

This document had described the way to inform the wpa_supplicant to do the WiFi connection by using the wpa_cli. The wpa_supplicant had supported all kinds of security connections and WPS defined in the 802.11 specification. So, we suggest use the wpa_supplicant to do the WiFi connection rather than the iwconfig wireless tool.

(A) Start WPA_SUPPLICANT

1. compile wpa_supplicant

```
#cd wpa_supplicant
#make
```

2. If compile fail like this:

```
...
...
../src/drivers/driver_nl80211.c:19:31: fatal error: netlink/genl/genl.h: No such file
#include <netlink/genl/genl.h>
      ^
compilation terminated.
make: *** [../src/drivers/driver_nl80211.o] Error 1
```

Try to find which library contains the missing file by:

```
#sudo apt-file search /netlink/genl/genl.h
```

If lost libnl, install the library:

```
# sudo apt-get install libnl-3-dev
```

Re-compile wpa_supplicant

```
# make
```

3. Start wpa_supplicant

- i. Run wpa_supplicant in background:

If cfg80211:

```
# wpa_supplicant -Dnl80211 -iwlan0 -c ../../wpa_0_8.conf -B
```

Or wireless extensions:

```
# wpa_supplicant -Dwext -iwlan0 -c ../../wpa_0_8.conf -B
```

- ii. Run wpa_supplicant in background with debug message (This may affect the performance, only used in debug purpose.)

If cfg80211:

```
# wpa_supplicant -Dnl80211 -iwlan0 -c ../../wpa_0_8.conf -dd &
```

Or wireless extension:

```
# wpa_supplicant -Dwext -iwlan0 -c ../../wpa_0_8.conf -dd &
```

(B) WPA_CLI commands

1. Scanning AP and See Results

```
# wpa_cli -p/var/run/wpa_supplicant scan
# wpa_cli -p/var/run/wpa_supplicant scan_results
```

2. Connect to AP

a. OPEN

```
# wpa_cli -p/var/run/wpa_supplicant remove_network 0
# wpa_cli -p/var/run/wpa_supplicant ap_scan 1
# wpa_cli -p/var/run/wpa_supplicant add_network
# wpa_cli -p/var/run/wpa_supplicant set_network 0 ssid "dlink"
# wpa_cli -p/var/run/wpa_supplicant set_network 0 key_mgmt NONE
# wpa_cli -p/var/run/wpa_supplicant select_network 0
```

b. WEP40 with open system

```
# wpa_cli -p/var/run/wpa_supplicant remove_network 0
# wpa_cli -p/var/run/wpa_supplicant ap_scan 1
# wpa_cli -p/var/run/wpa_supplicant add_network
# wpa_cli -p/var/run/wpa_supplicant set_network 0 ssid "dlink"
# wpa_cli -p/var/run/wpa_supplicant set_network 0 key_mgmt NONE
# wpa_cli -p/var/run/wpa_supplicant set_network 0 wep_key0 1234567890
# wpa_cli -p/var/run/wpa_supplicant set_network 0 wep_tx_keyidx 0
# wpa_cli -p/var/run/wpa_supplicant select_network 0
```

c. WEP40 with shared key mode

```
# wpa_cli -p/var/run/wpa_supplicant remove_network 0
# wpa_cli -p/var/run/wpa_supplicant ap_scan 1
# wpa_cli -p/var/run/wpa_supplicant add_network
# wpa_cli -p/var/run/wpa_supplicant set_network 0 ssid "dlink"
# wpa_cli -p/var/run/wpa_supplicant set_network 0 key_mgmt NONE
# wpa_cli -p/var/run/wpa_supplicant set_network 0 wep_key0 1234567890
# wpa_cli -p/var/run/wpa_supplicant set_network 0 wep_tx_keyidx 0
# wpa_cli -p/var/run/wpa_supplicant set_network 0 auth_alg SHARED
# wpa_cli -p/var/run/wpa_supplicant select_network 0
```

d. WEP104 with open system

```
# wpa_cli -p/var/run/wpa_supplicant remove_network 0
# wpa_cli -p/var/run/wpa_supplicant ap_scan 1
# wpa_cli -p/var/run/wpa_supplicant add_network
# wpa_cli -p/var/run/wpa_supplicant set_network 0 ssid "dlink"
# wpa_cli -p/var/run/wpa_supplicant set_network 0 key_mgmt NONE
# wpa_cli -p/var/run/wpa_supplicant set_network 0 wep_key0
12345678901234567890123456
# wpa_cli -p/var/run/wpa_supplicant set_network 0 wep_tx_keyidx 0
# wpa_cli -p/var/run/wpa_supplicant select_network 0
```

e. WEP104 with shared key mode

```
# wpa_cli -p/var/run/wpa_supplicant remove_network 0
# wpa_cli -p/var/run/wpa_supplicant ap_scan 1
# wpa_cli -p/var/run/wpa_supplicant add_network
# wpa_cli -p/var/run/wpa_supplicant set_network 0 ssid "dlink"
# wpa_cli -p/var/run/wpa_supplicant set_network 0 key_mgmt NONE
# wpa_cli -p/var/run/wpa_supplicant set_network 0 wep_key0
```

```
12345678901234567890123456
# wpa_cli -p/var/run/wpa_supplicant set_network 0 wep_tx_keyidx 0
# wpa_cli -p/var/run/wpa_supplicant set_network 0 auth_alg SHARED
# wpa_cli -p/var/run/wpa_supplicant select_network 0
```

- (1) If wep key is ASCII type, use the following cmd:
For WEP40

```
# wpa_cli -p/var/run/wpa_supplicant set_network 0 wep_key0 "12345"
For WEP104
```

```
# wpa_cli -p/var/run/wpa_supplicant set_network 0 wep_key0
"1234567890123"
```

- (2) WEP key index is X from 0 to 3, change X for other key index and select it.

```
# wpa_cli -p/var/run/wpa_supplicant set_network 0 wep_keyX
12345678901234567890123456
# wpa_cli -p/var/run/wpa_supplicant set_network 0 wep_tx_keyidx X
```

f. TKIP and AES

```
# wpa_cli -p/var/run/wpa_supplicant remove_network 0
# wpa_cli -p/var/run/wpa_supplicant ap_scan 1
# wpa_cli -p/var/run/wpa_supplicant add_network
# wpa_cli -p/var/run/wpa_supplicant set_network 0 ssid "dlink"
# wpa_cli -p/var/run/wpa_supplicant set_network 0 key_mgmt WPA-PSK
# wpa_cli -p/var/run/wpa_supplicant set_network 0 psk "12345678"
# wpa_cli -p/var/run/wpa_supplicant select_network 0
```

3. Ad-hoc mode

a. OPEN

```
# wpa_cli -p/var/run/wpa_supplicant scan
# wpa_cli -p/var/run/wpa_supplicant scan_results
# wpa_cli -p/var/run/wpa_supplicant remove_network 0
# wpa_cli -p/var/run/wpa_supplicant ap_scan 2
# wpa_cli -p/var/run/wpa_supplicant add_network
# wpa_cli -p/var/run/wpa_supplicant set_network 0 ssid "Adhoc_test"
# wpa_cli -p/var/run/wpa_supplicant set_network 0 mode 1
# wpa_cli -p/var/run/wpa_supplicant set_network 0 key_mgmt NONE
# wpa_cli -p/var/run/wpa_supplicant set_network 0 frequency 2412
# wpa_cli -p/var/run/wpa_supplicant select_network 0
```

#frequency is to set the channel frequency for Ad-hoc master.

b. WEP40

```
# wpa_cli -p/var/run/wpa_supplicant scan
# wpa_cli -p/var/run/wpa_supplicant scan_results
# wpa_cli -p/var/run/wpa_supplicant remove_network 0
# wpa_cli -p/var/run/wpa_supplicant ap_scan 2
# wpa_cli -p/var/run/wpa_supplicant add_network
# wpa_cli -p/var/run/wpa_supplicant set_network 0 ssid "Adhoc_test"
# wpa_cli -p/var/run/wpa_supplicant set_network 0 mode 1
# wpa_cli -p/var/run/wpa_supplicant set_network 0 key_mgmt NONE
# wpa_cli -p/var/run/wpa_supplicant set_network 0 wep_key0 1234567890
# wpa_cli -p/var/run/wpa_supplicant set_network 0 wep_tx_keyidx 0
# wpa_cli -p/var/run/wpa_supplicant set_network 0 frequency 2412
```

```
# wpa_cli -p/var/run/wpa_supplicant select_network 0
```

c. WEP104

```
# wpa_cli -p/var/run/wpa_supplicant scan  
# wpa_cli -p/var/run/wpa_supplicant scan_results  
# wpa_cli -p/var/run/wpa_supplicant remove_network 0  
# wpa_cli -p/var/run/wpa_supplicant ap_scan 2  
# wpa_cli -p/var/run/wpa_supplicant add_network  
# wpa_cli -p/var/run/wpa_supplicant set_network 0 ssid "Adhoc_test"  
# wpa_cli -p/var/run/wpa_supplicant set_network 0 mode 1  
# wpa_cli -p/var/run/wpa_supplicant set_network 0 key_mgmt NONE  
# wpa_cli -p/var/run/wpa_supplicant set_network 0 wep_key0  
12345678901234567890123456  
# wpa_cli -p/var/run/wpa_supplicant set_network 0 wep_tx_keyidx 0  
# wpa_cli -p/var/run/wpa_supplicant set_network 0 frequency 2412  
# wpa_cli -p/var/run/wpa_supplicant select_network 0
```

4. Save the Current Connection AP configuration file

```
# wpa_cli -p/var/run/wpa_supplicant save_config
```

5. WPS Connection

(1) Push Button:

```
# wpa_cli -p/var/run/wpa_supplicant remove_network 0  
# wpa_cli -p/var/run/wpa_supplicant wps_pbc any
```

(2) Pin Code:

```
# wpa_cli -p/var/run/wpa_supplicant remove_network 0  
# wpa_cli -p/var/run/wpa_supplicant wps_pin any 12345670
```

Or

```
# wpa_cli -p/var/run/wpa_supplicant remove_network 0  
# wpa_cli -p/var/run/wpa_supplicant wps_pin any
```

6. Get Current Status of wpa_supplicant

```
# wpa_cli -p/var/run/wpa_supplicant status
```

7. Disable current network connection

```
# wpa_cli -p/var/run/wpa_supplicant disable_network 0
```

(C) Using WPA_SUPPLICANT by WPA_CLI (Control interface commands)

1. Start **wpa_cli** control interface:

```
# wpa_cli
```

2. Commands:

PING

This command can be used to test whether wpa_supplicant is replying to the control interface commands.

The expected reply is PONG if the connection is open and wpa_supplicant is processing commands.

STATUS

Request current status information. The output is a text block with each line in variable=value format. For example:

```
bssid=02:00:01:02:03:04
ssid=test network
pairwise_cipher=CCMP
group_cipher=CCMP
key_mgmt=WPA-PSK
wpa_state=COMPLETED
```

LIST_NETWORKS

List configured networks.

network id / ssid / bssid / flags

0 example network any [CURRENT]

(note: fields are separated with tabs)

SCAN

Request a new BSS scan.

SCAN_RESULTS

Get the latest scan results.

```
bssid / frequency / signal level / flags / ssid
00:09:5b:95:e0:4e 2412 208 [WPA-PSK-CCMP] jkm private
02:55:24:33:77:a3 2462 187 [WPA-PSK-TKIP] testing
00:09:5b:95:e0:4f 2412 209 jkm guest
```

(note: fields are separated with tabs)

ADD_NETWORK

Add a new network. This command creates a new network with empty configuration. The new network is

disabled and once it has been configured it can be enabled with ENABLE_NETWORK command.

ADD_

NETWORK returns the [network id](#) of the new network or FAIL on failure

SELECT_NETWORK <network id>

Select a network (disable others). Network id can be received from the LIST_NETWORKS command output.

ENABLE_NETWORK <network id>

Enable a network. Network id can be received from the LIST_NETWORKS command output.

DISABLE_NETWORK <network id>

Disable a network. Network id can be received from the LIST_NETWORKS command output.
Special
network id **all** can be used to disable all network.

REMOVE_NETWORK <network id>

Remove a network. Network id can be received from the LIST_NETWORKS command output.
Special
network id **all** can be used to remove all network.

SET_NETWORK <network id> <variable> <value>

Set network variables. Network id can be received from the LIST_NETWORKS command output.
This command uses the same variables and data formats as the configuration file.

- ssid (network name, SSID)
- psk (WPA passphrase or pre-shared key)
- key_mgmt (key management protocol, NONE, WPA-PSK, WPA-EAP)
- proto (WPA WPA2)
- pairwise (CCMP TKIP)
- group (CCMP TKIP WEP40 WEP104)
- wep_key0 (set wep key for key index 0)
- wep_tx_keyidx (select wep key index)
- frequency (Channel frequency in megahertz (MHz) for IBSS)

GET_NETWORK <network id> <variable>

Get network variables. Network id can be received from the LIST_NETWORKS command output.

SAVE_CONFIG

Save the current configuration.

AP_SCAN <ap_scan value>

Change ap_scan value: 0 = no scanning, 1 = wpa_supplicant requests scans and uses scan results to select the AP, 2 = wpa_supplicant does not use scanning and just requests driver to associate and take care of AP selection