

You are here: [Wiki mainpage](#) / [Scripting](#) / [SSH/Telnet & The CLI](#) / [wl \(wireless options\) command](#)

'wl' is probably the most important tool in the routers firmware to access and control wireless hardware. It can be accessed from a [telnet](#) or [SSH](#) terminal or through the Web Interface. Tools such as [Wiviz](#) or [Kismet Server/Drone](#) use it to configure the radios, monitor packets and extract information directly from them.

## Setup

No setup is required as 'wl' comes with all Broadcom DD-WRT installations.

(For Atheros Hardware, this functionality is absent/less developed, and users will need to use the 'ifconfig' and 'wl\_atheros' commands instead.

See these threads: [wl command "not found"](#), [Regarding "wl command not found" and others](#))

## Contents

- [1 Setup](#)
- [2 Usage](#)
- [3 Common Tasks](#)
  - ◆ [3.1 Toggle radio on/off, restart, etc](#)
- [4 Alphabetical list of commands](#)
  - ◆ [4.1 aciargs](#)
  - ◆ [4.2 addwep](#)
  - ◆ [4.3 aes](#)
  - ◆ [4.4 antdiv](#)
  - ◆ [4.5 ap](#)
  - ◆ [4.6 assoc](#)
  - ◆ [4.7 assoclist](#)
  - ◆ [4.8 atten](#)
  - ◆ [4.9 authe\\_sta\\_list](#)
  - ◆ [4.10 autho\\_sta\\_list](#)
  - ◆ [4.11 authorize](#)
  - ◆ [4.12 band](#)
  - ◆ [4.13 bands](#)
  - ◆ [4.14 bssid](#)
  - ◆ [4.15 cap](#)
  - ◆ [4.16 channel](#)
  - ◆ [4.17 channel\\_qa](#)
  - ◆ [4.18 channel\\_qa\\_start](#)
  - ◆ [4.19 clk](#)
  - ◆ [4.20 cmds](#)
  - ◆ [4.21 country](#)

- ◆ [4.22 constraint](#)
- ◆ [4.23 crsuprs](#)
- ◆ [4.24 csa](#)
- ◆ [4.25 cwmin](#)
- ◆ [4.26 cwmax](#)
- ◆ [4.27 disassoc](#)
- ◆ [4.28 dtim](#)
- ◆ [4.29 dump](#)
- ◆ [4.30 deauthenticate](#)
- ◆ [4.31 deauthorize](#)
- ◆ [4.32 eap](#)
- ◆ [4.33 chanlist](#)
- ◆ [4.34 channels](#)
- ◆ [4.35](#)  
[channels in country](#)
- ◆ [4.36 curpower](#)
- ◆ [4.37 evm](#)
- ◆ [4.38 frameburst](#)
- ◆ [4.39 fasttimer](#)
- ◆ [4.40 frag](#)
- ◆ [4.41 fqacurcy](#)
- ◆ [4.42 glacialtimer](#)
- ◆ [4.43 gmode](#)
- ◆ [4.44](#)  
[gmode protection](#)
- ◆ [4.45](#)  
[gmode protection control](#)
- ◆ [4.46](#)  
[gmode protection cts](#)
- ◆ [4.47](#)  
[gmode protection override](#)
- ◆ [4.48 ignore bcns](#)
- ◆ [4.49 int](#)
- ◆ [4.50 interference](#)
- ◆ [4.51 infra](#)
- ◆ [4.52 isup](#)
- ◆ [4.53 join](#)
- ◆ [4.54 keys](#)
- ◆ [4.55 lazywds](#)
- ◆ [4.56 lbt](#)
- ◆ [4.57 legacy\\_erp](#)
- ◆ [4.58 locale](#)
- ◆ [4.59 lrl](#)
- ◆ [4.60 mac](#)
- ◆ [4.61 macmode](#)
- ◆ [4.62 macreg](#)
- ◆ [4.63 measure\\_req](#)
- ◆ [4.64 monitor](#)
- ◆ [4.65 mrate](#)
- ◆ [4.66 msglevel](#)

- ◆ [4.67 noise](#)
- ◆ [4.68 nvdump](#)
- ◆ [4.69 nvget](#)
- ◆ [4.70 nvset](#)
- ◆ [4.71 passive](#)
- ◆ [4.72 phylist](#)
- ◆ [4.73 phyreg](#)
- ◆ [4.74 radioreg](#)
- ◆ [4.75 phytype](#)
- ◆ [4.76 pktent](#)
- ◆ [4.77 plcphdr](#)
- ◆ [4.78 PM](#)
- ◆ [4.79 powerindex](#)
- ◆ [4.80](#)
  - [prb resp timeout](#)
- ◆ [4.81 primary key](#)
- ◆ [4.82 promisc](#)
- ◆ [4.83 pwr percent](#)
- ◆ [4.84 quiet](#)
- ◆ [4.85 radar](#)
- ◆ [4.86 radio](#)
- ◆ [4.87 rate](#)
- ◆ [4.88 ratedump](#)
- ◆ [4.89 rateparam](#)
- ◆ [4.90 rateset](#)
- ◆ [4.91 reboot](#)
- ◆ [4.92 revinfo](#)
- ◆ [4.93 regulatory](#)
- ◆ [4.94 roam delta](#)
- ◆ [4.95 rm rep](#)
- ◆ [4.96 rm req](#)
- ◆ [4.97 rmwep](#)
- ◆ [4.98](#)
  - [roam scan period](#)
- ◆ [4.99 roam trigger](#)
- ◆ [4.100 rssi](#)
- ◆ [4.101 rssidump](#)
- ◆ [4.102 rts](#)
- ◆ [4.103](#)
  - [scan channel time](#)
- ◆ [4.104](#)
  - [scan home time](#)
- ◆ [4.105 scan nprobes](#)
- ◆ [4.106](#)
  - [scan passive time](#)
- ◆ [4.107](#)
  - [scan unassoc time](#)
- ◆ [4.108 scanresults](#)
- ◆ [4.109 scansuppress](#)
- ◆ [4.110 scb timeout](#)

- ◆ [4.111 scbdump](#)
- ◆ [4.112 scan](#)
- ◆ [4.113 set\\_pmk](#)
- ◆ [4.114 shmem](#)
- ◆ [4.115 shortslot](#)
- ◆ [4.116 shortslot\\_override](#)
- ◆ [4.117 shortslot\\_restrict](#)
- ◆ [4.118 slowtimer](#)
- ◆ [4.119 spect](#)
- ◆ [4.120 srdump](#)
- ◆ [4.121 srl](#)
- ◆ [4.122 ssid](#)
- ◆ [4.123 sta\\_info](#)
- ◆ [4.124 status](#)
- ◆ [4.125 suprates](#)
- ◆ [4.126 tkip](#)
- ◆ [4.127 tkip\\_countermeasures](#)
- ◆ [4.128 tsc](#)
- ◆ [4.129 tssi](#)
- ◆ [4.130 txant](#)
- ◆ [4.131 txpathpwr](#)
- ◆ [4.132 txpwr](#)
- ◆ [4.133 txpwr1](#)
- ◆ [4.134 txpwrlimit](#)
- ◆ [4.135 ucflags](#)
- ◆ [4.136 upgrade](#)
- ◆ [4.137 ver](#)
- ◆ [4.138 wake](#)
- ◆ [4.139 wds](#)
- ◆ [4.140 wds\\_remote\\_mac](#)
- ◆ [4.141 wds\\_wpa\\_role](#)
- ◆ [4.142 wds\\_wpa\\_role\\_old](#)
- ◆ [4.143 wep](#)
- ◆ [4.144 wepstatus](#)
- ◆ [4.145 wet](#)
- ◆ [4.146 wsec](#)
- ◆ [4.147 wsec\\_test](#)
- ◆ [4.148 wme](#)
- ◆ [4.149 wsec\\_restrict](#)
- ◆ [4.150 wpa\\_auth](#)
- [5 External links](#)

## Usage

Most of the wireless options can be accessed using the program "wl" via the console window. This program has many more options than our WRT is able to process. Some can only be used in Client Mode and others only in Access Point (AP) Mode.

```
Usage: wl [-a|i <adapter>] [-hu] <command> [arguments]
-a, -i      adapter name or number
-h, -u      this message
```

Examples:

```
~ # wl ssid
~ # wl txpwr1 -o -m 35
```

**Note** For NEWD-2 builds, you must specify the interface when performing a command.

Example:

```
~ # wl -i eth1 status
```

If you have multiple radios, then each one will have it's own interface, i.e eth2, eth3

## Common Tasks

### Toggle radio on/off, restart, etc

- ~ # wl up

reinitialize adapter and mark as up

- ~ # wl down

reset adapter and mark as down

- ~ # wl out

mark adapter down but do not reset hardware.

- ~ # wl radio

toggle radio on/off

```
~ # wl radio
radio is on (WL_RADIO_SW_DISABLE 0 WL_RADIO_HW_DISABLE 0)
~ # wl radio off
~ # wl radio
radio is off (WL_RADIO_SW_DISABLE 1 WL_RADIO_HW_DISABLE 0)
~ # wl radio on
```

- ~ # wl restart

restart the driver. (Driver must already be down)

## Alphabetical list of commands

### aciargs

Get/Set various aci tuning parameters. Choices are:

```
enter: CRS glitch trigger level to start detecting ACI
exit:  CRS glitch trigger level to exit ACI mode
glitch Seconds interval between ACI scans when glitchcount is continuously high
spin:  Num microsecs to delay between rssi samples
Usage: wl aciargs [enter x][exit x][spin x][glitch x]
```

### addwep

Set an encryption key.

```
The key must be 5, 13 or 16 bytes long, or
10, 26, 32, or 64 hex digits long. The encryption algorithm is
automatically selected based on the key size. keytype is accepted
only when key length is 16 bytes/32 hex digits and specifies
whether AES-OCB or AES-CCM encryption is used. Default is ccm.
addwep <keyindex> <keydata> [ocb | ccm] [notx] [xx:xx:xx:xx:xx:xx]
```

### aes

Set AES options.

```
wl aes [options]
[on|enable|1]  enable AES
[off|disable|0] disable AES
[sw|software] perform AES in software
[hw|hardware] perform AES in hardware
```

### antdiv

Sets which antenna to use to receive on.

```
0 - force use of antenna 0
1 - force use of antenna 1
3 - automatic selection of antenna diversity
```

Note: can be used in conjunction with 'wl txant' to set the transmit antenna.

Toggle radio on/off, restart, etc

## ap

Set AP mode: 0 (STA) or 1 (AP)

## assoc

Print information about current network association. (also known as "status")

```
SSID: "XXXX"  
Mode: Managed   RSSI: -48 dBm   noise: -97 dBm   Channel: 6  
BSSID: XX:XX:XX:XX:XX:XX       Capability: ESS  
Supported Rates: [ 1(b) 2(b) 5.5 11 ]
```

## assoclist

AP only: Get the list of associated MAC addresses.

## atten

Set the transmit attenuation for B band.

```
Args: bb radio txctl1.  
auto to revert to automatic control
```

## authe\_sta\_list

Get authenticated sta mac address list

## autho\_sta\_list

Get authorized sta mac address list

## authorize

Restrict traffic to 802.1X packets until 802.1X authorization succeeds

## band

Returns or sets the current band

```
auto - auto switch between available bands (default)  
a - force use of 802.11a band  
b - force use of 802.11b band
```

## bands

Return the list of available 802.11 bands

## bssid

Get the BSSID value, error if STA and not associated

## cap

Display WL Capabilities

```
ap sta wet led wme pio 802.11d 802.11h rm cqa afterburner
```

## channel

Set the channel:

```
valid channels for 802.11b/g (2.4GHz band) are 1 through 14
valid channels for 802.11a (5 GHz band) are:
    36, 40, 44, 48, 52, 56, 60, 64,
    100, 104, 108, 112, 116, 120, 124, 128, 132, 136, 140,
    149, 153, 157, 161,
    184, 188, 192, 196, 200, 204, 208, 212, 216
```

## channel\_qa

Get last channel quality measurement

## channel\_qa\_start

Start a channel quality measurement

## clk

set board clock state. return error for set\_clk attempt if the driver is not down

```
0: clock off
1: clock on
```



## Wl\_command

### cmds

Lists all the wl commands available

*Original German* Generiert eine Kommando-Kurtzübersicht Die meisten wireless-Optionen können über das Programm "wl" auch über die Console eingestellt und überwacht werden  
Eine übersicht über alle wl- Comandos erhält man mit der Eingabe von:

```
~ # wl cmds
ver          cmds          up            down
out          clk            restart      reboot
ucflags     radio         dump         srdump
nvdump      nvset         nvget        revinfo
msglevel    PM            wake         promisc
monitor     frag          rts          cwmin
cwmax       srl           lrl          rate
mrate       infra         ap           bssid
channel     tssi         txpwr        txpwrl
txpathpwr   txpwrlimit   powerindex   atten
phyreg      radioreg     shmem        macreg
antdiv      txant        plcphdr      phytype
scbdump     ratedump     rateparam    wepstatus
primary_key addwep        rmwep        wep
tkip        aes           keys         tsc
wsec_test   tkip_countermeasures wsec_restrict eap
authorize   deauthorize   deauthenticate wsec
wpa_auth    set_pmk       scan         passive
regulatory  spect         scanresults  assoc
status      disassoc     chanlist     channels
channels_in_country curpower     scansuppress evm
rateset     roam_trigger  roam_delta   roam_scan_period
suprates    scan_channel_time scan_unassoc_time scan_home_time
scan_passive_time scan_nprobes  prb_resp_timeout channel_qa
channel_qa_start country       locale       join
ssid        mac           macmode      wds
lazywds     noise        fqacurcy     crsuprs
int         lbt          band         bands
phylist     shortslot    shortslot_override shortslot_restrict
ignore_bcns pktcnt       upgrade      gmode
gmode_protection gmode_protection_control gmode_protection_cts gmode_protection_override
legacy_erp  scb_timeout  assoclist    rssi
isup        fasttimer    slowtimer    glacialtimer
radar       rssidump     interference  aciargs
frameburst  pwr_percent  wet          dtim
wds_remote_mac wds_wpa_role_old wds_wpa_role  authe_sta_list
autho_sta_list measure_req  quiet        csa
constraint  rm_req       rm_rep       wme
sta_info    cap
~ #
```

This program has many more options than our WRT is able to process.  
Some can only be used in Client Mode and others only in Access Point (AP) Mode.

### country

Select Country code for use with 802.11d

## Wl\_command

Use either long name or abbreviation from ISO 3166.

Use 'wl country list [band(a or b)]' for the list of supported countries

### **constraint**

Send an 802.11h Power Constraint IE

Usage: wl constraint 1-255 db

### **crsuprs**

Manufacturing test: set carrier suppression mode.

carriersuprs syntax is: crsuprs <channel>  
Arg is channel number 1-14, or 0 to stop the test.

### **csa**

Send an 802.11h channel switch announcement

Usage wl csa <mode> <when (in TBTTs)> <channel>

### **cwmin**

Set the cwmin. (integer [1, 255])

### **cwmax**

Set the cwmax. (integer [256, 2047])

### **disassoc**

Disassociate from the current BSS/IBSS.

### **dtim**

Get/Set DTIM

### **dump**

print driver software state and chip registers to stdout

### **country**

## Wl\_command

```
~ # wl dump
wl0: Aug  2 2004 14:32:51 version 3.60.13.0

resets 27
perm_etheraddr 00:90:4c:xx:xx:xx cur_etheraddr 00:12:17:xx:xx:xx
board 0x101, board rev 1.0
wsec 0 auth 0 wsec_index -1 wep_algo 0
rate_override 0
antdiv_override 1 txant 1
current_bss.BSSID 00:fb:dd:xx:xx:xx
current_bss.SSID "www.fbn-dd.de (HSS)"
assoc_state 0 associated 1
~ #
```

## deauthenticate

Deauthenticate a STA from the AP with optional reason code (AP ONLY)

## deauthorize

Do not restrict traffic to 802.1X packets until 802.1X authorization succeeds

## eap

Restrict traffic to 802.1X packets until 802.1X authorization succeeds

```
0 - disable
1 - enable
```

## chanlist

Return valid channels for the current settings is nice.

## channels

Return valid channels for the current settings.

## channels\_in\_country

Return valid channels for the country specified.

```
Arg 1 is the country abbreviation
Arg 2 is the band(a or b) (editors note: only b seems to be recognized in dd-wrt v22)
```

```
Example: wl channels_in_country JP b
```

## curpower

Return current tx power settings

## evm

Start an EVM test on the given channel, or stop EVM test.

Arg 1 is channel number 1-14, or "off" or 0 to stop the test.  
Arg 2 is optional rate (1, 2, 5.5 or 11)

## frameburst

Disable/Enable frameburst mode

## fasttimer

Get/Set High frequency watchdog timeout (tx\_power) [15 sec]

## frag

Set the fragmentation threshold. (integer [256, 2346])

## fqacurcy

Manufacturing test: set frequency accuracy mode.

freqacurcy syntax is: fqacurcy <channel>  
Arg is channel number 1-14, or 0 to stop the test.

## glacialtimer

Get/Set Very Low frequency watchdog timeout (measurelo) [120 sec]

## gmode

Set the 54g Mode

LegacyB|Auto||GOnly|BDeferred|Performance|LRS|Afterburner

- LegacyB -
- Auto [default]

## WI\_command

- BDeferred -
- Performance -
- LRS - Limited Rate Support used to improve compatibility with older 802.11b cards.
- Afterburner -

### **gmode\_protection**

Get G protection mode.

```
0 - disabled
1 - enabled
```

### **gmode\_protection\_control**

Get/Set 11g protection mode control alg.

```
(0=always off, 1=monitor local association, 2=monitor overlapping BSS)
```

### **gmode\_protection\_cts**

Get/Set 11g protection type to CTS

```
(0=disable, 1=enable)
```

### **gmode\_protection\_override**

Get/Set 11g protection mode override.

```
(-1=auto, 0=disable, 1=enable)
```

### **ignore\_bcns**

AP only (G mode): Check for beacons without NONERP element (0=Examine beacons, 1=Ignore beacons)

### **int**

Interrupt Test - remember to precede by 'wl down' and follow by 'wl up'

### **interference**

Get/Set interference mitigation mode. Choices are:

```
0 = none
```

## WI\_command

```
1 = non wlan
2 = wlan manual
3 = wlan automatic
```

### **infra**

Set Infrastructure mode: 0 (ad-hoc IBSS) or 1 (managed BSS)

### **isup**

Get driver operational state (0=down, 1=up)

### **join**

Join a specified network SSID.

Join syntax is:

```
join <ssid> [key xxxxx] [imode bss|ibss] [amode open|shared|wpa|wpapsk|wpanone]
```

### **keys**

Prints a list of the current WEP keys

### **lazywds**

Set or get "lazy" WDS mode (dynamically grant WDS membership to anyone).

### **lbt**

Loopback Test - remember to precede by 'wl down' and follow by 'wl up'

### **legacy\_erp**

Get/Set 11g legacy ERP inclusion (0=disable, 1=enable)

### **locale**

OBSOLETE: use "wl country"

```
Select the country:
Worldwide
```

interference

## Wl\_command

Thailand  
Israel  
Jordan  
China  
Japan  
USA/Canada/ANZ  
Europe  
USALow  
JapanHigh  
All

### **lrl**

lrl Set the long retry limit. (integer [1, 255])

### **mac**

Set or get the list of source MAC address matches.

```
wl mac xx:xx:xx:xx:xx:xx [xx:xx:xx:xx:xx:xx ...]  
To Clear the list: wl mac none
```

### **macmode**

Set the mode of the MAC list.

```
0 - Disable MAC address matching.  
1 - Deny association to stations on the MAC list.  
2 - Allow association to stations on the MAC list.
```

### **macreg**

Get/Set any mac registers(include IHR and SB)

```
macreg offset size[2,4] [value]
```

### **measure\_req**

Send an 802.11h measurement request.

```
Usage: wl measure_req <type> <target MAC addr>  
Measurement types are: TPC, Basic, CCA, RPI  
Target MAC addr format is xx:xx:xx:xx:xx:xx
```

## monitor

set monitor mode

```
0 - disable
1 - enable active monitor mode (interface still operates)
```

## mrata

force a fixed multicast rate:

```
valid values for 802.11a are (6, 9, 12, 18, 24, 36, 48, 54)
valid values for 802.11b are (1, 2, 5.5, 11)
valid values for 802.11g are (1, 2, 5.5, 6, 9, 11, 12, 18, 24, 36, 48, 54)
-1 (default) means automatically determine the best rate
```

- After using `wl mrata` set the multicast rate, how to check if it has been correctly setted? How to get the current mrata value? [Skygunner](#)

## msglevel

set driver console debugging message bitvector

```
type 'wl msglevel ?' for values
```

## noise

Get noise (moving average) right after tx in dBm

## nvdump

print nvram variables to stdout No function. Use **nvram show!**

## nvget

get the value of an nvram variable

## nvset

set an nvram variable

```
name=value (no spaces around '=')
```



## **passive**

Puts scan engine into passive mode

## **phylist**

Return the list of available phytypes

## **phyreg**

Get/Set a phy register.

## **radioreg**

Get/Set a radio register.

## **phytype**

Get phy type

## **pktcnt**

Get the summary of good and bad packets.

## **plcphdr**

Set the plcp header.

`"long" or "auto" or "debug"`

## **PM**

set driver power management mode:

0: CAM (constantly awake)  
1: PS (power-save)  
2: FAST PS mode

## powerindex

Set the transmit power for A band(0-63).

-1 - default value

## prb\_resp\_timeout

Get/Set probe response timeout

## primary\_key

Set or get index of primary key

## promisc

set promiscuous mode ethernet address reception

0 - disable  
1 - enable

## pwr\_percent

Get/Set power output percentage

## quiet

Send an 802.11h quiet command.

Usage: wl quiet <TBTTs until start>, <duration (in TUs)>, <offset (in TUs)>

## radar

Enable/Disable radar

## radio

When used without arguments, toggles radio on/off.

Arguments:

up - reinitialize adapter and mark as up  
down - reset adapter and mark as down  
out - mark adapter down but do not reset hardware

## WI\_command

restart -

Examples:

```
wl radio
wl radio up
```

## rate

force a fixed rate:

```
valid values for 802.11a are (6, 9, 12, 18, 24, 36, 48, 54)
valid values for 802.11b are (1, 2, 5.5, 11)
valid values for 802.11g are (1, 2, 5.5, 6, 9, 11, 12, 18, 24, 36, 48, 54)
-1 (default) means automatically determine the best rate
```

## ratedump

print driver rate selection tunables and

```
per-scb state to stdout, valid scb values are
0 through NSCB-1
```

## rateparam

set driver rate selection tunables

```
arg 1: tunable id
arg 2: tunable value
```

## rateset

Returns or sets the supported and basic rateset.

```
(b) indicates basic
With no args, returns the rateset. Args are
rateset "default" | "all" | <arbitrary rateset>
    default - driver defaults
    all - all rates are basic rates
    arbitrary rateset - list of rates
List of rates are in Mbps and each rate is optionally followed
by "(b)" or "b" for a Basic rate. Example: 1(b) 2b 5.5 11
At least one rate must be Basic for a legal rateset.
```

## reboot

Reboot platform

## revinfo

get hardware revision information

```
~ # wl revinfo
vendorid: 0x14e4
deviceid: 0x4320
radiorev: 0x22050000
chipnum: 0x4712
chiprev: 0x1
corerev: 0x7
boardid: 0x101
boardvendor: 0x14e4
boardrev: 0x10
driverrev: 0x33c0d00
ucoderev: 0x1180016
bus: 0x0
~ #
```

## regulatory

Get/Set regulatory domain mode (802.11d). Driver must be down.

## roam\_delta

Set the roam candidate qualification delta. (integer)

## rm\_rep

Get current radio measurement report

## rm\_req

Request a radio measurement of type basic, cca, or rpi

specify a series of measurement types each followed by options.

example: `wl rm_req cca -c 1 -d 50 cca -c 6 cca -c 11`

Options:

- t n numeric token id for measurement set or measurement
- c n channel
- d n duration in TUs (1024 us)
- p parallel flag, measurement starts at the same time as previous

Each measurement specified uses the same channel and duration as the previous unless a new channel or duration is specified.

**rmwep**

Remove the encryption key at the specified key index.

**roam\_scan\_period**

Set the roam candidate qualification delta. (integer)

**roam\_trigger**

Set the roam trigger RSSI threshold. (integer)

**rss**

Get the current RSSI (signal strength) value.

In AP mode, you must specify the mac address of the wireless card who's signal signal you wish to monitor. You can use `wl assoclist` to get the client mac list.

```
wl rssi <MAC_ADDR_OF_CLIENT>
```

In client mode there is no need to specify the MAC address of the AP as it will just use the AP that you are associated with.

See also: Signal Strength on the [Script Examples](#) page for a one line script to display a bar graph of the current signal level.

**ssidump**

Dump rssi values from aci scans

**rts**

Set the RTS threshold. (integer [0, 2347])

**scan\_channel\_time**

Get/Set scan channel time

## **scan\_home\_time**

Get/Set scan home channel dwell time

## **scan\_nprobes**

Get/Set scan parameter for number of probes to use per channel scanned

## **scan\_passive\_time**

Get/Set passive scan channel dwell time

## **scan\_unassoc\_time**

Get/Set unassociated scan channel dwell time

## **scanresults**

Return results from last scan.

## **scansuppress**

Suppress all scans for testing.

```
0 - allow scans  
1 - suppress scans
```

## **scb\_timeout**

AP only: inactivity timeout value for authenticated stas

## **scbdump**

print driver scb state to stdout

## **scan**

Initiate a scan.

```
Default an active scan across all channels for any SSID.  
Optional arg: SSID, the SSID to scan.
```

## Wl\_command

Options:

-s S, --ssid=S	SSID to scan
-t ST, --scan_type=ST	[active passive] scan type
--bss_type=BT	[bss/infra ibss/adhoc] bss type to scan
-b MAC, --bssid=MAC	particular BSSID MAC address to scan, xx:xx:xx:xx:xx:xx
-n N, --nprobes=N	number of probes per scanned channel
-a N, --active=N	dwll time per channel for active scanning
-p N, --passive=N	dwll time per channel for passive scanning
-h N, --home=N	dwll time for the home channel between channel scans
-c L, --channels=L	comma or space separated list of channels to scan

NOTE: 'wl scan' does not work in AP Mode. To scan please use:

```
wl ap 0
wl scan
wl scanresults
wl ap 1 (back to AP mode)
```

## set\_pmk

Set passphrase for PMK in driver-resident supplicant.

## shmem

Get/Set a shared memory location.

## shortslot

Get/Set 11g Short Slot Timing mode. (-1=auto, 0=long, 1=short)

## shortslot\_override

Get/Set 11g Short Slot Timing mode override. (-1=auto, 0=long, 1=short)

## shortslot\_restrict

Get/Set AP Restriction on associations for 11g Short Slot Timing capable STAs.

```
0 - Do not restrict association based on ShortSlot capability
1 - Restrict association to STAs with ShortSlot capability
```

## slowtimer

Get/Set Low frequency watchdog timeout (nrssislope) [60 sec]

## spect

Get/Set 802.11h Spectrum Management mode.

- 0 - Off
- 1 - Loose interpretation of spec - may join non-11h APs
- 2 - Strict interpretation of spec - may not join non-11h APs

## srdump

print contents of SPROM to stdout (Functions neither in AP, nor nor in Client mode ~or something~)???  
*Original German: (weder im AP, noch im Client-Modus eine Ausgabe)???*

## srl

srl Set the short retry limit. (integer [1, 255])

## ssid

Set or get the current SSID.

Setting will initiate an assoication attempt if in infrastructure mode,  
or join/creation of an IBSS if in IBSS mode,  
or creation of a BSS if in AP mode.

## sta\_info

wl sta\_info <xx:xx:xx:xx:xx:xx>

## status

Print information about current network association. (also known as "assoc")

## suprates

Returns or sets the 11g override for the supported rateset.

With no args, returns the rateset. Args are a list of rates,  
or 0 or -1 to specify an empty rateset to clear the override.  
List of rates are in Mbps, example: 1 2 5.5 11



## tkip

Set TKIP options.

```
wl tkip [options]
[on|enable|1]    enable TKIP
[off|disable|0] disable TKIP
[sw|software]   perform TKIP in software
[hw|hardware]   perform TKIP in hardware
```

## tkip\_countermeasures

Enable or disable TKIP countermeasures (TKIP-enabled AP only)

```
0 - disable
1 - enable
```

## tsc

Print Tx Sequence Counter for key at specified key index.

## tssi

Get the tssi value from radio

## txant

Set the transmit antenna

```
0 - force use of antenna 0
1 - force use of antenna 1
3 - use the RX antenna selection that was in force during
    the most recently received good PLCP header
```

Note: can be used in conjunction with 'wl antdiv' to set the receive antenna

## txpathpwr

Turn the tx path power on or off on 2050 radios

## txpwr

Set transmit power in milliwatts. Range [1, 84]. (Deprecated: Use txpwr1 instead)

## WI\_command

This can be set to a value above 84, but it may damage your hardware especially over prolonged use -- use with caution.

### txpwr1

Set tx power in in various units. Choose one of (default: dbm):

```
-d dbm units
-q quarter dbm units
-m milliwatt units
```

Can be combined with:

```
-o turn on override to disable regulatory and other limitations
```

### txpwrlimit

Return current tx power limit

### ucflags

Get/Set ucode flags

### upgrade

Upgrade the firmware on an embedded device

### ver

Version information

```
~ # wl ver
wl:      3.60 RC13.0
        wl0: Aug  2 2004 14:32:51 version 3.60.13.0
~ #
```

### wake

set driver power-save mode sleep state:

```
0: core-managed
1: awake
```

## wds

Set or get the list of WDS member MAC addresses.

```
Set using a space separated list of MAC addresses.  
wl wds xx:xx:xx:xx:xx:xx [xx:xx:xx:xx:xx:xx ...]
```

## wds\_remote\_mac

Get WDS link remote endpoint's MAC address

## wds\_wpa\_role

Get/Set WDS link local endpoint's WPA role

## wds\_wpa\_role\_old

Get WDS link local endpoint's WPA role (old)

## wep

Set WEP options.

```
wl wep [options]  
[on|enable|1] enable WEP  
[off|disable|0] disable WEP  
[sw|software] perform WEP in software  
[hw|hardware] perform WEP in hardware
```

## wepstatus

Set or Get WEP status

```
wepstatus [on|off]
```

## wet

Get/Set wireless ethernet bridging mode

## wsec

Wireless security bit vector

## Wl\_command

```
1 - WEP enabled
2 - TKIP enabled
4 - AES enabled
8 - WSEC in software
```

### wsec\_test

Generate wsec errors

```
wsec_test <test_type> <keyindex|xx:xx:xx:xx:xx:xx>
type 'wl wsec_test ?' for test_types
```

### wme

Set WME (Wireless Multimedia Extensions) mode (0=off, 1=on)

### wsec\_restrict

Drop unencrypted packets if WSEC is enabled

```
0 - disable
1 - enable
```

### wpa\_auth

WPA authorization mode

```
[none|0]      none
[unspecified|1] WPA 802.1X
[psk|2]      WPA PSK
[disable|255] disable WPA
```

## External links

- [\[1\]](#) broadcom-wl-drv (auf Englisch)
- [\[2\]](#) broadcom-wl package(auf Englisch)