Black•Vault HSM

Authenticode

Integration Guide

9565 Soquel Drive
Aptos, CA 95003
Phone +1 831.688.1021
1 877.ENGAGE4 (364.2434)
sales@engageinc.com

1. Disclaimer and Warranty

Engage Black is a business unit of Engage Communication.

Engage Communication makes no representations or warranties with respect to the contents of this document and specifically disclaims any implied warranties of merchantability of fitness for any particular purpose. Information in this document is subject to change without notice and does not represent a commitment on the part of Engage Communication, Inc. Product specifications are subject to change without notice. Engage Communication assumes no responsibility for any inaccuracies in this document or for any obligation to update the information in this document.

Engage Communications, Inc.9565 Soquel Drive Aptos, CA 95003Phone +1(831) 688-1021
http://www.engageblack.com/http://www.engageinc.com/

2. Table of Contents

3. Introduction

The BlackVault Hardware Security Module (HSM) integrates with Microsoft Authenticode to enable you to identify the publisher of a software component before it is downloaded from the Internet, and to verify that no one has altered the code after it has been signed. Microsoft 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.

The benefits of using an HSM with Microsoft Authenticode include:

- Protection for the organizational credentials of the software publisher.
- Secure storage of the private key.
    - Signing code within a cryptographically secure environment

- FIPS 140-2 level 3 validated hardware.

3.1. Supported Operating Systems

Supported operating systems

OS Name	Version	32 bit	64 bit
Windows	7	X	X
	Server 2008 R2 x64		X
	8.1	X	X
	Server 2012 x64		X
	Server 2012 R2 x64		X
	10	X	X
	Server 2016 x64		X

4. Procedure

To proceed the following is needed:

- BlackVault HSM
- BlackVault Card Set
- BlackVault HSM Setup CD
- A client computer that has a supported Operating System installed.

Additionally, the BlackVault must be Initialized and Configured properly (see section 6.3 and 6.4 of the BlackVault HSM User Guide for more details)

To setup Authenticode with the BlackVault HSM:

- Initialize the HSM
- Run the Setup Wizard (included on the setup CD) to install the BlackVault HSM Libraries onto the client machine and configure Authenticode.
- Validate operation (i.e. create test key …. Etc.)

You can find information about how to initialize the BlackVault HSM in the BlackVault User Guide

The following assumes you already initialized the BlackVault HSM and are installing this software on a machine that does not already work with the BlackVault.

4.1. Install BlackVault Library and integrate with Authenticode

The BlackVault communicates over the network using PKCS#11. For Windows to communicate with the BlackVault HSM, the BlackVault HSM's PKCS#11 library, CNG library, and pkcs.dat must be installed first.

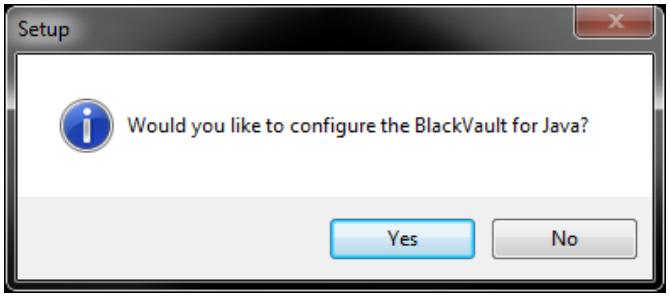
1. Copy the BlackVault Setup CD to a directory on your system. In that directory do the following:

2. Run bv-setup.exe

3. Windows asks if you want the installer to make changes to computer, select yes.

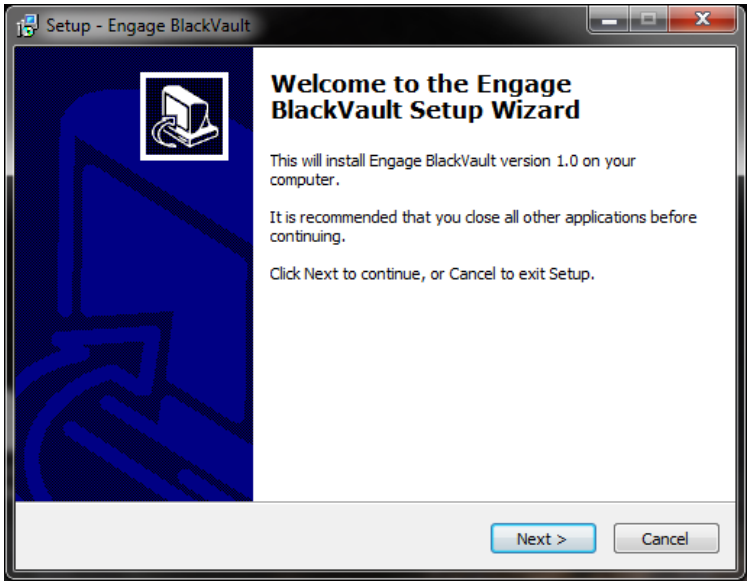
4. The installer asks, “Would you like to configure the BlackVault for Java?” select yes or no.

a. if no is selected, skip step 6

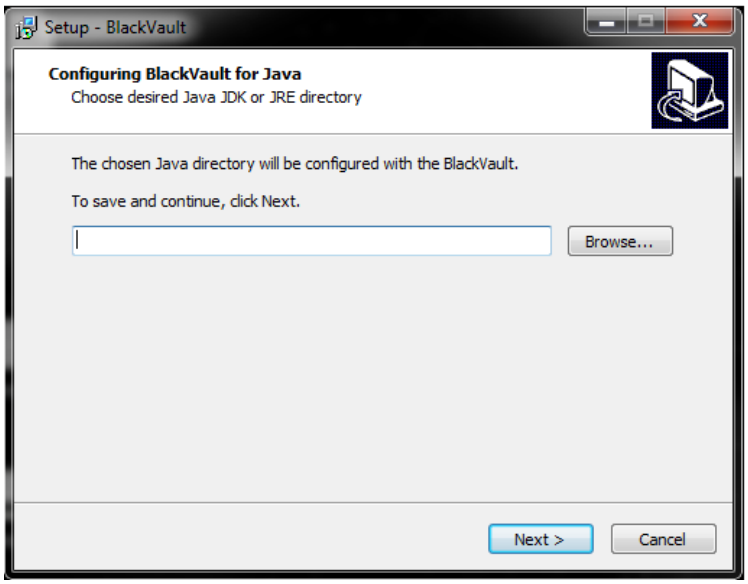


b. If yes is selected, perform step 6

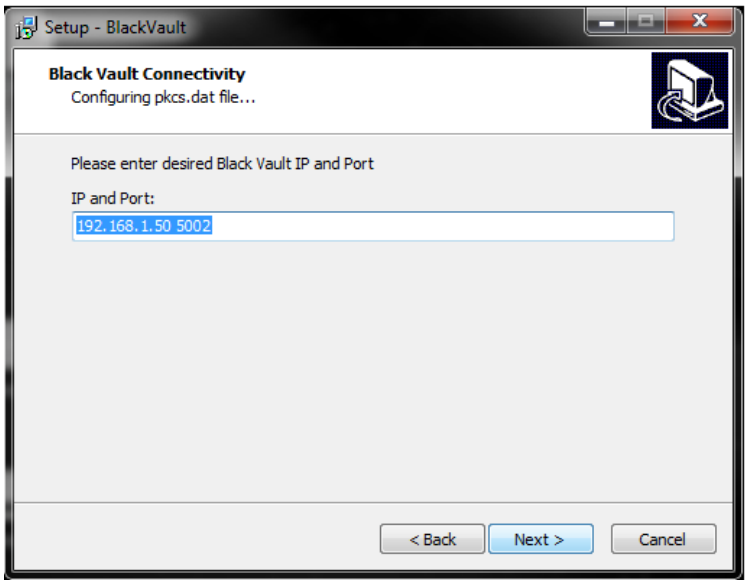
5. The first window you will see is the overview window. It goes over what will be installed. Select Next to continue.



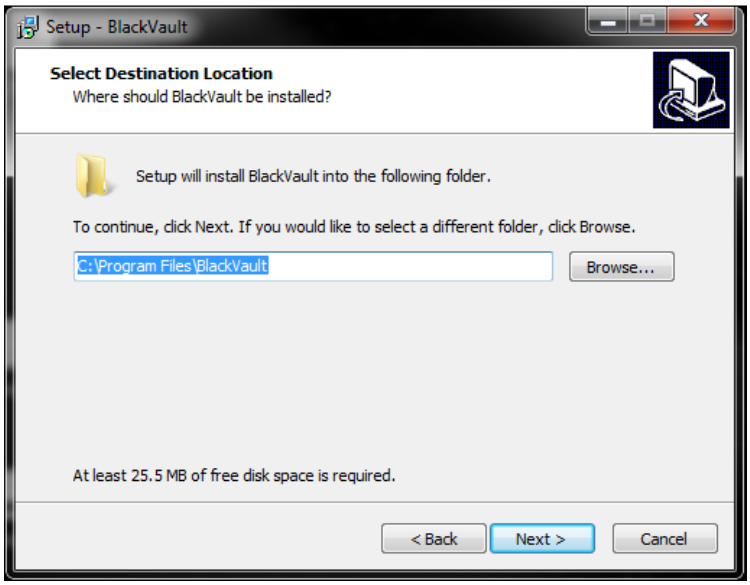
6. Next the installer asks for the Java directory currently being used on Windows. Browse for the desired top-level Java directory (usually in C:/Program Files/Java/jre8 or C:/Program Files(x86)/Java/jre8) then press next



7. Next the installer asks for the BlackVault IP address and TLS Port. Enter those there and press next.

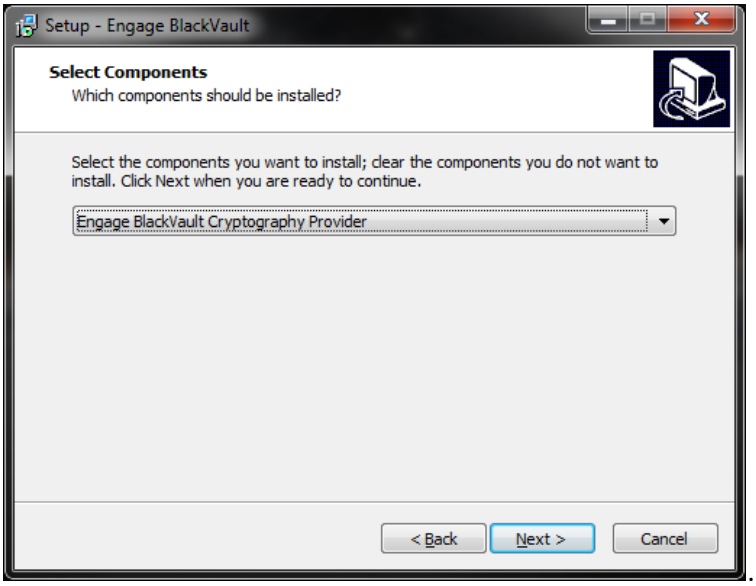


8. Next, the installer asks for the location of the directory to install the necessary files to. Either use the default location, or to select a new location, click browse and specify the new location.

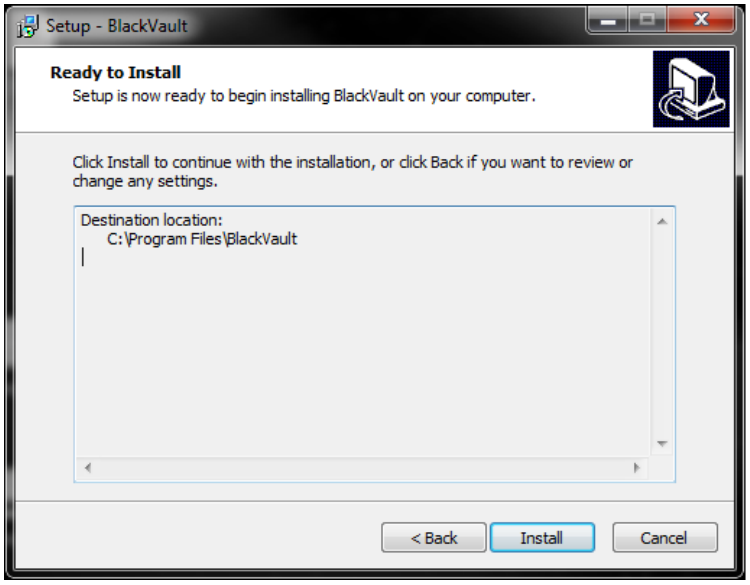


9. Next, the installer asks for which components to install, the Engage BlackVault Cryptography Provider (CNG/

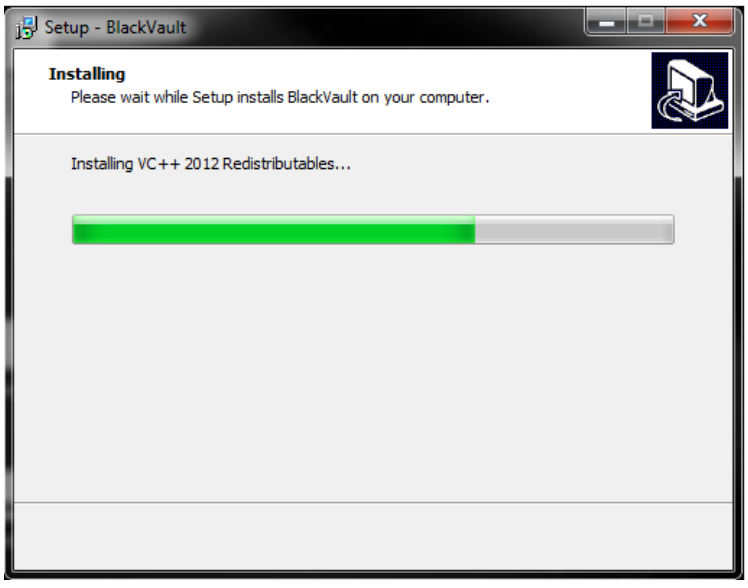
and PKCS#11 libraries), or just the PKCS#11 Library

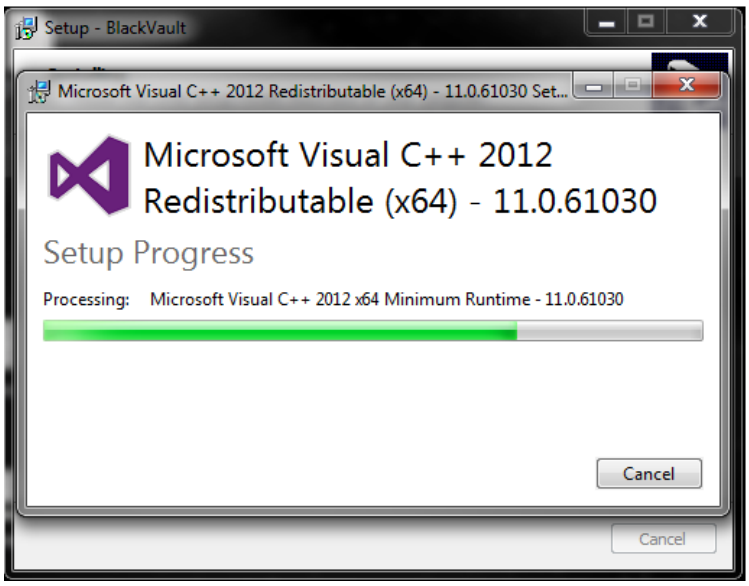


10. The installer then gives a summary of items to be installed press install to continue.

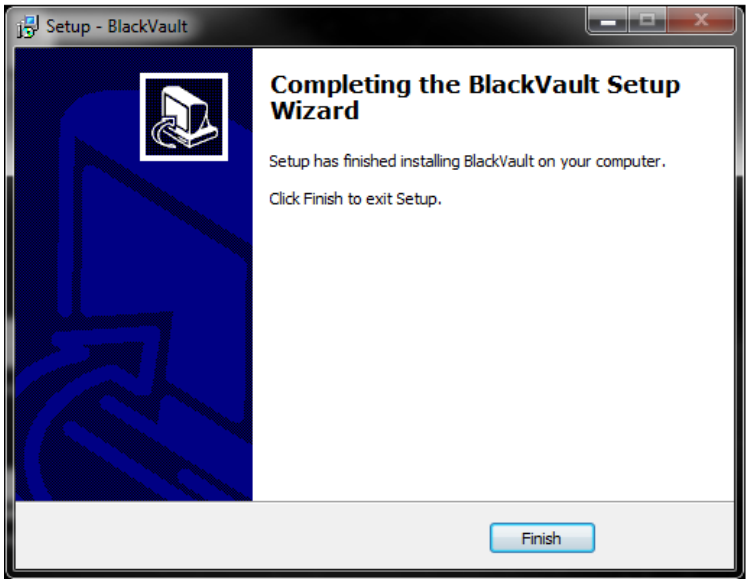


11. The installer then installs all the components, including a Microsoft Visual C++ Redistributable (if not already installed)





12. Once the installer finishes press finish.



4.2. Signing with Microsoft Authenticode and the BlackVault HSM

To perform code signing per industry best practices, along with creating and storing the key inside a secure HSM, a code signing certificate associated with the key is required. This section first describes how to create a key, and then how to create the certificate using a self-signed certificate authority managed by Openssl. If you require a chain of trust to a Certificate Authority (CA), replace the openssl commands with taking the CSR to the CA and have the CA sign your CSR.

For self-signed certificates, first download and install Openssl for windows found here.

1. Create a key

- In a command prompt run the command: bvtool genkey -n NAME -t TYPE -s SIZE -c CURVE -x
  - -n desired name of keys to be made
  - -t type of key aes, rsa, ec, dsa, generic
  - -s size of key if aes, rsa, dsa or generic is chosen
  - -c curve name if ec is chosen
  - -x use ANSI x9.31 for RSA key generation
- You do not have to use all of the arguments, only the relevant ones.
- Example:
  - Bvtool genkey -n NANE -t RSA -s 2048 -x
  - Bvtool genkey -n NAME -t EC -c prime256v1

2. Create a certificate

- In a command prompt run the command: Certreq –new file.inf csr.pem where:
  - "file.inf" is the inf file that specifies the key and other information about the key. For more information look at the "bvrsaex.inf" and "bvecex.inf" in section 5.1
  - "csr.pem" is the output certificate signing request file.
  - To use an existing key, in the .inf file, change "useexistingkeyset" to "true".

3. Create the Openssl CA using the following command:

- openssl req -x509 -newkey rsa:4096 -keyout rootkey.pem -out rootcert.cer -days 365 -subj "/C=US/ST=state/L=city/O=company/OU=division/CN=common name"

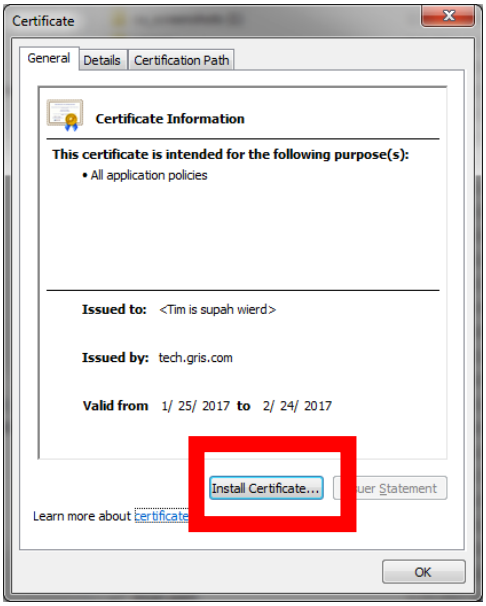
4. Sign the certificate created in step 1 with the CA created in step 2 using the following command:

- openssl x509 -req -in csr.pem -sha256 -extfile usercert.cnf -CA rootcert.pem -CAkey rootkey.pem -CAcreateserial -out mycert.cer -outform der

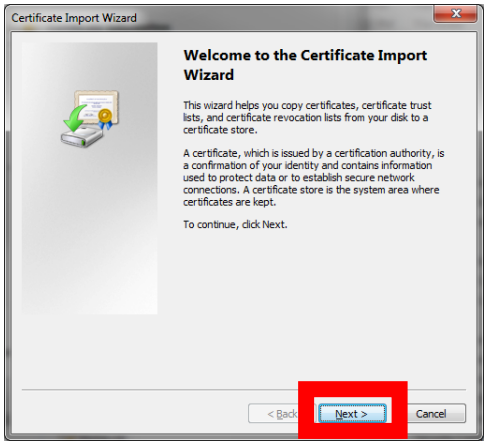
5. Import the root certificate and the certificate just generated into windows.

- To do this double click the CA certificate **rootcert.cer**

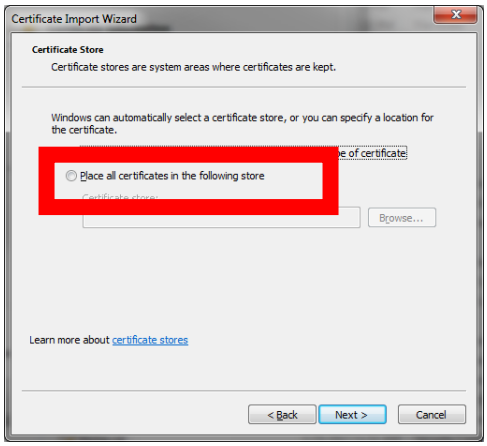
6. In the certificate dialogue box that shows up click **Install Certificate**



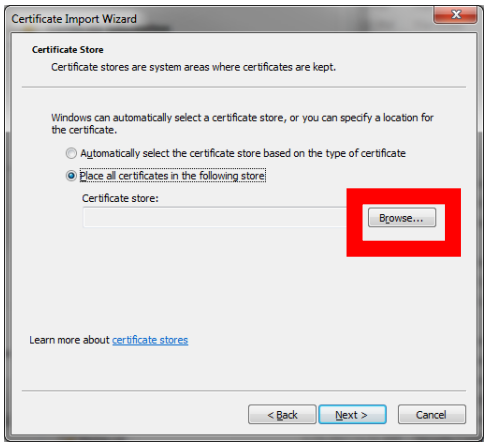
- Click **next**



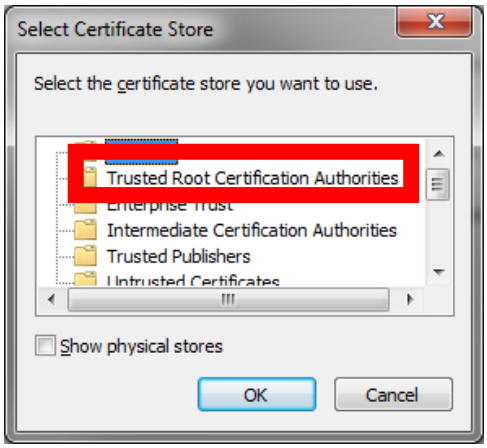
- The screen below is then displayed and select the radio button **Place all certificates in the following store**



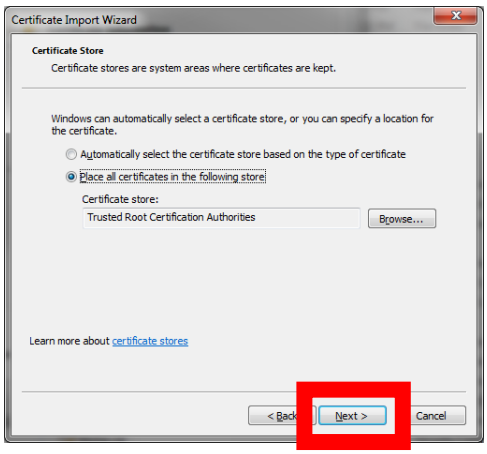
- Click **browse**



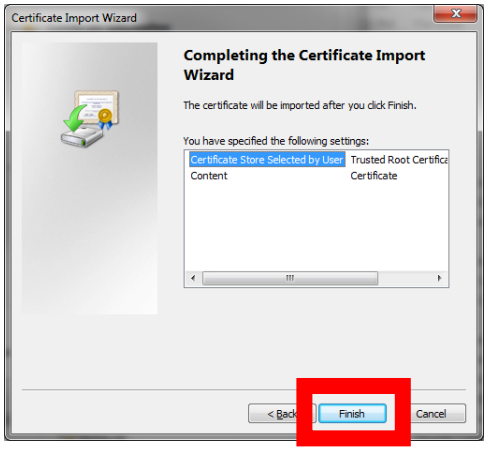
- Select **Trusted Root Certification Authorities** and select **OK.**



- Click **next**

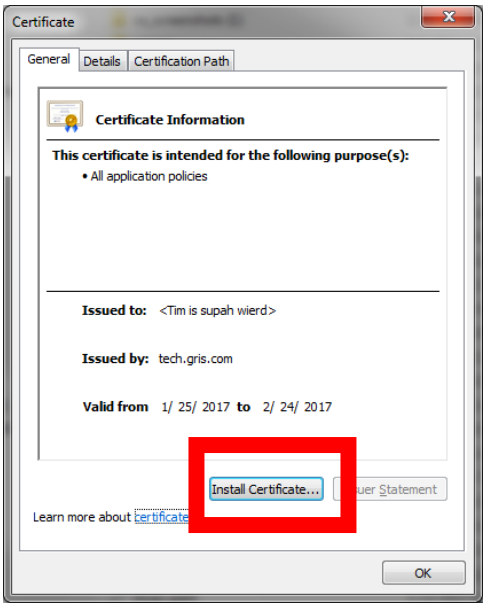


- Click **Finish**

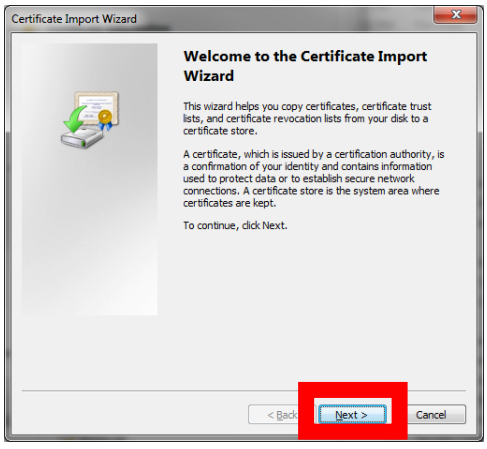


7. Then double click the certificate created **mycert.cer**

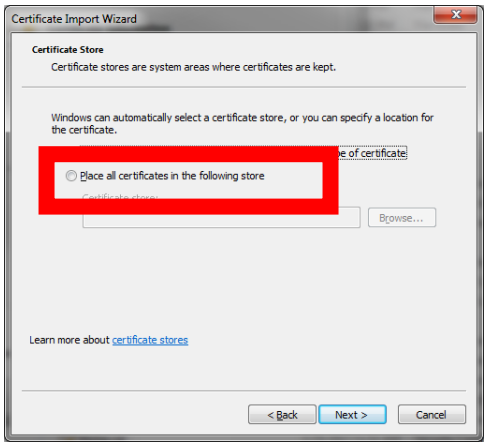
- In the certificate dialogue box displayed, click **Install Certificate**.



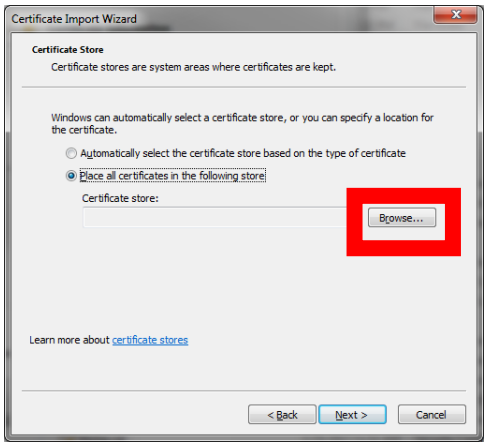
- Click **Next**



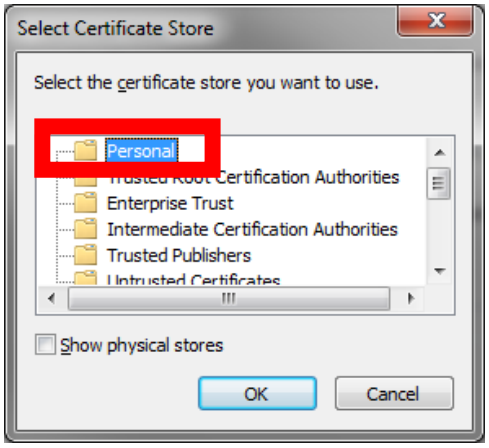
- Now select the radio button **Place all certificates in the following store**



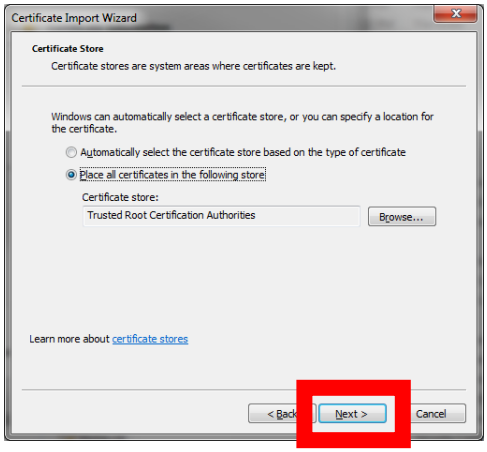
- In the new dialog box, click **Browse**.



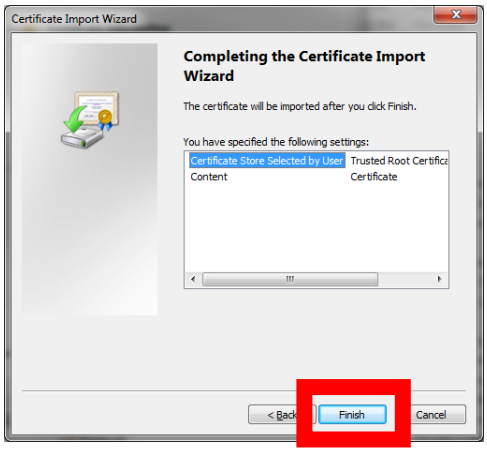
- Then select **Personal** and click **OK.**



- Click **next**



- Click **Finish**



8. Run the command “bv_associate.exe”

- In a command prompt run the command “bv_associate.exe “Certificate CN” “key name” where:
  - “Certificate CN” is the common name of the certificate installed
    - If you need to find the common name of the certificate see section 5.2
  - “key name” is the name of the key located on the BlackVault

9. Sign a file.

- In a command prompt run the command: “signtool.exe sign /debug /s MY /n "Certificate CN” file.exe”where:
  - “sign” is the call used to sign with signtool
  - “/debug” turns on debug information, in case debugging is needed
  - “/s MY” is the cert store, don’t change this field
  - “/n “Certificate CN”” is the common name of the certificate, so it knows which certificate to use to sign the file
    - If you need to find the common name of the certificate see section 5.2
  - file.exe is the demo file to sign. Replace with any exe or dll file needing signature.

10. Verify the signature

- In a command prompt run the command “signtool verify /v /pa pumpkin-2.7.2.exe”where:
  - “verify” is the call used to verify with signtool
  - “/v” means verbose, don’t change this field
  - “/pa” is the default Authenticode verification policy, don’t change this field
  - “pumpkin-2.7.2” is the demo file signed in the previous step. Replace it with the name of the actual file signed.

5. Appendix

5.1. .inf examples

The examples in this section show how certreq inf files should be formatted. Examples of a 2048 RSA key and a P256 ECDSA key are provided. For more information about certreq nad inf files please visit Microsoft documentation found here.

5.1.1. Bvrsaex

The following is an example of a 2048 RSA key. Simply copy and paste it (from [NewRequest] to UseExistingKeySet = false) into a blank notepad, change any parameters desired, then save the file as “file.inf” (where file is a name provided by you)

[NewRequest]

Subject = "CN=<Common Name Goes Here>"

KeyContainer = Key_Name_Goes_Here

HashAlgorithm = Sha256

ProviderName = "Engage BlackVault Cryptography Provider"

UseExistingKeySet = true

5.1.2. bvecex

The following is an example of a P256 ECDSA key. Simply copy and paste it (from [NewRequest] to UseExistingKeySet = true) into a blank notepad, change any parameters desired, then save the file as “file.inf” (where file is a name provided by you)

[NewRequest]

Subject = "CN=<Common Name Goes Here>"

KeyContainer = Key_Name_Goes_Here

HashAlgorithm = Sha256

ProviderName = "Engage BlackVault Cryptography Provider"

UseExistingKeySet = true

5.2. How to find common name of installed certificate

1. Open a Command Prompt window.
2. Type mmc and press the ENTER key.
3. On the File menu, click Add/Remove Snap In.
4. In the Add Standalone Snap-in dialog box, select Certificates.
5. Click Add.
6. In the Certificates snap-in dialog box, select My user account and click Next..
7. In the Select Computer dialog box, click Finish.
8. On the Add/Remove Snap-in dialog box, click OK.
9. In the Console Root window, click Certificates (Local Computer) to view the certificate stores for the computer.
10. Select Personal to view your personal certificates
11. Click certificates
12. In the windows that pops up, look for the certificate that has a little key in the image of a certificate
13. In the column “Issued to”, this is the common name.