

U S E R M A N U A L

Copyright (C) 2008-2010, Marvell International Ltd.
All Rights Reserved

1) FOR DRIVER BUILD ← 编译驱动

Goto source code directory wlan_src/.
make [clean] build
The driver and utility binaries can be found in ../bin_xxxx directory.
The driver code supports Linux kernel up to 2.6.33.

2) FOR DRIVER INSTALL ← 安装驱动

a) Copy firmware image sd8786_uapsta.bin | ... to /lib/firmware/mrvl/ directory,

create the directory if it doesn't exist.

b) Install WLAN driver,

For example, to install SD8786 driver,

insmod wlan.ko
insmod sd8786.ko [drv_mode=3] [fw_name=mrvl/sd8786_uapsta.bin]

To load driver in STA only mode

insmod wlan.ko
insmod sd8786.ko drv_mode=1 [fw_name=mrvl/sd8786.bin]

To load driver in uAP only mode

insmod wlan.ko
insmod sd8786.ko drv_mode=2 [fw_name=mrvl/sd8786_uap.bin]

To switch mode between STA only, uAP only and uAPSTA in run time

echo drv_mode=1 > /proc/mwlan/config // STA mode
echo drv_mode=2 > /proc/mwlan/config // uAP mode
echo drv_mode=3 > /proc/mwlan/config // uAPSTA mode

c) Uninstall WLAN driver,
ifconfig wlanX down
ifconfig uapX down
rmmmod sd8xxx
rmmmod wlan

wifi模块的STA和AP模式介绍。

1): AP, 也就是无线接入点, 是一个无线网络的创建者, 是网络的中心节点。一般家庭或办公室使用的无线路由器就是一个AP。wifi模块的AP模式
2): STA站点, 每一个连接到无线网络中的终端(如笔记本电脑、PDA及其它可以联网的用户设备)都可称为一个站点。

To load driver with MFG firmware file, use mfg_mode=1 when insmod WLAN driver and
specify MFG firmware name if needed.

There are some other parameters for debugging purpose. Use modinfo to check details.

mac_addr=xx:xx:xx:xx:xx:xx <override the MAC address (in hex)>
auto_ds=0|1|2 <use MLAN default | enable auto deepsleep | disable auto deepsleep>
ps_mode=0|1|2 <use MLAN default | enable IEEE PS mode | disable IEEE PS mode>
max_tx_buf=2048|4096|8192 <maximum AMSDU Tx buffer size>
pm_keep_power=1|0 <PM keep power in suspend | PM no power in suspend>
cfl1d=0|1|2 <use MLAN default | enable l1d | disable l1d>

3) FOR DRIVER PROC & DEBUG ← 调试驱动

README..txt

The following info are provided in /proc/net/mwlan/mlanX/info,
on kernel 2.6.24 or later, the entry is /proc/mwlan/mlanX/info.

```
driver_name = "wlan"
driver_version = <chip id, firmware version and driver version>
interface_name = "mlanX"
bss_mode = "Ad-hoc" | "Managed" | "Auto" | "Unknown"
media_state = "Disconnected" | "Connected"
mac_address = <6-byte adapter MAC address>
multicast_count = <multicast address count>
essid = <current SSID>
bssid = <current BSSID>
channel = <current channel>
region_code = <current region code>
multicast_address[n] = <multicast address>
num_tx_bytes = <number of bytes sent to device>
num_rx_bytes = <number of bytes received from device and sent to kernel>
num_tx_pkts = <number of packets sent to device>
num_rx_pkts = <number of packets received from device and sent to
kernel>
num_tx_pkts_dropped = <number of Tx packets dropped by driver>
num_rx_pkts_dropped = <number of Rx packets dropped by driver>
num_tx_pkts_err = <number of Tx packets failed to send to device>
num_rx_pkts_err = <number of Rx packets failed to receive from device>
carrier "on" | "off"
tx queue "stopped" | "started"
```

The following debug info are provided in /proc/net/mwlan/mlanX/debug.
on kernel 2.6.24 or later, the entry is /proc/mwlan/mlanX/debug.

```
int_counter = <interrupt count, cleared when interrupt handled>
wmm_ac_vo = <number of packets sent to device from WMM AcVo queue>
wmm_ac_vi = <number of packets sent to device from WMM AcVi queue>
wmm_ac_be = <number of packets sent to device from WMM AcBE queue>
wmm_ac_bk = <number of packets sent to device from WMM AcBK queue>
max_tx_buf_size = <maximum Tx buffer size>
tx_buf_size = <current Tx buffer size>
curr_tx_buf_size = <current Tx buffer size in FW>
ps_mode = <0/1, CAM mode/PS mode>
ps_state = <0/1/2/3, full power state/awake state/pre-sleep state/sleep
state>
is_deep_sleep = <0/1, not deep sleep state/deep sleep state>
wakeup_dev_req = <0/1, wakeup device not required/required>
wakeup_tries = <wakeup device count, cleared when device awake>
hs_configured = <0/1, host sleep not configured/configured>
hs_activated = <0/1, extended host sleep not activated/activated>
num_tx_timeout = <number of Tx timeout>
num_cmd_timeout = <number of timeout commands>
timeout_cmd_id = <command id of the last timeout command>
timeout_cmd_act = <command action of the last timeout command>
last_cmd_id = <command id of the last several commands sent to device>
last_cmd_act = <command action of the last several commands sent to
device>
last_cmd_index = <0 based last command index>
last_cmd_resp_id = <command id of the last several command responses
received from device>
```

README..txt

```
last_cmd_resp_index = <0 based last command response index>
last_event = <event id of the last several events received from device>
last_event_index = <0 based last event index>
num_cmd_h2c_fail = <number of commands failed to send to device>
num_cmd_sleep_cfm_fail = <number of sleep confirm failed to send to
device>
num_tx_h2c_fail = <number of data packets failed to send to device>
num_evt_deauth = <number of deauthenticated events received from device>
num_evt_disassoc = <number of disassociated events received from device>
num_evt_link_lost = <number of link lost events received from device>
num_cmd_deauth = <number of deauthenticate commands sent to device>
num_cmd_assoc_ok = <number of associate commands with success return>
num_cmd_assoc_fail = <number of associate commands with failure return>
cmd_sent = <0/1, send command resources available/sending command to
device>
data_sent = <0/1, send data resources available/sending data to device>
mp_rd_bitmap = <SDIO multi-port read bitmap>
mp_wr_bitmap = <SDIO multi-port write bitmap>
cmd_resp_received = <0/1, no cmd response to process/response received
and yet to process>
event_received = <0/1, no event to process/event received and yet to
process>
ioctl_pending = <number of ioctl pending>
tx_pending = <number of Tx packet pending>
rx_pending = <number of Rx packet pending>
malloc_count = <number of malloc done>
lock_count = <number of lock used>
```

Use dmesg or cat /var/log/debug to check driver debug messages.

Update /proc/sys/kernel/printk to change message log levels.

For example,

```
echo 6 > /proc/sys/kernel/printk    (messages with a higher priority
than 6                                will be printed to the console)
echo 15 > /proc/sys/kernel/printk    (all messages will be printed to
console)
```

4) FOR IWPRIV COMMAND

NAME

This manual describes the usage of private commands used in Marvell MLAN Linux Driver.

To use parameters as hex format, a '0x' must precede it for the parameters to be parsed properly.

SYNOPSIS

iwpriv <mlanX> <command> [sub-command] ...

```
iwpriv mlanX version
iwpriv mlanX verext
iwpriv mlanX getsignal [m] [n]
iwpriv mlanX antcfg [m]
iwpriv mlanX regioncode [n]
```

```
iwpriv wlanX wscfg [m]
iwpriv wlanX esuppmode
iwpriv wlanX passphrase <ssid/psk/passphrase>
iwpriv wlanX httcfg <m>
iwpriv wlanX htcapinfo <m>
iwpriv wlanX addbapara <m> <n> <o>
iwpriv wlanX aggrpriortbl <n>
iwpriv wlanX addbareject <n>
iwpriv wlanX txbufcfg <n>
iwpriv wlanX amsduaggrctrl <n>
iwpriv wlanX mpactrl [tx_ena] [rx_ena] [tx_size] [rx_size] [tx_ports]
[rx_ports]
iwpriv wlanX atimwindow [n]
iwpriv wlanX deepsleep [n] [m]
iwpriv wlanX hscfg [condition [[GPIO# [gap]]]]
iwpriv wlanX hssetpara condition [GPIO# [gap]]
iwpriv wlanX deauth [n]
iwpriv wlanX radioctrl
iwpriv wlanX reassoctrl [n]
iwpriv wlanX adhocaes
iwpriv wlanX bandcfg [1] [m] [n] [o]
iwpriv wlanX getlog
iwpriv wlanX lldcfg
iwpriv wlanX lldclrtbl
iwpriv wlanX wmmcfg [n]
iwpriv wlanX txpowercfg [<RateIndex> [<MinPwr> [<MaxPwr> <step>]]]
iwpriv wlanX qoscfg
iwpriv wlanX getdata rate
iwpriv wlanX txratecfg [n]
iwpriv wlanX bcninterval [n]
iwpriv wlanX sysclock [clk1] [clk2] [clk3] [clk4]
iwpriv wlanX ldofcfg [n]
iwpriv wlanX drvdbg [n] [m]
iwpriv wlanX warmreset
iwpriv wlanX regrdwr <type> <offset> [value]
iwpriv wlanX rdeeprom <offset> <length>
iwpriv wlanX memrdwr <address> [value]
iwpriv wlanX inactivityto <n> <m> <l> [k]
iwpriv wlanX bcats <traffic_type> [<timeshare_interval> <bt_time>]
iwpriv wlanX sdioclock <n>
iwpriv wlanX sdcmd52rw <FN no.> <address> [data]
iwpriv wlanX scancfg [t] [m] [p] [s] [a] [b]
iwpriv wlanX vsiecfg <action> <id> [<mask> [data1] ... [dataN]]
iwpriv wlanX sleepd [n]
iwpriv wlanX pscfg [k] [d] [1] ...
iwpriv wlanX sleepparams [<p1> <p2> <p3> <p4> <p5> <p6>]
iwpriv wlanX authtype [n]
iwpriv wlanX powercons [n]
```

DESCRIPTION

Those commands are used to send additional commands to the Marvell WLAN card via the Linux device driver.

The wlanX parameter specifies the network device that is to be used to perform this command on. It could be wlan0, wlan1 etc.

README..txt

version

This is used to get the current version of the driver and the firmware.

verext

Retrieve and display an extended version string from the firmware

Usage:

```
iwpriv wlanX verext [#]
```

where [#] is an optional argument to retrieve a specific version string, omission of the argument retrieves the 0 indexed string.

getsignal

This command gets the last and average value of RSSI, SNR and NF of Beacon and Data.

Note: This command is available only when STA is connected.

where value of m is:

- 1 -- RSSI (Receive Signal Strength Indication)
- 2 -- SNR (Signal to Noise Ratio)
- 3 -- NF (Noise Floor)

where value of n is:

- 1 -- Beacon last
- 2 -- Beacon average
- 3 -- Data last
- 4 -- Data average

Examples:

```
iwpriv wlan0 getsignal 1          : Get the RSSI info (beacon
last, beacon                        average, data last and data
average)
iwpriv wlan0 getsignal 3 4         : Get the NF of data average
iwpriv wlan0 getsignal 2 1         : Get the SNR of beacon last
iwpriv wlan0 getsignal             : Get all of the signal info
wlan0      getsignal:-32 -33 -35 -36 67 59 63 56 -99 -92
-98 -92
RSSI info: beacon last -32, beacon average -33, data last -35,
data average -36
SNR info: beacon last 67, beacon average 59, data last 63, data
average 56
NF info: beacon last -99, beacon average -92, data last -98,
data average -92
```

antcfg

This command is used to set/get the mode of Tx/Rx antenna.

where value of m is:

- 1 -- Tx/Rx antenna 1.
- 2 -- Tx/Rx antenna 2.
- 0xFFFF -- Tx/Rx antenna diversity.

Examples:

```
iwpriv wlan0 antcfg             : Get Tx/Rx antenna mode
iwpriv wlan0 antcfg 1           : Set Tx/Rx antenna 1
iwpriv wlan0 antcfg 0xFFFF     : Set Tx/Rx antenna diversity
```

README..txt

regioncode

This command is used to set/get the region code in the station.

Note: This command should be issued at beginning before band/channel selection and association.

where value is 'region code' for various regions like
USA FCC, Canada IC, Europe ETSI, Japan ...

The special code (0xff) is used for Japan to support channel 1-14 in B/G/N mode.

Examples:

```
iwpriv wlan0 regioncode      : Get region code
iwpriv wlan0 regioncode 0x10 : Set region code to USA (0x10)
```

wwscfg

This command is used to set/get the WWS (World Wide Safe) mode.

where value of m is:

```
0      -- Disable WWS mode (default)
1      -- Enable WWS mode
```

Examples:

```
iwpriv wlan0 wwscfg          : Get WWS mode
iwpriv wlan0 wwscfg 1        : Enable WWS mode
iwpriv wlan0 wwscfg 0        : Disable WWS mode
```

esuppmode

This command is used to get the current RSN mode and active pairwise/group cipher for WPA/WPA2 mode.

Note: This command is available only when STA is connected.

These are bits settings used to indicate each RSN mode.

```
Bit 0    : No RSN
Bit 1-2   : RFU
Bit 3     : WPA
Bit 4     : WPA-NONE
Bit 5     : WPA2
Bit 6     : AES CCKM
Bit 7-15  : RFU
```

These are bits settings used to indicate each pairwise and group cipher.

```
Bit 0     : RFU
Bit 1     : RFU
Bit 2     : TKIP
Bit 3     : AES CCKM
Bit 2-7   : RFU
```

Example:

```
iwpriv wlan0 esuppmode      : Get RSN mode and
pairwise/group cipher
8 4 4
(The current RSN mode is WPA, active pairwise cipher is TKIP and
```

README..txt
active group cipher is TKIP.)

passphrase

This command is used to set/get passphrase for WPA-PSK/WPA2-PSK mode.

Where <n>

ASCII string for ssid/passphrase/psk.

- 1) "0;<ssid=valid ssid>" - This will get the passphrase, AKMP for specified ssid, if none specified then it will get all.

Example:

```
iwpriv wlan0 passphrase "0;ssid=marvell"
```

- 2) "1;<psk=64 byte hexpsk>;<passphrase=1-63 byte passphrase>
<ssid=valid ssid>" - Passphrase and psk cannot be provided for the

same SSID.

This command takes only one SSID at a time, If ssid= is present it should contain

a passphrase or psk. If no arguments are provided then AKMP=802.1x, and passphrase

should be provided after association.

End of each parameter should be followed by a ';' (except for the last parameter)

as the delimiter. If ';' or '/' has to be used in an SSID then a '/' should be preceded

to ';' or '/' as a escape.

Examples:

```
iwpriv wlan0 passphrase "1;ssid=mrwlAP;passphrase=abcdefgd"  
iwpriv wlan0 passphrase "1;ssid=mrwl AP;psk=<64 bytes hexpsk>"
```

If user wants to input the ssid as "mrwl; AP" then command has to be

```
iwpriv wlan0 passphrase "1;ssid=mrwl/; AP;passphrase=abcdefgh"
```

If user wants to input the ssid as "///;" then command has to be

```
iwpriv wlan0 passphrase "1;ssid=/////;;passphrase=abcdefgh"
```

- 3) "2;<ssid=valid ssid>" - This will clear the passphrase for specified ssid, if none specified then it will clear all.

Examples:

```
iwpriv wlan0 passphrase "2;ssid=marvell"  
iwpriv wlan0 passphrase "2"
```

: Clear all profiles and disable embedded supplicant

httxcfg

This command is used to configure various 11n specific configuration for transmit (such as Short GI, Channel BW and Green field support)

where <m>

This is a bitmap and should be used as following

Bit 15-7: Reserved set to 0

Bit 6: Short GI in 40 Mhz enable/disable

Bit 5: Short GI in 20 Mhz enable/disable

README..txt

Bit 4: Green field enable/disable
Bit 3-2: Reserved set to 0
Bit 1: 20/40 Mhz enable disable.
Bit 0: Reserved set to 0

When Bit 1 is set then firmware could transmit in 20Mhz or 40Mhz based on rate adaptation. When this bit is reset then firmware will only transmit in 20Mhz.

```
iwpriv wlanX httxcfg 0x62
```

This will enable 20/40 and Short GI but will disable Green field.

```
iwpriv wlanX httxcfg 0x30
```

This will enable Short GI and Green field.

The default value is 0x20

Note:- If 20/40 MHz support is disabled in htcapinfo, device will not transmit in 40 MHz even 20/40 MHz is enabled in httxcfg.

htcapinfo

This command is used to configure some of parameters in HTCInfo IE (such as Short GI, Channel BW, and Green field support)

where <m>

This is a bitmap and should be used as following

Bit 29: Green field enable/disable
Bit 26: Rx STBC Support enable/disable. (As we support single spatial stream only 1 bit is used for Rx STBC)
Bit 24: Short GI in 40 Mhz enable/disable
Bit 23: Short GI in 20 Mhz enable/disable
Bit 17: 20/40 Mhz enable disable.
Bit 8: Enable/disable 40Mhz Intolarent bit in ht capinfo.
0 will reset this bit and 1 will set this bit in htcapinfo attached in assoc request.
All others are reserved and should be set to 0.

Setting of any other bits will return error.

```
iwpriv wlanX htcapinfo 0x1820000
```

This will enable Short GI, Channel BW to 20/40 and disable Green field support.

```
iwpriv wlanX htcapinfo 0x800000
```

This will enable Short GI, Channel BW to 20 only, No Rx STBC support and disable Green field support.

The default value is 0x4800000

Note:- This command can be issued any time but it will only come to effect from next association. (as HTCInfo is sent only during Association).

addbapara

This command can be used to update the default ADDBA parameters.

README..txt

where <m> is <timeout>
<timeout> - This is the block ack timeout for ADDBA request.
0 : Disable (recommended for throughput test)
1 - 65535 : Block Ack Timeout in TU

where <n> is <txwinsize>
<txwinsize> - Window size for ADDBA request. (32 is recommended and default value)

where <o> is <rxwinsize>
<rxwinsize> - Window size for ADDBA response. (16 is recommended value for most APs, 64

is recommended for AP85)

Current window size limit for Tx as well as Rx is 1023.

eg:
iwpriv wlanX addbapara - This command will get the current addba params
iwpriv wlanX addbapara 1000 64 5 - This will change the ADDBA timeout to (1000 * 1024) us,
txwinsize to 64 and rxwinsize to 5.

The default setting is 65535 32 16.

In case the ADDBA timeout value is updated then a ADDBA is sent for all streams to update the timeout value.

In case txwinsize and/or rxwinsize is updated, the effect could only be seen on next ADDBA request/response. The current streams will not be affected with this change.

aggrpriortbl

This command is used set/get the priority table for AMPDU/AMSDU traffic per tid.

This command can also be used to disable AMPDU/AMSDU for a given tid.

In case of AMPDU this priority table will be used to setup block ack (to make sure the highest priority tid always uses AMPDU as we have limited AMPDU streams)

where <m0> <n0> <m1> <n1> ... <m7> <n7>

<mx> - This is priority for Tid0 for AMPDU packet. A priority could be any values between 0 - 7, 0xff to disable aggregation.

<n7> - This is priority for Tid7 for AMSDU packet. A priority could be any values between 0 - 7, 0xff to disable aggregation.

eg:
iwpriv wlanX aggrpriortbl - This command will get the current Priority table for AMPDU and AMSDU.

README..txt

255 255 255>. This is read as

<2 2 0 0 1 1 3 3 4 4 5 5 255

"Prio for AMSDU for Tid0"

<"Prio for AMPDU for Tid0"

"Prio for AMSDU for Tid1" and so on

"Prio for AMPDU for Tid1"

iwpriv mlanX aggrpriortbl 2 2 0 0 1 1 3 3 4 4 5 5 255 255 255 255 -

This will set the priority table

for AMPDU and AMSDU

Priority for Tid0/AMPDU = 2,

Tid0/AMSDU = 2, Tid1/AMPDU = 0, Tid1/AMSDU = 0

and so on. Aggregation for Tid6

and Tid7 are disabled.

Here higher the priority number,

higher the priority (i.e. 7

has higher priority than 6).

Similarly for AMSDU.

iwpriv mlanX aggrpriortbl 0xff 2 0xff 0 0xff 1 0xff 3 0xff 4 0xff 5 0xff
0xff 0xff 0xff - This will disable

AMPDU for all the TIDs but will

still keep AMSDU enabled to Tid0 to Tid5

The default setting is 1 1 2 2 0 0 3 3 4 4 5 5 255 255 255 255.

A delBA should be seen in case a disable happens on a TID for which
AMPDU stream
is currently setup.

Note:- This command should only be issue in disconnected state.

addbareject

This command is used set/get the addbareject table for all the TIDs.

This command can also be used to enable rejection of ADDBA requests for
a given tid.

where <m0> <m1> ... <m7>

<mX> - This can be 0/1 for TidX. 1 enables rejection of ADDBA request
for TidX and

0 would accept any ADDBAs for TidX.

eg:

iwpriv mlanX addbareject - This command will get the current table.

[0 0 0 0 0 0 0 0]. ADDBA would be accepted for all TIDs. This is the
default state.

iwpriv mlanX addbareject 0 0 1 1 0 0 0 0 - This command will accept
ADDDBA requests for

Tid [0,1,4,5,6,7] and reject ADDBA requests for Tid [2,3]

iwpriv mlanX addbareject 1 1 1 1 1 1 1 1 - This will enable rejection of
ADDDBA requests for
all Tids.

Note:- This command should only be issue in disconnected state.

README..txt

txbufcfg

This command can be used to set max transmit buffer size of firmware.
Increasing this buffer size is recommended for AMSDU packets.

where <n> is <buffer size in byte>

<buffer size> - This can be 2048/4096/8192, default is 2048.

eg:

iwpriv wlanX txbufcfg - This will display the current buffer size.

iwpriv wlanX txbufcfg 8192 - This will change the tx buffer size of 8192.

For AMSDU for work for multiple packets we may need the max tx buffer size to be 4096/8192.

Note:- This command should be issue in disconnected state.

Otherwise, new setting will be effected in next time associate.

The actual tx buf size will depends on AP's capability and max transmit buffer size.

amsduaggrctrl

This command could be used to enable/disable a feature where firmware gives feedback to driver

regarding the optimal AMSDU buffer size to use with the current rate.

Firmware will use the

current rate to decide the buffer size we could transmit. The max buffer size will still be

limited by buffer size provided in txbufcfg. (i.e. if the txbufcfg is 4K, then we could only transmit

4K/2K AMSDU packets, if the txbufcfg is 8K then we could transmit 8k/4k/2k based on current rate)

If enabled AMSDU buffer size at various rates will be as follows

1. Legacy B/G rate.
No AMSDU aggregation.

2. BW20 HT Rate:
When TX rate goes down,
MCS 7, 6, 5, 4:
a 8K aggregation size (if TX buffer size is 8K)
b 4K aggregation size (if TX buffer size is 4K)
c 2K aggregation size (if TX buffer size is 2K)

MCS 3, 2:
a 4K aggregation size (if TX buffer size is 8K/4K)
b 2K aggregation size (if TX buffer size is 2K)

MCS 1, 0:
a No aggregation

When TX rate goes up,
MCS 7, 6, 5:

README..txt

a 8K aggregation size (if TX buffer size is 8K)
b 4K aggregation size (if TX buffer size is 4K)
c 2K aggregation size (if TX buffer size is 2K)

MCS 4, 3:

a 4K aggregation size (if TX buffer size is 8K/4K)
b 2K aggregation size (if TX buffer size is 2K)

MCS 2, 1, 0:

a No aggregation

3. BW40 HT Rate:

When TX rate goes down,

MCS 7, 6, 5, 4, 3, 2, 1:

a 8K aggregation size (if TX buffer size is 8K)
b 4K aggregation size (if TX buffer size is 4K)
c 2K aggregation size (if TX buffer size is 2K)

MCS 0:

a No aggregation

When TX rate goes up,

MCS 7, 6, 5, 4, 3:

a 8K aggregation size (if TX buffer size is 8K)
b 4K aggregation size (if TX buffer size is 4K)
c 2K aggregation size (if TX buffer size is 2K)

MCS 2, 1, 0:

a No aggregation

where <n> is 0/1 (for disable/enable)

eg:

iwpriv wlanx amsduaggrctrl 1 - Enable this feature

iwpriv wlanx amsduaggrctrl 0 - Disable this feature

iwpriv wlanx amsduaggrctrl - This will get the enable/disable flag and the current AMSDU buffer size). The AMSDU buffer size returned is

only

valid after association as before association there is no rate info.

Note:- This command to enable/disable could be given anytime

(before/after

association). This feature is enabled by default by the

driver during

initialization.

mpactrl

This command is used to set/get the Tx, Rx SDIO aggregation parameters.

Note: The parameters can be set only in disconnected state.

Usage:

iwpriv wlanX mpactrl [tx_ena] [rx_ena] [tx_size] [rx_size]
[tx_ports] [rx_ports]

where the parameter are:

[tx_ena]: Enable/disable (1/0) Tx MP-A

README..txt

[rx_ena]: Enable/disable (1/0) Rx MP-A
[tx_size]: Size of Tx MP-A buffer
[rx_size]: Size of Rx MP-A buffer
[tx_ports]: Max ports (1-8) for Tx MP-A
[rx_ports]: Max ports (1-8) for Rx MP-A

default values are 1 1 8192 16384 8 8

The MP-A may be disabled by default at build time if the MMC driver byte mode patch is not available in kernel.

Examples:

```
iwpriv wlan0 mpactrl          : Get MP aggregation parameters
iwpriv wlan0 mpactrl 0 0      : Disable MP aggregation for Tx, Rx
respectively
iwpriv wlan0 mpactrl 1 1 2048 2048 8 8
                               : Enable MP aggregation for Tx, Rx
                               : Set Tx, Rx buffer size to 2048
bytes
                               : Set maximum Tx ports to 8 and
maximum Rx ports to 8
```

atimwindow

This command is used to set/get the ATIM window value in the station.
The range of ATIM window is 0 - 50.

Note: This command should be issued before ad-hoc start/join and ad-hoc power save on.

Examples:

```
iwpriv wlan0 atimwindow 20    : Set atimwindow to 20
iwpriv wlan0 atimwindow       : Get atimwindow
```

deepsleep

This command is used to set/get auto deep sleep mode.

Usage:

```
iwpriv wlanX deepsleep [n] [m]
```

where the parameters are:

[n]: Enable/disable auto deep sleep mode (1/0)
[m]: Idle time in milliseconds after which firmware will put the

device

in deep sleep mode. Default value is 100 ms.

Examples:

```
iwpriv wlan0 deepsleep          : Display auto deep sleep mode
iwpriv wlan0 deepsleep 1        : Enable auto deep sleep mode,
idle time unchanged
iwpriv wlan0 deepsleep 0        : Disable auto deep sleep mode
iwpriv wlan0 deepsleep 1 500    : Enable auto deep sleep mode
with idle time 500 ms
```

hscfg

This command is used to configure the host sleep parameters.

Usage:

README..txt

iwpriv wlanX hscfg [condition [[GPIO# [gap]]]]

This command takes one (condition), two (condition and GPIO#) or three (condition, GPIO# and gap) parameters for set. If no parameter provided, get is performed.

where Condition is:

bit 0 = 1 -- broadcast data
bit 1 = 1 -- unicast data
bit 2 = 1 -- mac event
bit 3 = 1 -- multicast packet
bit 4 = 1 -- Wakeup when all the 16 STAs with special IE added

to database.

bit 5 = 1 -- Wakeup when a new STA is added to the database.

The host sleep mode will be canceled if condition is set to -1. The default is 0xf.

where GPIO is the pin number of GPIO used to wakeup the host. It could be any valid GPIO pin# (e.g. 0-7) or 0xff (interface, e.g. SDIO will be used instead). The default is 0xff.

where Gap is the gap in milliseconds between wakeup signal and wakeup event or 0xff for special setting. The default is 0.

The host sleep set except for cancellation will be blocked if host sleep is already activated.

Examples:

iwpriv wlan0 hscfg : Get current host sleep mode
iwpriv wlan0 hscfg -1 : Cancel host sleep mode
iwpriv wlan0 hscfg 3 : Broadcast and unicast data
 Use GPIO and gap set

previously

iwpriv wlan0 hscfg 2 3 : Unicast data
 Use GPIO 3 and gap set

previously

iwpriv wlan0 hscfg 2 1 0xa0 : Unicast data
 Use GPIO 1 and gap 160 ms
iwpriv wlan0 hscfg 2 0xff : Unicast data
 Use interface (e.g. SDIO)
 Use gap set previously
iwpriv wlan0 hscfg 4 3 0xff : MAC event
 Use GPIO 3
 Special host sleep mode
iwpriv wlan0 hscfg 1 0xff 0xff : Broadcast data
 Use interface (e.g. SDIO)
 Special host sleep mode

hsssetpara

This command is used to set host sleep parameters.

README..txt

Usage:

```
iwpriv wlanX hsetpara Condition [GPIO# [gap]]
```

Note:

- 1) The usages of parameters are the same as "hscfg" command.
- 2) The parameters will be saved in the driver and be used when host

suspends.

deauth

This command is used to send a de-authentication to an arbitrary AP. If [n] is omitted, the driver will deauth the associated AP. If in ad-hoc mode this command is used to stop beacon transmission from the station and go into idle state.

When <n> is supplied as a MAC address, the driver will deauth the specified AP. If the AP address matches the driver's associated AP, the driver will disconnect. Otherwise, the driver remains connected.

radioctrl

This command is used to turn on/off the radio.

Note: The radio can be disabled only in disconnected state.

where value of n is:

```
0  -- Disable
1  -- Enable
```

Examples:

```
iwpriv wlan0 radioctrl 1      : Turn the radio on
iwpriv wlan0 radioctrl        : Get radio status
```

reassoctrl

This command is used to turn on/off re-association in driver.

Usage:

```
iwpriv wlanX reassoctrl [n]
```

Where value of n is:

```
0  -- Disable
1  -- Enable
```

Examples:

```
iwpriv wlan0 reassoctrl      : Get re-association status
iwpriv wlan0 reassoctrl 1    : Turn re-association on
```

adhocaes

This command is used to set/get the AES key, when the station is in ad-hoc mode.

Note: This command is only available in disconnected state.

where value can be any 16 byte value.

Examples:

```
iwpriv wlan0 adhocaes        : Get ad-hoc aes key
iwpriv wlan0 adhocaes "1;12345678901234567890123456789012"
                               : Set ad-hoc aes key
iwpriv wlan0 adhocaes 2      : Clear ad-hoc aes key
```

bandcfg

This command is used to set/get infra/ad-hoc band.

Note: This command is only available in disconnected state.

Usage:

```
iwpriv wlanX bandcfg [l] [m] [n] [o]
```

where the parameters:

[l]: Infrastructure band

bit 0: B

bit 1: G

bit 2: A

bit 3: GN

bit 4: AN

[m]: Ad-hoc start band

bit 0: B

bit 1: G

bit 2: A

bit 3: GN

bit 4: AN

[n]: Ad-hoc start channel

[o]: 0 - no secondary channel

1 - secondary channel is above

3 - secondary channel is below

Examples:

```
ad-hoc      iwpriv wlan0 bandcfg          : Get infra/ad-hoc band and
start band  iwpriv wlan0 bandcfg          : Get infra/ad-hoc band and
to 6        iwpriv wlan0 bandcfg          : Get infra/ad-hoc band and
ad-hoc start band  iwpriv wlan0 bandcfg 1      : Set infra band to B only
channel to 6 and   iwpriv wlan0 bandcfg 3 2 6  : Set infra band to B/G, ad-hoc
                  to G and ad-hoc start channel
                  to B/G/GN,
                  secondary channel to above
```

getlog

This command is used to get the statistics available in the station.

l1dcfg

This command is used to control l1D. No argument is used to get.

where value of n is:

0 -- Disable

1 -- Enable

Examples:

```
iwpriv wlan0 l1dcfg 1      : Enable l1D
iwpriv wlan0 l1dcfg        : Get l1D status
```

l1dclrtbl

This command is used to clear the 11D channel table.

Usage:

```
iwpriv wlanX 11dclrtbl
```

wmmcfg

This command is used to control WMM. No argument is used to get.

where value of n is:

```
0  -- Disable
1  -- Enable
```

Examples:

```
iwpriv wlan0 wmmcfg 1      : Enable WMM
iwpriv wlan0 wmmcfg        : Get WMM status
```

txpowercfg

This command is used to get/set the Tx power configuration.

Where

<RateIndex> - Data rate index

```
0      1 Mbps
1      2 Mbps
2      5.5 Mbps
3      11 Mbps
4      6 Mbps
5      9 Mbps
6      12 Mbps
7      18 Mbps
8      24 Mbps
9      36 Mbps
10     48 Mbps
11     54 Mbps
12     MCS0 (BW20)
13     MCS1 (BW20)
14     MCS2 (BW20)
15     MCS3 (BW20)
16     MCS4 (BW20)
17     MCS5 (BW20)
18     MCS6 (BW20)
19     MCS7 (BW20)
140    MCS0 (BW40)
141    MCS1 (BW40)
142    MCS2 (BW40)
143    MCS3 (BW40)
144    MCS4 (BW40)
145    MCS5 (BW40)
146    MCS6 (BW40)
147    MCS7 (BW40)
0xff   Default
```

<MinPwr> - Minimum power level in dBm

<MaxPwr> - Maximum power level in dBm

<step> - Power step

Note: Firmware may adjust the setting if over limit, use get command to check the current setting.

README..txt

Examples:

```
configuration      iwpriv wlan0 txpowercfg 0xff          : Default power
data rate 54 Mbps  iwpriv wlan0 txpowercfg 11 12         : Set power level 12 dBm to
16 dBm with        iwpriv wlan0 txpowercfg 7 11 16 1     : Set power level 11 dBm to
Mbps                                                       step 1 to data rate 18
                  iwpriv wlan0 txpowercfg               : Get current configuration
18 2               wlan0 txpowercfg:2 3 13 18 2 1 1 13 18 2 0 0 13
17 2               10 11 13 15 2 8 9 13 16 2 6 7 13 17 2 4 5 13
13 17 2            17 19 13 15 2 15 16 13 16 2 13 14 13 17 2 12 12
140 140 13 14 1    145 147 13 14 1 143 144 13 14 1 141 142 13 14 1
```

2 -> First rate index is 5.5 Mbps.
3 -> Last rate index is 11 Mbps.
13 -> Min Tx power value is 13 dBm.
18 -> Max Tx power value is 18 dBm.
2 -> Power adjustment step value is 2.

Similarly

17 -> First rate index is MCS5 (BW20).
19 -> Last rate index is MCS7 (BW20).
13 -> Min Tx power value is 13 dBm.
15 -> Max Tx power value is 15 dBm.
2 -> Power adjustment step value is 2.

so on...

qoscfg

This command sets WMM IE QOS info when an argument is given, and gets current WMM IE QOS info when no argument is given.

Examples:

```
iwpriv wlan0 qoscfg 0x0f          : Set WMM IE QOS info to 0x0f
iwpriv wlan0 qoscfg               : Get WMM IE QOS info
```

getdatarate

This command is used to get the data rate (index) being used in last Tx packet and last Rx packet.

txratecfg

This command is used to set/get the transmit data rate.

Note:

1) The data rate can be set only after association.

2) If the reassoc is OFF driver reset the data rate to auto if the connection state is disconnected.

README..txt

Please note that user has to re-issue the set data rate command if the driver is disconnected.

3) If the reassoc is ON driver remembers the data rate set by the user, if the driver is disconnected user does not have to re-issue the set data rate again.

Where <n>

data rate	Data rate
0	1 Mbps
1	2 Mbps
2	5.5 Mbps
3	11 Mbps
4	6 Mbps
5	9 Mbps
6	12 Mbps
7	18 Mbps
8	24 Mbps
9	36 Mbps
10	48 Mbps
11	54 Mbps
12	MCS0
13	MCS1
14	MCS2
15	MCS3
16	MCS4
17	MCS5
18	MCS6
19	MCS7
44	MCS32
0xff	Auto

Examples:

```
iwpriv wlan0 txratecfg 3      : Set fixed Tx rate to 11 Mbps
iwpriv wlan0 txratecfg 11     : Set fixed Tx rate to 54 Mbps
iwpriv wlan0 txratecfg 15     : Set fixed Tx rate to MCS3
iwpriv wlan0 txratecfg 0xff   : Disable fixed rate and uses
```

auto rate

```
iwpriv wlan0 txratecfg      : Read the current data rate
```

setting

bcninterval

This command is used to set/get the beacon interval in ad-hoc mode. The valid beacon interval is between 20 - 1000, default beacon interval is 100.

Where <n>

Beacon interval in TU (Time Unit: 1024 us).

Examples:

```
iwpriv wlan0 bcninterval 200 : Set ad-hoc beacon interval to
```

200

```
iwpriv wlan0 bcninterval      : Get ad-hoc beacon interval
```

sysclock

README..txt

This command is used to set/get system clocks in MHz.
The current system clock, configurable system clocks and all of the supported system clocks will be returned if no parameter provided.

Examples:

```
iwpriv wlan0 sysclock          : Get system clocks
80 80 128 128 128 5 11 16 20 22 32 40 44 64 80 106 128 160
(The current system clock is 80 MHz.
```

```
The configurable system clocks of non-security, security,
```

non-security

```
A-MPDU and security A-MPDU are 80 MHz, 128 MHz, 128 MHz and 128
```

MHz.

```
The supported system clocks are 5 MHz, 11 MHz, ..., 160 MHz.)
```

```
iwpriv wlanX sysclock 80      : Set system clock in
```

non-security mode

```
to 80 MHz, no change for
```

others

```
iwpriv wlanX sysclock 0 0 128 : Set system clock in
```

non-security A-MPDU

```
mode to 128 MHz, no changes
```

for others

ldocfg

This command is used to set/get internal/external core power voltage source.

By default firmware uses internal LDO for 1.2V core power supply.

The current setting will be returned if no parameter provided.

Usage:

```
iwpriv wlanX ldocfg [n]
```

where the parameter is,

```
0      -- internal
1      -- external
```

drvdbg

This command is used to set/get the bit masks of driver debug message control.

Usage:

```
iwpriv wlanX drvdbg [n] [m]
```

Where the parameter <n> is the generic debug message control bit mask.

The following types of driver debug messages can be dynamically enabled

or

disabled by setting or clearing the corresponding bits,

```
bit 0:  MMSG          PRINTM(MMSG,...)
bit 1:  MFATAL        PRINTM(MFATAL,...)
bit 2:  MERROR        PRINTM(MERROR,...)
bit 3:  MDATA         PRINTM(MDATA,...)
bit 4:  MCMND         PRINTM(MCMND,...)
bit 5:  MEVENT        PRINTM(MEVENT,...)
bit 6:  MINTR         PRINTM(MINTR,...)
bit 7:  MIOCTL        PRINTM(MIOCTL,...)
```

...

README..txt

```

bit 16: MDAT_D          PRINTM(MDAT_D,...),
DBG_HEXDUMP(MDAT_D,...)
bit 17: MCMD_D          PRINTM(MCMD_D,...),
DBG_HEXDUMP(MCMD_D,...)
bit 18: MFW_D           PRINTM(MFW_D,...)
...
bit 28: MENTRY          PRINTM(MENTRY,...), ENTER(), LEAVE()
bit 29: MWARN           PRINTM(MWARN,...)
bit 30: MINFO           PRINTM(MINFO,...)

```

Where the parameter <m> is the extended interface module debug message control bit mask. The following types of debug messages can be controlled.

```

bit 0: MIF_D            PRINTM(MIF_D,...),
DBG_HEXDUMP(MIF_D,...)

```

If CONFIG_DEBUG=2, all kinds of debug messages can be configured. By default all debug messages are enabled except for MEVENT and MIF_D.

If CONFIG_DEBUG=1, all kinds of debug messages can be configured except for MENTRY, MWARN and MINFO. By default MMSG, MFATAL and MERROR are enabled.

Some special debug messages,

```

'*'          // MLAN driver ISR is called (bit 6 MINTR
enabled)
'|'          // PS awake event is received (bit 5 MEVENT
enabled)
'_'          // PS sleep event is received (bit 5 MEVENT
enabled)
'+'          // PS sleep confirm is sent (bit 5 MEVENT
enabled)

```

Examples:

```

iwpriv wlan0 drvdbg          : Get the current driver debug
masks
iwpriv wlan0 drvdbg 0 0      : Disable all the debug messages
iwpriv wlan0 drvdbg 7        : Enable MMSG, MFATAL and MERROR
messages,
no change for if debug control
iwpriv wlan0 drvdbg 3 1      : Enable MMSG and MFATAL
messages,
enable MIF_D message
iwpriv wlan0 drvdbg -1 -1    : Enable all the debug messages

```

warmreset

This command is used for warm reset of the interface.

Usage:

```
iwpriv wlanX warmreset
```

regrdwr

This command is used to read/write the adapter register.

Usage:

README..txt

`iwpriv wlanX regdwr <type> <offset> [value]`

where the parameters are,

`<type>`: 1:MAC/SOC, 2:BBP, 3:RF, 4:PMIC, 5:CAU
`<offset>`: offset of register
`[value]`: value to be written

Note: If highest bit of a 32-bit value needs to be set, use negative input.

Examples:

`iwpriv wlan0 regdwr 1 0xa060` : Read the MAC register
`iwpriv wlan0 regdwr 1 0xa060 0x12` : Write the MAC register
`iwpriv wlan0 regdwr 1 0xa794 -0x80000000`
: Write 0x80000000 to MAC

register

`iwpriv wlan0 regdwr 1 0xa794 -0x00000001`
: Write 0xffffffff to MAC

register

rdeeprom

This command is used to read the EEPROM contents of the card.

Usage:

`iwpriv wlanX rdeeprom <offset> <length>`

where the parameters are,

`<offset>`: multiples of 4
`<length>`: 4-20, multiples of 4

Example:

`iwpriv wlan0 rdeeprom 0 20` : Read 20 bytes of EEPROM data

from offset 0

memrdwr

This command is used to read/write the adapter memory.

Usage:

`iwpriv wlanX memrdwr <address> [value]`

where the parameters are,

`<address>`: memory address
`[value]`: value to be written

Note: If highest bit of a 32-bit value needs to be set, use negative input.

Examples:

`iwpriv wlan0 memrdwr -0x80000000`
: Read memory address 0x80000000
`iwpriv wlan0 memrdwr -0x80000000 -0x00000001`
: Write 0xffffffff to memory

address 0x80000000

inactivityto

This command is used to set/get the inactivity timeout value, which specifies

when WLAN device is put to sleep.

Usage:

iwpriv wlanX inactivityto <n> <m> <l> [k]

where the parameter are:

<n>: timeout unit in microseconds.

<m>: Inactivity timeout for unicast data.

<l>: Inactivity timeout for multicast data.

[k]: Inactivity timeout for new Rx traffic after PS notification

to AP.

Examples:

iwpriv wlan0 inactivityto : Get the timeout value

iwpriv wlan0 inactivityto 1000 2 3 : Set timeout unit to 1000

us (1 ms),

inactivity timeout for

unicast data is 2 ms,

inactivity timeout for

multicast data is 3 ms

bcats

This command is used to set/get the BCA timeshare parameters.

This command only works after BCA has been enabled.

Usage:

iwpriv wlanX bcats <traffic_type> [<timeshare_interval>

<bt_time>]

where:

<traffic_type>: 0 - Wlan and bluetooth are low priority.

1 - Wlan and bluetooth are high priority.

2 - Wlan and bluetooth are medium priority.

3 - Wlan and bluetooth are medium high priority.

0xffff - Reset fairshare.

If <timeshare_interval> value is not multiple of 10 then floor

value

is taken and the valid range is <20 ... 60,000> in milliseconds.

If <bt_time> value is not multiple of 10 then floor value is

taken

and the valid range is <0 ... timeshare_interval value> in

milliseconds.

Examples:

iwpriv wlan0 bcats 1

: Get the BCA timeshare settings

when wlan

and bluetooth are set to high

priority

iwpriv wlan0 bcats 0xffff

: Reset fairshare, disable all

modes above

that are running, and restore

arbitration

table register values to

before the user

enabled any of the above

fairshare modes.

```
iwpriv wlan0 bcats 1 30 20 : Set wlan and bluetooth to high
priority,                    wlan timeshare_interval to 30
ms and                       bt_time to 20 ms
```

sdio'clock

Turn On(1) or Off(0) the SDIO clock.

Usage:

```
iwpriv wlanX sdio'clock 1 (on)
iwpriv wlanX sdio'clock 0 (off)
iwpriv wlanX sdio'clock (get the current clock state)
```

sdcmd52rw

This command is used to read/write a controller register in Secure Digital I/O Interfaces.

Usage:

```
iwpriv wlanX sdcmd52rw <function number> <register address>
[value]
```

For SDIO MMC driver, only function 0 and 1 access is allowed. And there is a limitation for function 0 write, only vendor specific CCCR registers (0xf0 -0xff) are permitted.

Examples:

```
iwpriv wlan0 sdcmd52rw 1 3
iwpriv wlan0 sdcmd52rw 1 1 0x3f
```

scancfg

This command is used to set/get scan configuration parameters.

Usage:

```
iwpriv wlanX scancfg [t] [m] [p] [s] [a] [b]
```

where the parameters:

[t]: Scan Type (0: Unchanged, 1: Active, 2: Passive, default Active)

[m]: Scan Mode (0: Unchanged, 1: BSS, 2: IBSS, 3: Any, default Any)

[p]: Scan Probes (0: Unchanged, 1-4: Number of probes per channel, default 4)

[s]: Specific Scan Time (0: Unchanged, n: Value in ms, default 110 ms, max 500 ms)

[a]: Active Scan Time (0: Unchanged, n: Value in ms, default 200 ms, max 500 ms)

[b]: Passive Scan Time (0: Unchanged, n: Value in ms, default 200 ms, max 2000 ms)

No change if the parameter is 0 or the parameter is not provided.

Examples:

```
iwpriv wlan0 scancfg : Get all the current scan
configuration settings
```


README..txt

`iwpriv wlan0 scancfg 1 3` : Set scan type to active and scan mode to any, all the other scan configurations are unchanged
`iwpriv wlan0 scancfg 0 1 2 200` : Set scan mode to BSS, number of probes to 2 and specific scan time to 200 ms, all the other scan configurations are unchanged

vsiecfg

This command is used to get/add/remove vendor specific IE.

Usage:

`iwpriv wlanX vsiecfg <action> <id> [<mask> [data1] ... [dataN]]`

where:

`<action>`: 0/1/2: Get/Add/Remove
`<id>`: 0-7: IE index in the driver IE array
`[mask]`: Bit 0: IE for scan
Bit 1: IE for associate
Bit 2: IE for ad-hoc
`[data1]...[dataN]`: IE data

Note: The max data length is 254-byte, IE ID (221) and length is not included.

The max total length of vendor specific IEs for scan/assoc/ad-hoc is 512-byte.

Examples:

`iwpriv wlan0 vsiecfg 0 2` : Get the 3rd IE
`iwpriv wlan0 vsiecfg 1 0 4 0x12 0x34` : Add IE in 1st position for ad-hoc
`iwpriv wlan0 vsiecfg 1 6 3 0x00 0x50 0x43 0x20 0xFF 0xFE` : Add IE in 7th position for scan and associate
`iwpriv wlan0 vsiecfg 2 1` : Remove the 2nd IE

sleepd

This command is used to configure the sleep period of the WLAN device.

Usage:

`iwpriv wlanX sleepd [<period>]`

Where the parameter is:

period: sleep period in milliseconds. Range 10~60. 0 for disable.

Examples:

`iwpriv wlan0 sleepd` : Get sleep period configuration
`iwpriv wlan0 sleepd 10` : Set sleep period to 10 ms

pscfg

This command is used to set/get PS configuration parameters.

README..txt

Usage:

```
iwpriv wlanX pscfg [k] [d] [l] ...
```

Where the parameters:

[k]: Keep alive null packet interval (0: Unchanged, -1: Disable,
n: Interval in seconds)
[d]: DTIM interval (0: Unchanged,
1-5: Value,
65534: DTIM will be ignored, listen interval
will be used,
65533: Closest DTIM to the listen interval
period will be used)
[l]: Local listen interval (0: Unchanged,
-1: Disable,
1-49: Value in beacon intervals,
>= 50: Value in TUs)
[a]: Ad-hoc awake period (0: Unchanged, 1-31: Beacon interval,
255: Firmware
out) will go to sleep after beacon send
[b]: Beacon miss timeout (0: Unchanged, 1-50: Value in
milliseconds, 65535: Disable)
[p]: Delay to PS (0-65535: Value in milliseconds, default
1000ms)
[m]: PS mode (0: Unchanged, 1: Auto mode, 2: PS-Poll mode, 3: PS
Null mode)
No change if parameters are not provided.

Examples:

```
iwpriv wlan0 pscfg : Get all the current PS
configuration settings
iwpriv wlan0 pscfg 3 4 : Set PS keep alive null packet
interval to 3 seconds and DTIM interval to 4, all
the other configurations are unchanged
iwpriv wlan0 pscfg 0 0xffff 10 0 20 : Disable DTIM interval, set
local listen interval to 10 beacon intervals and beacon
miss interval to 20, all the other configurations
are unchanged
iwpriv wlan0 pscfg 0 0 0 0 0 50 : Set delay to PS to 50 ms, keep
the others unchanged
```

sleepparams

This command is used to set the sleepclock configurations

Usage:

```
iwpriv wlanX sleepparams [<p1> <p2> <p3> <p4> <p5> <p6>]
```

where:

p1 is Sleep clock error in ppm (0-65535)
p2 is Wakeup offset in usec (0-65535)
p3 is Clock stabilization time in usec (0-65535)

README..txt

p4 is Control periodic calibration (0-2)
p5 is Control the use of external sleep clock (0-2)
p6 is reserved for debug (0-65535)

Examples:

```
sleepclock configuration
iwpriv wlan0 sleepparams                : Get current
sleepclock configuration
iwpriv wlan0 sleepparams 10 1000 2000 1 0 128 : Set sleepclock
configuration
```

authtype

This command is used to set/get authentication type.

Usage:

```
iwpriv wlanX authtype [n]
```

where <n>

0: 802.11 open system authentication
1: 802.11 shared key authentication
255: allow open system or shared key authentication (default)

Examples:

```
key authentication
iwpriv wlan0 authtype 0           : use open system authentication
iwpriv wlan0 authtype 1           : use shared key authentication
iwpriv wlan0 authtype 255         : allow open system or shared
key authentication
iwpriv wlan0 authtype             : get current setting
```

powercons

This command is used to set the local transmit power constraint.
Value is in dbm unit. This command is only used for ad-hoc start.

Usage:

```
iwpriv wlanX powercons [n]
```

Examples:

```
12 dbm
iwpriv wlanX powercons           : get the current setting
iwpriv wlanX powercons 12        : set local power constraint to
```

=====

USER MANUAL FOR MLANCONFIG

NAME

mlanconfig - configure the additional parameters available for the Marvell mdriver.

SYNOPSIS

```
mlanconfig -v
mlanconfig <wlanX> <command> [parameters] ...
```

```
mlanconfig wlanX hostcmd <bg_scan.conf> bgscfg
mlanconfig wlanX hostcmd <requesttpc.conf> requesttpc
mlanconfig wlanX hostcmd <crypto_test.conf> crypto_test
mlanconfig wlanX hostcmd <subevent.conf> subevent_get
```

README..txt

```
mlanconfig mlanX hostcmd <subevent.conf> subevent_set
mlanconfig mlanX hostcmd <auto_tx.conf> auto_tx_get
mlanconfig mlanX hostcmd <auto_tx.conf> nat_keep_alive
mlanconfig mlanX hostcmd <auto_tx.conf> auto_tx_unreg
mlanconfig mlanx hostcmd <cmd_stadb.conf> set_stadb_ie/set_stadb_oui
mlanconfig mlanx hostcmd <cmd_stadb.conf>
get_stadb_ie_all/get_stadb_ie/get_stadb_oui
mlanconfig mlanx hostcmd <cmd_stadb.conf> clear_stadb_ie_all/clear_stadb_ie
mlanconfig mlanX hostcmd <txrate_cfg.conf> txrate_cfg_get
mlanconfig mlanX hostcmd <txrate_cfg.conf> txrate_cfg_set_bg
mlanconfig mlanX hostcmd <txrate_cfg.conf> txrate_cfg_set_bgn
mlanconfig mlanX hostcmd <txpwrlimit_cfg.conf> txpwrlimit_cfg_get
mlanconfig mlanX hostcmd <txpwrlimit_cfg.conf> txpwrlimit_2g_cfg_set
mlanconfig mlanX hostcmd <txpwrlimit_cfg.conf> txpwrlimit_5g_cfg_set
mlanconfig mlanX hostcmd <11n_2040coex.conf> 2040coex
mlanconfig mlanX hostcmd <robust_btc.conf> robust_btc_get
mlanconfig mlanX hostcmd <robust_btc.conf> robust_btc_enable
mlanconfig mlanX hostcmd <robust_btc.conf> robust_btc_disable
mlanconfig mlanX arpfilter <arpfilter.conf>
mlanconfig mlanX mefcfg <mef.conf>
mlanconfig mlanX cfgdata <register type> <conf file>
mlanconfig mlanX sdcmd52rw <FN no.> <address> [data]
mlanconfig mlanX sdcmd53rw <FN no.> <address> <mode> <blksize> <blknum> [data1]
... [dataN]
mlanconfig mlanX setuserscan [ARGS]
mlanconfig mlanX getscantable
mlanconfig mlanX addts <filename.conf> <section# of tspec> <timeout in ms>
mlanconfig mlanX delts <filename.conf> <section# of tspec>
mlanconfig mlanX qconfig set msdu <lifetime in TUs> [Queue Id: 0-3]
mlanconfig mlanX qconfig get [Queue Id: 0-3]
mlanconfig mlanX qconfig def [Queue Id: 0-3]
mlanconfig mlanX qstats on [Queue Id: 0-3]
mlanconfig mlanX qstats off [Queue Id: 0-3]
mlanconfig mlanX qstats get [Queue Id: 0-3]
mlanconfig mlanX qstatus
mlanconfig mlanX ts_status
mlanconfig mlanX regrdwr <type> <offset> [value]
mlanconfig mlanX memrdwr <address> [value]
mlanconfig mlanX customie <index> <mask> <IE buffer>
```

DESCRIPTION

Those commands are used in Marvell specific application called mlanconfig.

=====

-v

This command is used to display the version of mlanconfig utility.
Usage:

```
mlanconfig -v
```

hostcmd bgscfg

This command is used to configure the various parameters for PPS/UAPSD or normal background scan.

Usage:

```
mlanconfig mlanX hostcmd config/bg_scan.conf bgscfg
```

README..txt

hostcmd requesttpc

This command is used to request 802.11H TPC info.

Usage:

mlanconfig mlanX hostcmd config/requesttpc.conf requesttpc

hostcmd crypto_test

This command is used to test the encryption/decryption API of the firmware.

Usage:

mlanconfig mlanX hostcmd config/crypto_test.conf crypto_test

hostcmd subevent_get

hostcmd subevent_set

This command is used to get/set the configurations for event descriptor interface command.

subevent_get: get subscribed event parameters

subevent_set: set subscribed event parameters

Usage:

mlanconfig mlanX hostcmd config/subevent.conf subevent_get

mlanconfig mlanX hostcmd config/subevent.conf subevent_set

hostcmd auto_tx_get

hostcmd nat_keep_alive

hostcmd auto_tx_unreg

This command is used to configures the Frame Auto Transmission parameters.

auto_tx_get: get auto_tx parameters

nat_keep_alive: register to firmware for sending NAT Keep Alive packet

auto_tx_unreg: unregister to firmware auto_tx

Usage:

mlanconfig mlanX hostcmd config/auto_tx.conf auto_tx_get

mlanconfig mlanX hostcmd config/auto_tx.conf nat_keep_alive

mlanconfig mlanX hostcmd config/auto_tx.conf auto_tx_unreg

hostcmd set_stadb_ie

This command will set the IE data specified below for the mac address.

Usage:

mlanconfig mlanX hostcmd <cmd_stadb.conf> set_stadb_ie

hostcmd set_stadb_oui

This command will set a OUI specified to compare with the vendor specific

IE received in beacon/probe request/probe response. If the OUI matches then that IE will be stored in the data base.

Usage:

mlanconfig mlanX hostcmd <cmd_stadb.conf> set_stadb_oui

hostcmd get_stadb_ie_all

This command will get the mac address of all the stas whose IEs are

stored in
the data base.

Usage:

mlanconfig mlanX hostcmd <cmd_stadb.conf> get_stadb_ie_all

hostcmd get_stadb_ie

This command will get the IE data for the specific mac adress requested.

Usage:

mlanconfig mlanX hostcmd <cmd_stadb.conf> get_stadb_ie

hostcmd get_stadb_oui

This command will get the current comapare OUI set in the firmware.

Usage:

mlanconfig mlanX hostcmd <cmd_stadb.conf> get_stadb_oui

hostcmd clear_stadb_ie_all

This command will clear all the mac address and IEs for all the STAs
stored in
the data base.

Usage:

mlanconfig mlanX hostcmd <cmd_stadb.conf> clear_stadb_ie_all

hostcmd clear_stadb_ie

This command will only clear the mac address and IE from the data base,
if the
mac adress specifid matches an entry in data base.

Usage:

mlanconfig mlanX hostcmd <cmd_stadb.conf> clear_stadb_ie

hostcmd txrate_cfg_get

hostcmd txrate_cfg_set_bg

hostcmd txrate_cfg_set_bgn

This command is used to set/get the transmit data rate.

Usage:

mlanconfig mlanX hostcmd config/txrate_cfg.conf txrate_cfg_get

mlanconfig mlanX hostcmd config/txrate_cfg.conf

txrate_cfg_set_bg

mlanconfig mlanX hostcmd config/txrate_cfg.conf

txrate_cfg_set_bgn

hostcmd txpwrlimit_cfg_get

hostcmd txpwrlimit_2g_cfg_set

hostcmd txpwrlimit_5g_cfg_set

This command is used to set/get the configuration data of Tx power
limitation.

Note: The configuration set should be issued when STA is disconnected.

Usage:

mlanconfig mlanX hostcmd config/txpwrlimit_cfg.conf

txpwrlimit_cfg_get

README..txt

```
mlanconfig mlanX hostcmd config/txpwrlimit_cfg.conf
txpwrlimit_2g_cfg_set
mlanconfig mlanX hostcmd config/txpwrlimit_cfg.conf
txpwrlimit_5g_cfg_set
```

hostcmd 2040coex

This command is used to send the 11n 20/40 Coex command to firmware.
Firmware will send 11n 20/40 Coex management action frame to AP.

Usage:

```
mlanconfig mlanX hostcmd config/11n_2040coex.conf 2040coex
```

hostcmd robust_btc_get

hostcmd robust_btc_enable

hostcmd robust_btc_disable

This command is used to get/set Robust BT Coex.

robust_btc_get: get the current configuration

robust_btc_enable: enable and set the Robust BT Coex timing

robust_btc_disable: disable the Robust BT Coex

Usage:

```
mlanconfig mlanX hostcmd config/robust_btc.conf robust_btc_get
```

```
mlanconfig mlanX hostcmd config/robust_btc.conf
```

robust_btc_enable

```
mlanconfig mlanX hostcmd config/robust_btc.conf
```

robust_btc_disable

arpfilter

This command is used to configure the ARP filtering parameters.

Usage:

```
mlanconfig mlanX arpfilter config/arpfilter.conf
```

mefcfg

This command is used to set MEF settings.

Usage:

```
mlanconfig mlanX mefcfg config/mef.conf
```

cfgdata

This command is used to set/get the configuration data to/from firmware.

Usage:

```
mlanconfig mlanX cfgdata <type> <.conf file name>
```

This command is used to set the cfg data in the .conf file to
firmware.

```
mlanconfig mlanX cfgdata <type>
```

This command is used to get the cfg data from firmware and
display
on to the console.

Where the value of <type> field is:

1 -- Optimal Register download and <.conf file name> is
or_data.conf

2 -- Cal data download and <.conf file name> is cal_data.conf

3 -- PMIC data download and <.conf file name> is
pmic_data.conf

sdcmd52rw

This command is used to read/write a controller register in
Secure Digital I/O Interfaces.

Usage:

mlanconfig mlanX sdcmd52rw <function number> <register address>
[value]

For SDIO MMC driver, only function 0 and 1 access is allowed. And there
is a limitation for function 0 write, only vendor specific CCCR
registers
(0xf0 -0xff) are permitted.

Examples:

mlanconfig mlan0 sdcmd52rw 1 3
mlanconfig mlan0 sdcmd52rw 1 1 0x3f

sdcmd53rw

This command is used to issue a CMD53 read/write data in
Secure Digital I/O Interfaces.

Usage:

mlanconfig mlanX sdcmd53rw <func> <address> <mode> <blksize>
<blknum> [data1] ... [dataN]

where the parameters are,

<func>: function number (0/1/2/..)
<address>: data address
<mode>: byte mode/block mode (0/1)
<blksize>: block size (32/64/..512, NA for byte mode)
<blknum>: block number or byte number
<data1> ... <dataN>: data for write

Note: The total data length is block size * block number for block mode
or byte number for byte mode. The max data length is 2000-byte.
For write the data pattern will be duplicated to data buffer.

Examples:

mlanconfig mlan0 sdcmd53rw 0 0x8000 1 64 2
mlanconfig mlan0 sdcmd53rw 1 0x10000 0 1 5 0x0a 0x0b 0x0c 0x0d

0x0e

setuserscan

Initiate a customized scan and retrieve the results

Usage:

mlanconfig mlanX setuserscan [ARGS]

Where [ARGS]:

ssid="[SSID]" specify a SSID filter for the scan
chan=[chan#][band][mode] where band is [a,b,g,n] and mode is
blank for active or 'p' for passive
bssid=xx:xx:xx:xx:xx:xx specify a BSSID filter for the scan

	README..txt
* wc="[WILDCARD SSID]"	specify a UNIX pattern matching filter (using
	and ?) for SSIDs found in a broadcast probe
(0) keep=[0 or 1]	keep the previous scan results (1), discard
dur=[scan time]	time to scan for each channel in milliseconds
probes=[#]	number of probe requests to send on each chan
	for each broadcast probe required and each
SSID	
	specific probe required (1-4)
type=[1, 2, 3]	BSS type: 1 (Infra), 2 (Adhoc), 3 (Any)

Any combination of the above arguments can be supplied on the command line.

If the chan token is absent, a full channel scan will be completed by driver.

If the dur or probes tokens are absent, the driver default setting will be used. The bssid and ssid fields, if blank, will produce an unfiltered scan.

The type field will default to 3 (Any) and the keep field will default to 0 (Discard).

Examples:

- 1) Perform an active scan on channels 1, 6, and 11 in the 'g' band:
 setuserscan chan=1g,6g,11g
- 2) Perform a passive scan on channel 11 for 20 ms:
 setuserscan chan=11gp dur=20
- 3) Perform an active scan on channels 1, 6, and 11; and a passive scan
on channel 36 in the 'a' band:
 setuserscan chan=1g,6g,11g,36ap
- 4) Perform an active scan on channel 6 and 36 for a specific SSID:
 setuserscan chan=6g,36a ssid="TestAP"
- 5) Scan all available channels (B/G/N, A bands) for a specific BSSID,
keep the current scan table intact, update existing or append new scan
data: setuserscan bssid=00:50:43:20:12:82 keep=1
- 6) Scan channel 6, for all infrastructure networks, sending two probe
requests. Keep the previous scan table intact. Update any duplicate
BSSID/SSID matches with the new scan data:
 setuserscan chan=6g type=1 probes=2 keep=1
- 7) Scan channel 1 and 6, for all networks matching the Mrvl*AP
or AP*Mrvl? patterns and for MrvlTst SSID. Generate 3 broadcast
probes for the patterns and 3 SSID specific probes for MrvlTst on
both channel 1 and channel 6.
 setuserscan chan=1g,6g probes=3 wc="Mrvl*AP" wc="AP*Mrvl?"
ssid="MrvlTst"

- 8) Scan all the channels for specified band.
setuserscan chan=0g

All entries in the scan table (not just the new scan data when keep=1) will be displayed upon completion by use of the getscantable ioctl.

getscantable

Display the current contents of the driver scan table

Usage:

```
mlanconfig mlanX getscantable
mlanconfig mlanX getscantable [#]
mlanconfig mlanX getscantable tsf
mlanconfig mlanX getscantable help
```

- 1) Without argument, the entire scantable is displayed.
- 2) Specifying a # will display detailed information about a specific

scan

- table entry. '0' displays driver cached information regarding the current association (if any).
- 3) The tsf argument will display the entire scan table with the recorded TSF timestamp for the entry.
 - 4) The help argument will display the legend for the capability field.

addts

Send an ADDTS command to the associated AP.

Process a given conf file for a specific TSPEC data block. Send the TSPEC along with any other IEs to the driver/firmware for transmission in an ADDTS request to the associated AP.

Return the execution status of the command as well as the ADDTS response from the AP if any.

Usage:

```
mlanconfig mlanX addts <filename.conf> <section# of tspec>
```

<timeout(ms)>

delts

Send a DELTS command to the associated AP.

Process a given conf file for a specific TSPEC data block. Send the TSPEC along with any other IEs to the driver/firmware for transmission in a DELTS request to the associated AP.

Return the execution status of the command. There is no response to a DELTS from the AP.

Usage:

```
mlanconfig mlanX delts <filename.conf> <section# of tspec>
```

qconfig

Send a WMM AC Queue configuration command to get/set/default params

Configure or get the parameters of a WMM AC queue. The command takes

README..txt

an optional Queue Id as a last parameter. Without the queue id, all queues will be acted upon.

Usage:

0-3] mlanconfig mlanX qconfig set msdu <lifetime in TUs> [Queue Id:
mlanconfig mlanX qconfig get [Queue Id: 0-3]
mlanconfig mlanX qconfig def [Queue Id: 0-3]

qstats

Turn on/off or retrieve and clear the queue statistics for an AC

the Turn the queue statistics collection on/off for a given AC or retrieve
command current accumulated stats and clear them from the firmware. The
takes an optional Queue Id as a last parameter. Without the queue id,
all queues will be acted upon.

Usage:

mlanconfig mlanX qstats on [Queue Id: 0-3]
mlanconfig mlanX qstats off [Queue Id: 0-3]
mlanconfig mlanX qstats get [Queue Id: 0-3]

qstatus

This command retrieves the current status of the WMM queues. If WMM
is enabled then it displays the information for each AC in a table.

Usage:

mlanconfig mlanX qstatus

ts_status

This command queries the FW for the status of TSIDs 0 through 7
configured via call admission control and displays the results in a
table.

Usage:

mlanconfig mlanX ts_status

regrdwr

This command is used to read/write the adapter register.

Usage:

mlanconfig mlanX regrdwr <type> <offset> [value]

where the parameters are,

<type>: 1:MAC/SOC, 2:BBP, 3:RF, 4:PMIC, 5:CAU
<offset>: offset of register
[value]: value to be written

Examples:

mlanconfig mlan0 regrdwr 1 0xa060 : Read the MAC register
mlanconfig mlan0 regrdwr 1 0xa794 0x80000000
: Write 0x80000000 to MAC

register

README..txt

memrdwr

This command is used to read/write the adapter memory.

Usage:

```
mlanconfig mlanX memrdwr <address> [value]
```

where the parameters are,

```
<address>:  memory address
[value]:     value to be written
```

Examples:

```
mlanconfig mlan0 memrdwr 0x80000000 : Read memory address
0x80000000
mlanconfig mlan0 memrdwr 0x80000000 0xffffffff
                                          : Write 0xffffffff to
memory address 0x80000000
```

customie

This command is used to set or get custom IEs for management frames.

Usage : customie [INDEX] [MASK] [IEBuffer]

empty - Get all IE settings

INDEX: 0 - Get/Set IE index 0 setting

1 - Get/Set IE index 1 setting

2 - Get/Set IE index 2 setting

3 - Get/Set IE index 3 setting

-1 - Append/Delete IE automatically

Delete will delete the IE from the matching IE buffer

Append will append the IE to the buffer with the same mask

MASK : Management subtype mask value as per bit definitions

: Bit 0 - Association request

: Bit 1 - Association response

: Bit 2 - Reassociation request

: Bit 3 - Reassociation response

: Bit 4 - Probe request

: Bit 5 - Probe response

: Bit 8 - Beacon

MASK : MASK = 0 to clear the mask and the IE buffer

IEBuffer : IE Buffer in hex (max 256 bytes)

The Buffer should not be space separated.

Example:

```
./mlanconfig mlan0 customie
```

Get IE buffer, subtype mask settings for all indices [0-3].

```
./mlanconfig mlan0 customie 1
```

Get IE buffer and subtype mask for the Index = 1.

```
./mlanconfig mlan0 customie 2 0
```

Clear IE buffer and mask value for Index = 2.

```
./mlanconfig mlan0 customie 3 0x101 0xdd051234567890
```

Set IE buffer and mask value for Index = 3.

```
./mlanconfig mlan0 customie -1 0x101 0xdd051234567890
```

Append the specified IEBuffer at index with mask value of 0x101.

README..txt

- ./mланconfig mлан0 customie -1 0 0xdd051234567890
Delete the specified IEBuffer from all the IEs at 0-3 index.
- ./mланconfig mлан0 customie 2 0 0xdd051234567890
Delete the specified IEBuffer from the IEs at index 2.

U S E R M A N U A L F O R M L A N 2 0 4 0 C O E X

NAME

mлан2040coex - This application handles the 11n 20/40 coexistence operation for the Marvell mdriver

SYNOPSIS

mлан2040coex [-i <intfname>] [hvB]
(If intfname not present then mлан0 assumed)
-h = Help
-v = Version
-B = Run the process in background

U S E R M A N U A L F O R M L A N E V E N T

NAME

mланevent.exe

This tool can be used to listen for and obtain events from the driver through the netlink layer.

Supported events

STA_DEAUTH
STA_ASSOC
BSS_START
BSS_IDLE
BSS_ACTIVE

Details of events

STA_DEAUTH

For this event, the following information is shown:
+ Deauthenticated STA MAC address.
+ Reason for deauthentication.

STA_ASSOC

For this event, the following information is shown:
+ STA MAC address.

BSS_START

README..txt

For this event, the following information is shown:
+ AP MAC address.

BSS_IDLE

For this event, there is no associated information.

BSS_ACTIVE

For this event, there is no associated information.
=====