

# **Package WLAN**

## **Version 3.10.18**

Frank Meyer  
email: [frank@fli4l.de](mailto:frank@fli4l.de)

The fli4l-Team  
email: [team@fli4l.de](mailto:team@fli4l.de)

September 15, 2019

# Contents

<b>1. Documentation Of Package WLAN</b>	<b>3</b>
1.1. WLAN - Support For Wireless-LAN . . . . .	3
1.1.1. WLAN Configuration . . . . .	3
1.1.2. Examples . . . . .	7
1.1.3. Virtual Accesspoint (VAP) (experimental) . . . . .	9
1.1.4. Switching WLAN on and off based on daytime with easycron . . . . .	9
1.1.5. Donations . . . . .	9
<b>A. Appendix For Package WLAN</b>	<b>10</b>
<b>List of Figures</b>	<b>11</b>
<b>List of Tables</b>	<b>12</b>
<b>Index</b>	<b>13</b>

# 1. Documentation Of Package WLAN

## 1.1. WLAN - Support For Wireless-LAN

When using PCI Cards please be sure to use a mainboard that at least complies to the specifications of PCI 2.2. Older mainboards that only support PCI 2.1 or less can produce diverse errors. Either the computer does not start at all (it even can't be switched on) or the WLAN card is not found on PCI scan.

WLAN cards are addressed as wlanX in base.txt's IP\_NET\_X\_DEV. If only one WLAN card is installed its name is wlan0.

### 1.1.1. WLAN Configuration

**OPT\_WLAN** Default setting: `OPT_WLAN='no'`

Activates package Wireless LAN.

**WLAN\_WEBGUI** Default setting: `WLAN_WEBGUI='yes'`

Aktivates the web interface for package Wireless LAN.

**WLAN\_REGDOMAIN** This variable determines the country settings. Valid values are ISO 3166-1 alpha-2 country codes i.e. 'DE' for Germany. In most countries different legal presets apply for frequency channels and transmission power.

**WLAN\_N** Number of independent WLAN Configurations. If this is set to '1' behavior is like in earlier fli4l versions when only one configuration was allowed.

**WLAN\_x\_MAC** MAC address of the WLAN card in this notation:

XX:XX:XX:XX:XX:XX

Each X is a Hex Digit of the MAC Address for the card that belongs to this configuration. If none of the MAC addresses entered here matches a particular card the configuration in `WLAN_1_*` will be applied to this card and a warning message containing the card's MAC address will be displayed. Enter this address in the config file to assure that the web interface will work without problems.

**WLAN\_x\_MAC\_OVERRIDE** This setting changes the MAC address of the WLAN card. This is used to connect to a WLAN where MAC-filtering is active without changing the filters. This is useful for WAN connections that are fixed to i.e. the MAC address of the WLAN-USB-stick delivered by the provider.

**WLAN\_x\_ESSID** The Service Set Identifier (SSID) is the name for your wireless lan. The string has a maximum length of 32 characters and is also called "Network Name". It is configured in the access point of a wireless lan and is used by all clients accessing it. The SSID has to be identical for all joining nodes also for Ad-Hoc networking.

**WLAN\_x\_MODE** Sets the WLAN mode to be used by the card.

Default setting: `WLAN_x_MODE='ad-hoc'`

Possible values:

ad-hoc	wireless net without Access-Point
managed	managed wireless net with several cells
master	the WLAN card is working as an Access-Point

`WLAN_x_MODE='master'` will only working with adequate WLAN drivers.

**WLAN\_x\_NOESSID** Deactivates sending ESSID during beacon frames. Only valid with `hostap_*` driver and Firmware  $\geq 1.6.3$  in `WLAN_MODE='master'`

This feature is optional and has to be added manually in `config/wlan.txt`.

**WLAN\_x\_CHANNEL** Sets the transmission channel of the network.

Default setting: `WLAN_x_CHANNEL='1'`

Possible values: 1-13 and 36,40,44,48,52,56,60,64,100,104,108,112,116,120,124,128,132,136,140

Please read the documentation of your WLAN card to find out which channels are allowed to use in your country. You are responsible for any rights violation by using channels not allowed. In Germany channels 1-13 in 2,4 GHz frequency range (Modes: b and g) are allowed. Channels 36-140 (see above) are legally allowed in 5 GHz frequency range.

Value '0' is also valid if `WLAN_x_MODE='managed'` is set. This does not set a particular channel explicitly but searches for an AP on all valid channels. You may add the character a,b or g to the channel (i.e. 5g) which will then determine the operating mode and frequency range.

Adding 'n' or 'N' triggers usage of 802.11n for according WLAN cards. Lower case means: 20 MHz channel width, upper Case means: 40 MHz channel width.

Upper case for a/b/g will activate proprietary WLAN turbo modes with some drivers (at the moment only `ath_pci`). This option is experimental and may be removed at any time.

**WLAN\_x\_RATE** Sets transmission speed of the network.

Default setting: `WLAN_x_RATE='auto'`

Possible values: 1,2,5.5,11,auto - rates in Megabit/s

Depending on the card this rates can be chosen in addition: 6,9,12,18,24,36,48 and 54. Some 54 MBit cards don't accept rate settings. In this case specify 'auto' here.

**WLAN\_x\_RTS** Activates RTS/CTS handshake. This option is useful in bigger WLANs with a lot of clients if those clients can't receive each other but only the AP. If this option is activated the client will start each transmission with a RTS query to get permission for the actual data transmission. It gets a CTS from the AP then providing permission to send. This way every client knows another client is transmitting without receiving the other client. Collisions are minimized because it is ensured that only one client is transmitting data at a time. This option only makes sense in the situation described above because of additional overhead and thus decreasing bandwidth. Bandwidth can raise by avoiding collision though.

This feature is optional and has to be added manually in config/wlan.txt.

**WLAN\_x\_ENC\_N (deprecated)** Sets the number of Wireless Encryption Key's (WEP).

Possible values: 0-4

**WLAN\_x\_ENC\_x (deprecated)** Sets Wireless Encryption Keys.

Possible values:

XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-XX	128 Bit Hex-Key (X=0-F)
XXXX-XXXX-XX	64 Bit Hex-Key (X=0-F)
s:<5 characters>	64 Bit
s:<6-13 characters>	128 Bit
P:<1-64 characters>	128 Bit

Using s:text is **not** compatible with the passphrase of the Windows drivers. Use a hex key instead! Windows mostly uses hex keys **without** hyphens '-'. Using P:<text> is compatible to passphrases of most (if not all) Windows WLAN drivers but **only** in 128 Bit mode. Linux allows to mix key length which Windows drivers usually do **not**!

**WLAN\_x\_ENC\_ACTIVE (deprecated)** Sets the active wireless encryption key.

Possible values: 1-4

This variable must be set if WLAN\_x\_ENC\_N is greater than 0. In other cases it's optional.

**WLAN\_x\_ENC\_MODE (deprecated)** Activates the encryption mode.

Possible values:

on/off	with or without encryption
open	also accepts unencrypted packets
restricted	only accepts encrypted packets

Most reasonable value: 'restricted'

This feature is optional and has to be added manually in config/wlan.txt. If this variable is not set the default 'off' will be assumed if no WEP key was defined and 'restricted' if at least one key is defined.

**WLAN\_x\_WPA\_KEY\_MGMT** If you want to use WPA instead of WEP encryption set the WPA mode here. At the moment only WPA with a pre-shared key between client and AP (WPA-PSK) is supported. This key should be chosen carefully and not too short because that would allow for dictionary attacks.

In *managed* mode all cards supported by WPA-Supplicant ([http://hostap.epitest.fi/wpa\\_supplicant/](http://hostap.epitest.fi/wpa_supplicant/)) and in *master* mode all cards supported by Hostapd (<http://hostap.epitest.fi/hostapd/>) are accepted.

Cards based on chipsets by Atheros and on the hostap-driver were tested successfully in managed and master mode. In theory also atmel and some other cards are possible if developers of third-party opt-packages adapt their packages accordingly.

**WLAN\_x\_WPA\_PSK** Specify the pre-shared key to be used for communication between client and Access-Point here. The key is provided as a passphrase at a minimum length of 16 characters and a maximum length of 63 characters. The following characters are supported:

## 1. Documentation Of Package WLAN

a-z A-Z 0-9 ! # \$ % & ( ) \* + , - . / : ; < = > ? @ [ \ ] ^ \_ ` { | } ~

**WLAN\_x\_WPA\_TYPE** Choose between '1' for WPA1 mode, '2' for WPA2 (IEEE 802.11i) mode and '3' for both - the client can decide to use WPA1 or WPA2. Only WPA2 should be used if the hardware supports it.

**WLAN\_x\_WPA\_ENCRYPTION** Encryption protocols TKIP and the enhanced version CCMP (AES-CTR/CBC-MAC Protocol, mostly called AES in short) can be used. CCMP eventually won't be supported by older WLAN hardware. You may also specify both.

**WLAN\_x\_WPA\_DEBUG** In case of problems with WPA set this variable to 'yes' for the daemon to provide more verbose output for debugging.

**WLAN\_x\_AP** Registers the node with an Access-Point.

Specify the MAC address of the Access-Points here. If WLAN mode "master" was chosen before keep this variable empty. This option only makes sense if `fi4l` can't find the AP by itself or should be bound to a preferred Access-Point. Only to be used in WLAN mode "managed".

This feature is optional and has to be added manually in `config/wlan.txt`.

**WLAN\_x\_ACL\_POLICY** Access Control List (ACL) Policy.

Default setting: `WLAN_x_ACL_POLICY='allow'`

Describes what action should be taken for the provided MAC addresses:

deny    None of the addresses listed here gets access  
allow    Only the addresses listed here get access  
open    All MAC addresses get access independent on filter

Unfortunately `WLAN_ACLs` are only supported well by the `hostap_*` driver. As an alternative you may use the firewall rules that have improved a lot since `fi4l` version 3.0.x.

**WLAN\_x\_ACL\_MAC\_N** Number of restricted WLAN stations.

Default setting: `WLAN_x_ACL_MAC_N='0'`

A number greater than 0 activates the Access Control List.

**WLAN\_x\_ACL\_MAC\_x** MAC address in notation `XX:XX:XX:XX:XX:XX`  
(example: `00:00:E8:83:72:92`)

**WLAN\_x\_DIVERSITY** Trigger manual antenna diversity.

Default setting: `WLAN_x_DIVERSITY='no'` (automatic)

**WLAN\_x\_DIVERSITY\_RX** The receiving antenna to be used.

Default setting: `WLAN_x_DIVERSITY_RX='1'`

0 = Automatic  
1 = Antenna 1  
2 = Antenna 2

**WLAN\_x\_DIVERSITY\_TX** The transmitting antenna to be used.

Default setting: WLAN\_x\_DIVERSITY\_TX='1'

**WLAN\_x\_WPS** Activates WPS support. Push-Button and PIN are possible. If you don't only want to use the console it makes sense to activate WLAN\_WEBGUI.

Default setting: WLAN\_x\_WPS='no'

**WLAN\_x\_PSKFILE** With PSKFILE activated other client related keys can be used beside WLAN\_x\_WPA\_PSK pre-shared keys. At the moment the function WLAN\_x\_WPS uses this file to provide individual keys to clients.

If the file is deactivated WPS clients using it can not connect to the Access Point anymore.

WPS-Clients connected with deactivated file are not affected.

Default setting: WLAN\_x\_PSKFILE='yes'

**WLAN\_x\_BRIDGE** As an alternative to package ADVANCED\_NETWORKING you may specify the bridge to which the WLAN should be bound here.

Example: WLAN\_x\_BRIDGE='br0'

Attention: Use either ADVANCED\_NETWORKING or this setting and **not both!**

### 1.1.2. Examples

#### Connecting to an Access Point via WPA

```
OPT_WLAN='yes'
WLAN_N='1'
WLAN_1_MAC='00:0F:A3:xx:xx:xx'
WLAN_1_ESSID='foo'
WLAN_1_MODE='managed'           # connection to an Access Point
WLAN_1_CHANNEL='1'
WLAN_1_RATE='auto'
#
# WPA Configuration
#
WLAN_1_ENC_N='0'                # no WEP
WLAN_1_WPA_KEY_MGMT='WPA-PSK'   # WPA pre shared key
WLAN_1_WPA_TYPE='1'             # WPA 1
WLAN_1_WPA_ENCRYPTION='TKIP'
WLAN_1_WPA_PSK='your best passphrase choice ever (16-63 characters)'
#
# irrelevant in WPA context
#
WLAN_1_ENC_N='0'
WLAN_1_ENC_ACTIVE='1'
WLAN_1_ACL_POLICY='allow'
WLAN_1_ACL_MAC_N='0'
```

#### Access Point with WPA2 Encryption

```
OPT_WLAN='yes'
```

## 1. Documentation Of Package WLAN

```
WLAN_N='1'
WLAN_1_MAC='00:0F:A3:xx:xx:xx'
WLAN_1_ESSID='foo'
WLAN_1_MODE='master'           # Access Point
WLAN_1_CHANNEL='1g'           # Channel 1, Mode 'g' on an
                                # Atheros card

WLAN_1_RATE='auto'
#
# WPA Configuration
#
WLAN_1_ENC_N='0'               # no WEP
WLAN_1_WPA_KEY_MGMT='WPA-PSK' # WPA pre shared key
WLAN_1_WPA_TYPE='2'           # WPA 2
WLAN_1_WPA_ENCRYPTION='CCMP'
WLAN_1_WPA_PSK='your best passphrase choice ever (16-63 characters)'
#
# MAC based Access Control to AP
#
WLAN_1_ACL_POLICY='allow'
WLAN_1_ACL_MAC_N='0'
#
# irrelevant in WPA context
#
WLAN_1_ENC_ACTIVE='1'
```

### Access Point with WEP Encryption

```
OPT_WLAN='yes'
WLAN_N='1'
WLAN_1_MAC='00:0F:A3:xx:xx:xx'
WLAN_1_ESSID='foo'
WLAN_1_MODE='master'           # Access Point
WLAN_1_CHANNEL='1'
WLAN_1_RATE='auto'
#
# WEP Configuration
#
WLAN_1_WPA_KEY_MGMT=''         # no WPA
WLAN_1_ENC_N='4'               # 4 WEP-Keys
WLAN_1_ENC_1='...'
WLAN_1_ENC_2='...'
WLAN_1_ENC_3='...'
WLAN_1_ENC_4='...'
WLAN_1_ENC_ACTIVE='1'         # first key is active
#
# MAC based Access Control to AP
#
WLAN_1_ACL_POLICY='allow'
WLAN_1_ACL_MAC_N='0'
#
# irrelevant for WEP Configuration
#
WLAN_1_WPA_TYPE='2'
```

```
WLAN_1_WPA_ENCRYPTION='CCMP'  
WLAN_1_WPA_PSK='...'
```

### 1.1.3. Virtual Accesspoint (VAP) (experimental)

Certain WLAN cards (driver: ath\_pci, ath5k, ath9k, ath9k\_htc) can be split into a maximum of 4 virtual WLAN cards. (VAP)

WLAN configuration of virtual APs is arbitrary except for one condition: Channel and MAC address have to be the same. Based on the multiplexed MAC address the card that should be splitted is identified. If more physical cards are present this can be done repeatedly for each of them.

The base device's name is still wlan0, the next in VAP mode will be wlan0v2 a.s.o. For binding to a bridge please use WLAN\_x\_BRIDGE='br0'!

The maximum at the moment is up to 8x master depending on card and driver.

### 1.1.4. Switching WLAN on and off based on daytime with easycron

By using the package *easycron* (Page ??) WLAN may be switched on and off based on a time schedule.

```
EASYCRON_N='2'  
EASYCRON_1_CUSTOM = ''      # switch off every evening at 12PM  
EASYCRON_1_COMMAND = '/usr/sbin/wlanconfig.sh wlan0 down'  
EASYCRON_1_TIME    = '* 24 * * *'  
  
EASYCRON_2_CUSTOM = ''      # and on at 8 AM.  
EASYCRON_2_COMMAND = '/usr/sbin/wlanconfig.sh wlan0'  
EASYCRON_2_TIME    = '* 8 * * *'
```

### 1.1.5. Donations

Due to a generous donation of 2 Ralink 2500 based WLAN cards with RT25xx chipset those cards can be used in ad-hoc and managed modes. Use rt2500 as the driver in base.txt.

The cards where donated by:

Computer Contor, Pilgrimstein 24a, 35037 Marburg

## **A. Appendix For Package WLAN**

## List of Figures

## List of Tables

# Index

OPT\_WLAN, [3](#)

WLAN\_N, [3](#)

WLAN\_REGDOMAIN, [3](#)

WLAN\_WEBGUI, [3](#)

WLAN\_x\_ACL\_MAC\_N, [6](#)

WLAN\_x\_ACL\_MAC\_x, [6](#)

WLAN\_x\_ACL\_POLICY, [6](#)

WLAN\_x\_AP, [6](#)

WLAN\_x\_BRIDGE, [7](#)

WLAN\_x\_CHANNEL, [4](#)

WLAN\_x\_DIVERSITY, [6](#)

WLAN\_x\_DIVERSITY\_RX, [6](#)

WLAN\_x\_DIVERSITY\_TX, [6](#)

WLAN\_x\_ENC\_ACTIVE, [5](#)

WLAN\_x\_ENC\_MODE, [5](#)

WLAN\_x\_ENC\_N, [5](#)

WLAN\_x\_ENC\_x, [5](#)

WLAN\_x\_ESSID, [3](#)

WLAN\_x\_MAC, [3](#)

WLAN\_x\_MAC\_OVERRIDE, [3](#)

WLAN\_x\_MODE, [3](#)

WLAN\_x\_NOESSID, [4](#)

WLAN\_x\_PSKFILE, [7](#)

WLAN\_x\_RATE, [4](#)

WLAN\_x\_RTS, [4](#)

WLAN\_x\_WPA\_DEBUG, [6](#)

WLAN\_x\_WPA\_ENCRYPTION, [6](#)

WLAN\_x\_WPA\_KEY\_MGMT, [5](#)

WLAN\_x\_WPA\_PSK, [5](#)

WLAN\_x\_WPA\_TYPE, [6](#)

WLAN\_x\_WPS, [7](#)