THALES

Microsoft Authenticode: Integration Guide

THALES LUNA HSM

THALES DATA PROTECTION ON DEMAND

Document Information

Document Part Number	007-009988-001
Revision	P
Release Date	14 October 2020

Trademarks, Copyrights, and Third-Party Software

CONTENTS

Overview

This document covers the necessary information to install, configure, and integrate Microsoft Authenticode with Luna Hardware Security Modules (HSM) or an HSM on Demand (HSMoD) Service. Luna HSMs come as on premise hardware HSMs widely known as Luna HSM and a Data Protection on Demand (DPoD) cloud offering HSM on Demand Service.

Microsoft Authenticode permits end users to identify who has published a software component and verify that no one has tampered with it before downloading it from the Internet. Authenticode guarantees the integrity of published software.

Authenticode relies on proven cryptographic techniques and the use of one or more private keys to sign and time-stamp the published software. It is important to maintain the confidentiality of these keys. Luna Hardware Security Module (HSM) or an HSMoD service integrates with Microsoft Authenticode to provide a trusted system for protecting the organizational credentials of the software publisher. Luna HSMs secures the code-signing key within an industry standard FIPS 140-2 level 3 validated HSM.

The benefits of using Luna HSMs with Authenticode include:

- Secure generation, storage and protection of the Identity signing private key on FIPS 140-2 level 3 validated hardware*.
- Full life cycle management of the keys.
- HSM audit trail.

  NOTE: HSM on Demand services do not have access to the secure audit trail.

- Protection for the organizational credentials of the software publisher.
- Secure storage of the private key.
- Provision of a trusted time-stamp to Authenticode.

*FIPS 140-2 validation in progress for HSMoD services

Supported Platforms

Below is the list of the platforms tested with the following HSMs:

Luna HSM: Luna HSM appliances are purposefully designed to provide a balance of security, high performance, and usability that makes them an ideal choice for enterprise, financial, and government organizations. Luna HSMs physically and logically secure cryptographic keys and accelerate cryptographic processing. The Luna HSM on premise offerings include the Luna Network HSM, Luna PCIe HSM, and Luna USB HSMs. Luna HSMs are also available for access as an offering from cloud service providers such as IBM cloud HSM and AWS cloud HSM classic.
This integration is supported with Luna HSM on the following operating systems:

- Windows 2019 Server
- Windows 2016 Server
- Windows Server 2012R2

  NOTE: This integration is tested with Luna Clients in FIPS and HA mode.

Thales Data Protection on Demand (DPOD): is a cloud-based platform that provides on-demand HSM and Key Management services through a simple graphical user interface. With DPOD, security is simple, cost effective and easy to manage because there is no hardware to buy, deploy and maintain. As an Application Owner, you click and deploy services, generate usage reports and maintain just the services you need.

This integration is supported/verified with Thales DPOD on the following operating systems:

- Windows 2019 Server
- Windows 2016 Server
- Windows Server 2012R2

Prerequisites

Before starting the integration of Microsoft Authenticode with Luna HSM or HSM on Demand Service, ensure you have completed configuring the Luna Network HSM or provisioning HSM on Demand Service as per the requirement.

Configuring Luna Network HSM

Before you get started ensure the following:

1. Luna Network HSM appliance and a secure admin password.
2. Luna Network HSM, and a hostname, suitable for your network.
3. Luna Network HSM network parameters are set to work with your network.
4. Initialize the HSM on the Luna Network HSM appliance.
5. Create and exchange certificates between the Luna Network HSM and your Client system.
6. Create a partition on the HSM that will be later used by Microsoft Authenticode.
7. Register a client for the system and assign the client to the partition to create an NTLS connection. Initialize Crypto Officer and Crypto User roles for the registered partition.
8. Ensure that the partition is successfully registered and configured. The command to see the registered partition is:

   ```
   lunacm.exe
   ```

   Example output:

   ```
   Available HSMs:
   Slot Id ->              0
   Label ->                HSM7
   Serial Number ->        1213475834614
   Model ->                LunaSA 7.3.0
   Firmware Version ->     7.3.0
   Configuration ->        Luna User Partition With SO (PW) Signing With
   Cloning Mode
   Slot Description ->     Net Token Slot
   Current Slot Id: 0
   ```

NOTE: Follow the Luna Network HSM documentation for detailed steps for creating NTLS connection, initializing the partitions and various user roles.

Using Luna HSM in FIPS Mode

Under FIPS 186-3/4, the RSA methods permitted for generating keys are 186-3 with primes and 186-3 with aux primes. This means that RSA PKCS and X9.31 key generation is no longer approved for operation in a FIPS-compliant HSM. If you are using the Luna HSM in FIPS mode, you have to make the following change in configuration file:

```
[Misc]
RSAKeyGenMechRemap = 1
```

The above setting redirects the older calling mechanism to a new approved mechanism when the Luna HSM is in FIPS mode.

Provisioning your HSM on Demand Service

This service enables your client machine to access an HSM application partition for storing cryptographic objects used by your applications. Application partitions can be assigned to a single client, or multiple clients can be assigned to, and share, a single application partition.

You need to provision your application partition by initializing the following roles:

- **Security Officer (SO)** - Responsible for setting the partition policies and for creating the Crypto Officer.
- **Crypto Officer (CO)** - Responsible for creating, modifying, and deleting crypto objects within the partition. The CO can use the crypto objects and create an optional, limited-capability role called Crypto User that can use the crypto objects but cannot modify them.
- **Crypto User (CU)** – An optional role that can use crypto objects while performing cryptographic operations.

NOTE: Refer the “Thales Data Protection on Demand Application Owner Quick Start Guide” for configuring the HSM on Demand service and creating a service client. The HSM service client package is a zip file containing system information needed to connect your client machine to an existing HSM on Demand service.

HSM on Demand Service can be configured in the following scenarios:

- **User wants to use DPoD Client to access service partition:** Execute steps 1-4 and 10.
- **User wants to use Luna Client to access the service partition:** Execute steps 1-10.
- **User wants to use existing Luna Client to access the service partition in Hybrid mode with Luna Partition:** Execute steps 1-10.

NOTE: Last two scenarios are supported for Universal Client only, starting from Luna Client v10.1.0 onwards.

To Configure DPoD HSM on Demand service with/without Luna Client

1. Transfer the downloaded .zip file to your Client workstation using pscp, scp, or other secure means.

2. Extract the .zip file into a directory on your client workstation.
3. Extract or untar the appropriate client package for your operating system. Do not extract to a new subdirectory; place the files in the DPoD client install directory.

   ```
   [Windows]
   cvclient-min.zip
   [Linux]
   cvclient-min.tar
   # tar -xvf cvclient-min.tar
   ```

4. Run the provided script to create a new configuration file containing information required by the HSMoD service.

   ```
   [Windows]
   Right-click setenv.cmd and select Run as Administrator.
   [Linux]
   Source the setenv script.
   # source ./setenv
   ```

   NOTE: Run the LunaCM utility available in the DPoD client and verify the service partition is listed. If you need to configure DPoD service partition with existing Luna Client follow further steps.

5. Copy the server and partition certificates from the DPoD client directory to your Luna client certificates directory:

   DPoD Certificates:

   ```
   server-certificate.pem
   partition-ca-certificate.pem
   partition-certificate.pem
   ```

   LunaClient Certificate Directory:

   [Windows default]

   ```
   C:\Program Files\Safenet\Lunaclient\cert\
   ```

   [Linux default]

   ```
   /usr/safenet/lunaclient/cert/
   ```

6. Open the configuration file from the DPoD client directory and copy the **XTC** and **REST** section.

   [Windows]

   ```
   crystoki.ini
   ```

   [Linux]

   ```
   Chrystoki.conf
   ```

7. Edit the Luna Client configuration file and add the **XTC** and **REST** section. In both sections you need to change only server and partition certificates path from step 5. Do not change any other entries provided in **XTC** and **REST** section.

```
[XTC]
. . .
PartitionCAPath=<LunaClient_cert_directory>\partition-ca-certificate.pem
PartitionCertPath00=<LunaClient_cert_directory>\partition-certificate.pem
. . .

[REST]
. . .
SSLClientSideVerifyFile=<LunaClient_cert_directory>\server-certificate.pem
. . .
```

8. Edit the following entry from the **Misc** section and update the correct path for the **plugins** directory:

```
Misc]
PluginModuleDir=<LunaClient_plugins_directory>

[Windows Default]

C:\Program Files\Safenet\Lunaclient\plugins\

[Linux Default]

/usr/safenet/lunaclient/plugins/
```

Save the configuration file. If you wish, you can now safely delete the extracted DPoD client directory.

9. Reset the **ChrystokiConfigurationPath** environment variable and point back to the location of the Luna Client configuration file.

[Windows]

In the Control Panel, search for "environment" and select **Edit the system environment variables**. Click **Environment Variables**. In both list boxes for the current user and system variables, edit **ChrystokiConfigurationPath** and point to the **crystoki.ini** file in the Luna client install directory.

[Linux]

Either open a new shell session, or export the environment variable for the current session pointing to the location of the **Chrystoki.conf** file:

```
# export ChrystokiConfigurationPath=/etc/
```

10. Run the **LunaCM** utility and verify the service partition is listed. If you already have a Luna Partition before configuring the DPoD service partition, both Luna and DPoD service partition will be listed.

Constraints on HSM on Demand Services

To use HSM on Demand Service in FIPS mode

HSMoD services operate in a FIPS and non-FIPS mode. If your organization requires non-FIPS algorithms for your operations, ensure you enable the **Allow non-FIPS approved algorithms** check box when configuring your HSM on Demand service. The FIPS mode is enabled by default.

Refer to the *Mechanism List* in the *SDK Reference Guide* for more information about available FIPS and non-FIPS algorithms.

To verify HSM on Demand <slot value>

LunaCM commands work on the current slot. If there is only one slot, then it is always the current slot. If you are completing an integration using HSMoD services, you need to verify which slot on the HSMoD service you send commands to. If there is more than one slot, then use the slot set command to direct a command to a specified slot. You can use slot list to determine which slot numbers are in use by which HSMoD service.

To use Thales HSM in FIPS Mode

Under FIPS 186-3/4, the RSA methods permitted for generating keys are 186-3 with primes and 186-3 with aux primes. This means that RSA PKCS and X9.31 key generation is no longer approved for a FIPS-compliant HSM. If you are using the Luna HSM or HSM on Demand service in FIPS mode, you have to make the following change to the configuration file:

```
[Misc]

RSAKeyGenMechRemap=1
```

The above setting redirects the older calling mechanism to a new approved mechanism when Luna HSM or HSMoD is in FIPS mode.

> **NOTE:** For Universal Client, above setting is not required. This setting is applicable for Luna Client 7.x only.

> **NOTE:** This remapping is automatic if you are using Luna HSM Client 10.1 and above, and the configuration file entry is ignored.

Setting up Microsoft Authenticode

Install the Windows SDK

The Authenticode programs (makecert, cert2spc, etc.,) are installed with Microsoft Visual Studio and Microsoft Windows SDK.

Refer to the Microsoft Windows SDK installation documentation for further information.

Install the Office Smart Tags SDK

To demonstrate the Authenticode technology, this integration guide requires the Microsoft Office Smart Tags SDK. Refer to the Microsoft Office Smart Tags SDK installation documentation for further information.

Integrating Luna HSM with Microsoft Authenticode

Microsoft Authenticode permits end users to identify who published a software component and verify that no one has tampered with the software component before downloading the object from the internet.

This integration guide contains the following topics:

> Configure SafeNet Cryptographic Storage Provider

> Configure SafeNet Key Storage Provider (KSP)

> Sign and time stamp the code

Configure SafeNet Cryptographic Storage Provider

To use Microsoft Authenticode with a Luna HSM, configure the SafeNet Cryptographic Service Provider (CSP) to generate the certificates for Microsoft Authenticode. Configure the Luna Cryptographic Service Provider (CSP) on Windows Server 2012 R2/2016/2019.

Run the command, *register.exe* to register Luna CSP. The general form of command is:

```
<Luna Client Installation Directory>\CSP>register.exe
```

Register the Luna Cryptographic Services for Microsoft Windows. The general form of command is:

```
<Luna Client Installation Directory>\CSP>register.exe /l
```

Configure SafeNet Key Storage Provider (KSP)

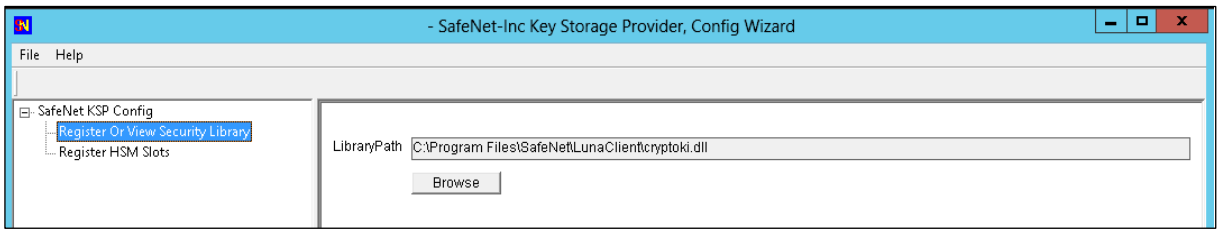
You must configure the SafeNet Key Storage Provider (KSP) to allow the user account and system to access the Luna HSM or HSM on Demand Service.

> If using a Luna HSM, the KSP package must be installed during the Luna Client software installation.

> If using an HSM on Demand (HSMoD) service, the KSP package is included in the HSMoD service client package inside the /KSP directory.

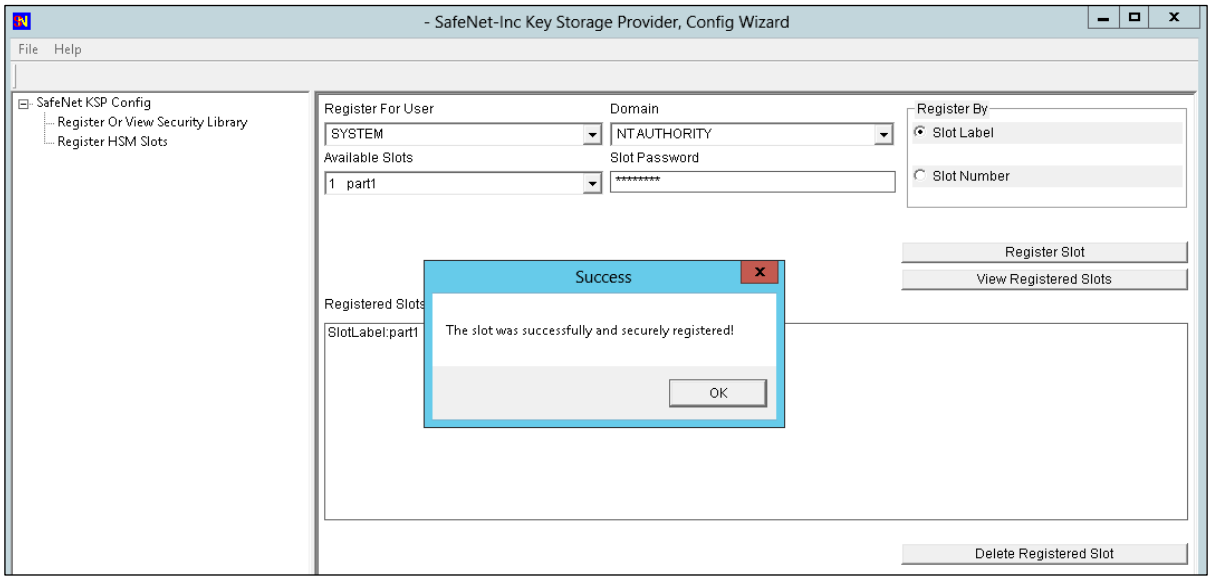
To configure the SafeNet Key Storage Provider:

1. Navigate to the <Luna HSM Client installation Directory>/KSP directory.
2. Run the KspConfig.exe (KSP configuration wizard).
3. Double-click Register Or View Security Library and browse the library `cryptoki.dll` from the Luna HSM Client installation directory or HSMoD service client package. Click **Register**.



4. On successful registration, you will see a message: **Success registering the security library**.
5. Double-click **Register HSM Slots** on the left side of the pane and enter the Slot (Partition) password.
6. Click **Register Slot** to register the slot for Domain\User. On successful registration, you will see a message: **The slot was successfully and securely registered**.

7. Register the same slot for **NT AUTHORITY\SYSTEM**.



8. Generate a certificate using the `makecert` command and the Luna CSP `"Luna Cryptographic Services for Microsoft Windows".`

9. `makecert -sk mykey -sp "Luna Cryptographic Services for Microsoft Windows" -n "CN=MS-AuthCode" -r -ss mystore Test.cer -a sha256 -len 2048`

   where:

   -sk The location of the subject's key container which holds the private key.

   -sp Subject CryptoAPI's provider name.

   -n The name and details of the publisher's certificate.

   -ss The name of the subject's certificate store in which the generated certificate will be stored.

   -a <algorithm> The signature's digest algorithm. <md5|sha1|sha256|sha384|sha512>. Default to 'sha1'

   -len <number> Generated Key Length (Bits) Default to '2048' for 'RSA' and '512' for 'DSS'

10. Create a Software Publishing Certificate (SPC) from the generated certificate.

```
cert2spc Test.cer Test.spc
```

Sign and time stamp the code

Sign and time stamp the code using the signtool sign command and the Software Publishing Certificate (SPC). To sign and time stamp the code:

Sign and time stamp the code using signtool:

```
signtool sign /v /f Certificate /p Pin /csp "Cryptographic Service Provider Name" /k "Key Container Name" /t timestamp URL "File to be signed"
```

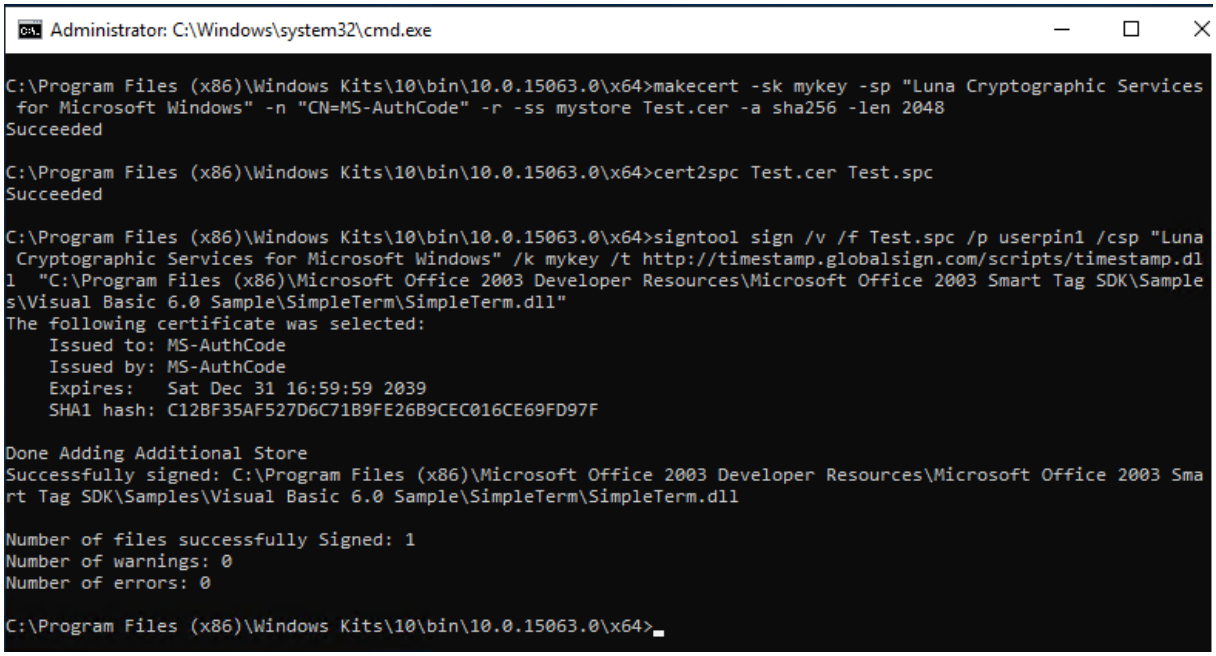
where:

/f Publisher's Certificate.

/p HSM partition password.

/k Container Name that contains the signing key.

/t URL used for Time Stamping.



Integrating Luna HSM with MS Strong Name

Strong Name is the part of Microsoft SDK that offers a powerful mechanism for giving .NET Framework assemblies unique identities. To get a valid strong name, an assembly is strong-name signed during the build process. This is done using the private key that corresponds to the public key in the strong name. The strong name signature can then be verified using the public key.

This integration guide contains the following topics:Sign a .NET Assembly

> Configure SafeNet Cryptographic Storage Provider

> Configure SafeNet Key Storage Provider (KSP)

> Sign a .NET Assembly

Configure SafeNet Cryptographic Storage Provider

To use Microsoft Strong Name with a Luna HSM configure the SafeNet Cryptographic Service Provider (CSP) to generate the keys and certificates for Microsoft Authenticode.

Configure Luna Cryptographic Service Provider (CSP) on Windows Server.

> Open the command prompt and run register.exe to register Luna CSP. The general form of command is given below:

  <Luna Client Installation Directory>\CSP>register.exe

> To register the Luna Cryptographic Services for Microsoft Windows. The general form of command is given below:

  <Luna Client Installation Directory>\CSP>register.exe /l

Configure SafeNet Key Storage Provider (KSP)

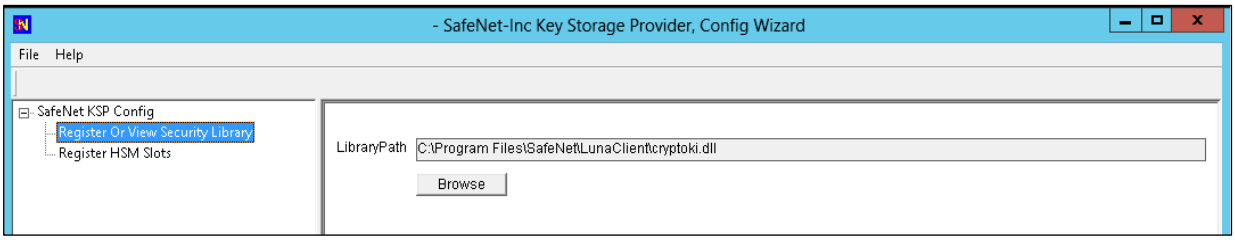
You must configure the SafeNet Key Storage Provider (KSP) to allow the user account and system to access the Luna HSM or HSM on Demand Service.

> If using a Luna HSM, the KSP package must be installed during the Luna Client software installation.

> If using an HSM on Demand (HSMoD) service, the KSP package is included in the HSMoD service client package inside of the /KSP folder.

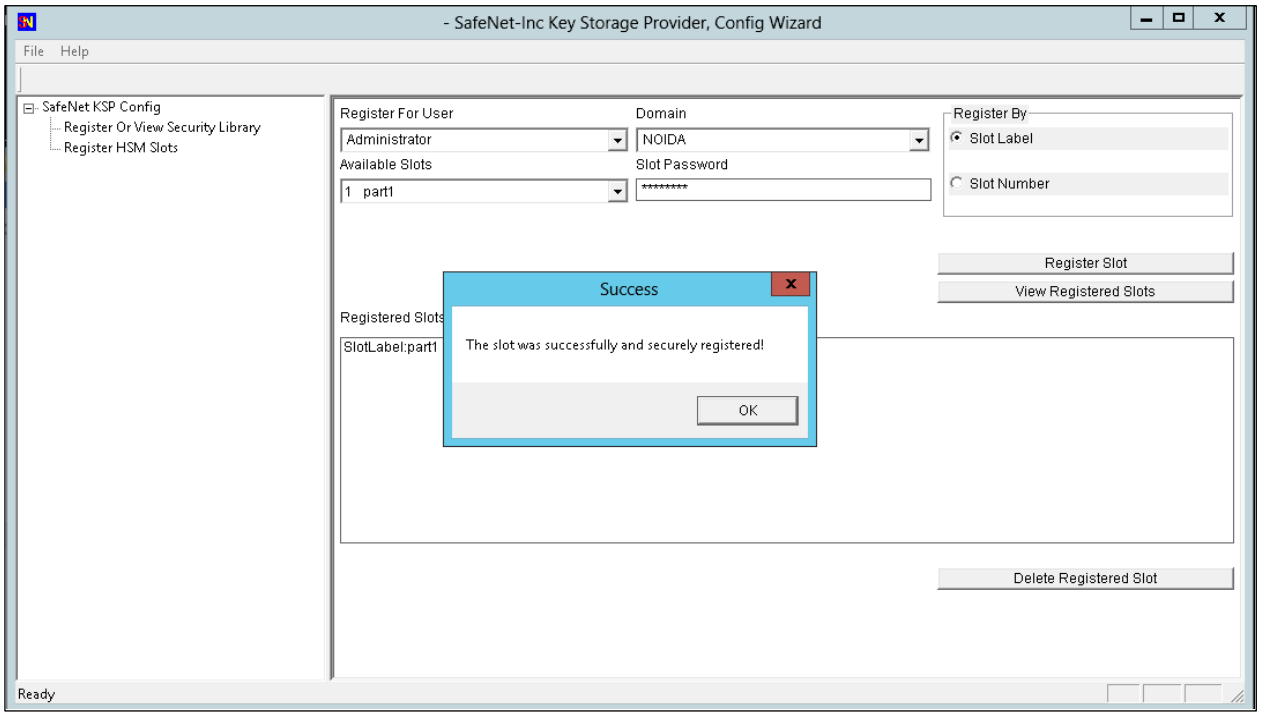
To configure the SafeNet Key Storage Provider

1. Navigate to the <Luna HSM Client installation Directory>/KSP directory.
2. Run the KspConfig.exe (KSP configuration wizard).
3. Double-click Register Or View Security Library.
4. Browse the library cryptoki.dll from the Luna HSM Client installation directory or HSMoD service client package and click Register.

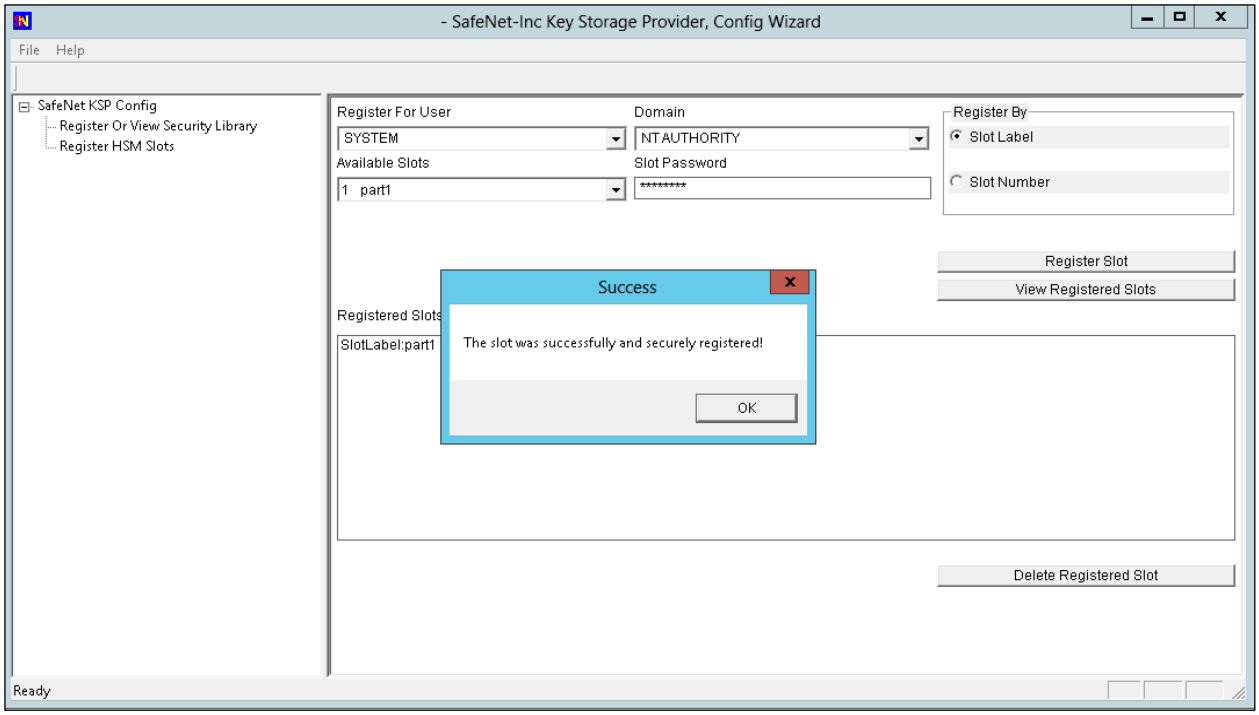


5. On successful registration, you will see a message **Success registering the security library**.
6. Double-click **Register HSM Slots** on the left side of the pane.
7. Enter the Slot (Partition) password.

8. Click **Register Slot** to register the slot for Domain\User. On successful registration, a message "**The slot was successfully and securely registered**" displays.



9. Register the same slot for **NT AUTHORITY\SYSTEM**.

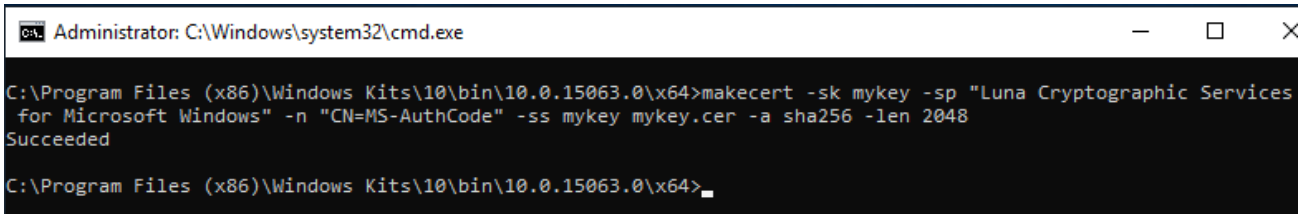


10. Generate a certificate using the `makecert` command and the Luna CSP `"Luna Cryptographic Services for Microsoft Windows"`.

```
makecert -sk <keyContainer> -sp "Luna Cryptographic Services for Microsoft Windows" –n "CN=Common Name" -ss <certStore> CertificateName.cer -a sha256 -len 2048
```

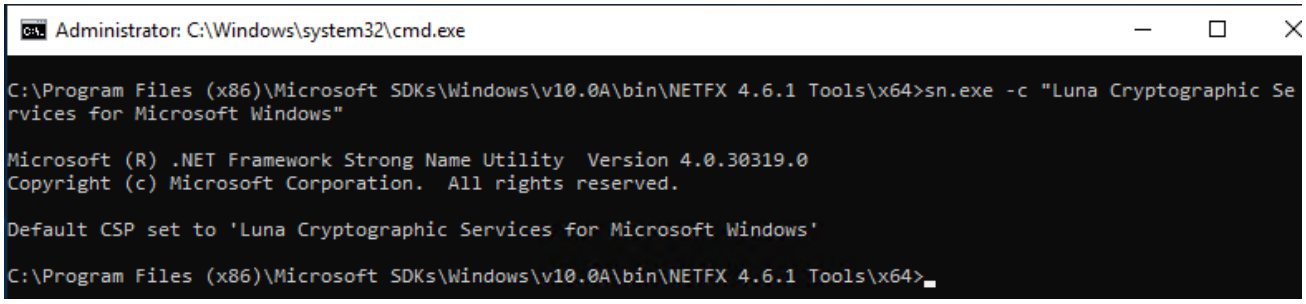
where:

-sk	The location of the subject's key container which holds the private key
-sp	Subject CryptoAPI's provider name
-n	Name and details of the signer's certificate
-ss	Name of the subject's certificate store in which the generated certificate will be stored.
-a <algorithm>	Signature's digest algorithm. <md5|sha1|sha256|sha384|sha512>. Default to 'sha1'
-len <number>	Generated Key Length (Bits) Default to '2048' for 'RSA' and '512' for 'DSS'



11. Make the Luna CSP the default CSP to use with Microsoft Strong Name using the following command:

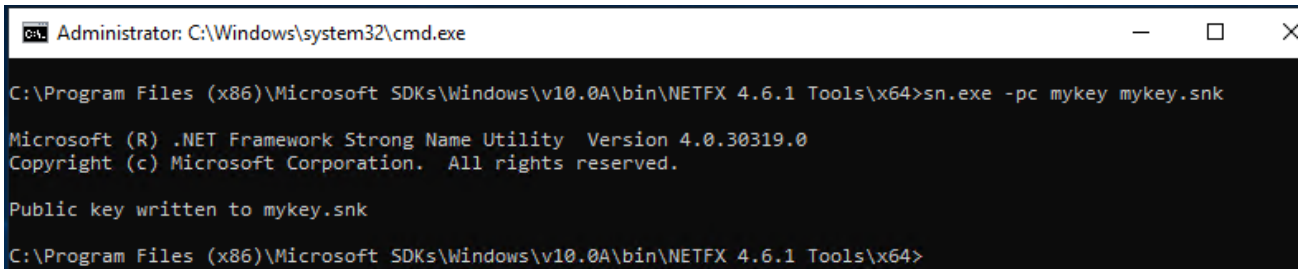
```
sn.exe -c "Luna Cryptographic Services for Microsoft Windows"
```



12. Extract the public key from the key-pair generated in step 2 using the following command:

```
sn.exe -pc mykey mykey.snk
```

where "mykey" is the name of key container and "mykey.snk" is name of public key file.



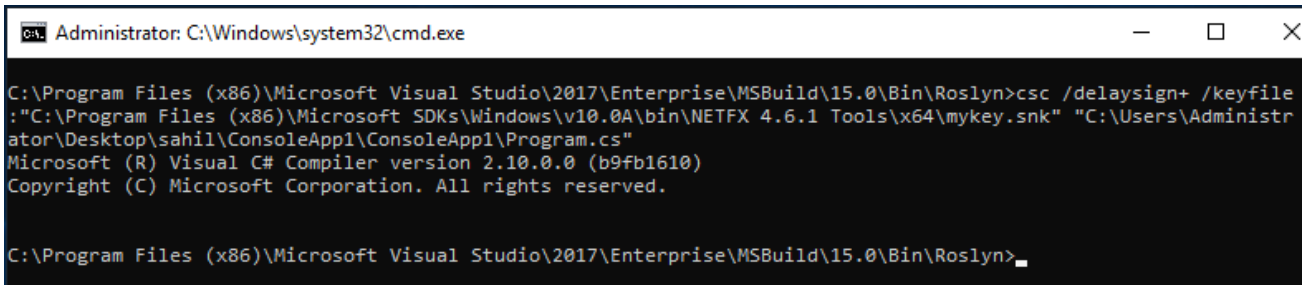
Sign a .NET Assembly

You can use MS Strong by signing a .NET Assembly. To sign a .NET assembly:

1. Write any C# program. Open the Visual Studio command prompt and compile the program. Delay sign the generated exe file. Use the following command:

   ```
   csc /delaysign+ /keyfile:"C:\Program Files (x86)\Microsoft
   SDKs\Windows\v8.1A\bin\NETFX 4.5.1 Tools\x64\mykey.snk"
   C:\Users\Administrator\Desktop\myapp.cs
   ```

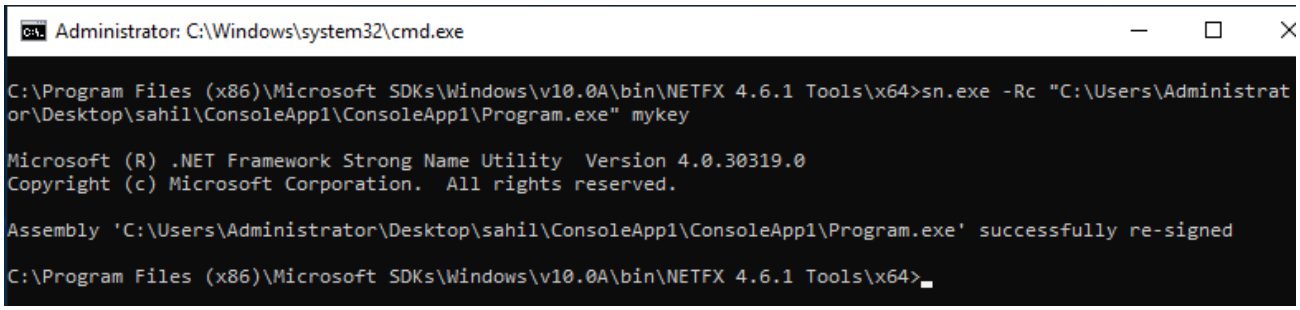
   Where "/keyfile" is the public key extracting from the key-pair in the previous command.



2. Sign the generated exe with Strong Name::

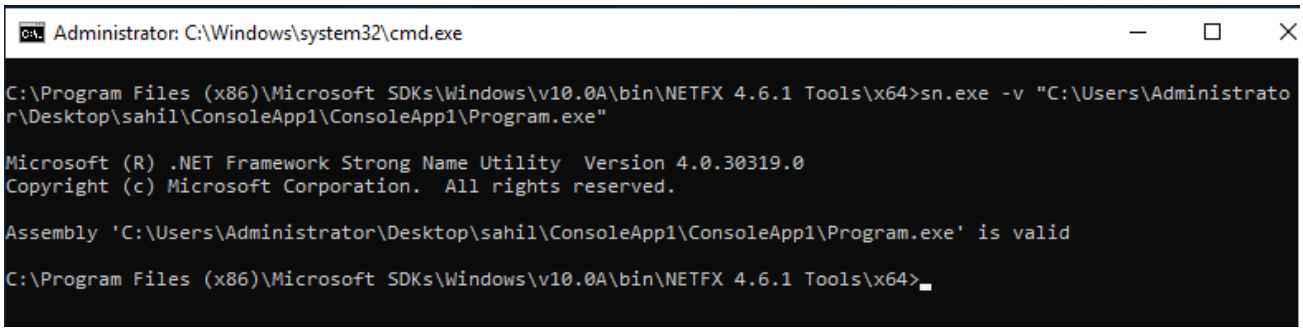
   ```
   sn.exe -Rc C:\Users\Administrator\Desktop\myapp.exe mykey
   ```

   Where "mykey" is the key container in which you have generated the key-pair.



3. Verify the assembly is Strong Name signed using the following command:

   ```
   sn.exe -v C:\Users\Administrator\Desktop\myapp.exe
   ```



   Strong Name is successfully signed and verified with .Net assembly using the key-pair generated on Luna HSM.

Integrating Luna HSM with MS Mage/ClickOnce

Microsoft's Mage.exe is a Manifest Generation and Editing command line Tool for .NET Framework applications. There is also a UI version MageUI.exe. A typical use is manually creating your ClickOnce deployment manifests. This guide assumes that you have a Windows application ready for deployment. This application will be referred to as **AppToDeploy**. For more details, refer Microsoft Documentation.

> **NOTE:** Luna HSM is used to secure the signing keys so that your signing keys never access by any unauthorized entity. Mage.exe is a 32 bit application so you have to use the 32 bit Luna Client with 32 bit CSP.

This integration guide contains the following topics:

> Configure SafeNet Cryptographic Storage Provider

> Deploy an application with Mage.exe command line tool

Configure SafeNet Cryptographic Storage Provider

To use Microsoft Mage/ClickOnce with a Luna HSM configure the SafeNet Cryptographic Service Provider (CSP) to generate the keys and certificates for Microsoft Mage/ClickOnce.

To configure the SafeNet CSP:

1. Install Luna Cryptographic Service Provider (CSP) on Windows Server.
   i. Open the command prompt and run *register.exe* to register Luna CSP. The general form of command is `<Luna Client Installation Directory>\win32\csp>register.exe`
   ii. To register the Luna Cryptographic Services for Microsoft Windows. The general form of command is `<Luna Client Installation Directory>\win32\csp>register.exe /l`
2. Generate a certificate using the `makecert` command and the Luna CSP `"Luna Cryptographic Services for Microsoft Windows".`

   ```
   makecert -sk <keyContainer> -sp "Luna Cryptographic Services for Microsoft Windows" –n "CN=Common Name" -ss <certStore> CertificateName.cer
   ```

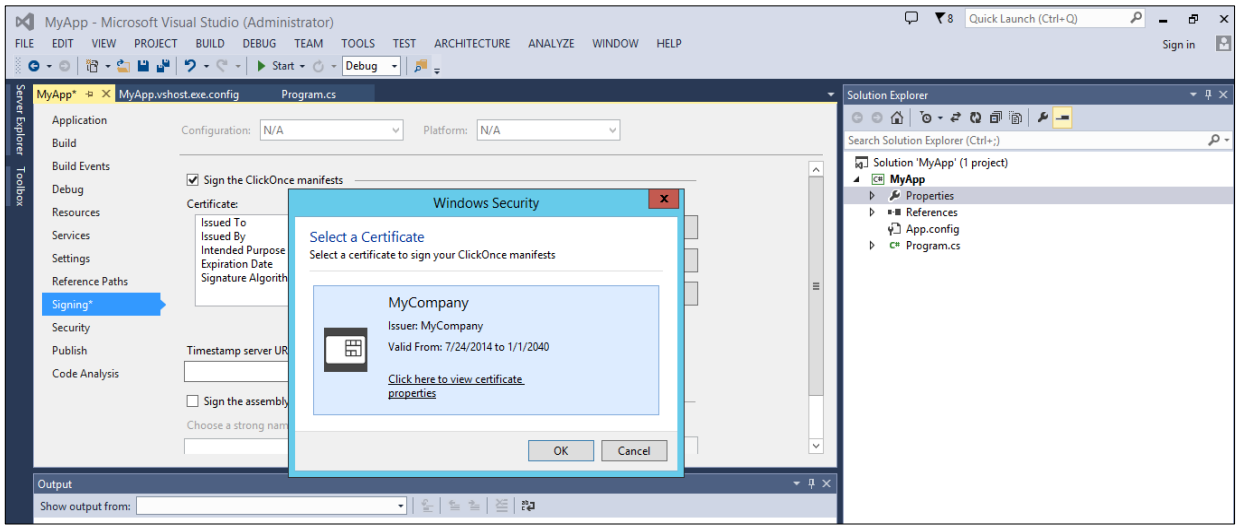
   where:

-sk	The location of the subject's key container which holds the private key
-sp	Subject CryptoAPI's provider name
-n	The name and details of the signer's certificate
-ss	The name of the subject's certificate store in which the generated certificate will be stored.

   > **NOTE:** Use "My" which is the default user cert store where the application is looking for certificate.

3. After generating the certificate, use this certificate in Visual Studio to sign the Application/Deployment manifest. Open the Properties window of the project and click on the Signing and then select Sign the ClickOnce manifests.

4. Click on **Select from Store…** and click **OK** after choosing the certificate that was generated in step 1.

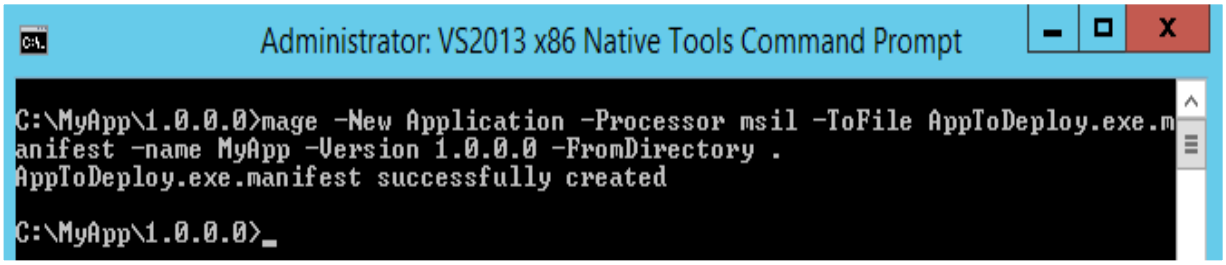


Deploy an application with Mage.exe command line tool

To deploy applications using the Mage.exe command line tool:

1. Create a directory to store your **ClickOnce** deployment files. In this directory, create a version subdirectory. If this is the first time application is deployed, name the version subdirectory 1.0.0.0.
2. Copy all of application files to the version subdirectory, including executable files, assemblies, resources, and data files. If necessary, create additional subdirectories that contain additional files.
3. Open the Windows SDK or Visual Studio command prompt and change to the version subdirectory.
4. Create the application manifest with a call to Mage.exe. The following statement creates an application manifest for code compiled to run on the msil processor.

```
mage -New Application -Processor msil -ToFile AppToDeploy.exe.manifest -
name "MyApp" -Version 1.0.0.0 -FromDirectory .
```

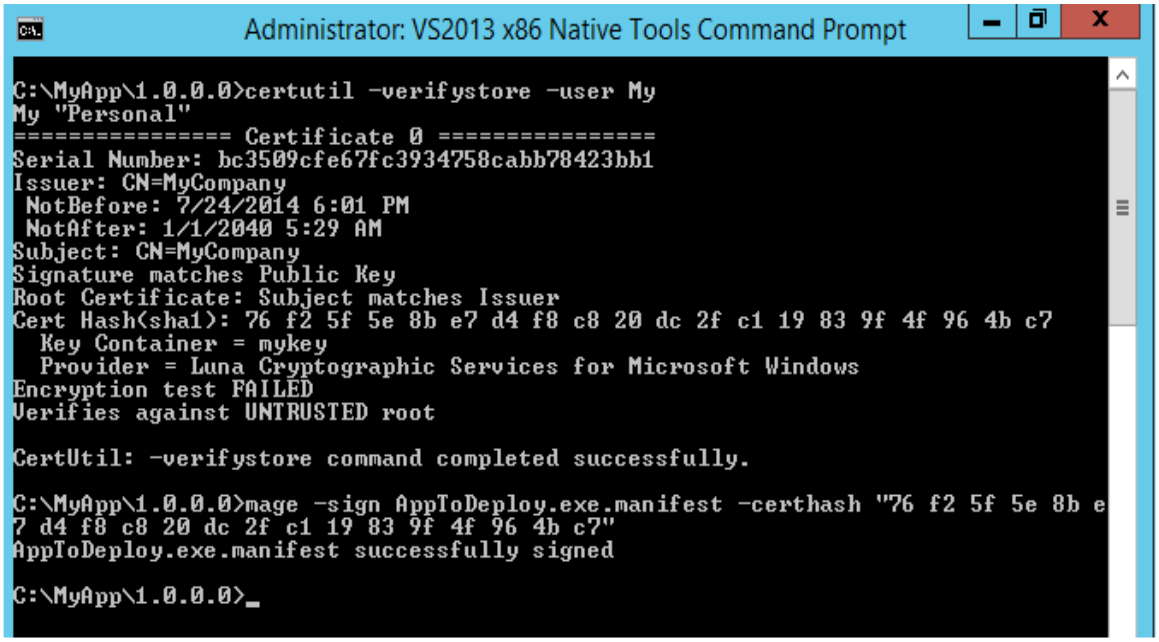


5. Sign the application manifest with Authenticode certificate.

```
mage -Sign AppToDeploy.exe.manifest -CertHash "Certificate Hash"
```

6. Run the certutil.exe to know the certificate hash value the command would be `Certutil -verifystore -user My`.

   `My` is the user cert store where the certificate is generated using the `makecert` command.

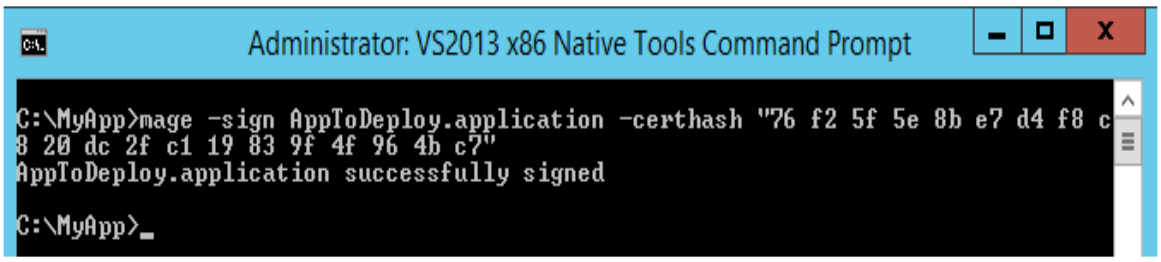


7. Change to the root of the deployment directory.
8. Generate the deployment manifest with a call to Mage.exe. By default, Mage.exe marks your ClickOnce deployment as an installed application so that it can be run online as well as offline. To make the application available only when the user is online, use the -Install option with a value of false. If default option is used and the user installs the application from a website or file share, ensure that the value of -ProviderUrl option points to the location of the application manifest.

```
mage -New Deployment -Processor msil -Install true -Publisher "My Company"
-ProviderUrl "\\myServer\myShare\AppToDeploy.application" -AppManifest
1.0.0.0\AppToDeploy.exe.manifest -ToFile AppToDeploy.application
```

9. Sign the deployment manifest with the Authenticode certificate.

```
mage -Sign AppToDeploy.application -CertHash "Certificate Hash"
```



10. Copy all the files in the deployment directory to the deployment destination or media. This may be either a folder on a website or FTP site, a file share, or a CD-ROM.
11. Provide your users with the URL, UNC, or physical media required to install your application. If you provide a URL or a UNC, you must give your users the full path to the deployment manifest. For example, if AppToDeploy is deployed to **`http://webserver01/`** in the AppToDeploy directory, the full URL path would be `http://webserver01/AppToDeploy/AppToDeploy.application`.

Integrating Luna HSM with Microsoft HCK

Microsoft's Windows Certification Program is designed to help your company deliver compatible and reliable systems, software, and hardware products. Luna HSM is used to secure the signing keys so that your signing keys cannot be accessed by any unauthorized entity. Microsoft HCK uses RSA keys for signing packages. This integration guide contains the following topics:

> Configure SafeNet Cryptographic Storage Provider

> Sign the package using CA signed certificate

> Sign the package using self-signed certificate

Configure SafeNet Cryptographic Storage Provider

To use Microsoft Mage/ClickOnce with a Luna HSM configure the SafeNet Cryptographic Service Provider (CSP) to generate the keys and certificates for Microsoft Mage/ClickOnce. To configure the SafeNet CSP:

1. Install Luna Cryptographic Service Provider (CSP) on Windows Server.
   a. Open the command prompt and run *register.exe* to register Luna CSP. The general form of command is given below:

      ```
      <Luna Client Installation Directory>\win32\csp>register.exe
      ```

   b. To register the Luna Cryptographic Services for Microsoft Windows. The general form of command is given below

      ```
      <Luna Client Installation Directory>\win32\csp>register.exe /l
      ```

2. To verify the registered cryptographic providers, browse to `"C:\Windows\SysWOW64"` and execute `"certutil –csplist"`
3. To integrate the Luna HSM with Microsoft HCK, the Luna CSP Luna Cryptographic Services for Microsoft Windows must be used to generate the certificate. The certificate must be signed and the signer certificate must be present in the "Trusted Root Certificate Authority". You can use the CA signed certificate of self-signed certificate both. There are two methods that you can use to generate the signing certificate:

Sign the package using CA signed certificate

To sign the package using a CA signed certificate:

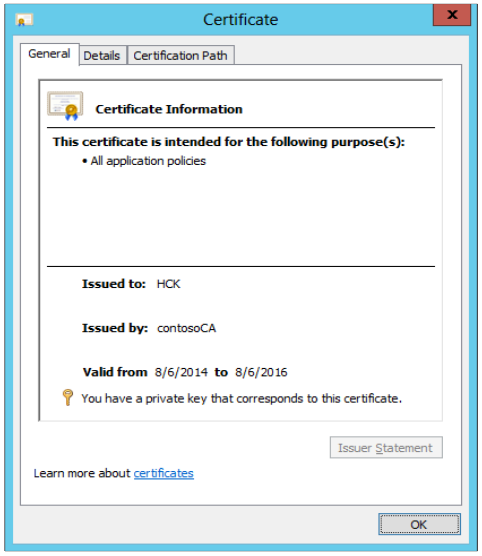
1. Create an inf file with the following attributes:

   ```
   [Version]

   Signature="$Windows NT$"

   [NewRequest]
   ```

```
Subject = "C=US,O=SafeNet,CN=HCK,OU=HCKIntegration"

KeySpec = 1

KeyLength = 2048

Exportable = FALSE

MachineKeySet = TRUE

KeyContainer = HCK

ProviderName = "Luna Cryptographic Services for Microsoft Windows"

ProviderType = 1

KeyUsage = 0x04
```

2. Generate a certificate request using the inf file. Use the 32 bit **certreq** utility from the `"C:\Windows\SysWOW64"` directory. You will see a success message after the execution.
3. Take the generated certificate request to a CA and get it signed to obtain a signed certificate.
4. Now we have to import this obtained certificate in the user's personal certificate store. As this setup is 32 bit, use the 32 bit Microsoft Certificate manager console.

   `C:\Windows\SysWOW64\certmgr.msc`
5. Right-click on **Personal** -> **All Task** -> **Import** and follow the instruction to import the signed certificate. Verify the certificate is successfully imported.
6. Double-click the certificate and confirm that there is a private key mapped with this certificate. Check the message at the bottom. If the private key is not mapped correctly, repair the certificate using the "certutil –repairstore" utility.
7. Open the certificate, browse to the **Details** tab, and select the **Serial number** field.
8. Copy the serial number or thumb print and execute the `"certutil -repairstore -user My "SerialNumber or ThumbPrint"` command from the **SysWOW64** directory to map the private key (on the HSM) with the certificate.
9. After the `repairstore` command has been successfully executed, refresh the certificate manager snap in, open the certificate, and confirm the message at the bottom is displayed.



Sign the package using self-signed certificate

To sign the package using a self-signed certificate:

1. Use the makecert utility to generate a self-signed certificate. Browse to the "C:\Program Files (x86)\Windows Kits\8.1\bin\x86" directory and execute the following command:

   ```
   makecert -sk <keyContainer> -sp "Luna Cryptographic Services for Microsoft Windows" -r  -n "CN=Common Name" -ss <certStore> CertificateName.cer
   ```

   where:

   -sk The location of the subject's key container which holds the private key

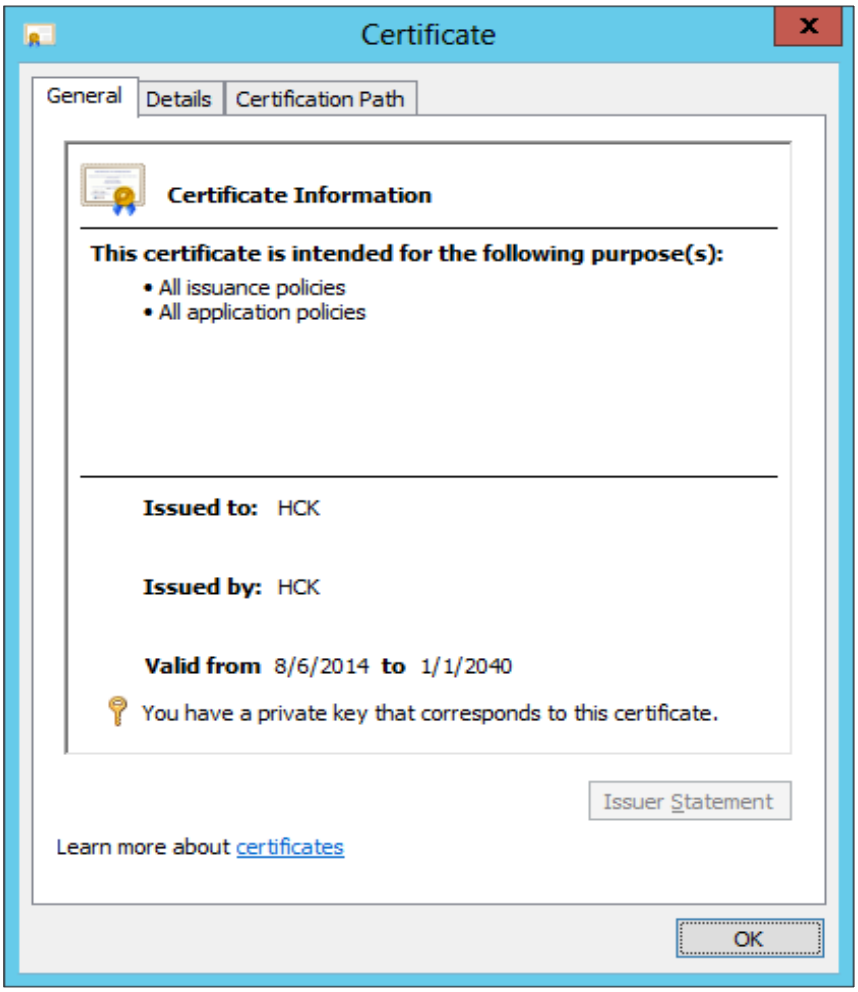
   -sp Subject CryptoAPI's provider name

   -n The name and details of the signer's certificate

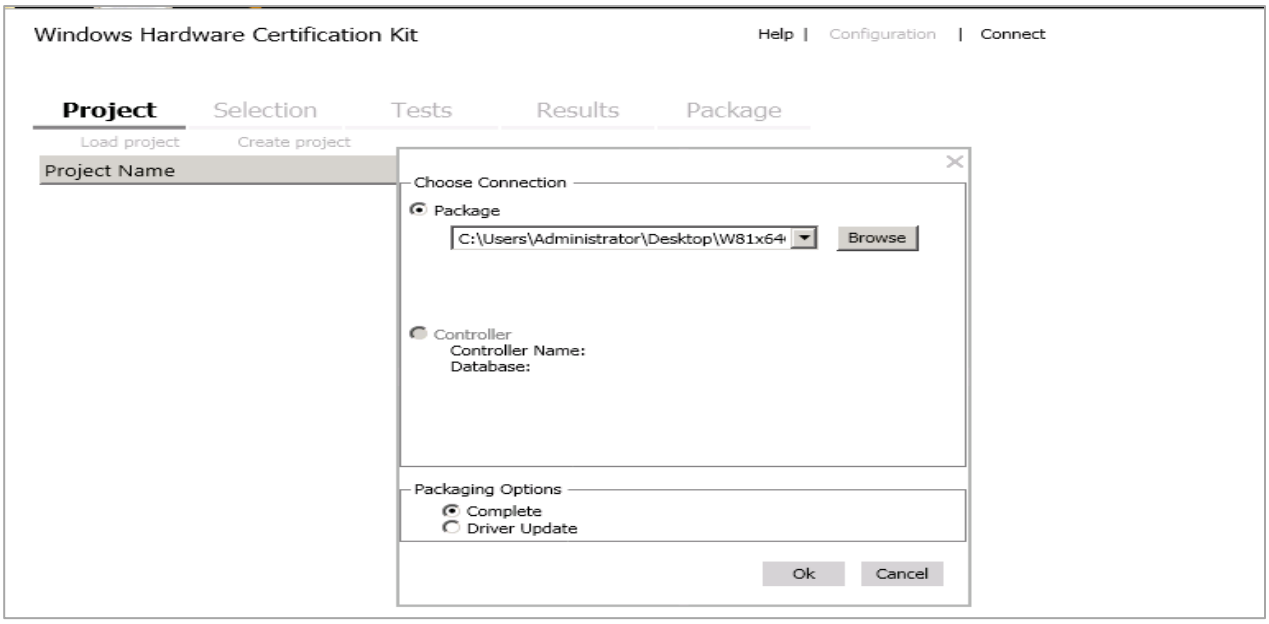
   -ss The name of the subject's certificate store in which the generated certificate will be stored. Use "My" which is the default user cert store where the application is looking for certificate.

2. Open the **certmgr.msc** from the "C:\Windows\SysWOW64" directory and export the generated certificate from the Personal folder.

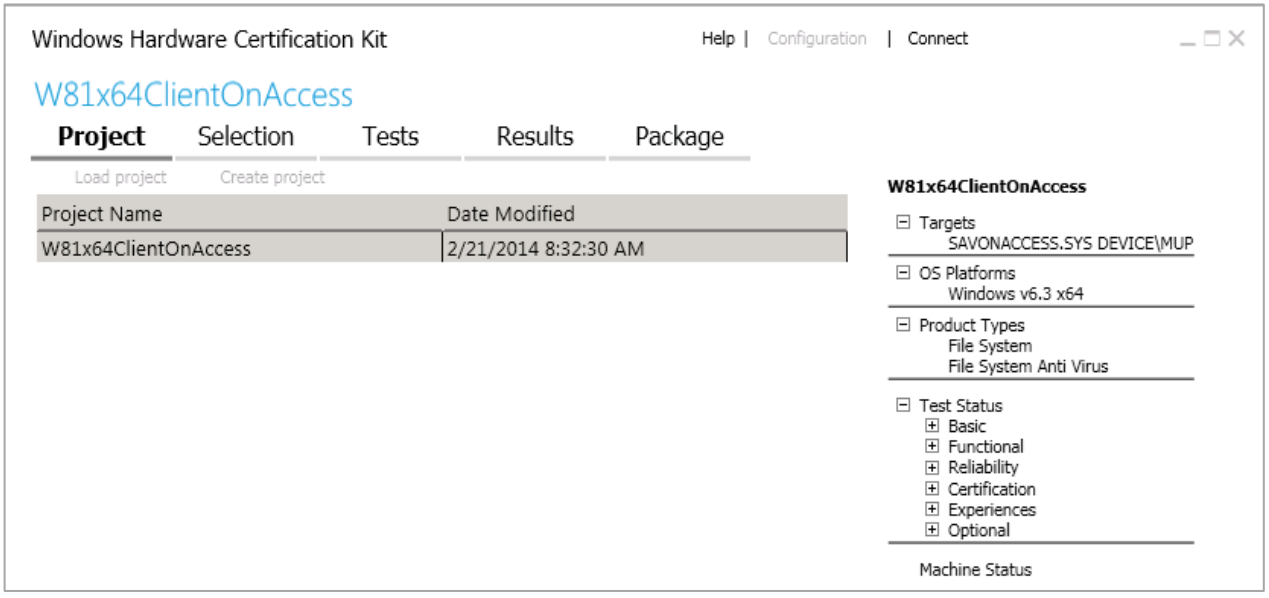
3. Import the certificate in the **Trusted Root Certificate Authority** folder. Verify that certificate imported successfully.

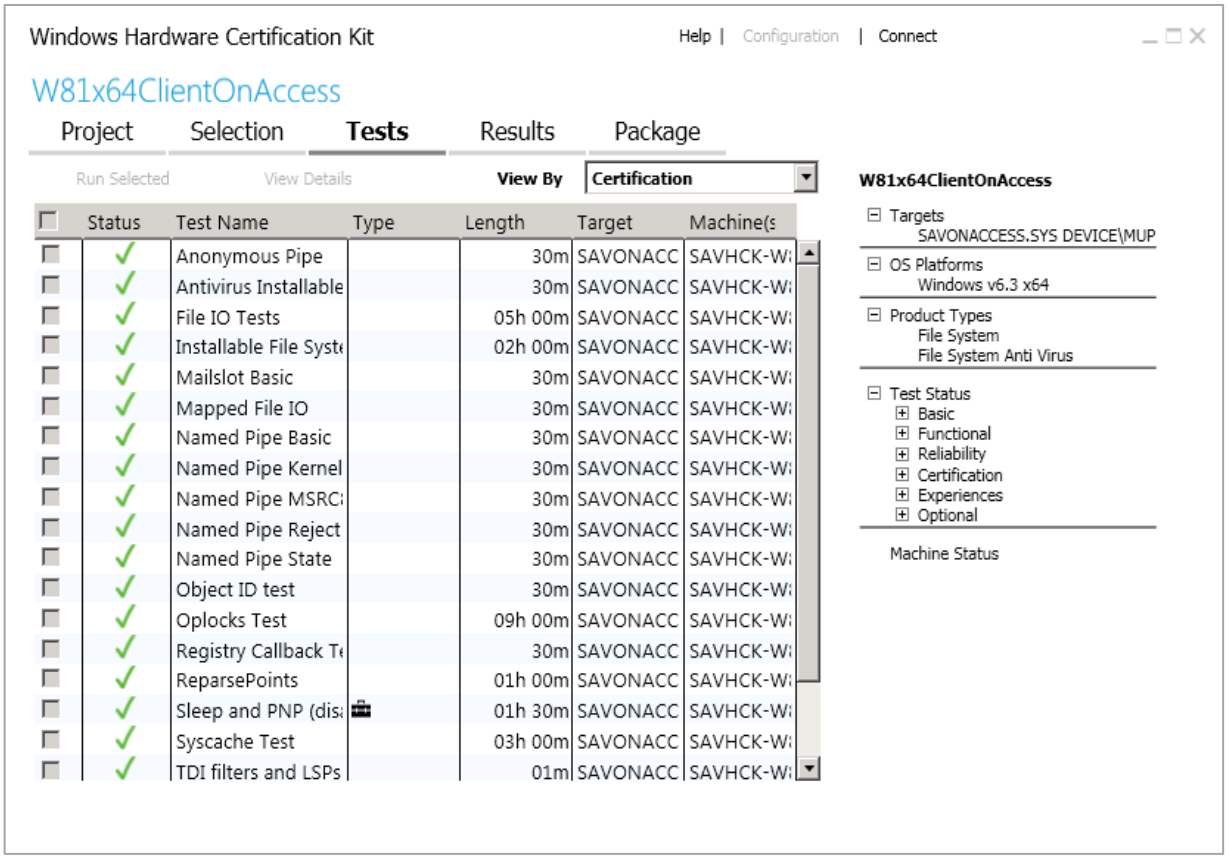


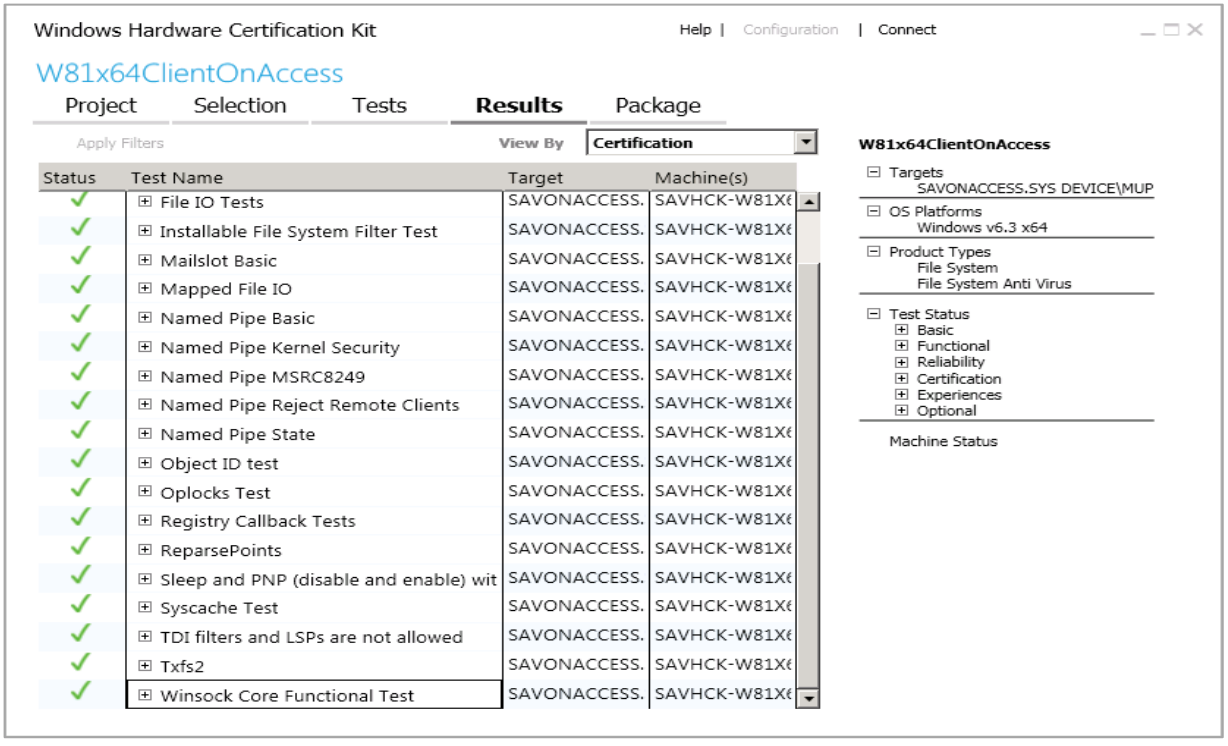
4. Now, when the certificate and the private key is ready for signing, open Windows Hardware Certification Kit and import the project to sign.



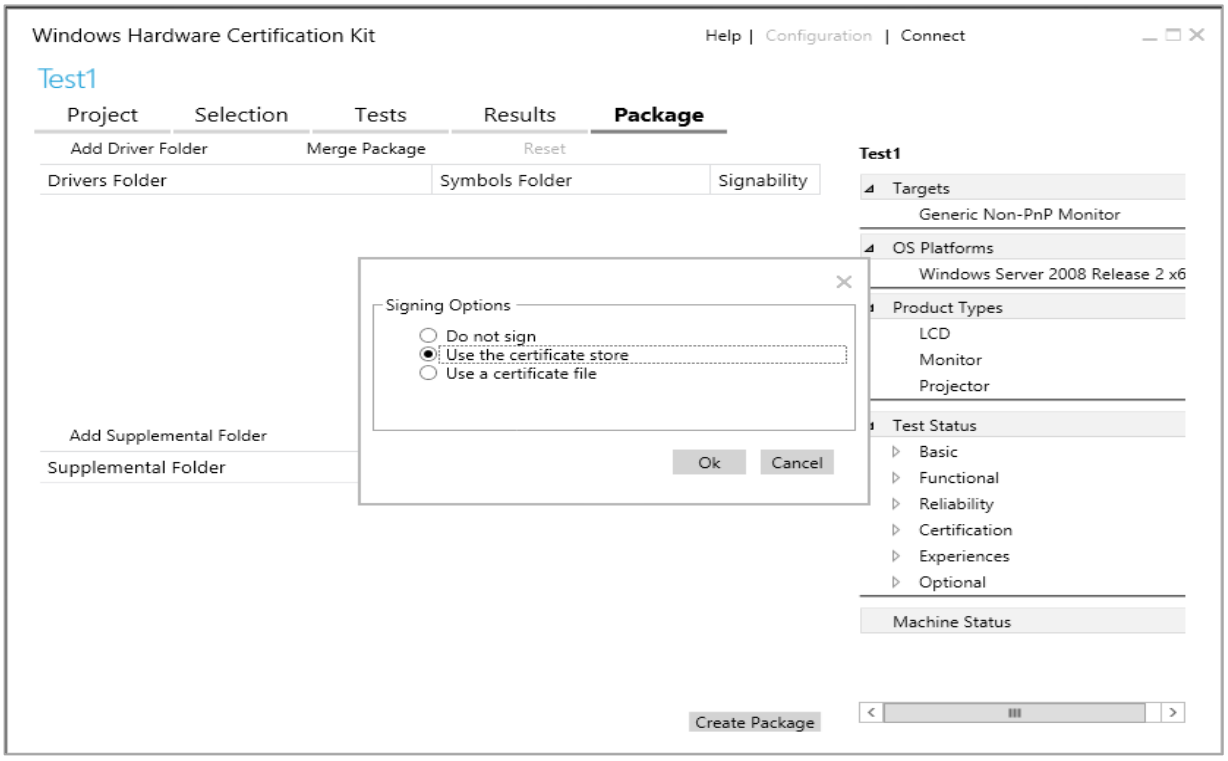
5. Navigate through various tabs to verify the project imported is correct.



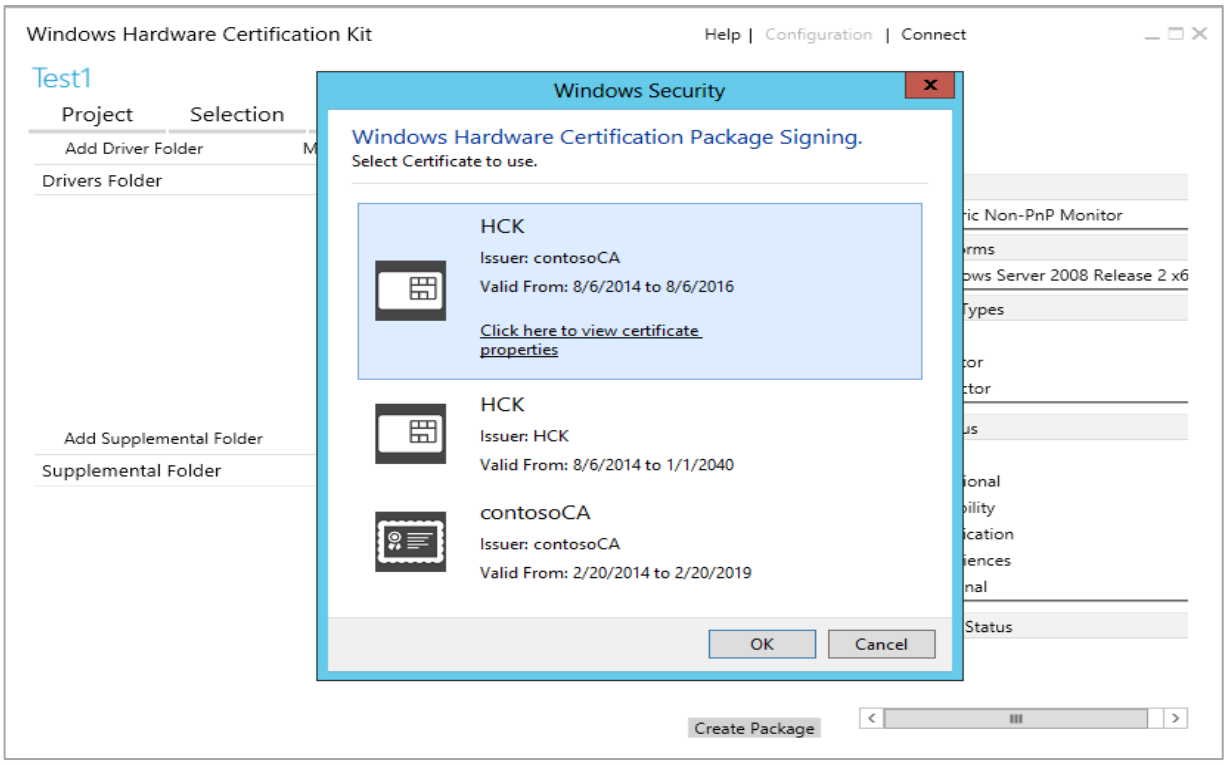




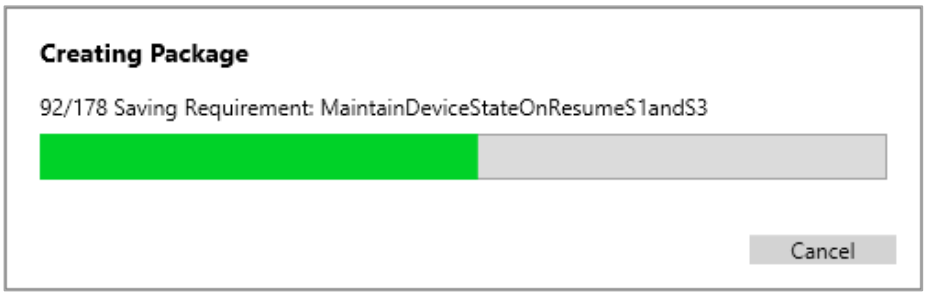
6. After verification, go to the package tab and click on create package to sign the package. It prompts for **Signing Options** Select **Use certificate Store** and click **OK.**



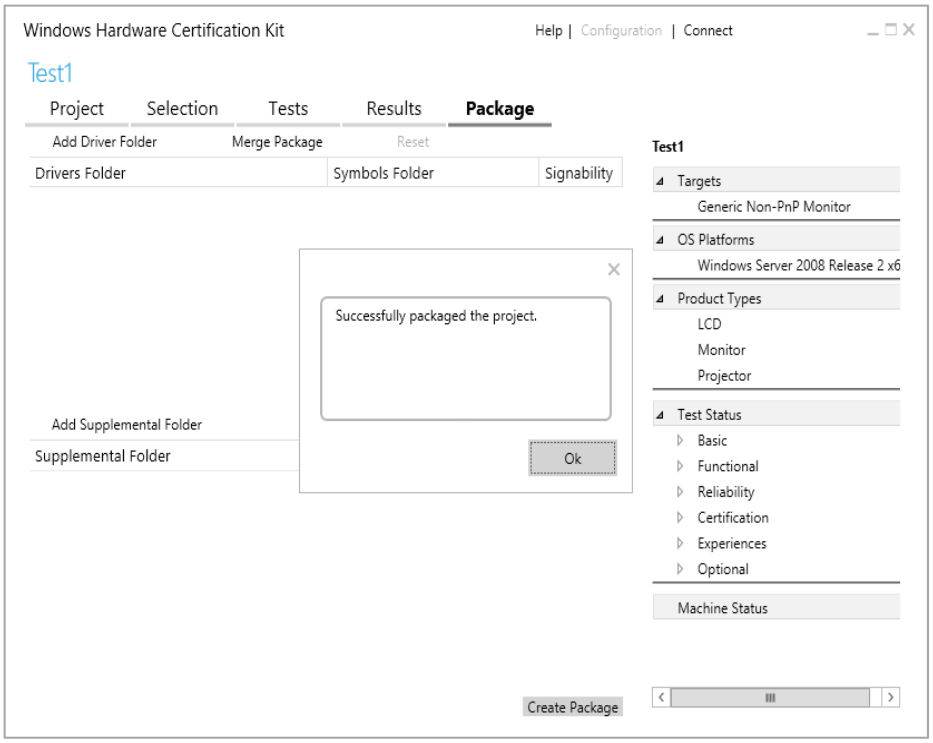
7. Select the signing certificate. From the pop up, select the certificate that was imported earlier on the local machine's personal certificate store and click **OK**.



8. Select a location to save the signed package and click **Save**.
9. Click **Save** to start signing. Signing starts with a **Creating Package** window.



10. A success message appears and you can verify the signed package in the location you saved it.



Troubleshooting

Problem

You encounter the following error when running the makecert command on an HSM in FIPS mode.

Error: CryptHashPublicKeyInfo failed => 0x80090005 (-2146893819) Failed.

Solution

The cert always has an MD5 hash. Configure the Luna CSP to do MD5 in software. The command is:

```
<Luna Client Installation Directory>\CSP > Register.exe /algorithms
```

It prompts you to register the various algorithms; you need to register the MD5 algorithms in software.

Contacting Customer Support

If you encounter a problem while installing, registering, or operating this product, refer to the documentation. If you cannot resolve the issue, contact your supplier or Thales Customer Support. Thales Customer Support operates 24 hours a day, 7 days a week. Your level of access to this service is governed by the support plan arrangements made between Thales and your organization. Please consult this support plan for further information about your entitlements, including the hours when telephone support is available to you.

Customer Support Portal

The Customer Support Portal, at https://supportportal.thalesgroup.com, is a database where you can find solutions for most common problems. The Customer Support Portal is a comprehensive, fully searchable repository of support resources, including software and firmware downloads, release notes listing known problems and workarounds, a knowledge base, FAQs, product documentation, technical notes, and more. You can also use the portal to create and manage support cases.

> **NOTE:** You require an account to access the Customer Support Portal. To create a new account, go to the portal and click on the **REGISTER** link.

Telephone Support

If you have an urgent problem, or cannot access the Customer Support Portal, you can contact Thales Customer Support by telephone at +1 410-931-7520. Additional local telephone support numbers are listed on the support portal.

Email Support

You can also contact technical support by email at technical.support.DIS@thalesgroup.com.