Linux Wireless Support

**iwconfig for wireless configuration**

The iwconfig tool is like ifconfig and ethtool for wireless cards. You can view and set the basic Wi-Fi network details, such as the SSID, channel, and encryption. There's also many advanced settings you can view and change, including receive sensitivity, RTS/CTS, fragmentation, and retries. Here are some commands you may want to try:

Display the wireless settings of your interfaces, including the interface names you'll need for other commands:

**iwconfig**

Set the ESSID (Extended Service Set Identifier) or network name:

**iwconfig essid**

Example: **iwconfig eth0 essid "my network"**

Example: **iwconfig eth0 essid any**

Set the wireless mode:

**iwconfig eth0 mode Managed**

**iwconfig eth0 mode Ad-Hoc**

Set the wireless channel/freq of the radio (1-11):

**iwconfig eth0 channel 3**

**iwconfig eth0 channel auto**

**iwconfig eth0 freq 2.422G**

Input a WEP encryption key (WPA/WPA2 isn't supported yet; for this you need wpa\_supplicant):

**iwconfig eth0 key**

Only allow the adapter to connect to an AP with the MAC address you specify:

**iwconfig ap**

Example: **iwconfig eth0 ap 00:60:1D:01:23:45**

Set the transmit power of the radio, if supported by the wireless card, in dBm format by default or mW when specified:

**iwconfig txpower**

Example: **iwconfig eth0 txpower 15**

Example: **iwconfig eth0 txpower 30mW**

* **Wireless Statistics (extracted from /proc/net/wireless)**

**cat /proc/net/wireless**

* **iw is a new**[**nl80211**](https://wireless.wiki.kernel.org/en/developers/documentation/nl80211)**based CLI configuration utility for wireless devices**

**Help**

Just enter

iw help

on your command line and it will print out the commands it supports.

**Getting device capabilities**

Use the following to get device capabilities for all devices, such as band information (2.4 GHz, and 5 GHz), and 802.11n information:

iw list

**Scanning**

iw dev wlan0 scan

**Listening to events**

Just use

iw event

When debugging, it can be useful to see the auth/assoc/deauth/disassoc frames, use

iw event -f

and sometimes timing information is also useful:

iw event -t

**Getting link status**

To determine if you are connected to an AP or not and if you are the last TX rate used you can use the command below.

Example output when associated to a legacy (non-802.11n) AP:

iw dev wlan0 link

Connected to 04:21:b0:e8:c8:8b (on wlan0)

SSID: attwifi

freq: 2437

RX: 2272 bytes (18 packets)

TX: 232 bytes (3 packets)

signal: -57 dBm

tx bitrate: 36.0 MBit/s

Example output when associated to an 802.11n AP:

iw dev wlan0 link

Connected to 68:7f:74:3b:b0:01 (on wlan0)

SSID: tesla-5g-bcm

freq: 5745

RX: 30206 bytes (201 packets)

TX: 4084 bytes (23 packets)

signal: -31 dBm

tx bitrate: 300.0 MBit/s MCS 15 40Mhz short GI

Example output when not connected to an AP:

iw dev wlan0 link

Not connected.

**Establishing a basic connection**

You can use *iw* to connect to an AP directly if and only if the AP has:

* No encryption

To connect to an AP that has encryption disabled steps

sudo stop network-manager

sudo gedit /etc/network/interfaces

Changes in /etc/network/Interfaces

***Auto wlan0***

***iface wlan0 inet dhcp***

sudo restart network-manager

ip link show wlan0

sudo ip link set wlan0 up

ip link show wlan0

sudo iw wlan0 scan

sudo iw wlan0 connect MiPhone

iw wlan0 link

sudo iw wlan0 disconnect

To connect to an AP that has encryption disabled, where its SSID is *foo*:

iw wlan0 connect cdac123

Suppose you have two APs with the SSID *foo*, and you know the one you want to connect to is on the frequency 2432, you can specify the frequency to use:

iw wlan0 connect foo 2432

**Getting station statistics**

To get station statistic information such as the amount of tx/rx bytes, the last TX bitrate (including MCS rate) you can do:

$ iw dev wlan1 station dump

Station 12:34:56:78:9a:bc (on wlan0)

inactive time: 304 ms

rx bytes: 18816

rx packets: 75

tx bytes: 5386

tx packets: 21

signal: -29 dBm

tx bitrate: 54.0 MBit/s

**Getting station statistics against a peer**

If you want to get specific statistics against a peer you station is communicating with you can use the following:

sudo iw dev wlan1 station get <peer-MAC-address>

In the case of a STA the above <peer-MAC-address> would be the MAC address of your AP.

**Modifying transmit bitrates**

iw supports modifying TX bitrates, both legacy and HT MCS rates. It does this by masking in the allowed bitrates, and also lets you clear the mask.

**Modifying tx legacy bitrates**

You can set preference for transmitting using only certain legacy bitrates. For example:

iw wlan0 set bitrates legacy-2.4 12 18 24

Here's how to enable what some folks call “Purge G” which disables 802.11b associations:

iw wlan0 set bitrates legacy-2.4 6 12 24

**Modifying tx HT MCS bitrates**

Setting preference for transmitting using MCS rates is supported by letting you specify the band and MCS rate. Note that whether or not the device actually listens to your petition will vary depending on the device driver and cooperation from the firmware. For example:

iw dev wlan0 set bitrates mcs-5 4

iw dev wlan0 set bitrates mcs-2.4 10

To clear all tx bitrates and set things back to normal:

iw dev wlan0 set bitrates mcs-2.4

iw dev wlan0 set bitrates mcs-5

**Setting TX power**

You can set the txpower by using either the device interface name of the respective phy.

iw dev <devname> set txpower <auto|fixed|limit> [<tx power in mBm>]

iw phy <phyname> set txpower <auto|fixed|limit> [<tx power in mBm>]

(Note that the value this commands takes is in *milli*bel-milliwatts (mBm) instead of the commonly used *deci*bel-milliwatts (dBm). <power in mBm> = 100 \* <power in dBm>)

**Power save**

To enable [power save](https://wireless.wiki.kernel.org/en/developers/documentation/ieee80211/power-savings) by default you can use:

sudo iw dev wlan0 set power\_save on

For mac80211 drivers this means [Dynamic Power Save](https://wireless.wiki.kernel.org/en/users/documentation/dynamic-power-save) gets enabled.

To query the current power save settings you can use:

iw dev wlan0 get power\_save

Wireless Extensions CLI tools Method

This process is often used as the fall back position if the gui tools fail to work. The following example uses eth1 for the wifi interface though this may be wlan0, ath0 or something else in your case.

Preparation

Ensure that network manager has been stopped and then bring the interface down.

sudo service network-manager stop

sudo ip link set eth1 down

Configuration

1. Switch the card into ad hoc mode

sudo iwconfig eth1 mode ad-hoc

1. Set the channel/frequency that you want to use.

sudo iwconfig eth1 channel 4

1. Add the name (ssid) for the network you want to create/join. Use single quotes if there is a space in the name.

sudo iwconfig eth1 essid 'name'

1. Add a WEP encryption key

sudo iwconfig eth1 key 1234567890

Activation

1. Bring the interface back up

sudo ip link set eth1 up

1. Start dhclient to get an address

sudo dhclient eth1

1. If you want to do it manually, you will have to make up an IP address.

sudo ip addr add 169.254.34.2/16 dev eth1

You should now be connected to the ad hoc network. Use the iwconfig command to view the cell being used. This must match the same cell on the other PCs that you want to connect to.

Testing

Once you have two or more computers connected to the ad hoc network, all of them showing the same cell number and having different IP addresses, then try pinging one from the others.

ping 169.254.34.2

If that doesn't work then try specifying the interface name along with the address.

ping -I eth1 169.254.34.2

If you have more than one interface with link-local addresses there can be some routing confusion, so I recommend that you bring down these unused interfaces while connected to non-DHCP networks. If the other interfaces are all on DHCP networks, such as in the case of [InternetConnectionSharing](https://help.ubuntu.com/community/InternetConnectionSharing) this is not necessary.

sudo ifdown eth0

Example Interfaces File

Currently Network Manager does not support static address settings and Network-Admin does not support ad hoc networks. So instead we can configure the network in the /etc/network/interfaces file. The following is an example.

auto lo

iface lo inet loopback

auto eth1

iface eth1 inet static

wireless-mode ad-hoc

wireless-channel 4

wireless-essid 'name'

wireless-key 1234567890

address 192.168.0.2

netmask 255.255.255.0

gateway 192.168.0.1

# **File Server**

One of the most common ways to network Ubuntu and Windows computers is to configure Samba as a File Server. This section covers setting up a *Samba* server to share files with Windows clients.

The server will be configured to share files with any client on the network without prompting for a password. If your environment requires stricter Access Controls see [Securing File and Print Server](https://help.ubuntu.com/lts/serverguide/samba-fileprint-security.html).

1. [Installation](https://help.ubuntu.com/lts/serverguide/samba-fileserver.html#samba-fileserver-installation)
2. [Configuration](https://help.ubuntu.com/lts/serverguide/samba-fileserver.html#samba-fileserver-configuration)
3. [Resources](https://help.ubuntu.com/lts/serverguide/samba-fileserver.html#samba-fileserver-resources)

## Installation

The first step is to install the *samba* package. From a terminal prompt enter:

sudo apt install samba

That's all there is to it; you are now ready to configure Samba to share files.

## Configuration

The main Samba configuration file is located in /etc/samba/smb.conf. The default configuration file has a significant amount of comments in order to document various configuration directives.

Not all the available options are included in the default configuration file. See the smb.conf *man* page or the [Samba HOWTO Collection](http://samba.org/samba/docs/man/Samba-HOWTO-Collection/) for more details.

1. First, edit the following key/value pairs in the *[global]* section of /etc/samba/smb.conf:
2. workgroup = EXAMPLE
3. ...
4. security = user

The *security* parameter is farther down in the [global] section, and is commented by default. Also, change *EXAMPLE* to better match your environment.

1. Create a new section at the bottom of the file, or uncomment one of the examples, for the directory to be shared:
2. [share]
3. comment = Ubuntu File Server Share
4. path = /srv/samba/share
5. browsable = yes
6. guest ok = yes
7. read only = no
8. create mask = 0755
   1. *comment:* a short description of the share. Adjust to fit your needs.
   2. *path:* the path to the directory to share.

This example uses /srv/samba/sharename because, according to the *Filesystem Hierarchy Standard (FHS)*, [/srv](http://www.pathname.com/fhs/pub/fhs-2.3.html#SRVDATAFORSERVICESPROVIDEDBYSYSTEM) is where site-specific data should be served. Technically Samba shares can be placed anywhere on the filesystem as long as the permissions are correct, but adhering to standards is recommended.

* 1. *browsable:* enables Windows clients to browse the shared directory using *Windows Explorer*.
  2. *guest ok:* allows clients to connect to the share without supplying a password.
  3. *read only:* determines if the share is read only or if write privileges are granted. Write privileges are allowed only when the value is *no*, as is seen in this example. If the value is *yes*, then access to the share is read only.
  4. *create mask:* determines the permissions new files will have when created.

1. Now that *Samba* is configured, the directory needs to be created and the permissions changed. From a terminal enter:
2. sudo mkdir -p /srv/samba/share
3. sudo chown nobody:nogroup /srv/samba/share/

The *-p* switch tells mkdir to create the entire directory tree if it doesn't exist.

1. Finally, restart the *samba* services to enable the new configuration:
2. sudo service smbd restart
3. sudo service nmbd restart

Once again, the above configuration gives all access to any client on the local network. For a more secure configuration see [Securing File and Print Server](https://help.ubuntu.com/lts/serverguide/samba-fileprint-security.html).

From a Windows client you should now be able to browse to the Ubuntu file server and see the shared directory. If your client doesn't show your share automatically, try to access your server by its IP address, e.g. \\192.168.1.1, in a Windows Explorer window. To check that everything is working try creating a directory from Windows.

To create additional shares simply create new *[dir]* sections in /etc/samba/smb.conf, and restart *Samba*. Just make sure that the directory you want to share actually exists and the permissions are correct.

The file share named *"[share]"* and the path /srv/samba/share are just examples. Adjust the share and path names to fit your environment. It is a good idea to name a share after a directory on the file system. Another example would be a share name of *[qa]* with a path of /srv/samba/qa.

To access your network share

**sudo apt-get install smbclient**

**# List all shares:**

**smbclient -L //<HOST\_IP\_OR\_NAME>/<folder\_name> -U <user>**

**# connect:**

**smbclient //<HOST\_IP\_OR\_NAME>/<folder\_name> -U <user>**

To access your network share use your username (<user\_name>) and password through the path "smb://<HOST\_IP\_OR\_NAME>/<folder\_name>/" (Linux users) or "\\<HOST\_IP\_OR\_NAME>\<folder\_name>\" (Windows users). Note that "<folder\_name>" value is passed in "[<folder\_name>]", in other words, the share name you entered in "/etc/samba/smb.conf".

1. Note: The default user group of samba is "WORKGROUP".

**To see Routing Table:-**

Command :- route -n

**To Add Route in Routing Table:-**

Command :-

ip route add <Destination Network>/subnet-mask via <Gateway>

eg:-

ip route add 10.42.0.0/24 via 10.42.0.1

**To Delete Route from Routing Table:-**

Command :-

ip route flush <Destination Network>/subnet-mask via <Gateway>

eg:-

ip route flush 10.42.0.0/24 via 10.42.0.1

ip route flush 0.0.0.0/0 via 10.42.0.1

**Adding wpa\_passphrase generated configuration in wpa\_supplicant.conf:-**

wpa\_passphrase <ssid> <key> > /etc/wpa\_supplicant/wpa\_supplicant.conf

wpa\_passphrase jio 12345678 > /etc/wpa\_supplicant/wpa\_supplicant.conf

or

use below command and then write generated configuration manually in /etc/wpa\_supplicant/wpa\_supplicant.conf file

**To add wpa\_passphrase :-**

wpa\_passphrase <SSID> <PASSWORD>

eg:-

wpa\_passphrase jio 12345678

**it will generate configuration**

network={

ssid="jio"

#psk="12345678"

psk=b97ed542efb9f42310e606dd05ab87019439501e5a48446214b1d8e0ea722b2c

}

then write generated configuration manually in /etc/wpa\_supplicant/wpa\_supplicant.conf file