Technical Information

Broadcom WL Tool

802.11 WL Tool

Broadcom Confidential

80211-TI300-R

5300 California Avenue • Irvine, CA 92617 • Phone: 949-926-5000 • Fax: 949-926-5203

07/13/07

REVISION HISTORY

Revision	*Date*	*Change Description*
80211-TI300-R	07/13/07	Initial release

Broadcom Confidential

Broadcom Corporation
5300 California Avenue
Irvine, CA 92617

© 2007 by Broadcom Corporation
All rights reserved
Printed in the U.S.A.

Broadcom®, the pulse logo, Connecting everything®, and the Connecting everything logo are among the trademarks of Broadcom Corporation and/or its affiliates in the United States, certain other countries and/or the EU. Any other trademarks or trade names mentioned are the property of their respective owners.

TABLE OF CONTENTS

Broadcom Confidential

Broadcom Confidential

Broadcom Confidential

Broadcom Confidential

Broadcom Confidential

Broadcom Confidential

This document describes the different commands available in the Broadcom WLAN client utility.

Before you can begin using the utility to start a build, you must have the following tools available/installed on your Linux build machine:

- New version of the Broadcom client driver Bills 4.10.47 or later
- The latest Broadcom BCM43XX WLAN adapter
- Windows® or Linux®operating system

COMMANDS—GENERAL

SYNTAX

The command syntax is as follows:

```
wl [-a|i <adapter>] [-h] [-d|u|x] <command> [arguments]
```

where

-h	is this message and command descriptions.
-h [cmd]	is the command description for cmd.
-a, -i	is the adapter name or number.
-d	is the output format signed integer.
-u	is the output format unsigned integer.
-x	is the output format hexadecimal.

The [h,u] option is used only to print out the Help topic, and the [a |] option is needed only if you have multiple adapters.

OPTIONS

To view the list of options for the utility, run **wl**.

Broadcom Confidential

COMMANDS—DESCRIPTION

The following lists the available individual commands/options and describes what they do and how they are used.

VERSION INFORMATION AND LIST OF COMMANDS

ver

Returns the version information of the utility.

Syntax: **wl ver**

Example: `4.150 RC6.0 wl0: May 8 2007 20:35:49 version 4.150.6.0`

cmds

Generates a short list of available commands.

Syntax: **wl cmds**

Example:

a_rate	counters	nvset	set_pmk
a_mrate	csscantimer	nvget	scan
ap	closed	nvram_get	spect
atten	closednet	noise	scanresults
ampdu_tid	cac_addts	nphy_test_tssi	status
ampdu_send_addba	cac_delts	nphy_test_tssi_offs	scansuppress
ampdu_send_delba	cac_tslist	nphy_rssiant	suprates
ampdu_clear_dump	cac_tspec	nvotpw	scan_channel_time
antdiv	down	nrate	scan_unassoc_time
addwep	dump	nphy_caldump	scan_home_time
authorize	dfs_channel_forced	nphy_antsel	scan_passive_time
auth	dpt_deny	out	scan_nprobes
assoc	dpt_endpoint	otpw	ssid
autocountry_default	deauthorize	otpdump	shortslot
assoclist	deauthenticate	PM	shortslot_override
arpoe	disassoc	promisc	shortslot_restrict
arp_ol	dfs_status	powerindex	scb_timeout
arp_peerage	dtim	phyreg	slowtimer
arp_table_clear	del_ie	plcphdr	sta_info
arp_hostip	diag	phytype	staname
arp_hostip_clear	eap	primary_key	tssi

Broadcom Confidential

arp_stats
arp_stats_clear
authe_sta_list
autho_sta_list
assoc_pref
add_ie
assoc_info
autochannel
abminrate
ap_isolate
apname
bg_rate
bg_mrate
bssid
bssmax
band
bands
bi
bcmerrorstr
bss
cmds
clk
cwmin
cwmax
channel
cur_mcsset
chanspecs
chanspec
cur_etheraddr
chanlist
channels
channels_in_country
curpower
channel_qa
channel_qa_start
country
crsuprs
evm
eventing
event_msgs
eap_restrict
frag
fqacurcy
fasttimer
frameburst
freqtrack
gpioout
gmode
gmode_protection
gmode_protection_control
gmode_protection_override
glacialtimer
infra
iscan_s
iscan_c
iscanresults
ignore_bcns
isup
interference
join
join_pref
keys
lrl
lazywds
longtrain
legacy_erp
lifetime
list_ie
msglevel
monitor
mrate
macreg
mac
macmode
passive
prb_resp_timeout
phylist
pktcnt
protection_control
phy_rssi_ant
pwr_percent
pmkid_info
phy_rxiqest
pkteng_start
pkteng_stop
quiet
restart
reboot
radio
revinfo
rts
rate
radioreg
rateparam
rmwep
regulatory
rateset
roam_trigger
roam_delta
roam_scan_period
rssi
radar
radarargs
radarargs40
rm_req
rm_rep
reinit
rand
reset_d11cnts
rifs
srclear
txpwr
txpwr1
txpathpwr
txpwrlimit
txant
tsc
tkip_countermeasures
txinstpwr
toe
toe_ol
toe_stats
toe_stats_clear
txfifo_sz
up
ucflags
ucantdiv
upgrade
ver
wake
wepstatus
wsec_test
wsec_restrict
wsec
wpa_auth
wpa_cap
wds
wet
wds_remote_mac
wds_wpa_role_old
wds_wpa_role
wme
wme_ac
wme_apsd
wme_apsd_sta
wme_dp
wme_counters
wme_clear_counters

Broadcom Confidential

csa	measure_req	srdump
constraint	malloc_dump	srwrite
cap	mimo_txbw	srl
chan_info	nvram_dump	shmem

list

Lists all installed wireless adapters.

Syntax: **wl list**

Example:

```
1: wl1 MAC: 00:90:4B:7A:7A:AC
```

xlist

Lists all installed network adapters.

Syntax: **wl xlist**

Example:

```
0: ??0 802.3 0035 {0159A4F2-4EC8-4F75-8DB8-A74547B9D1A5} MAC:
00:0F:1F:CE:91:AB
1: wl1 wireless 0012 {2611C167-BB4F-40FA-A6A7-890348F9E104} MAC:
00:90:4B:7A:7A:AC
```

ASSOCIATION CONTROL

join

Joins a specified network.

Syntax:

join <name|ssid> [key xxxxx] [imode bss|ibss] [amode open|shared|auto|wpa|wpapsk|wpanone|wpa2|wpa2psk]

Example: join Broadcom imode infra amode open

If the AP is not configured with Wired Equivalent Privacy (WEP) security, no WEP key is required. Otherwise, specify either

wep xxxx

-or-

wepkey xxxx

The amode or authentication mode choices are open or shared.

Broadcom Confidential

Note:

- For infrastructure mode, the choices are `bss`, `managed`, or `infra`, and for ad hoc mode, the choices are `ibss` or `ad-hoc`.
- IEEE Std 802.11 supports two subtypes of network authentication services: open system and shared key. Under open system authentication, any wireless station can request authentication. The station that needs to authenticate with another wireless station sends an authentication management frame that contains the identity of the sending station. The receiving station then sends back a frame that indicates whether it recognizes the identity of the sending station. Under shared key authentication, each wireless station is assumed to have received a secret shared key over a secure channel that is independent from the 802.11 wireless network communications channel.

up

Reinitializes and marks the adapter up (operational).

Syntax: **wl up**

This command makes the interface operational. It does all the necessary initialization to bring up the interface. Some of the tasks associated with this command are:

- Configure PCI/PCMCIA here to allow manufacturer hot-swap: down, hot-swap (chip power cycle), up.
- Read the PHY revision.
- Set the soft interrupt mask.
- Bring the interface up in each frequency band.
- Initialize the default rate, channel, and type-dependent information.
- Initialize the basic rate look-up.
- Save, suspend, disable interrupts, and turn the radio off.
- Start a one-second watchdog.
- Start the activity LED timer.

down

Resets and marks the adapter down (disabled).

Syntax: **wl down**

This command disables the interface. Some of the tasks associated with this command are:

- Disassociate.
- Turn the radio off.
- Cancel the watchdog timer.
- Cancel the activity timer.
- Cancel any active scan.
- Cancel any IBSS timer.
- Cancel any association timer.
- Flush the TX control queue.
- Reclaim the SCBS.
- If an AP, flush PS-POLL response (MSDU) packet queues and also flush PSPOLL.

Broadcom Confidential

- Response (MPDU) packet queues.
- Restore to a known good default state.

out

Marks the adapter down, but does not reset the hardware (disabled).

Syntax: **wl out**

On dual-band cards, the card must be band-locked before use.

INITIALIZING/RESTARTING

restart

Restarts the driver.

Syntax: **wl restart**

The driver must already be down (you should do a **wl down** before restarting the driver). Otherwise, the following message is returned:

```
wl restart
: Not down
```

RADIO CONTROL

radio

Turns the radio on or off using a software switch.

Syntax: **wl radio** `on/off`

Typing **wl radio** returns the current state of the radio. For example, 0x0000 when ON or 0x0005 when OFF, and so on.

DEBUGGING/STATUS/CONFIGURATION

eventing

Set/get the 128-bit hexadecimal filter bitmask for MAC event reporting up to application layer.

Note: The **eventing** command has been deprecated. Use **event_msgs** instead.

event_msgs

Sets/gets the 128-bit hexadecimal filter bit mask for MAC event reporting (through packet indications).

Default: 0

Broadcom Confidential

Takes a 128-bit vector, which selectively enables or disables the reporting of MAC events through the packet data path. For example, setting bit locations 0 and 3 would enable the reporting of WLC_E_SET_SSID and WLC_E_AUTH event messages, and so on.

Syntax: **wl event_msgs**

Event messages bit vector:

```
WLC_E_SET_SSID          0     /* indicates status of set SSID */
WLC_E_JOIN              1     /* differentiates join IBSS from found (WLC_E_START) IBSS */
WLC_E_START             2     /* STA founded an IBSS or AP started a BSS */
WLC_E_AUTH              3     /* 802.11 AUTH request */
WLC_E_AUTH_IND          4     /* 802.11 AUTH indication */
WLC_E_DEAUTH            5     /* 802.11 DEAUTH request */
WLC_E_DEAUTH_IND        6     /* 802.11 DEAUTH indication */
WLC_E_ASSOC             7     /* 802.11 ASSOC request */
WLC_E_ASSOC_IND         8     /* 802.11 ASSOC indication */
WLC_E_REASSOC           9     /* 802.11 REASSOC request */
WLC_E_REASSOC_IND       10    /* 802.11 REASSOC indication */
WLC_E_DISASSOC          11    /* 802.11 DISASSOC request */
WLC_E_DISASSOC_IND      12    /* 802.11 DISASSOC indication */
WLC_E_QUIET_START       13    /* 802.11h Quiet period started */
WLC_E_QUIET_END         14    /* 802.11h Quiet period ended */
WLC_E_BEACON_RX         15    /* BEACONS received/lost indication */
WLC_E_LINK              16    /* generic link indication */
WLC_E_MIC_ERROR         17    /* TKIP MIC error occurred */
WLC_E_NDIS_LINK         18    /* NDIS style link indication */
WLC_E_ROAM              19    /* roam attempt occurred: indicate status & reason */
WLC_E_TXFAIL            20    /* change in dot11FailedCount (txfail) */
WLC_E_PMKID_CACHE       21    /* WPA2 pmkid cache indication */
```

Broadcom Confidential

```
WLC_E_RETROGRADE_TSF   22     /* current AP's TSF value went backward */
WLC_E_PRUNE            23     /* AP was pruned from join list for reason */
WLC_E_AUTOAUTH         24     /* report AutoAuth table entry match for join attempt */
WLC_E_EAPOL_MSG        25     /* Event encapsulating an EAPOL message */
WLC_E_SCAN_COMPLETE    26     /* Scan results are ready or scan was aborted */
WLC_E_ADDTS_IND        27     /* indicate to host addts fail/success */
WLC_E_DELTS_IND        28     /* indicate to host delts fail/success */
WLC_E_BCNSENT_IND      29     /* indicate to host of beacon transmit */
WLC_E_BCNRX_MSG        30     /* Send the received beacon up to the host */
WLC_E_BCNLOST_MSG      31     /* indicate to host loss of beacon */
WLC_E_ROAM_PREP        32     /* before attempting to roam */
WLC_E_PFN_NET_FOUND    33     /* PFN network found event */
WLC_E_PFN_NET_LOST     34     /* PFN network lost event */
```

counters

Returns the driver counter values.

Syntax: **wl counters**

Example return:

```
txframe 92289 txbyte 7637260 txretrans 830936 txerror 0 rxframe 90957 rxbyte 665
3890 rxerror 17
txprshort 4398 txdmawar 0 txnobuf 0 txnoassoc 0 txchit 116 txcmiss 92173
reset 14926 txserr 0 txphyerr 1 txphycrs 0 txfail 689
d11_txfrag 509883 d11_txmulti 12 d11_txretry 267653 d11_txretrie 266858
d11_txrts 0 d11_txnocts 0 d11_txnoack 829661 d11_txfrmsnt 284515
rxcrc 3824120 rxnobuf 0 rxnondata 0 rxbadds 0 rxbadcm 0 rxdup 1017 rxfragerr 0
rxrunt 14 rxgiant 0 rxnoscb 0 rxbadproto 0 rxbadsrcmac 3
d11_rxfrag 2280968 d11_rxmulti 1591082 d11_rxundec 0
rxctl 2140668 rxbadda 0 rxfilter 0
rxuflo: 0 0 0 0 0 0
txallfrm 1407603 txrtsfrm 0 txctsfrm 17113 txackfrm 217150
txdnlfrm 0 txbcnfrm 0 txtplunfl 0 txphyerr 1
```

Broadcom Confidential

```
txfunfl: 0 0 0 0 0 0

tkipmicfaill 0 tkipicverr 0 tkipcntrmsr 0
tkipreplay 0 ccmpfmterr 0 ccmpreplay 0
ccmpundec 0 fourwayfail 0 wepundec 0
wepicverr 0 decsuccess 0 rxundec 0

rxfrmtoolong 828457 rxfrmtooshrt 8166 rxinvmachdr 1374201 rxbadfcs 3824120
rxbadplcp 5332054 rxcrsglitch 36254836 rxstrt 16714715 rxdfrmucastmbss 92241
rxmfrmucastmbss 124909 rxcfrmucast 284475 rxrtsucast 0 rxctsucast 0
rxackucast 284475 rxdfrmocast 82398 rxmfrmocast 1280135 rxcfrmocast 450757
rxrtsocast 19235 rxctsocast 137269 rxdfrmmcast 486848 rxmfrmmcast 9259123
rxcfrmmcast 4 rxbeaconmbss 1310371 rxdfrmucastobss 0 rxbeaconobss 7698574
rxrsptmout 827723 bcntxcancl 0 rxf0ovfl 0 rxf1ovfl 0
rxf2ovfl 0 txsfovfl 0 pmqovfl 0
rxcgprqfrm 0 rxcgprsqovfl 0 txcgprsfail 0 txcgprssuc 0
prs_timeout 0 rxnack 0 frmscons 0 txnack 0 txglitch_nack 0
txburst 0 txphyerror 0
txchanrej 237
```

staname

Gets/sets the station name.

Syntax: **wl staname**

Returns your machine name. If the STA name has not been set by the operating system, a **get staname** command returns a NULL string. The maximum STA name length (set/get) is 15 bytes.

apname

Gets the current associated AP name. If the client is not associated to an AP, a stale AP name might be returned. The maximum AP name length is 15 bytes.

Syntax: **wl apname**

dump

Prints the driver software state and chip registers to STDOUT.

Syntax: **wl dump**

Example return:

```
wl0: May 8 2007 20:35:49 version 4.150.6.0
```

Broadcom Confidential

```
resets 14932
perm_etheraddr 00:90:4c:99:02:35 cur_etheraddr 00:90:4c:99:02:35
board 0x46d, board rev 4.11
rate_override: A 0, B 0
antdiv_override -1 (3 default) txant 3

BSS Config 0: "foo"
enable 1 up 1 wlif 0x00000000 ""
wsec 0x0 auth 0 wsec_index -1 wep_algo 0

current_bss.BSSID 00:10:18:91:0e:56
current_bss.SSID "foo"
assoc_state 0 sta_associated 1
```

srdump

Prints the contents of SPROM to STDOUT (dumps 64 16-bit words of the SROM present on-board). For details of the individual locations, check the Broadcom SROM memory map for that specific design. Memory maps are different, depending on the type of the design (for example, Mini PCI, Cardbus, PCMCIA, and so on).

Syntax: **wl srdump**

Example return:

```
0x3001  0x0000  0x046d  0x14e4  0x4329  0x8000  0x0002  0x0000
 srom[008]:  0x1000  0x1800  0x0000  0x0000  0xffff  0xffff  0xffff  0xffff
 srom[016]:  0xffff  0xffff  0xffff  0xffff  0xffff  0xffff  0xffff  0xffff
 srom[024]:  0xffff  0xffff  0xffff  0xffff  0xffff  0xffff  0xffff  0xffff
 srom[032]:  0x5372  0x004b  0x0200  0x0000  0x0003  0x0000  0x0090  0x4c99
 srom[040]:  0x0235  0x0000  0x0000  0xffff  0xffff  0xffff  0x0007  0x0202
 srom[048]:  0xff02  0x4a4a  0x5b5b  0xffff  0xffff  0xffff  0xffff  0xffff
 srom[056]:  0xffff  0xff3c  0xffff  0xffff  0xffff  0xffff  0xffff  0xffff
 srom[064]:  0x0000  0x0000  0x0000  0x0000  0x0000  0xffff  0xffff  0xffff
 srom[072]:  0xffff  0xffff  0xffff  0xffff  0xffff  0xffff  0xffff  0xffff
 srom[080]:  0xffff  0xffff  0xffff  0xffff  0xffff  0xffff  0xffff  0x0000
 srom[088]:  0x0000  0x0000  0x0000  0x0000  0xffff  0xffff  0xffff  0xffff
 srom[096]:  0xffff  0xffff  0xffff  0xffff  0xffff  0xffff  0xffff  0xffff
 srom[104]:  0xffff  0xffff  0xffff  0xffff  0xffff  0xffff  0xffff  0xffff
 srom[112]:  0xffff  0xffff  0xffff  0xffff  0xffff  0xffff  0xffff  0xffff
 srom[120]:  0xffff  0xffff  0xffff  0xffff  0xffff  0xffff  0xffff  0xffff
```

Broadcom Confidential

```
srom[128]:  0xffff  0xffff  0xffff  0xffff  0xffff  0xffff  0xffff  0xffff
srom[136]:  0xffff  0xffff  0xffff  0xffff  0xffff  0xffff  0xffff  0xffff
srom[144]:  0xffff  0xffff  0xffff  0xffff  0xffff  0xffff  0xffff  0xffff
srom[152]:  0xffff  0xffff  0xffff  0xffff  0x0000  0x0000  0x0000  0xffff
srom[160]:  0xffff  0xffff  0xffff  0xffff  0xffff  0x0000  0x0000  0x0000
srom[168]:  0x0000  0x0000  0x0000  0x0000  0x0000  0xffff  0xffff  0xffff
srom[176]:  0xffff  0xffff  0xffff  0xffff  0xffff  0xffff  0xffff  0xffff
srom[184]:  0xffff  0xffff  0xffff  0xffff  0xffff  0xffff  0xffff  0xffff
srom[192]:  0xffff  0xffff  0xffff  0xffff  0xffff  0x0000  0x0000  0x0000
srom[200]:  0x0000  0xffff  0xffff  0xffff  0xffff  0xffff  0xffff  0xffff
srom[208]:  0xffff  0xffff  0xffff  0xffff  0xffff  0xffff  0xffff  0xffff
srom[216]:  0xffff  0xffff  0xffff  0xaf04
```

nvset

Sets an NVRAM variable.

Syntax: **wl nvset** <name = value>

nvget

Gets the value of an NVRAM variable.

Syntax: **wl nvget** <name>

nvram_get

Gets the value of an NVRAM variable (only available in manufacturing test builds).

Syntax: **wl nvget** <name>

revinfo

Gets the hardware revision information.

Syntax: **wl revinfo**

Example return:

vendorid 0x14e4
deviceid 0x4329
radiorev 0x42055000
chipnum 0x4321
chiprev 0x1
corerev 0xb
boardid 0x46d

Broadcom Confidential

boardvendor 0x14e4

boardrev 0x4b

driverrev 0x4960c00

ucoderev 0x19a00d8

bus 0x1

msglevel

Sets the driver console debugging message-bit vector.

Syntax: **wl msglevel** N

N can be a list of numbers such as the following:

0x0001 error, err - error

0x0002 trace - traces the code

0x0004 prhdrs - prints the headers for the routines

0x0008 prpkt - dumps the individual packets

.....

0x0040 oid - dumps the OIDs as they are called

0x0080 rate - rate related information

0x0100 assoc, as - association related message/routines

promisc

Sets Promiscuous mode Ethernet address reception.

Syntax: **wl promisc** <*value*>

where *value*

0 = Disable

1 = Enable

When set, the address filter accepts all received frames. When cleared, the address filter accepts only those frames that match the BSSID or local MAC address.

monitor

Sets the adapter to Monitor mode.

Syntax: **wl monitor** <*value*>

where *value*

0 = Disable

1 = Enable active monitor mode (interface still operates)

Broadcom Confidential

When set, the address filter accepts all received frames (when cleared, the address filter accepts only those frames that match the BSSID or local MAC address), accepts all received control frames that are accepted by the address filter, and accepts all beacon and probe response frames without regard to the source address.

phylist

Returns the list of available PHY types.

Syntax: **wl phylist**

Example return:

```
g
```

phyreg

Gets/sets a PHY register.

Syntax: **wl phyreg** *offset* [`value`] [`band`]

radioreg

Gets/sets a radio register.

Syntax: **wl radioreg** *offset* [`value`] [`band`]

shmem

Gets/sets a shared memory location.

Syntax: **wl shmem** *offset* [`value`] [`band`]

macreg

Gets/sets any MAC register (include IHR and SB).

Syntax: **wl macreg** *offset size* [2, 4] [`value`] [`band`]

plcphdr

Gets/sets the PLCP header.

Syntax: **wl plcphdr** <`option`>

where option = long, auto, or debug

Broadcom Confidential

phytype

Gets/sets the PHY type.

Syntax: **wl phytype**

PHY Type A	0
PHY Type B	1
PHY Type G	2
PHY Type N	4
UNKNOWN PHY	15

sta_info

Gets STA information.

This command is AP specific and gets the rate-set of each associated wireless station.

Syntax: **wl sta_info** <MAC address of the WLAN interface of the AP>

Example:

wl sta_info 00:90:4B:7A:7A:AC

STA 00:90:4B:7A:7A:AC:

rateset [1 2 5.5 6 9 11 12 18 24 36 48 54]

idle 0 seconds

in network 38 seconds

state: AUTHENTICATED ASSOCIATED

flags 0x1b: BRCM WME

cap

Driver capabilities.

Syntax: **wl cap**

scb_timeout

Inactivity time-out value for authenticated STAs.

Syntax: **wl scb_timout** <value>

AP only: inactivity time-out value for authenticated STAs. Running **wl scb_timout** without any arguments displays the following string on the serial console that is attached to the AP/wireless router (from the busybox hash prompt #).

Broadcom Confidential

shortslot

Gets the current IEEE 802.11g Short Slot Timing mode.

Syntax: **wl shortslot**

Options are:

0 = long

1 = short

shortslot_override

Gets/sets the IEEE 802.11g Short Slot Timing mode override.

Syntax: **wl shortslot_override**

Options are:

1 = auto

0 = long

1 = short

shortslot_restrict

Gets/sets the AP restriction on assoc for 802.11g short slot.

Syntax: **wl shortslot_restrict**

Timing capable STAs:

0 = do not restrict association based on short-slot capability

1 = restrict association to STAs with short-slot capability

ignore_bcns

AP only (802.11g mode):

Checks for beacons without the NONERP element.

Syntax: **wl ignore_bcns**

0 = examine beacons

1 = ignore beacons

This command returns whether the AP will ignore beacons from other BSSs on the same channel. If the capability to ignore beacons is disabled, and the AP detects another BSS on the same channel, the AP drops to long-slot timing if the other BSS supports only long-slot timing. In accord with the IEEE 802.11g draft specification, an IEEE 802.11g AP can ignore these beacons.

Broadcom Confidential

pktcnt

Gets the summary of good and bad packets.

Syntax: **wl pktcnt**

Receive: good packet 0, bad packet 0

Transmit: good packet 0, bad packet 0

upgrade

Upgrades the firmware on an embedded device.

Syntax: **wl upgrade**

infra

Sets the device to operate in either Infrastructure mode or ad hoc mode.

Syntax: **wl infra** <value>

where value is one of the following:

0 = IBSS (ad hoc)

1 = BSS (infrastructure)

ap

Sets the AP mode.

Syntax: **wl ap** <value>

where *value* is one of the following:

0 = STA

1 = AP

bssid

Gets the BSSID value.

Syntax: **wl bssid**

The basic service set identifier is the 48-bit MAC (hardware) address of the WLAN interface in the AP that serves the stations in a basic service set (BSS).

dfs_status

Gets the DFS status.

Syntax: **wl dfs_status**

Example return:

Broadcom Confidential

```
state IDLE time elapsed 0ms radar channel cleared by dfs none
```

bi

Gets/sets the beacon period.

Syntax: **wl bi**

Returns the set beacon interval (BI). If the beacon interval is set for 100 ms, 100 is returned.

stats

Dumps the statistics for the IEEE 802.11 interface.

Syntax: **wl stats**

Use **oid_802_11_statistics** to dump statistics for the IEEE 802.11 interface.

Tx Counters:

```
Frag Count 8724 Mcast Frame Count 0
Fail Count 0 RetryCount 0
Multiple Retry Count 10 RTS Success Count 0
RTS Failure Count 0 ACK Failure Count 0
```

Rx Counters:

```
Frame Dup Count 0 Frag Count 0
Mcast Frame Count 295297 FCS Error Count 0
```

rcvok

Returns the number of frames received without errors.

Syntax: **wl rcvok**

Use **OID_GEN_RCV_OK** to get 'Frames received without errors'.

xmitok

Returns the number of frames transmitted without errors.

Syntax: **wl xmitok**

Use **OID_GEN_XMIT_OK** to get 'Frames transmitted without errors'.

rcvdropped

Returns the number of frames missed, no buffer.

Use **rndis_oid_gen_rcv_no_buffer** to get 'Frames missed, no buffer'.

Broadcom Confidential

config

Gets and sets the default configuration.

Syntax: **wl config**

Using **oid_802_11_configuration**

Syntax: **config** [beacon beacon_value atim atim_value freq freq_in_Khz]

The frequency must be in kilohertz. Invalid frequencies will be rejected.

Example to set parameters:

```
wl config beacon 100 atim 0 freq 2412000
```

Example to get parameters:

```
wl config Beacon Period 100 kusecs ATIM Window 0 kusecs Channel Freq 2412000 Khz (2.412
GHz) Hop pattern 0x0 Hop set 0x0 Dwell time 0 kusecs
```

nettypes_supported

Gets supported network types.

Syntax: **wl nettypes_supported**

Example:

```
wl nettypes_supported
Network types: count 2
0: Ndis802_11OFDM24
1: Ndis802_11DS
```

nettype

Gets/sets the network type in use.

Syntax: **wl nettype**

Example return:

```
Ndis802_11OFDM24
```

Broadcom Confidential

QUALITY OF SERVICE (QOS) RELATED COMMANDS

cwmin

Sets the cwmin.

Syntax : **wl cwmin** <limit>

where

limit is an integer [1, 255]).

Cwmin is the size of the Minimum Contention Window QoS prioritizations. This parameter is input to the algorithm that determines the initial random backoff wait time ("window") for retry of a transmission. The value specified for cwmin is the upper limit (in milliseconds) of a range from which the initial random backoff wait time is determined. The first random number generated will be a number between 0 and the number specified here.

If the first random backoff wait time expires before the data frame is sent, a retry counter is incremented and the random backoff value (window) is doubled. Doubling will continue until the size of the random backoff value reaches the number defined for cwmax.

The value for cwmin must be lower than the value for "cwmax" (cwmin < cwmax).

cwmax

Sets the cwmax.

Syntax : **wl cwmax** <limit>

where

limit is an integer [256, 2047]).

Cwmax is the Maximum Contention Window for QoS prioritizations. The value specified in cwmax is the upper limit (in milliseconds) for the doubling of the random backoff value. This doubling continues until either the data frame is sent or the cwmax size is reached. After cwmax size is reached, retries will continue until the maximum allowable number of retries is reached. The value for cwmax must be greater than or equal to the value for "cwmin" (cwmax >= cwmin).

TRANSMISSION RETRY CONTROL

srl

Sets the short retry limit.

Syntax : **wl srl** <limit>

where

limit is an integer [1, 255].

This command indicates the number of retransmission attempts for frames shorter than the RTS threshold. If this number is reduced, frames are discarded more quickly, so the buffer space requirement is lower. If this number is increased, retransmitting up to the limit takes longer and might cause the TCP to throttle back on the data rate.

Broadcom Confidential

lrl

Sets the long retry limit.

Syntax : **wl lrl** <limit>

where

limit is an integer [1, 255].

This command indicates the number of retransmission attempts for frames longer than the RTS threshold. If this number is reduced, frames are discarded more quickly, so the buffer space requirement is lower. If this number is increased, retransmitting up to the limit takes longer and might cause the TCP to throttle back on the data rate.

RATE PARAMETERS

rate

Forces a fixed rate.

Syntax: **wl rate** <rate>

For *rate*, the valid values are:

IEEE 802.11a operation: 6, 9, 12, 18, 24, 36, 48, 54

IEEE 802.11b operation: 1, 2, 5.5, 11

IEEE 802.11g operation: 1, 2, 5.5, 6, 9, 11, 12, 18, 24, 36, 48, 54

-1 (default): automatically determines the best rate.

bg_mrate/a_mrate

Forces a fixed multicast rate.

Syntax: **wl bg/a_mrate** <rate>

For *rate,* the valid values are:

IEEE 802.11a operation: 6, 9, 12, 18, 24, 36, 48, 54

IEEE 802.11b operation: 1, 2, 5.5, 11

IEEE 802.11g operation: 1, 2, 5.5, 6, 9, 11, 12, 18, 24, 36, 48, 54

-1 (default): automatically determines the best rate.

a_rate

Forces a fixed rate for the A PHY.

Syntax: **wl a_rate** <rate>

For *rate* the valid values are:

IEEE 802.11a : 6, 9, 12, 18, 24, 36, 48, 54

Broadcom Confidential

-1 (default): automatically determines the best rate

The rate option without any arguments returns the current negotiated rate between a STA and an AP in Infrastructure mode, or the weighted average of the last 32 frames sent in ad hoc mode between two stations. The best choice is to keep the value at –1, which is the Autonegotiate mode. To override the current rate, add the required rate argument. Then, every packet sent out has this new rate stamped in it.

mrate

Forces a fixed multicast rate.

Syntax: **wl mrate**

Valid values are:

IEEE 802.11a operation: 6, 9, 12, 18, 24, 36, 48, 54

IEEE 802.11b operation: 1, 2, 5.5, 11

IEEE 802.11g operation: 1, 2, 5.5, 6, 9, 11, 12, 18, 24, 36, 48, 54

-1 (default): automatically determines the best rate.

bg_rate

Forces a fixed rate for the IEEE 802.11b/g PHY.

Valid values are:

IEEE 802.11b operation: 1, 2, 5.5, 11

IEEE 802.11g operation: 1, 2, 5.5, 6, 9, 11, 12, 18, 24, 36, 48, 54

-1 (default): automatically determines the best rate.

band

Returns or sets the current frequency band.

auto (default): automatically switches between available bands

a: forces the use of the IEEE 802.11a frequency band

b: forces the use of the IEEE 802.11b frequency band

rateparam

Sets driver rate selection tunables.

Syntax: **wl rateparam** `<arg1> <arg2>`

arg 1: tunable id

arg 2: tunable value

rateset

Returns or sets the supported and basic ratesets.

Broadcom Confidential

Syntax: **wl rateset**

With no arguments, returns the rateset.

Example return:

```
[ 1(b) 2(b) 5.5(b) 6 9 11(b) 12 18 24 36 48 54 ]
```

Arguments 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.
```

Note: To change the rateset, you must "down" the driver with **wl down**, change the rateset and then use **wl up** to bring up the driver.

suprates

Gets/sets the IEEE 802.11g override for the supported rateset.

Syntax: **wl suprates**

With no arguments, returns the rateset. Arguments are either a list of rates, or a 0 or –1 to specify an empty rateset to clear the override. Listed rates are in Mbps. Example: `1 2 5.5 11`

abminrate

Gets/sets the Afterburner minimum rate threshold.

Syntax: **wl abminrate**

Example return: 36 Mbps.

ANTENNA CONTROLS

antdiv

Sets the antenna diversity property for RX.

Syntax: **wl antdiv** `<value>`

0 = Forces the use of antenna 0.

1 = Forces the use of antenna 1.

3 = Automatic selection of antenna diversity.

Broadcom Confidential

An antenna diversity receiver could be receiving the wireless/RF signal on one or more antennas having distinct characteristics (for example, location, radiation pattern and/or polarization). In such situations, when one antenna is experiencing bad reception conditions (deep or flat fading), the other should probably not. Therefore, when the right antenna is set at the right moment, the diversity receiver behaves as if it is receiving a continuously perfect signal.

txant

Sets the transmit antenna.

Syntax: wl txant <value>

0 = Forces the use of antenna 0.

1 = Forces the use of antenna 1.

3 = Uses the RX antenna selection that was in force during the most recently received good PLCP header.

SECURITY AND ENCRYPTION CONTROLS

wepstatus

Sets or gets the WEP status.

Syntax: **wl wepstatus** <on/off>

primary_key

Sets or gets the index of the primary key.

Syntax: **wl primary_key**

addwep

Sets an encryption key.

Syntax: **wl addwep**

The OID_802_11_ADD_WEP OID requests the miniport driver to set an IEEE 802.11 Wired Equivalent Privacy (WEP) key to a specified value.

addwep <keyindex> <keydata>

The key must be 5, 13, or 16 bytes long, or 10, 26, 32, or 64 hexadecimal digits long. The encryption algorithm is automatically selected based on the key size. The typed key is accepted only when the key length is 16 bytes/32 hexadecimal 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*]

rmwep

Removes the encryption key at the specified key index.

Syntax: **wl rmwep**

Broadcom Confidential

tsc

Prints the TX SEQ counter for the key at a specified key index

Syntax: **wl tsc**

wsec_test

Generates **wsec** errors.

Syntax: **wl wsec_test** <test_type> <keyindex|xx:xx:xx:xx:xx:xx>

Example return:

```
wsec test_type may be a number or name from the following set:
0x0001 mic_error
0x0002 replay
```

tkip_countermeasures

Enables or disables TKIP countermeasures.

Syntax: **wl tkip_countermeasures**

(TKIP-enabled AP only)

0 = Disable

1 = Enable

wsec_restrict

Drops unencrypted packets if **wsec** is enabled.

Syntax: **wl wsec_restrict**

0 = Disable

1 = Enable

eap

Restricts traffic to IEEE 802.1X packets (until IEEE 802.1X authorization succeeds).

Syntax: **wl eap**

0 = Disable

1 = Enable

authorize

Restricts traffic to IEEE 802.1X packets.

Syntax: **wl authorize**

Broadcom Confidential

deauthorize

Does not restrict traffic to IEEE 802.1X packets.

Syntax: **wl deauthorize**

deauthenticate

Deauthenticates a STA from the AP with optional reason code.

Syntax: **wl deauthenticate**

(AP ONLY)

wsec

Returns/sets a value containing a bit vector representing which wireless security modes are currently enabled.

Syntax: **wl wsec**

Bit 0 = WEP enabled

Bit 1 = TKIP enabled

Bit 2 = AES enabled

Bit 3 = WSEC in software

auth

Gets/sets the IEEE 802.11 authentication type.

Syntax: **wl auth**

0 = Open system

1 = Shared key

wpa_auth

Bit vector of WPA authorization modes.

Syntax: **wl wpa_auth**

1 = WPA-NONE

2 = WPA-802.1X/WPA-Enterprise

4 = WPA-PSK/WPA-Personal

wpa_cap

Sets/gets IEEE 802.11i RSN capabilities.

Syntax: **wl wpa_cap**

Broadcom Confidential

set_pmk

Sets the passphrase for PMK (in-driver resident supplicant).

Syntax: **wl set_pmk**

leap

Sets the parameters for LEAP authentication.

Syntax: **wl leap** <ssid> <user name> <password> [domain]

mac

Sets or gets the list of source MAC address matches.

Syntax: **wl mac** *xx:xx:xx:xx:xx:xx* [xx:xx:xx:xx:xx:xx ...]

To clear the list, type:

wl mac none

Values are:

0 = Allow association to stations on the MAC list.

1 = Deny association to stations on the MAC list.

Multiple access points having the same SSID could cause undesirable results during tests because the client and the server could intermittently roam to different access points. To gain control over such behavior, the utility supports the **mac** and **deny** commands.

For the AP driver, the list controls which stations are allowed to authenticate. For the STA driver, the list allows/restricts association to a particular set of BSSIDs. You can add multiple MAC addresses to the list and then allow/deny access of these based on your deny status.

Broadcom Confidential

By default, association works normally and all valid targets are considered. If you specify a list of BSSIDs, then all other BSSIDs will be ignored before attempting a join. If you turn the list into a deny list by specifying:

```
wl deny 1
```

Then BSSIDs specified in the list will be ignored before a join is attempted.

To clear the list and return to normal functionality, do the following:

```
wl mac none
```

macmode

Sets the mode of the MAC list.

Syntax: **wl macmode**

0 = Disable MAC address matching.

1 = Deny association to stations on the MAC list.

2 = Allow association to stations on the MAC list.

encryptstrength

Gets the current WEP key length.

Syntax: **wl encryptstrength**

decryptstatus

Gets the status of WEP decryption.

Syntax: **wl decryptstatus**

One of:

1 = Success

2 = Failure

3 = Unknown

addkey

Sets an IEEE 802.11 Wired Equivalent Privacy (WEP) key.

Syntax: **wl addkey**

The OID_802_11_ADD_KEY OID requests the miniport driver to set an IEEE 802.11 Wired Equivalent Privacy (WEP) key to a specified value.

Syntax: **addkey** <keyindex> <keydata> [notx] [xx:xx:xx:xx:xx:xx]

wepdefault

Resets the WEP keys to their power-on defaults.

Broadcom Confidential

Syntax: **wl addkey**

pmkid_info

Returns the PMKID table.

Syntax: **wl pmkid_info**

Options:

```
-s S, --ssid=SSSID 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=MACparticular BSSID MAC address to scan, xx:xx:xx:xx:xx:xx
-n N, --nprobes=Nnumber of probes per scanned channel
-a N, --active=Ndwell time per channel for active scanning
-p N, --passive=Ndwell time per channel for passive scanning
-h N, --home=Ndwell time for the home channel between channel scans
-c L, --channels=Lcomma or space separated list of channels to scan
```

passive

Puts the scan engine into Passive mode.

Syntax: **wl passive**

Scanresults

Returns the results from the last scan.

Syntax: **wl scanresults**

scan_channel_time

Gets/sets the scan channel time.

Syntax: **wl scan_channel_time**

scan_unassoc_time

Gets/sets the *unassoc scan channel* dwell time.

Syntax: **wl scan_unassoc_time**

scan_home_time

Gets/sets the *scan home channel* dwell time.

Syntax: **wl scan_home_time**

Broadcom Confidential

scan_passive_time

Gets/sets the *passive scan channel* dwell time.

Syntax: **wl scan_passive_time**

scan_nprobes

Gets/sets the *scan* parameter for number of probes to use (per channel scanned).

Syntax: **wl scan_nprobes**

scansuppress

Suppresses all scans for testing

Syntax: **wl scansuppress**

0= Allow scans

SCAN CONTROLS

scan

Initiates a scan.

Syntax: **wl scan**

The default scan is an active scan across all channels for any SSID.

Optional argument to scan a specific SSID: `SSID`

1 = Suppress scans

ASSOCIATION AND STATUS

isup

Gets the operational state of the driver.

Syntax: **wl isup**

0 = Down

1 = Up

assoclist

Gets the list of associated MAC addresses.

Syntax: **wl assoclist** */* AP Side*

Example: Linux */

Broadcom Confidential

AP only: Gets the list of associated MAC addresses.

Typing the above displays the following string on the serial console attached to the AP/wireless router (from the BusyBox hash prompt #):

assoc

Prints information about the current network association to the AP with the specified BSSID.

Syntax: **wl channels**

shownetworks

Pretty-prints the BSSID list.

Syntax: **wl shownetworks**

authe_sta_list

Gets the authenticated STA MAC address list.

Syntax: **wl authe_sta_list**

autho_sta_list

Gets the authorized STA MAC address list.

Syntax: **wl autho_sta_list**

join

Joins a specified network SSID.

Syntax: **join** <ssid> [key xxxxx] [imode bss|ibss] [amode open|shared|wpa|wpapsk|wpa2|wpa2psk|wpanone]

In the current releases, to use this feature, you must know the SSID of the access point you want to join.

Example: `join KILROY imode infra amode open`

The *amode* or Authentication mode choices are *open* or *shared*.

Under shared key authentication, each wireless station is assumed to have received a secret shared key over a secure channel that is independent from the IEEE 802.11 wireless network communications channel.

ssid

Sets or gets the SSID of a wireless network connection profile.

Syntax: **wl ssid**

This setting initiates an association attempt if in Infrastructure mode, or joins/creates an IBSS if in IBSS (ad hoc) mode, or creates a BSS if in AP mode.

Broadcom Confidential

closednet

Sets/gets the BSS closed network attribute.

Syntax: **wl closednet**

When this setting is enabled (=1), the adapter does not send out a probe response as a result of having received a Broadcast Probe request.

assoc_info

Returns the *assoc req* and *resp* information [STA only].

Syntax: **wl assoc_info**

closed

Hides the network from active scans, 0 or 1.

Syntax: **wl closed**

0 = Open

1 = Hide

bss set/get bss

Enabled status: Up/Down

Syntax: **wl bss**

join_pref

Sets/gets *join* target preferences.

Syntax: **wl join_pref**

assoc_pref

Sets/gets the association preference.

Syntax: **wl assoc_pref** [auto|a|b|g]

CHANNEL AND BAND CONTROL PARAMETERS

chan_info

Channel information.

Syntax: **wl chan_info**

Broadcom Confidential

autochannel

Automatic channel selection.

Syntax: **wl autochannel**

1 = Issue a channel scanning.

2 = Set channel based on the channel scanning result.

Without an argument to only show the channel selected, **ssid** must be set to null before this process, and RF must be up.

auto

Switches between available frequency bands (default).

Syntax: **wl auto**

a = Force use of IEEE 802.11a band

b = Force use of IEEE 802.11b band

bands

Returns the list of available IEEE 802.11 bands.

Syntax: **wl bands**

where **bands** options are as follows:

0 = Auto-Select

1 = 5 GHz

2 = 2.4 GHz

3 = All bands

channel

Sets the working channel.

Syntax: **wl channel** <channel #>

This command sets the default channel. As an AP, the default channel is used as the channel on which the AP found the network. For a STA, the default channel is used as the channel on which the STA creates an IBSS (ad hoc) network.

Valid channels for 802.11a 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.

Valid channels for 802.11b/g are 1–14.

Broadcom Confidential

channels

Returns valid channels for the current settings.

Syntax: **wl channels**

Example return for US country code/locale:

1 2 3 4 5 6 7 8 9 10 11 and so on

channels_in_country

Returns valid channels for the country specified.

Syntax: **wl channels_in_country**

Argument 1 is the country abbreviation.

Argument 2 is the band (a or b).

ROAM CONTROLS

roam_trigger

Sets the *roam trigger* RSSI threshold.

Syntax: **wl roam_trigger** [`integer [, a/b]`]

Example return:

```
roam_trigger is 0xffba(-70)
```

roam_delta

Sets the *roam candidate* qualification delta.

Syntax: **wl roam_delta** [`integer [, a/b]`]

Example return:

```
roam_delta is 0x0014(20)
```

roam_scan_period

Sets the *roam candidate* qualification delta.

Syntax: **wl roam_scan_period** `<integer>`

prb_resp_timeout

Gets/sets the *probe response* timeout.

Syntax: **wl prb_resp_timeout**

Broadcom Confidential

REGULATORY TEST AND MEASUREMENTS

evm

Starts/stops an EVM test on the given channel.

Syntax: **wl evm**

Argument 1 is the channel number 1–14, or "off" or 0 to stop the test.

Argument 2 is an optional rate (1, 2, 5.5, or 11) in Mbps.

Argument 3 is an optional TX power in milliwatts (range [1, 84]).

To perform EVM tests using the wl:

1. Enter the command **wl down** to disable the wireless driver.
2. Turn on EVM mode on any channel. To do this, enter the command **wl evm 2**, where 2 is the channel on which you want to transmit.
3. When you are done with EVM, issue the command **wl evm 0** and then **wl up** to bring the wireless driver back to normal operation.

pm

Sets Driver Power Management mode.

Syntax: **wl PM** <value>

0 = CAM (constantly awake)

1 = PS (power-save)

2 = FAST PS mode

In the Broadcom WLAN driver, FAST PS mode consumes slightly more power than the standard PS Mode, but can operate at higher rates (close to 54g rates).

The difference between FAST PS and PS is very dependent on the usage model. If there is no traffic, there is no difference. If the traffic is a steady stream (some packets every DTIM), there is no difference. If the traffic is a stream that happens less often than every DTIM, then the adapter consumes more power in FAST PS mode and less power but more latency in PS mode. The amount of power difference depends on the frequency of traffic.

wake

Sets DRIVER POWER-SAVE MODE SLEEP state.

Syntax: **wl wake** <value>

Where *value* choices are:

0 = Core-managed

1 = Awake

Only for STA:

Broadcom Confidential

wake = TRUE(1= awake) OR FALSE(0=core managed);

When this field is set to 1, the MAC exits the SLEEP state immediately.

When the field is set to 0, the MAC enters the SLEEP state as appropriate. The host must write this field to 1 when the MAC is functioning as an AP.

tssi

Gets the TSSI value from the radio.

Syntax: **wl tssi**

The need to measure signal levels in wireless infrastructure equipment is critical to adjust transceiver automatic gain control (AGC) circuits, in the receiver (RX) for maximum sensitivity to the varying inputs from the mobile users, and in the transmitter (TX) so that the output power is maintained at its optimum level for performance mask, power-amplifier (PA) efficiency and linearity, and government regulations.

Within an RX, the received signal strength indication (RSSI) is used to adjust the gain of the RX to extend the dynamic range to 100 dB. For the TX, accurately controlling the transmit signal power with a transmitted signal strength indication (TSSI) at RF frequencies at the higher power levels significantly eases the implementation of controls for PA operating level for maximum efficiency.

txpwr

Sets TX power in milliwatts.

Syntax: **wl txpwr** <value in mW>

Value in mW = Range [1, 84]

When TX power is set, it overrides the current setting (SPROM, locale, and so on) and forces the transmitter to always use this new value for power output.

Note: For dual-band cards, you must bandlock to your required band to make this command work.

Example: First run **wl band b** to lock in the 2.4-GHz band, and then use this command.

txpwr1

Sets the TX power in various units.

Syntax: **wl txpwr1**

Choose one of:

-d dbm units (default)

-q quarter dbm units

-m milliwatt units

Can be combined with:

Broadcom Confidential

-o (Turns on the override to disable regulatory and other limitations.)

Use `wl txpwr -1` to restore defaults.

txpathpwr

Turns the TX path power on or off on BCM2050 radios.

Syntax: **wl txpathpwr**

txpwrlimit

Returns the current TX power limit.

Syntax: **wl txpwrlimit**

powerindex

Sets the transmit power for a band (0–63).

Syntax: **wl powerindex**

-1 = Default value

rssi

Gets the current RSSI value.

Syntax: **wl rssi** /* *STA side* */

For an AP, you must specify the MAC address of the STA.

Note: The MAC layer operates together with the physical layer (PHY) by sampling the transmitted energy over the medium transmitting the data. The PHY uses a clear channel assessment (CCA) algorithm to determine if the channel is clear. This function is accomplished by measuring the RF energy at the antenna and determining the strength of the received signal. This measured signal is commonly known as RSSI. If the received signal strength is below a specified threshold, the channel is declared clear, and the MAC layer is given the clear channel status for data transmission. If the RF energy is above the threshold, data transmissions are deferred in accordance with the protocol rules. The standard provides another option for CCA that can be alone or with the RSSI measurement.

pwr_percent

Gets/sets the power output percentage.

Syntax: **wl pwr_percent**

To set a specified percentage such as 60 percent of the full (100 percent) power:

wl pwr_percent 60 = Sets the new value.

wl pwr_percent = Gets the new setting.

pwr_percent is 60(0x3c)

Broadcom Confidential

diag

Diagnostic test index.

Syntax: **wl diag**

1 = Interrupt

2 = Loopback

3 = Memory

4 = LED

Precede the **diag** command by the **wl down** command, and then follow with **wl up**.

rand

Gets a 2-byte random number from the MAC PRNG.

Syntax: **wl rand**

Usage: `wl rand`

reset_d11cnts

Resets the IEEE 802.11 MIB counters.

Syntax: **wl reset_d11cnts**

cac_addts

Adds TSPEC (an error is returned if the STA is not associated or WME (WMM) is not enabled).

Syntax: **wl cac_addts**

cac_delts

Deletes TSPEC (an error is returned if the STA is not associated or WME (WMM) is not enabled).

Syntax: wl cac_delts

regulatory

Gets/sets the Regulatory Domain mode (IEEE 802.11d)

Syntax: **wl regulatory**

Note: The driver must be down.

spect

Gets/sets the IEEE 802.11h Spectrum Management mode.

Broadcom Confidential

Syntax: **wl spect**

0 = Off

1 = Loose interpretation of IEEE 802.11h specification (may join non-IEEE 802.11h APs)

2 = Strict interpretation of IEEE 802.11h specification (may not join non-IEEE 802.11h APs)

3 = Disable IEEE 802.11h and enable IEEE 802.11d

4 = Loose interpretation of IEEE 802.11h+d specification) may join non-IEEE 802.11h APs)

noise

Gets the noise level (MoVING average) in dBm immediately after TX.

Syntax: **wl noise**

fqacurcy

A manufacturing test that sets the Frequency Accuracy mode.

Syntax: **wl fqacurcy** <channel>

The argument is channel number 1–14 or 0 to stop the test.

This command is used to measure the center frequency of the carrier. After this mode is set, the user can use a spectrum analyzer to measure the accuracy of the frequency for a specific channel.

crsuprs

A manufacturing test that sets the Carrier Suppression mode.

Syntax: **wl crsuprs** <channel>

The argument is channel number 1–14 or 0 to stop the test.

This command is used to measure the lo-leakage/carrier. The carrier is required to be 15 dB below the peak power spectrum. The transmitter transmits a repetitive 01 data sequence with scrambler disabled using DQPSK modulation.

longtrain

A manufacturing test that sets the Longtraining mode.

Syntax: **wl longtrain** <channel>

The argument is a band channel number or 0 to stop the test.

band, **wl_band**, **wlc_get_band**, **wlc_set_band**, returns or sets the current band.

clk

Sets the board CLOCK state.

Syntax: **wl clk**

Broadcom Confidential

0 = Clock OFF

1 = Clock ON

channel_qa

Gets the last channel quality measurement.

Syntax: **wl channel_qa**

channel_qa_start

Starts a channel quality measurement.

Syntax: **wl channel_qa_start**

freqtrack

Sets the Frequency Tracking mode.

Syntax: **wl freqtrack**

lifetime

Sets the lifetime parameter (milliseconds) for each AC.

Syntax: wl lifetime

wl lifetime `be|bk|vi|vo` [`<value>`]

measure_req

Sends an IEEE 802.11h measurement request.

Syntax: **wl measure_req** `<type> <target MAC addr>`

Measurement types are: TPC, Basic, CCA, RPI

The target MAC address format is: `xx:xx:xx:xx:xx:xx`

quiet

Sends an IEEE 802.11h **quiet** command.

Syntax: **wl quiet** `<TBTTs until start>, <duration (in TUs)>, <offset (in TUs)>`

csa

Sends an IEEE 802.11h channel switch announcement.

Syntax: **wl csa** `<mode> <when (in TBTTs)> <channel>`

Broadcom Confidential

constraint

Sends an IEEE 802.11h power constraint information element (IE).

Usage: `wl constraint 1-255 db`

dtim

Gets/sets the delivery traffic indication message (DTIM).

Syntax: **wl dtim**

rm_req

Requests a radio measurement of type basic, CCA, or RPI.

Syntax: **wl rm_req** cca -c 1 -d 50 cca -c 6 cca -c 11

Specifies a series of measurement types each followed by options.

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.

rm_rep

Gets the current radio measurement report.

Syntax: **wl rm_rep**

rssidump

Dumps RSSI values from ACI scans.

Syntax: **wl rssidump**

Broadcom Confidential

interference

Gets/sets Interference Mitigation mode.

Syntax: **wl interference**

Choices are:

0 = Bone

1 = Non-WLAN

2 = WLAN manual

3 = WLAN automatic

```
Auto wireless LAN Interference mitigation is enabled and not active.
```

This is a BCM4306 or better PHY set of modes. It only applies to the GPHY, and not the APHY or BPHY.

Usage: `wl interference`

```
Auto Wireless LAN Interference mitigation is enabled and active.
```

1 = Use for non-wireless device in the same band. Use for strong non-IEEE 802.11 interference such as some 2.4-GHz phones or 2.4-GHz audio/visual wireless transmitters. This setting greatly reduces the sensitivity of Broadcom devices, especially for the BCM4306B0, but less so for the BCM4306C0.

2 = Implements ACI (Adjacent Channel Interference) mode, which makes the Broadcom receiver less sensitive to IEEE 802.11 traffic on adjacent channels but at some cost to sensitivity on the Broadcom channel. Choosing this option manually turns on ACI mode.

3 = Turned ON. The driver automatically determines whether to turn on ACI mode. When this option is chosen, a **wl interference** query will tell you that you are in interference mode 3, the Auto Wireless LAN Interference mode, and if the automatic mechanism *currently* has ACI active. The *and active* part of the message means that at the time that the **wl interference** command was run, the Auto WLAN mode had ACI on.

frameburst

Disables/enables Frame Burst mode.

Syntax: **wl frameburst** <`on`/`off`>

When frame bursting is enabled, multiple frames are sent with a minimum interframe gap, which enhances network efficiency and reduce overhead. Windows STA drivers are able to perform frame bursting. Frame bursting is disabled by default.

curpower

Returns the current TX power settings.

Syntax: **wl curpower**

```
-q (quiet): estimated power only.
```

Broadcom Confidential

COUNTRY AND LOCALE

country

Selects the country code for use with IEEE 802.11d.

Syntax: **wl country**

Use either the long name or the abbreviation from ISO 3166.

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

WIRELESS DISTRIBUTION SYSTEM

wds

Sets or gets the list of WDS member MAC addresses.

Syntax: **wl wds**

Set using a space separated list of MAC addresses.

```
wl wds xx:xx:xx:xx:xx:xx [xx:xx:xx:xx:xx:xx ...]
```

lazywds

Sets or gets Lazy WDS mode.

Syntax: **wl lazywds**

Returns/sets the value of the lazy WDS setting.

0 = Lazy WDS is disabled. WDS partners must be set explicitly.

1 = Lazy WDS is enabled and the AP will accept WDS partners from any MAC address.

wds_remote_mac

Gets the MAC address of the WDS link remote endpoint.

Syntax: **wl wds_remote_mac**

wds_wpa_role_old

Gets the WPA role (old) of the WDS link local endpoint.

Syntax: **wl wds_wpa_role_old**

wds_wpa_role

Gets/sets the WPA role of the WDS link local endpoint.

Syntax: **wl wds_wpa_role**

Broadcom Confidential

MODE CONTROLS

gmode

Sets the 54g™ mode.

Syntax: **wl gmode** <mode>

```
Mode = LegacyB | Auto | GOnly | BDeferred | Performance |LRS
```

LegacyB

Rateset: 1b, 2b, 5.5, 11

Preamble: Long

Short slot: Off

In LegacyB g-mode operation, only CCK rates are allowed in the network, and only 1 and 2 Mbps are basic, so that legacy IEEE 802.11 devices can join (older than IEEE 802.11b). In this mode, the IEEE 802.11g AP or IBSS will not include an ERP IE or an IE. This mode is intended to look as much like an early IEEE 802.11b network as possible to allow interoperability with devices that have trouble with any of the newer specification changes. The AP advertises and uses only IEEE 802.11b CCK rates. IEEE 802.11g clients can still associate but will only operate at IEEE 802.11b rates.

Summary:

- Uses IEEE 802.11 long slot timing only.
- Only 1 and 2 Mbps rates are basic rates, so that all legacy devices can join.
- For IEEE 802.11b networks only.

LRS

Rateset: 1b, 2b, 5.5b, 11b (CCK only)

Extended rateset: 6, 9, 12, 18, 24, 36, 48, 54

Preamble: Long

Shortslot: Auto

In LRS g-mode operation, all IEEE 802.11g rates are available, but only CCK rates are basic to allow IEEE 802.11 devices to join. The rateset is split, with only four rates in the *supported rates* IE to allow interoperability with IEEE 802.11 devices that have trouble with more than four rates in the IE (take the full 12 rates, throwing away none, and split the four CCK rates into the first rate element and put all the OFDM rates in the ESR).

Summary:

- Allows both short and long slot timings (11b and 11g).
- No IEEE 802.11 devices (legacy) can join.
- Supports devices that can handle *only* four CCK rates in the rateset.

Auto [default]

Rateset: 1b, 2b, 5.5b, 11b, 18, 24, 36, 54

Extended rateset: 6, 9, 12, 48

Preamble: Long

Broadcom Confidential

Short slot: Auto

This mode allows for maximum compatibility and is fully compliant with the IEEE 802.11g specification. All 12 IEEE 802.11g rates (1(b), 2(b), 5.5(b), 6, 9, 11(b), 12, 18, 24, 36, 48, and 54) will be advertised, but the basic rateset will include only 1, 2, 5.5, and 11, so that legacy IEEE 802.11b clients can associate (no-OFDM). This mode defaults to high-speed IEEE 802.11g operation by using short slot timing. The AP will switch to long slot timing if an IEEE 802.11b or an IEEE 802.11g client that does not support short slot timing enters the network.

When set to auto, the AP advertises this short slot timing in the beacon frames. As long as there are only 11g clients associated, short slot timing will be used, and throughput will be high. If a client that does not support short-slot timing (such as a legacy 11b client) joins the BSS, the AP will cease to advertise short slot. All STA devices in the BSS will then start to use the normal IEEE 802.11b interpacket timings. If all non-short-slot timing STAs leave the BSS, the AP and remaining STAs will revert to using short slot timing.

Summary:

Use of IEEE 802.11g short slot timing is automatic:

- If no IEEE 802.11b clients are associated, short slot timing will be used.
- If an IEEE 802.11b client associates, then the AP will use long slot timing.
- Because no OFDM rates are in the basic rateset, the maximum IEEE 802.11g performance cannot be achieved.

IEEE 802.11g mode performance

Rate set: 1b, 2b, 5.5b, 6b, 9, 11b, 12b, 18, 24b, 36, 48, 54

Preamble: Short required

Short slot: On and required

In IEEE 802.11g mode operation, all 12 IEEE 802.11g rates are available, and the rates are *not* split. OFDM Basic rates are present, so CCK-only (IEEE 802.11b) devices cannot join.

The rates are not split, so that legacy 54g drivers can see all the rates, not just a good subset. Putting more than eight rates in the Supported Rates element is not included in the IEEE 802.11 specification or the IEEE 802.11(a, b, or g) specification, but the practice works with all Broadcom drivers.

As in the Auto setup, short preambles are required to join. In addition, short slot support is required to join, and short slot operation is always on in the network.

In the normal automatic setup, short slot operation attention is given to overlapping BSSs, but in this mode it is not.

Summary:

- Use of IEEE 802.11g short slot timing is mandated.
- Use of IEEE 802.11g short preamble is mandated.
- IEEE 802.11b clients cannot associate because of the above two mandates.
- Designed to use the maximum bandwidth only on an IEEE 802.11g network topology.

wet

Gets/sets Wireless ENET Bridging mode.

Broadcom Confidential

Syntax: **wl wet**

Default wet is off:

wl wet on

wl wet

wet is 1 (on).

To change it back to default off state:

wl wet off

wl wet

wet is 0 (off).

Note: Older IEEE 802.11 devices (supporting only 1 and 2 Mbps) cannot join in this mode.

gMODE PROTECTION CONTROLS

gmode_protection

Gets gmode protection.

Syntax: **wl gmode_protection** *on*/*off*

0 = Disabled

1 = Enabled

It is a query command used to determine whether protection mechanisms are currently being used.

0 = Protection mechanisms are not currently being used.

1 = Protection mechanisms are currently being used.

The IEEE 802.11g standard uses orthogonal frequency division multiplexing (OFDM) to attain its high data speed. To protect IEEE 802.11b users, IEEE 802.11g is required to also send a protection signal based on the longer complementary code keying (CCK). Omitting the protection signal ensures high data speeds for IEEE 802.11g users at the cost of locking out IEEE 802.11b users.

Without such protection, an IEEE 802.11b user would be blocked by an invisible flow of IEEE 802.11g data and would assume that the wireless network had crashed.

Features:

- As required for Wi-Fi compliance, protection mechanisms are enabled automatically whenever an IEEE 802.11b STA joins the BSS.
- If no IEEE 802.11b STA joins the BSS, then no protection mechanism will be used, and full IEEE 802.11g performance will be attained.
- The default protection mechanism is not CTS-to-self.
- Typing this command without any parameters displays the following string on the console:

Broadcom Confidential

```
gmode_protection is 0 (off)
```

- Mixed environments (ERP and Legacy PHYs [IEEE 802.11, IEEE 802.11b, and IEEE 802.11g coexisting]) require a protection mechanism.
- ERP ONLY STAs use a short-slot time to improve performance.
- The IEEE 802.11g specification defines a *gmode protection* mechanism as one that involves prefixing each OFDM IEEE 802 data frame with an RTS/CTS CCK frame sequence. The duration fields of the RTS and CTS frames should allow the IEEE 802.11b node to correctly set its NAV and avoid collisions with the subsequent OFDM frames.
- In accordance with the specification, the data frames should be sent at one of the basic rates, but with a CCK-only basic rateset.
- STAs are expected to automatically honor the bit announced in BSS beacons and should require no configuration.

gmode_protection_control

Gets/sets the 11g protection mode control algorithm.

Syntax: **wl gmode_protection_control** <option>

0 = Always off.

1 = Monitor local association.

2 = Monitor overlapping BSSs.

gmode_protection_cts

Gets/sets 11g protection type to CTS.

Syntax: **wl gmode_protection_cts** *on*/*off*

0 = Disable

1 = Enable

gmode_protection_override

Gets/sets 11g protection mode override.

Syntax: **wl gmode_protection_override** <option>

-1: auto = Protection will automatically be used if either an IEEE 802.11b STA associates to the AP, or the AP detects another legacy IEEE 802.11b BSS.

0: off = Turns OFF protection on the 54g AP. Protection mechanisms will *never* be used.

1: on = Turns ON protection on the 54g AP. Protection mechanisms will *always* be used.

The driver default is g_protection_override == AUTO, but can be set to 0/1 to force protection off/on. Typing this command without any parameters displays the following string on the console.

```
gmode_protection_override is -1
```

This indicates that the override is set for auto.

There are three modes:

- No protection (default in 54g). This mode is configured with wl gmode_protection_override 0.

Broadcom Confidential

- RTS/CTS, when legacy IEEE 802.11b STA is associated. This mode is configured with:

  `wl ignore_bcns TRUE (default)`

  `wl gmode_protection_override -1 (AUTO /default)`

- RTS/CTS, when overlapping legacy IEEE 802.11b operation is detected (`ignore_bcns=FALSE` may be required for Wi-Fi compliance for IEEE 802.11g devices). This mode is configured with:

  `wl ignore_bcns FALSE`

  `wl gmode_protection_override -1 (AUTO /default)`

legacy_erp

Gets/sets 11g legacy ERP inclusion.

Syntax: **wl legacy_erp**

0 = Disable

1 = Enable

The command gets/sets the legacy ERP inclusion flag for the driver for NonERP element advertisement. If set, include legacy ERP information element ID 47 along with IEEE 802.11g information element ID 42. The beacon sender shall set b0 (NonERP_present) and b1 (use_protection) for the use of this element. An ERP STA that is aware of a non-ERP STA shall set bit0 of NonERP information element true and transmit this information in a subsequent beacon frame.

infra

Gets/sets the Infrastructure mode.

Syntax: **wl infra**

bss, infra, or managed for Infrastructure BSS

ibss, or ad hoc for IBSS

auto for either

RADAR CONTROLS

radar

Enables/disables radar.

Syntax: **wl radar**

radarargs

Gets/sets radar parameters in order.

Syntax: **wl radarargs**

Broadcom Confidential

WME CONTROLS

wme

Sets Wireless Multimedia Extensions (WME) mode.

Syntax: **wl wme** `ap|sta [be|bk|vi|vo [ecwmax|ecwmin|txop|aifsn|acm <value>] ...]`

0 = Off

1 = On

-1 = Auto

wme_apsd

Sets Automatic Power Save Delivery (APSD) mode on the AP.

Syntax: **wl wme_apsd**

0 = Off

1 = On

wme_apsd_sta

Sets APSD parameters on the STA (the driver must be down).

Syntax: **wl wme_apsd_sta** `<max_sp_len> <be> <bk> <vi> <vo>`

where

<max_sp_len> = number of frames per USP: 0 (all), 2, 4, or 6 <xx>

0 = Disable

1 = Enable U-APSD per AC

wme_dp

Sets AC queue discard policy.

Syntax: **wl rssidump**

Usage: `wl wme_dp <be> <bk> <vi> <vo>`

<xx>: value 0 for newest-first, 1 for oldest-first.

wme_counters

Prints the WME statistics.

Syntax: **wl wme_counters**

Broadcom Confidential

INFORMATION ELEMENT CONTROLS

add_ie

Adds a vendor-proprietary IE to management packets.

Syntax: **wl add_ie** <pktflag> *length OUI hexdata* <pktflag>:

Bit 0 = Beacons

Bit 1 = Probe response

Bit 2 = Associate/reassociate response

Bit 3 = Authenticate response

Example: `wl add_ie 3 10 00:90:4C 0101050c121a03` to add this IE to beacons and probe responses.

del_ie

Deletes a vendor-proprietary IE from management packets.

Syntax: **wl del_ie** <pktflag> *length OUI hexdata* <pktflag>

Bit 0 = Beacons

Bit 1 = Probe response

Bit 2 = Associate/reassociate response

Bit 3 = Authenticate response

Example: `wl del_ie 3 10 00:90:4C 0101050c121a03`

list_ie

Dumps the list of vendor-proprietary IEs.

Syntax: **wl list_ie**

NVRAM/SROM WRITE CONTROLS

otpw

Writes an SROM image to the on-chip OTP.

Syntax: **wl otpw**

nvotpw

Writes NVRAM to the on-chip OTP.

Syntax: **wl nvotpw** *file*

Broadcom Confidential

legacylink

Sets the IBSS legacy link behavior.

Syntax: **wl legacylink**

Options are:

0 = Disable

1 = Enable

listen

Sets or queries the listen time in units of beacon interval.

Syntax: **wl listen**

RNDIS RELATED

rndismac

Gets/sets the current MAC address.

Syntax: **wl rndismac**

Set allowed with RNDIS-USB only.

MAC address example: `00:11:20:11:33:33` (colons optional)

NDIS RELATED

ndisscan

Initiates a broadcast SSID scan across all channels (no SSID argument).

Syntax: **wl ndisscan**

ndis_frag

Gets/sets the fragmentation threshold.

Syntax: **wl ndis_frag**

Range: [256–2346]

ndis_rts

Gets/sets the RTS threshold.

Syntax: **wl ndis_rts**

Broadcom Confidential

MIMO SPECIFIC

chanspec

Sets the channel, channel bandwidth, control sideband, and frequency band.

Syntax: **wl chanspec** *N* `[band a|b][ctl sideband u (upper) | l (lower)]`

Set <`channel`>`[a,b][n][u,l]`

Channel number (0–224)

Band: a = 5 GHz, b = 2 GHz, default to 2 GHz if channel <= 14

Bandwidth: n = 10 MHz, none for 20 MHz and 40 MHz

ctl sideband, l = lower, u = upper

-or-

Set channel with legacy format. Options are:

-c channel number (0–224)

-b band (5(a) or 2(b/g))

-w bandwidth, 10,20 or 40

-s ctl sideband, –1 = lower, 0 = none, 1 = upper

Example typical command sequence:

```
wl down
wl chanspec -c 3 -b 2 -w 40 -s -1
wl up
```

where

-c specifies channel

-b specifies band

-w specifies bandwidth

-s specifies control side band on the AP. After this, the STA can join the AP.

After the STA is associated with the AP, and they can ping each other, you can tune the N-PHY rate on both the AP and STA side at the same time by using the following command:

For 40-MHz channels, the channel number *N* is the channel number of the control channel, not the center frequency channel number.

The `band a|b` choice is optional and is assumed to be *b* (2.4 GHz) when channel number N ≤14, and *a* (5 GHz) when channel number N > 14.

The `control sideband u|l` choice is only used when specifying a 40-MHz channel. For a 20-MHz channel, do not append a *u* or *l*.

Example for a 5-GHz band, 40-MHz channel pair 36–40, with 40 as control:

Broadcom Confidential

```
wl chanspec 40u
```

Example for a 5-GHz band, 40-MHz channel pair 36–40, with 36 as control:

```
wl chanspec 36l
```

Example for a 5-GHz band, 20-MHz channel 36

```
wl chanspec 36
```

Example for a 2.4-GHz band, 20-MHz channel 6

```
wl chanspec 6
```

Or, you can use the old CHANSPEC option parameters:

```
wl chanspec -c [channel number] -b [band 2|5] -w [bw 20|40] -s [ctl sideband
0(none)|1(upper)|-1(lower)]
```

cur_mcsset

Gets the current modulation coding scheme (MCS) set.

Syntax: **wl cur_mcsset**

Gets the current MCSSET if associated, else default MCSSET.

Example return: `MCS SET: [0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15]`

The modulation coding scheme includes variables such as the number of spatial streams, modulation, and the data rate on each stream. Radios establishing and maintaining a link must automatically negotiate the optimum MCS based on channel conditions and then continuously adjust the selection of MCS as conditions change due to interference, motion, fading, and other events.

The following table shows an example of how MCSs are specified.

MCS Index	*Modulation*	*Code Rate*	N_{BPSC}[a] *(iSS)*[b]	N_{SD}[c]	N_{SP}[d]	N_{CBPS}[e]	N_{DPSC}[f]	*Data Rate (Mbps)*	
								800 ns GI[g]	*400 ns GI*
0	BPSC	1/2	1	108	6	108	54	13.5	15.0
1	QPSK	1/2	2	108	6	216	108	27.0	30.0
2	QPSK	3/4	2	108	6	216	162	40.5	45.0
3	16-QAM	1/2	4	108	6	432	216	54.0	60.0
4	16-QAM	3/4	4	108	6	432	324	81.0	90.0
5	64-QAM	2/3	6	108	6	648	432	108.0	120.0
6	64-QAM	3/4	6	108	6	648	486	121.5	135.0
7	64-QAM	5/6	6	108	6	648	540	135.0	150.0

a. Number of coded bits per single carrier.

b. Number of coded bits per single carrier for each spatial stream, iSS.

c. Number of data subcarriers.

Broadcom Confidential

d. Number of pilot subcarriers.

e. Number of coded bits per symbol.

f. Number of data bits per symbol.

g. Guard interval (GI) is the time delay used by the receiver to let the reflections in the channel settle before sampling data bits.

chanspecs

Gets all the valid CHANSPECS.

Syntax: **wl chanspecs**

Default: all within the current locale

Options/arguments:

-b band (*5*(*a*) *or 2*(*b*/*g*))

-w bandwidth, 10, 20 or 40

[-c country_abbrev]

```
0x2b01 channel 1 2G  20MHZ
0x2b02 channel 2 2G  20MHZ
0x2b03 channel 3 2G  20MHZ
0x2b04 channel 4 2G  20MHZ
0x2b05 channel 5 2G  20MHZ
0x2b06 channel 6 2G  20MHZ
0x2b07 channel 7 2G  20MHZ
0x2b08 channel 8 2G  20MHZ
0x2b09 channel 9 2G  20MHZ
0x2b0a channel 10 2G  20MHZ
0x2b0b channel 11 2G  20MHZ

0x2e03 channel 3 2G  40MHZ(upper)
0x2e04 channel 4 2G  40MHZ(upper)
0x2e05 channel 5 2G  40MHZ(upper)
0x2e06 channel 6 2G  40MHZ(upper)
0x2e07 channel 7 2G  40MHZ(upper)
0x2e08 channel 8 2G  40MHZ(upper)
0x2e09 channel 9 2G  40MHZ(upper)
0x2d03 channel 3 2G  40MHZ(lower)
0x2d04 channel 4 2G  40MHZ(lower)
0x2d05 channel 5 2G  40MHZ(lower)
0x2d06 channel 6 2G  40MHZ(lower)
0x2d07 channel 7 2G  40MHZ(lower)
0x2d08 channel 8 2G  40MHZ(lower)
0x2d09 channel 9 2G  40MHZ(lower)

0x1b24 channel 36 5G  20MHZ
0x1b28 channel 40 5G  20MHZ
0x1b2c channel 44 5G  20MHZ
0x1b30 channel 48 5G  20MHZ
0x1b34 channel 52 5G  20MHZ
```

Broadcom Confidential

```
0x1b38 channel 56 5G  20MHZ
0x1b3c channel 60 5G  20MHZ
0x1b40 channel 64 5G  20MHZ
0x1b95 channel 149 5G 20MHZ
0x1b99 channel 153 5G 20MHZ
0x1b9d channel 157 5G 20MHZ
0x1ba1 channel 161 5G 20MHZ
0x1ba5 channel 165 5G 20MHZ

0x1e26 channel 38 5G  40MHZ(upper)
0x1e2e channel 46 5G  40MHZ(upper)
0x1e36 channel 54 5G  40MHZ(upper)
0x1e3e channel 62 5G  40MHZ(upper)
0x1e97 channel 151 5G 40MHZ(upper)
0x1e9f channel 159 5G 40MHZ(upper)

0x1d26 channel 38 5G  40MHZ(lower)
0x1d2e channel 46 5G  40MHZ(lower)
0x1d36 channel 54 5G  40MHZ(lower)
0x1d3e channel 62 5G  40MHZ(lower)
0x1d97 channel 151 5G  40MHZ(lower)
0x1d9f channel 159 5G  40MHZ(lower)
```

For example, `0x2b01` indicates 2.4-GHz band with 20-MHz bandwidth operating with control channel 1, and so on.

nrate

Forces MIMO (IEEE 802.11n) rates.

Note: MCS indexes 0–7 default to CDD, 8–15 default to SDM.

Syntax: **wl nrate** *-r* [*legacy rate*] *-m* [mcs index] *-s* [stf mode 0=SISO, 1=CDD, 2=STBC, 3=SDM]

Default: Auto

When set, it applies to band-specific rate_override, and when a query, it gets the moving average or band-specific rate_override (if it is on).

For both legacy or MCS, the stf field shows the STF mode (and number of streams) in use.

0 - SISO, 1 stream

1 - CDD, 2 streams

2 - STBC, 2 streams

3 - SDM, 2 streams.

Other - invalid.

Note: For 2 × 2 solutions, the valid nrate is 1–15 and 32.

Broadcom Confidential

nphy_antsel

Gets/sets the antenna configuration.

Syntax: **wl nphy_antsel** (to get values of [utx urx dtx drx])

This command controls the antenna selection feature in the driver. It can enable/disable the various antenna selection algorithms (explained later) and/or manually select/override antenna configurations. The default is AUTO if SROM supports 3 antennas. This command is primarily for internal use. For external use, it should use simple syntax to set:

-1 = auto selection

Setting the antennas:

basic: **wl nphy_antsel** *cfg* (for example, –1 or 0x01)

advanced: **wl nphy_antsel** *utx urx dtx drx*

where

The argument *cfg* in the basic setting controls the antenna configuration for TX and RX of frames. A value of –1 means auto selection, and a value of 0xAB means fixed antennas (A and B are the antenna numbers used for RF chain 0 and 1, respectively).

The four arguments/results [utx urx dtx drx] are as follows:

utx = antenna configuration for transmission of unicast data frames. Set values can be either –1 (auto selection, Algorithm1) or 0xAB. The get value is the current ANTCFG selected by Algorithm1 including a flag "AUTO" or the user-specified TX ANTCFG override 0xAB.

urx = antenna configuration for reception of unicast data frames protected by RTS/CTS. Set values can be either –1 (auto selection, Algorithm2) or 0xAB. The get value is the current ANTCFG selected by Algorithm2 including a flag "AUTO" or the user-specified RX ANTCFG override 0xAB.

dtx = antenna configuration for transmission of frames that are *not* unicast data frames (also known as "default tx configuration"). Set values can be either –1 (auto selection, Algorithm3) or 0xAB. The get value is the current ANTCFG selected by Algorithm3 including a flag "AUTO" or the user-specified TX ANTCFG override 0xAB.

dtx = antenna configuration for reception of all frames, except RTS/CTS protected frames (also known as "default rx configuration"). Set values can be either –1 (auto selection, Algorithm3) or 0xAB. The get value is the current ANTCFG selected by Algorithm3 including a flag "AUTO" or the user-specified RX ANTCFG override 0xAB.

Note: The distinction between dtx and drx is mainly for backward compatibility with 2 × 3 CB "superswitch".

The query will also have a flag "AUTO" if auto selection is ON.

ampdu_tid

Enables/disables PER-TID AMPDU.

Syntax: **wl ampdu_tid** `<tid>` [`0/1`]

This command enables/disables AMPDU on a per-traffic identified (TID) basis.

Broadcom Confidential

ampdu_clear_dump

Clears AMPDU counters.

Syntax: **wl ampdu_clear_dump**

This command allows users to clear AMPDU-specific statistics (WLCNT or wireless related counters), examples of which are counters for WMM and AMSDU.

Broadcom Confidential

Broadcom Corporation

5300 California Avenue
Irvine, CA 92617
Phone: 949-926-5000
Fax: 949-926-5203

Broadcom® Corporation reserves the right to make changes without further notice to any products or data herein to improve reliability, function, or design. Information furnished by Broadcom Corporation is believed to be accurate and reliable. However, Broadcom Corporation does not assume any liability arising out of the application or use of this information, nor the application or use of any product or circuit described herein, neither does it convey any license under its patent rights nor the rights of others.