## NAME

**wlandebug** - set/query 802.11 wireless debugging messages

## SYNOPSIS

**wlandebug** [**-d** | **-i** *ifnet*] [**-flag**|+**flag** ...]

## DESCRIPTION

The **wlandebug** command is a tool for enabling and disabling debugging messages in the wlan(4) module.  Running **wlandebug** without any options will display the current messages enabled for the specified network interface (by default, ''wlan0').  The default debugging level for new interfaces can be set by specifying the **-d** option.  When run as the super-user **wlandebug** can be used to enable and/or disable debugging messages.

To enable debugging messages of a certain *type* use *+type*; to disable such messages use *-type*.  Multiple messages can be enabled and disabled with a single command.

Messages are organized in the following groups:

*debug*      general debugging facilities; equivalent to setting the debug parameter with ifconfig(8).

*dumppkts*  dump packet contents on transmit and receive.

*crypto*     crypto-related work.

*input*      errors encountered during input handling.

*xrate*      extended rate set handling (for 802.11g).

*elemid*     information element processing in 802.11 management frames.

*node*       management of per-station state.

*assoc*      802.11 station association processing; particularly useful to see when stations join and leave a BSS.

*auth*       802.11 station authentication processing.

*scan*       scanning operation; especially useful for debugging problems with not locating an access point.

*output*     errors encountered during output handling.

*state*     wlan(4) state machine operation.

*power*     802.11 power save operation; in hostap mode this enables copious information about buffered frames for stations operating in power save mode.

*hwmp*     trace operation of Hybrid Wireless Mesh Protocol processing.

*dot1xsm*     802.1x state machine operation; not presently meaningful as 802.1x protocol support is implemented in user mode by the hostapd(8) program.

*radius*     radius backend operation as it relates to 802.1x operation; not presently meaningful as 802.1x protocol support is implemented in user mode by the hostapd(8) program.

*raddump*     dump packets exchanged with the radius backend for 802.1x operation; not presently meaningful as 802.1x protocol support is implemented in user mode by the hostapd(8) program.

*mesh*     trace operation of 802.11s mesh protocol processing.

*wpa*     trace operation of the WPA protocol; only partly meaningful as WPA protocol support is mostly implemented in user mode by the hostapd(8) and wpa_supplicant(8) programs.

*acl*     trace operation of the Access Control List (ACL) support; see wlan_acl(4) for more details.

*wme*     trace operation of WME/WMM protocol processing.

*superg*     trace operation of Atheros SuperG protocol processing.

*doth*     trace operation of IEEE 802.11h protocol processing.

*inact*     trace station inactivity processing; in particular, show when stations associated to an access point are dropped due to inactivity.

*roam*     trace station mode roaming between access points.

*rate*     trace transmit rate control operation.

**EXAMPLES**

The following might be used to debug basic station mode operation:

　　　wlandebug -i wlan1 scan+auth+assoc

it enables debug messages while scanning, authenticating to an access point, and associating to an access point.

**SEE ALSO**
　　ifconfig(8)

　　The */usr/src/tools* directory contains some utilities that might be relevant to debug wireless issues.

**NOTES**
　　Different wireless drivers support different debugging messages.  Drivers such as ath(4) and ral(4) that depend on the wlan(4) module for 802.11 protocol processing typically support most of the debugging messages while devices that implement parts of the 802.11 protocol in firmware do not.

　　Some debugging messages are no longer meaningful because protocol processing has moved from the operating system to user mode programs such as hostapd(8) and wpa_supplicant(8).