I Wish I Knew How To …

Build an AppX Installer

For Microsoft Store with Xojo

December 2016 Edition (1.5)

By Eugene Dakin

Contributors: Michel Bujardet and David Cox

Table of Contents

Converting a 32-bit Xojo Desktop App to a Universal Windows App

The Windows 10 Anniversary Update (Version 1607) is making it possible for Xojo programmers to convert their Windows Desktop program to a UWP (Universal Windows Platform) that can be placed and sold in Windows Store. If you have questions, feel free to visit Xojo forums and ask questions!

Before we start the journey to create this installer, set aside a day (yes, this takes significant patience) to go through the steps to create an InnoSetup installer, and then build the AppX installer.

Assumptions for this article are that you know how to program Xojo and create a simple installer package (InnoSetup or other).

There are a few steps to setup the computer to be able to perform the conversion, and the rough explanation is that the Xojo executable file is placed in the Windows 10 Converter and an .appx package is created. Once the application is in an appx form, then it can be installed on your computer or uploaded to the Windows Store to be sold. The Windows store does not accept an exe from Xojo to be sold within the store, the application must be in an appx package.

Many thanks to Michel Bujardet and David Cox who have programs in the store and have shared their experience with the installation process and have programs in the Windows Store.

Professional Windows 10 or the Enterprise version of the Operating System are able to run the appx programs.

The condensed steps to submit a Xojo 32-bit app into the Windows Store are:

1) Register as an app developer
2) Check the Windows OS version, and it should be greater than 14393.10
3) Install the UWP DesktopAppConverter (DAC)
4) Install the appropriate Base Image File
5) Install Visual C++ runtime for desktop
6) Install the Windows SDK
7) Build the Xojo app with Innosetup
8) Built the Appx program with DesktopAppConverter
9) Install the certificate
10) Install the appx program – (This creates a local AppX installer)
11) Update the icons and update the installer
12) Submit to Store
    a. Have a website page explaining your program (like a pamphlet)
    b. Windows Store Privacy statement on website

Register as an app developer

Whether you are selling applications or are giving them away for free, you will need a Microsoft developer account which can be created at the following link.

https://developer.microsoft.com/en-us/store/register

This provides you with an account with Microsoft, and as of Nov 2016 this is a one-time fee of $20.00 U.S.D.

There are a few steps to confirm your identity and don't forget to fill in the part where Microsoft will send the funds for your sold program. This step may be as fast as an hour or can take a week.

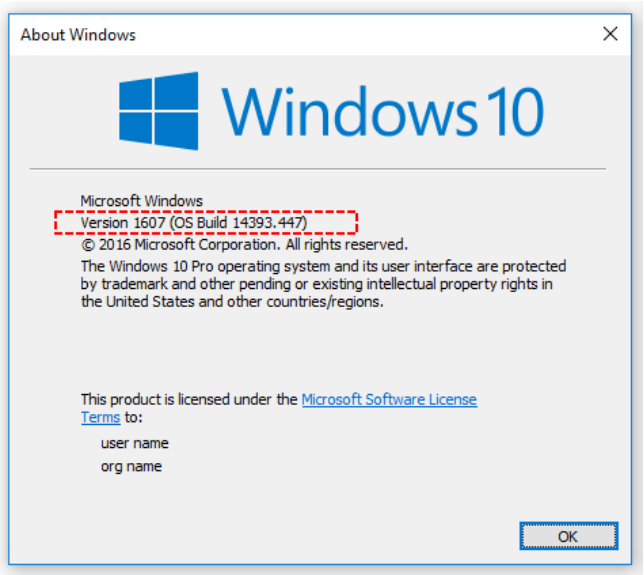
Windows 10 Anniversary Update

To run appx programs, the operating system must be running Windows 10 Anniversary update that has been launched on August 2nd 2016. If your computer has been automatically updating Windows 10 then it may already be installed.

To start the Windows Version program:

Press Win+R keys
Type winver
Press Enter

Figure 1. Winver Program



If the version is 1607 or higher, or the OS Build is 14393.10 or higher, then Windows 10 Anniversary Update is already installed.

Install UWP DesktopAppConverter (DAC)

Start the Windows Store program by typing Store in the search bar in the bottom left corner of the screen and type ***Desktop App Converter*** in the search bar, and download it for free. Here is the description information for the program:

Figure 2. UWP DesktopAppConverter Store Description

Desktop App Converter requires you to obtain a Windows 10 base image (.wim) from the Microsoft Download Center. Please visit https://aka.ms/converterimages to obtain the base image matching your host operating system.

Desktop App Converter is a tool that enables you to bring your existing desktop apps written for .NET 4.6.1 or Win32 to the Universal Windows Platform (UWP). You can run your desktop installers through the converter in an unattended (silent) mode and obtain an AppX package that you can sideload on your machine or upload to the Windows Store.

This developer tool is required to be run under administrative privileges.

Note: Each build of Windows must have a matching base image for the Desktop App Converter. With the file size being about 3.3 GB, you may want to dedicate one PC or a Virtual Machine for building the final product.

Press the ***Get*** button to download the free program.

When the DesktopAppConverter program is launched, a PowerShell Window (Command Prompt) box appears and shows the following information.

Figure 3. DesktopAppConverter Launch Information

```
SYNOPSIS
   DesktopAppConverter is a tool to create a Windows 10 Store-ready APPX package from an
application's installer that can run in silent mode.

USAGE
   DesktopAppConverter.exe <arguments>
   Use "DesktopAppConverter.exe -?" or visit http://go.microsoft.com/fwlink/?LinkID=785434 for
more information on arguments supported by this tool.

EXAMPLES
   1. Setup converter environment using a base image
     C:\> DesktopAppConverter.exe -Setup -BaseImage C:\BaseImage\BaseImage-<version>.wim
```

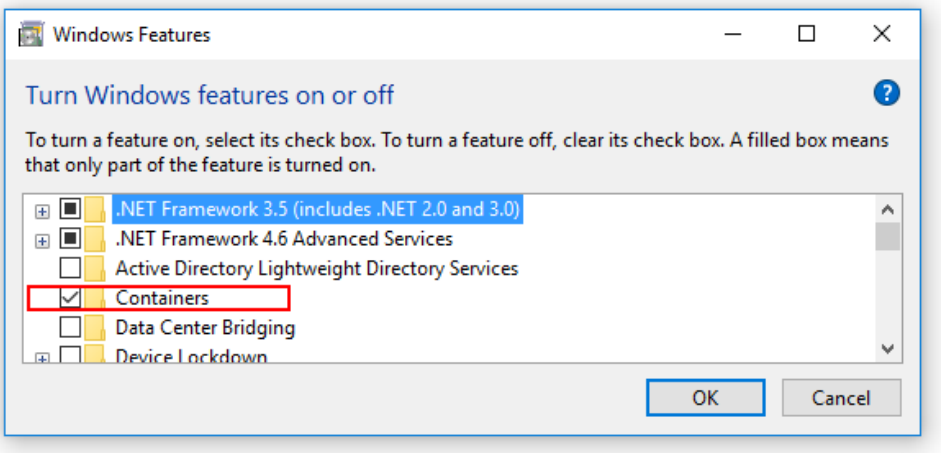
2. Convert an installer for a desktop app (MSI or EXE) to a UWP package (AppX)
C:\> DesktopAppConverter.exe -Installer C:\MySetup.MSI -Destination C:\MyAppxFolder -PackageName "MyApp" -Publisher "CN=<publisher_name>" -Version 1.0.0.0 -MakeAppx

This program requires administrative privileges to run. Please right click on the application tile and choose "Run as administrator" or run this program from an elevated cmd or powershell window.

- -Installer is the filename and path of our InnoSetup file that will be converted
- -Destination is where the built output files will be placed
- -PackageName will be the name of the installer package
- -Publisher is usually your name, because you are creating the program and installer
- -Version will be the version of the application
- -MakeAppX will create the folder with the files needed, such as the assets, manifest, etc. The AppX package will also be created
- -Verbose lets you know many of the steps that are being performed while the DesktopAppConverter program is executing
- -Sign will generate the certificates to sign the AppX package to load on your local computer without needing to upload it in the store

When the installer application is in MSI or EXE form, then it automatically supports the silent installation process.

This will take a while and the machine may need to be rebooted to install a feature called containers that was added by installing the base image in Step 1.

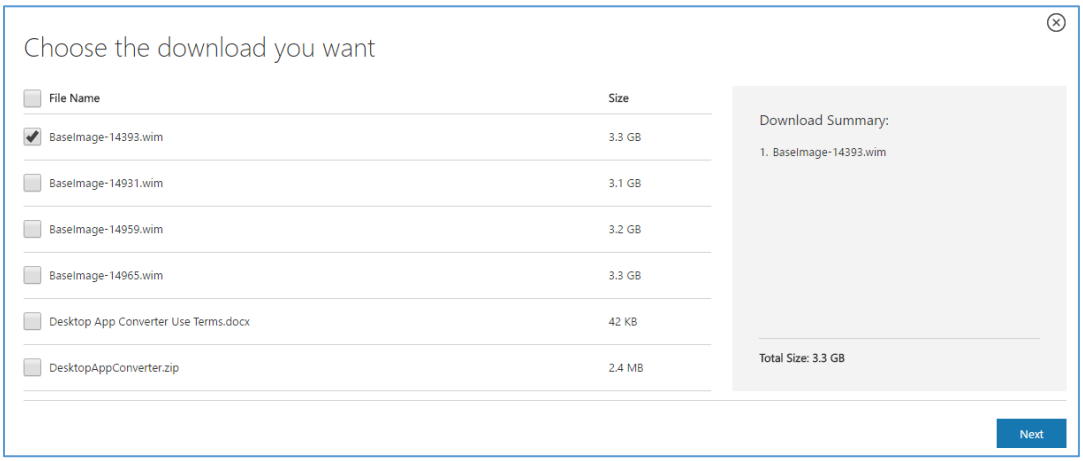


Containers is only available on Windows 10 Pro or Enterprise.

Desktop App Converter Base Images

Before the Windows 10 Base Image (.WIM) file is to be installed, it is important to know which version of Windows 10 is running. In the ***Windows 10 Anniversary Update*** section of this article is the command to retrieve this information. My machines version is Version 1607 (OS Build 14393.447).

Click on the following link to download the Desktop App Converter Base Image:
https://www.microsoft.com/en-us/download/details.aspx?id=54245



The files are very large, and the BaseImage-14393.wim is 3.3 Gigabytes. Double check your internet speed and hard-drive space before downloading.

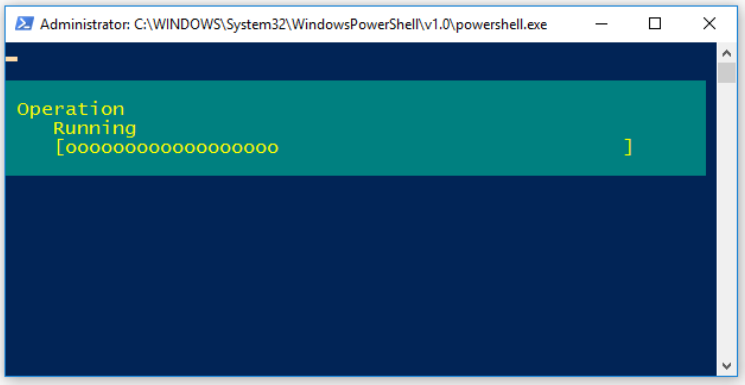
Once the file has been downloaded, then make a new folder to store the wim file. I created a new directory path at C:\MyApps\ and placed the wim file there.

Type PowerShell and run this program as an administrator.

Type the following commands:

1) Set-ExecutionPolicy bypass (and then press y afterwards)
2) DesktopAppConverter.exe -Setup -BaseImage BaseImage-14393.wim (it may ask you to restart the computer, and type y for restarting the computer). This may take a while. Once the computer restarts then the powershell will continue to install and there will be a screen like the below screen grab.

Figure 4. Installing BaseImage



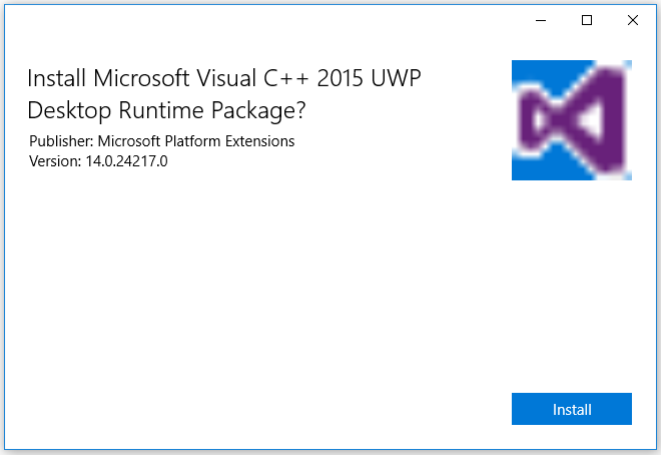
This may take a little while to install.

Install Visual C++ Runtime for Desktop Bridge

The Desktop App Converter tool is required because the InnoSetup installer exe package has added the necessary dll and setup files for our Xojo program. Theoretically, any installer msi or exe package should work and the InnoSetup installer is used for this example.

With Xojo updated for Windows, programs that use the desktop bridge will need have the Visual C++ dependencies downloaded. A Microsoft blog article describes this a little more. In my case, I will download the ***VC 14.0 framework packages for Desktop Bridge***. The link to download this package is located at: https://www.microsoft.com/en-us/download/details.aspx?id=53175 and has a file size of about 43 MB. The downloaded filename is vc_uwpdesktop.140.exe. Double-click the file to install it. Once the installation is complete, then these Visual C++ Runtime framework files will be located at: C:\Program Files (x86)\Microsoft SDKs\Windows Kits\10\ExtensionSDKs\Microsoft.VCLibs.Desktop\14.0\Appx\Retail\x86. There should be a file called Microsoft.VCLibs.x86.14.00.Desktop.appx, and double-click it to install it.

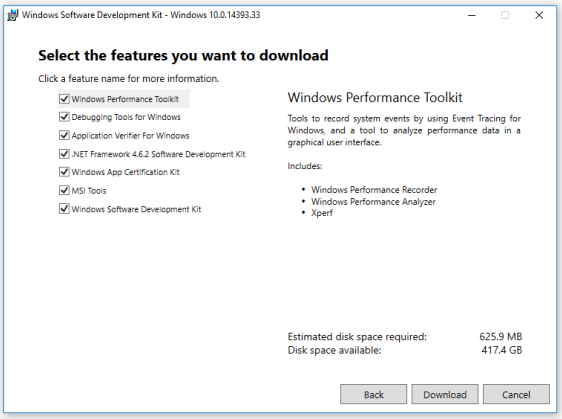
Figure 5. Install UWP Runtime Package



Install the Windows 10 SDK

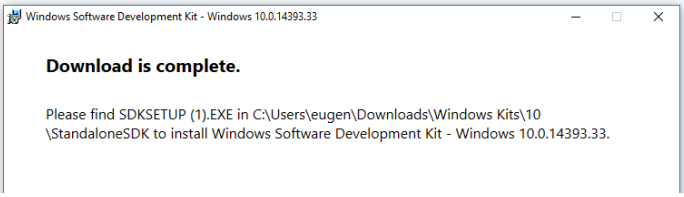
The next step is to install the Windows 10 Software Development Kit. Download the program at http://go.microsoft.com/fwlink/?LinkId=821375 and double click the program to run it. Install ALL of the 8 packages:

Figure 6. Install Windows 10 SDK Options



Once the file has been downloaded, you should receive a screen with this information.

Figure 7. Downloaded File



Run the sdksetup (1).exe file. It will be installed in the C:\Program Files (x86)\Windows Kits\10\ directory.

Building the Xojo Application

Type PowerShell in the search bar, and right-mouse-click on the icon and select ***Run as Administrator***.

Change the active folder within the powershell by typing:

- cd\
- cd MyApps

With InnoSetup containing all of the files in the SetupDemoAppx86.exe file, and copy this file to the C:\MyApps folder.

Code 1. Type this command in PowerShell (Administrative privileges) to Convert

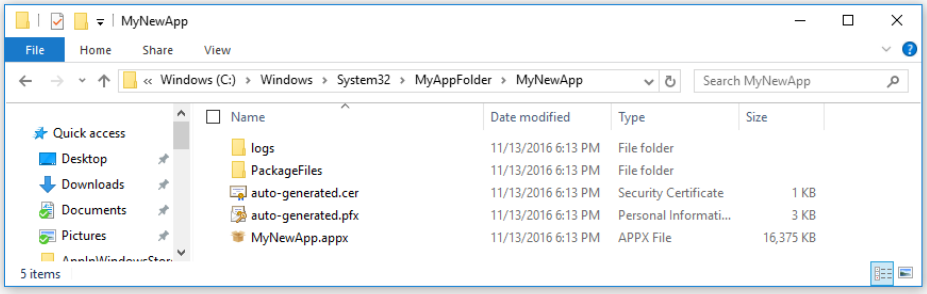
```
DesktopAppConverter.exe -Installer "C:\MyApps\SetupDemoAppx86.exe" -Destination "MyAppFolder" -PackageName "MyNewApp" -Publisher "CN=Eugene Dakin" -Version 1.0.0.0 -MakeAppx -Verbose -Sign -InstallerArguments "/verysilent"
```

- **-Installer** is the filename and path of our InnoSetup file that will be converted
- **-Destination** is where the built output files will be placed
- **-PackageName** will be the name of the installer package
- **-Publisher** is usually your name, because you are creating the program and installer
- **-Version** will be the version of the application
- **-MakeAppX** will create the folder with the files needed, such as the assets, manifest, etc. The AppX package will also be created
- **-Verbose** lets you know many of the steps that are being performed while the DesktopAppConverter program is executing
- **-Sign** will generate the certificates to sign the AppX package to load on your local computer without needing to upload it in the store

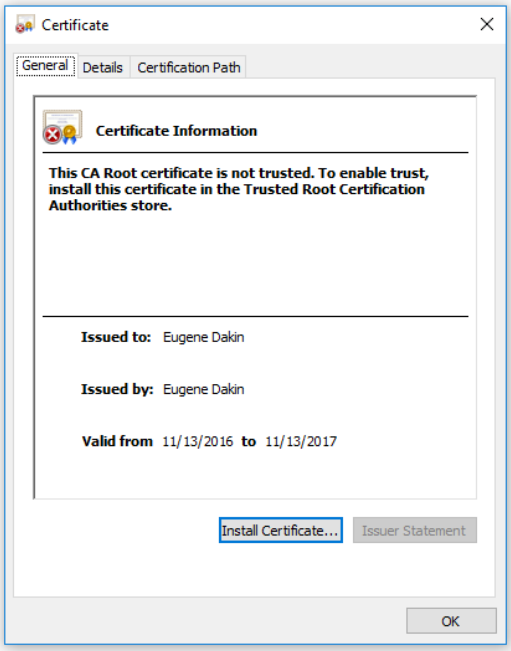
More arguments are available by typing the following line in the PowerShell:

```
DesktopAppConverter -?
```

There were some files that were generated at the location: C:\MyApps\MyAppFolder\MyNewApp

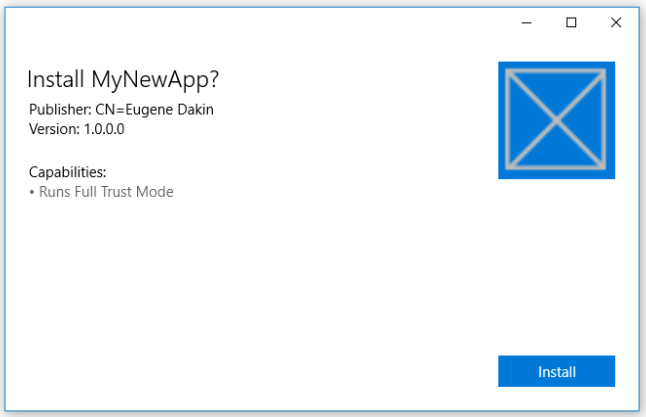


Before double-clicking on MyNewApp.appx, we need to install the trusted certificate. Double-click auto-generated.cer and press ***Install Certificate…***



Select the ***Local Machine*** option, and select ***Place all certificates in the following store***, and press ***Browse***. Next, select ***Trusted People*** and select ***OK***, press ***Next*** and then ***Finish***. If everything went correctly, you should get a messagebox that shows ***The import was successful***.

Double-click on the installer file MyNewApp.appx and the following installer screen should appear.



The installer will allow you to install it on your machine.

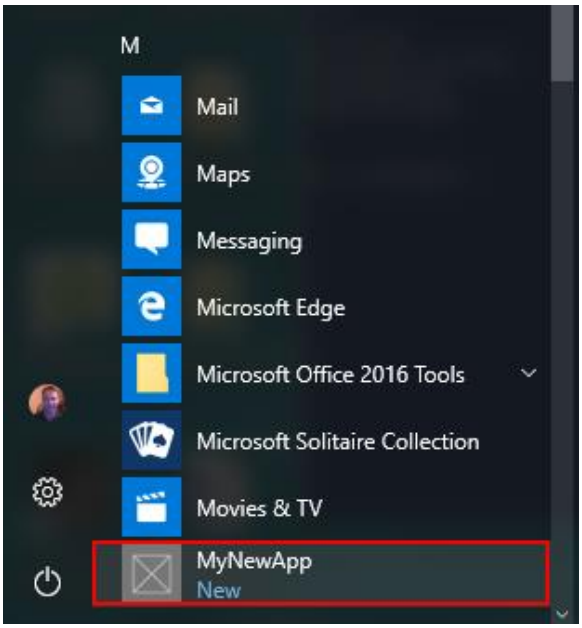
The program can be found by pressing the Windows Start Button and search for MyNewApp.

If you get an error message that has an error with something like this:

Appx Package was blocked because the provided package has the same identity as an already-installed package but the contents are different. Increment the version number of the package to be installed, or remove the old package for every user on the system before installing this package.

To solve this issue, either uninstall the program by going to **Control Panel->Uninstall a Program** and uninstall the program, or increment the number from 1.0.0.0 to another number and change the UUID.

Tada!!! Finished. You have created a basic installer to be installed on your Windows 10 computer. The application appears in the Start menu and if you right-mouse-click on the icon, then there is the ability to uninstall the program.

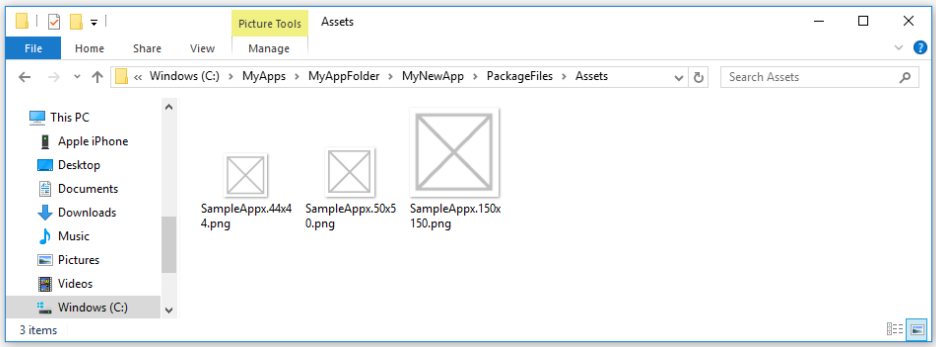


The application has been installed in the **C:\Program Files\WindowsApps** folder which is hidden from a user and administrator.

Important: Data can only be read from the **C:\Program Files\WindowsApps** folder. Data can be written to the **AppData** folder. The AppData folder is located at C:\Users\<username>\AppData\Local.

Adding Icons

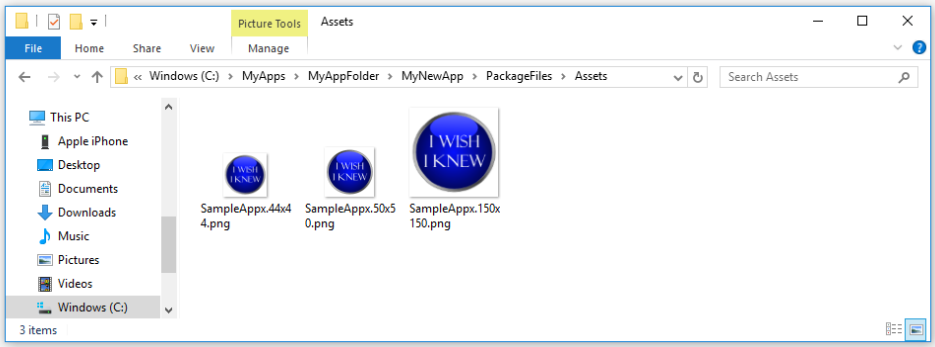
There are three default icons that have been created when the DesktopAppConverter executable file created the MyNewApp.appx installer, and they are located at C:\MyApps\MyAppFolder\MyNewApp\PackageFiles\Assets.



The default names are:

- SampleAppx.150x150.png
- SampleAppx.50x50.png
- SampleAppx.44x44.png

With these three files being PNG formatted pictures, I used Paint dot net and created my own pictures and then copied them in the same directory. Feel free to use your own picture editing software. Each of the files need to be 150x150 pixels, 50x50 pixels, and 44x44 pixels. Windows will use the appropriate size when running your program. Below are these images that were copied over the existing pictures.



Now that these images are in the correct folder, a command prompt (Not a PowerShell window) needs to be opened by typing **cmd** in the search bar.

Go to the directory where the makeappx.exe is located by typing the following commands:

- cd\
- cd Program Files (x86)
- cd Windows Kits
- cd 10
- cd bin
- cd x86 (go to x64 directory for a 64-bit Xojo program)

This should place the cursor in the directory C:\Program Files (x86)\Windows Kits\10\bin\x86\

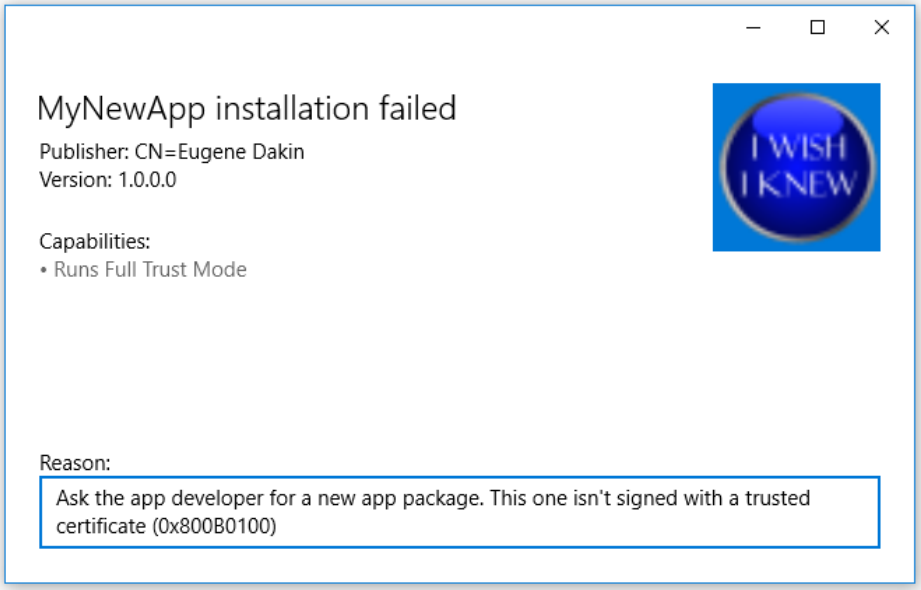
Type the following command to update MyNewApp.appx program with the new icons.

```
makeappx pack -d "C:\MyApps\MyAppFolder\MyNewApp\PackageFiles" -p
"C:\MyApps\MyAppFolder\MyNewApp\MyNewApp.appx”
```

This is one long line, and the makeappx updates MyNewApp.appx with the new icons. Below are the meanings for the three arguments used:

- **Pack** which means that a new package is to be created
- **-d** is the path to the current package files that contains our new icons that we want to update
- **-p** is the path where we want the completed appx package to be placed

If you try and run the new MyNewApp.appx program from the directory C:\MyApps\MyAppFolder\MyNewApp, then the following message will appear:



The makeappx command does not automatically sign our application like the DesktopAppConverter does. We need to run another command to sign the MyNewApp.appx installer package manually.

Note: Only sign an appx program if you are going to install it on your own machine (ie: side load) or submit the program to the Microsoft Tester to approve your program. If the program is to be uploaded to Microsoft Store, then do not sign the appx program, as the Microsoft Store will sign it for you.

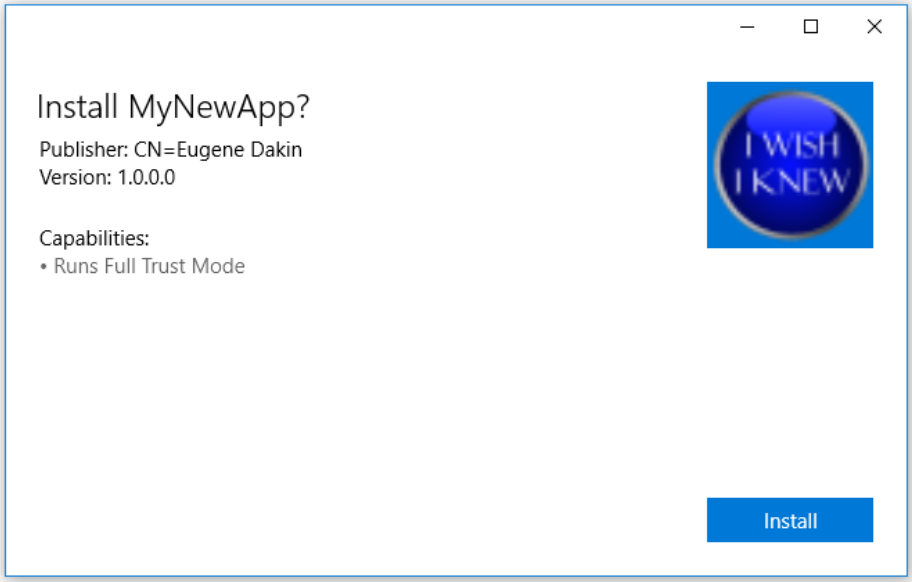
While the command prompt (cmd) window is opened and is at the directory C:\Program Files (x86)\Windows Kits\10\bin\x64\ , run the following command to sign our app with the new certificate:

```
signtool sign /a /v /fd SHA256 /f "C:\MyApps\MyAppFolder\MyNewApp\auto-generated.pfx" /p "123456" "C:\MyApps\MyAppFolder\MyNewApp\MyNewApp.appx
```

Parameters for signtool are shown below:

- **/a** searches all the certificates and uses the best signing certificate
- **/v** provides the user with verbose (many) messages
- **/fd** uses the SHA256 file digest algorithm for creating file signatures
- **/f** is the file containing the certificate
- **/p** is the password used to open the file, and the default is 123456

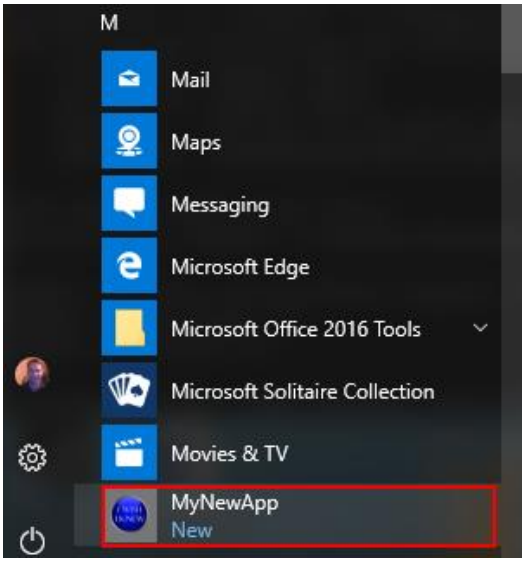
The last path is where the new appx file will be placed. The new installer application has been built and when the MyNewApp.appx file runs, the following installation happens:



If there is an issue with signing the app, then repeat the steps when running the auto-generated.cer certificate installation steps on the previous pages in the ***Building the Xojo Application*** section.

When your application is going to be reviewed by a Microsoft employee, then you will need to create a signed app for them to review. Usually the Appx is uploaded to a private specialized website where the Microsoft employee installs your program on their computer and verifies your program. When your Appx is uploaded to the Microsoft store then this final Appx program is not to be signed, as Microsoft will sign it prior to being added to the Microsoft Store.

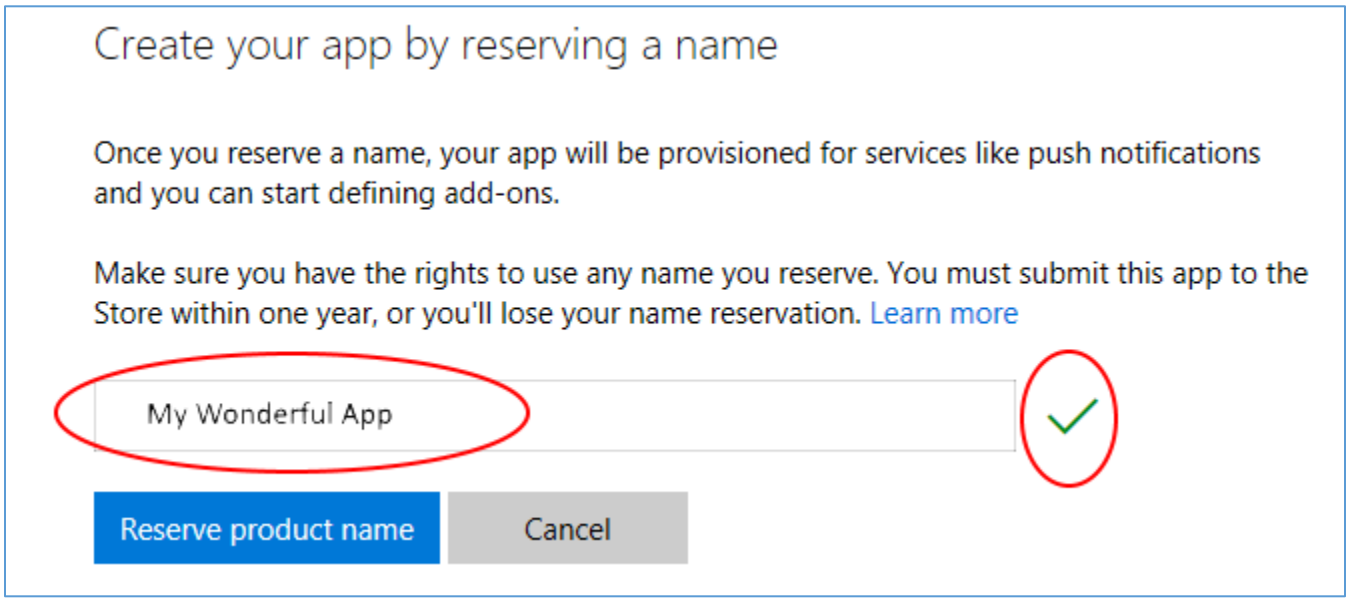
When the installer MyNewApp is added, the program runs, and the icon is located in the Start bar.



Submission to Microsoft

With your converted appx program that is working perfectly, the next step is to upload an unsigned version to Microsoft. Sign-in to your development account at: https://developer.microsoft.com/en-us/windows

The next step is to reserve the name for your application by going to **Dashboard overview** or **All apps** page and click **Create a new app.** Add your name in the line and press **Check availability**.



If **Check availability** turns into a green checkmark, then press **Reserve product name**. Names can be reserved for one year. With your program built, next clickn on Windows-> Apps->Overview and click on **My Wonderful App** and press **Start Submission**.

In the Packages section, add your appx package and upload it to Microsoft.

PublisherDisplayName Error

One of the errors I received had the following message:

> Package acceptance validation error: The PublisherDisplayName element in the app manifest of MyWonderfulApp.appx is CN=Eugene Dakin, which doesn't match your publisher display name: Eugene Dakin.

The AppxManifest xml file needs to be modified. This file is located at: C:\MyWIE\WIEFolder\WellInjectEstimator\PackageFiles. Open it with a regular text editor like WordPad and edit the one line to match the error that was shown on the website:

```
Change this line (remove CN=):
<PublisherDisplayName>CN=Eugene Dakin</PublisherDisplayName>

To this line
<PublisherDisplayName>Eugene Dakin</PublisherDisplayName>
```

Now rerun the MakeAppx command with all of the parameters to rebuild the program in the Command Prompt with Administrative Priviledges.

Note: There may be many modifications needed with the AppxManifest as some of the information will be assigned by Microsoft that needs to be modified in this manifest and then the MakeAppx command will need to be rerun.

There will likely be many changes to the AppxManifest through the various verification steps when your Appx program is being verified by Microsoft. Each time the AppxManifest is modified then the program makeappx program will need to be re-run to create an updated version of your Appx program with the new AppxManifest information. Resubmitting your application by uploading it within the Microsoft Dev website will provide additional information on changes that are needed within the AppxManifest.

Centennial Early Adopter Program Addendum

Another error flag that appeared was the following message when my appx program was uploaded to the Microsoft website.

> Package acceptance validation error: You need to accept the Centennial Early Adopter Program Addendum before you can submit this app.

Clicking on the link brings you to this website:
https://developer.microsoft.com/dashboard/account/agreements?agreementType=Centennial

This usually means that your account has not yet been signed up for Desktop Bridge. To signup for Desktop Bridge access, go to the following site and successfully submit your information:

https://developer.microsoft.com/en-us/windows/projects/campaigns/desktop-bridge

You will probably receive emails verifying your account fairly quickly, which is within a week or so.

Submit App Details

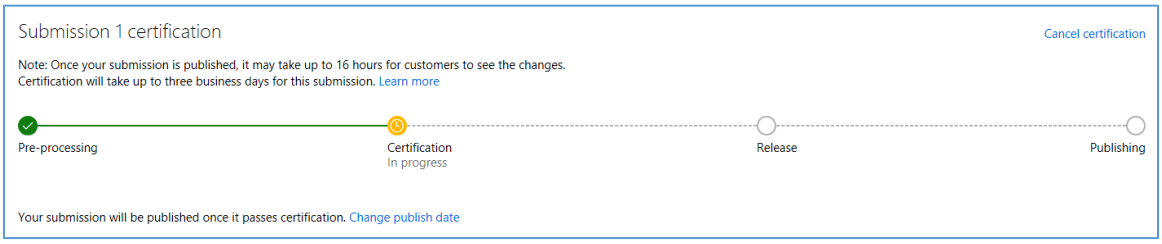
The following link is where you will provide your information along with application information and the Windows Store Privacy Statement for a Microsoft employee to review.

https://developer.microsoft.com/en-us/windows/projects/campaigns/desktop-bridge

If your website is missing information about your app and the Privacy Statement, then your program will not be accepted. Usually the Microsoft employee will ask you to resubmit the desktop-bridge form again.

Submission 1 certification

After your application has been tested, AppxManifest modified, payment options, and legal information has been completed, then your final unsigned Appx application will be submitted for certification. This is the last steps before your application is published to the store.



The status is shown for your application within the Microsoft Dev Centre by clicking on Dashboard-> Apps-> Your App Name -> Submissions -> In Certification view progress link.

Tips to Build your InnoSetup Installer

Some helpful hints are provided in this section when building the InnoSetup package for your program. It is recommended to install the dll's that are included beside the executable that Xojo creates. On the other hand, do not include the MS Runtime files in the InnoSetup installer package.

Read and Write Data

Most programs usually save a file for the user preferences, or license files, etc. It is recommended that this data be placed in the C:\Users\<username>\AppData\Roaming folder, which can read and write data safely.

To create a folder in this section, a code snippet to create the folder 'MyApp' with the total folder path C:\Users\<username>\AppData\Roaming\MyApp is:

```
Dim f as FolderItem = SpecialFolder.ApplicationData.Child("MyApp")
f.CreateAsFolder
```

Note: Sometimes the app data folder is not in the location C:\Users\<username>\AppData\Roaming\MyApp. Verify this by installing the Appx on your computer first and making sure the program accesses files the way it should.

It is here where JSON, text files, sqlite database, an other read/write data can be placed.

Note: Do not place writable data in the C:\Program Files (x86)\ directory. Data in this secure directory can be read but not written.

Frequently Asked Questions

- Do I need to purchase a certificate so sign my application for the Microsoft Store?
  - No, Microsoft will sign the AppX installer package for you. If the application is signed then it will likely be rejected.
- Do I need to sign my application if I am going to side-load it on my Windows 10 computer for testing?
  - Yes, you will need to sign your application with the generated certificate for installation on your Windows 10 computer for testing.
- Do I need to add icons to be accepted into the store?
  - Probably not, but icons provide professionalism to your application.
- Is the MakeAppx step necessary?
  - Yes, it is mandatory for the Microsoft Store. This is the way to add icons and update the required AppxManifest information to your application.
- What accompanying Xojo files need to be added to InnoSetup?
  - All of the files in the Lib folder, which includes the XojoGUIFramework32.dll. In the Output folder PackageFiles, drag a copy of XojoGUIFramework32.dll and place it next to the executable. Do not include the 40 windows DLL from the Extras>Windows Runtime>Files folder, and do not include the msvcp100.dll nor the msvcr100.dll files.
- Once my program has been added to Windows Store can I add a link in the forums?
  - Yes, feel free to put a link and a screen grab or two in Xojo forums by clicking on the https://forum.xojo.com/36287-xojo-apps-in-the-windows-store/0 link.
- If my app collects user data, do I need to have a privacy statement?
  - Yes, this is mandatory, and you may want to seek legal assistance if you are not familiar with legal terms and conditions. Below are two of many possible example Privacy Policies:

    PRIVACY STATEMENT
    We believe in the strict right to privacy. In other words, your information is your own, and you are entitled to expect others to respect it as your private property.
    We do not collect any information about you, and do not use tracking software such as cookies or other technical means.
    Servers automatically record IP addresses of computers connecting to them to be able to deliver pages, but none of this information is used for any tracking operation.
    We strictly forbid ourselves to collect, record or otherwise process any private data belonging to our visitors without their expressed consent.

  - Another Privacy Policy:

    ### Privacy Policy
    ### Windows Store: Privacy Policy
    MyCompany is committed to protecting your privacy. Most of our services can be enjoyed without having to communicate to MyCompany any personal data, but in some cases it may be required to provide certain information in order to

access our services. This privacy policy describes the collection and processing of data in such cases. This privacy statement applies only to the services made available by the MyCompany application in the Windows Store.

Collection and use of personal data

Personal information will be requested only in cases where such information is necessary to identify you or contact you. Typically this is done in cases where it is necessary to register in order to make use of a service, such as leaving comments or join a discussion. In such cases MyCompany services also gather information about the hardware and software used, such as IP address, browser type, operating system, domain name or Web site addresses from which a user has logged in. This information is used to improve the quality of service and to provide general statistics about MyCompany. MyCompany services also collect information on the pages visited by users within the service. Data related to visited pages are never linked to your personal information.

If you decide to give your consent, your data may be used for purposes of promotion of MyCompany activities. MyCompany may disclose personal information only if required by law or in case there is the firm belief that such action is necessary to:

- (1) comply with laws and local regulations;
- (2) protect and defend the rights or property of MyCompany.

Control and access to personal information

The personal information you provide to MyCompany for registration or for any other purpose is not shared with third parties without explicit permission from the you, except for the limited instances described above. Personal information will always be used for the purposes described above. MyCompany provides users with all the tools they need to make sure that personal information is correct and up-to-date.

Changes to this privacy policy

MyCompany reserves the right to modify this document. If we use your personal information differently than specified in this legal information provided at the time of collection of the data, users will be notified through announcements posted on MyCompany.

- Is it common for my Appx program to need many modifications to the AppxManifest file and be reloaded up through the Packages area in the Microsoft Dev Centre?
    - Yes, this is common as each step through the Microsoft security screening stages will need more updates that are specific to your Appx becoming closer to being accepted by the Appx store.
- How long does it take to have a Xojo application successfully added to the Microsoft Store?
    - If there are no problems or concerns with your application and all information has been created on your site such as promotional information on your program and a Microsoft Store Privacy Statement, then the application could be accepted by the store in as little as 1-month (yes, this is considered fast for the first program to be submitted).