MVS ClickOnce Signing Integration Document

CodeSign Secure can sign ClickOnce application manifests directly from Microsoft Visual Studio, using Encryption Consulting's Key Storage Provider (KSP) to reach a private key that never leaves the HSM. Every signing operation is recorded centrally, giving you strong key custody and a complete audit trail without changing the way your developers already publish applications.

Unlike signing a standalone executable, ClickOnce signing is driven by Visual Studio's publish wizard rather than by a command line tool. Visual Studio selects its signing certificate from the Windows certificate store of the current user, so the workflow has an additional prerequisite: the public certificate must first be exported from CodeSign Secure, imported into the personal certificate store, and then associated with the Encryption Consulting KSP so Windows knows which provider holds the matching private key.

In this guide, we will walk through the complete workflow — from installing the Encryption Consulting KSP through to publishing a signed ClickOnce application and confirming the result.

Before you begin, ensure that Microsoft Visual Studio is installed on the machine and that the project you intend to publish builds successfully.

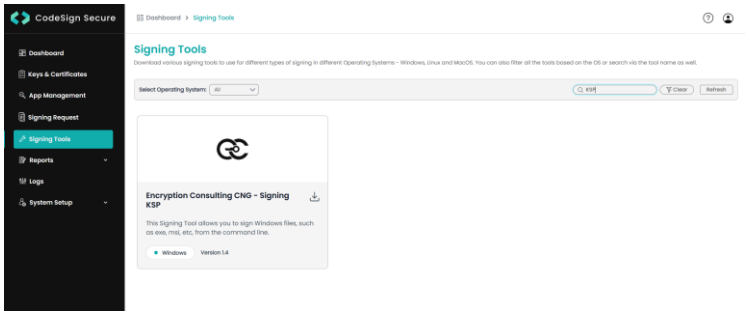
1. Set up CodeSign Secure KSP

The Encryption Consulting Key Storage Provider (KSP) for Windows is a software component that extends the Microsoft Cryptography API: Next Generation (CNG) framework. Its primary purpose is to enable Windows applications, including Visual Studio and the Windows certificate tools, to interact seamlessly with the cryptographic keys and certificates stored within an HSM. Installing the KSP also provides the ECGetCert.exe utility used later in this guide.

Steps:

Step 1: Download the EC KSP

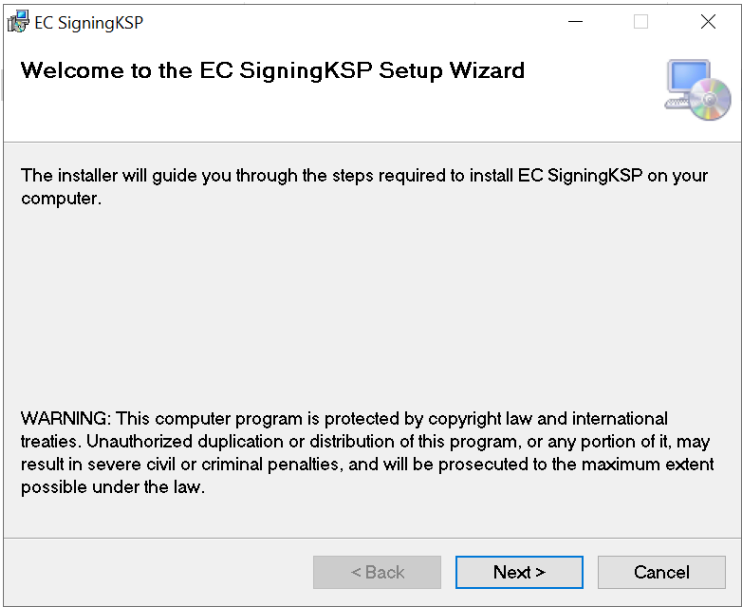
- Log in to the CodeSign Secure portal and navigate to the Signing Tools section to download "Encryption Consulting CNG-SigningKSP" (also listed as "EC KSP for Windows").



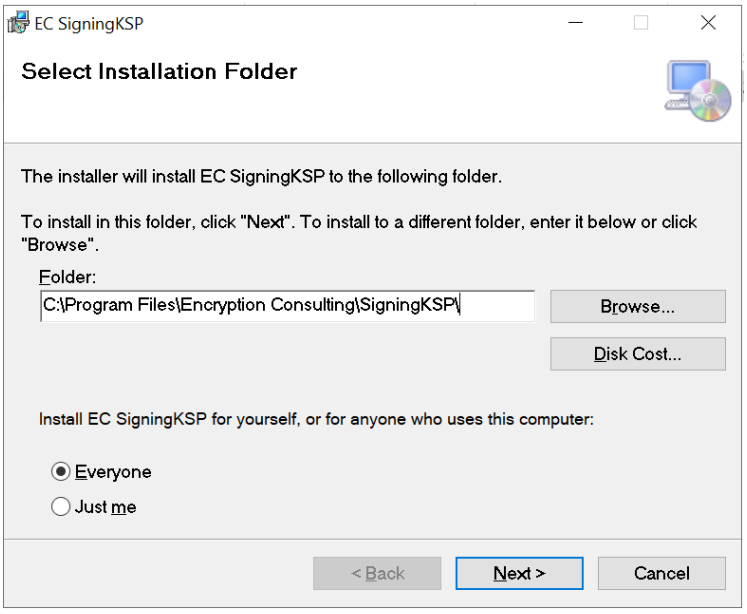
- Extract the zip file to get the "Setup.msi" file.

Step 2: Install the EC KSP

- Run the “Setup.msi” installer with Administrator privileges.

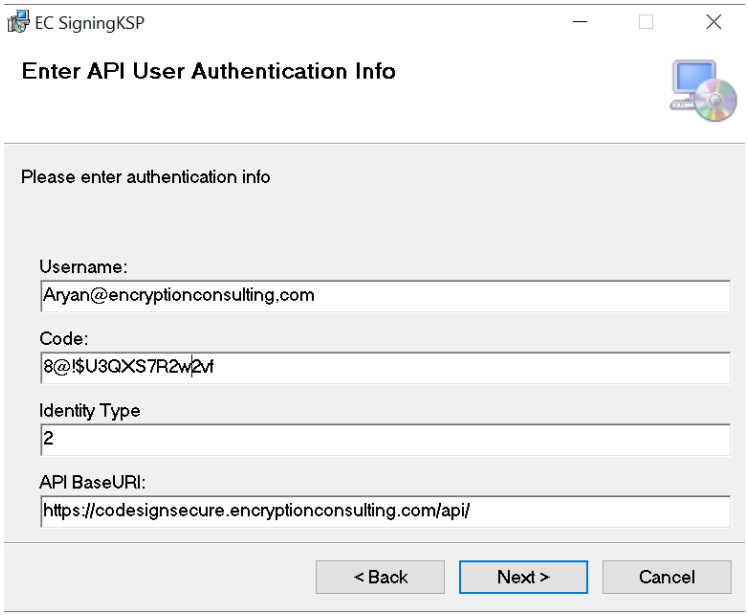


- Follow the on-screen prompts of the installation wizard.
  - Accept the End-User License Agreement.
  - Choose the installation directory (the default is C:\Program Files\Encryption Consulting\SigningKSP).
  - Choose whether you want to install the KSP for Everyone or just for the current user.

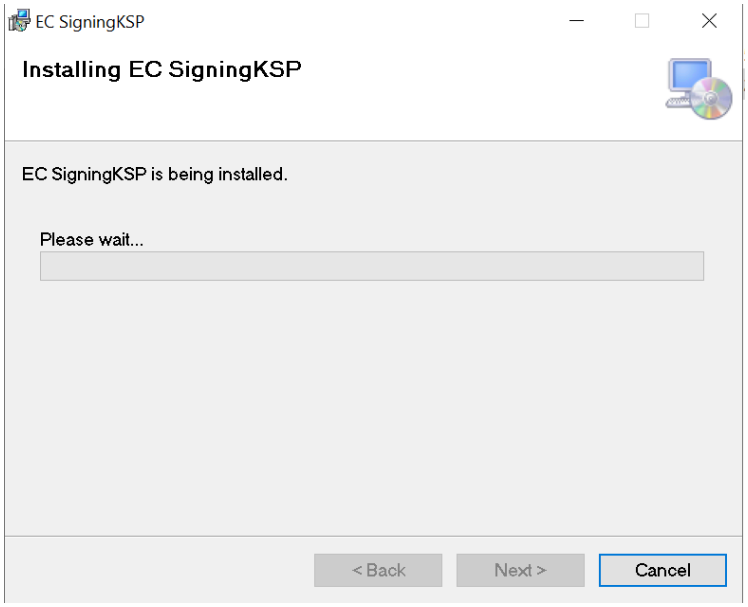


- Enter the prompted details such as:
  - **Username –** The username/email that you use to log in to the CodeSign Secure portal.

- **Code –** The secret code that you set at the time of setting up the CodeSign Secure solution (this is the code defined in your conf.ini configuration file).
- **IdentityType –** Keep this field as default (2). If your deployment authenticates against a local identity store, set this to 1 as described in the product documentation.
- **CodeSign Secure URL –** The URL to access the portal (remember to add "/api/" at the end of the URL). Leave the API BaseURL unchanged if it is already populated correctly.



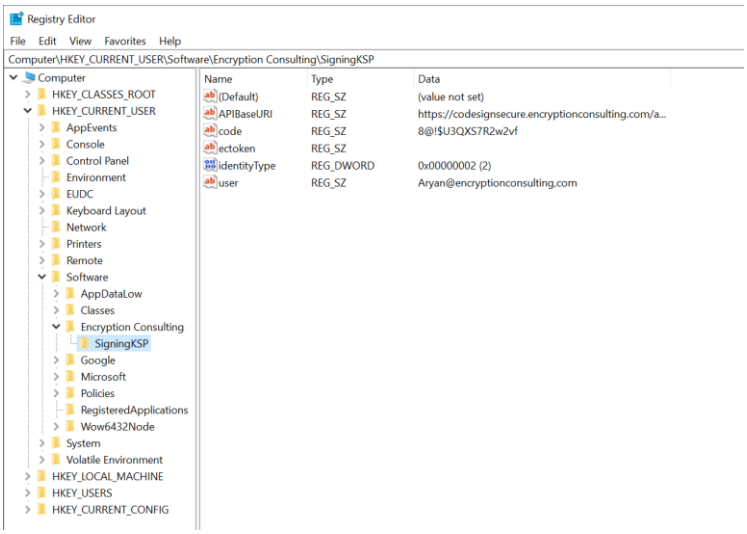
- Click Next and confirm the installation. When Windows asks whether you want to allow this program to make changes to your PC, click Yes.



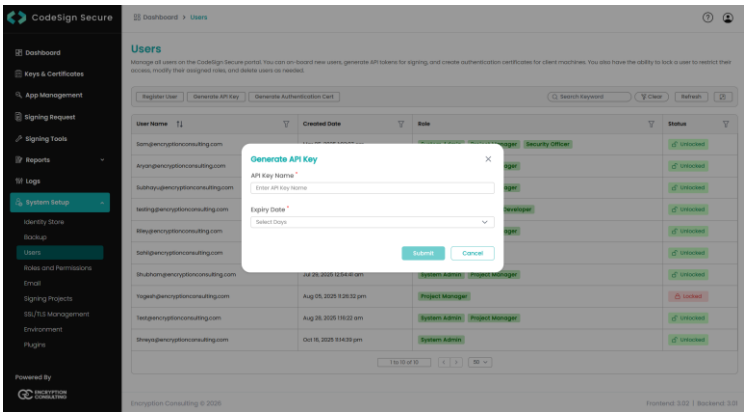
NOTE: Take note of the installation directory. The ECGetCert.exe utility used in Section 3 is located here.

Step 3: Configure the Registry Editor settings

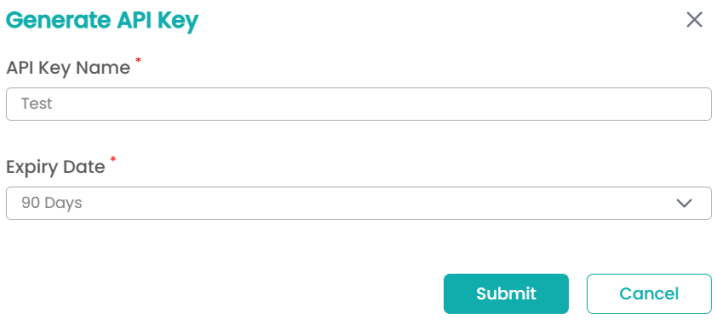
- Open the Registry Editor from the Start menu and navigate to HKEY_CURRENT_USER > Software > Encryption Consulting > SigningKSP.



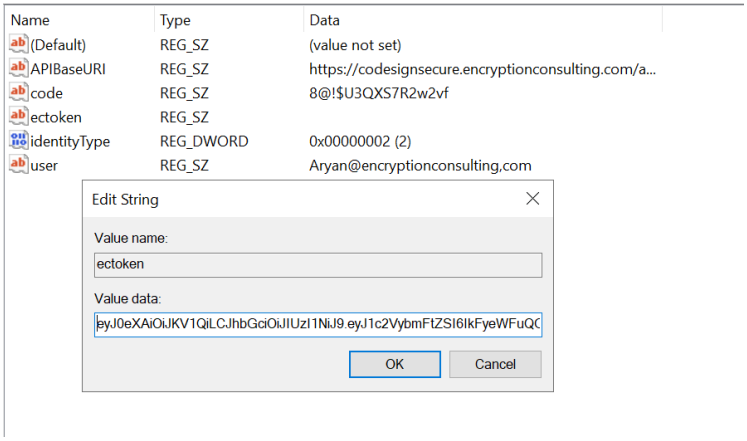
- Now open the CodeSign Secure portal and navigate to System Setup > User. Select the “Generate API Key” option.



- Create a token for your account by providing a name and the validity period. Remember to copy the token as it will be shown only once.



- Add this token to the “ectoken” field in the Registry Editor.



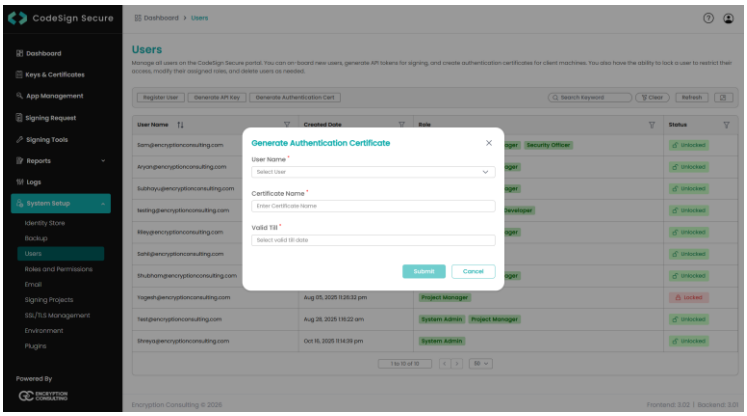
2. Set up P12 Authentication Certificate

Setting up a P12 Certificate involves configuring your environment variables to authenticate your client machine with Encryption Consulting's CodeSign Secure.

Steps:

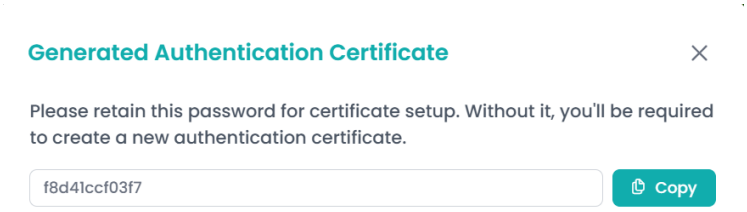
Step 1: Create a Machine Authentication Certificate

- Open the CodeSign Secure portal and navigate to System Setup > User. Select the "Generate Authentication Cert" option.



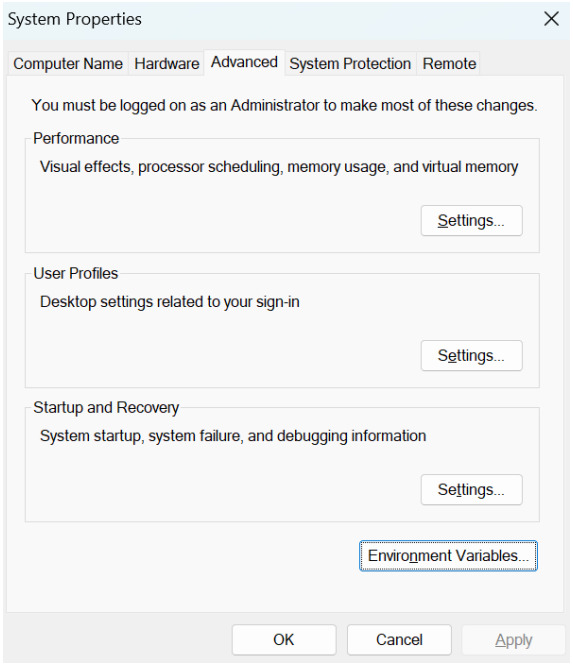
- Select the user name from the drop down and enter the details like certificate name and its expiry date.
- It will then provide you with a .pfx certificate file and also display the password to the certificate file.

NOTE: This password will be displayed only once. So you must copy and store it safely to perform the authentication with the CodeSign Secure server.

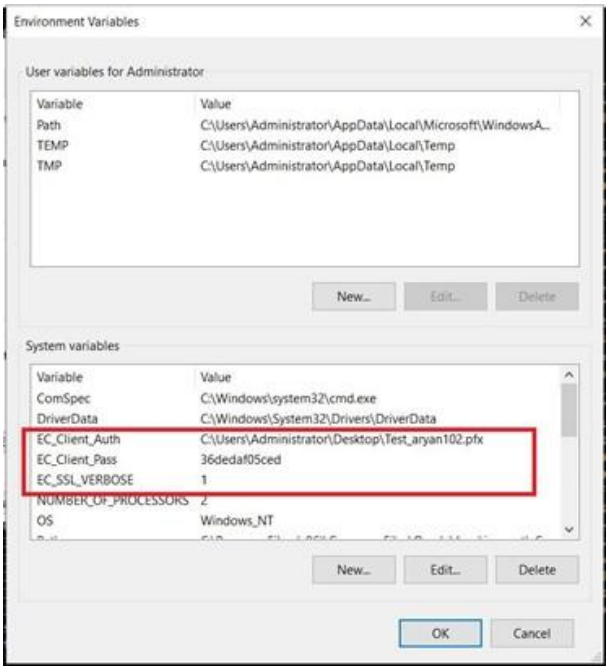


Step 2: Configure the Environment Variables

- Open the Environment Variables from your Start Menu.



- Add new system variables by clicking on the New button. Provide the following variable name and its corresponding details.
  - **EC_Client_Auth:** Corresponds to the path of your SSL Authentication certificate, which can be created from CodeSign Secure.
  - **EC_Client_Pass:** Corresponds to the password of your certificate, which is provided at the time of creation of the certificate.
  - **EC_SSL_VERBOSE:** Corresponds to the setting to either enable (1) or disable (0) the debugging output for EC KSP.



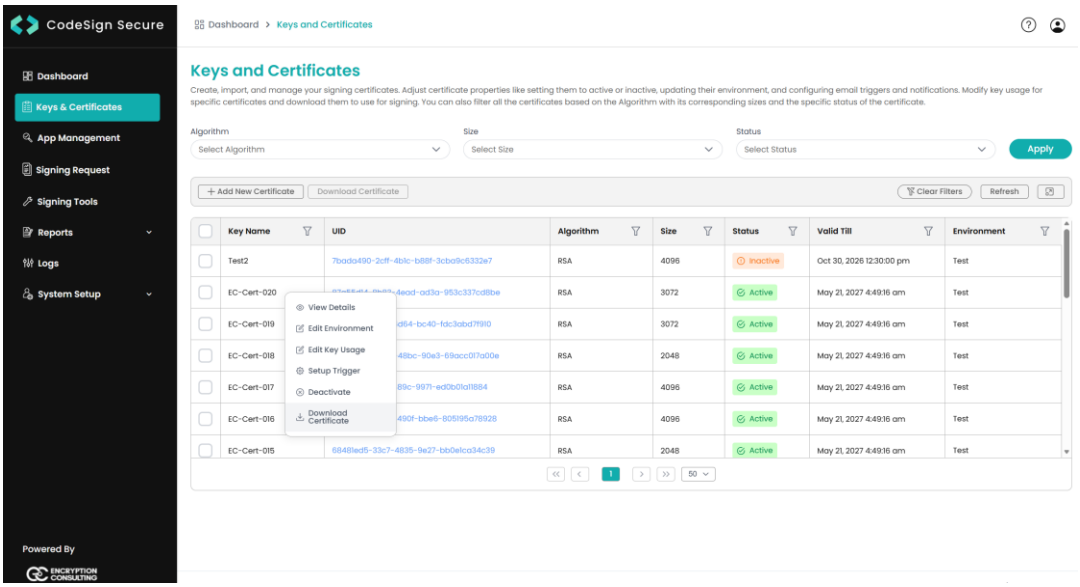
3. ## Export the Public Certificate

Visual Studio needs the public certificate present in the local certificate store before it can be selected for signing.

Steps:

Step 1: Get the Certificate from CodeSign Secure Portal

- Download the required certificate from the CodeSign Secure portal's Keys and Certificate section.



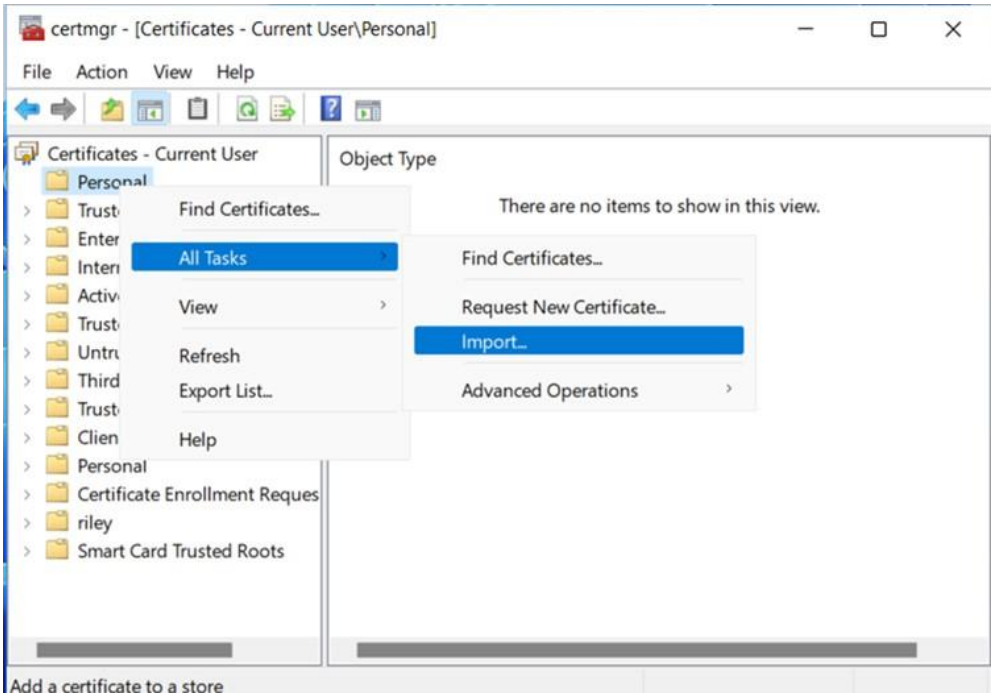
4. ## Import the Certificate into the Windows Certificate Store

Visual Studio selects signing certificates from the current user's personal certificate store, so the exported PEM file must now be imported there.

Steps:

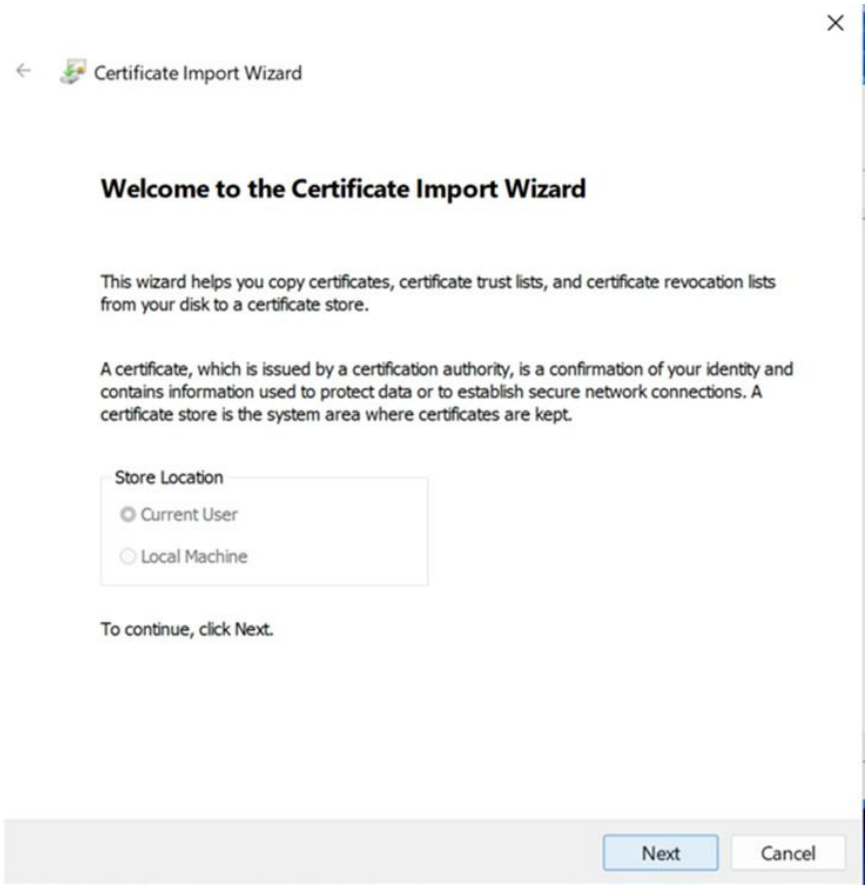
Step 1: Open the Certificate Manager

- Open certmgr.msc and navigate to Personal > Certificates. If there is no Certificates folder, just right click on Personal > All Tasks > Import.



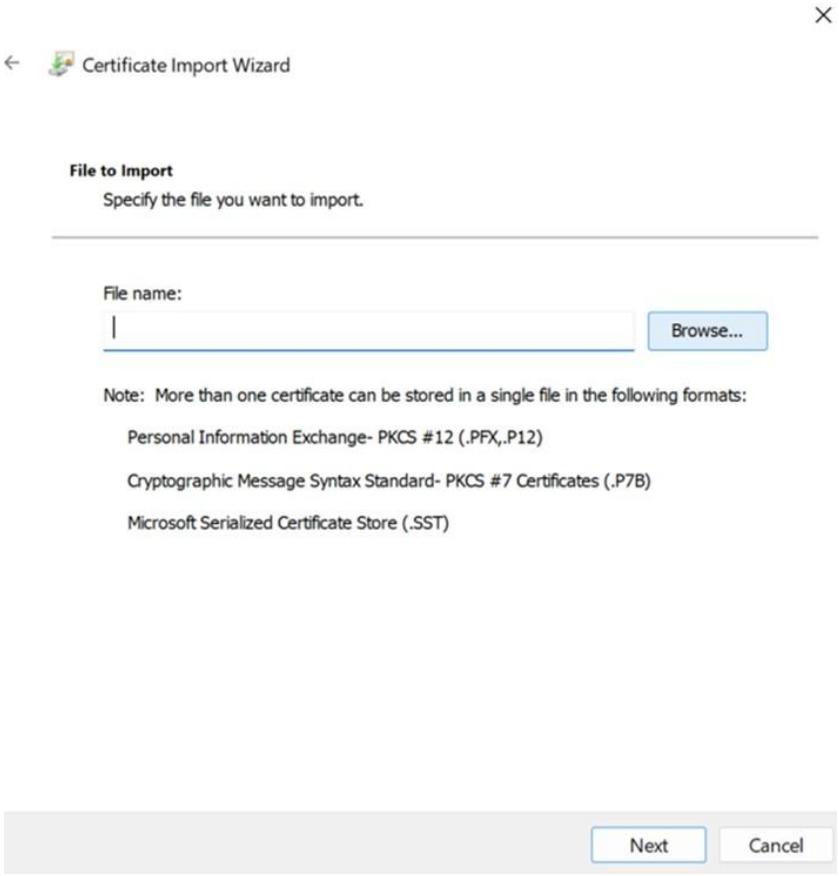
Step 2: Start the Certificate Import Wizard

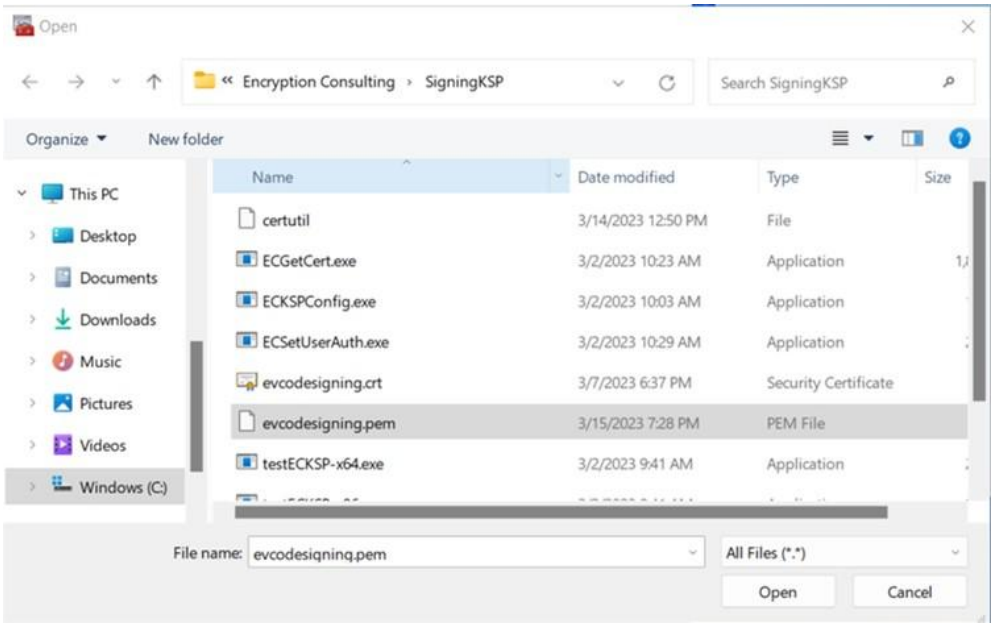
- A Certificate Import Wizard opens. Click on Next. The store location here is by default Current User.



Step 3: Select the Exported Certificate

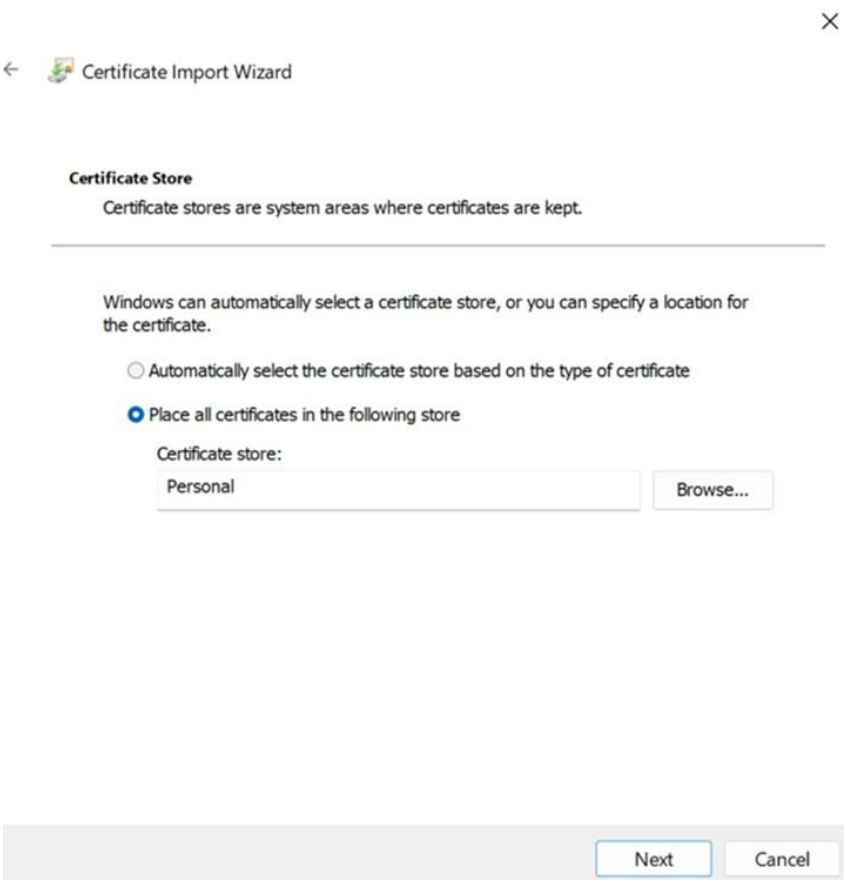
- In the next page, browse for the certificate. From there select evcodesigning.pem (certificatename.pem).
- If you are unable to see the file, select “All Files” at the bottom instead of “X.509 Certificate”. Once the certificate is selected, click Next.



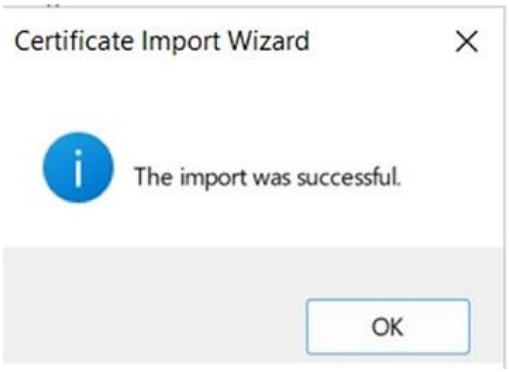


Step 4: Choose the Certificate Store

- On the next page, ensure that “Place all certificates in the following store” is selected, and under that, Certificate store is set to Personal.



- Click on Next and then click on Finish. You will see a dialogue box saying the import was successful.



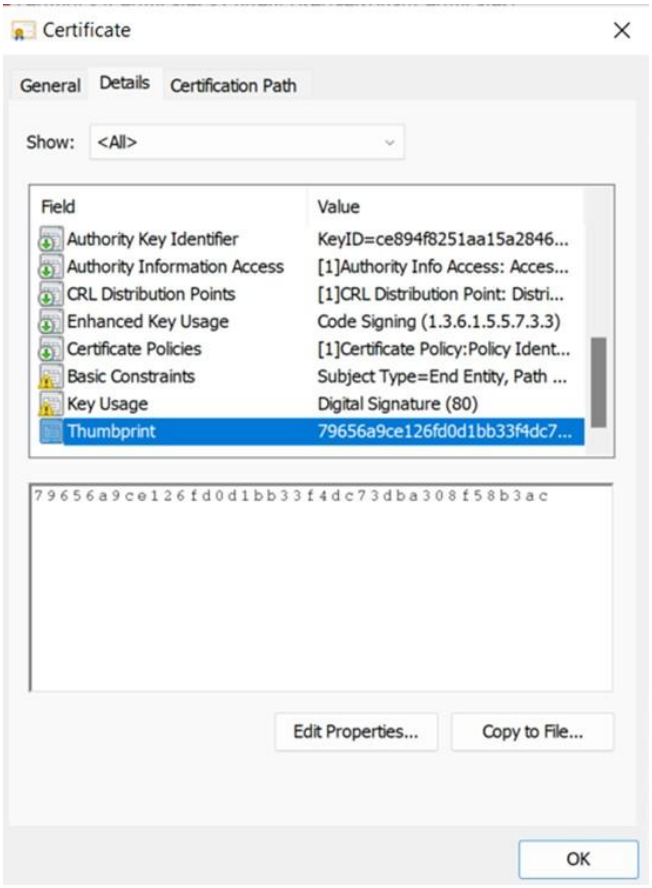
5. **Associate the Certificate with the EC KSP**

   At this point the certificate is in the store but Windows does not yet know which provider holds its private key. The certutil repairstore command binds the imported certificate to the Encryption Consulting Key Storage Provider, so that when Visual Studio signs with this certificate the operation is routed to the HSM.

   Steps:

   Step 1: Copy the Certificate Thumbprint

   - Once the certificate import is done, you need the thumbprint value of your certificate. Click on Personal > Certificates and then the imported certificate.
   - Navigate to the “Details” tab and scroll down to Thumbprint. You can copy the value.



NOTE: Remove any spaces from the copied thumbprint value before using it in the command below.

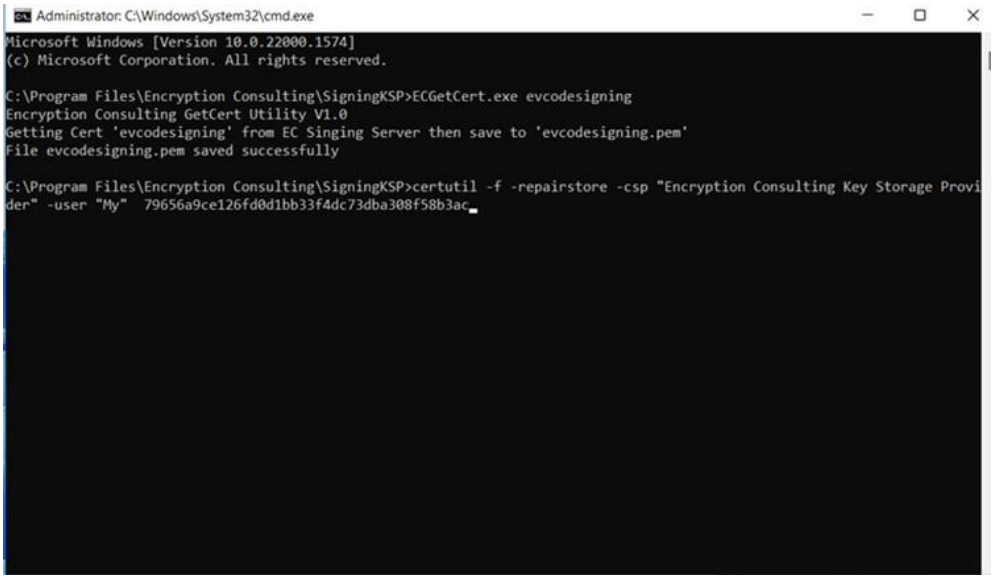
Step 2: Run the Certutil Repairstore Command

- Return to the command prompt. Run the following command, ensuring that you place the thumbprint of your certificate in your command.

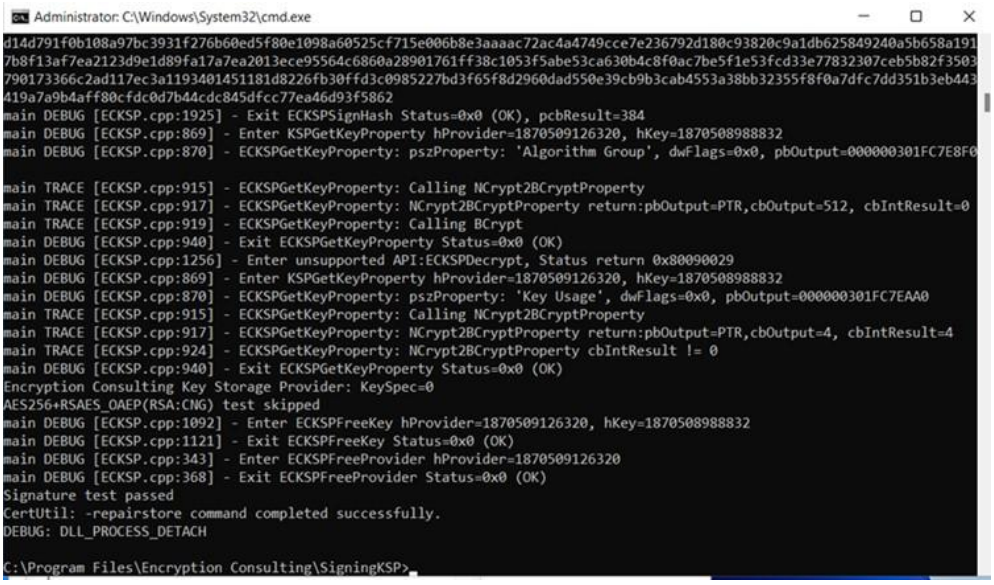
```
certutil -f -repairstore
  -csp "Encryption Consulting Key Storage Provider"
  -user "My" <thumbprint of your certificate>
```

- An example command is as below:

```
certutil -f -repairstore
  -csp "Encryption Consulting Key Storage Provider"
  -user "My" 79656a9ce126fd0d1bb33f4dc73dba308f58b3ac
```



- On success, certutil confirms that the certificate has been repaired and is now associated with the Encryption Consulting Key Storage Provider.



6. Publish the Project with ClickOnce in Visual Studio

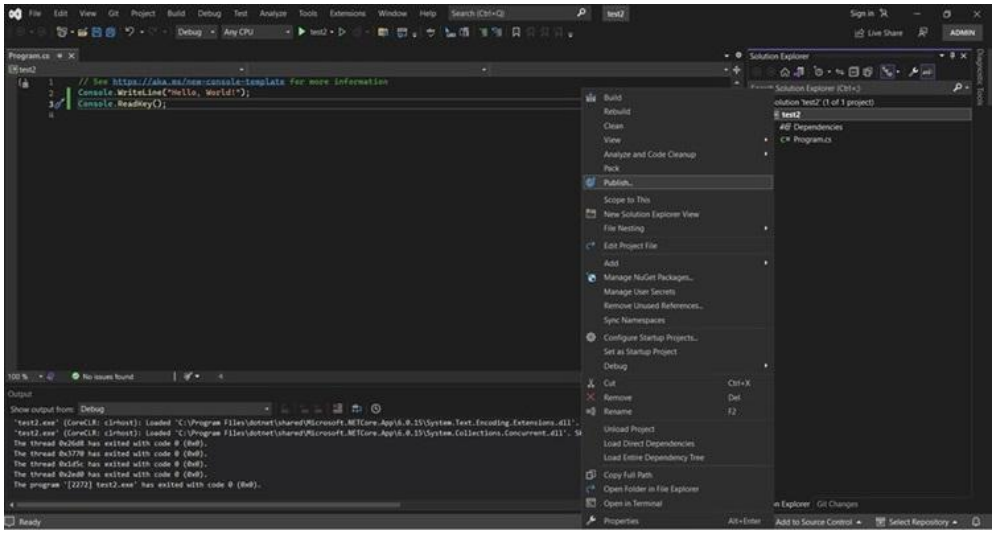
With the certificate in place and bound to the KSP, the remaining work is done entirely inside Visual Studio's publish wizard.

Steps:

Step 1: Open the Publish Wizard

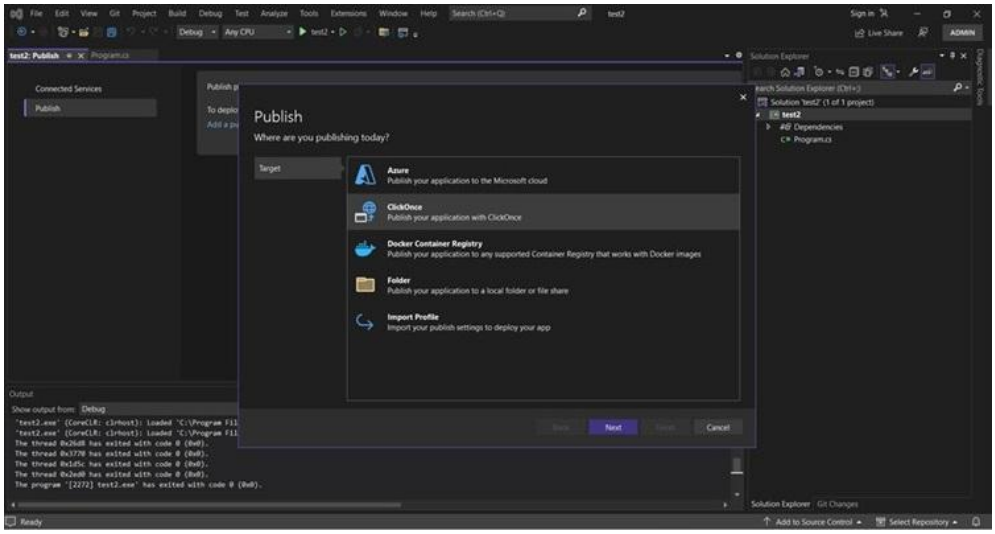
- Once the command runs successfully, navigate to the project in Visual Studio that you want to publish with ClickOnce.

- In the Solution Explorer, right click on your project and navigate to Publish. Click on it.



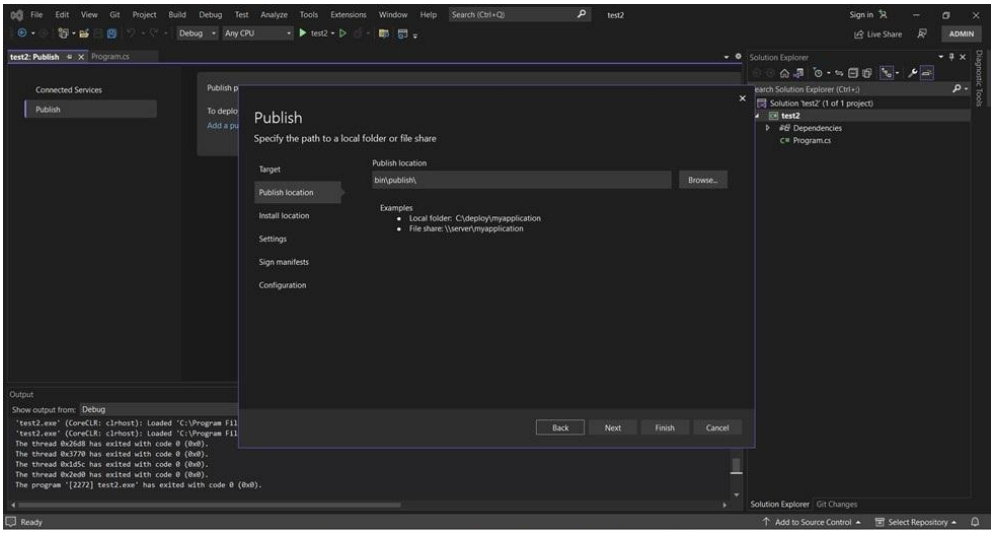
Step 2: Select the ClickOnce Target

- A new window opens. Select ClickOnce and click on Next.



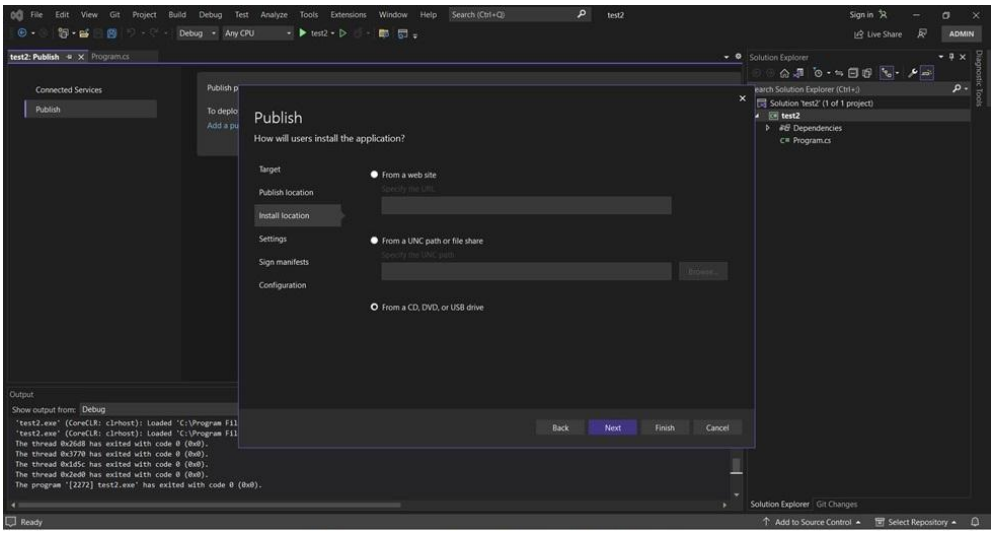
Step 3: Choose the Publish Location

- In the next page, you can choose a publish location or leave the default bin\publish and click on Next.



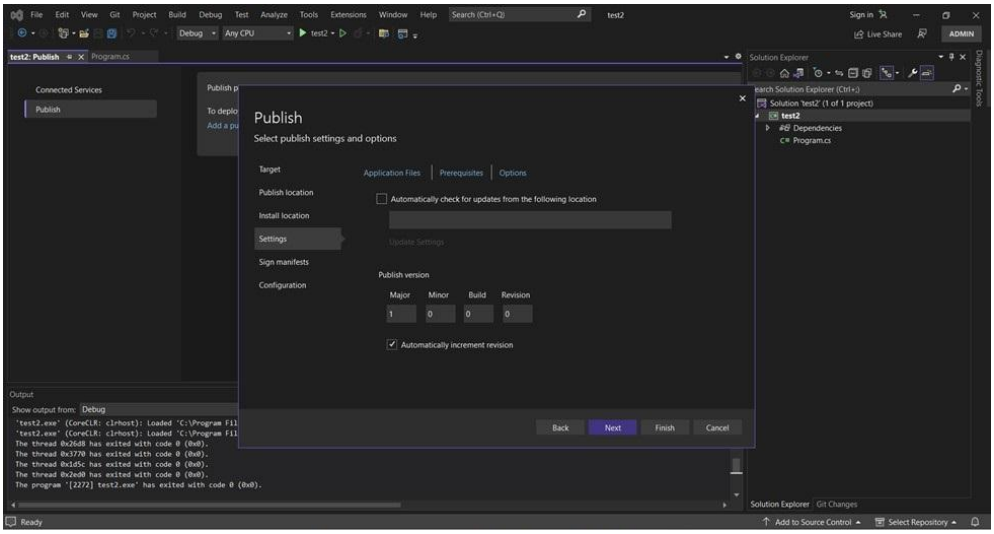
Step 4: Choose the Install Location

- You can choose the Install Location as per your choice or leave the default. Click on Next.



Step 5: Review the Publish Settings

- You can select your settings in the next tab as you like and click Next.



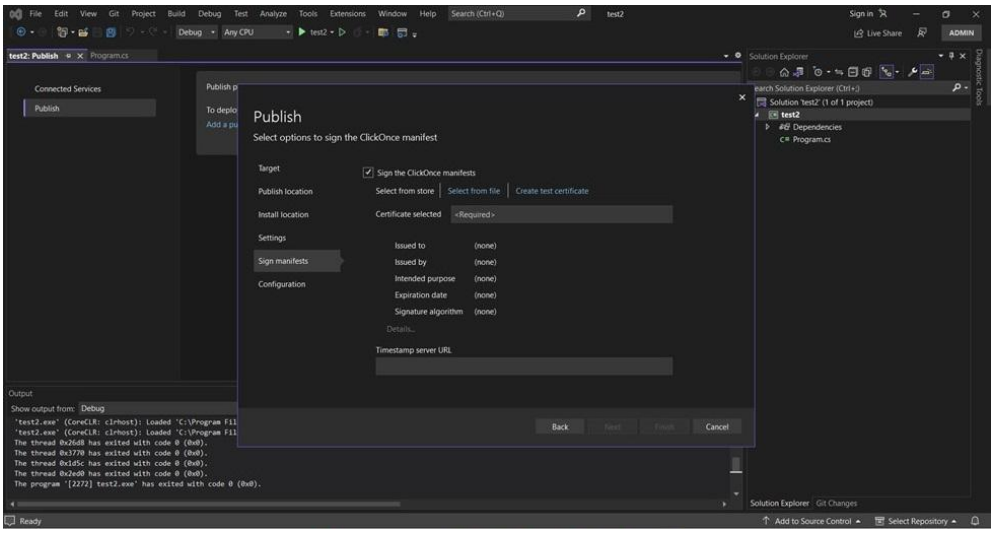
7. **Sign the ClickOnce Manifests**

This is the step where the signature is applied. Visual Studio reads the certificate from the personal store and, because of the association created in Section 5, the private key operation is performed inside the HSM through the Encryption Consulting KSP.

Steps:

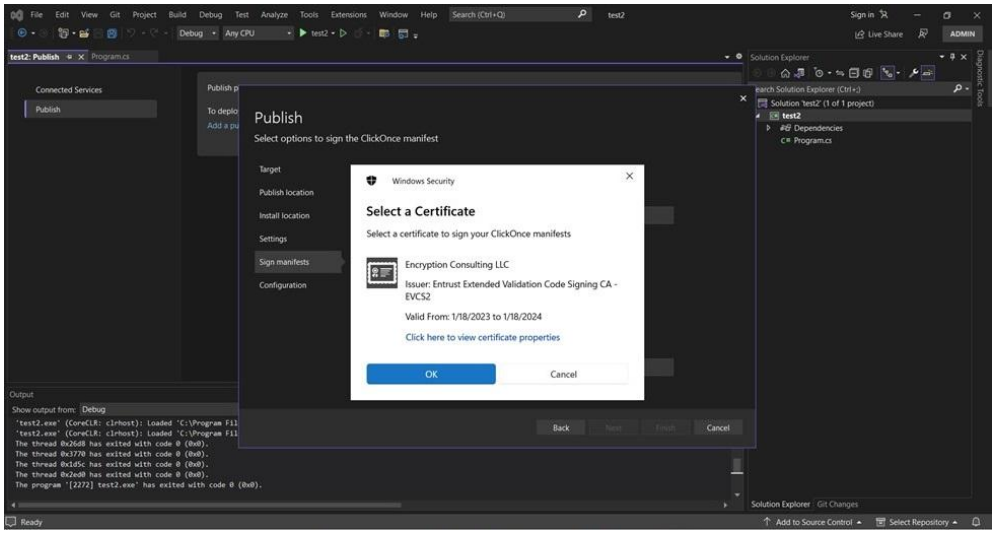
Step 1: Enable Manifest Signing

- In Sign manifests, check the box “Sign the ClickOnce manifests” and click on “Select certificate from store”.

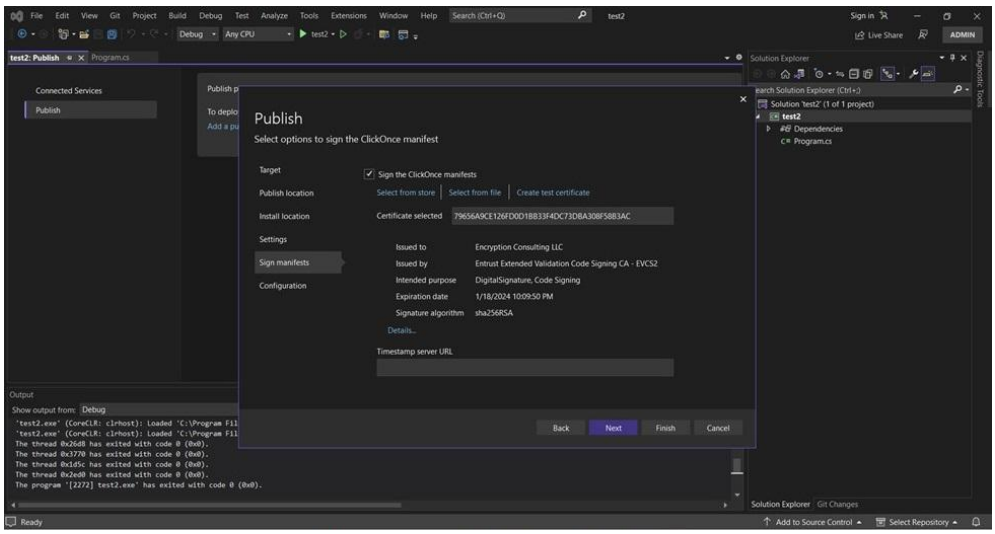


Step 2: Select the Certificate

- A dialogue box opens with the certificate which was initially imported. Click OK to proceed.



- You can now see the certificate details in Sign manifests.



NOTE: If the certificate does not appear in this dialogue, it was either imported into the wrong store or the certutil repairstore command in Section 5 did not complete successfully. Revisit those sections before continuing.

8. Complete the Publish and Verify

The final pages of the wizard create the publish profile and produce the signed ClickOnce deployment.

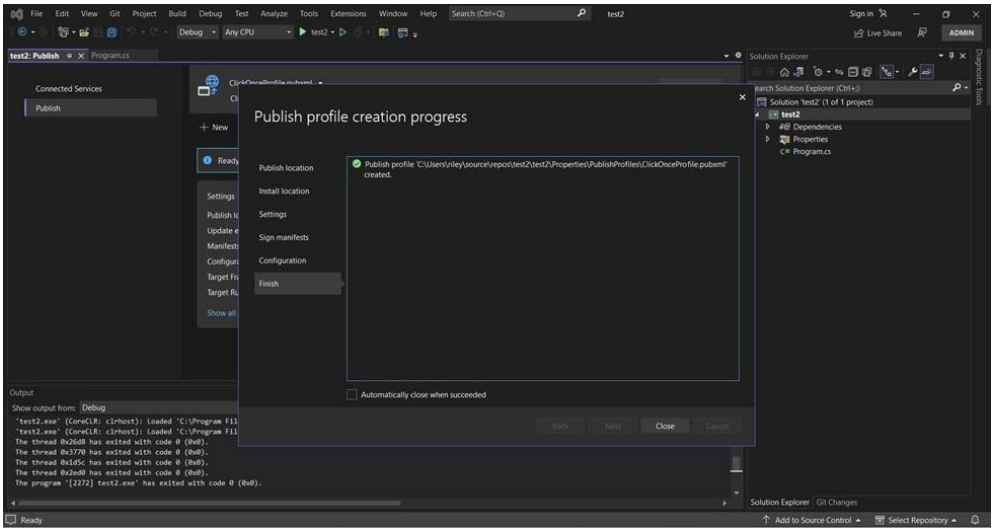
Steps:

Step 1: Finish the Wizard

- Click on Next to choose your configuration and click on Finish.

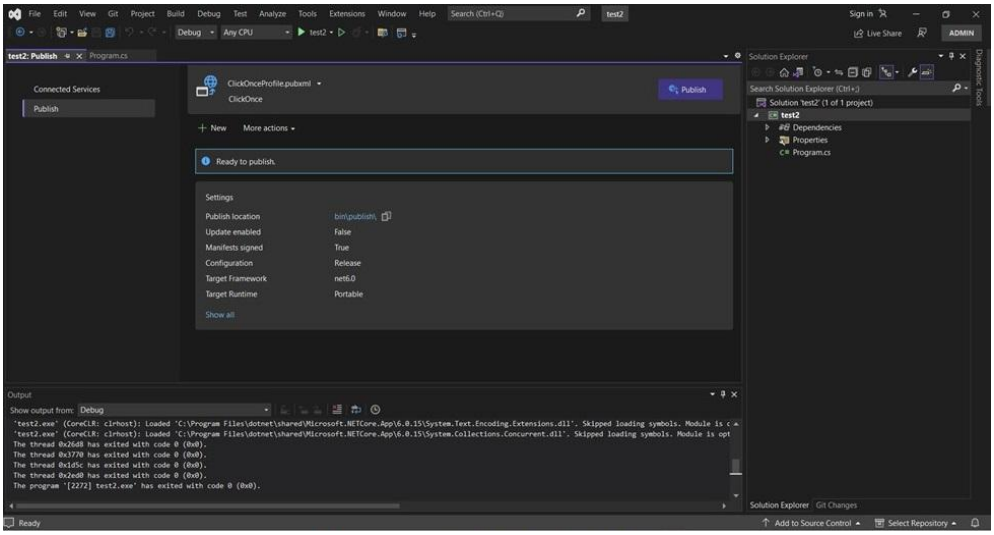
Step 2: Watch the Publish Profile Creation

- You will see the Publish profile creation progress and a green tick when successful.



Step 3: Confirm the Publish Profile

- You can see the Publish Profile created.



- We have successfully signed ClickOnce manifests with Visual Studio.

Step 4: Cross-Check in CodeSign Secure

- Open the CodeSign Secure portal and navigate to Reports > Signing Request Report, where every signing request performed through the KSP is recorded for audit purposes. Confirm that a signing request appears for the certificate you used.
- You can additionally inspect the generated .application and .manifest files in your publish directory, or right click the published application and review the Digital Signatures tab of its properties.