

How to self-sign your own PowerShell scripts

How to self-sign your own PowerShell scripts

Introduction

Since security has become more and more of a concern around the globe and more and more people have started to develop all kinds of applications and scripts, this tutorial will explain how you can sign scripts you have developed, and also briefly explain what PowerShell execution policies are and how to change it so that only signed scripts and binaries can be run on your machine.

A word about PowerShell execution policies

There are four different execution policy levels, and I will explain them briefly here.

- **Restricted**
Can not run any scripts. Default execution policy. Can run PS commands interactive only.
- **AllSigned**
Can run scripts. All scripts and configuration files must be signed by a publisher that you trust. Opens you to the risk of running signed (but possibly malicious) scripts, after confirming that you trust the publisher.
- **RemoteSigned**
Can run scripts. All scripts and configuration files downloaded from communication applications such as [Microsoft](#) Outlook, Internet Explorer and Outlook Express, must be signed by a publisher that you trust. Opens you to the risk of running malicious scripts not downloaded from these applications, without any warning prompts.
- **Unrestricted**
Can run scripts. All scripts and configuration files downloaded from communication applications run after confirming that you understand the file originated from the Internet. No digital signature is required. Opens you to the risk of running unsigned, possibly malicious scripts.

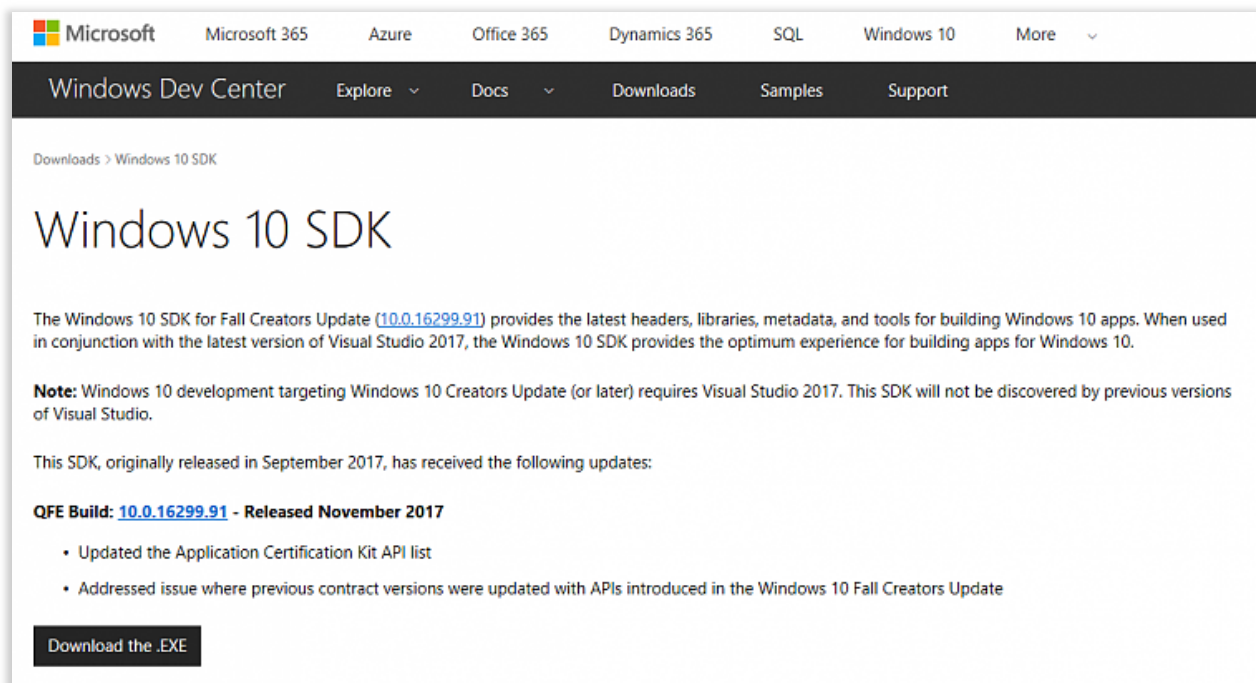
Getting the necessary tools for creating a self-signed

certificate

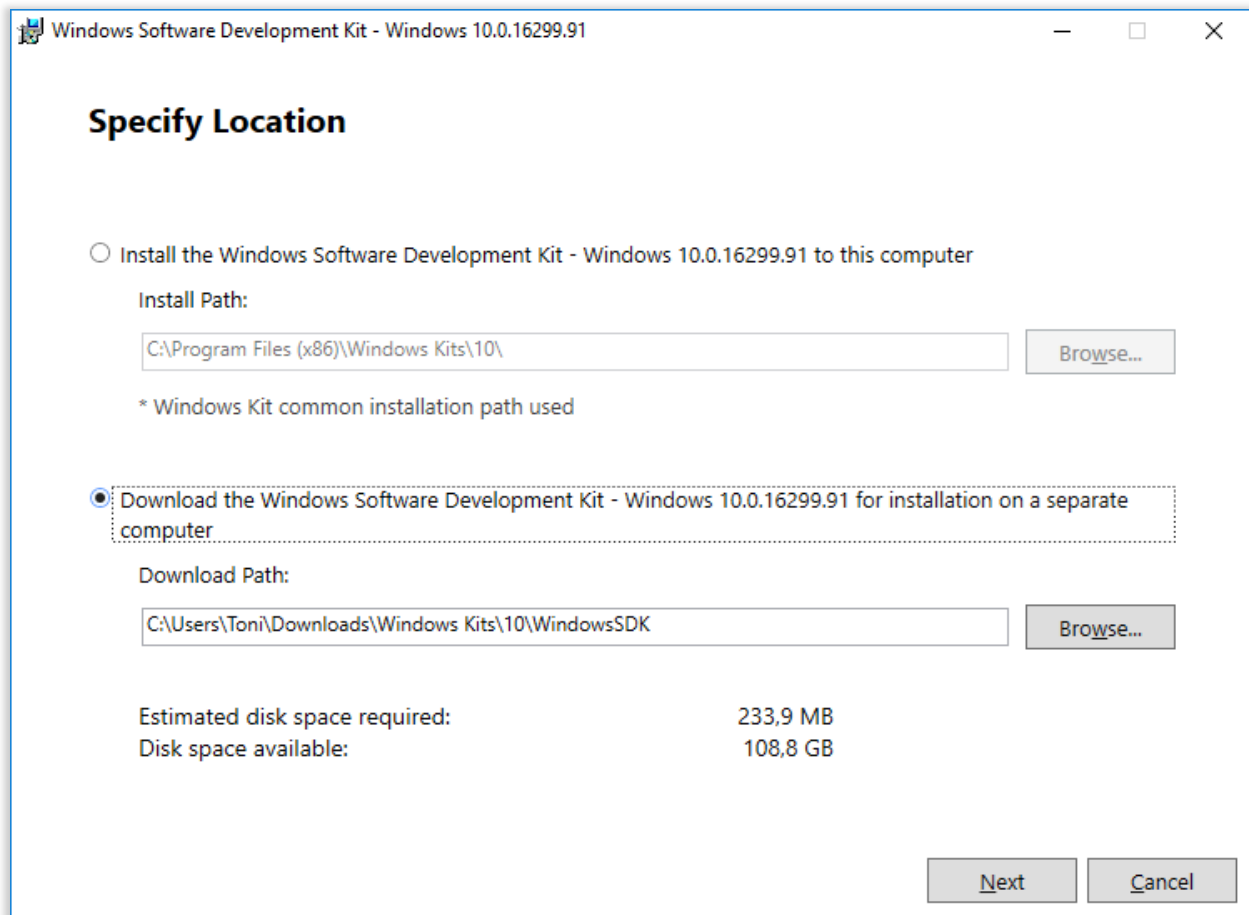
You will either need to have Visual Studio installed or manually download the Windows SDK. In order to save space and bandwidth, this section will explain how to get the latest SDK, download the minimal required files and only install one package from that download.

Steps to take for the download and installation:

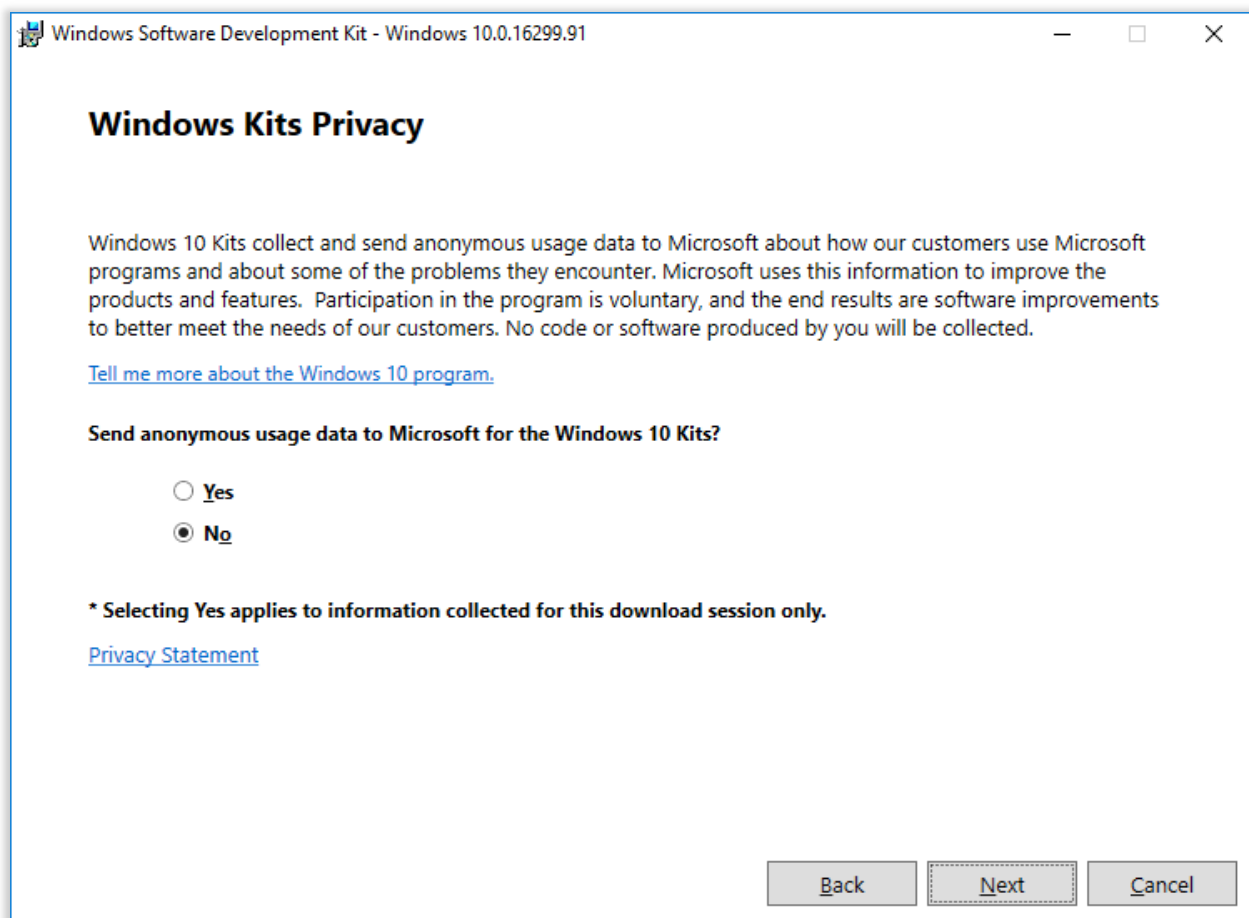
1. Using your web-browser navigate to:
[Download Windows 10 SDK - Latest](#)
or
[Download Windows 10 SDK - Archive](#)
2. Download the latest SDK installer for Windows (as of writing it is 10.0.16299.91 - Released November 2017)



3. Execute the installer
4. Select Download the Windows Software Development Kit and take note of the path and click next

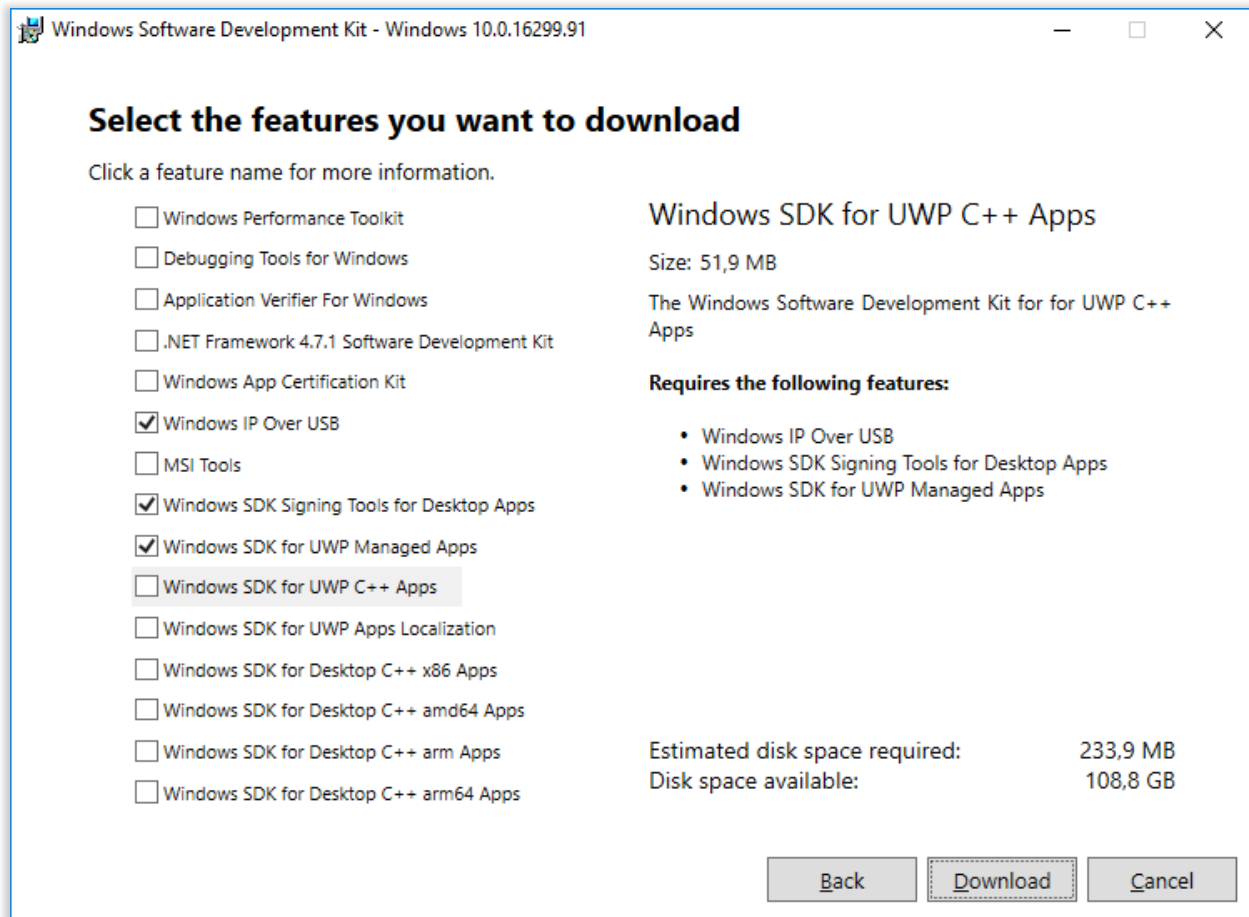


5. On Privacy page, click NO and then next



6. On Select the features you want to download page unselect everything, and then only select "Windows SDK for UWP managed apps". 2 other boxes will

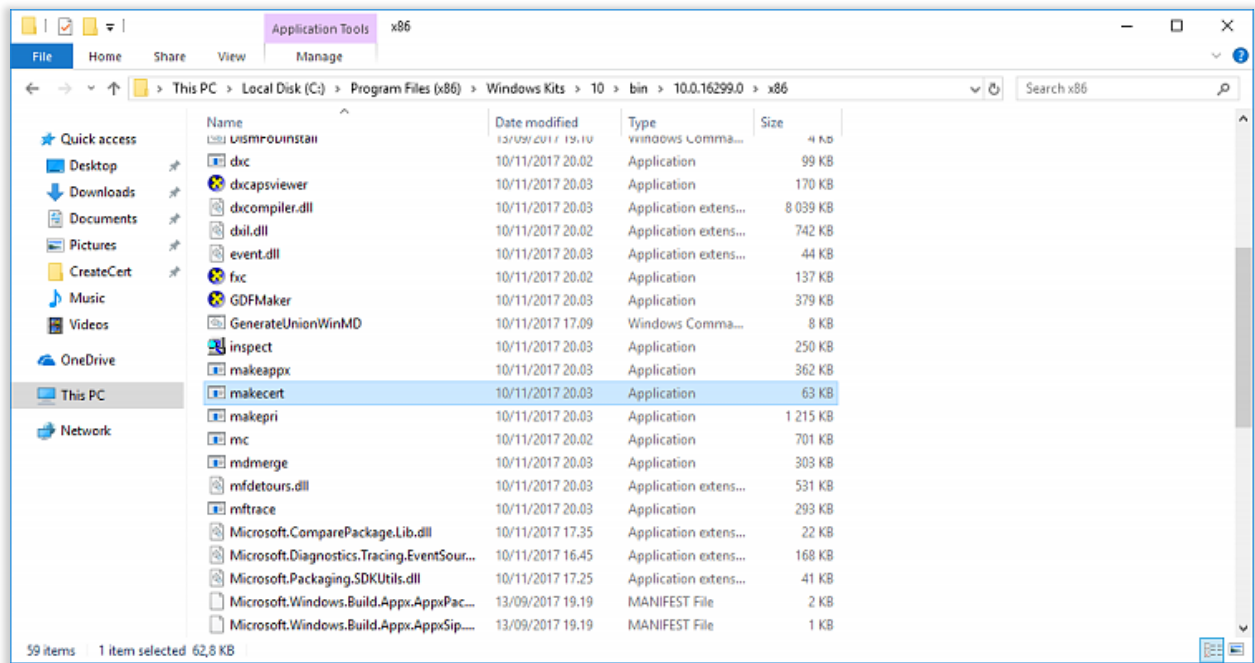
automatically be selected. Click Download.



- Once the download completes, navigate to the "Windows Kits" folder where you just downloaded the installers.
- Navigate deeper into the folder structure (\10\WindowsSDK\Installers)
- Locate the file "*Windows SDK for Windows Store Apps Tools-x86_en-us.msi*" and install it. Note, you will not get any prompts when the installer has completed.

Windows SDK for Windows Store Apps Legacy Tools-x86_en-us	10/04/2018 22.47	Windows Installer ...	288 KB
Windows SDK for Windows Store Apps Libs-x86_en-us	10/04/2018 22.48	Windows Installer ...	360 KB
Windows SDK for Windows Store Apps Metadata-x86_en-us	10/04/2018 22.47	Windows Installer ...	288 KB
Windows SDK for Windows Store Apps Tools-x86_en-us	10/04/2018 22.47	Windows Installer ...	352 KB
Windows SDK for Windows Store Apps-x86_en-us	10/04/2018 22.47	Windows Installer ...	296 KB
Windows SDK for Windows Store Managed Apps Libs-x86_en-us	10/04/2018 22.47	Windows Installer ...	288 KB

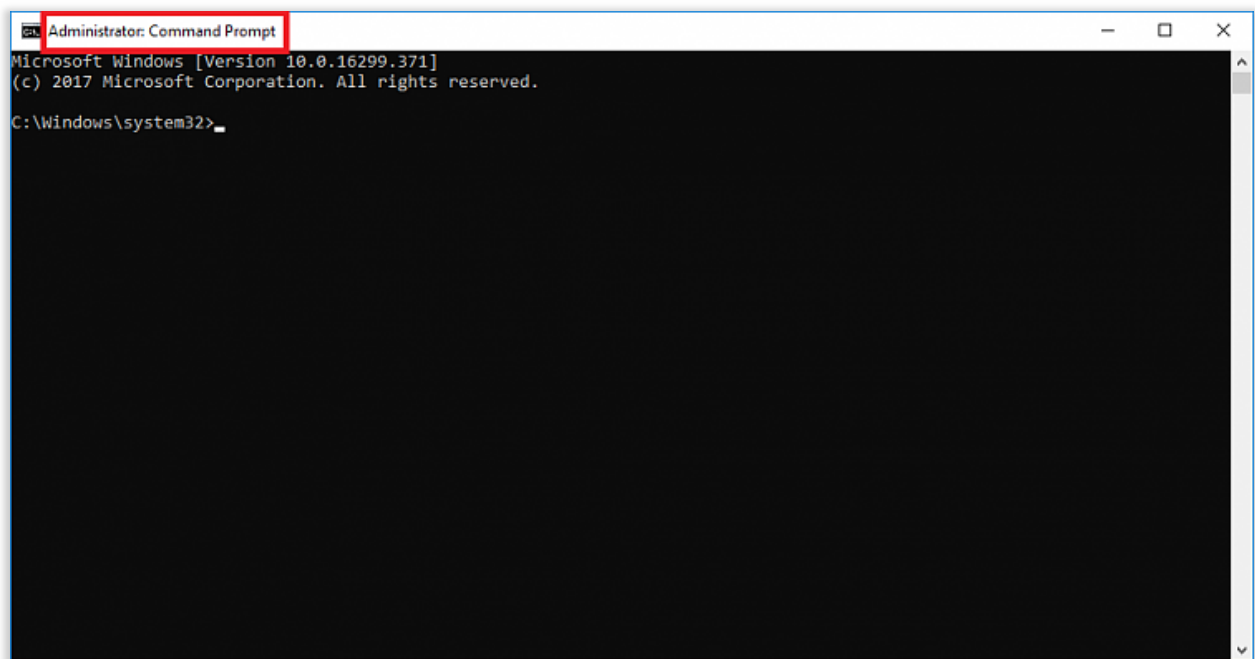
- Now you should have the required file installed (C:\Program Files (x86)\Windows Kits\10\bin\10.0.16299.0\x86\makecert.exe)



Creating the Certificate

Steps to take:

1. Open up Command Prompt with Elevated privileges.



2. Create a working directory

Code:

```
md \TEMP
```

3. Navigate to the working directory

Code:

```
cd \TEMP
```

4. Check path variables and take note of the last character.

Code:

```
set PATH
```

5. Create a temporary path to the folder where the required certificate tool (makecert.exe) is located.

If the last character in your path was a semicolon then enter this code:

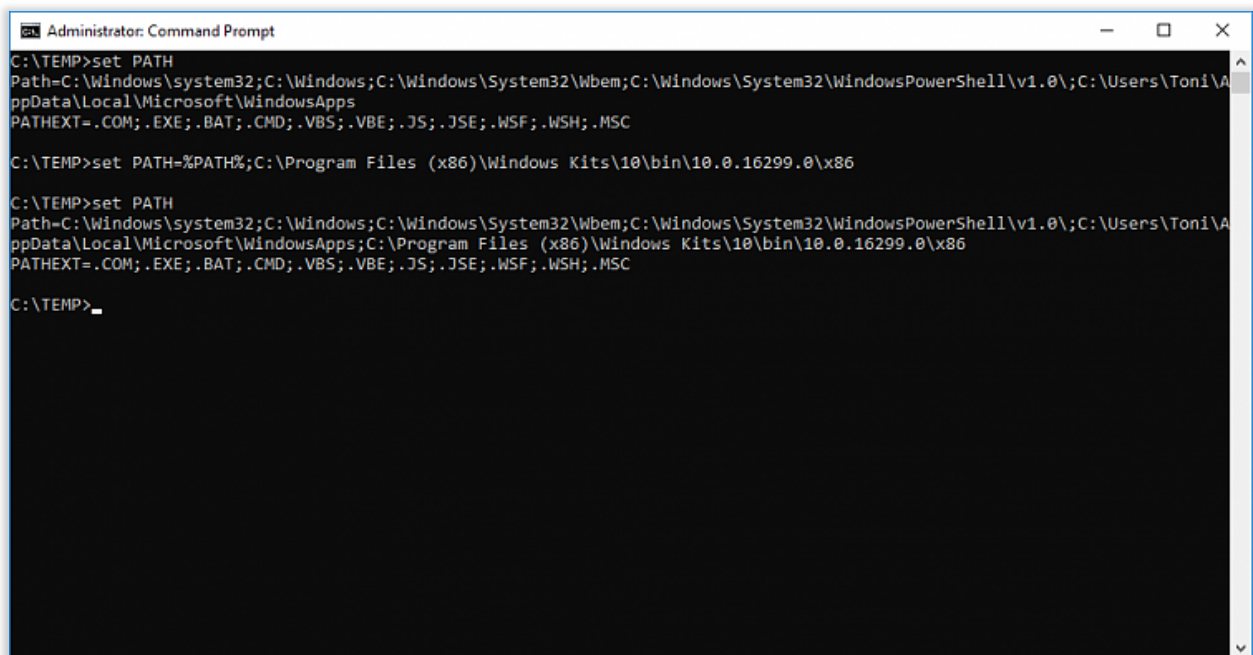
Code:

```
set PATH=%PATH%C:\Program Files (x86)\Windows Ki
```

else enter this code:

Code:

```
set PATH=%PATH%;C:\Program Files (x86)\Windows K
```



```
Administrator: Command Prompt
C:\TEMP>set PATH
Path=C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem;C:\Windows\System32\WindowsPowerShell\v1.0\;C:\Users\Ton1\AppData\Local\Microsoft\WindowsApps
PATHEXT=.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC

C:\TEMP>set PATH=%PATH%;C:\Program Files (x86)\Windows Kits\10\bin\10.0.16299.0\x86

C:\TEMP>set PATH
Path=C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem;C:\Windows\System32\WindowsPowerShell\v1.0\;C:\Users\Ton1\AppData\Local\Microsoft\WindowsApps;C:\Program Files (x86)\Windows Kits\10\bin\10.0.16299.0\x86
PATHEXT=.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC

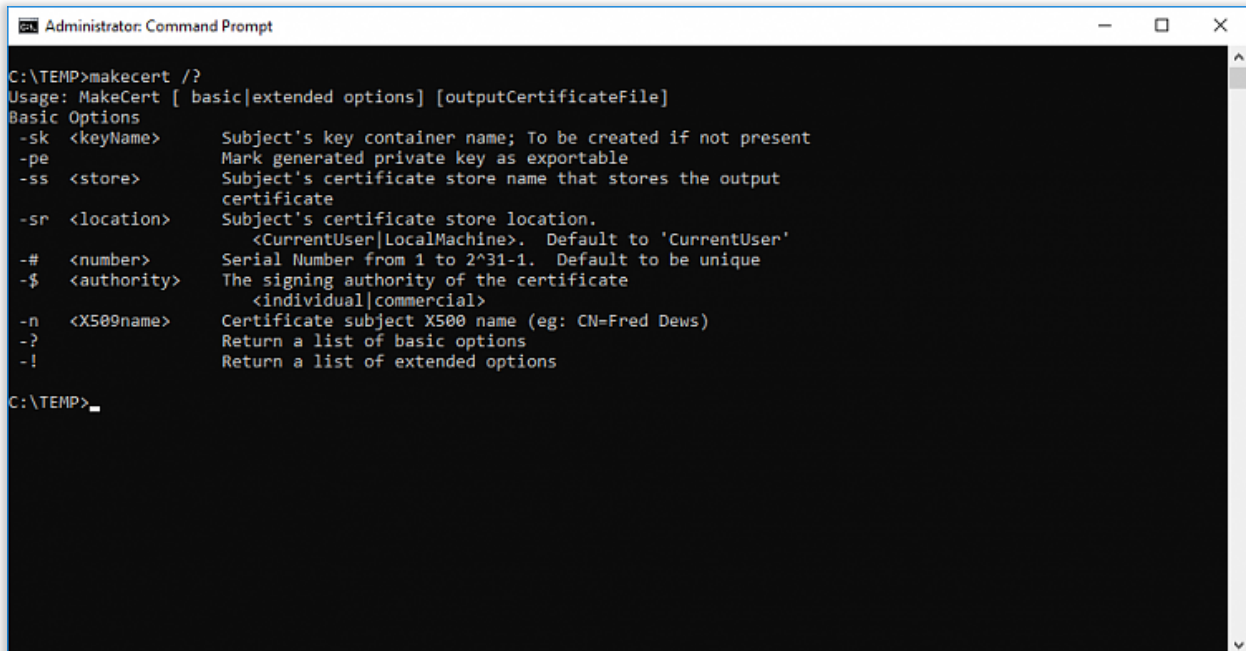
C:\TEMP>
```

You can also make the path stick permanently by using setx instead of set.

6. Verify that makecert.exe works

Code:

```
makecert /?
```



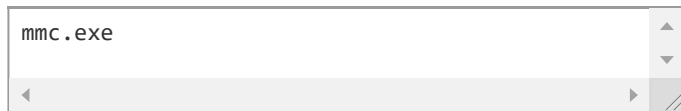
```
Administrator: Command Prompt

C:\TEMP>makecert /?
Usage: MakeCert [ basic|extended options] [outputCertificateFile]
Basic Options
-sk <keyName>      Subject's key container name; To be created if not present
-pe              Mark generated private key as exportable
-ss <store>       Subject's certificate store name that stores the output
                  certificate
-sr <location>    Subject's certificate store location.
                  <CurrentUser|LocalMachine>. Default to 'CurrentUser'
-# <number>      Serial Number from 1 to 2^31-1. Default to be unique
-$ <authority>   The signing authority of the certificate
                  <individual|commercial>
-n <X509name>    Certificate subject X509 name (eg: CN=Fred Dews)
-?              Return a list of basic options
-!              Return a list of extended options

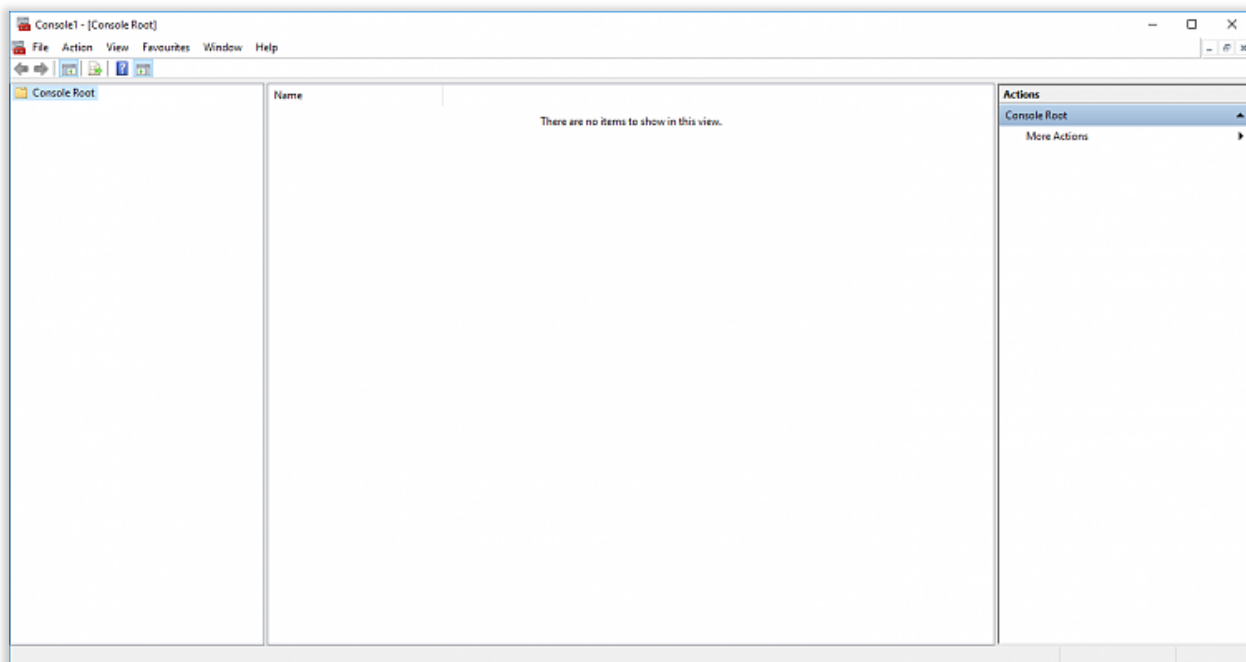
C:\TEMP>
```

7. Open [Microsoft Management Console](#).

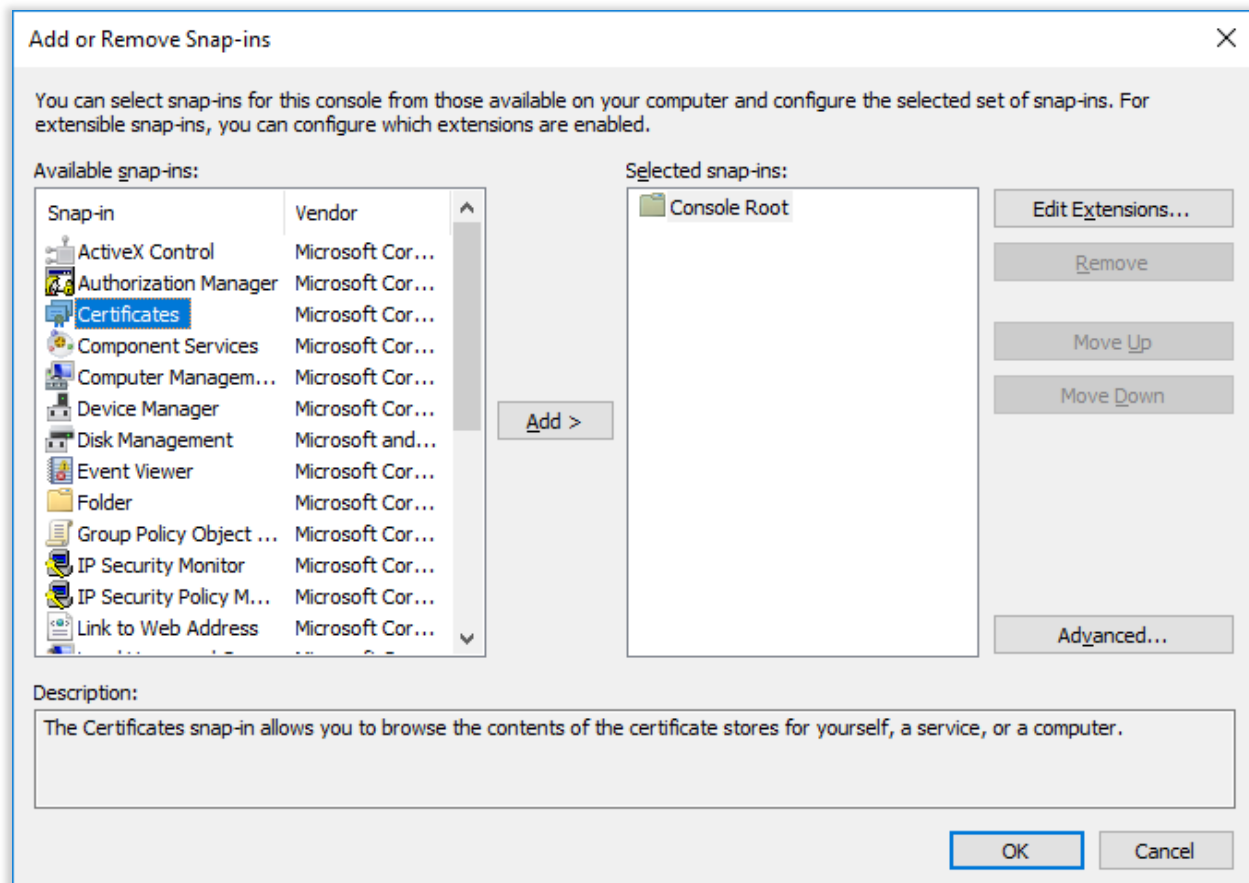
Code:

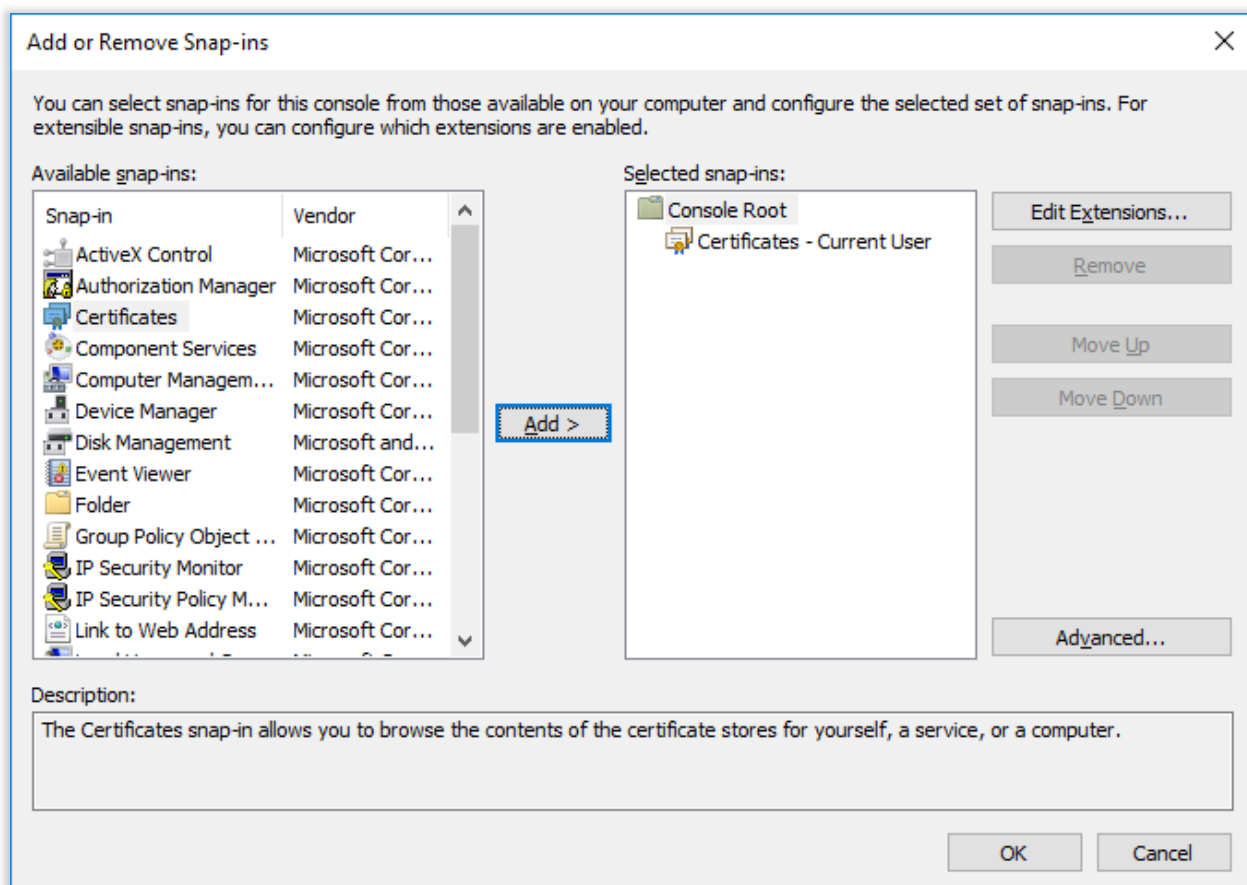
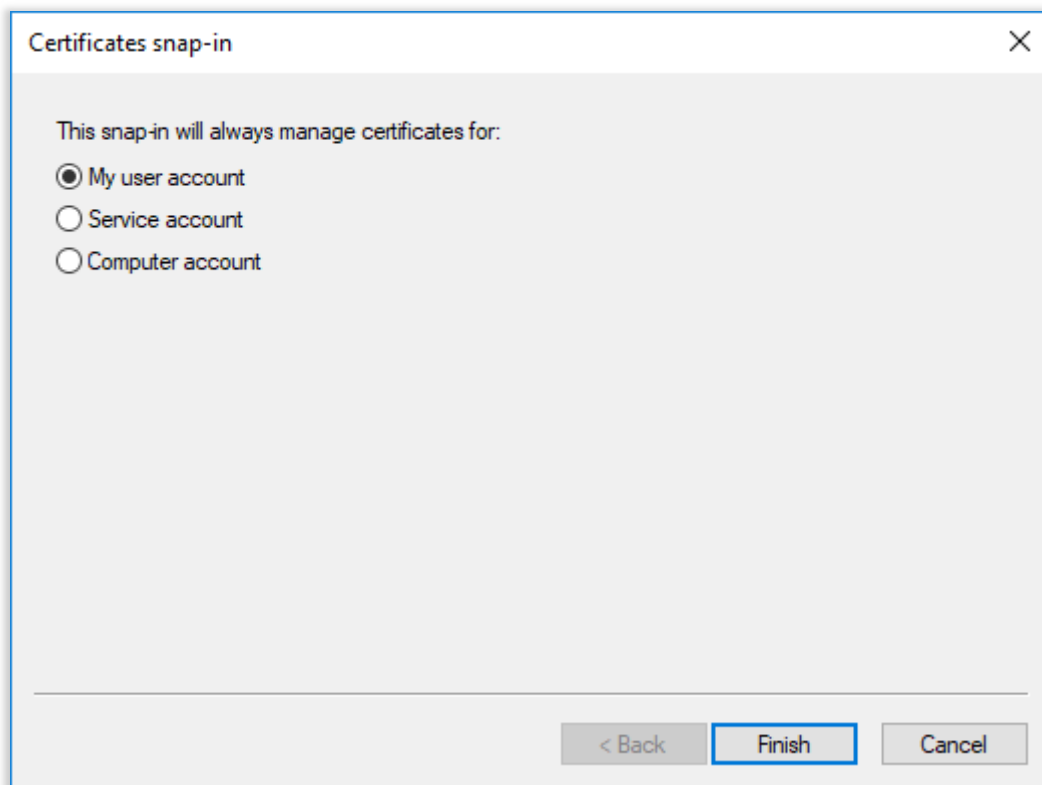


```
mmc.exe
```

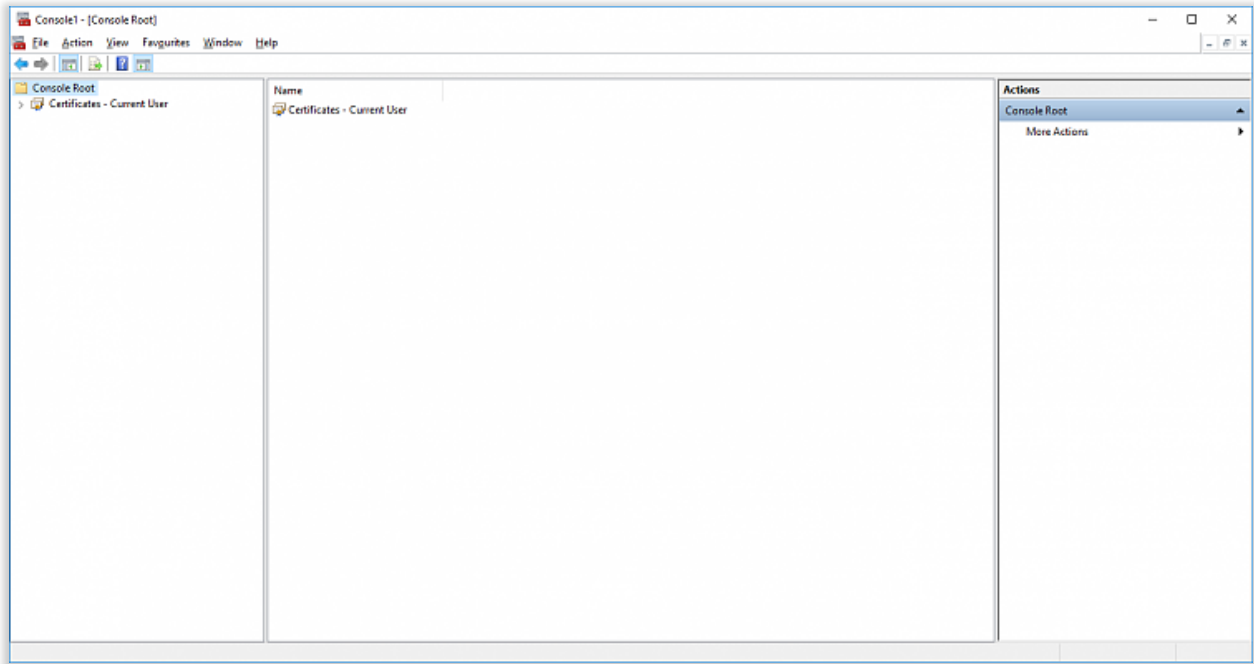


8. Click Add/Remove Certificate Snap-in from File-menu





9. Now your MMC should look like this

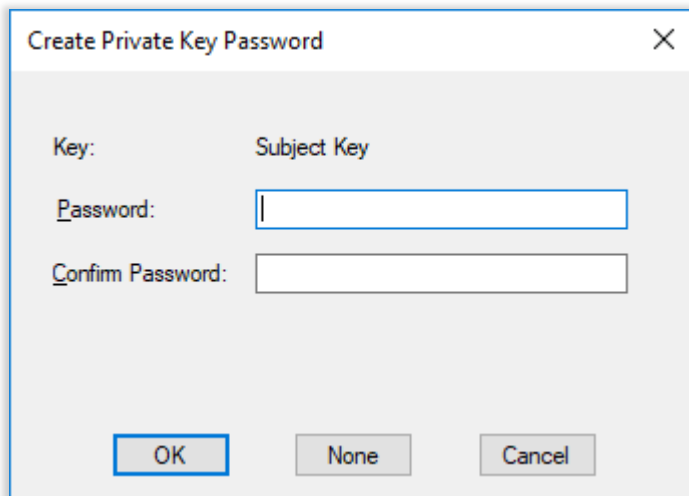


10. Create a Certificate Authority using CMD

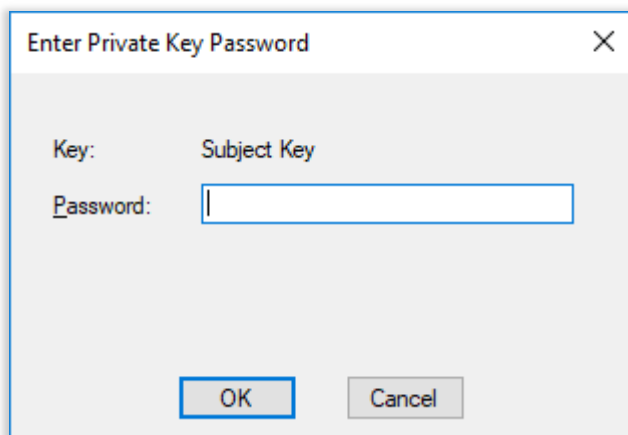
Code:

```
makecert -n "CN=PowerShell Local Certificate Roc"
```

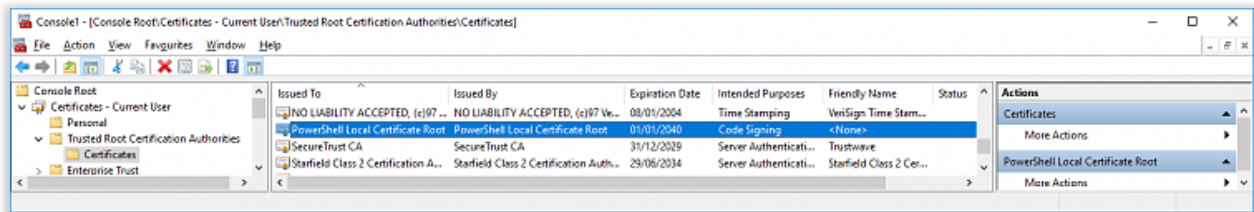
11. Give it a password



12. Verify your password



13. Now your MMC looks like this (may have to refresh by using F5-key)

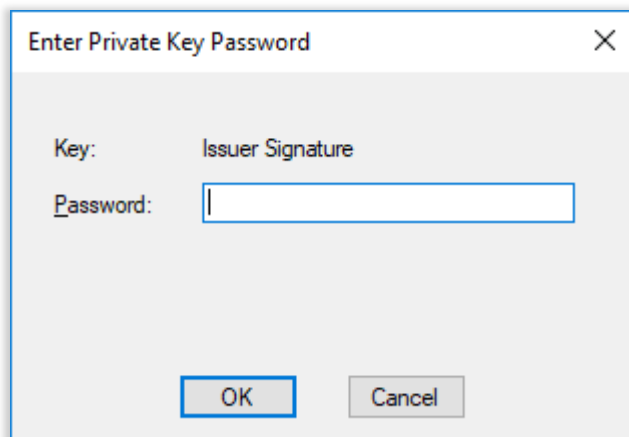


14. Create Personal Certificate

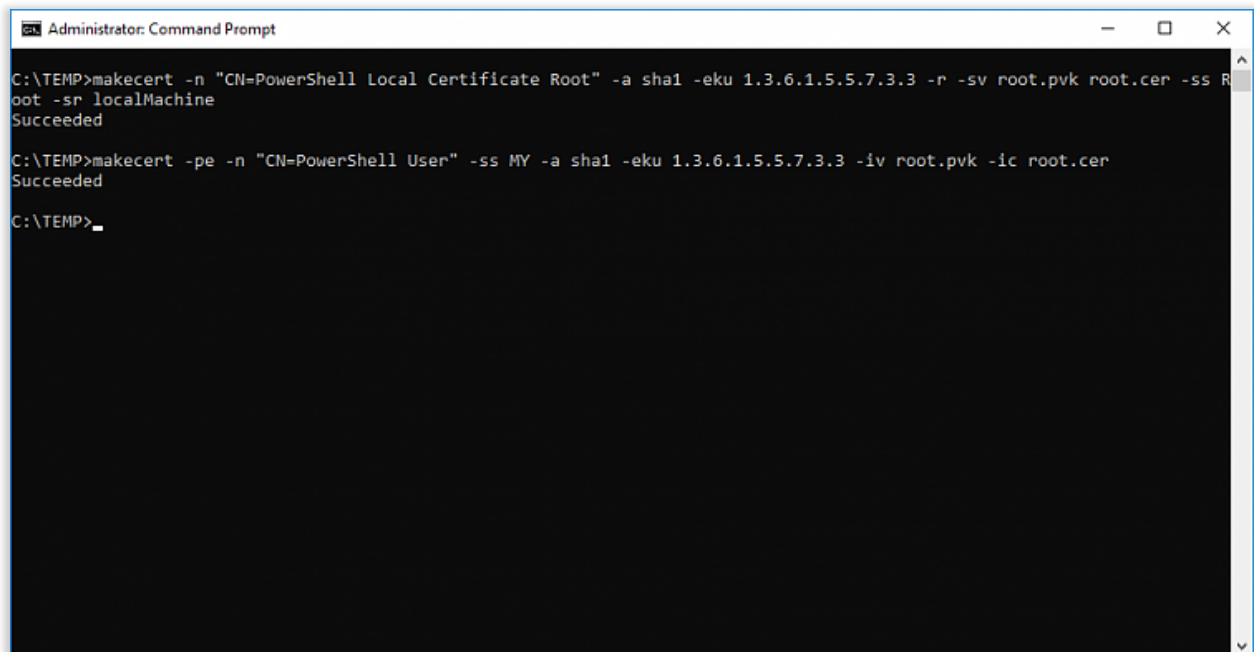
Code:

```
makecert -pe -n "CN=PowerShell User" -ss MY -a sha1 -eku 1.3.6.1.5.5.7.3.3 -iv root.pvk -ic root.cer
```

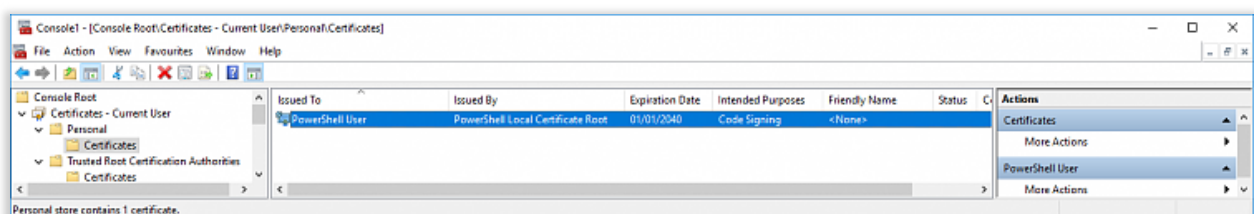
15. Enter your password



16. View of your console after entering the 2 commands



17. Now your MMC looks like this (may have to refresh by using F5-key)



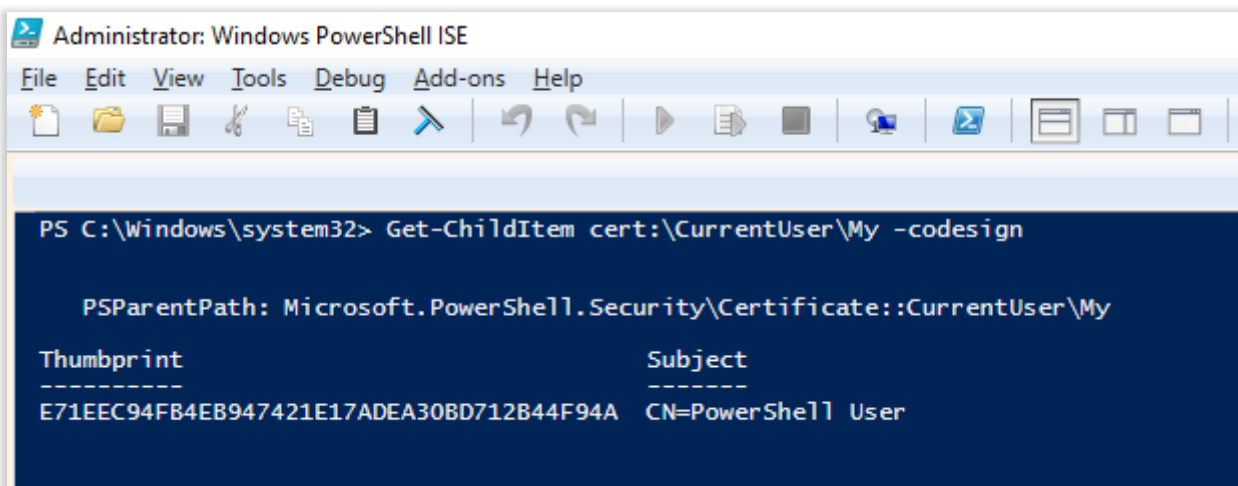
Verifying that we can sign PS scripts

Steps to take:

1. Open elevated PowerShell ISE
2. Checking that we can access our certificate

Code:

```
Get-ChildItem cert:\CurrentUser\My -codesign
```

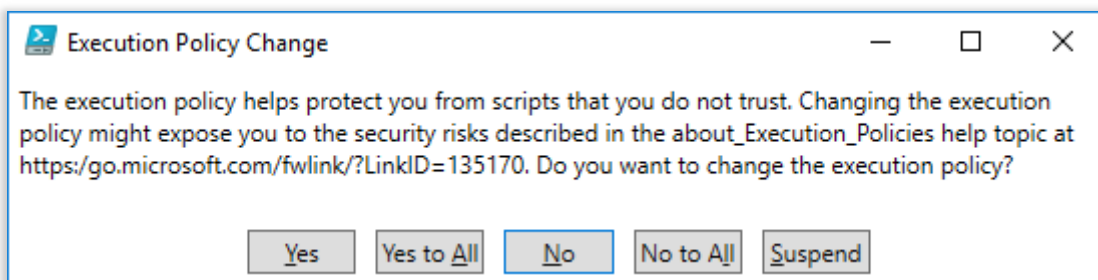


3. Changing execution policy to AllSigned

Code:

```
Set-ExecutionPolicy AllSigned
```

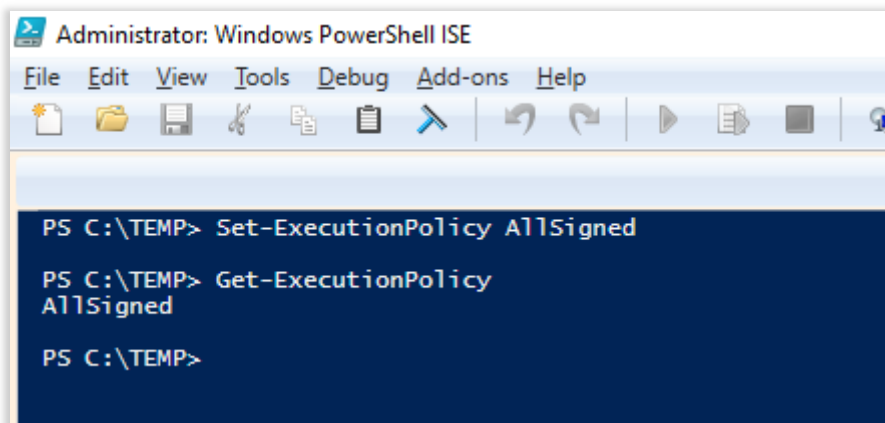
4. Answer Yes at the prompt



5. Verifying that everything got set properly

Code:

```
Get-ExecutionPolicy
```



6. Navigate to our working folder

Code:

```
cd \TEMP
```

7. Next we create a script that we will test, sign and test again

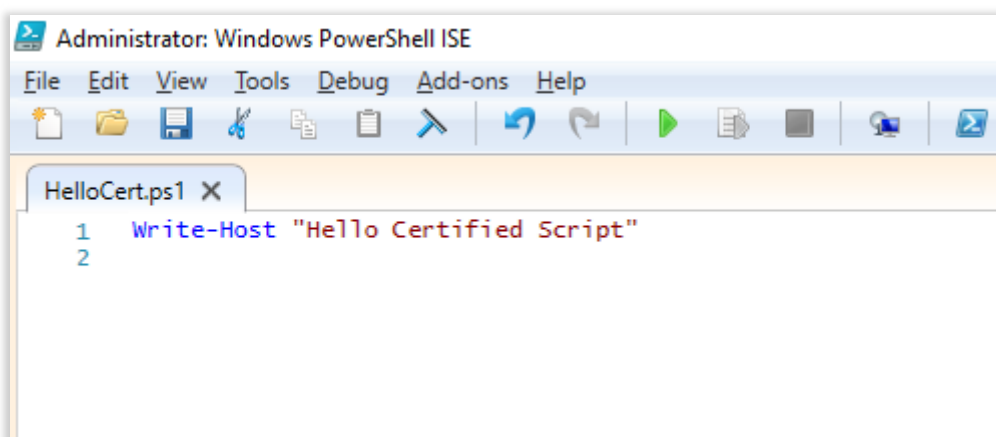
8. In PSISE select File>New

9. Enter the following code and press enter

Code:

```
Write-Host "Hello Certified Script"
```

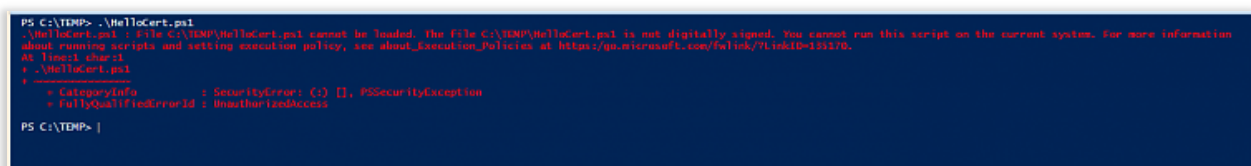
10. Save the script as HelloCert.ps1



11. Test the script (notice you will get an error)

Code:

```
'\HelloCert.ps1
```



12. Sign the script

Code:

```
Set-AuthenticodeSignature HelloCert.ps1 @(Get-Cl
```

```
PS C:\TEMP> Set-AuthenticodeSignature HelloCert.ps1 @(Get-ChildItem cert:\CurrentUser\My -codesign)[0]

Directory: C:\TEMP

SignerCertificate                Status                Path
-----
E71EEC94FB4EB947421E17ADEA30BD712B44F94A Valid                HelloCert.ps1

PS C:\TEMP>
```

13. Close the current HelloCert.ps1 you were earlier editing and open the file again (note the difference)

```
Administrator: Windows PowerShell ISE
File Edit View Tools Debug Add-ons Help

HelloCert.ps1 X
1 Write-Host "Hello Certified Script"
2
3 # SIG # Begin signature block
4 # MIIFuYJYKZIHvcNAQcCoIIFqjCCBaYCAQExCzAJBgUrDgMCGGUAMGkGCisGAQQB
5 # gjcCAQSGwzBZMDQGCisGAQQBggjCAR4wJgIDAQAABBAfzDtgWUsITrck0sYpfvNR
6 # AgEAAgEAAgEAAgEAAgEAMCEwCQYFKw4DAhoFAAQUotLR4y240uKK3MAmTiumOu+D
7 # JE2gggNCMIIDPjCCAiqgAwIBAgIQ6K94DUHgiB5EH9bT2az2bDAJBgUrDgMCHQUA
8 # MCwxKjAoBgNVBAMTIVBvd2VyU2h1bGwgTG9jYVwvQ2VydG1maWwNhdGUgUm9vdDAe
9 # Fw0xODA0MTEwNzEyMjBaFw0zOTEyMzU5NTI1aWBoxGDAwBgNVBAMTD1Bvd2Vy
10 # U2h1bGwgVXN1c2VydG1maWwNhdGUgUm9vdDQGCisGAQQBggjCAR4wJgIDAQAABBAfzDtgWUsITrck0sYpfvNR
11 # HL5KwHsbh2a/8316+Uw61vr7ngBzUiaBhwA92gEWq1zfHn84e/389SAr4mqbZ1i
12 # nIb51DKa4f4DP8WZ+XJ3m+rPgSEFWbPKnrg8EFWtH8QYkhhZLddrCKcXb3yniDc
13 # 5vtjafWlwzNQgtngSr2+qoEBre8t275M3Ri5jGbdHa4oAiaEmc1/Jy2B3/FW17Xg
14 # Pio2I2WhSOK01Nx1N8sP7LYQ2LDICMZbbrwQXsONjbQaONAooiZho0L6pXXDWcW/
15 # 5gnyaoCefnvWw/uUSXNCzIzBx27JUVqxzHJqcryRhsFQ6VPeITHTZ16YEkhqfCn
16 # t7vpf2nXDrGBAUECAwEAAaNMHqWewYDVR01BAwwCgYIKwYBBQUHAWMwXQYDVROB
17 # BFYwVIAQdNcR6z0wy6HCYT20Xh21/aEuMCwxKjAoBgNVBAMTIVBvd2VyU2h1bGwg
18 # TG9jYVwvQ2VydG1maWwNhdGUgUm9vdDQGCisGAQQBggjCAR4wJgIDAQAABBAfzDtgWUsITrck0sYpfvNR
19 # HQUAA4IBAQAIIY/GuDbQbrtvwBC+nL03t37ae5PoGTPUqvCRNqtYE3GPFxDeoBrco
20 # udrCoU70MilyiuD8I60AdLC/8qrH+G2JGe60c1pXYrLgZ0aIegcGJemXc+whik+o
21 # eCyydVF52VzUsBa8WpOGmMzXSR31IUQJNSQk60vzoFk5qIr80ND2YDP2V60bdvq
22 # SEtVumQfTtyLpg4BJoHKCw/pZM/U9C4W7yP05I5wvPNHZjsyhIVT3mto9Yy9W5T
23 # Colno570P+E/FmtuNVvgpvm+a7pgASZrvitkEre1+12VKL+JgeFcrsX7IOA8fanl
24 # ubDtRquX977H14pwRN8jU0j67Q61gyUzMYIB4TCCAd0CAQEWQDAsMSowKAYDVQQD
25 # EyFQb3dlc1NoZWxsIEExvY2FsIEN1cnRpdjYXR1IFJvb3QCE0iveA1B4Im+RB/W
26 # 09ms9mwwCQYFKw4DAhoFAKB4MBGcGCisGAQQBggjCAR4wJgIDAQAABBAfzDtgWUsITrck0sYpfvNR
27 # KoZIHvcNAQkDMQwGCisGAQQBggjCAR4wJgIDAQAABBAfzDtgWUsITrck0sYpfvNR
28 # gjcCARUwIwYJYKZIHvcNAQkEMRYEFBFGiwVwFE9E2q1DCydwasftBeBnMA0GCSqG
29 # SIb3DQEBAQUABIIBAAkVhnHEHm41jAuce/KErpe6h52K+GikFC/B796T1EsdhmH
30 # kXDH5wdJ51ITZey4mIzhUpeawYMcFUSXDik0EGiUURccHN81aLsCFR10NTYDQwpg
31 # vJGkH01qUSGZnt8xvKm//On0wN1L6jPzCMAQQfjYq3Jpkeb05+4JF2jDYpB9nc5R
32 # W0z4qZbsv7HFDtqIJ0ZjCzoVtCQng933be2myZzvSRz7vsWULEE7xXCy0vcjUqJa
33 # VLidRwQV43yNMMpNy3XxN8s1FA4kmp1k529XmQoJKxOWes/a31mvVuLwPQ4sx0g
34 # 24Vur2dy9wKC+5/wZ+c55+Kfy5zVLxrHRvNMO2s=
35 # SIG # End signature block
36
```

14. Run the script

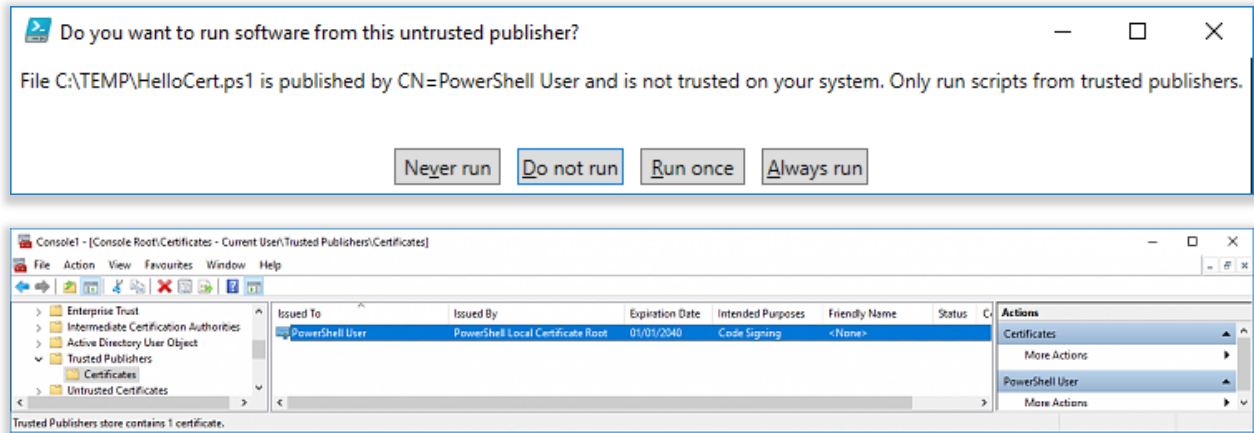
Code:

```
.\HelloCert.ps1
```

```
PS C:\TEMP> .\HelloCert.ps1
Hello Certified Script

PS C:\TEMP>
```

15. Press Always run to add the certificate to Trusted Publishers



16. All should work now as expected and you have maximum security enabled that still allows you to run scripts.

CONGRATULATIONS! You are now officially a PS-script signing guru!