

Purpose:

Build a small simple showcase lab in order to test the upcoming MSIX App Attach feature.

More info in the [official documentation](#) and [Azure Academy youtube video](#)

Disclaimer:

All sample scripts are provided for illustrative purpose and are based on my test environment and they don't represent official Microsoft best practices.

Useful links:

[Official Documentation](#)

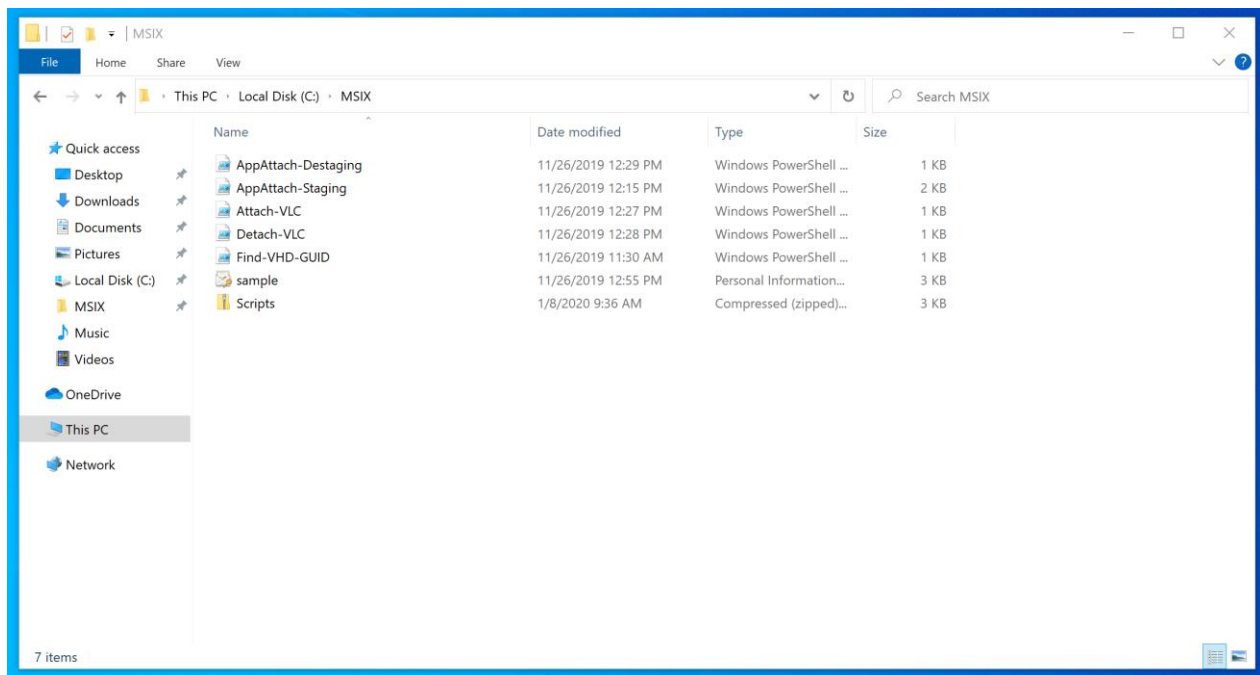
[Azure Academy video](#)

First of all we need the last Windows 10 2004 so you can install it on a physical machine, a virtual machine hosted by Hyper-V, VMware, Azure, AWS... It doesn't matter because MSIX app attach is a Windows 10 feature and is not linked with some specific feature available only on Windows Virtual Desktop.

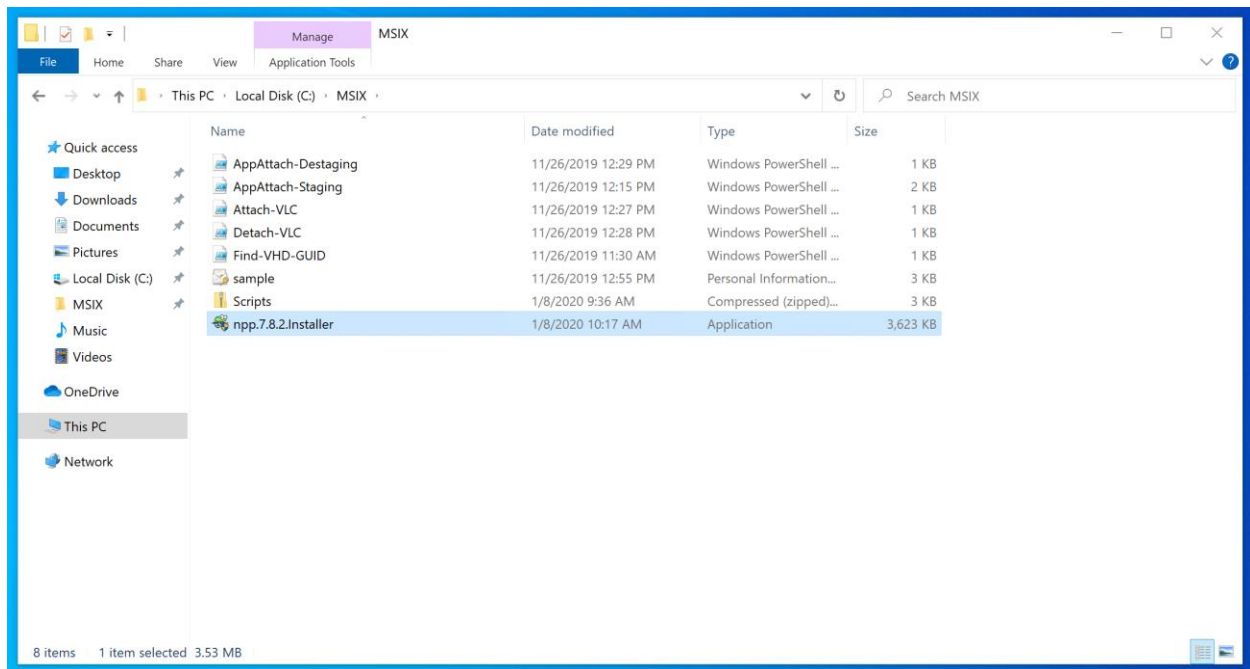
Then you can download the scripts.zip file that you can find in my Github repository

<https://github.com/mmoioli/WVD/tree/master/MSIX%20Lab>

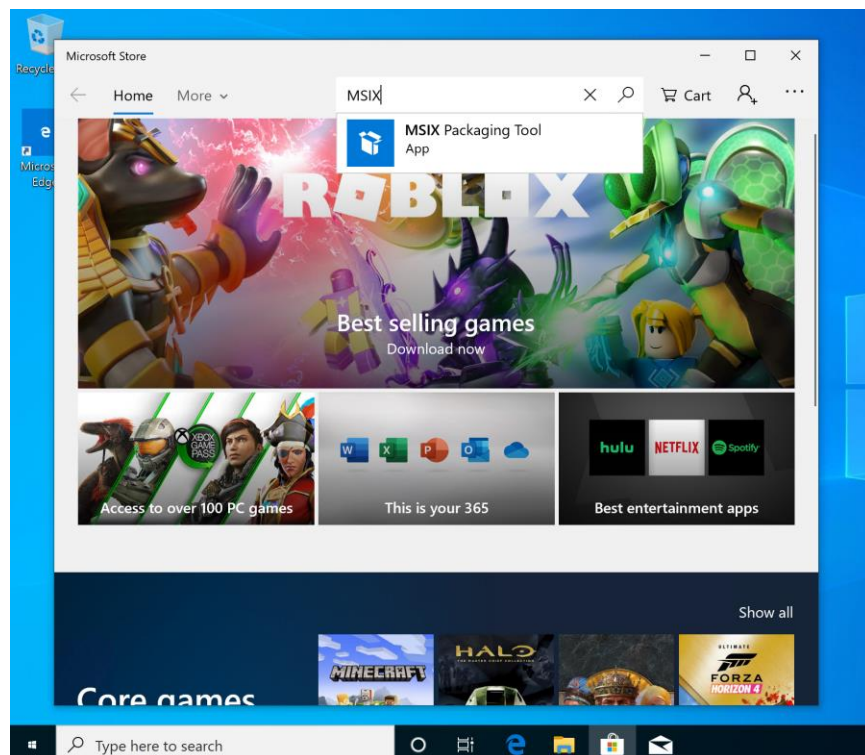
Inside the Windows 10, **create a C:\MSIX directory** and **copy the Scripts.zip** file and extract the files inside



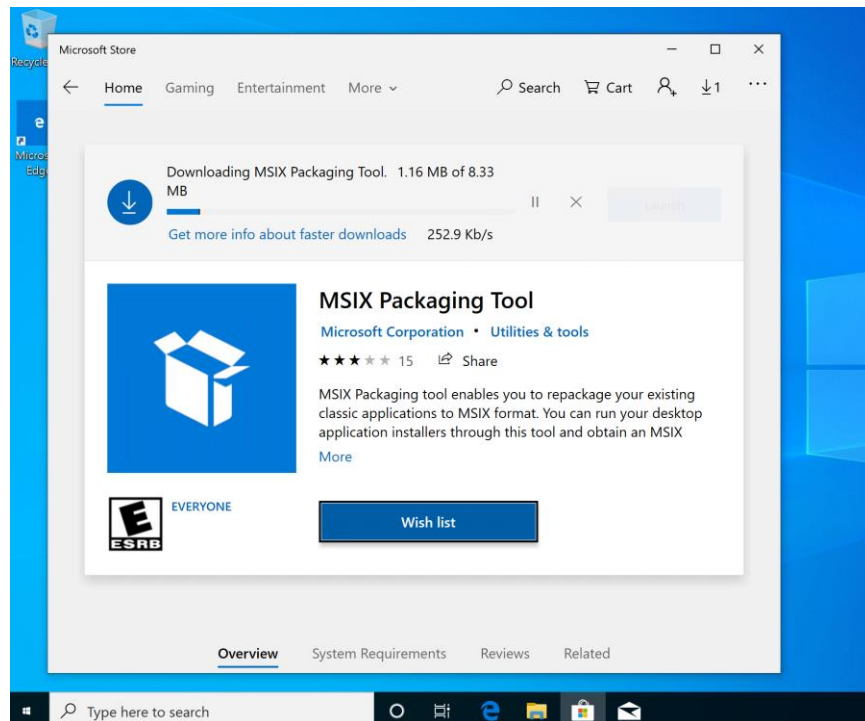
Download/Copy the installation files for an application like Notepad++



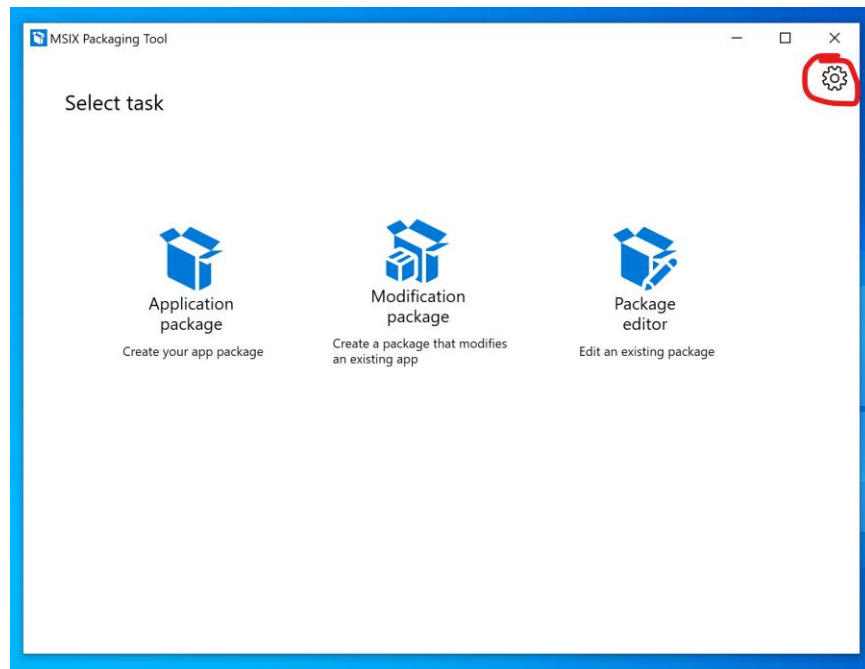
Open the Store App and search for MSIX Packaging Tool



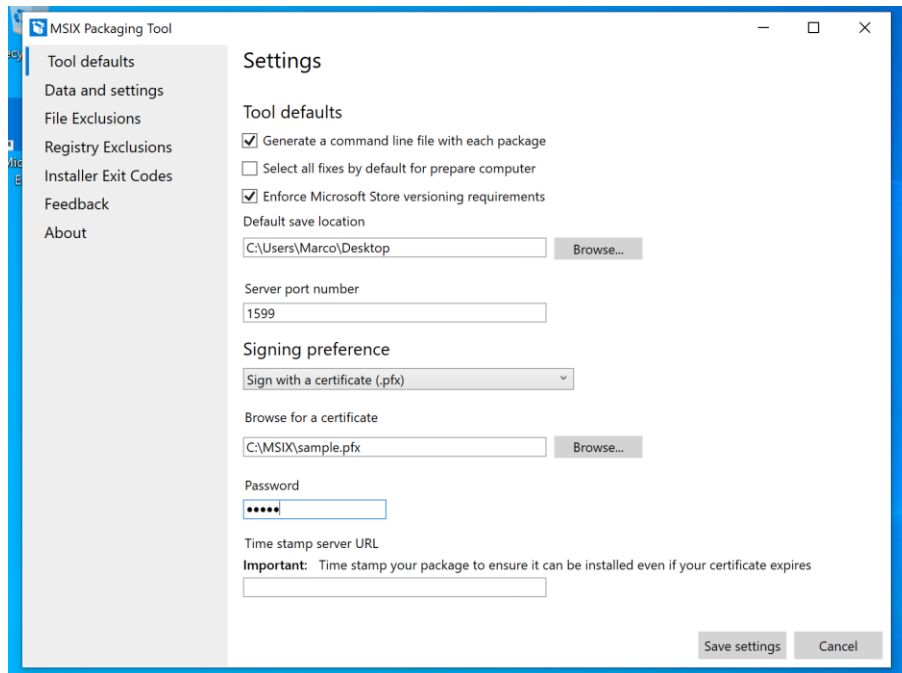
Install It



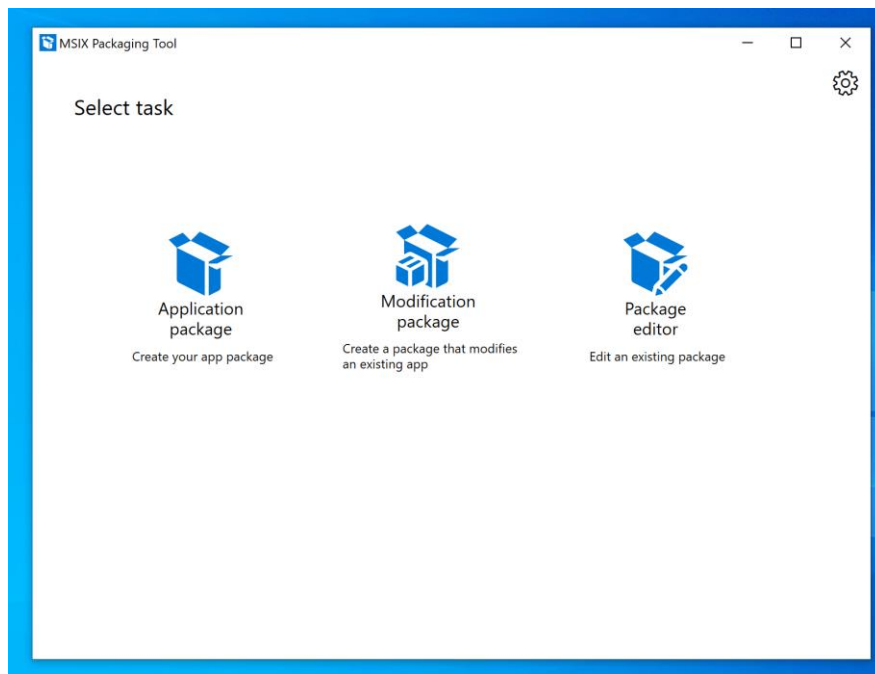
Open the app MSIX Packaging Tool and click on the wheel



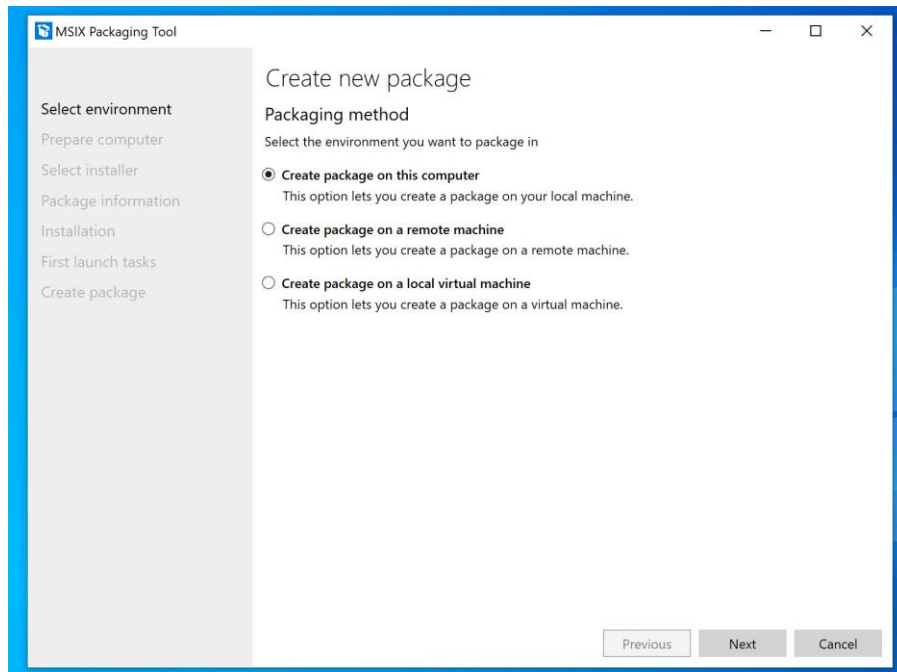
Select “Sign with a certificate (.pfx)”, Select the “sample.pfx” certification that you find in C:\MSIX and use the “MSIX!” password



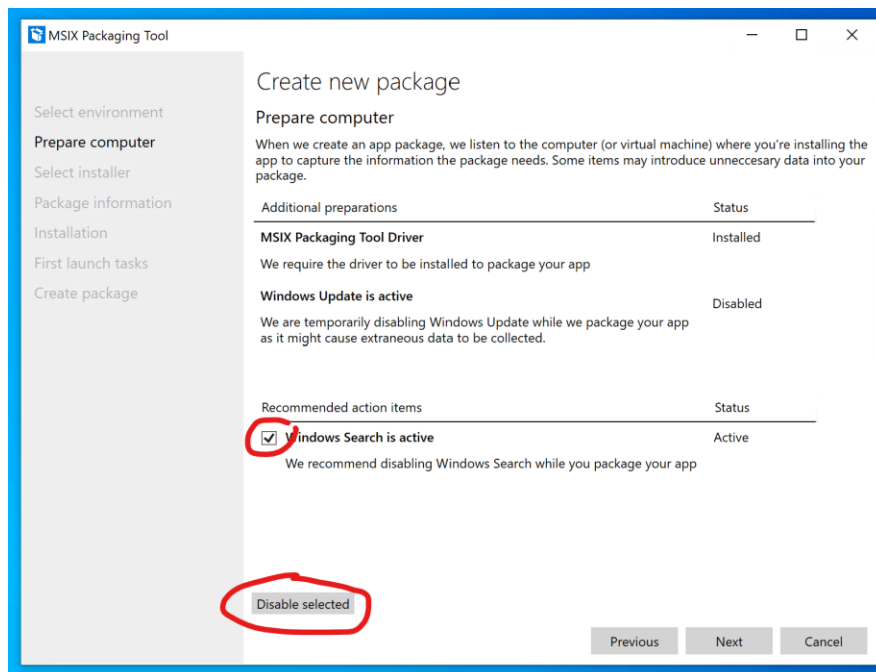
From the MSIX Packaging Tool click on “Application package”



Next



Click on every “Recommended action items” and on “Disable selected”



Select the Notepad++ installer

MSIX Packaging Tool

Create new package

Choose the installer you want to package

If you don't have an installer, click Next to create one. You'll choose and run the files you need to install your app, and the MSIX packaging tool will create the installer for you.

C:\MSIX\ppp.7.8.2.Installer.exe

Specify installer arguments (optional)

Warning: A template file will not be generated unless an installer argument to run silently is provided

Signing preference

Sign with a certificate (.pfx)

Browse for a certificate

C:\MSIX\sample.pfx

Password

•••••

Time stamp server URL

Important: Time stamp your package to ensure it can be installed even if your certificate expires

Insert the required package information

MSIX Packaging Tool

Create new package

Package information

Package name *

Package display name *

Publisher name *
 Subject of the certificate provided

Publisher display name *

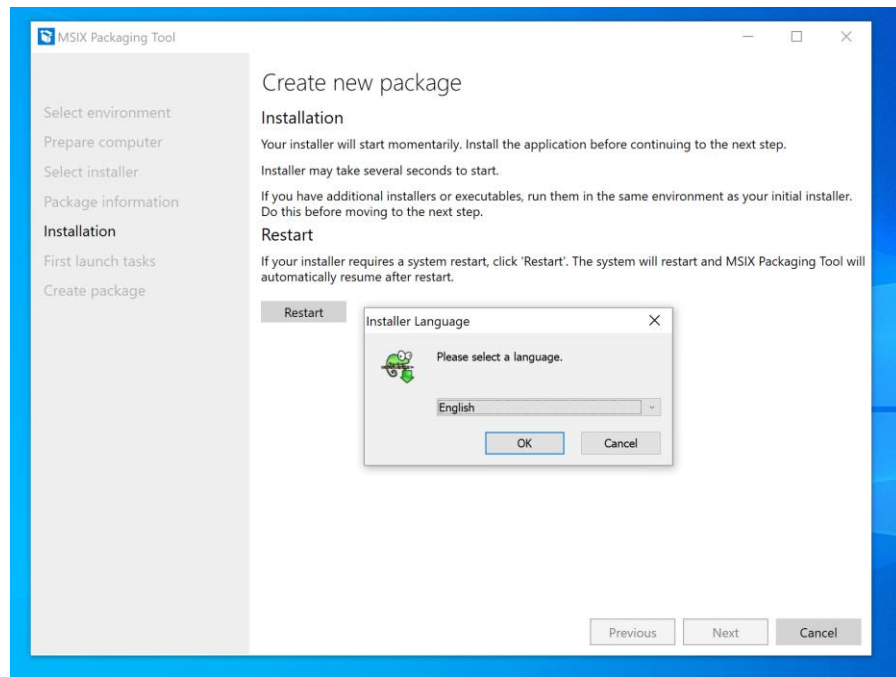
Version *
 . . .

Package Description

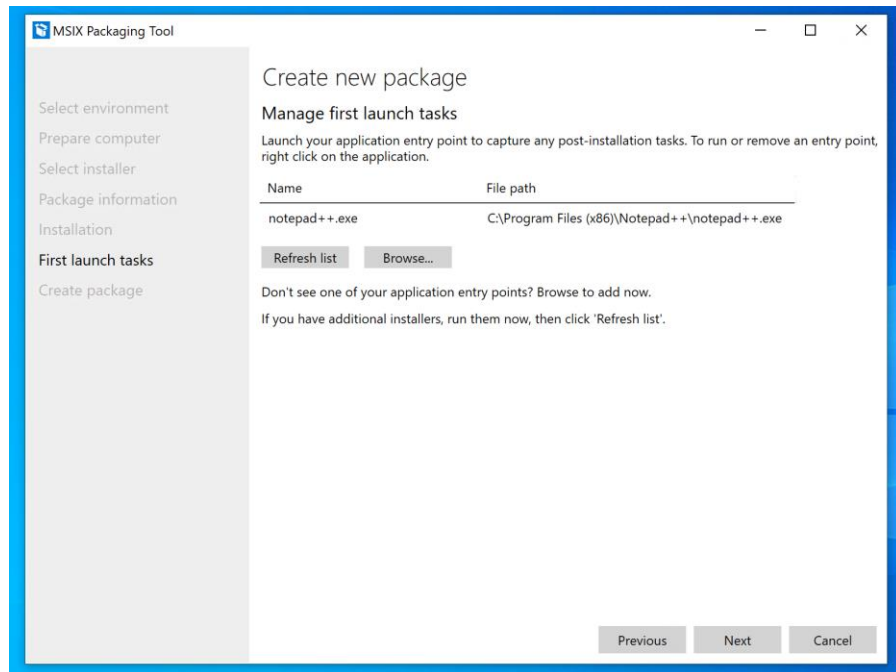
Installation location

Once you continue to the next page, installation will begin and you will not be able to return.

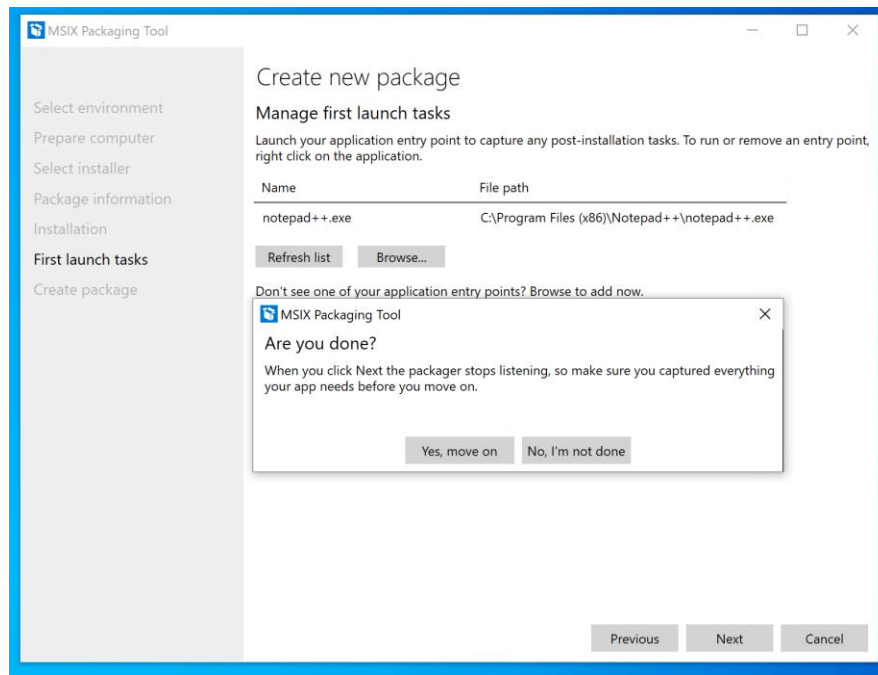
Now go through the installation process



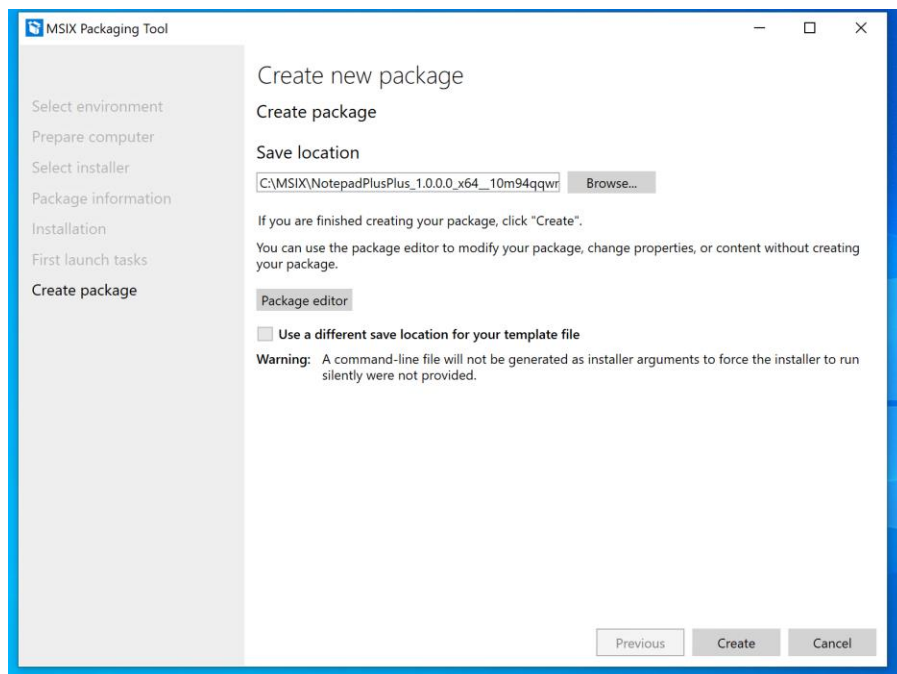
Launch one time the application in order to capture everything



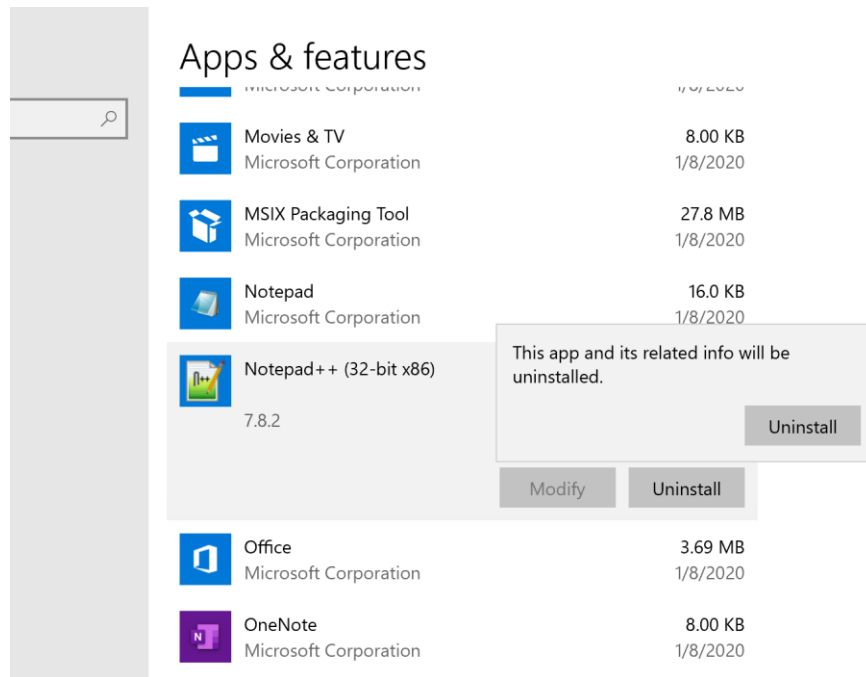
End the sequencing clicking on “Yes, move on”



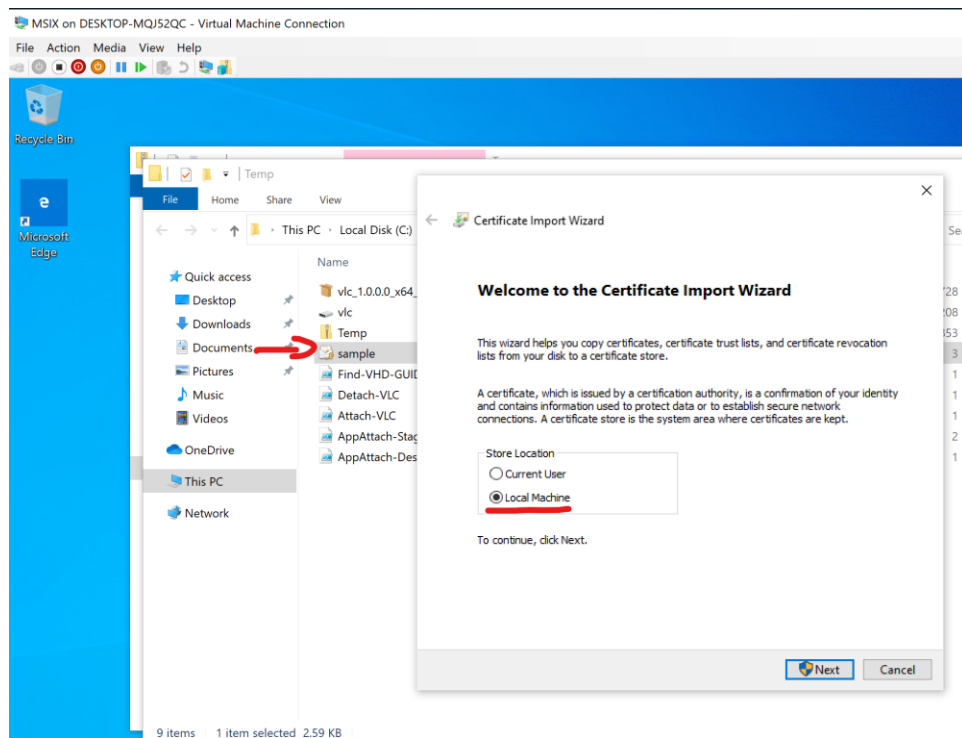
Save the package in C:\MSIX



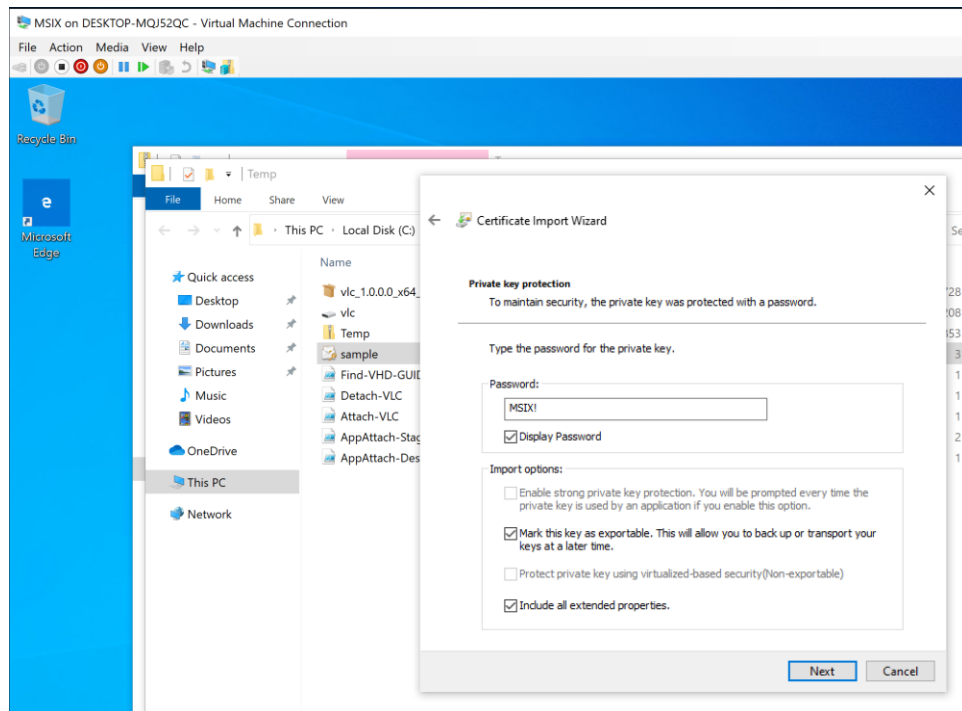
Now you can uninstall Notepad++ from Windows



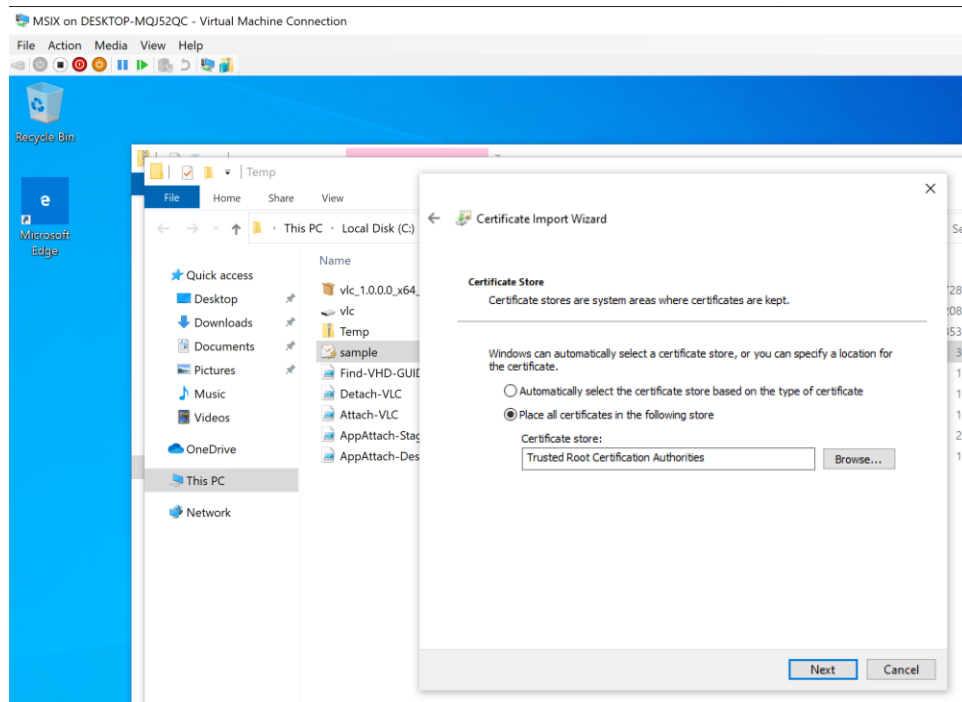
Double click **C:\temp\sample.pfx** and select “Local Machine”



The password is “MSIX!”



Place it in “**Trusted Root Certification Authorities**”



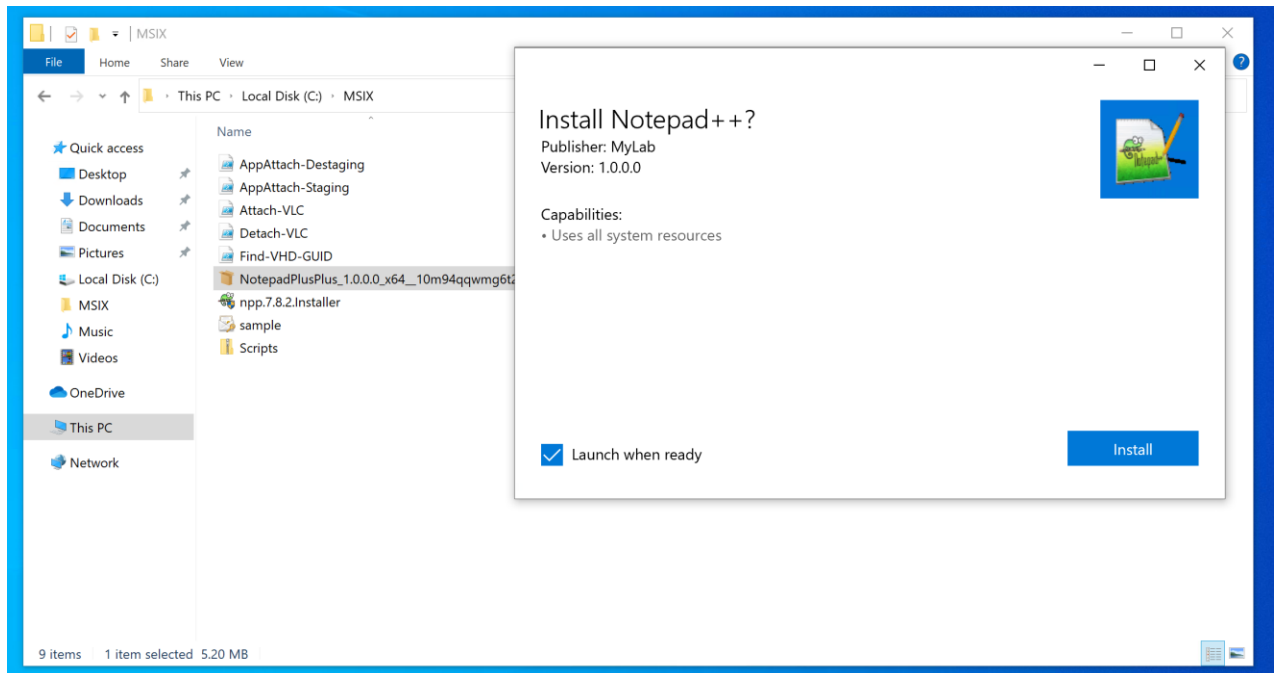
The certificate is a self signing certificate that I created using this command:

```
New-SelfSignedCertificate -DnsName sample.contoso.com -Type CodeSigning -  
CertStoreLocation Cert:\CurrentUser\My
```

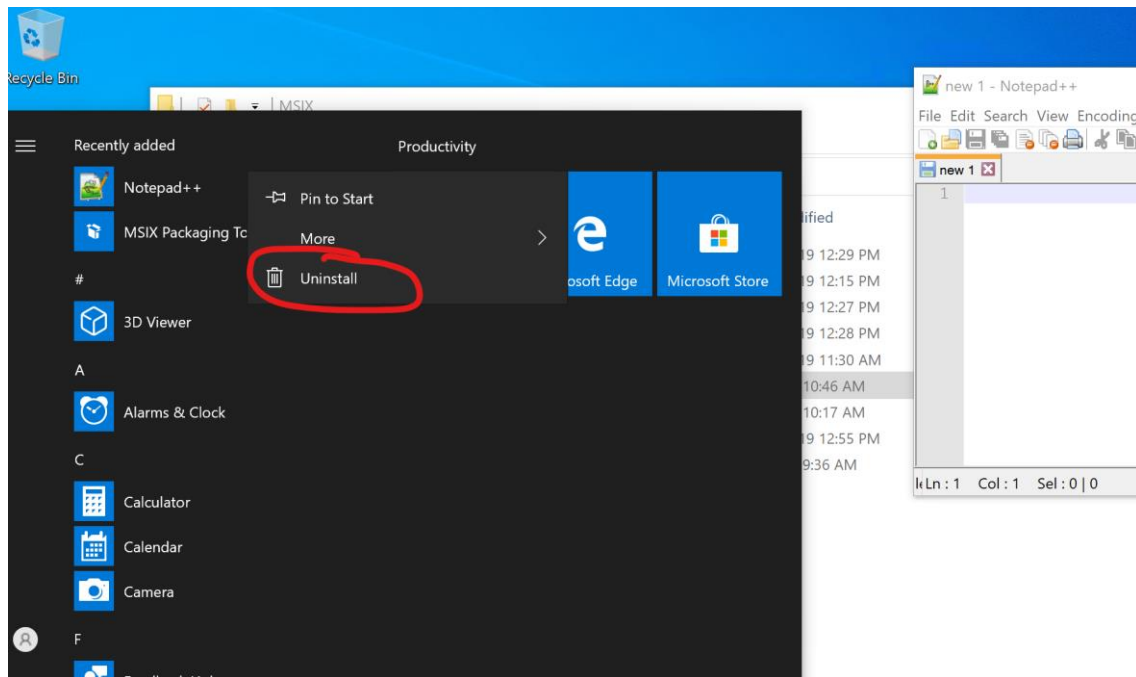
Use it only in a lab environment not in production!!!

The certificate is needed because when I created the MSIX package I signed it with this certificate.

So now you can try the MSIX package created: simply double click on it and install just to check that the MSIX package is good...



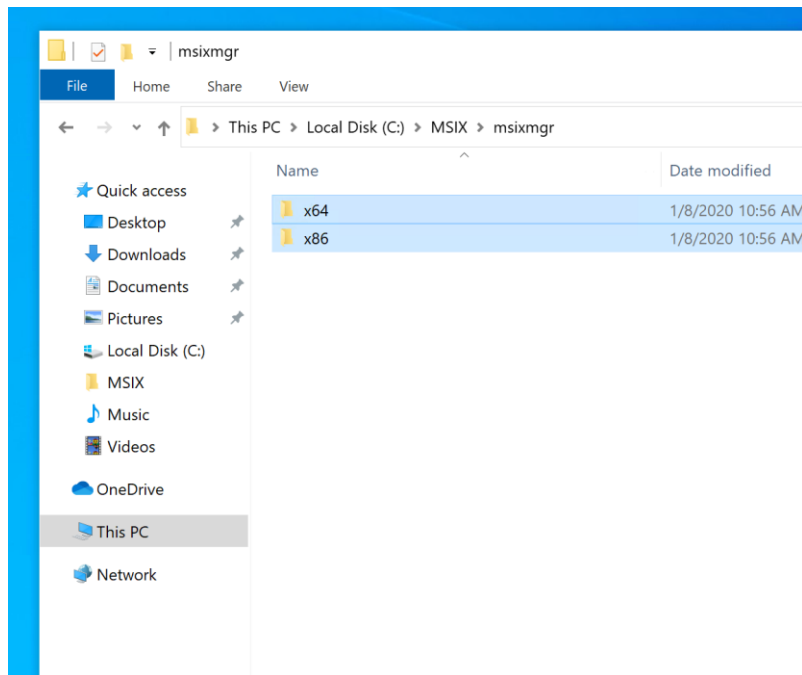
If everything is good you can use Notepad++ and you can uninstall it simply using a right click on the icon



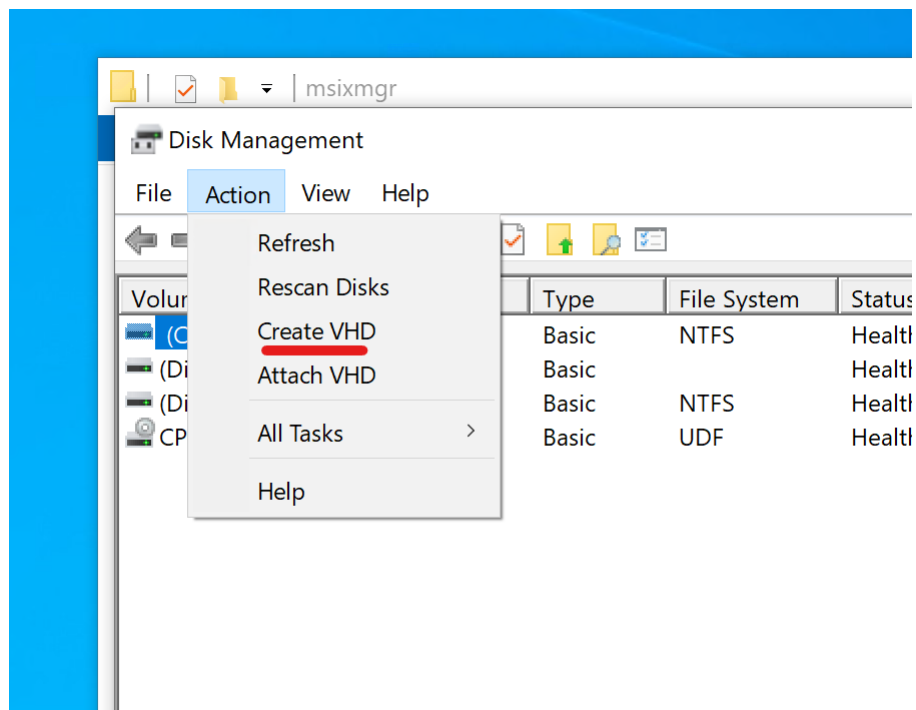
Ok now we have an MSIX package containing Notepad++ that is working but this is not AppAttach so let's continue...

Generate the VHD app package

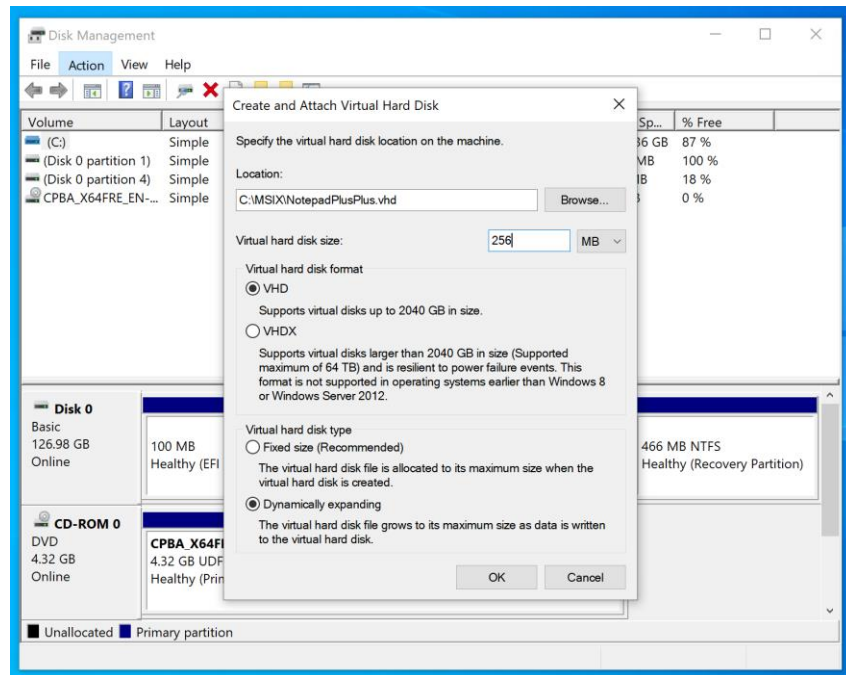
Download the last [msixmgr](#) tool and extract the files in C:\MSIX



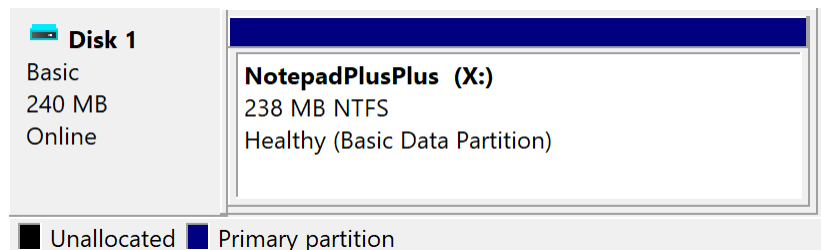
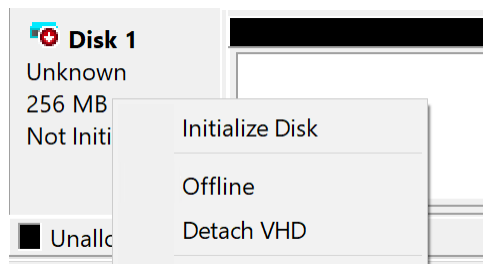
Now open diskmgr and Create a VHD



Create a Dynamically expanding VHD in C:\MSIX the size must be at least the size of the expanded MSIX package that we will insert

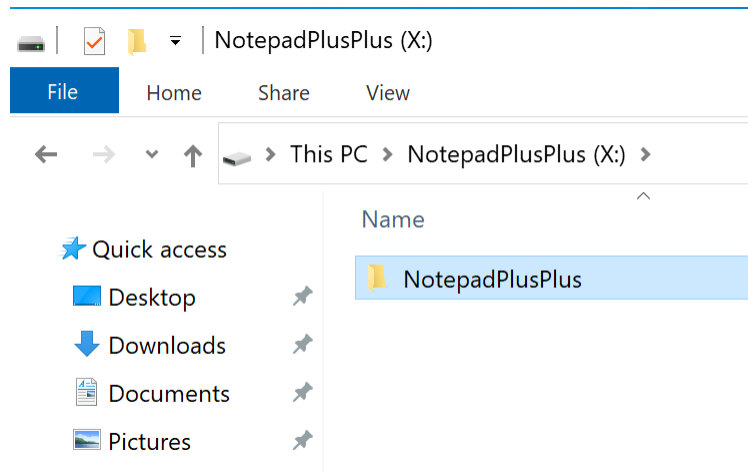


Now you can initialize the disk, create a partition and assign a letter like X



VERY IMPORTANT!!!

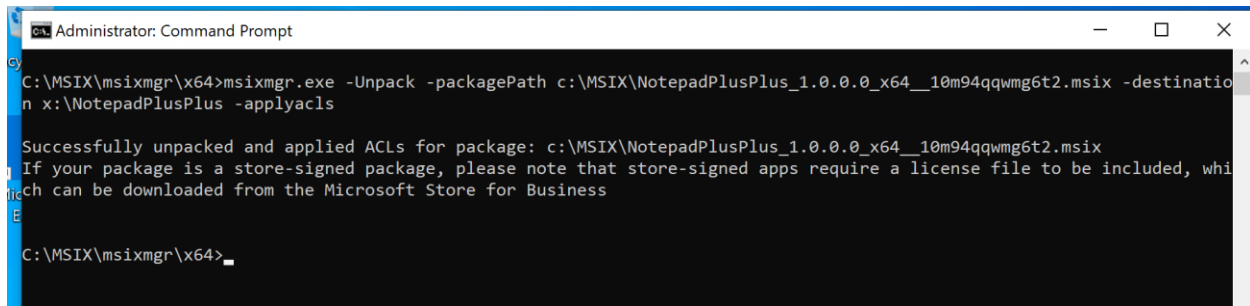
Create a folder inside the vhd the name is your choice, I used **X:\NotepadPlusPlus**



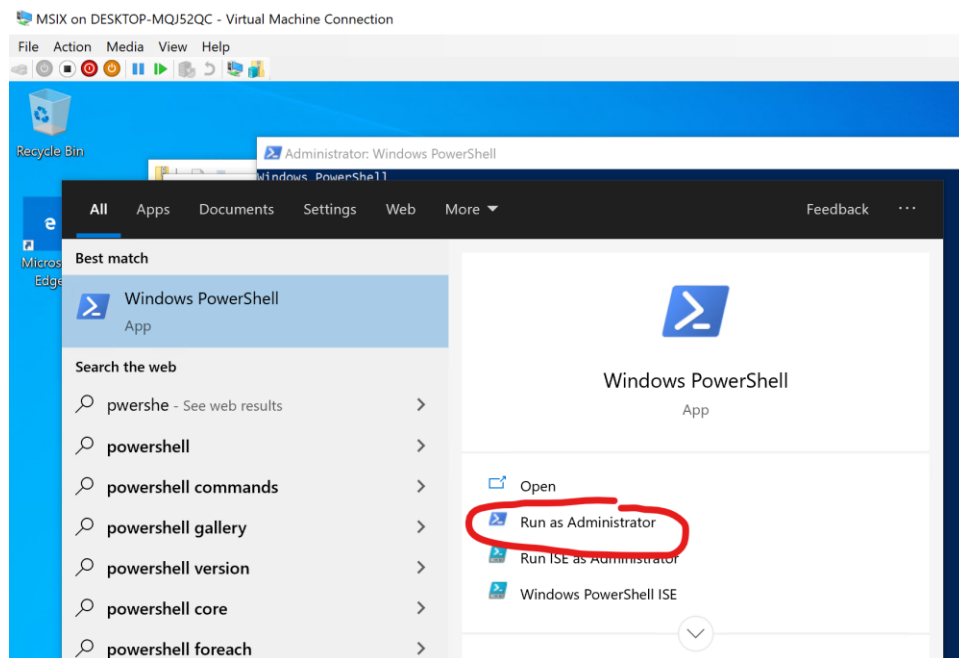
Open an elevated command prompt and type

msixmgr.exe -Unpack -packagePath c:\MSIX\NotepadPlusPlus_1.0.0.0_x64__10m94qqwmg6t2.msix -destination x:\NotepadPlusPlus -applyacls

NOTE: you must change the name of the msix package with the one generated by your packaging tool

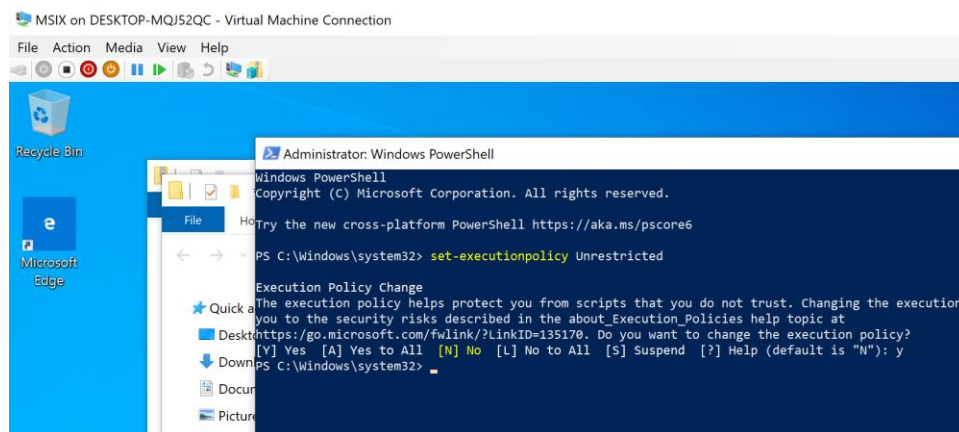


Now it's Powershell time so open it as an administrator



Just because this is a small and simple demo lab, I set the powershell execution policy as “Unrestricted”

Set-executionpolicy Unrestricted

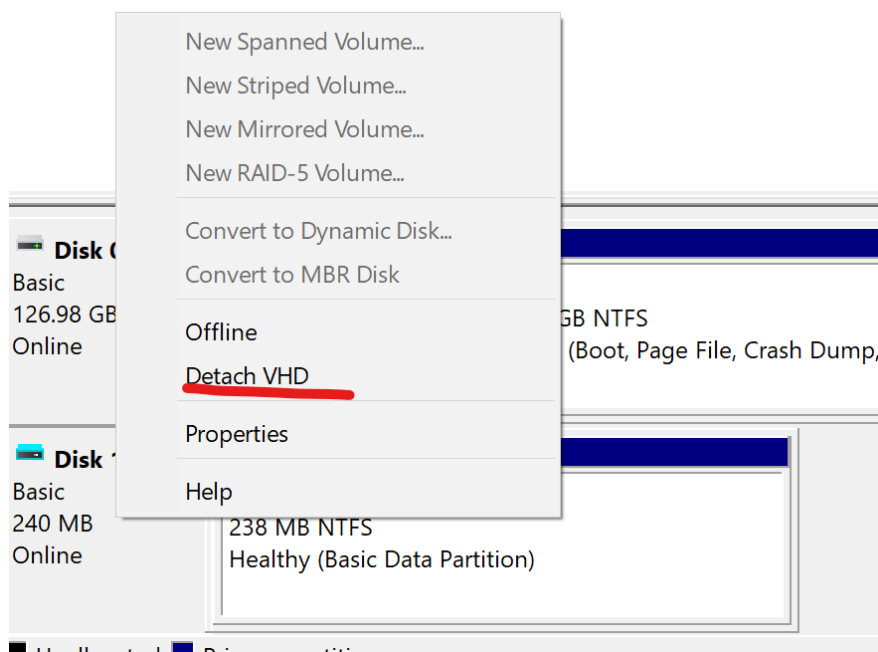


We need to retrieve the X: volume guid so the command is:

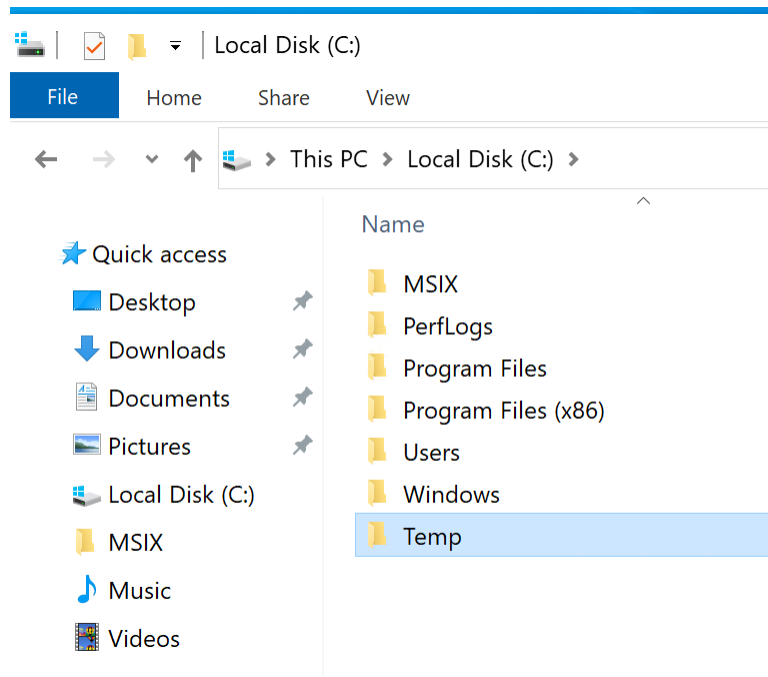
```
$volumeGuid = (((Get-volume -driveletter x).UniqueId).split('{')[1]).split('')[0]
```

```
Administrator: Windows PowerShell
PS C:\windows\system32> $volumeGuid = (((Get-volume -driveletter x).UniqueId).split('{')[1]).split('')[0]
PS C:\windows\system32> $volumeGuid
73e451a5-ea15-4fa8-beb5-94de2f98acf5
PS C:\windows\system32>
```

Now we can detach the VHD from the disk management tool

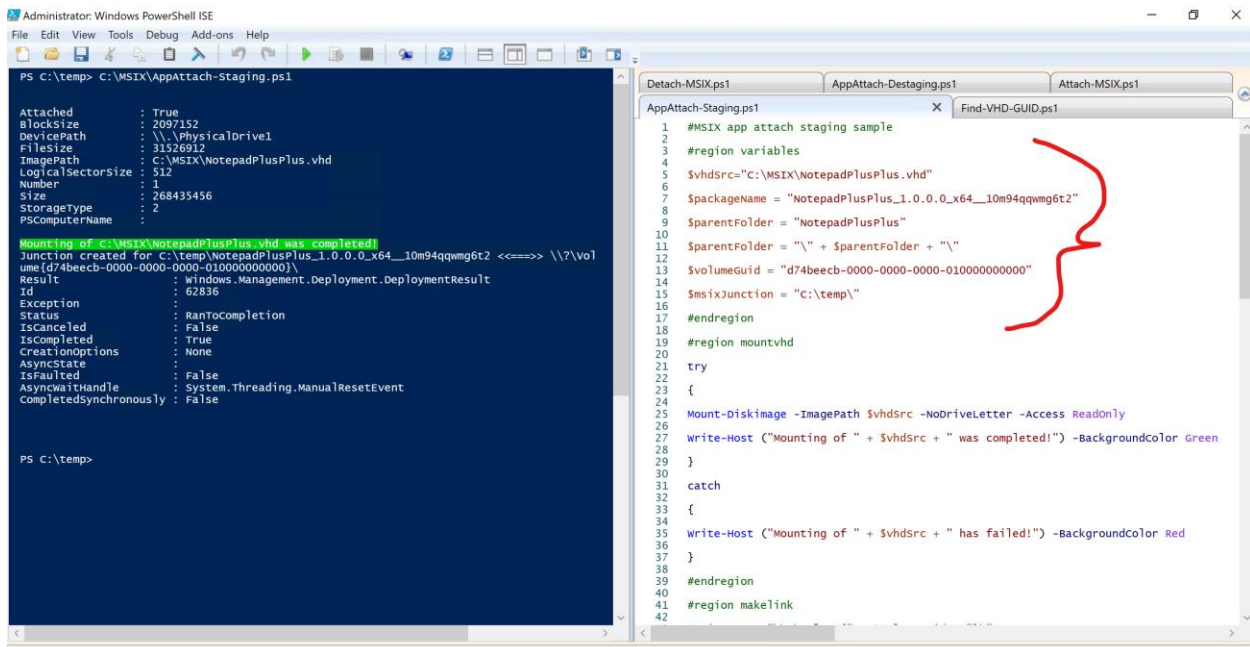


Create a Temp directory in C:



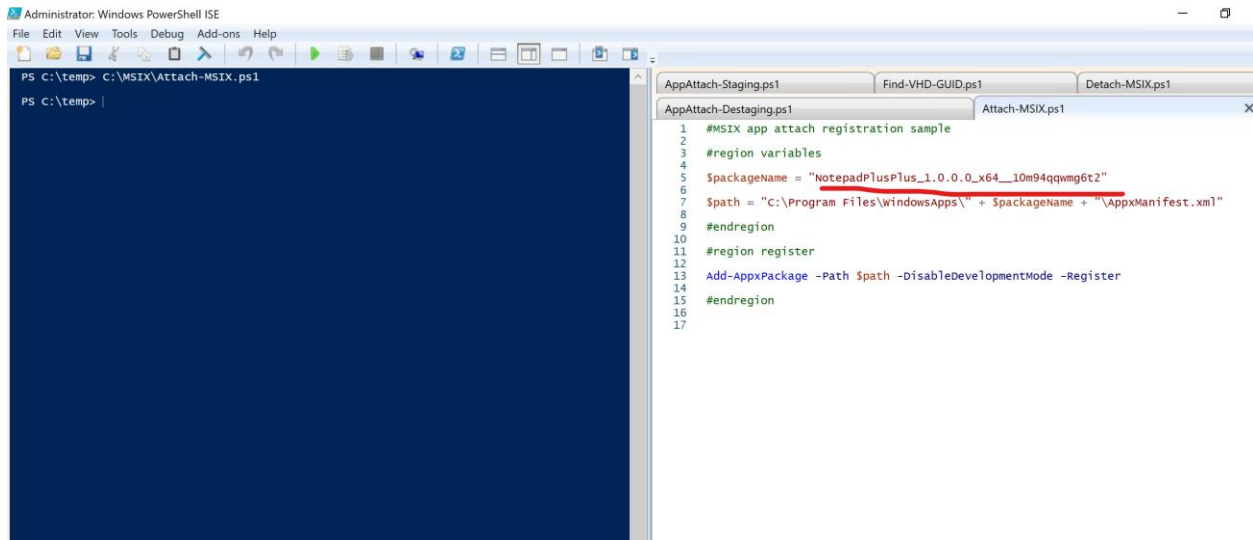
From C:\MSIX, run **.\AppAttach-Staging.ps1**

NOTE: be sure that the variables are matching the ones in your environment



After you can run **.\Attach-MSIX.ps1** and in a moment you will find **Notepad++ in the Start menu**

NOTE: be sure that the variables are fitting your environment



The screenshot shows the Windows PowerShell ISE interface. The left pane displays the command prompt with the command `C:\MSIX\Attach-MSIX.ps1` entered. The right pane shows the script `Attach-MSIX.ps1` with the following content:

```
1 #MSIX app attach registration sample
2
3 #region variables
4 $packageName = "NotepadPlusPlus_1.0.0.0_x64__10m94qqwmg6t2"
5 $path = "C:\Program Files\WindowsApps\" + $packageName + "\AppxManifest.xml"
6
7 #endregion
8
9 #region register
10
11 Add-AppxPackage -Path $path -DisableDevelopmentMode -Register
12
13 #endregion
14
15
16
17
```

If you want to detach it simply run **.\Detach-MSIX.ps1**

If you want also to clean the staging environment, run **.\AppAttach-Destaging.ps1**

NOTE: before running the scripts, be sure that the variables are fitting your environment

If you are interested in understanding what exactly is the mechanism, I suggest you to watch the [Azure Academy video](#) made by [Dean Cefola](#).