



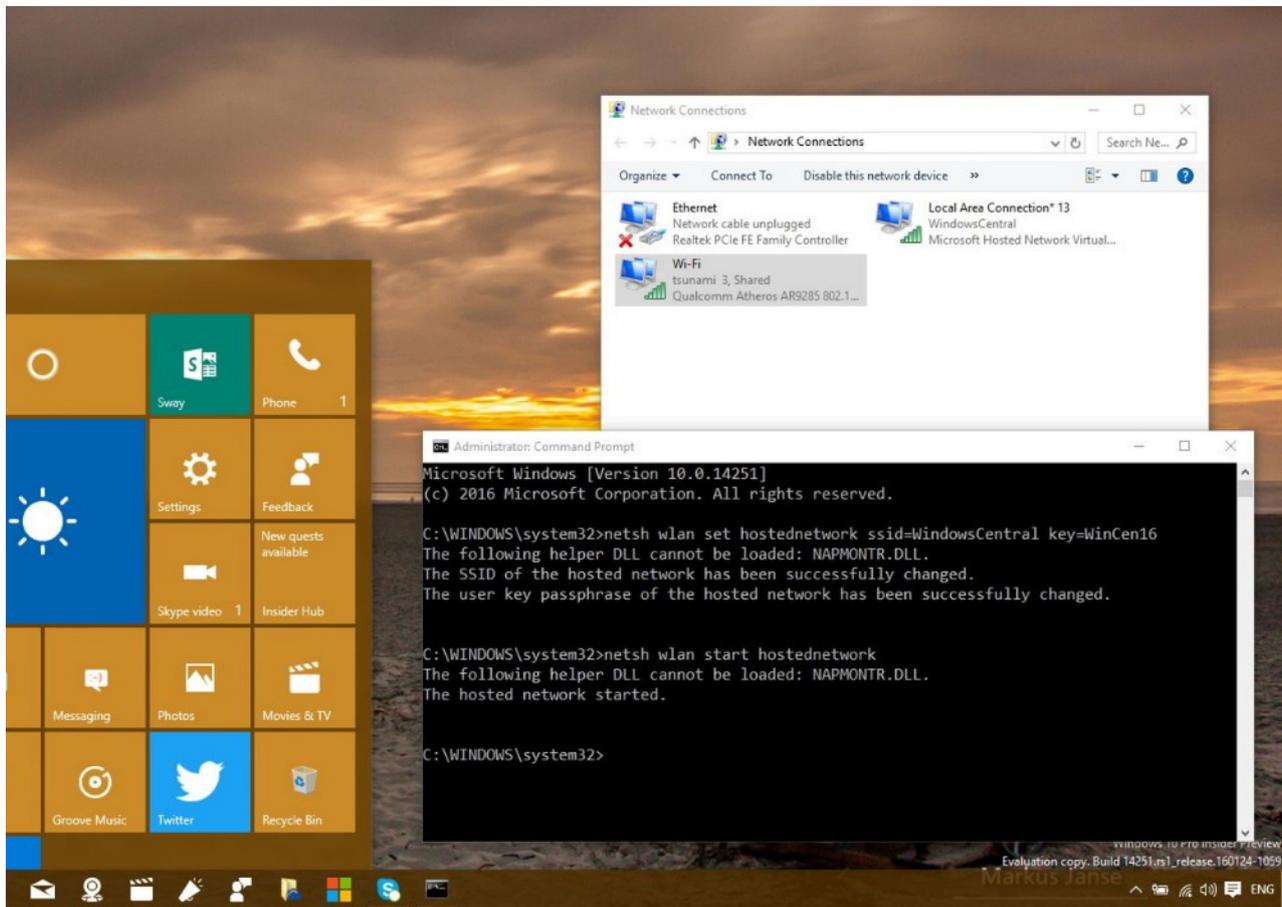
Part 31

-

HotSpot

How to turn your Windows 10 PC into a wireless hotspot

Windows 10 includes a feature called "Hosted Network" that allows you to turn your computer into a wireless hotspot, and this guide we'll show you how to do it.



Whether you're connecting to the internet using a wireless or wired adapter, similar to previous versions, Windows 10 allows you to share an internet connection with other devices with a feature called "Hosted Network". Hosted Network is a feature that comes included with the `netsh` (Network Shell) command-line utility. It's was previously introduced in Windows 7, and it allows you to use the operating system to create a virtual wireless adapter – something that Microsoft refers to "Virtual Wi-Fi" — and create a SoftAP, which is a software-based wireless access point.

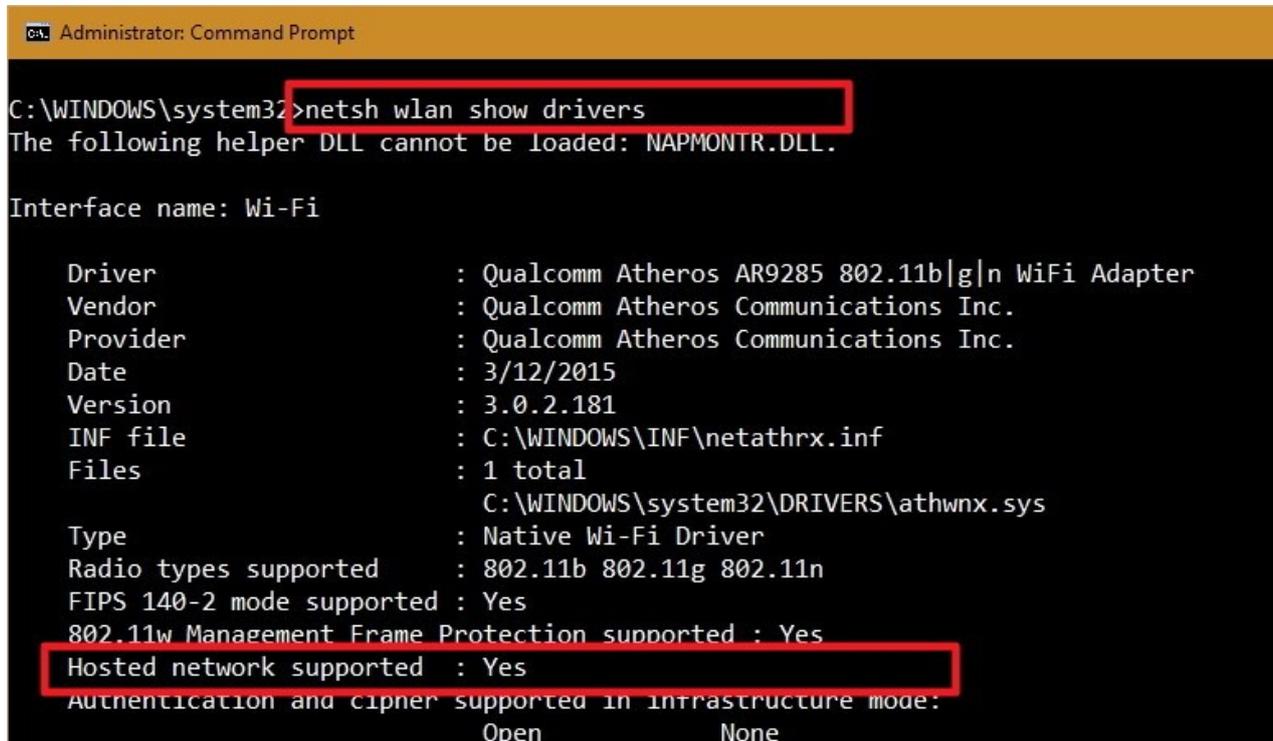
Through the combination of these two elements, your PC can take its internet connection (be it an ethernet connection or hookup through a cellular adapter) and share it with other wireless devices — essentially acting as a wireless hotspot.

You'll need to open the Command Prompt with administrator rights. To do this, use the **Windows key + X** keyboard shortcut, and select **Command Prompt (Run As Administrator)**.

How to check if your wireless adapter supports Hosted Networks

While some adapters include support for Hosted Network, you will first need to verify your computer's physical wireless adapter supports this feature using the following command:

```
NETSH WLAN show drivers
```



```
Administrator: Command Prompt
C:\WINDOWS\system32>netsh wlan show drivers
The following helper DLL cannot be loaded: NAPMONTR.DLL.

Interface name: Wi-Fi

Driver                : Qualcomm Atheros AR9285 802.11b|g|n WiFi Adapter
Vendor                : Qualcomm Atheros Communications Inc.
Provider              : Qualcomm Atheros Communications Inc.
Date                  : 3/12/2015
Version               : 3.0.2.181
INF file              : C:\WINDOWS\INF\netathrx.inf
Files                 : 1 total
                       C:\WINDOWS\system32\DRIVERS\athwnx.sys
Type                  : Native Wi-Fi Driver
Radio types supported : 802.11b 802.11g 802.11n
FIPS 140-2 mode supported : Yes
802.11w Management Frame Protection supported : Yes
Hosted network supported : Yes
Authentication and cipher supported in infrastructure mode:
Open                  None
```

If the generated output shows **Hosted network supported: Yes**, then you can continue with the guide. If your wireless adapter isn't supported, you could try using a USB wireless adapter that supports the feature.

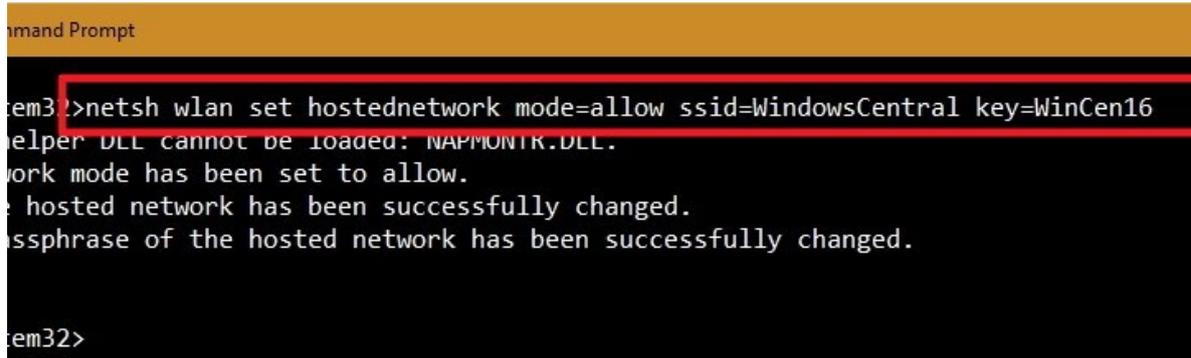
How to create a wireless Hosted Network

Creating a wireless hotspot in Windows 10 is relatively straightforward. Simply follow the steps below to configure a wireless Hosted Network:

1. While in Command Prompt (Admin) enter the following command:

```
NETSH WLAN set hostednetwork mode=allow ssid=Your_SSID key=Your_Passphrase
```

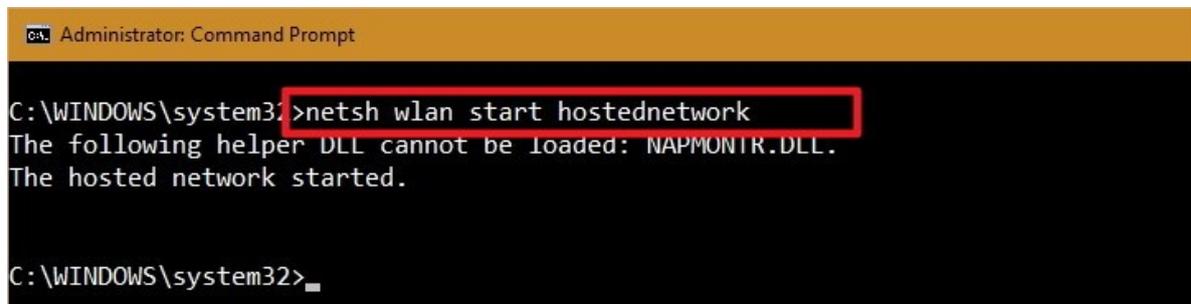
Where the **SSID** would be the name you want to identify your wireless network when trying to connect a new device, and the passphrase is the network security key you want users to use to connect to your network. Remember that the passphrase has to be at least 8 characters in length.



```
Command Prompt
C:\WINDOWS\system32>netsh wlan set hostednetwork mode=allow ssid=WindowsCentral key=WinCen16
The following helper DLL cannot be loaded: NAPMONTR.DLL.
The hosted network mode has been set to allow.
The hosted network has been successfully changed.
The passphrase of the hosted network has been successfully changed.
C:\WINDOWS\system32>
```

2. Once you created a Hosted Network, enter the following command to activate it:

```
NETSH WLAN start hostednetwork
```

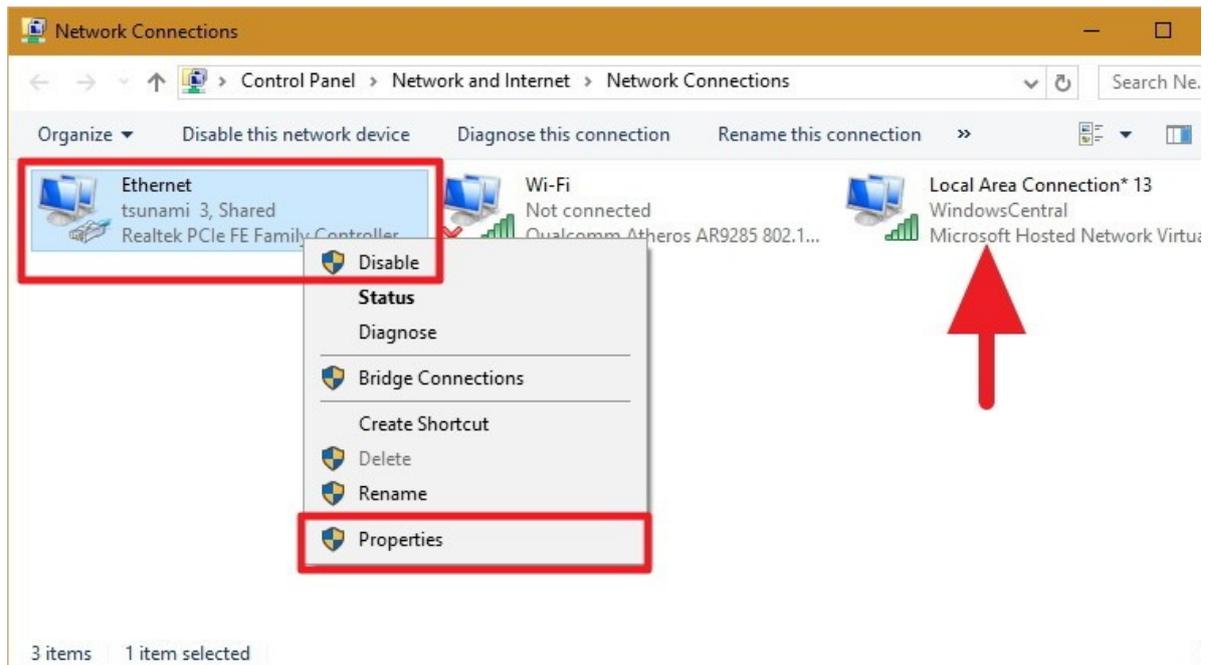


```
Administrator: Command Prompt
C:\WINDOWS\system32>netsh wlan start hostednetwork
The following helper DLL cannot be loaded: NAPMONTR.DLL.
The hosted network started.
C:\WINDOWS\system32>
```

How to share your internet connection with a Hosted Network

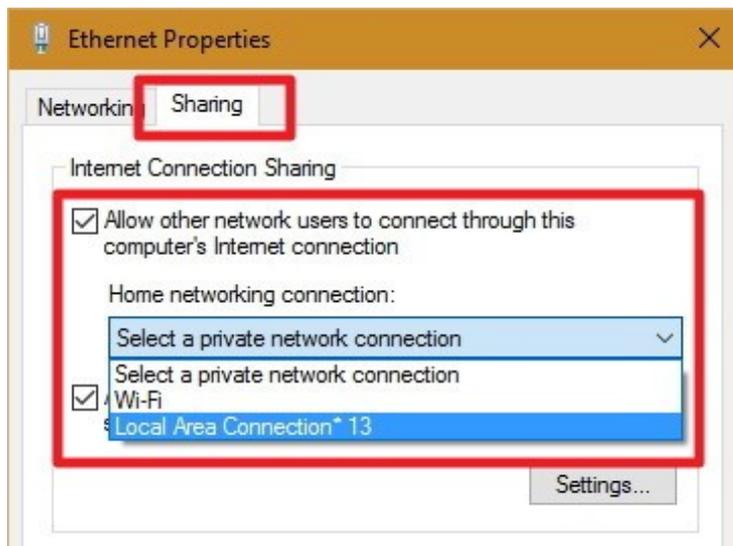
Up to here, you created and started a Hosted Network in your Windows 10 PC. However, any wireless capable device won't be able to access the internet just yet. The last thing you need to do is to share an internet connection using the "Internet Connection Sharing" feature from a physical network adapter.

1. Use the **Windows key + X** keyboard shortcut to open the Power User menu, and select **Network Connections**.
2. Next, right-click the network adapter with an internet connection – this could be a traditional Ethernet or wireless network adapter – select **Properties**.



Note: In Network Connections, you should now see a new our new **Microsoft Hosted Virtual Adapter** which is labeled **Local Area Connection* X**, and with the SSID name.

3. Click the **Sharing** tab.
4. Check the **Allow other network users to connect through this computer's Internet connection** option.
5. Next, from the **Home networking connection** drop-down menu select the **Microsoft Hosted Virtual Adapter**.



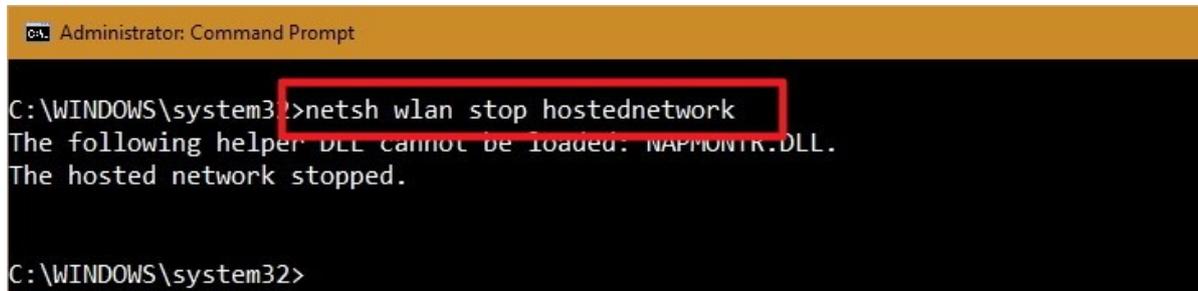
6. Click **OK** to finish.

At this point, you should be able to see and connect any wireless capable device to the newly created software access point, and with access to the internet.

How to stop sharing an internet connection with other devices

If you want to temporarily stop allowing other devices to connect wirelessly through your computer, you can type the following command in the Command Prompt and hit **Enter**:

```
NETSH WLAN stop hostednetwork
```

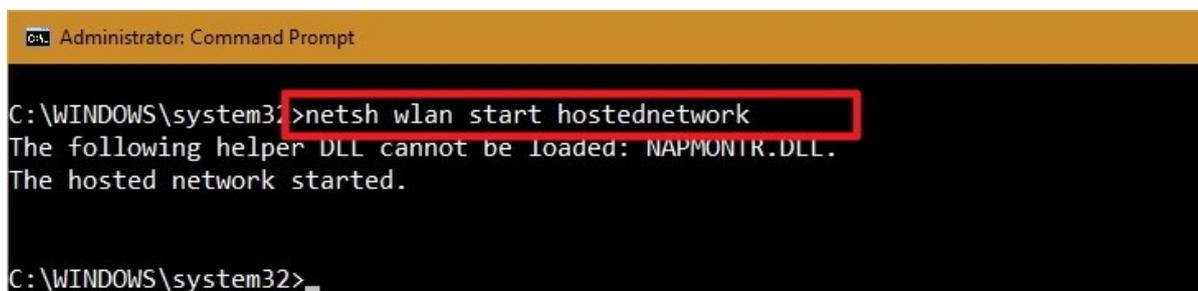


```
Administrator: Command Prompt
C:\WINDOWS\system32>netsh wlan stop hostednetwork
The following helper DLL cannot be loaded: NAPMONTR.DLL.
The hosted network stopped.

C:\WINDOWS\system32>
```

At any time, you can just use the **start** variant of the command to allow other devices to connect to the internet using your computer as an access point without extra configuration:

```
NETSH WLAN start hostednetwork
```



```
Administrator: Command Prompt
C:\WINDOWS\system32>netsh wlan start hostednetwork
The following helper DLL cannot be loaded: NAPMONTR.DLL.
The hosted network started.

C:\WINDOWS\system32>
```

Similarly, you can also use the following command to enable or disable a wireless Hosted Network:

```
NETSH WLAN set hostednetwork mode=allow
NETSH WLAN set hostednetwork mode= disallow
```

How to change a Hosted Network settings

In the case you want to change some of the current settings, such as SSID or network security you can use the following commands:

```
NETSH WLAN set hostednetwork ssid=Your_New_SSID
NETSH WLAN set hostednetwork key=Your_New_Passphrase
```

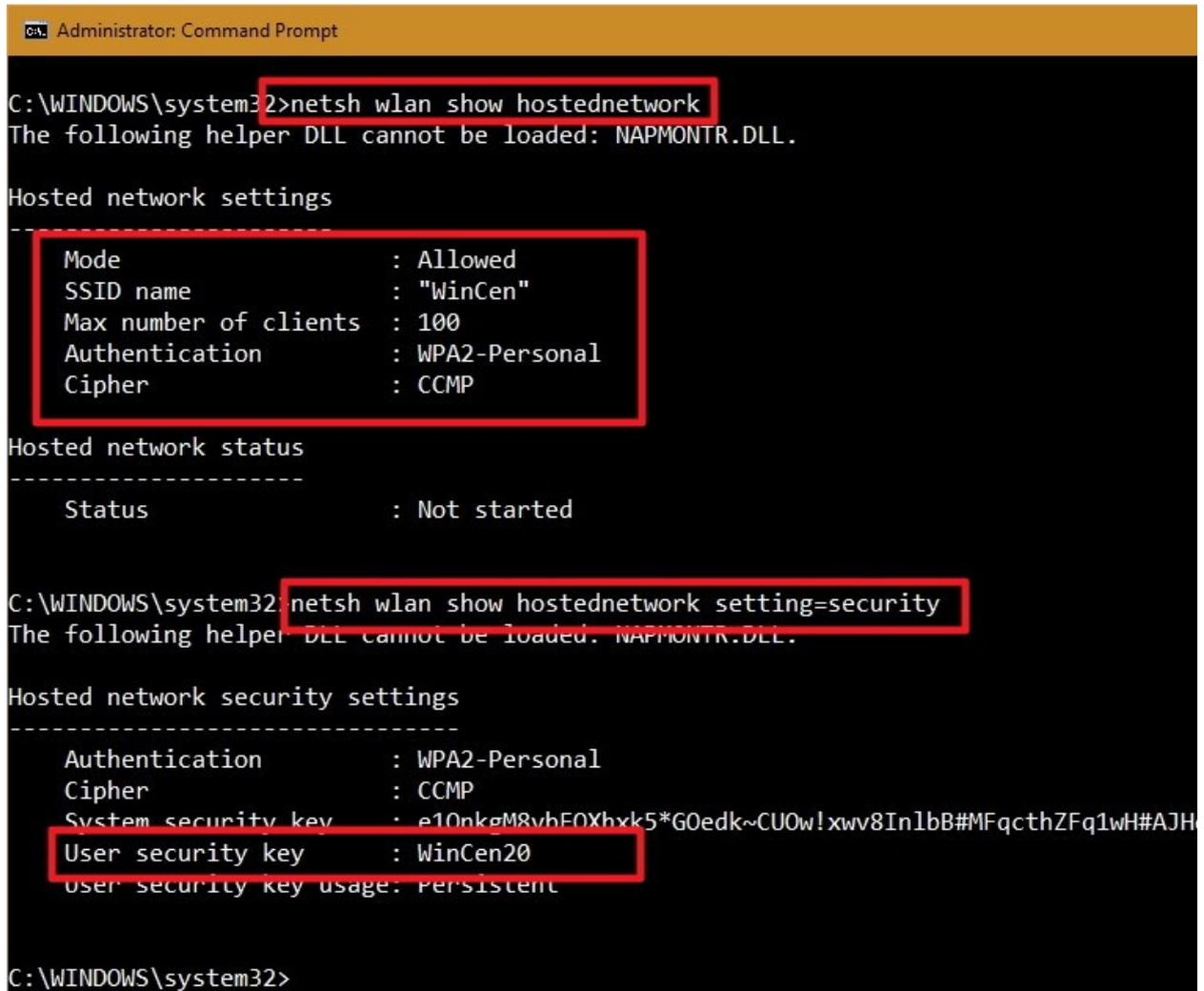
How to view the current Hosted Network settings

There are two commands to view the Hosted Network settings on your computer:
The following command shows the mode and SSID name in use, max number of clients that can connect, type of authentication, and cipher:

```
NETSH WLAN show hostednetwork
```

And the following command will also reveal the current network security key among other settings, similar to the previous command:

```
NETSH WLAN show hostednetwork setting=security
```



```
Administrator: Command Prompt
C:\WINDOWS\system32>netsh wlan show hostednetwork
The following helper DLL cannot be loaded: NAPMONTR.DLL.

Hosted network settings
-----
Mode                : Allowed
SSID name           : "WinCen"
Max number of clients : 100
Authentication      : WPA2-Personal
Cipher              : CCMP

Hosted network status
-----
Status              : Not started

C:\WINDOWS\system32>netsh wlan show hostednetwork setting=security
The following helper DLL cannot be loaded: NAPMONTR.DLL.

Hosted network security settings
-----
Authentication      : WPA2-Personal
Cipher              : CCMP
System security key : e10nkgM8vbFOXhvk5*G0edk~CU0w!xwv8In1bB#MFqcthZFq1wH#AJH
User security key   : WinCen20
User security key usage: Persistent

C:\WINDOWS\system32>
```

How to disable a wireless Hosted Network

While the setup of a wireless Hosted Network in Windows 10 is not very complicated, Microsoft doesn't make it very straightforward to remove the configurations when you no longer need the feature.

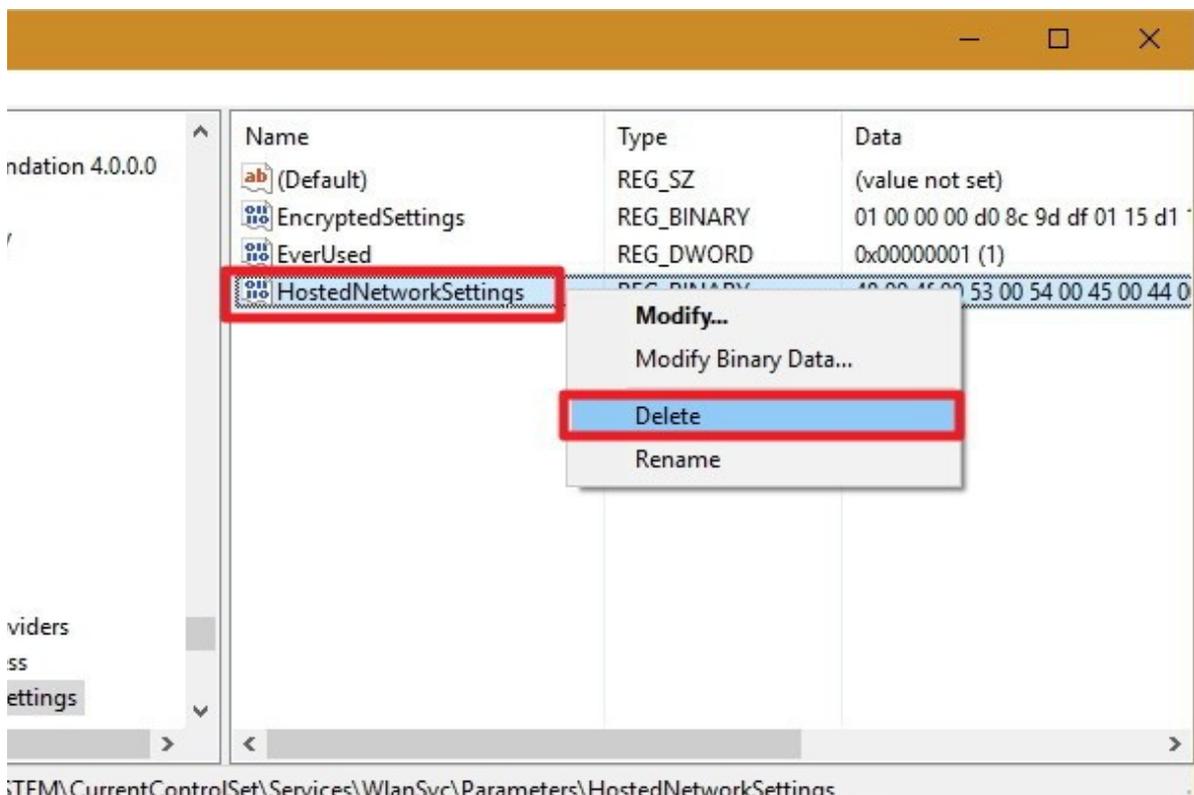
Although you can use the stop or disallow commands, these actions won't eliminate the settings from your computer. If you want to completely delete the Hosted Network settings in Windows 10, you'll need to modify the Registry.

1. Open the Start menu, do a search for **regedit**, hit **Enter**, and click **OK** to open the Registry with admin rights.

2. Scroll down the following path in the Registry:

HKEY_LOCAL_MACHINE\system\currentcontrolset\Services\WlanSvc\Parameters\HostedNetworkSettings

Right-click the **HostedNetworkSettings** DWORD key, select **Delete**, and click **Yes** to confirm deletion.



3. Restart your computer

4. Open to the Command Prompt and use the following command:

```
NETSH WLAN show hostednetwork
```

You will know that you have successfully deleted the settings when the **Settings** field reads **Not configured**.

```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.14251]
(c) 2016 Microsoft Corporation. All rights reserved.

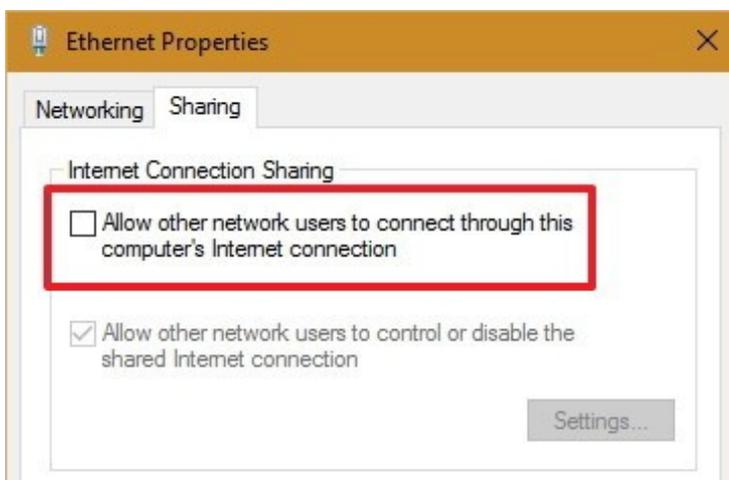
C:\WINDOWS\system32>netsh wlan show hostednetwork
The following helper DLL cannot be loaded: NAPMONTR.DLL.

Hosted network settings
-----
Mode                : Allowed
Settings            : <Not configured>

Hosted network status
-----
Status              : Not started

C:\WINDOWS\system32>
```

5. Make sure you turn off "Internet Connection Sharing" in the physical network adapter that was sharing the internet with other devices. Use the **Windows key + X** keyboard shortcut to open the Power User menu, and select **Network Connections**.
6. Right-click the network adapter, and select **Properties**.
7. Click the **Sharing** tab.
8. Uncheck the **Allow other network users to connect through this computer's Internet connection** option.



9. Click **OK** to complete the process.

Things you need to know

Although the wireless Hosted Network feature in Windows 10 allows you to implement an access point solution to share an internet connection with other devices, it's not meant to be a solution to replace a physical wireless access point.

Also, there are a few things you want to consider. For example, wireless speeds will dramatically be reduced compared to the rates provided from a physical access point. Perhaps it would not be a big deal for internet browsing, but downloading or transferring big files could be an issue for some users.

You also need to consider that your computer needs to be always turned on to act as a wireless access point. If the computer enters into sleep, hibernate, or restarts, your wireless hotspot will stop working, and you will need to start manually the feature using the

```
NETSH WLAN start hostednetwork command.
```

You cannot run a SoftAP and Ad Hoc at the same time on Windows. If you need to create a temporary network connection between two computers, setting up Ad Hoc will turn off SoftAP — you can run one or the other, not both at the same time.

Setting up a hotspot to access a wireless standalone Raspberry Pi

First check if your WiFi adapter on your laptop supports hosted networks

```
netsh wlan show drivers
```

If the generated output shows

```
Hosted network supported: Yes
```

then you can continue with the guide. If your wireless adapter isn't supported, you could try using a USB wireless adapter that supports the feature.

If multiple WiFi adapters are present, disable the ones that you do not want to be used (in my case: "Wi-Fi")

```
netsh interface set interface name="Wi-Fi" admin=disabled
```

1. create and start a WiFi hotspot with SSID and security key (needs to be at least 8 characters):

```
netsh wlan set hostednetwork mode=allow ssid=AP-MePi key=12345678
netsh wlan start hostednetwork
```

2. Assign a name to the hotspot in the "Network and sharing center > change adapter settings" (in my case I also call it "AP-MePi")

3. Set a static IP address for the WiFi network:

```
netsh interface ip set address "AP-MePi" static 192.168.159.1 255.255.255.0
196.168.159.1
```

4. Now, using a program that provides a DHCP server on Windows like DHCPServer from <http://www.dhcpserver.de/cms/>, I configured the DHCP server like this

```
[SETTINGS]
IPPOOL_1=192.168.159.2-254
IPBIND_1=192.168.159.1
```

5. If needed, you can re-enable the disabled WiFi adapters

```
netsh interface set interface name="myWiFi" admin=enabled
```

6. To stop the hosted network

```
netsh wlan stop hostednetwork
```

Raspbian WiFi Setup

Raspbian, since May 2016, checks the contents of the boot partition for a file called `wpa_supplicant.conf`, and will copy the file into `/etc/wpa_supplicant`, replacing any existing `wpa_supplicant.conf` file that may be there. The file in the boot directory is then removed. This can be used to enable headless setup, using the `wpa_supplicant.conf` settings detailed below.

SSH can be enabled by placing a file named 'ssh', without any extension, onto the boot partition of the SD card.

What is my IP Address?

If you just want to know the IP Address your Pi is using enter `hostname -I` on the command line.

Predictable Network Interface Names

Stretch introduces predictable network interface names which have been used in other distributions for some time.

Network interfaces will have names formed from a prefix `en` — Ethernet or `wl` — wlan followed by `x` indicating MAC and the MAC e.g. `enxb827eb123456` or `wlx00c140123456`. The onboard WiFi of the Pi3 and PiZeroW which is connected over `sdio` will however use the name `wlan0`. This can be beneficial to those using multiple network interfaces, however for most Pi users, with a single Ethernet and WiFi interface will make little difference. The previous names `eth0` and `wlan0` can be restored if you pass `net.ifnames=0` on the kernel command line in `/boot/cmdline.txt`.

Networking Files

Note: If you are running a recent Raspbian `/etc/network/interfaces` should be as below. If you have changed it PUT IT BACK. On Stretch just delete it - it effectively does NOTHING.

```
# interfaces(5) file used by ifup(8) and ifdown(8)

# Please note that this file is written to be used with dhcpcd
# For static IP, consult /etc/dhcpcd.conf and 'man dhcpcd.conf'

# Include files from /etc/network/interfaces.d:
source-directory /etc/network/interfaces.d
```

Stretch includes the `10-wpa_supplicant` hook, so the links to `wpa_supplicant` are not needed. The older settings are incompatible with Predictable Network Interface Names.

The default `/etc/network/interfaces` used by **Jessie** should be:

```
# interfaces(5) file used by ifup(8) and ifdown(8)

# Please note that this file is written to be used with dhcpcd
# For static IP, consult /etc/dhcpcd.conf and 'man dhcpcd.conf'

# Include files from /etc/network/interfaces.d:
source-directory /etc/network/interfaces.d

auto lo
iface lo inet loopback

iface eth0 inet manual

allow-hotplug wlan0
iface wlan0 inet manual
    wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
```

```
allow-hotplug wlan1
iface wlan1 inet manual
    wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
```

The file `/etc/wpa_supplicant/wpa_supplicant.conf` will be created/modified by the recommended setup methods, but can be setup by hand. Recent Raspbian have an option in `raspi-config` to enter SSID and Password. `/etc/wpa_supplicant/wpa_supplicant.conf` should contain something like the following after entering your SSID and password:

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
country=AU

network={
    ssid="ESSID"
    psk="Your_wifi_password"
}
```

If you need to connect to a **private network** (*i.e. no broadcast SSID*) include the line `scan_ssid=1` inside `network={...}`.

Note: If you want to connect to different networks (e.g. at work or home) you can include multiple `network={...}` entries.

If you have access to multiple networks and want to connect to a specific network, include the line `priority=100` inside the `network={...}` entry.

You can select a different priority for each network, the highest priority accessible network will be selected; the default is 0.

There are many other options which can be used see `man wpa_supplicant.conf`.

Advanced dhcpcd Configuration

Raspbian, by default, uses `dhcpcd` to manage network interfaces. This is automatic, and most users need do no more than specify the SSID and password for wireless networks. It is possible to configure how `dhcpcd` works; see `man dhcpcd.conf`.

The following describe some of the more common configurations:

Setup a Static IP Address

Leave `/etc/network/interfaces` at its default (as above).

Edit `/etc/dhcpcd.conf` as follows:

```
interface eth0
static ip_address=10.1.1.30/24
static routers=10.1.1.1
static domain_name_servers=10.1.1.1

interface wlan0
static ip_address=10.1.1.31/24
static routers=10.1.1.1
static domain_name_servers=10.1.1.1
```

`ip_address` is the address and size from the command above (or another unused address on the same network), `routers` is the address of your router aka gateway. `domain_name_servers` is the DNS address(es) from `/etc/resolv.conf`. (see `man dhcpcd.conf`)

Fallback profile

It is possible to configure a static profile within `dhcpcd` and fallback to it when DHCP lease fails. This is useful particularly for headless machines, where the static profile can be used as "recovery" profile to ensure that it is always possible to connect to the machine. The static profile is setup as any other Static IP Address

```
# define static profile
profile static_eth0
static ip_address=...
static routers=...
static domain_name_servers=...

# fallback to static profile on eth0
interface eth0
fallback static_eth0
```

Prevent dhcpcd from configuring an interface

This is often done to enable the Pi to act as an Access Point (which needs to be configured using other files), while allowing normal DHCP configuration on other interfaces. Add `deny interfaces wlan0` to the end of the file (but above any other added interface lines).

Use different wpa_supplicant files

It is possible to configure `dhcpcd` to use different `wpa_supplicant.conf` files for a specific wireless interface. Create a file named `wpa_supplicant-"$interface".conf` in `/etc/wpa_supplicant/` e.g. `wpa_supplicant-wlan0.conf` will only be used by `wlan0` `/etc/wpa_supplicant/wpa_supplicant.conf` will be used for any other wireless interfaces.

Connecting a Computer to the Pi

Recent versions of Raspbian (which use `dhcpcd`) allow `ssh` to work over a link-local address and `avahi` (which is a zeroconf implementation) enables programs to discover hosts running on a local network.

This means you can plug the Pi into a Computer (with an Ethernet cable) or a local network router and connect without knowing the IP address.

You can easily connect from Linux and OS X with `ssh pi@hostname.local` (the default hostname is `raspberrypi`) This should work with popular GUI `ssh` programs. This is sometimes problematic with some versions of Windows and networks which use `.local` in a non-standard way.

One drawback of direct connection is that the Pi will have no Internet access and the date will not be set. You can copy the date from the host by running `ssh pi@hostname.local sudo date -s$(date -Ins)` before connection.

Setup a Static IP Address

Questions about setting Static IP Address are among the most common on this site. There are very many tutorials (many wrong, obsolete or incomplete).

Find the Settings of your local Network

This is most easily done with the Pi itself, using DHCP, but can be done on any computer on your network, although the commands may differ on other systems. Run

```
ip -4 addr show | grep global
```

which should give an output like:

```
inet 10.1.1.30/24 brd 10.1.1.255 scope global eth0
inet 10.1.1.31/24 brd 10.1.1.255 scope global wlan0
```

The first address is the IP address of your Pi on the network, and the part after the slash is the network size. It is highly likely that yours will be a /24.

The second address is the brd (broadcast) address of the network.

Find the address of your router (or gateway)

```
ip route | grep default | awk '{print $3}'
10.1.1.1
```

Finally note down the address of your DNS server, which is often the same as your gateway.

```
Cat /etc/resolv.conf
# Generated by resolvconf
nameserver 10.1.1.1
```

Then follow **ONE** of the following options. There are other methods not documented here.

These are the most common on Raspbian. **In either method substitute the appropriate network interface name for eth0, wlan0 or [predictable network interface names](#).**

If you want to find the interface names, even if not connected, run the following command

```
ls /sys/class/net/
```

dhcpcd method

Leave `/etc/network/interfaces` at its default (as above).

Edit `/etc/dhcpcd.conf` as follows:

```
interface eth0
static ip_address=10.1.1.30/24
static routers=10.1.1.1
static domain_name_servers=10.1.1.1

interface wlan0
static ip_address=10.1.1.31/24
static routers=10.1.1.1
static domain_name_servers=10.1.1.1
```

`ip_address` is the address and size from the command above or another unused address on the same network, `routers` is the address of your router aka gateway. `domain_name_servers` is the DNS address(es) from `/etc/resolv.conf`.

Network Interfaces method

Configure a static network address on your Pi in `/etc/network/interfaces`

```
auto eth0
iface eth0 inet static
    address 10.1.1.30
    netmask 255.255.255.0
    gateway 10.1.1.1

allow-hotplug wlan0
iface wlan0 inet static
    address 10.1.1.31
    netmask 255.255.255.0
    gateway 10.1.1.1

wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
```

`address` is the address from the command above or another unused address on the same network, `netmask 255.255.255.0` corresponds to network size `/24`. `gateway` is the address of your router aka gateway.

You can also specify `dns-nameservers`, but this is generally not necessary. The broadcast is automatically derived from address and netmask and need not be specified.

You can set either or both `eth0`, `wlan0` or one of the predictable network interface names. Then **disable the DHCP client daemon** and **switch to standard Debian networking**:

```
sudo systemctl disable dhcpcd
sudo systemctl enable networking
```

Reboot for the changes to take effect:

```
sudo reboot
```

Short and foolproof method how to do this with Raspbian Jessie or later versions

- Open `/boot/cmdline.txt` and add `ip=192.168.1.20` to the end of the line.
- Create an empty file `/boot/ssh`
- Boot your Raspberry Pi
- On Linux start `ssh pi@192.168.1.20` the password is `raspberry`.

- Use Putty or SmarTTY on Windows to connect via ssh.

I'm using this to access my Raspbian which boots without a attached monitor, with power and ethernet only. After accessing it with a ssh shell I can continue my setup.