Windows® 10 IoT Core Retail Build with Code Signing Certificate

By Sean D. Liming and John R. Malin
Annabooks – www.annabooks.com

March, 2019

Windows 10 IoT Core Build 1809 Version 17763

Microsoft has made it easy to get started with Windows 10 IoT Core by offering free test release versions for popular platforms in the Raspberry Pi 2/3, MinnowBoard Turbot/MAX. QUALCOMM Dragon Board, and NXP i.MX6/i.MX7. These images are Insider builds that will be updated periodically to the latest release as they become available. These images include most of the features available in Windows 10 IoT Core, which provides a solid foundation for initial application development. Once you have proof of concept, the next step is to pick a production platform and create a custom image. Creating a custom image requires downloading the necessary kits: Windows Assessment and Deployment Kit (Windows ADK), Windows Driver Kit (WDK), Windows Software Development Kit (SDK), the IoT Core ADK add-on-kit, the IoT Core Packages, and a BSP for the hardware platform. There are two image-build-types: test and retail. Both allow you to build customized images. The test builds include extra development tools and are used for continuing application and system development. The test image is signed with Microsoft test certificates.

Once the application, drivers, and hardware have been developed, the final step is to create a shippable image, and this is where the retail build comes in. The retail builds remove the development tools from the image and all the CAB files must be crossed-signed with a company code signing certificate. The online documentation provides a high-level process to build a retail image, but the actual steps to get a code signing certificate and setting the cross-signing certificate is not 100% clear. This article looks to provide the steps in setting up the IoT Core image build system with the code-signing and cross-signing certificates.

Note: This information is intended for those working with Windows 10 IoT Core Build 1809 Version 17763, which is a Long Term Service Channel (LTSC) release You should already be familiar with the IoT Core ADK Add-on kit build process for 17763.

Code Certificate Purchase and Cross-Sign Certificate Setup

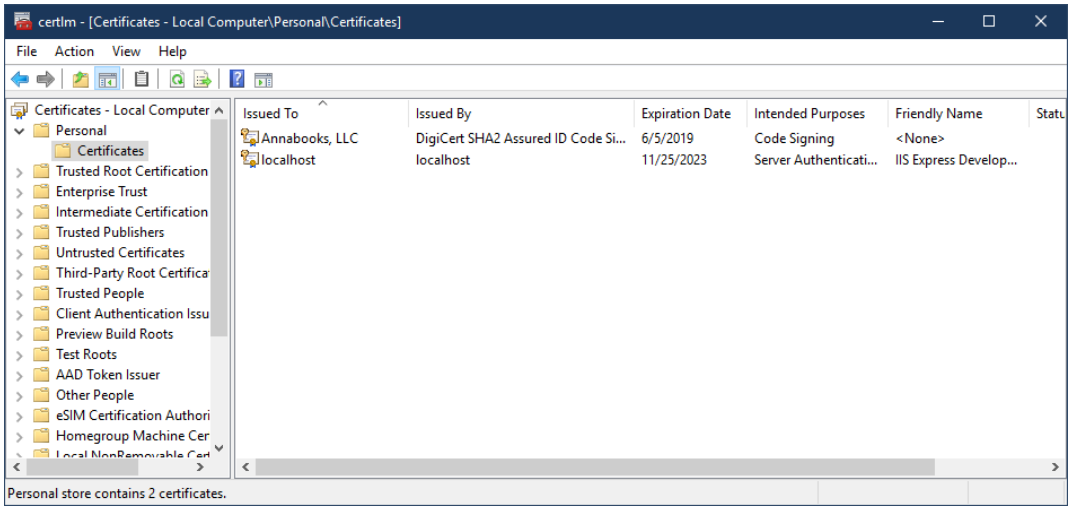
There are two certificates involved in the signing process. The code signing certificate is purchased by your company. The cross-signing certificate is referenced during the build process to validate the code signing certificate and sign the CAB files that make up the image.

1. First you need to purchase a code signing certificate from a certificate authority. There is a list of certificate providers available here:

https://docs.microsoft.com/en-us/windows-hardware/drivers/dashboard/get-a-code-signing-certificate

Note: The choice of provider is up to you and your company. We show a specific example only for clarity and not an endorsement of one provider over another.

There are two code signing certificate types: standard and extended validation (EV). You need to choose the one that best fits your needs. For example, the EV certificate is good for those selling or updating applications via the Windows Store. In most cases, the standard certificate is fine for building a retail IoT Core image. Once you purchase a company code signing certificate, you must install it on the development machine that is building the custom IoT Core image. Best practice is to install the certificate for both the local system and the current user. You can then view the certificate in MMC with Certificate snap-in (or certificate manager). Here is an example of the Annabooks certificate that has been installed:



The code signing certificate can be exported and put into another development machine if you have others working on applications or images.

2. Using the information in the MMC window, locate and download the cross-certificate from the webpage:

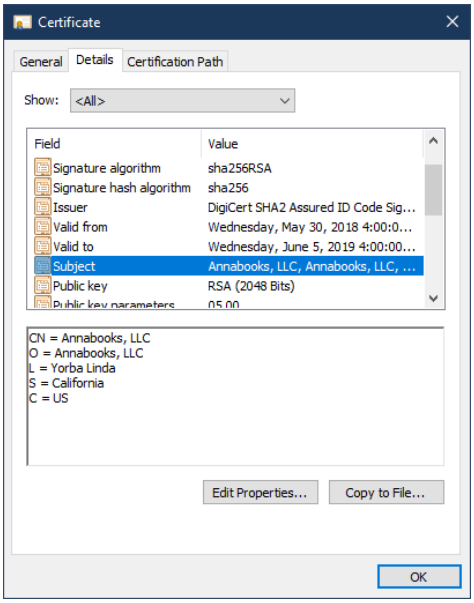
https://docs.microsoft.com/en-us/windows-hardware/drivers/install/cross-certificates-for-kernel-mode-code-signing

From the Annabooks certificate information above, the DigiCert Assured ID Root CA is the match.

3. Unzip the crt file, and place the file in the workspace: c:\<workspace>\Certs, where workspace is the folder location generated by the IoT Core ADK Add-on kit for your project.
4. Edit the IoTWorkspace.xml file found in c:\<workspace>
    a. In the <RetailSignToolParam / > tag, put in the string that calls the code signing certificate. Here is the format:

```
/s my /i "Issuer" /n "Subject" /ac "C:\<path>\CrossCertRoot.cer" /fd SHA256
```

The “issuer” and the “subject” come from the code signing certificate. If you open (double click) the certificate from the certificate manager window, and then click on the “Details” tab, you will see the necessary information for Issue and Subject.



Here is an example using the Annabooks certificate information with the MBM1 workspace:

```
<RetailSignToolParam>/s my /i "DigiCert SHA2 Assured ID Code Signing CA"
/n "Annabooks, LLC" /ac "C:\MBM1\Certs\DigiCert Assured ID Root CA.crt"
/fd SHA256</RetailSignToolParam>
```

The build process matches the cross-certificate with the install code signing certificate to sign the CAB files. The above line in the XML file provides the link between the two certificates for the build process to follow.

Note: The cross-signing certificate doesn't get installed into the system.

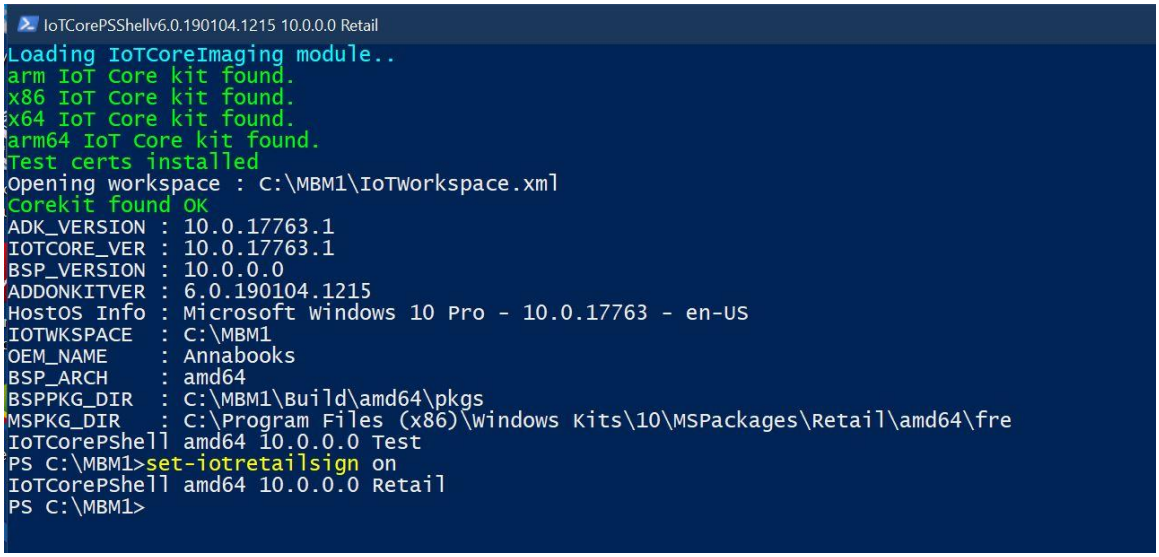
Building the Retail Image

With the certificates set up, the steps to create a retail image are:

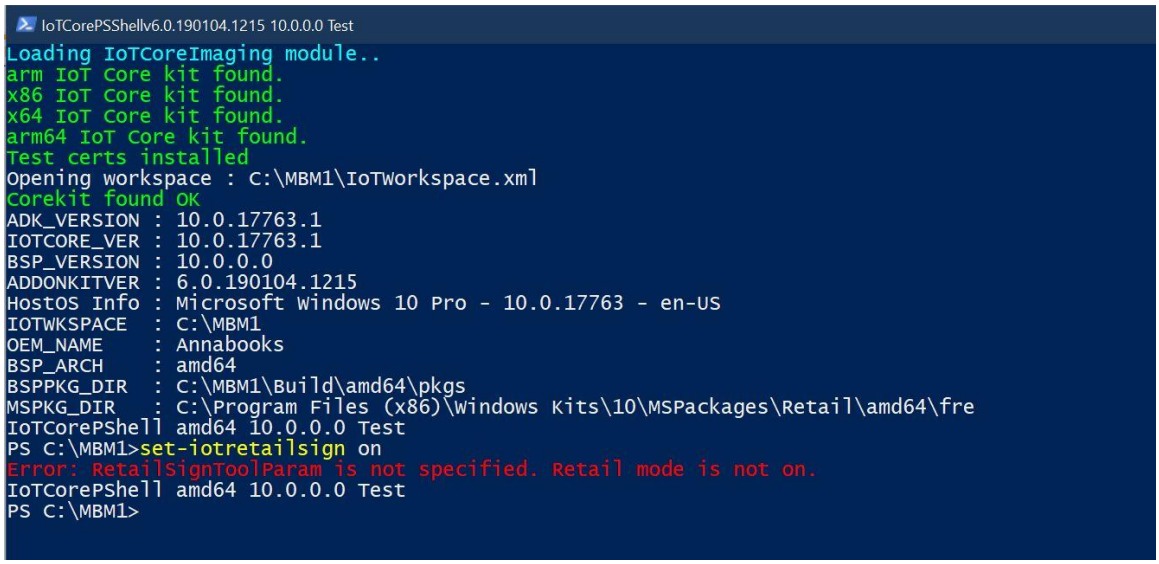
1. If you have been building test builds, delete the c:\<project>\Workspace\Build\<cpu> folder. This will clean out any CAB files signed with test certificates.
2. Run the IoTCorePShell.cmd, this will open a PowerShell window with Administrator privileges.
3. Within the IoTWorkspace.xml file, update with the cross-certificate information. Run the following command to enable signing:

    Set-IoTRetailSign On

Note: You will see a banner change from test to retail



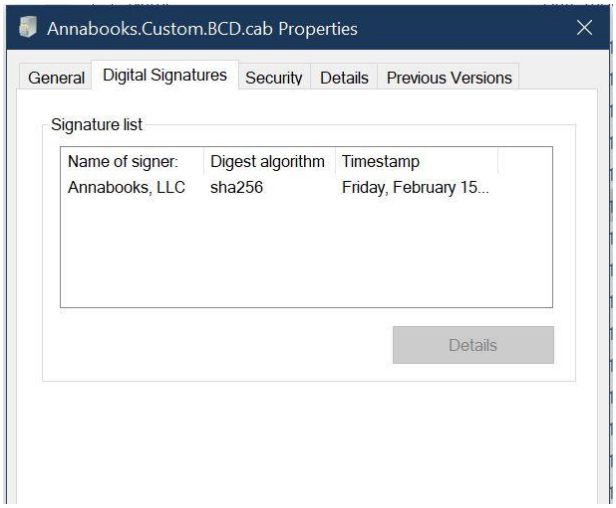
If you forget to put the cross-certificate information in the IoTWorkspace.xml file, you will get an error:



4. Run the following command to build the cab files and sign with the certificate:

   New-IoTCabPackage All

   If you check the properties of the CAB files, you will see that the Digital Signature has been applied:



5. If a vendor supplied CAB files for a BSP or applications, these CAB files will have to be re-signed with your certificate using the Redo-IoTCabSignatures command.
    a. Create a folder on the root of c:\cabs
    b. Run the following command to re-sign the cab files:

        Redo-IoTCabSignatures c:\<location of cab files> c:\cabs

Note: You will have to replace the original CAB files with the newly signed CAB files.

6. Once the cab packages have been generated, we can build the retail image, run the following command:

        New-IoTFFUImage myproject retail

The resulting image is ready for production. From insider builds, to custom test image, to final retail release, Windows IoT Core provides a path to go from concept to production. We would like to thank Microsoft Core OS team for helping to clarify the cross-signing string information.