

Setting WiFi up via the command line

This method is suitable if you don't have access to the graphical user interface normally used to set up WiFi on the Raspberry Pi. It is particularly suitable for use with a serial console cable if you don't have access to a screen or wired Ethernet network. Note also that no additional software is required; everything you need is already included on the Raspberry Pi.

Using raspi-config

The quickest way to enable wireless networking is to use the command line raspi-config tool.

```
sudo raspi-config
```

Select the **Network Options** item from the menu, then the **Wi-fi** option. On a fresh install, for regulatory purposes, you will need to specify the country in which the device is being used. Then set the SSID of the network, and the passphrase for the network. If you do not know the SSID of the network you want to connect to, see the next section on how to list available networks prior to running raspi-config.

Note that raspi-config does not provide a complete set of options for setting up wireless networking; you may need to refer to the extra sections below for more details if raspi-config fails to connect the Pi to your requested network.

Getting WiFi network details

To scan for WiFi networks, use the command `sudo iwlist wlan0 scan`. This will list all available WiFi networks, along with other useful information. Look out for:

1. 'ESSID:"testing"' is the name of the WiFi network.
2. 'IE: IEEE 802.11i/WPA2 Version 1' is the authentication used. In this case it's WPA2, the newer and more secure wireless standard which replaces WPA. This guide should work for WPA or WPA2, but may not work for WPA2 enterprise. For WEP hex keys, see the last example [here](#). You'll also need the password for the wireless network. For most home routers, this is found on a sticker on the back of the router. The ESSID (ssid) for the examples below is `testing` and the password (psk) is `testingPassword`.

Adding the network details to the Raspberry Pi

Open the `wpa-suppl icant` configuration file in nano:

```
sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
```

Go to the bottom of the file and add the following:

```
network={
    ssid="testing"
    psk="testingPassword"
}
```

The password can be configured either as the ASCII representation, in quotes as per the example above, or as a pre-encrypted 32 byte hexadecimal number. You can use the `wpa_passphrase` utility to generate an encrypted PSK. This takes the SSID and the password, and generates the encrypted PSK. With the example from above, you can generate the PSK with `wpa_passphrase "testing"`. Then you will be asked for the password of the WiFi network (in this case `testingPassword`). The output is as follows:

```
network={
    ssid="testing"
    #psk="testingPassword"

    psk=131e1e221f6e06e3911a2d11ff2fac9182665c004de85300f9cac208a6a8
    0531
}
```

Note that the plain text version of the code is present, but commented out. You should delete this line from the final `wpa_supplicant` file for extra security.

The `wpa_passphrase` tool requires a password with between 8 and 63 characters. For more complex passphrases you can extract the content of a text file and use it as input for `wpa_passphrase`, if the password is stored as plain text inside a file somewhere, by calling `wpa_passphrase "testing" < file_where_password_is_stored`. For extra security, you should delete the `file_where_password_is_stored` afterwards, so there is no plain text copy of the original password on the system.

To use the `wpa_passphrase` -encrypted PSK, you can either copy and paste the encrypted PSK into the `wpa_supplicant.conf` file, or redirect the tools output to the configuration file in one of two ways:

- Either change to `root` by executing `sudo su`, then call `wpa_passphrase "testing" >> /etc/wpa_supplicant/wpa_supplicant.conf` and enter the testing password when asked
- Or use `wpa_passphrase "testing" | sudo tee -a /etc/wpa_supplicant/wpa_supplicant.conf > /dev/null` and enter the testing password when asked; the redirection to `/dev/null` prevents `tee` from **also** outputting to the screen (standard output)

If you want to use one of these two options, **make sure you use `>>`, or use `-a` with `tee`** – either will **append** text to an existing file. Using a single chevron `>`, or omitting `-a` when using `tee`, will erase all contents and **then** append the output to the specified file.

Now save the file by pressing `Ctrl+X`, then `Y`, then finally press `Enter`.

Reconfigure the interface with `wpa_cli -i wlan0 reconfigure`.

You can verify whether it has successfully connected using `ifconfig wlan0`. If the `inet addr` field has an address beside it, the Raspberry Pi has connected to the network. If not, check that your password and ESSID are correct.

On the Raspberry Pi 3 Model B+, you will also need to set the country code, so that the 5G networking can choose the correct frequency bands. You can either use the `raspi-config` application and select the localisation option, or edit the

`wpa_supplicant.conf` file and add the following. (Note you need to replace 'GB' with the ISO code of your country. See [Wikipedia](https://en.wikipedia.org/wiki/ISO_3166-1_alpha-2) for a list of country codes.)

```
country=GB
```

Unsecured networks

If the network you are connecting to does not use a password, the

`wpa_supplicant` entry for the network will need to include the correct

`key_mgmt` entry. e.g.

```
network={
    ssid="testing"
    key_mgmt=NONE
}
```

Hidden networks

If you are using a hidden network, an extra option in the `wpa_supplicant file`, `scan_ssid`, may help connection.

```
network={
    ssid="yourHiddenSSID"
    scan_ssid=1
    psk="Your_wifi_password"
}
```

You can verify whether it has successfully connected using `ifconfig wlan0`. If the `inet addr` field has an address beside it, the Raspberry Pi has connected to the network. If not, check your password and ESSID are correct.

Adding multiple wireless network configurations

On recent versions of Raspbian, it is possible to set up multiple configurations for wireless networking. For example, you could set up one for home and one for school.

For example

```
network={
    ssid="SchoolNetworkSSID"
    psk="passwordSchool"
    id_str="school"
}

network={
    ssid="HomeNetworkSSID"
    psk="passwordHome"
    id_str="home"
}
```

If you have two networks in range, you can add the priority option to choose between them. The network in range, with the highest priority, will be the one that is connected.

```
network={
    ssid="HomeOneSSID"
    psk="passwordOne"
    priority=1
    id_str="homeOne"
}
```

```
network={
  ssid="HomeTwoSSID"
  psk="passwordTwo"
  priority=2
  id_str="homeTwo"
}
```

[VIEW/EDIT THIS PAGE ON GITHUB](#)
[READ OUR USAGE AND CONTRIBUTIONS POLICY](#)



ABOUT US

[About us](#)
[Our team](#)
[Governance](#)
[Safeguarding](#)
[Our supporters](#)
[Jobs](#)
[Contact us](#)

SUPPORT

[Help](#)
[Documentation](#)
[Projects](#)
[Training](#)
[Downloads](#)
[Research](#)
[FAQ](#)

Sign up to our newsletter

RASPBERRY PI FOUNDATION
UK REGISTERED CHARITY 1129409

[Privacy](#) [Cookies](#) [Trademark rules and brand guidelines](#)