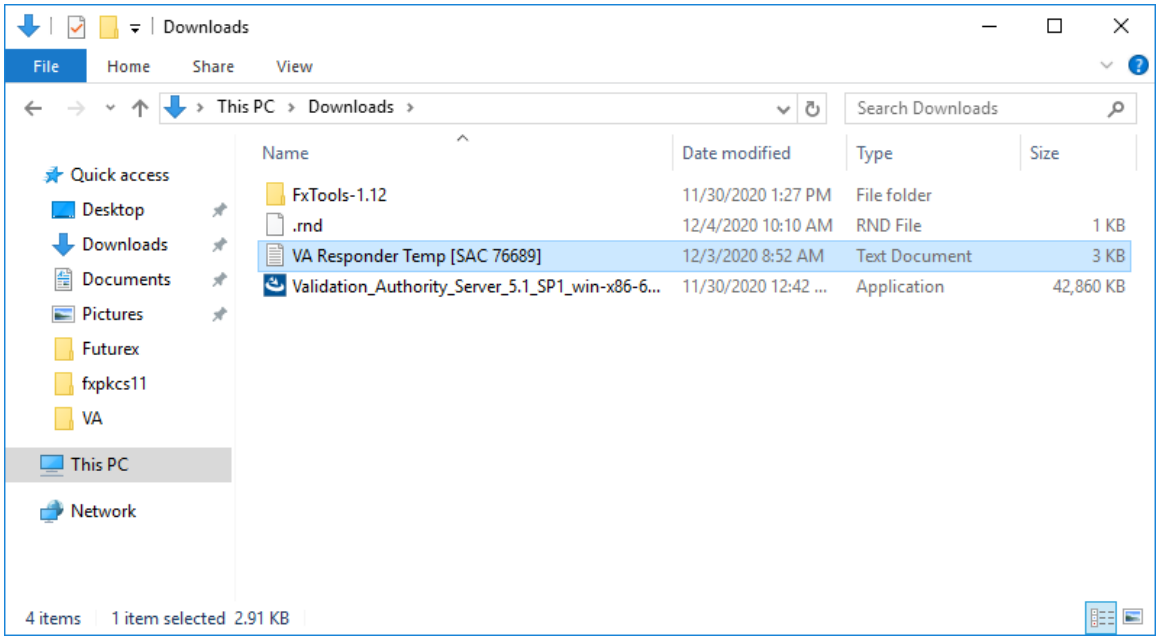


[9.2.2] Install the Responder product license

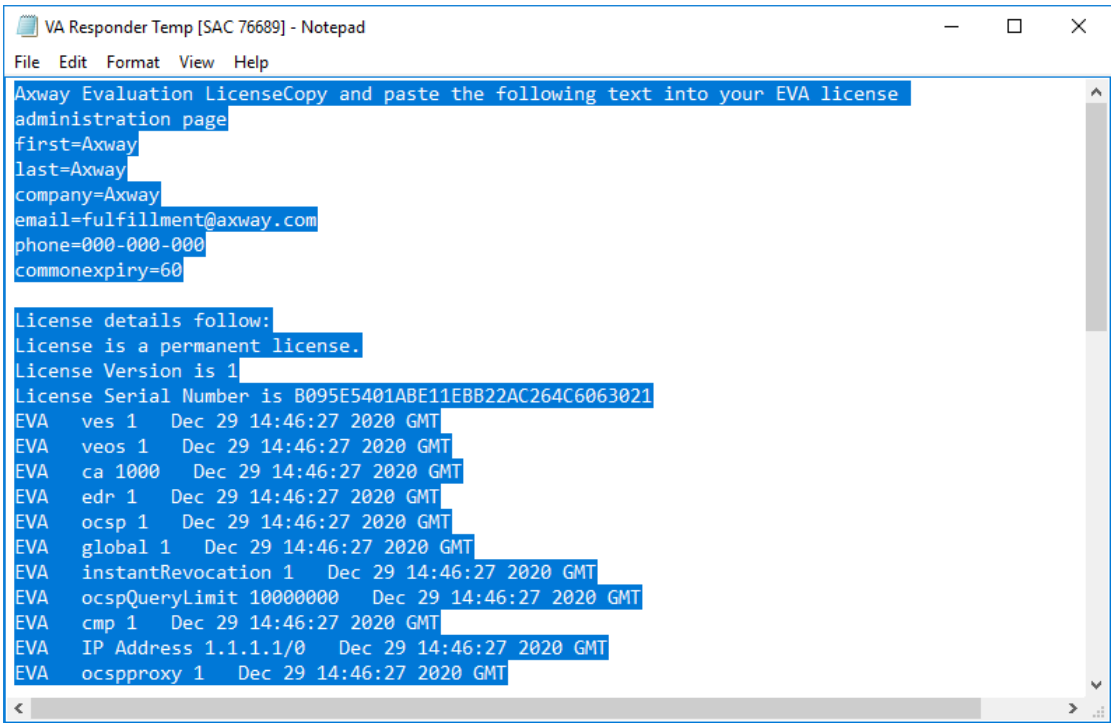
Select the *Enter License* menu on the left. You will see a blank text area where you can paste in a product license.



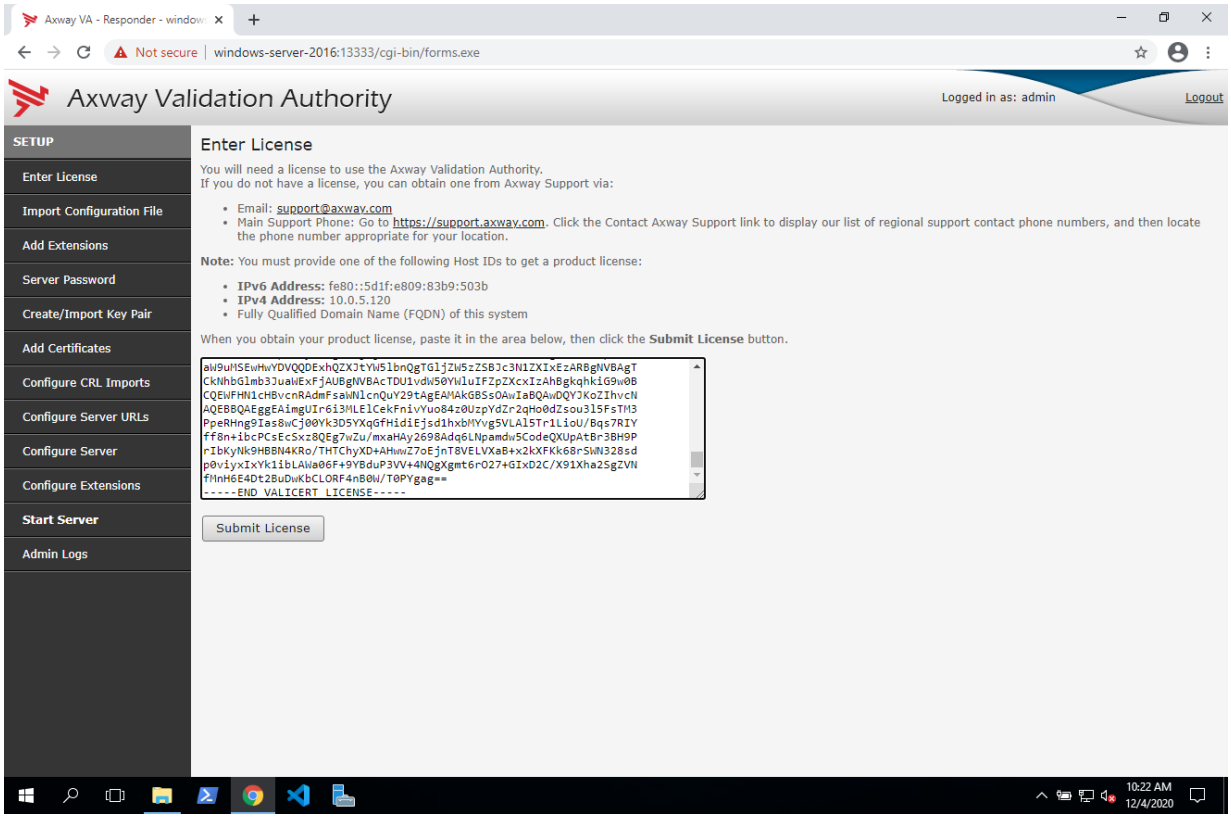
In the file manager for your system, find the "VA Responder Temp" license file that was provided by Axway Global Support.



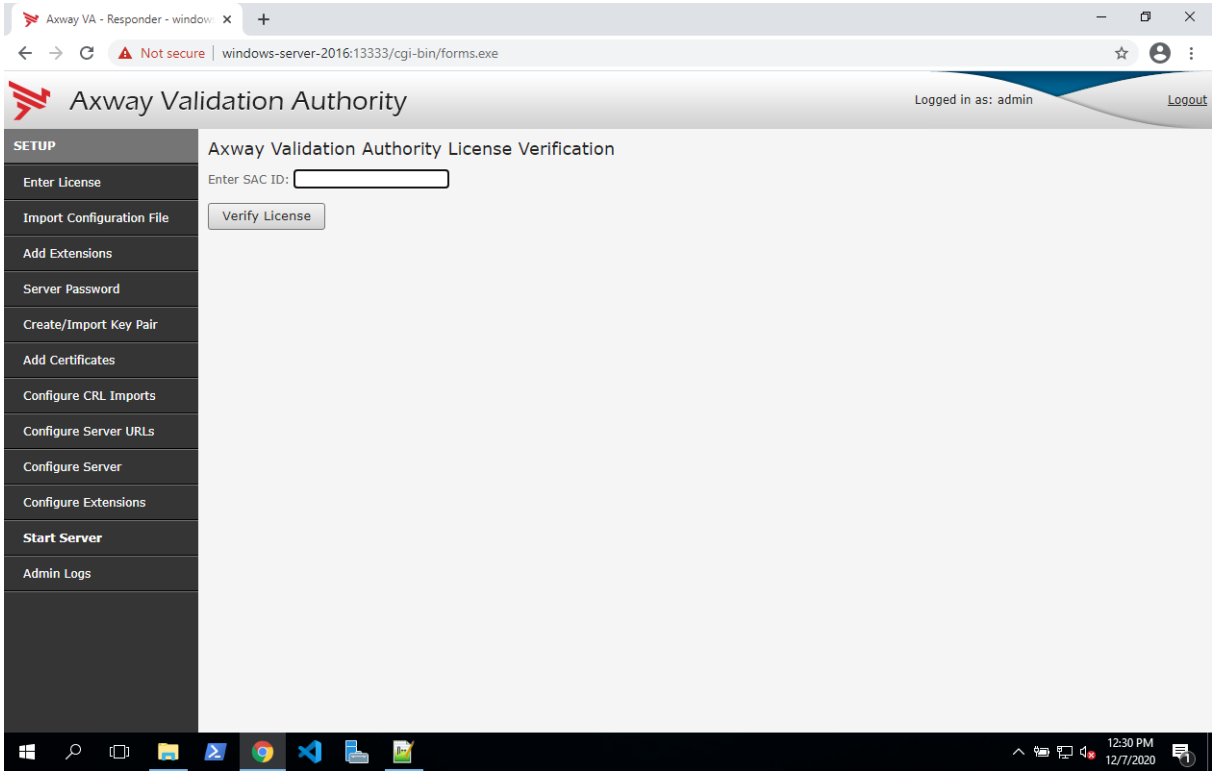
Double-click the "VA Responder Temp" license file to open it. Then type **Ctrl+A** to Select All, then **Ctrl+C** to copy to the clipboard.



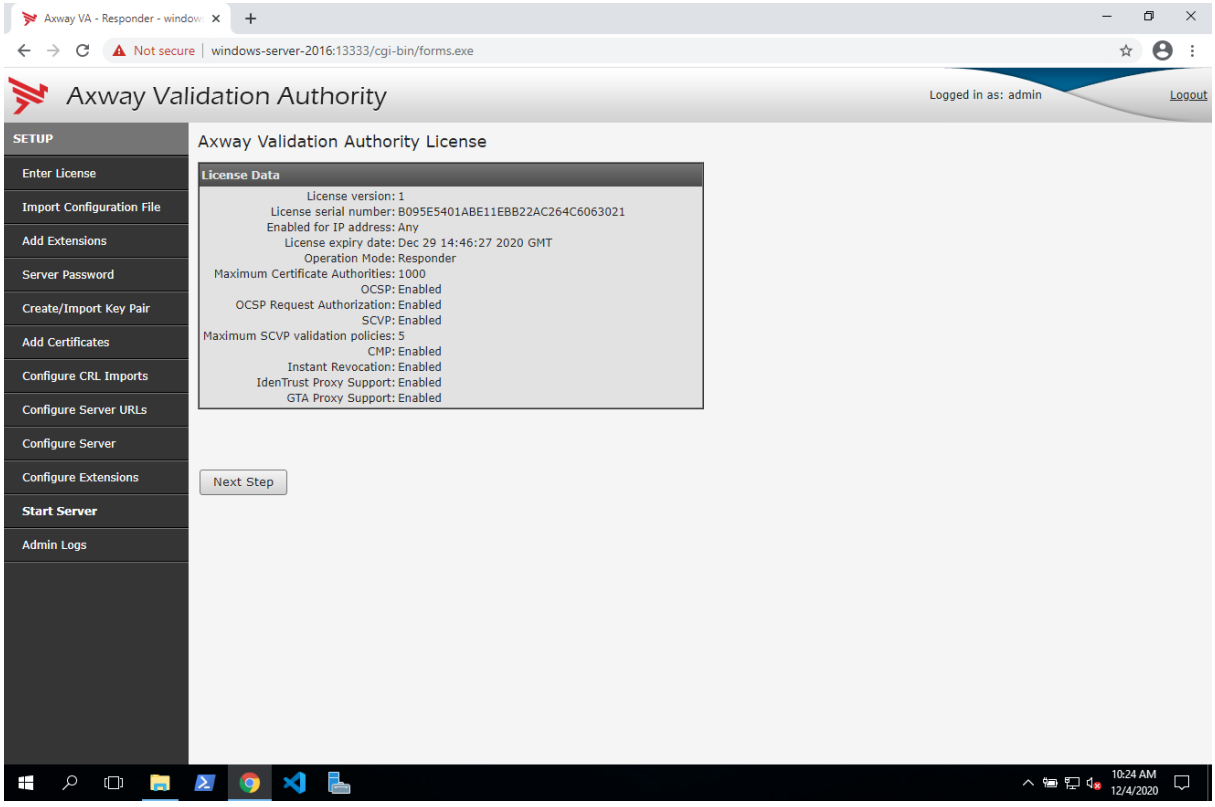
Paste the license information into the blank text area on the *Enter License* page in the admin UI, then click **Submit License**.



Enter the SAC ID that was provided by Axway Global Support, then click **Verify License**.

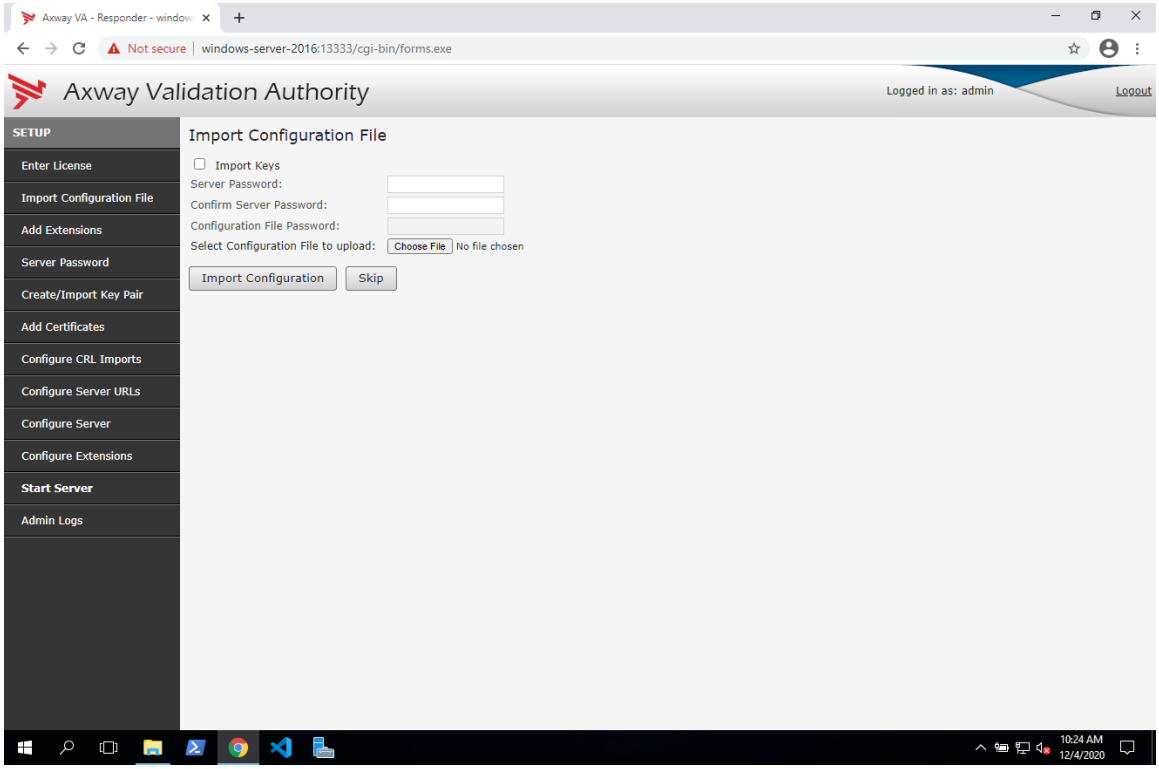


If the submission is successful, the license information will be detailed to review on the *Axway Validation Authority License* page. Click **Next Step** once you have finished reviewing the information.

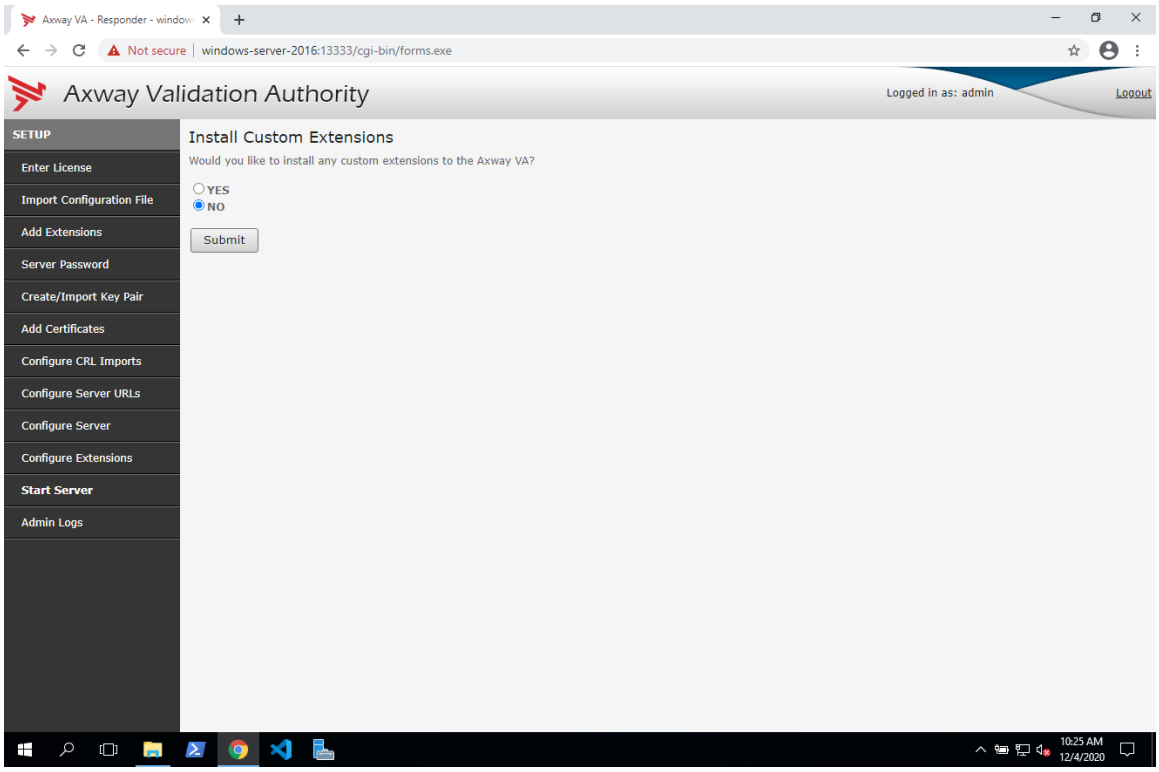


[9.2.3] Bypass optional configurations

On the *Import Configuration File* page, click **Skip**.

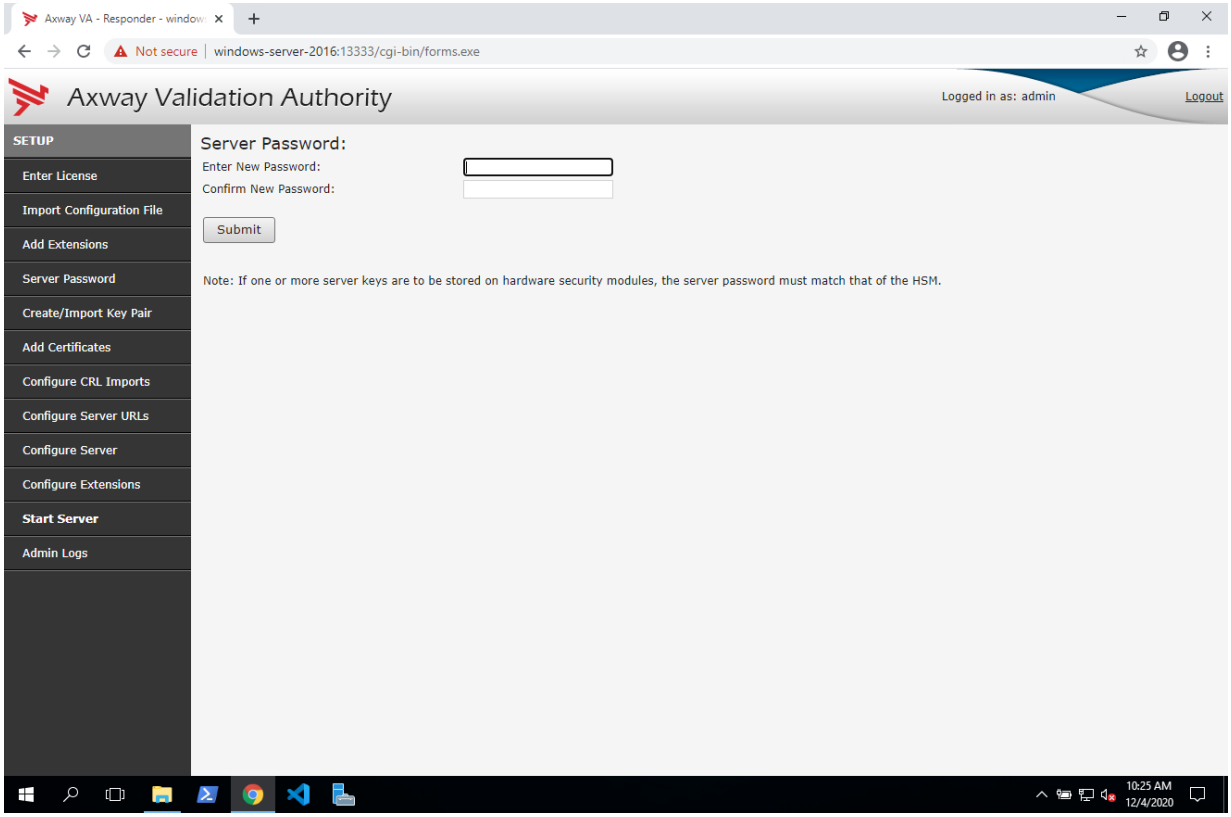


On the *Install Custom Extensions* page, select **NO**, then click **Submit**.

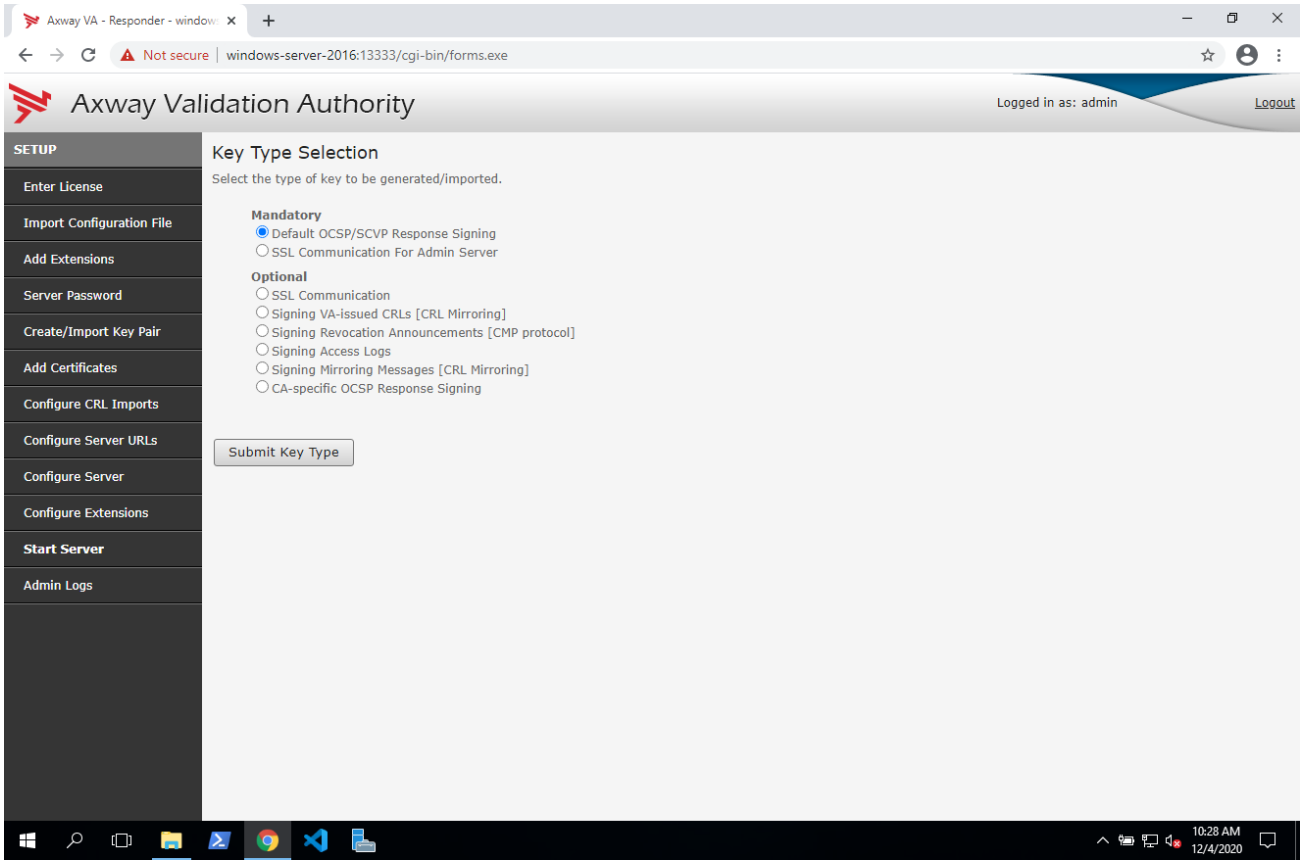


[9.2.4] Change the server password

To prevent unauthorized access to the VA Server, change the server password.



1. If you already created a server password, type it into the **Enter Current Server Password** field. Otherwise, leave the field blank and go to the next step.
2. Type the password you want to use in **Enter New Password**.
   The password must be at least 8 characters long and contain one uppercase, one lowercase, one digit, one special character.
3. Verify the new password by typing it into **Confirm New Password** and click **Submit**.
4. Click **Next Step** to continue with the initial configuration.
   The *Key Type Selection* page displays.
   IMPORTANT: Because you are using VA Server with an HSM device conforming to PKCS #11, you must configure VA Server to use the same password that you assigned to the HSM.

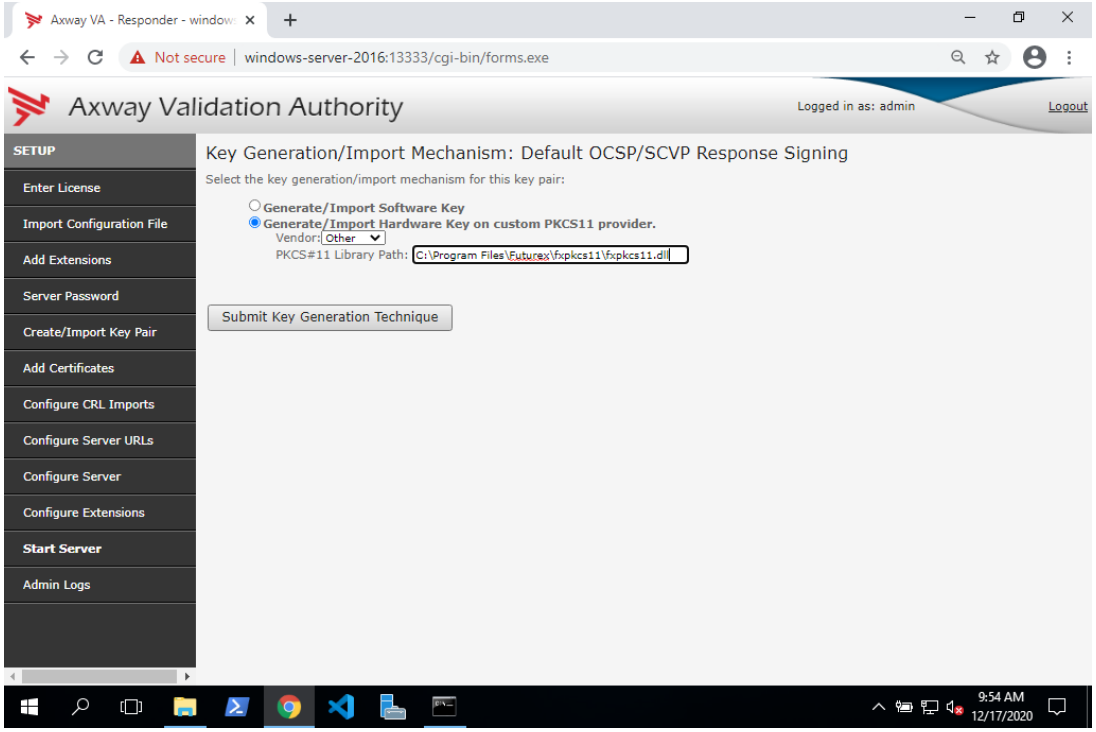


[9.2.5] Create an OCSP and SCVP signing key pair

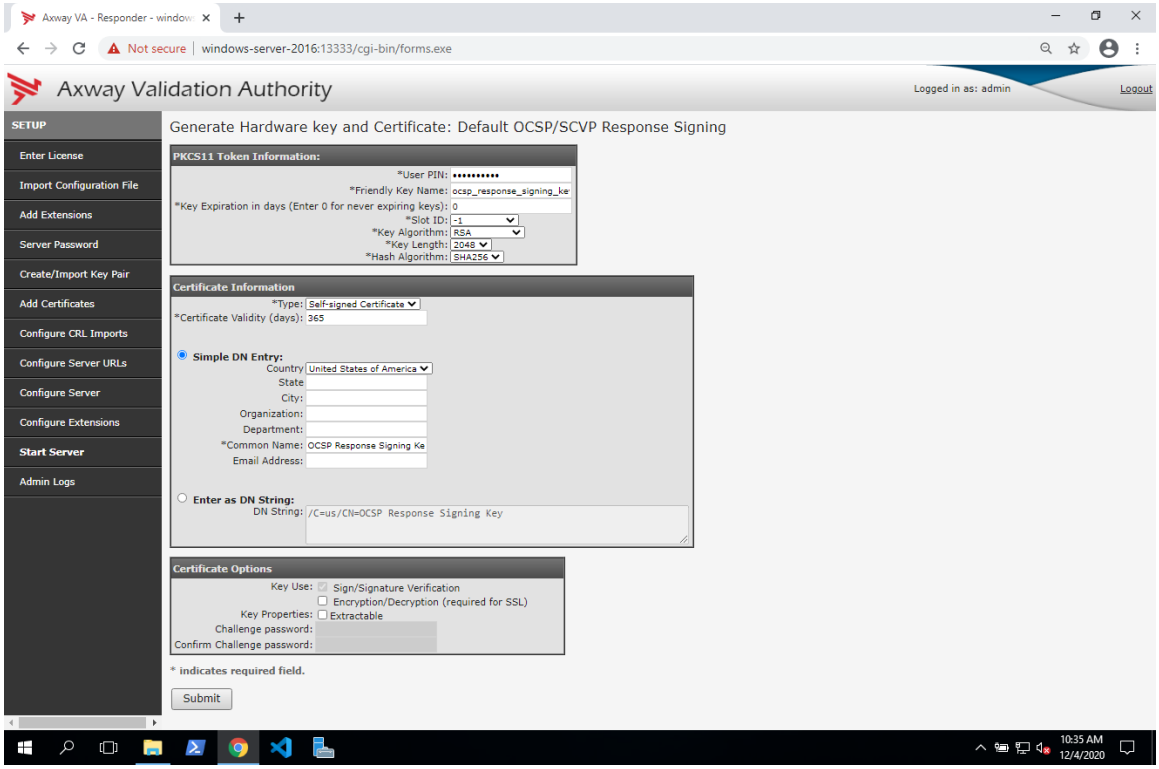
Because it is mandatory for you to generate a public/private key pair for signing OCSP and SCVP responses when operating as a Responder, this key type is assigned as the default.

1. Click **Submit Key Type**.
2. The *Key Generation/Import Mechanism* page displays.

3. Select the **Generate/Import Hardware Key on custom PKCS11 provider** option, set the **Vendor** as "Other", and type in the location of the Futurex PKCS #11 library. Then click **Submit Key Generation Technique**.

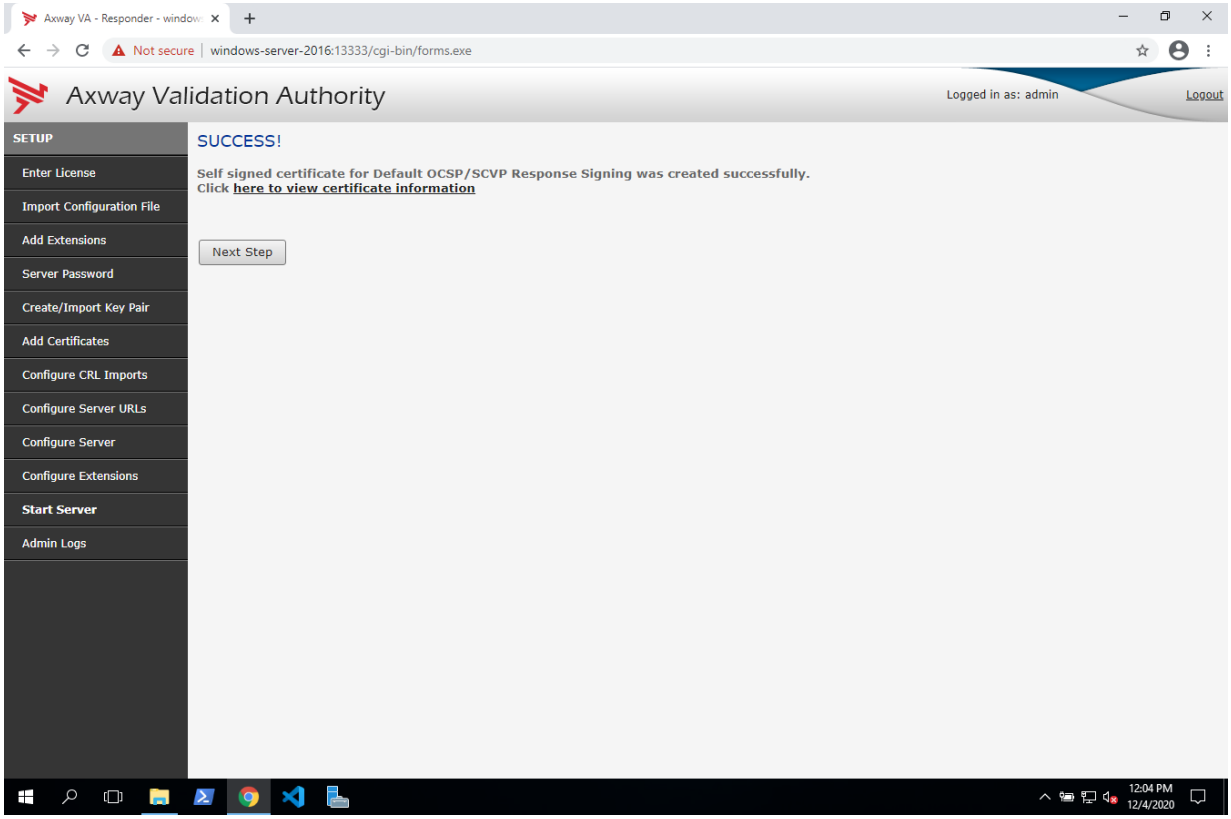


4. For this integration, select **Generate new private key** so that Axway VA will create a key pair on the HSM with the key attributes that it requires.
   This will pull up the following page:

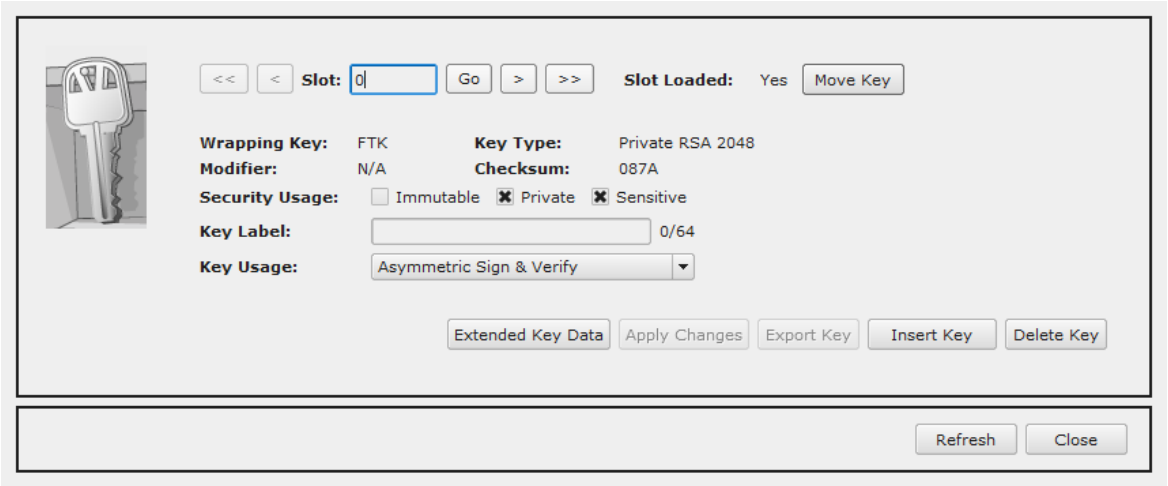


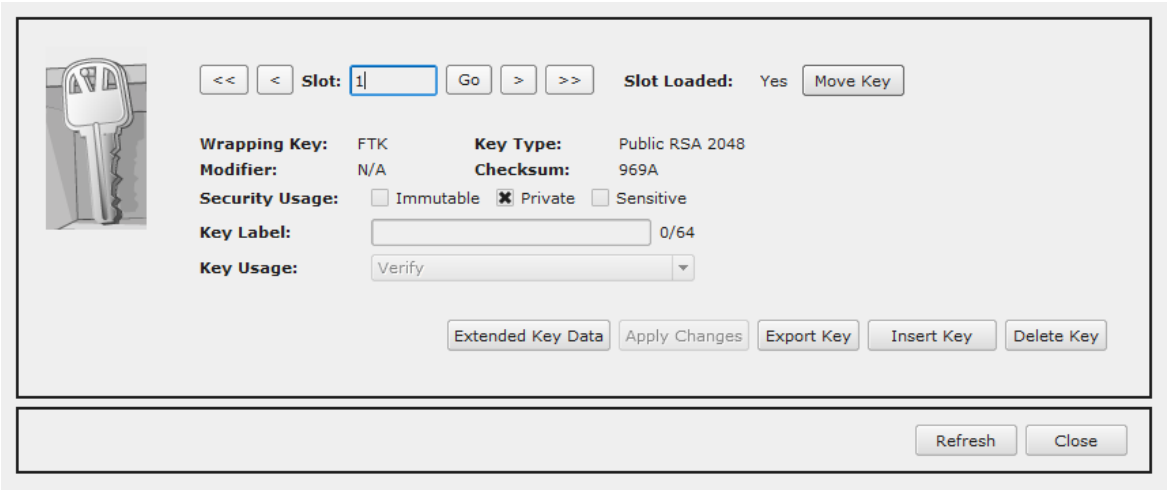
5. Fill in all of the required information, then click **Submit**.
   NOTE: In the **User PIN** field you must specify the password of the HSM Identity that is configured in the Futurex PKCS #11 (FXPKCS11) configuration file.
   NOTE: All of the **Certificate Options** should be left as their default values.

If Axway VA was able to successfully create the OCSP/SCVP Response Signing key on the HSM you will see the following message:



In Excrypt Manager, we can see the two keys that Axway VA created in the Key Storage Table. One of them is a private key, and the other is a public key:





NOTE: The keys can also be viewed using Futurex Command Line Interface (FXCLI), or PKCS11Manager, which comes packaged with the Futurex PKCS #11 (FXPKCS11) installation.

[9.2.6] Configure SSL communication for the admin server

In this section, before performing any configurations in the Axway VA admin UI, we'll first be completing the following actions directly on the HSM using FXCLI:

1. Generate a key pair
2. Export a signing request (CSR)
3. Sign the CSR with a test CA

After these steps are completed, the remaining configurations in this section will be performed on the machine that is running Axway VA.

[9.2.6.1] FXCLI configuration steps

1. Run the **HSM CLI** program.
2. Set the TLS configuration to Anonymous using the following command:

```
$ tls config --anonymous=true
result:
    status: success
    statusCode: 0
tlsConfig:
    anonymous: false
    enabled: false
    verifyDepth: 1
```

NOTE: Anonymous TLS is being used here to help simplify the demonstration. Using Anonymous is not recommended in a production setting. If you choose to connect to the HSM anonymously, you must enable the "Anonymous" setting for the HSM's production port.

3. Connect to the HSM via TCP.

```
$ connect tcp -c 10.0.5.223:9100
[2020-12-07 16:57:12]   INFO   Connected to 10.0.5.223:9100.
[2020-12-07 16:57:12]   INFO   10.0.5.223:9100 handshake successful.
Connected to '10.0.5.223:9100'.
result:
    status: success
    statusCode: 0
```

4. Login with the default admin identities.

```
$ login user
  Username> Admin1
  Password>[2020-12-03 10:53:58]   INFO   Successfully logged in user 'Admin1'.
Successfully logged in as 'Admin1'.
result:
    status: success
    statusCode: 0
dualFactor:
    wanted: false
loggedIn: true
fullyLoggedIn: false
numLogins: 1
loginsRemaining: 1
identities: "Admin1"
roles: "Single Admin"
[2020-12-03 10:53:58]   INFO   Successfully seeded local OpenSSL context with random data.


$ login user
  Username> Admin2
  Password>[2020-12-03 10:54:07]   INFO   Successfully logged in user 'Admin2'.
Successfully logged in as 'Admin2'.
result:
    status: success
    statusCode: 0
dualFactor:
    wanted: false
loggedIn: true
fullyLoggedIn: true
numLogins: 2
loginsRemaining: 0
identities:
    - "Admin1"
    - "Admin2"
roles:
    - "Administrator"
    - "Key Manager"
    - "Operations"
    - "Settings Manager"
    - "Single Admin"
```

5. Create a new key pair in the next available slot on the HSM.

```
$ generate --algo RSA --bits 2048 --name AxwaySslKeyPair --slot next --usage mak
Generated key in board slot.
result:
    status: success
    statusCode: 0
keySlot:
    slot: 2
    name: "AxwaySslKeyPair"
    kcv: "26484FC3"
    algorithm: RSA
    bits: 2048
    usage: Sign,Verify
    startValidity: "1971-01-01 00:00:00"
    endValidity: "2999-01-01 00:00:00"
    exportable: true
    clearExportable: false
    passwordExportable: false
    requiresAuth: false
    modifiable: true
```

6. Add a PKCS #11 label to the private key.

```
keytable extdata --slot 2 --p11-attr label --p11-value "AxwaySslForAdminServer"
```

NOTE: The `generate` command in step 5 set "AxwaySslKeyPair" as the *HSM* label for the key pair. However, Axway VA cannot find the key using the HSM label. It must find it using a *PKCS #11* label. That is why it is necessary to run the `keytable extdata` command above, which sets the PKCS #11 label in a separate field from where the HSM label is set.

7. Generate a certificate signing request (CSR).

```
$ x509 req --private-slot AxwaySslKeyPair --out AxwaySslCSR.pem --dn
'O=Futurex\CN=AxwaySslForAdminServer'
Saved CSR file 'AxwaySslCSR.pem'.
result:
    status: success
    statusCode: 0
request: |-
    -----BEGIN CERTIFICATE REQUEST-----
    MIICeDCCAWACAQAwMzEQMA4GA1UEChMHRnV0dXJleDEfMB0GA1UEAxMWQXh3YXlT
    c2xGb3JBZG1pblNlcnZlcjCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEB
    AMbHeWpulBU5gcZywKJRTAXgu0+aPCSGzORe6K1CDNkFXZ0g2zfYcONCZ6dG5F60
    6M1piEaEHkMNzLBaSn2F1bBvj5ecFBxyAoWmqYsF7R7o+Q7hFr7Qudz0anT09Qqt
    pt885wWcfH6lFhDwtpoT2bMcEmcEUgrlJYgg7NHkJKournhkjBA2CJ06UAHE/qOC
    DXIptWeJOws9mUaU7sNXEDfuwuy9qAoRP4H0dRhT+NL/GUcwu2zcnAMr+UgVXvnw
    NIpIp22/zCDiUyGJmP3mMcBurk9sjnaE3OgCWvbU30crMBtJyUhXFJAlnqcjEHtt
    1v+CxzoZikYimFEors/k+vsCAwEAAaAAMA0GCSqGSIb3DQEBCwUAA4IBAQC0ugsT
    p3MgfmT8VBMCF51M4Dh5J8U3iKNqOESYcr7hCHASzn6jpeom5o5tdZVxnzTfRS5x
    VY0MSSm6WOZUjDJqpAtWczaKJ46Dlfy2kabwec/MZfurgJfcjTRGHnPTuipdwXTk
    0GUTuAraEU+Jg287QHbnMmPyPBWskEKdWT7rgYVvzvF5H6LvtWYPUfHAUTk7OQjW
    MvRE2B5eoe9iDKlD1TjfHXuaqA+bFLyadM/iTLtfRTRoangO6WinRrPDEG8AZwja
    IfyUfmxHalSdInsqefY2u8VGlE4q81V7j1Gsgzc4M3Uq4wkk4zUnT7kpDlCvBTvB
    WcmHLxk2N+bdz/ho
    -----END CERTIFICATE REQUEST-----
```

8. Sign the CSR with a certificate authority (CA) certificate.

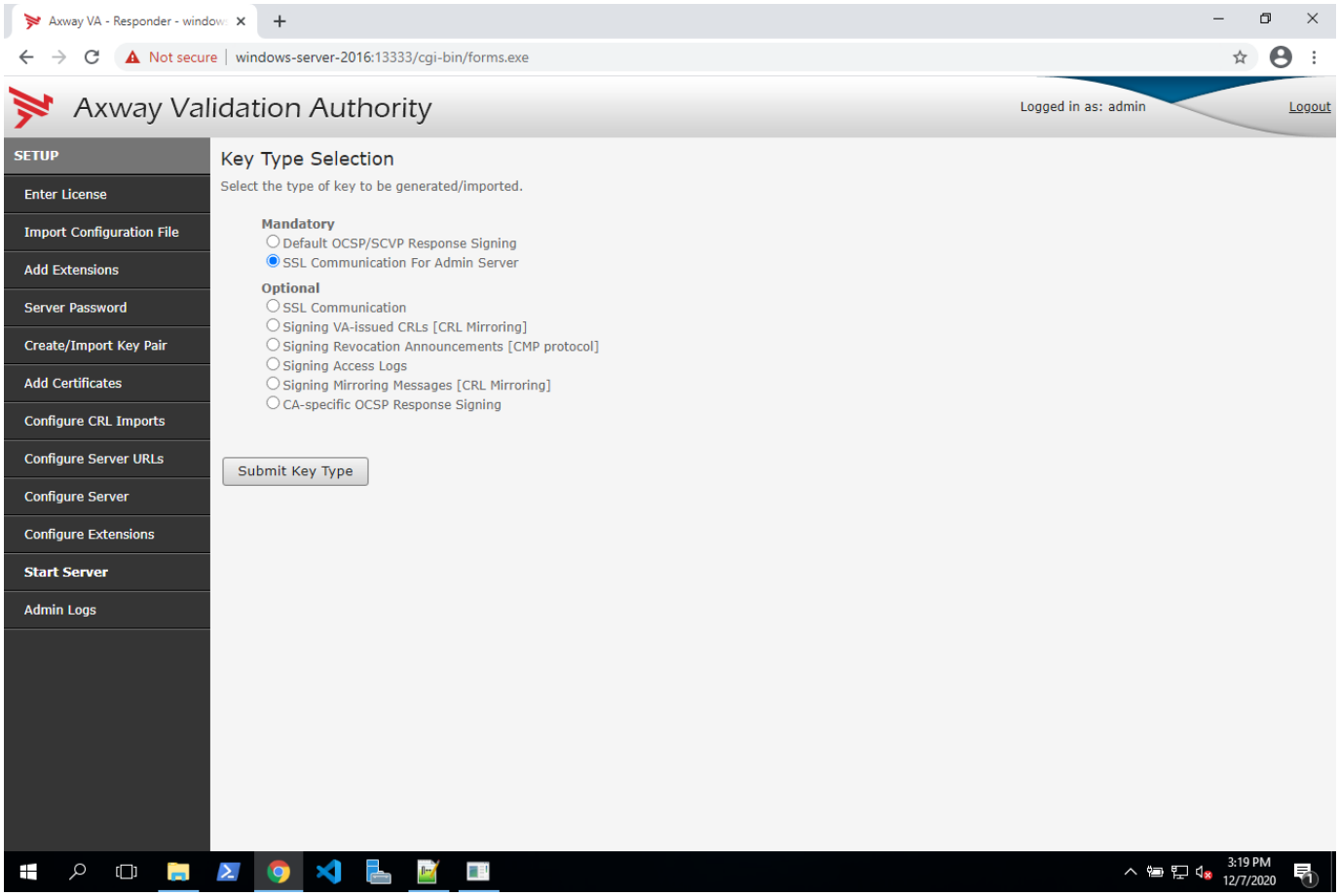
```
$ x509 sign --private-slot 2 --issuer C:\Futurex\sandbox\AxwayTlsCa.pem --csr
C:\Futurex\sandbox\AxwaySslCSR.pem --eku Server --key-usage DigitalSignature --key-usage
KeyAgreement --ca false --dn 'O=Futurex\CN=AxwaySignedSslForAdminServer' --out
C:\Futurex\sandbox\AxwaySignedSsl.pem
Output certificate to 'AxwaySignedSsl.pem'.
result:
    status: success
    statusCode: 0
certificate: |-
    -----BEGIN CERTIFICATE-----
    MIIDBTCCAe2gAwIBAgIJAMWV+GNGM+vSMA0GCSqGSIb3DQEBCwUAMCExEDAOBgNV
    BAoTB0Z1dHVyZXgxDTALBgNVBAMTBFJvb3QwIhgPMjAyMDEyMDIxOTIxMDhaGA8y
    MDIxMTIwMzE5MjEwOFowKzEQMA4GA1UEChMHRnV0dXJleDEXMBUGA1UEAxMOQXh3
    YXlUbHNTZXJ2ZXIwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQC/mUfV
    Pj+DWM9q1iz0k/TbJCWP9OxDIbJ5y4Lg7hVUb5wT3bZ+UrZ57t7bvyR5tQLtFwgj
    fbHrQxwAwPPc3G3AoU0crRPpmA86AXOvVHPm/R4Up5LpweO4AezCZlrGwetkop5z
    QMA824R581p0zxULu5HrpS0Ye0qBcCFJnAKuWUG5HDV0TFLLwyfhRFnOnIvM666Z
    incWYVRTxgrgjMtRKN84M1qqOldF8VuO4ba/CcdVLbIlkMRIfS4S69cTw1r5kYVx
    Do6xrEocgG4NR5nD//mC8oh0jgNRnFMbAEEHOUdOQICdJJ5onjNGq6LcnuOdmP3v
    nR2wzJr1OOOHi9ZZAgMBAAGjMjAwMAwGA1UdEwEB/wQCMAAwCwYDVR0PBAQDAgOI
    MBMGA1UdJQQMMAoGCCsGAQUFBwMBMA0GCSqGSIb3DQEBCwUAA4IBAQC1HlPjr5qV
    kgcGaqDKZmLMJnMstDJzvjZmTv3fV/p15k5Lg8/P14g755PbGmgzC74PntuFAu+A
    GReiIwXxY6hgA6l2+xLA6wigv+HGpuswKO1Q0Q4s3RYqeUwWWFQawE2d/LxK0XVp
    IrkzfzxbuPvHR/Ofy+B0CzHKXFwcckt4lY0TmWqkvWNjrGSY8am6UI8ZPbTnkX5R
    4mN0rqrQ9rJEsvjC06R40AhpC5JFcB5Vvgux7mOf1lEBmry4f+INufQTdT7yQf/3
    8Cur17QNX1E5ImUzG+DwJ9SncJ3hl44fBYLG1pAy7uUyhnATOWcPMC4k6RZkacXk
    ZgPGY1XbWRWM
    -----END CERTIFICATE-----
```

NOTE: The CA certificate that is being used to sign the Axway VA certificate was also created using FXCLI.

Next, we will import this signed certificate in the Axway VA admin UI.

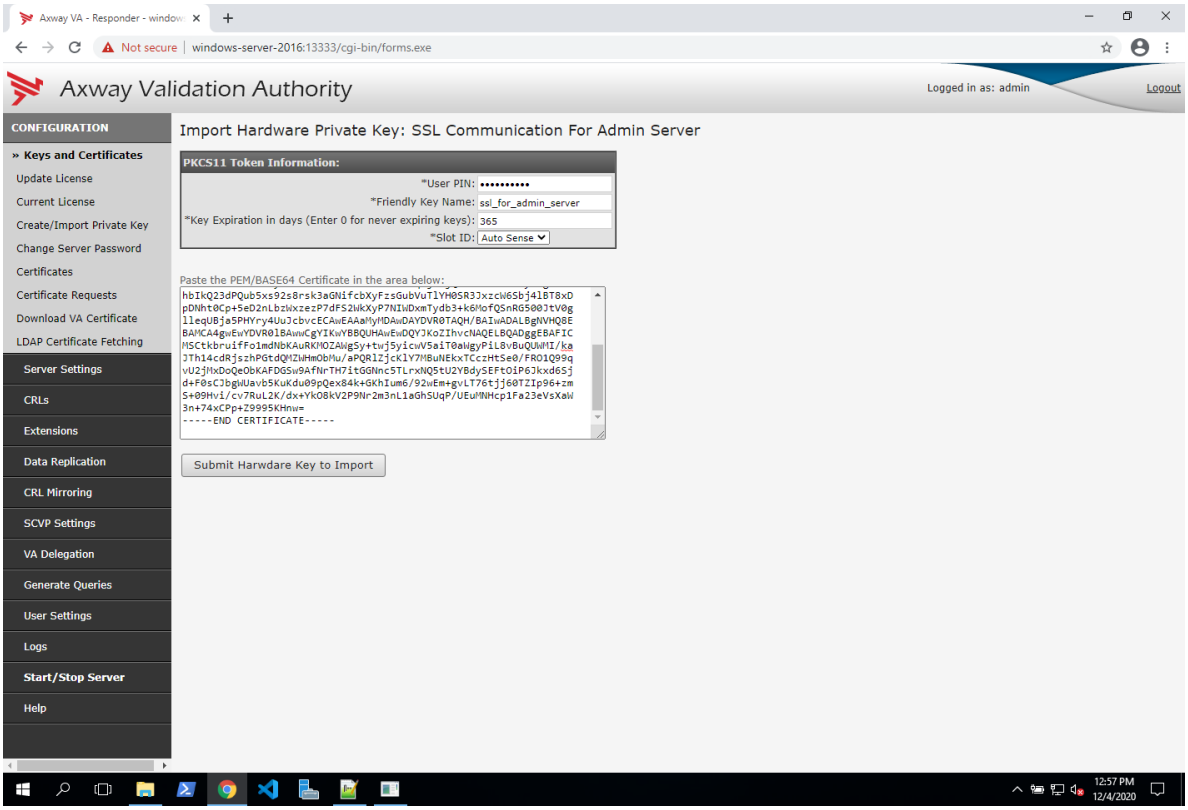
[9.2.6.2] Axway VA configuration steps

1. First, the CA that signed the certificate we'll import needs to be installed on the machine where Axway VA Server is installed. The CA should be installed in the "Trusted Root Certificate Authorities" store for Windows, or your browser's equivalent store.
2. Login to the VA admin UI
3. Go to the *Create/Import Private Key* menu, select **SSL Communication For Admin Server**, then click **Submit Key Type**.

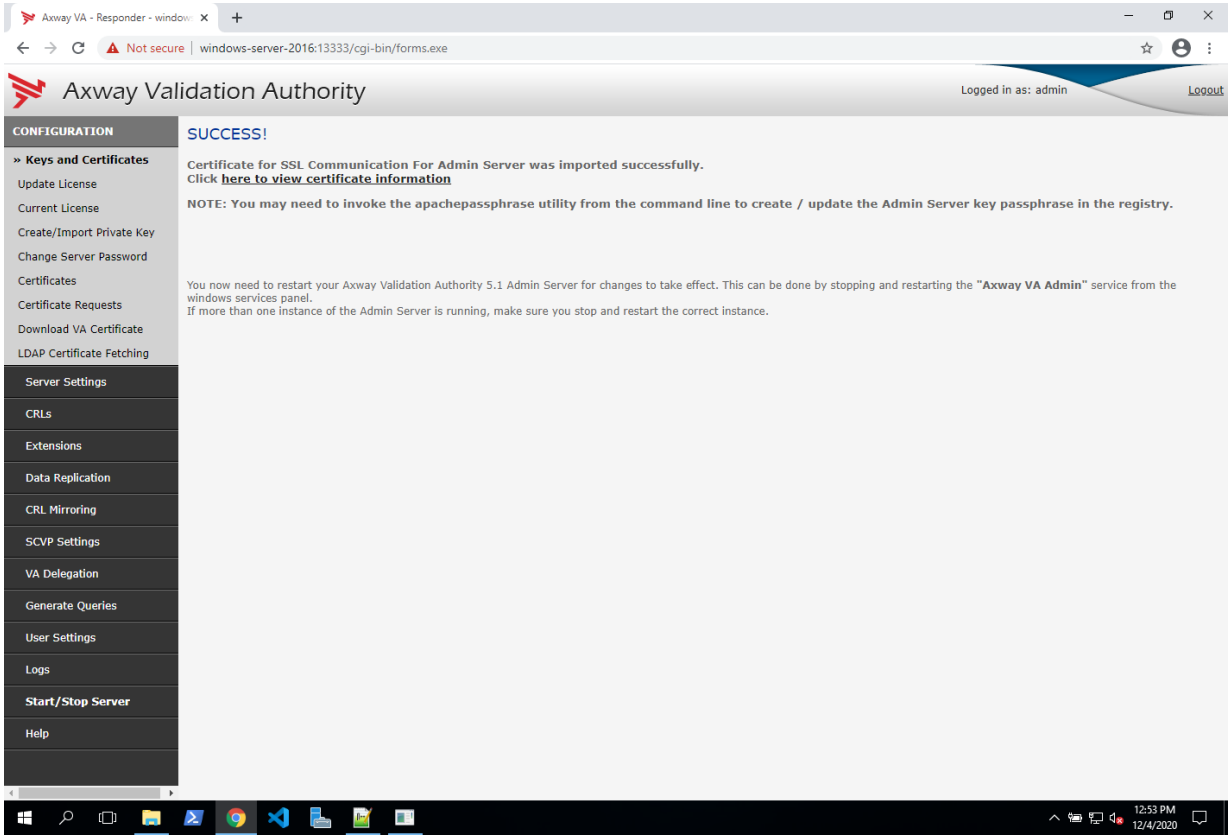


4. For the key generation/import mechanism, select **Hardware Key Generation/Import using Other**, then click **Submit Key Generation Technique**.
5. Select **Import previously generated private key**, then click **Submit Key Generation Or Import**.
6. Fill in all of the **PKCS11 Token Information** fields, paste in the PEM/BASE64 Certificate that we signed in the previous section, then click **Submit Hardware Key to Import**.

   NOTE: In the **Friendly Key Name** field, set the value to the PKCS #11 label of the key. Also, the **Slot ID** field must be set to "Auto Sense". If these two fields are not set correctly, Axway VA will not find the private key associated with the signed certificate on the HSM.



If the certificate import is successful you will see the following message:



7. Start a command prompt as administrator and call apachepassphrase.

```
apachepassphrase -set "<VA Server password>"
```

   This sets the password in the registry. The Apache HTTP server will read it from there using apachepassphrase during startup automatically.

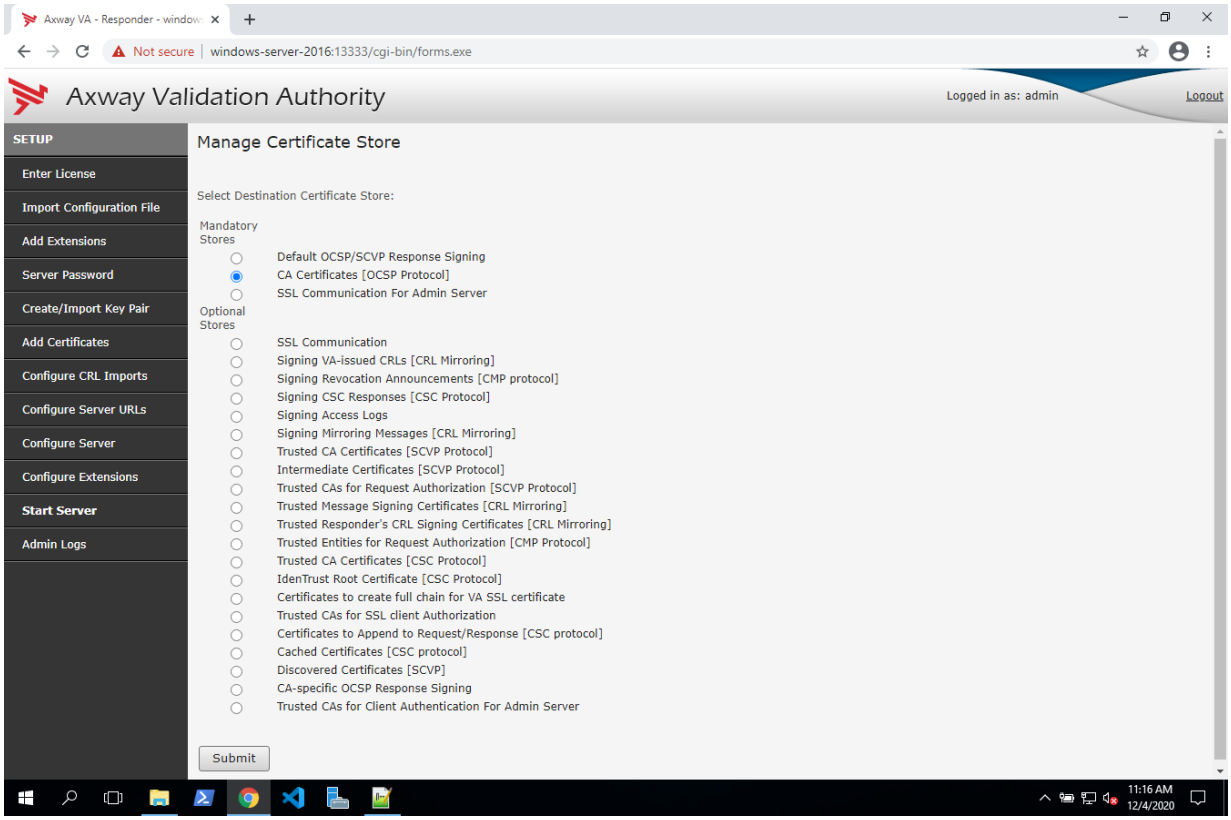
8. Restart “Axway VA Admin” service in the Service Control Panel for changes to take effect.

[10] TEST CRL SIGNING

In this section we will test CRL signing and OCSP Database creation. To simplify this demonstration, we'll pull certificates from a Defense Information Systems Agency (DISA) repository.

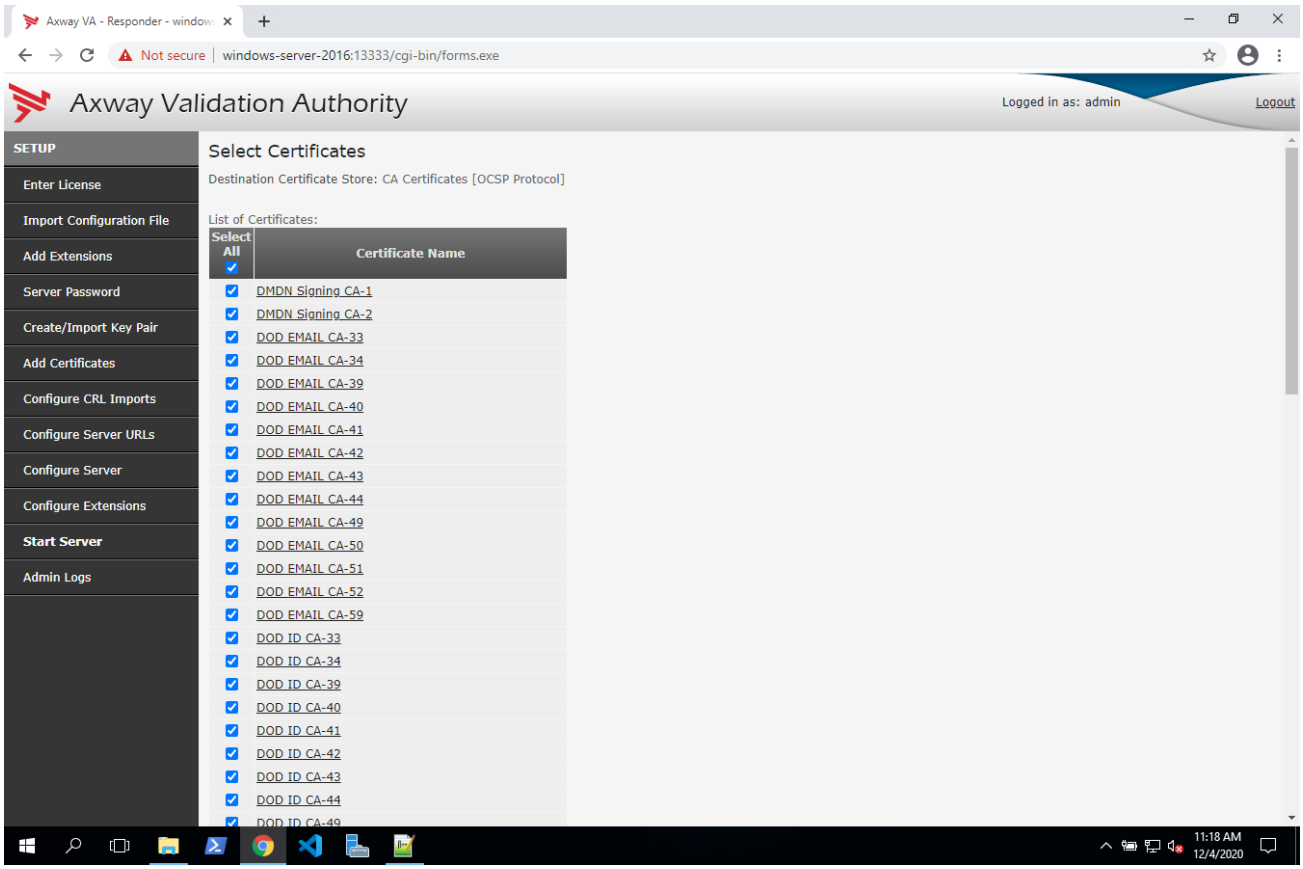
[10.1] PULL CERTIFICATES FROM A DISA LDAP SERVER

1. Go to the *Add Certificates* menu, select **CA Certificates [OCSP Protocol]**, then click **Submit**.



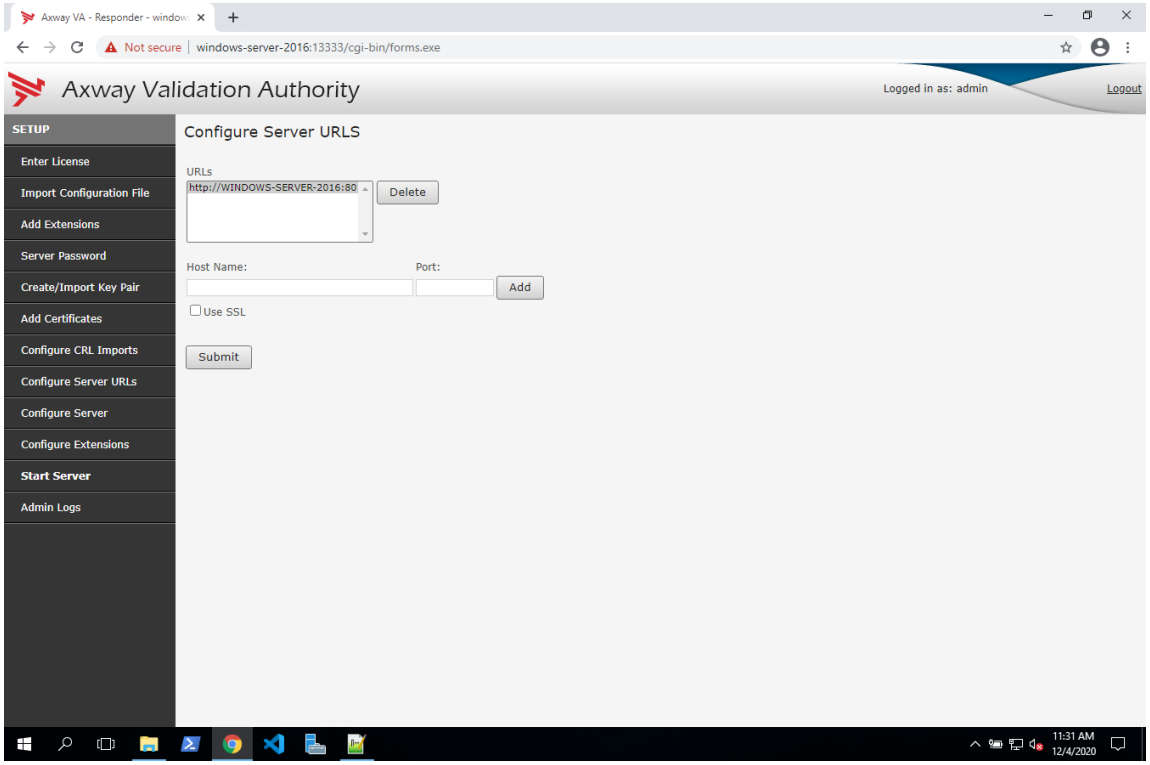
2. Select **LDAP Server**, then click **Submit Certificate Import Method**.
3. On the **Important Certificates from LDAP Server** page, set the **Host Name** to "crl.chamb.disa.mil". Leave all of the other fields as default and click **Get LDAP Certificates**.
   NOTE: At the time of this writing, DISA supports port 389 for importing certificates from their LDAP server. However, recently they announced that soon they will only support Secure LDAP (LDAPS), which uses port 636. If port 389 does not work for you, attempt to use port 636 anonymously instead.

4. If the VA Server connects to the LDAP server successfully, you will see a list of certificates on the next page. Scroll to the bottom and click **Submit Certificates**.

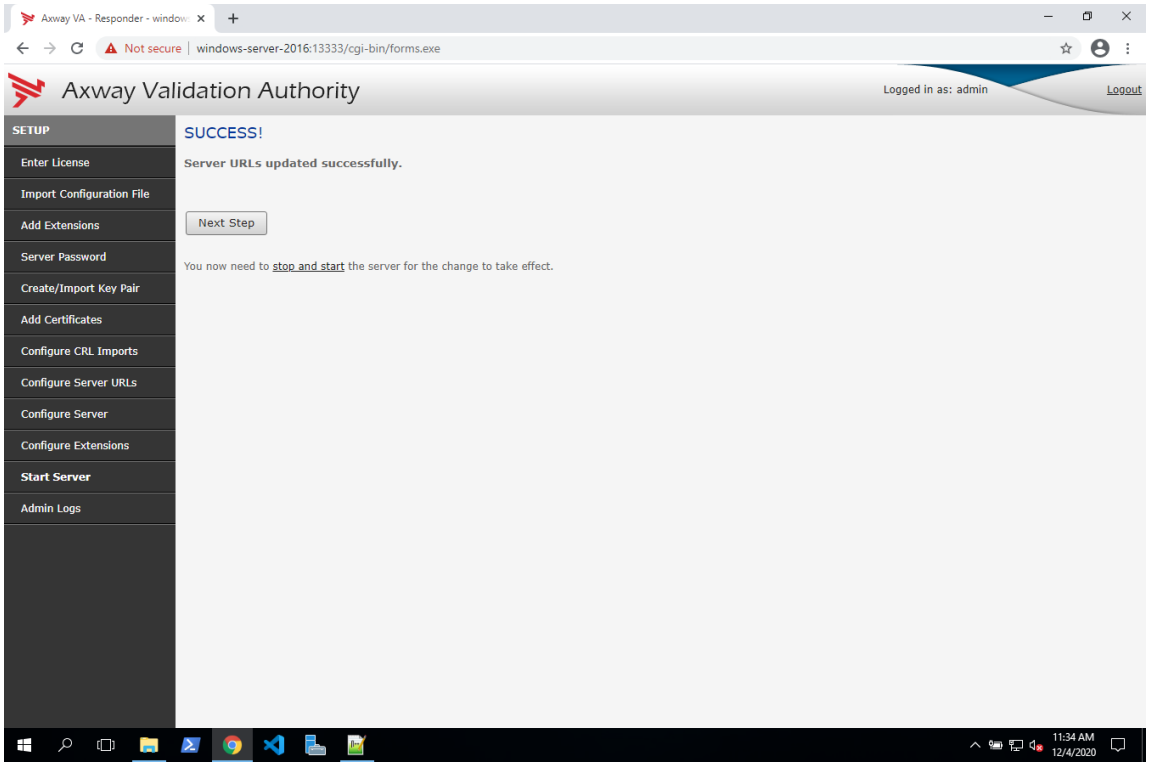


5. Expect to see the following error:
   ```
   Failed to import one or more certificates. Please refer to admin logs for details
   ```
   This can be disregarded. It just means that at least one certificate out of approximately 50 failed to load. Click **Go Back**.
6. Scroll to the bottom of the Configure VA Certificate Store page and click **Next Step**.
7. On the Configure CRL Imports page, leave **in an LDAP Directory** selected as the CRL Source and click **Add CRL Source**.
8. On the **Configure CRL Import (LDAP)** page we'll see that the **LDAP Host** field is auto-populated with the address we previously entered. Leave all of the fields as-is and click **Find Available CRLs** at the bottom.
9. Scroll to the bottom of the **Available CRLs for Import** page and click **Schedule Import of Checked CRLS**.
10. Click **Next Step** on the **Configure CRL Imports** page.
11. On the **Configure Server URLS** page everything can be left as default as long as port 80 is available on the machine (by default the server URL is configured to use port 80). If port 80 is taken you can either free it up so that it can be used by Axway VA, or you can configure a different port. Once you've finished configuring the server URLs, click **Submit**.
    NOTE: On Windows, sometimes the IIS service will have port 80 reserved. On Linux, sometimes the

Apache service will have port 80 reserved.



If the request is successful you will see the following message:

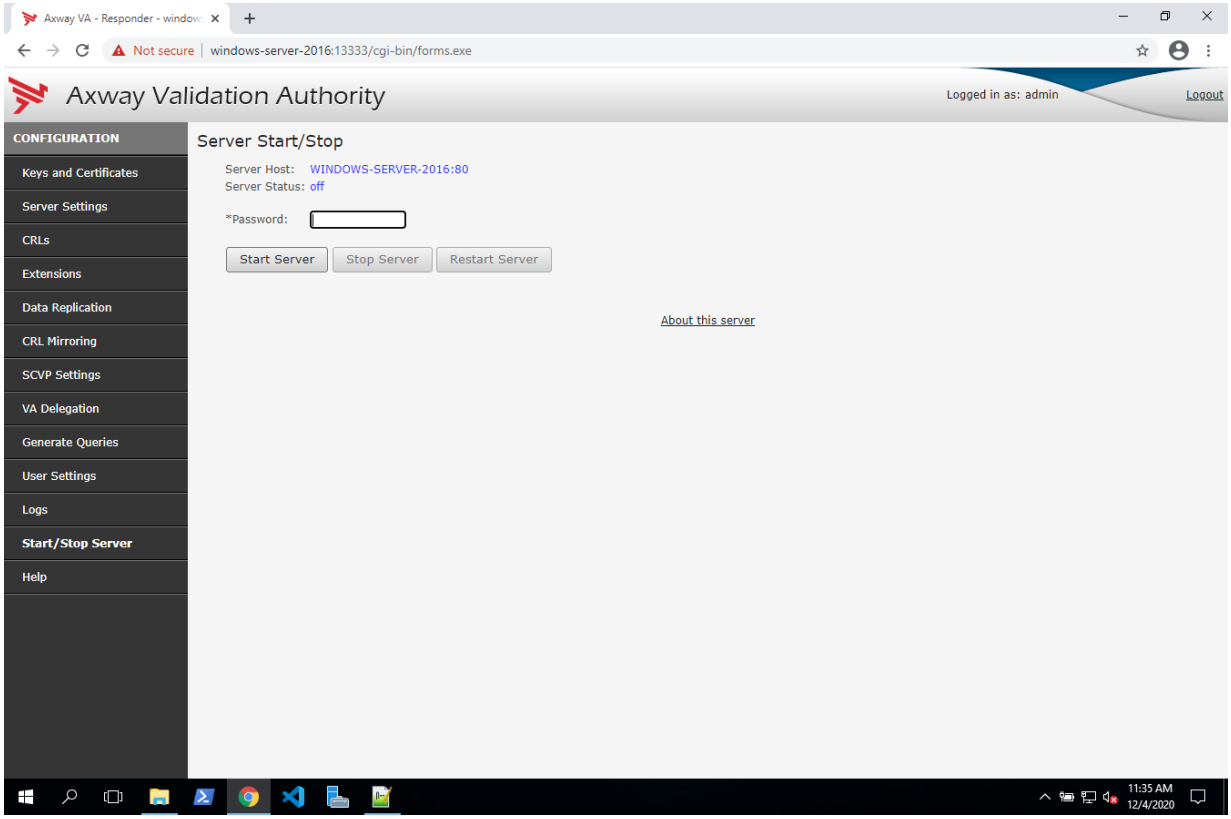


12. Click **Next Step**.

13. Leave all of the settings as default on the **VA Responder Server Configuration Parameters** page and click **Submit Configuration Parameters**.
    You should see a message that says:
    ```
    The Server configuration has been successfully updated.
    ```
    Click **Next Step**.

[10.2] START THE SERVER

On the *Start/Stop Server* page, type in the password of the server, then click **Start Server**.



If the server starts successfully the **Start Server** button will become grayed out and the **Stop Server** button will become clickable.

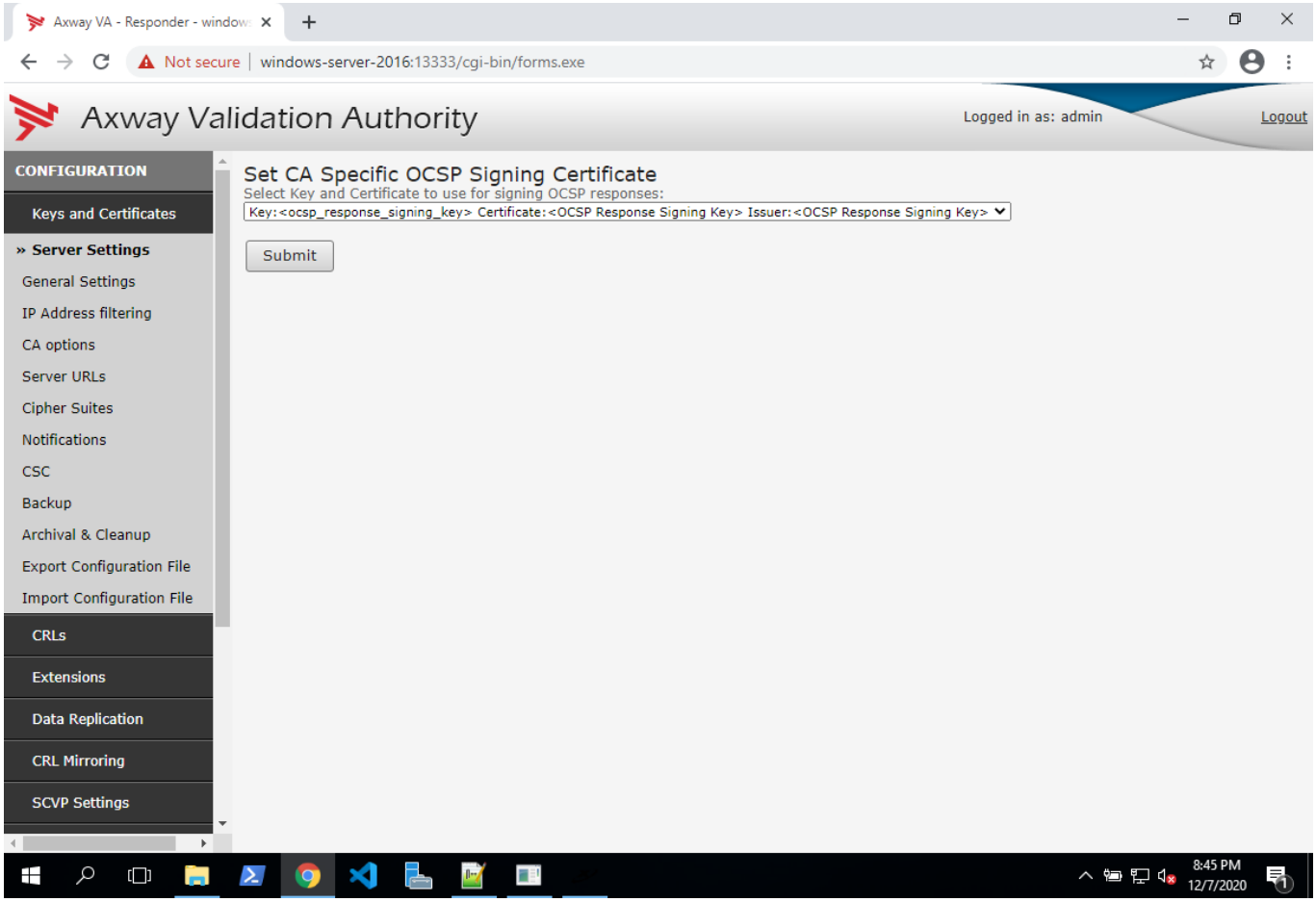
[10.3] TEST CRL SIGNING AND OCSP DATABASE CREATION

1. Go to *Server Settings -> CA options*.
2. Select the **DOD EMAIL CA-41 CA**, then click **Configure CA Options** at the top of the page.
3. On the **VA Responder CA Options Configuration** page there are two settings that need to be modified:

a. Under **OCSP Response Settings**, change the **Validity period of CRL** to the next **7 days**.

b. Under **Pre-computation Options**, check the **Pre-compute OCSP Data** checkbox, then select **Only Revoked Certificates**.

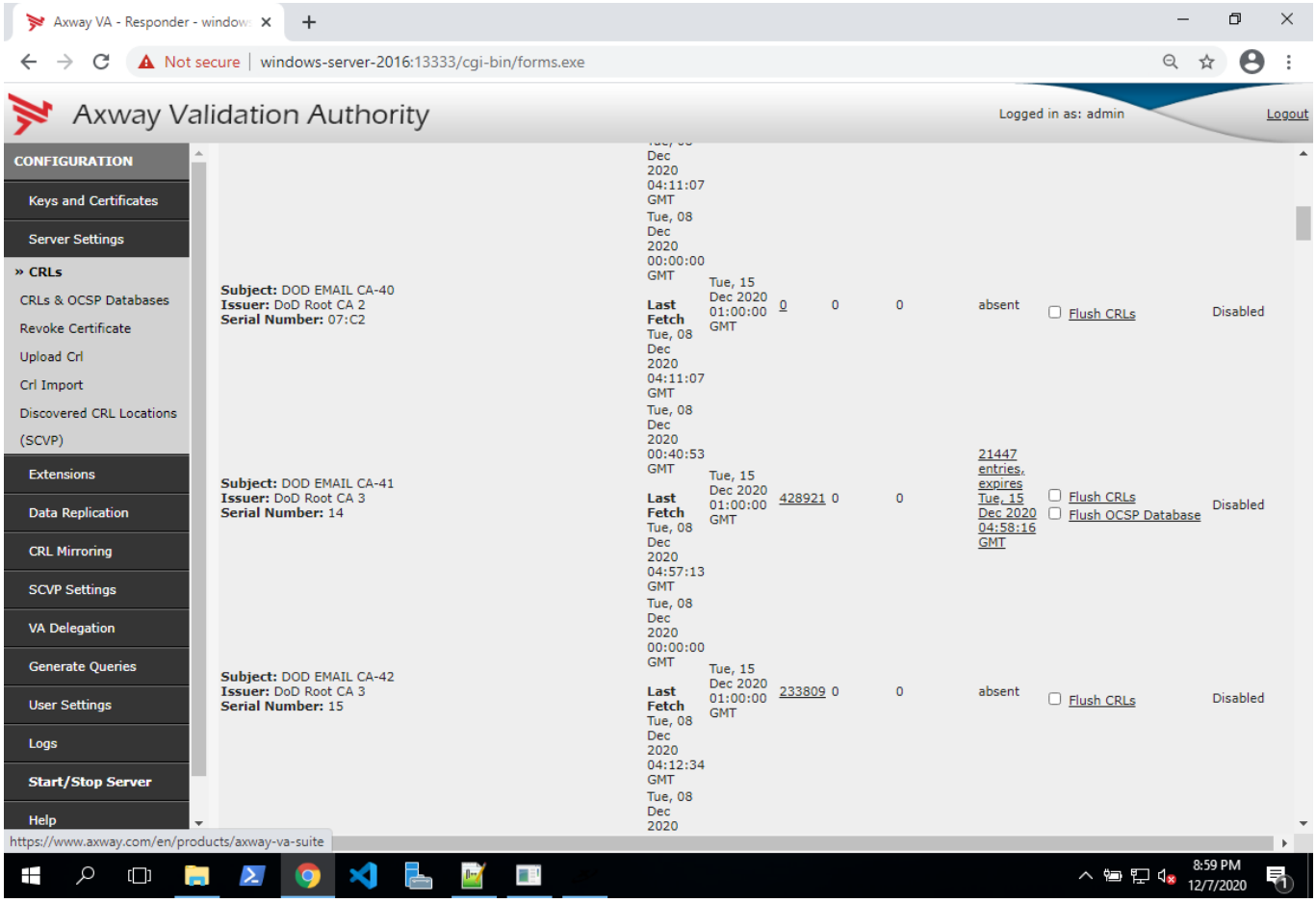
4. Click **Submit CA Configuration Parameters** at the bottom of the page. You should see a message that the CA configuration options have been successfully modified.
5. Go back to *Server Settings -> CA options*.
6. Select the **DOD EMAIL CA-41 CA**, then click **Configure CA Specific OCSP Signing Certificate** at the top of the page.
7. On the **Set CA Specific OCSP Signing Certificate** page we can see the OCSP signing key that we created earlier on the HSM.



Click **Submit** and you should see a message saying that it successfully set CA Specific OCSP Signing certificate/key.

8. Go to the **Start/Stop Server** page, enter the password, then click **Stop Server**.
9. Go to *CRLs -> CRLs & OCSP Databases*. Find **DOD EMAIL CA-41** and click **Flush CRLs**.
10. Disregard the warning and proceed by clicking **Flush CRL and OCSP DB Information**. You should see a message that the CRLs and OCSP databases for the specified CA have been cleaned successfully.

11. Go to the **Start/Stop Server** page, enter the password, then click **Start Server**.
12. Go to *CRLs -> CRLs & OCSP Databases*. Find **DOD EMAIL CA-41** and in the **OCSP response database** field you should something similar to the following once the CRLs have finished downloading and the OCSP database has been successfully created.



This result confirms that VA Server was able to use the OCSP response signing key stored on the HSM to sign the CRLs that were downloaded for DOD EMAIL CA-41.

APPENDIX A: USING THE GUARDIAN SERIES 3 TO CONFIGURE THE HSM

[10.4] SETTING UP THE GUARDIAN SERIES 3 TO MANAGE CLIENT FUTUREX HSM'S

If a user has multiple HSMs, the Guardian Series 3 can be used to create and manage device groups, provide load balancing, configuration management capabilities, peering, redundancy, and notifications for client Futurex devices.

Preconditions for Futurex Device Group Configuration Through the Guardian Series 3

In order to connect client Futurex HSMs for management by the Guardian Series 3, a number of preconditions for all of the involved HSMs must be met.

NOTE: Futurex certificates will be used for the connection between the Guardian Series 3 and the HSMs in the following sections. Futurex certificates are preloaded on every unit. There is a private key and associated signed-certificate, which is signed under a Customer "X" Futurex TLS CA tree. In conjunction with a client certificate signed under the same CA, these certificates can be used for secure communications with a Futurex unit without the need for generating and managing certificates on a customer-managed CA. If you wish to utilize a user CA, please refer to the relevant Administrator's guide.

Preconditions for Client Futurex HSMs

1. The HSM must be network-attached, with an IP address configured and an Ethernet cable plugged into a local area network.
2. If using user certificates, the HSM must have a major key loaded. If Futurex certificates are utilized this precondition does not apply.
3. If using TLS between the HSM and the Guardian Series 3, the HSM must have the proper TLS settings enabled. If a mutually authenticated connection is to be established, these settings must match on the Guardian Series 3. Otherwise, selecting this connection type will result in a failure to add the device to the group.
4. The HSM must be signed using the same root certificate as the Guardian Series 3. This is automatic if using Futurex certificates.
5. The HSM must have the same date and time settings as the Guardian Series 3, as well as other units in the device group. The date and time settings are synced automatically when you sign in to the Device Group on the Guardian Series 3, so no user configuration is required for this.
6. All HSMs in the device group must be of the same model, and they must have the same firmware version and feature set.

Preconditions for Guardian Series 3

In order to add a client Futurex HSM to a device group, the following preconditions must first be met.

1. The Guardian Series 3 must be network-attached, with an IP address configured and an Ethernet cable plugged into a local area network.

2. If using user certificates, the Guardian Series 3 must have a major key loaded. If Futurex certificates are utilized this precondition does not apply.
3. If using TLS between the Guardian Series 3 and HSM, the Guardian Series 3 must have the proper TLS settings enabled. If a mutually authenticated connection is to be established, these settings must match on all client HSMs. Otherwise, selecting the connection type will result in a failure to add the device to the group.
4. The Guardian Series 3 must be signed using the same root certificate as the client Futurex device. This is automatic if using Futurex certificates.
5. The Guardian Series 3 should have the same date and time settings as all units in the device group. The date and time settings are synced automatically when you sign in to the Device Group on the Guardian Series 3, so no user configuration is required for this.
6. The Guardian-required Host API commands must be enabled.

Creating a Client Futurex Device Group

Device groups help simplify the management of information on multiple client Futurex devices by controlling them through a single interface. The devices need to be associated with groups in order to harness the Guardian Series 3 for replication, synchronization, load balancing, monitoring, failover, and alerting features. Use the following procedures to create a device group and add devices.

1. Select Encryption Devices from the left toolbar. Click the Add Group button at the bottom of the window to open the Encryption Device Group window.

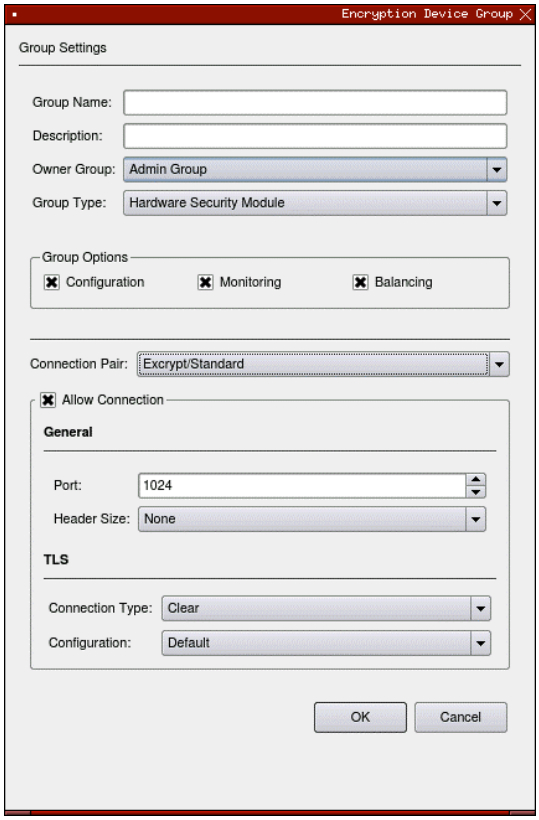


FIGURE: ENCRYPTION DEVICE GROUP WINDOW

2. Enter a Group Name in the associated field.
3. Enter a Description of the group in the associated field.
4. Select the desired Owner Group from the drop-down menu.
5. Select the Group Type.
    - For this use case you will select **Hardware Security Module**: Excrypt SSP9000, Excrypt SSP9000 Enterprise, Excrypt Plus, Excrypt SSP Enterprise v.2, or Vectera Plus devices.

    NOTE: As mentioned previously, devices in the Hardware Security Module group may only be added to groups of like devices.

6. Define Group Options.
    - **Configuration**: Allows you to remotely configure all Futurex HSMs in group.
    - **Monitoring**: Allows you to monitor all Futurex HSMs in group.
    - **Balancing**: API calls sent to this group will be load-balanced between all devices in the group.
7. Choose the Connection Pair using the drop-down menu. The connection pairs available will vary depending on the type of device group. For PKCS #11, only the Excrypt/Standard connection pair is needed. The HTTP and International connection pairs should be disabled.
    - **Excrypt/Standard**: used to connect with the Excrypt or Standard APIs for transaction processing using Futurex HSMs
    - **HTTP**: used to connect with the client Futurex device's web management portal, or the Registration Authority in the case of KMES Series units with Registration Authority functionality enabled, or to the device's RESTful web API
    - **International**: the connection pair used to connect with the International API for transaction processing using Futurex HSMs, when the Excrypt Universal Interface license is enabled
8. Check Allow Connection and choose the Port and Header Size, if applicable.
9. Select the Connection Type for each connection pair from the drop-down menu. The options are Clear, SSL, or Anonymous TLS, but **SSL** should be used and is the default.
10. Click OK to create the group.

Adding Devices to a Device Group

How to Add a Device to a Device Group

Groups are defined by device type. When selecting a device to add, chose the group of the same model, as it is not possible to mix and match different devices within the same group.

1. Select the group to add the client device to.
2. Click the Add Device button at the bottom of the screen. The Encryption Device window will appear.

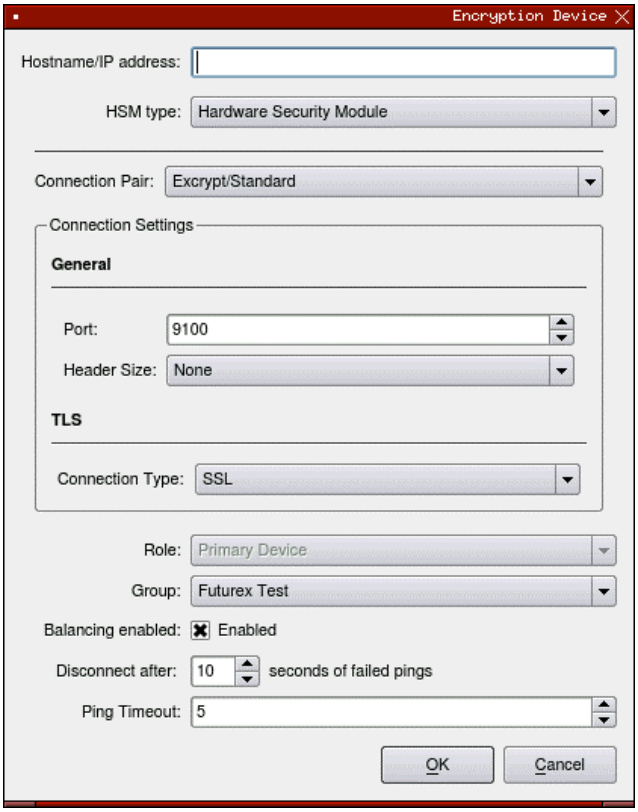


FIGURE: ENCRYPTION DEVICE WINDOW

3. Enter the Hostname of IP address of the client device.

   NOTE: HSMs managed by the Guardian Series 3 in a single group must be using the same firmware version and feature set.

   NOTE: All of the remaining settings in this menu (steps 4-13) should be kept as default if using Futurex certificates.

4. Select the Connection Pair using the drop-down menu. This allows you to set the proper TLS pair for the device in question.
5. Define the Port that the client devices are configured to operate on. There is no need to specify a Header Size.
6. Designate the desired Connection Type and Configuration using the drop-down menus.
7. Select the Role of the device from the associated drop-down menu. This specifies the device's use in the assigned group. Only the Primary Device role will be available for the first device added to the group.

   NOTE: The differences between the 3 main device role types are described below:

   - Primary Device – Designates a device as a primary device in the device group. The configuration details on this device will automatically be replicated to any additional devices added to the device group. The primary device also functions in the same role as a production device.
   - Production Device – Designating a device as a production device will cause it to begin actively processing transactions as soon as it has been synchronized with the group. Multiple production devices may be added to an individual device group.

- **Backup Device** – Designating a device as a backup device will cause it to remain synchronized with the group, but not process transactions, until a production device is removed from service, at which point it will automatically begin processing transactions. The use of backup devices is optional, and multiple backup devices may be added to an individual device group.

8. Select the desired Group from the drop-down menu.
9. To enable balancing, check the box next to Balancing Enabled. This allows the Guardian to evenly distribute requests to devices in the group.
10. Set the number of seconds of failed pings before the Guardian considers the device to be disconnected.
11. Set the desired number of seconds for the ping timeout. The ping timeout is the amount of time before an individual ping is open.
12. Click OK to save changes.

The Details window will open, displaying the connection status for the device, as well as the connection details. Users will be given the option to export this information once the process is complete.

This window can also be reopened by right-clicking on the encryption device and selecting Show Connection Status.

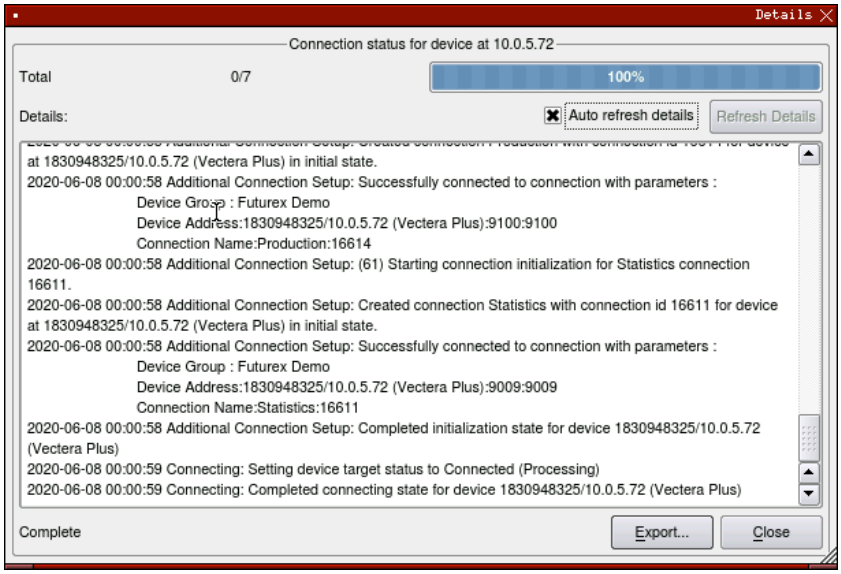


FIGURE: CONNECTION STATUS DETAILS

Troubleshooting Failed Connections

If the connection is failing these are some of the things that you should check:

- Is the Device Group and Device enabled?
- Are the Admin and Excrypt TLS ports configured on the HSM?
- Are the Guardian Series 3 and the HSM using the same CA tree? If using Futurex certificates, they both need to be utilizing either RSA or ECC CA.

NOTE: If port 9100 is failing to connect, there is a problem with the Excrypt port configuration. If port 9009 is failing to connect, there is a problem with the Admin port configuration.

[10.5] CONFIGURING THE HSM THROUGH THE GUARDIAN

Load Futurex Key

For this step you will need to be logged in with an identity that has a role with permissions ***Major Keys:Load****. The default Administrator role and Admin identities can be used.*

The FTK is used to wrap all keys stored on the HSM used with PKCS #11. If using multiple HSMs in a cluster, the same FTK can be used for syncing HSMs. Before an HSM can be used with PKCS #11, it must have an FTK.

Note that this process can also be completed using the Excrypt Manager, FXCLI, the Excrypt Touch or the Guardian Series 3. The instructions that follow will be for the Guardian Series 3. For more information about how to load the FTK into an HSM using the other tools/devices, please see the relevant Administrative Guide.

After logging in, go to the *Encryption Devices* page. Then, right-click on the device group and select "Remote Manage…".

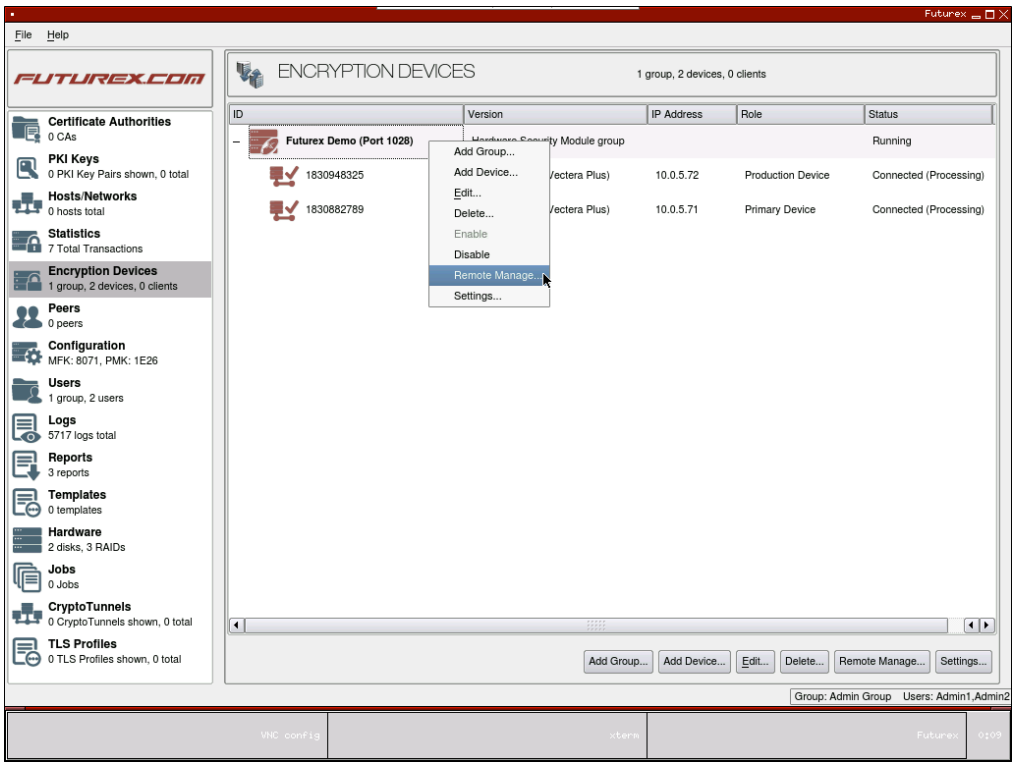


FIGURE: REMOTE MANAGE OPTION

This will pull up the login screen, from which you can log in to the selected device. Once logged in, select **Keys** in the left-hand menu. This will bring you to the **Major Keys** tab. Once there, click on "Load" next to the FTK.

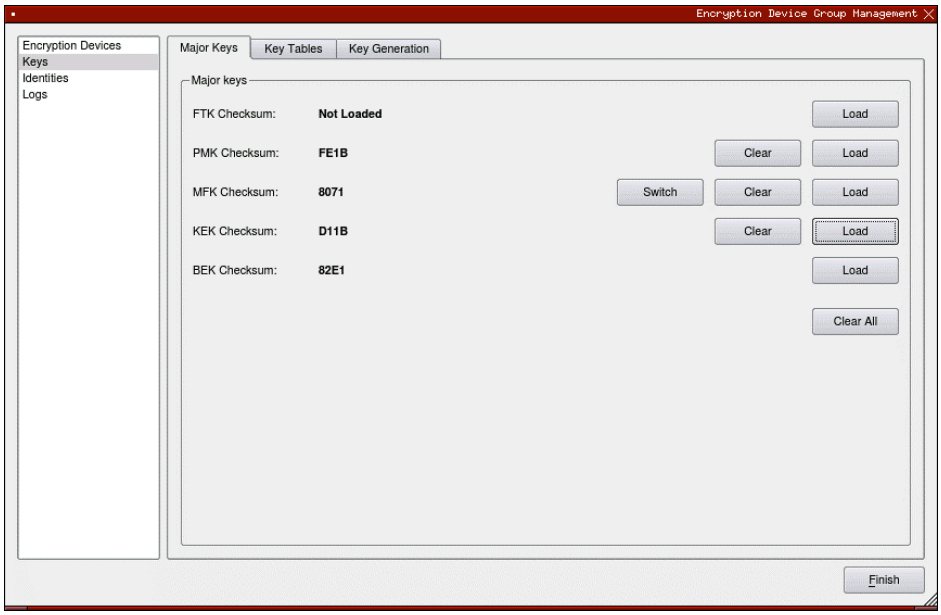


FIGURE: MAJOR KEYS TAB

The first menu in the wizard will have you select the Algorithm, Key length, and Key parts that you want to use for the key that you're loading. Then you will load each of the key parts. For each of the key parts, you will receive confirmation that it was loaded successfully.

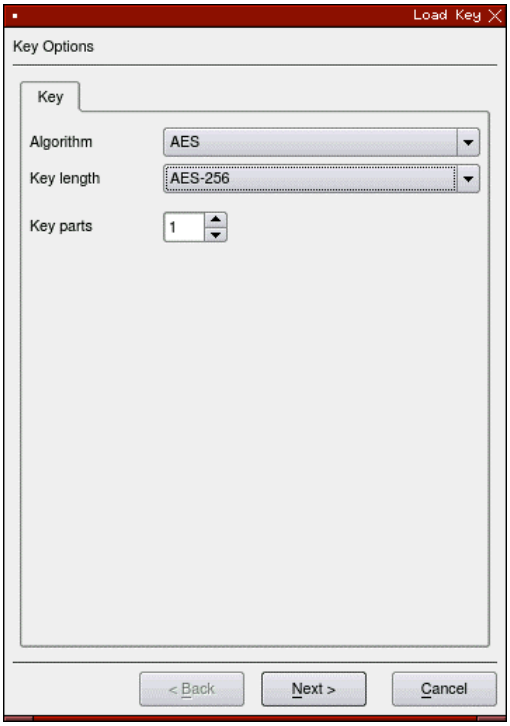


FIGURE: KEY OPTIONS IN LOAD KEY WINDOW

After all key parts have been loaded, you will receive a Final Key Checksum.

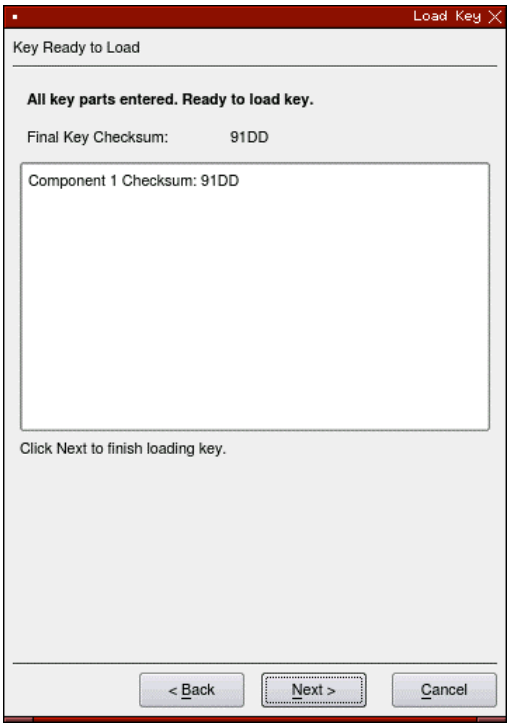


FIGURE: FINAL KEY CHECKSUM IN LOAD KEY WINDOW

After clicking “Next” on the previous screen, the dialogue below will confirm that the key was created successfully.

Configure a Transaction Processing Connection

For this step you will need to be logged in with an identity that has a role with permissions ***Role:Add****,* ***Role:Assign All Permissions****,* ***Role:Modify****,* ***Keys:All Slots****, and* ***Command Settings:Excrypt****. The default Administrator role and Admin identities can be used.*

NOTE: For the purposes of this integration guide you can consider the terms "Application Partition" and "Role" to be synonymous. For more information regarding Application Partitions, Roles, and Identities, please refer to the relevant Administrator's guide.

Configure a Transaction Processing Connection

Before an application logs in to the HSM with an authenticated user, it first connects as an unauthenticated user under the “Anonymous” Application Partition. For this reason, it is necessary to take steps to harden the “Anonymous” Application Partition. These three things need to be configured for the “Anonymous” partition:

1. It should not have access to the “All Slots” permissions.
2. It should not have access to any key slots.
3. Only the PKCS #11 communication commands should be enabled.

While still logged in to the Device Group, navigate to the Identities menu, and then the Application Partition Management tab.

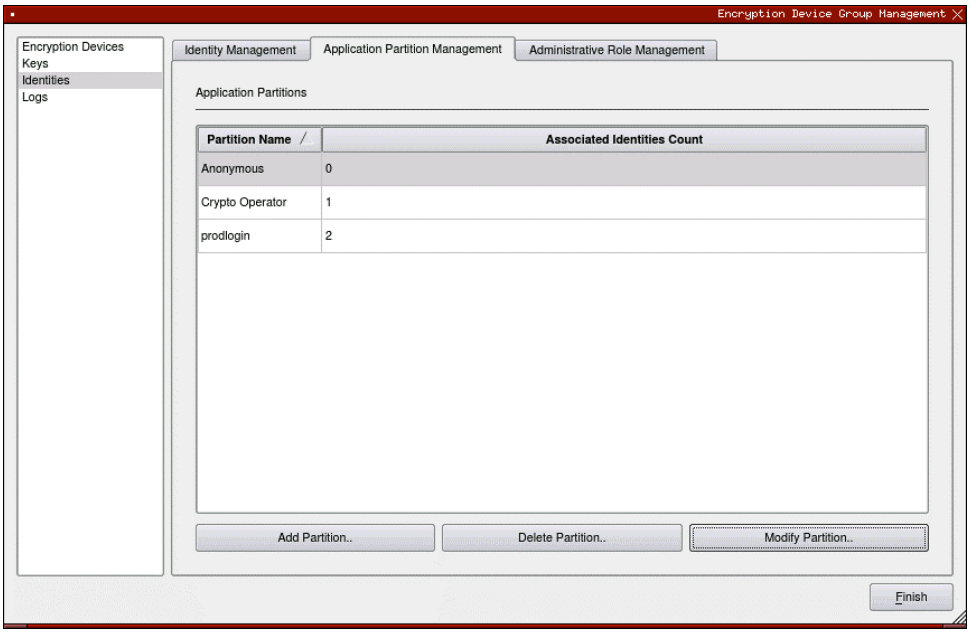


FIGURE: APPLICATION PARTITION MANAGEMENT TAB

Select the "Anonymous" Application Partition, and click *Modify Partition*, which will pull up this menu.

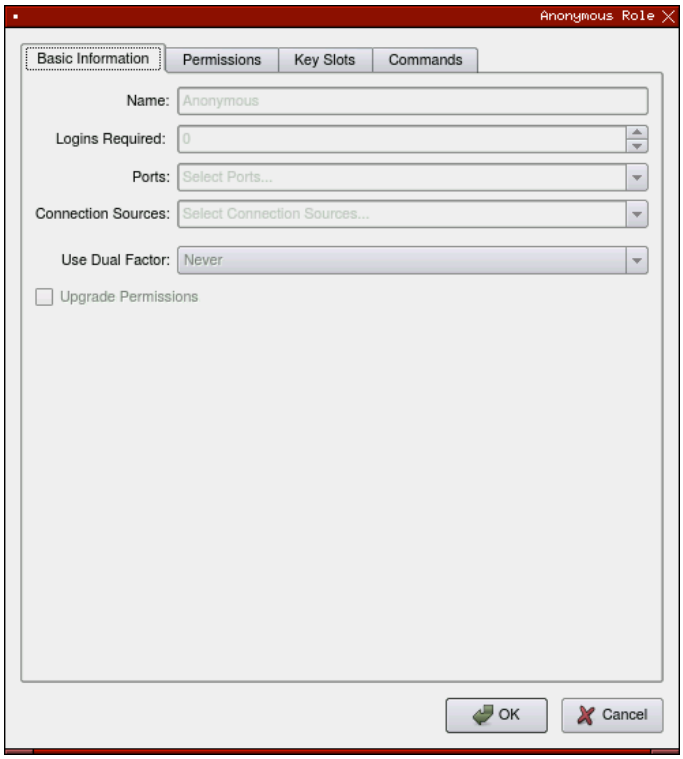


FIGURE: BASIC INFORMATION IN THE ANONYMOUS ROLE WINDOW

Navigate to the “Permissions” tab and ensure that the “All Slots” key permission is unchecked. None of the other key permissions should be enabled either.

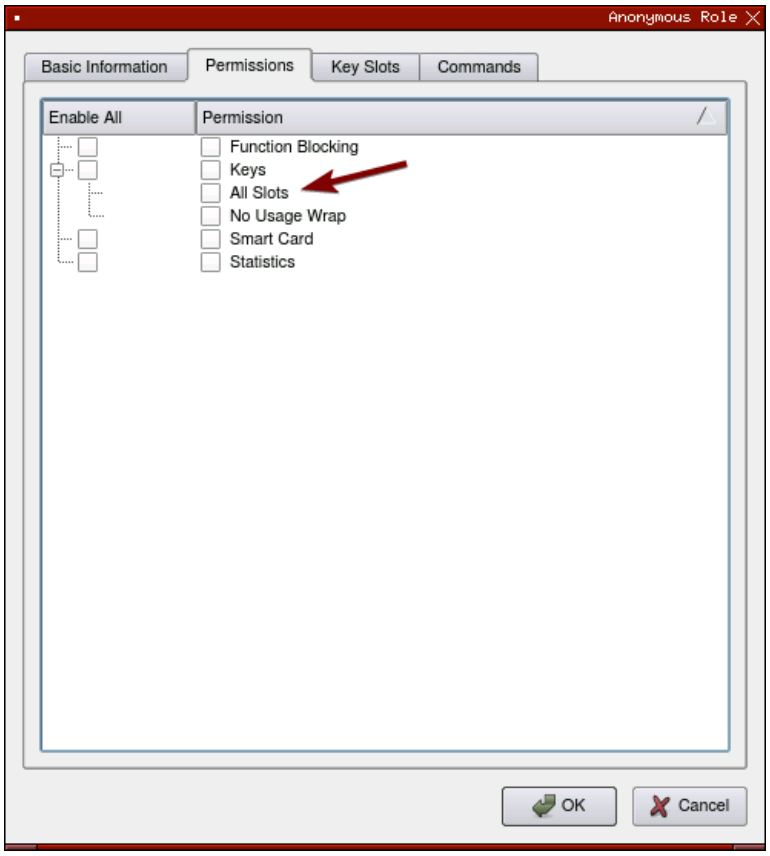


FIGURE: "ALL SLOTS" KEY PERMISSION

Under the “Key Slots” tab you need to ensure that there are no key ranges specified. By default, the Anonymous Application Partition has access to the entire range of key slots on the HSM.

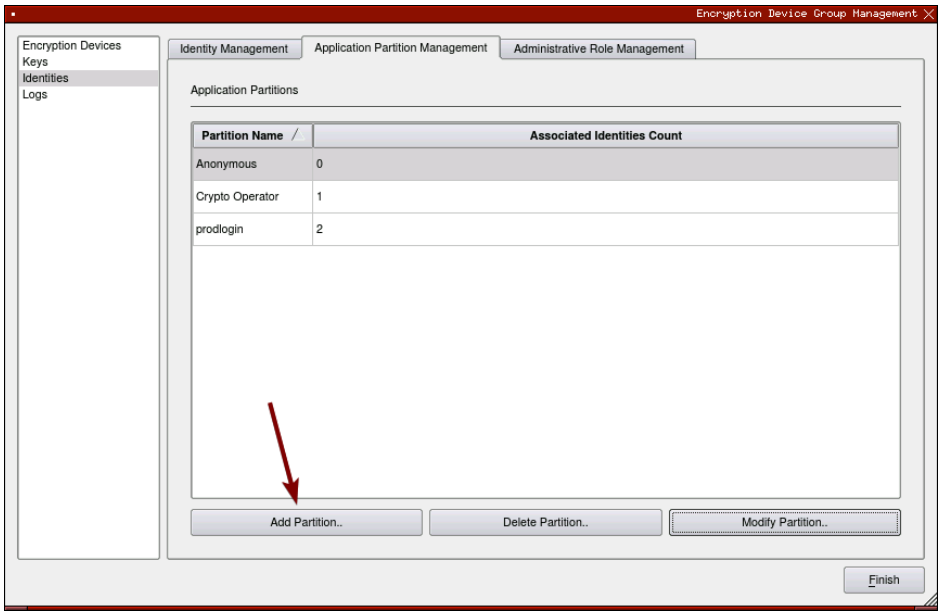
Lastly, under the “Commands” tab make sure that only the following PKCS #11 Communication commands are enabled for the Application Partition that you created:

- ECHO: Communication Test/Retrieve Version
- PRMD: Retrieve HSM restrictions
- RAND: Generate random data
- HASH: Retrieve device serial
- GPKM: Retrieve key table information
- GPKS: General purpose key settings get/change
- GPKR: General purpose key settings get (read-only)

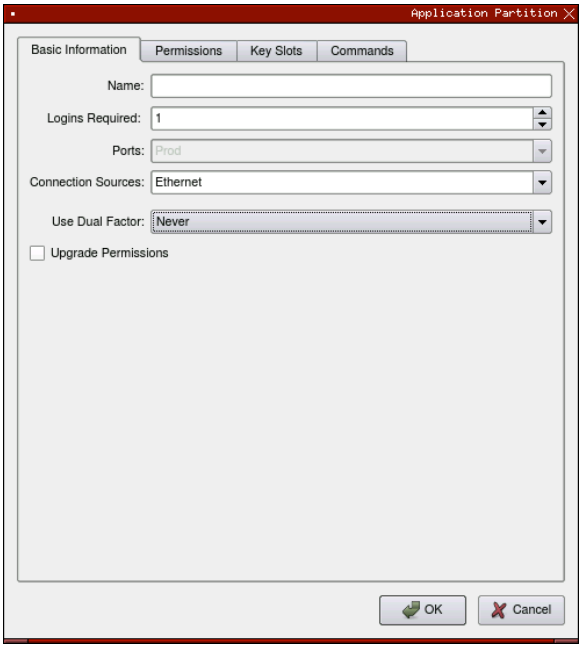
Create an Application Partition

In order for application segregation to occur on the HSM, an Application Partition must be created specifically for your use-case. Application partitions are used to segment the permissions and keys on an HSM between applications. The process for configuring a new application partition is outlined in the following steps:

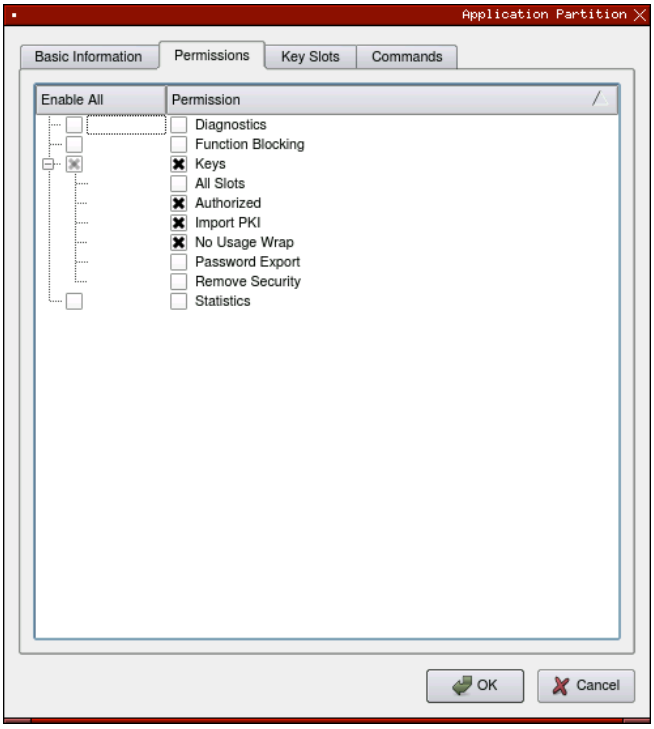
From the Application Partitions tab and click the Add Partition button at the bottom of the menu.



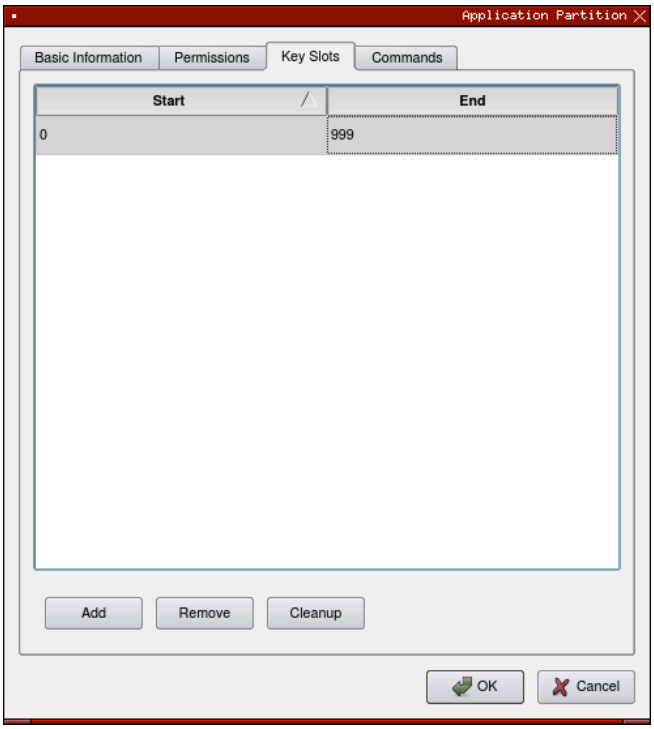
Fill in all of the fields in the “Basic Information” tab, as shown below. The information that is essential is Logins Required being set to “1”, the Ports being set to “Prod”, and the Connection Sources being set to “Ethernet”.



Under the "Permissions" tab, select the Key permissions shown in the screenshot below. The Authorized permission allows for keys that require login. The Import PKI permission allows trusting an external PKI, which is used by some applications to allow for PKI symmetric key wrapping (It is not recommended to enable unless using this use case). The No Usage Wrap permission allows for interoperable key wrapping without defining key usage as part of the wrapped key (This is only recommended if exchanging keys with external entities or using the HSM to wrap externally used keys).



Under Key Slots, it is recommended that you create a range of 1000 total keys (here we've specified the key range 0-999), which do not overlap with another Application Partition. Within this range, there must be ranges for both symmetric and asymmetric keys. If more keys are required by the application, configure accordingly.



Based on application requirements there are particular functions that need to be enabled on the Application Partition in order to utilize the HSMs functionality. The most often used commands are included below. These can be enabled under the "Commands" tab.

PKCS #11 Communication Commands

- **ECHO**: Communication Test/Retrieve Version
- **PRMD**: Retrieve HSM restrictions
- **RAND**: Generate random data
- **HASH**: Retrieve device serial
- **GPKM**: Retrieve key table information
- **GPKS**: General purpose key settings get/change
- **GPKR**: General purpose key settings get (read-only)

Key Operations Commands

- **APFP**: Generate PKI Public Key from Private Key
- **ASYL**: Load asymmetric key into key table
- **GECC**: Generate an ECC Key Pair
- **GPCA**: General purpose add certificate to key table
- **GPGS**: General purpose generate symmetric key
- **GPKA**: General purpose key add
- **GPKD**: General purpose key slot delete/clear
- **GRSA**: Generate RSA Private and Public Key
- **LRSA**: Load key into RSA Key Table
- **RPFP**: Get public components from RSA private key

Interoperable Key Wrapping

- **GPKU**: General purpose key unwrap (unrestricted)
- **GPUK**: General purpose key unwrap (preserves key usage)
- **GPKW**: General purpose key wrap (unrestricted)
- **GPWK**: General purpose key wrap (preserves key usage)

Data Encryption Commands

- **ADPK**: PKI Decrypt Trusted Public Key
- **GHSH**: Generate a Hash (Message Digest)
- **GPED**: General purpose data encrypt and decrypt
- **GPGC**: General purpose generate cryptogram from key slot
- **GPMC**: General purpose MAC (Message Authentication Code)
- **GPSR**: General purpose RSA encrypt/decrypt or sign/verify with recovery
- **HMAC**: Generate a hash-based message authentication code
- **RDPK**: Get Clear Public Key from Cryptogram

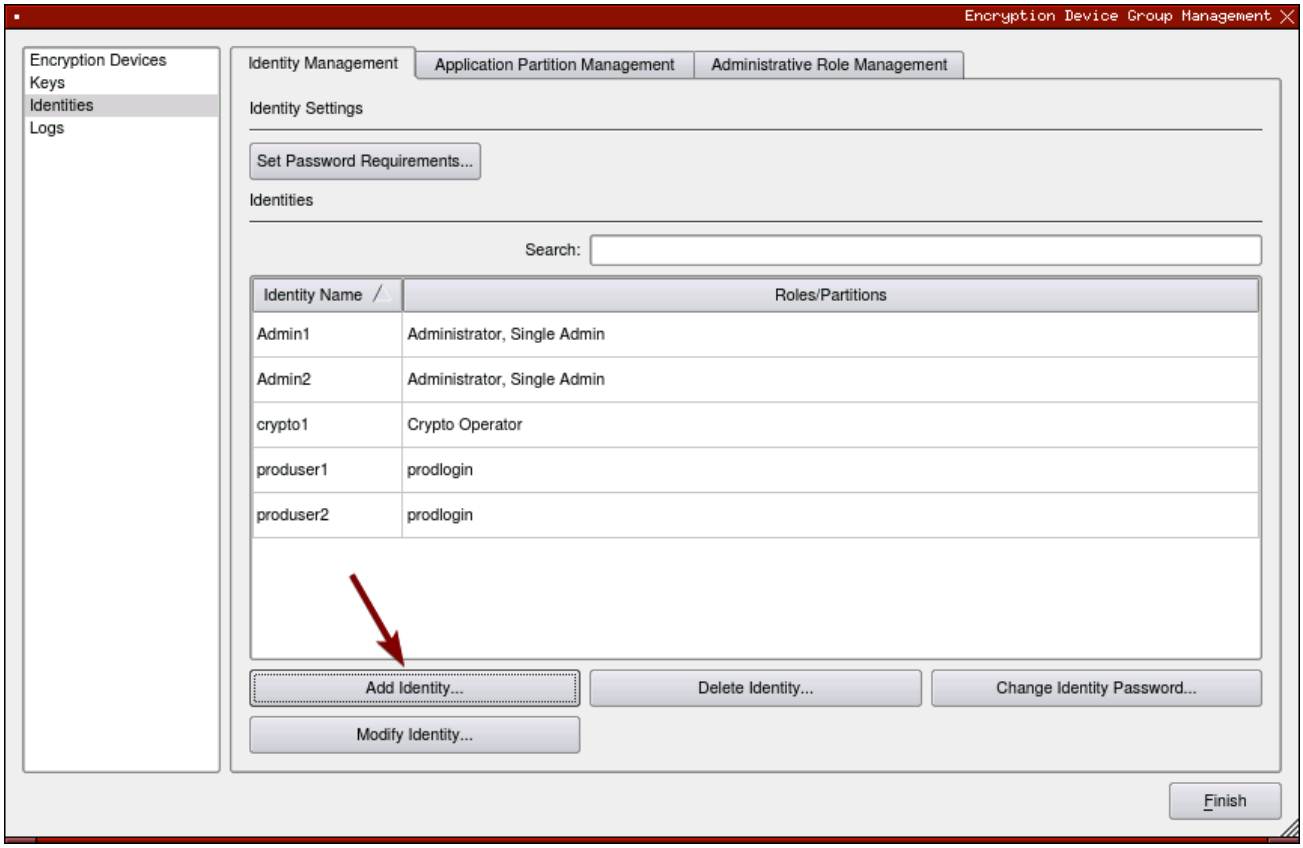
Signing Commands

- **ASYS**: Generate a Signature Using a Private Key
- **ASYV**: Verify a Signature Using a Public Key
- **GPSV**: General purpose data sign and verify
- **RSAS**: Generate a Signature Using a Private Key

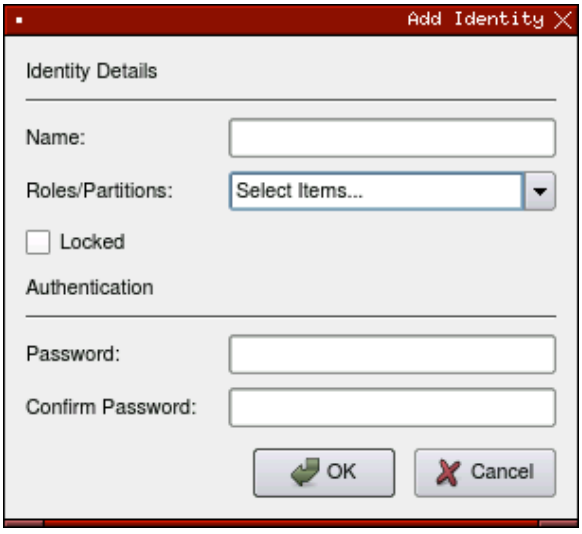
Create new Identity and associate it with the newly created Application Partition

For this step you will need to be logged in with an identity that has a role with permissions ***Identity:Add****. The default Administrator role and Admin identities can be used.*

A new identity must be created, which will need to be associated with the Application Partition created in step 7.5. To create this new identity, go to the *Identity Management* tab, and click "Add Identity...".



Specify a name for the new Identity, and in the Roles dropdown select the name of the Application Partition created in the previous step. This will associate this new Identity with that Application Partition.

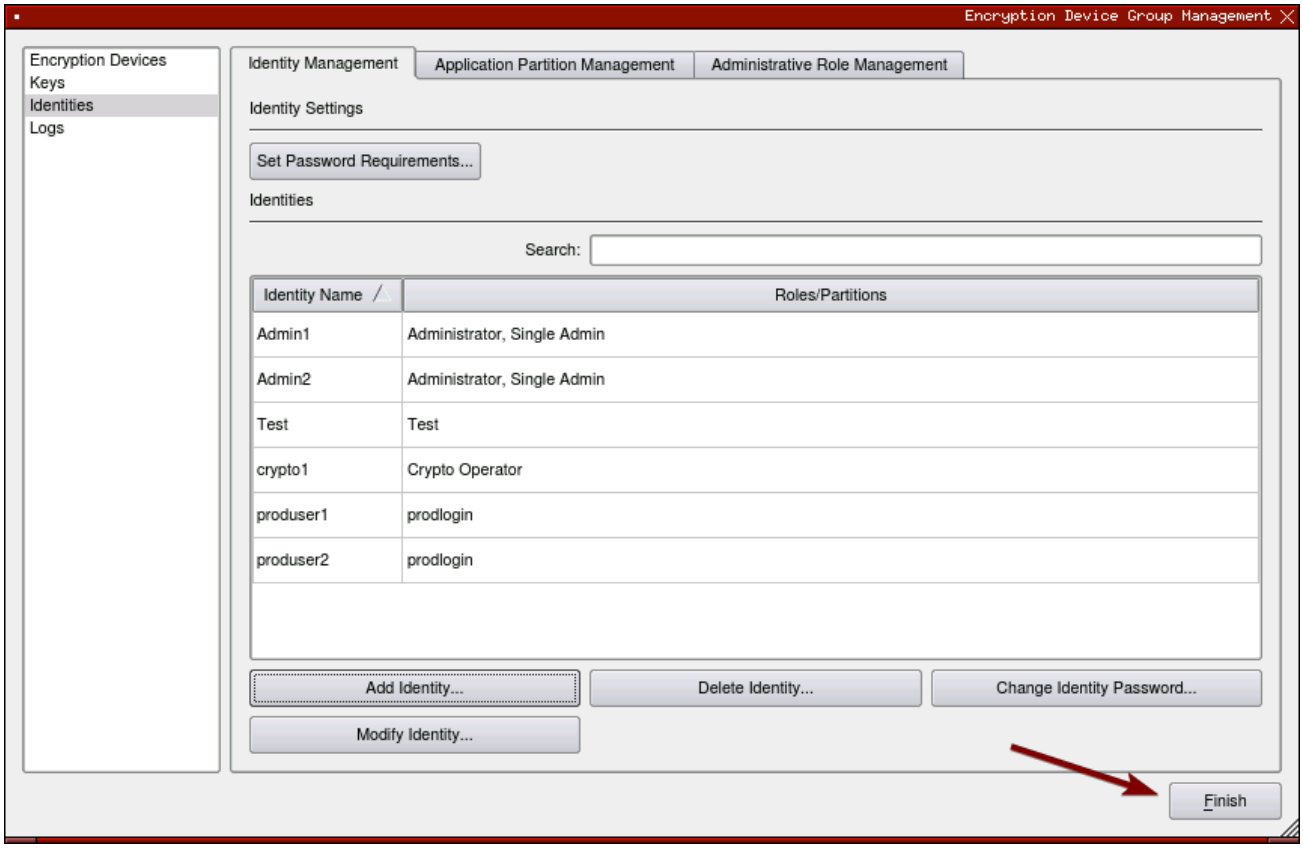


This new identity must be set in the fxpkcs11.cfg file, in the following section:

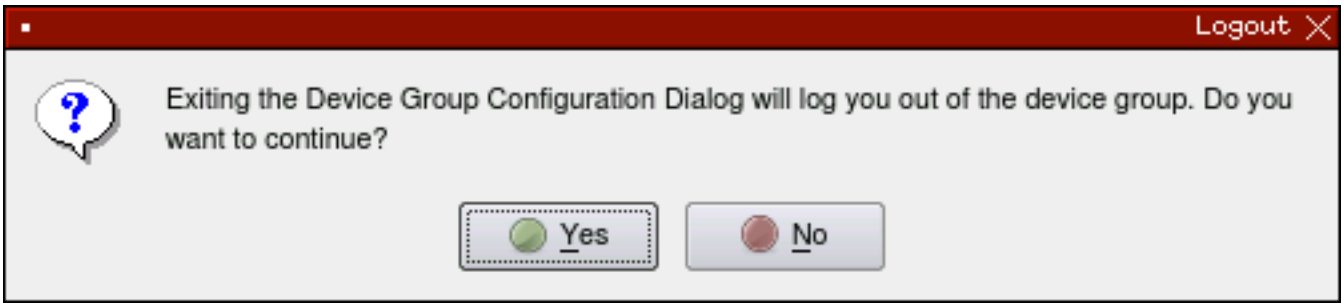
```
# HSM crypto operator identity name
<CRYPTO-OPR> [insert name of Identity that you created] </CRYPTO-OPR>

# Production connection
<PROD-ENABLED> YES </PROD-ENABLED>
<PROD-PORT> 9100 </PROD-PORT>
```

NOTE: Crypto Operator in the fxpkcs11.cfg file must match exactly the name of the identity created in the HSM.

Click the "Finish" button to exit out of this menu and log out of the device group.



Click "Yes" at the following prompt.



Configure TLS Authentication

For this step you will need to be logged in with an identity that has a role with permissions ***Keys:All Slots****,* ***Management Commands:Certificates****,* ***Management Commands:Keys****,* ***Security:TLS Sign****, and* ***TLS Settings:Upload Key****. The default Administrator role and Admin identities can be used.*

Enable Server-Side Authentication (Option 1)

Mutually authenticating to the HSM using client certificates is recommended, but server-side authentication is also supported. To enable server-side authentication go to *SSL/TLS Setup*, then select the Excrypt Port and enable the "Allow Anonymous" setting.

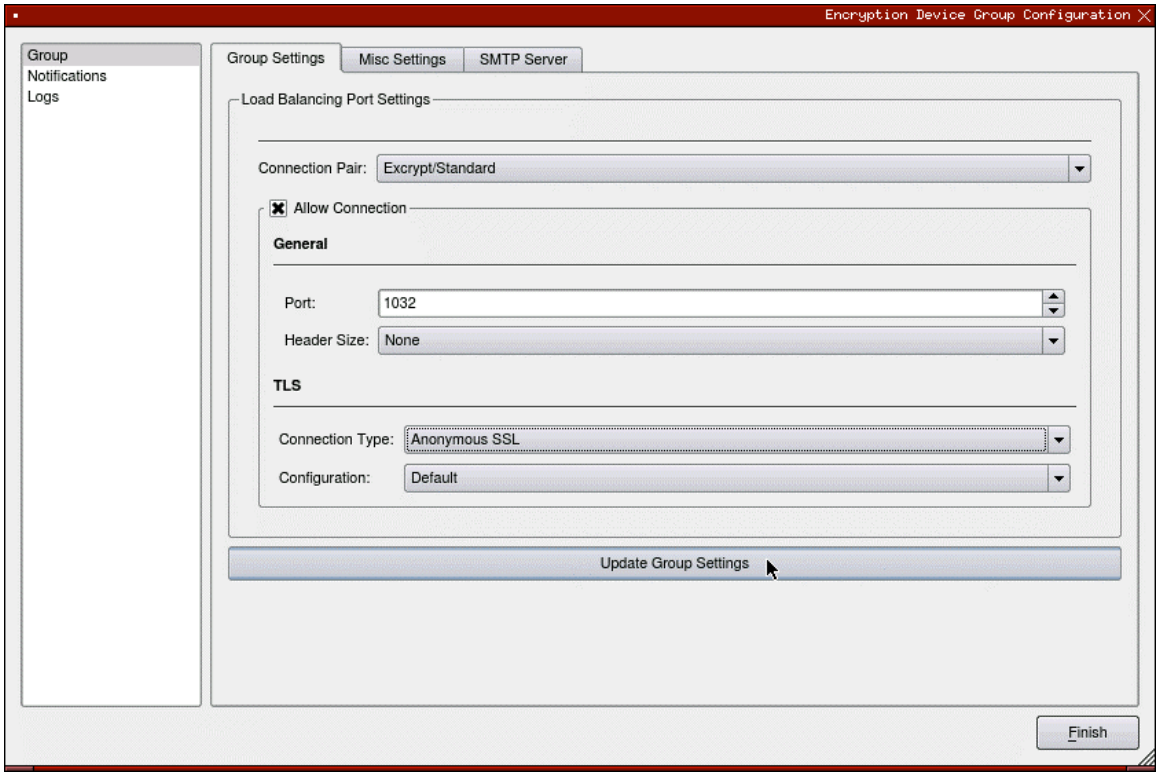


FIGURE: GROUP SETTINGS

You will receive confirmation that the device group settings have been successfully updated. Click "OK", then "Finish", to once again log out of the device group.

Create Connection Certificates for Mutual Authentication (Option 2)

To create client certificates for mutual authentication, refer to section 7.7.

NOTE: Because you're going directly to an HSM to create the client certificates, it may cause the device to drop out of sync. To re-sync, simply log on to the Guardian, right-click on the device, and select "Reconnect…".

APPENDIX B: 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 will help do 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 cannot be found 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

XCEPTIONAL SUPPORT

24x7x365

Toll-Free: 1-800-251-5112

E-mail: support@futurex.com

SOLUTIONS ARCHITECT

E-mail: solutions@futurex.com