

NAME

net80211 - standard interface to IEEE 802.11 devices

SYNOPSIS

```
#include <sys/types.h>
#include <sys/socket.h>
#include <net/if.h>
#include <net/ethernet.h>
#include <net80211/ieee80211_ioctl.h>
```

DESCRIPTION

This section describes the standard programming interface to configure and retrieve status information for IEEE 802.11 devices that depend on the wlan(4) module for operation. The interface is via one of the following ioctl(2) calls on a socket:

SIOCG80211 Get configuration or status information.

SIOCS80211 Set configuration information.

These requests are made via a modified *ifreq* structure. This structure is defined as follows:

```
struct ieee80211req {
    char          i_name[IFNAMSIZ];      /* if_name, e.g. "wi0" */
    uint16_t     i_type;                 /* req type */
    int16_t      i_val;                  /* Index or simple value */
    int16_t      i_len;                  /* Index or simple value */
    void         *i_data;                 /* Extra data */
};
```

Requests that are not supported by the underlying device return -1 and set the global variable `errno` to `EOPNOTSUPP`. `SIOCG80211` requests that return data to an application place small values in *i_val* or in a user-specified buffer pointed to by *i_data*. When an indirect buffer is used *i_len* specifies how large the indirect buffer is and on return it is set by the system to the actual amount of data returned.

`SIOCS80211` requests use a similar scheme with data passed to the system taken either from *i_val* or an indirect buffer pointed to by *i_data*.

For `SIOCG80211` the following values of *i_type* are valid:

IEEE80211_IOC_AMPDU

Return whether or not AMPDU is enabled in *i_val*. AMPDU is an aggregation scheme that is

part of the 802.11n specification and is used only when operating on an HT channel. The value returned is one of: 0 (AMPDU disabled), 1 (AMPDU enabled for transmit), 2 (AMPDU enabled for receive), and 3 (AMPDU enabled for transmit and receive). The 802.11n specification says a compliant station must receive AMPDU but may not support transmitting AMPDU frames. Disabling AMPDU receive is mainly useful for testing and working around bugs.

IEEE80211_IOC_AMPDU_DENSITY

Return the minimum density for bursting AMPDU frames in *i_val*. The value returned is one of: 0 (no time restriction), 1 (1/4 usec), 2 (1/2 usec), 3 (1 usec), 4 (2 usec), 5 (4 usec), 6 (8 usec), and 7 (16 usec).

IEEE80211_IOC_AMPDU_LIMIT

Return the limit on the size of AMPDU frames in *i_val*. The value returned is one of: 0 (8 kilobytes), 1 (16 kilobytes), 2 (32 kilobytes), and 3 (64 kilobytes).

IEEE80211_IOC_AMSDU

Return whether or not AMSDU is enabled in *i_val*. AMSDU is an aggregation scheme that is part of the 802.11n specification and is used only when operating on an HT channel. The value returned is one of: 0 (AMSDU disabled), 1 (AMSDU enabled for transmit), 2 (AMSDU enabled for receive), and 3 (AMSDU enabled for transmit and receive). The 802.11n specification says a compliant station must receive AMSDU but may not support transmitting AMSDU frames. Disabling AMSDU receive is mainly useful for testing and working around bugs.

IEEE80211_IOC_AMSDU_LIMIT

Return the limit on the size of AMSDU frames in *i_val*. The value returned is one of: 3839 (bytes) and 7935 (bytes). Note these values are specified by 802.11n; arbitrary values are not allowed.

IEEE80211_IOC_APBRIDGE

Return whether AP bridging is enabled in *i_val*. Normally packets sent between stations associated to the same access point are delivered without going through system layers that do packet filtering; when AP bridging is disabled packets are passed up the system to be forwarded using some other mechanism. This value will be non-zero when AP bridging is enabled and otherwise zero.

IEEE80211_IOC_APPIE

Return an application information element via *i_data*. Application IE's are maintained for many 802.11 frames; the request must identify the frame to return an IE for in *i_val*. For example, to retrieve the IE sent in each Beacon frame *i_val* would be set to `IEEE80211_FC0_SUBTYPE_BEACON | IEEE80211_FC0_TYPE_MGT`. If no information

element is installed then `EINVAL` is returned. If the data buffer for returning data is too small to hold the information element the value is truncated; this permits querying the presence of data by requesting zero bytes of data be returned.

IEEE80211_IOC_AUTHMODE

Return the current authentication mode in *i_val*. Valid values are `IEEE80211_AUTH_NONE` (no authentication), `IEEE80211_AUTH_OPEN` (open authentication), `IEEE80211_AUTH_SHARED` (shared key authentication), `IEEE80211_AUTH_8021X` (802.1x only authentication), and `IEEE80211_AUTH_WPA` (WPA/802.11i/802.1x authentication).

IEEE80211_IOC_BEACON_INTERVAL

Return the time between Beacon frames (in TU) in *i_val*.

IEEE80211_IOC_BGSCAN

Return whether background scanning is enabled in *i_val*. When this value is non-zero and operating in station mode the station will periodically leave the current channel and scan for neighboring stations. See also `IEEE80211_IOC_BGSCAN_IDLE` and `IEEE80211_IOC_BGSCAN_INTERVAL`.

IEEE80211_IOC_BGSCAN_IDLE

Return in *i_val* the minimum time (msecs) a station must be idle (i.e. not transmitting or receiving frames) before it will do a background scan. See also `IEEE80211_IOC_BGSCAN_INTERVAL`.

IEEE80211_IOC_BGSCAN_INTERVAL

Return in *i_val* the minimum time (seconds) between background scan operations. See also `IEEE80211_IOC_BGSCAN_IDLE`.

IEEE80211_IOC_BMISSTHRESHOLD

Return in *i_val* the number of consecutive missed Beacon frames before the system will attempt to roam to a different/better access point.

IEEE80211_IOC_BSSID

Return the MAC address for the current BSS identifier via *i_data*. When the interface is running, the bssid is either the value configured locally (e.g. for an IBSS network started by the local station) or the value adopted when joining an existing network. For WDS interfaces this value is the address of the remote station. When the interface is not running, the bssid returned is the desired bssid, if any, that has been configured.

IEEE80211_IOC_BURST

Return whether or not packet bursting is enabled in *i_val*. If this value is non-zero then the system will try to send packets closely spaced to improve throughput.

IEEE80211_IOC_CHANINFO

Return the set of available channels via *i_data*. Note this data should be used by user applications to map between channel specifications (frequency and attributes) and IEEE channel numbers as user applications may not have the necessary information to do this directly (e.g. for 900MHz radios, operation in the Public Safety Band).

IEEE80211_IOC_CHANLIST

Return the current list of usable channels via *i_data*. The channel list is returned as a bit vector with bit N set to 1 if IEEE channel number N is available for use.

IEEE80211_IOC_CHANNEL

Return the IEEE channel number of the current channel in *i_val*. Note this request is deprecated; use IEEE80211_IOC_CURCHAN instead.

IEEE80211_IOC_COUNTERMEASURES

Return whether TKIP Countermeasures are enabled in *i_val*. This value will be non-zero when Countermeasures are enabled and otherwise zero.

IEEE80211_IOC_CURCHAN

Return information for the current channel via *i_data*. This information includes the IEEE channel number, the frequency, and attributes that describe the operating constraints (e.g. Passive Scan, DFS, usage restrictions).

IEEE80211_IOC_DEVCAPS

Return device capabilities in the data buffer pointed at by *i_data*. The buffer must be large enough to return the number of available channels but otherwise may be made small to limit how much information is returned.

IEEE80211_IOC_DFS

Return whether or not Dynamic Frequency Selection (DFS) is enabled in *i_val*. DFS embodies several facilities including detection of overlapping radar signals, dynamic transmit power control, and channel selection according to a least-congested criteria. DFS support is mandatory for some 5GHz frequencies in certain locales (e.g. ETSI). By default DFS is enabled according to the regulatory definitions and the current country code, regdomain, and channel.

IEEE80211_IOC_DOTD

Return whether or not 802.11d support is enabled in *i_val*. When 802.11d is enabled in station

mode, Beacon frames that advertise a country code different than the currently configured country code will cause an event to be dispatched to user applications. This event can be used by the station to adopt that country code and operate according to the associated regulatory constraints. When operating as an access point with 802.11d enabled the Beacon and ProbeResponse frames transmitted will advertise the current regulatory domain settings.

IEEE80211_IOC_DOTH

Return whether 802.11h support is enabled in *i_val*. When 802.11h is enabled Beacon and ProbeResponse frames will have the SpectrumMgt bit set in the capabilities field and country and power constraint information elements will be present. 802.11h support also includes handling Channel Switch Announcements (CSA) which are a mechanism to coordinate channel changes by an access point. By default 802.11h is enabled if the device is capable.

IEEE80211_IOC_DROPUNENCRYPTED

Return, in *i_val*, whether unencrypted packets transmit/received should be discarded. This value will be zero if unencrypted packets will be accepted and non-zero if they are to be discarded.

IEEE80211_IOC_DTIM_PERIOD

Return the period (in beacon intervals) between DTIM events in *i_val*.

IEEE80211_IOC_DWDS

Return, in *i_val*, whether or not Dynamic WDS support is enabled. Dynamic WDS is a facility by which packets may be tunneled over normal Infrastructure BSS associations using 4-address (WDS) frames.

IEEE80211_IOC_FF

Return, in *i_val*, whether Atheros fast-frames support is enabled. Fast-frames is a non-standard protocol extension that aggregates multiple frames to improve throughput. Note that enabling fast-frames support does not guarantee use; the client and access point must negotiate its use.

IEEE80211_IOC_FRAGTHRESHOLD

Return, in *i_val*, the threshold (in bytes) for enabling fragmentation frames. Packets larger than this value will automatically be split into multiple fragmented frames that are sent one after the other.

IEEE80211_IOC_GREENFIELD

Return, in *i_val*, whether or not Greenfield preamble use is enabled. This setting is meaningful only when operating with 802.11n on an HT channel.

IEEE80211_IOC_HIDESSID

Return, in *i_val*, whether SSID hiding/cloaking is enabled. SSID hiding is only meaningful when operating as an access point. When this is enabled Beacon frames do not include the SSID and ProbeRequest frames are not answered unless they include the AP's SSID. This value will be non-zero when SSID hiding is enabled and otherwise zero.

IEEE80211_IOC_HTCOMPAT

Return, in *i_val*, whether or not 802.11n compatibility support is enabled. The 802.11n protocol specification went through several incompatible iterations. Some vendors implemented 11n support to older specifications that will not interoperate with a purely 11n-compliant station. In particular the information elements included in management frames for old devices are different. When compatibility support is enabled both standard and compatible data will be provided and/or accepted.

IEEE80211_IOC_HTCONF

Return the setting for automatic promotion of HT channels in *i_val*. Promotion happens when the system must select a channel and may choose between legacy, HT20, and HT40 operation (e.g. when scanning). Valid values are: 0 (do not promote, use legacy), 1 (promote to HT20), and 2 (promote to HT40).

IEEE80211_IOC_HTPROTMODE

Return, in *i_val*, the technique used to protect HT frames in a mixed 802.11n network. Valid values are: IEEE80211_PROTMODE_OFF (no protection enabled) and IEEE80211_PROTMODE_RTSCCTS (send RTS and wait for CTS).

IEEE80211_IOC_HWMP_MAXHOPS

Return the maximum acceptable hop count in an HWMP path in *i_val*.

IEEE80211_IOC_HWMP_ROOTMODE

Return the setting for Mesh root mode operation in *i_val*. Valid values are: IEEE80211_HWMP_ROOTMODE_DISABLED (root mode is disabled), IEEE80211_HWMP_ROOTMODE_NORMAL (send broadcast Path Request frames), IEEE80211_HWMP_ROOTMODE_PROACTIVE (send broadcast Path Request frames and force replies) and IEEE80211_HWMP_ROOTMODE_RANN (send broadcast Root Announcement (RANN) frames).

IEEE80211_IOC_IC_NAME

Return the underlying hardware device(9) name in the buffer pointed to by *i_data* and the name length including terminating NUL character in *i_len*. If the buffer length is too small to hold the full name EINVAL will be returned.

IEEE80211_IOC_INACTIVITY

Return whether or not the system handles inactivity processing in *i_val*. When inactivity processing is enabled the system will track stations that have not transmitted frames and periodically probe them to check if they are still present. Stations that are inactive and do not respond to probes are dropped.

IEEE80211_IOC_MACCMD

Return information about the state of the MAC address access control list (ACL) system. There are two requests supported: **IEEE80211_MACCMD_POLICY** (to retrieve the current policy in *i_val*), and **IEEE80211_MACCMD_LIST** to retrieve the list installed/active ACL's via *i_data*. The `wlan_acl(4)` module must be installed and enabled or `EINVAL` will be returned.

IEEE80211_IOC_MESH_AP

Return whether or not Mesh AP support is enabled in *i_val*.

IEEE80211_IOC_MESH_ID

Return the Mesh ID in the buffer pointed to by *i_data*.

IEEE80211_IOC_MESH_FWRD

Return whether or not packet forwarding support is enabled in *i_val*.

IEEE80211_IOC_MESH_PP_METRIC

Return the link metric protocol in the buffer pointed to by *i_data*.

IEEE80211_IOC_MESH_PP_PATH

Return the path selection protocol in the buffer pointed to by *i_data*.

IEEE80211_IOC_MESH_RTCMD

Return information about the state of the Mesh routing tables. One request is supported: **IEEE80211_MESH_RTCMD_LIST** to retrieve the contents of the routing table in the buffer pointed to by *i_data*.

IEEE80211_IOC_MESH_TTL

Return, in *i_val*, the Mesh Time To Live (TTL) setting installed in packets transmitted by this mesh node.

IEEE80211_IOC_NUMSSIDS

Return the number of SSIDs supported in *i_val*.

IEEE80211_IOC_NUMWEPKEYS

Return the number of WEP keys supported in *i_val*.

IEEE80211_IOC_POWERSAVE

Return the current powersaving mode in *i_val*. Valid values are IEEE80211_POWERSAVE_OFF (power save operation is disabled) and IEEE80211_POWERSAVE_ON (power save operation is enabled).

IEEE80211_IOC_POWERSAVESLEEP

Return the powersave sleep time in TU in *i_val*. This value is also termed the listen interval and represents the maximum time a station will sleep before waking to retrieve packets buffered by an access point.

IEEE80211_IOC_PRIVACY

Return the current MLME setting for PRIVACY in *i_val*. When PRIVACY is enabled all data packets must be encrypted. This value will be zero if PRIVACY is disabled and non-zero when PRIVACY is enabled.

IEEE80211_IOC_PROTMODE

Return the current 802.11g protection mode in *i_val*. Protection is the mechanism used to safeguard 802.11b stations operating on an 802.11g network. Valid values are IEEE80211_PROTMODE_OFF (no protection enabled), IEEE80211_PROTMODE_CTS (send CTS to yourself), and IEEE80211_PROTMODE_RTSCCTS (send RTS and wait for CTS).

IEEE80211_IOC_PUREG

Return whether “pure 11g” mode is enabled in *i_val*. This setting is meaningful only for access point operation; when non-zero, 802.11b stations will not be allowed to associate.

IEEE80211_IOC_PUREN

Return whether “pure 11n” mode is enabled in *i_val*. This setting is meaningful only for access point operation; when non-zero, legacy (non-11n capable) stations will not be allowed to associate.

IEEE80211_IOC_REGDOMAIN

Return the regulatory state in the buffer pointed to by *i_data*.

IEEE80211_IOC_RIFS

Return whether or not Reduced InterFrame Spacing (RIFS) is enabled in *i_val*. This setting is meaningful only when operating with 802.11n on an HT channel.

IEEE80211_IOC_ROAM

Return station roaming parameters in the buffer pointed to by *i_data*.

IEEE80211_IOC_ROAMING

Return the current roaming mode in *i_val*. Roaming mode specifies which entity controls operation of the MLME state machine when operating as a station in an Infrastructure BSS. Valid values are: IEEE80211_ROAMING_DEVICE (driver/firmware is in control), IEEE80211_ROAMING_AUTO (host 802.11 layer is in control), and IEEE80211_ROAMING_MANUAL (application is in control).

IEEE80211_IOC_RTSTHRESHOLD

Return the threshold (in bytes) for enabling transmission of RTS frames in *i_val*. Packets larger than this value will automatically have an RTS frame sent preceding it to reduce the likelihood of packet loss.

IEEE80211_IOC_SCAN_RESULTS

Return the current contents of the scan cache in the data area pointed to by *i_data*.

IEEE80211_IOC_SCANVALID

Return in *i_val* how long (in seconds) results from a scan operation will be considered valid. When scan results are no longer valid and they are needed (e.g. to roam) the system will initiate a scan operation to replenish the scan cache.

IEEE80211_IOC_SHORTGI

Return whether or not Short Guard Interval (SGI) is enabled in *i_val*. Note SGI is only used when operating with 802.11n on an HT channel.

IEEE80211_IOC_SMPS

Return the Spatial Multiplexing Power Save (SMPS) setting in *i_val*. This setting is meaningful only when operating with 802.11n on an HT channel. Valid values are: IEEE80211_HTCAP_SMPS_DYNAMIC (Dynamic SMPS is enabled), IEEE80211_HTCAP_SMPS_ENA (Static SMPS is enabled), and IEEE80211_HTCAP_SMPS_OFF (SMPS is disabled).

IEEE80211_IOC_SSID

Return the requested SSID in the buffer pointed to by *i_data*. If *i_val* is ≥ 0 then the request refers to the configured value for that slot. Generally, 0 is the only valid value, but some interfaces support more SSIDs.

IEEE80211_IOC_STA_INFO

Return information about the current state of the specified station(s) via *i_data*. The MAC

address of a single station may be passed in or, if the broadcast address is supplied, information about all stations will be returned. If a single station is requested and the MAC address is unknown then ENOENT will be returned.

IEEE80211_IOC_STA_STATS

Return collected statistics for the specified station via *i_data*. The MAC address of the desired station is passed in; if it is unknown ENOENT will be returned.

IEEE80211_IOC_STA_VLAN

Return any VLAN tag assigned to a station via *i_data*.

IEEE80211_IOC_TDMA_SLOT

Return the slot number for the station in *i_val*. Slot number zero is the master station in a TDMA network.

IEEE80211_IOC_TDMA_SLOTCNT

Return the count of slots in the TDMA network in *i_val*.

IEEE80211_IOC_TDMA_SLOTLEN

Return the length (in usecs) of the TDMA slot assigned to each station in the network in *i_val*.

IEEE80211_IOC_TDMA_BINTERVAL

Return the number of superframes between Beacon frames in *i_val*. A TDMA network with N slots and slot length T has a superframe of NxT.

IEEE80211_IOC_TSN

Return whether or not Transitional Security Network (TSN) is enabled in *i_val*.

IEEE80211_IOC_TURBOP

Return whether Atheros Dynamic Turbo mode is enabled in *i_val*. Dynamic Turbo mode is a non-standard protocol extension available only on Atheros devices where channel width is dynamically changed between 20MHz and 40MHz. Note that enabling Dynamic Turbo mode support does not guarantee use; both client and access point must use Atheros devices and support must be enabled on both sides.

IEEE80211_IOC_TXPARAMS

Return transmit parameters in the buffer pointed to by *i_data*.

IEEE80211_IOC_TXPOWER

Return the transmit power limit in .5 dBm units in *i_val*. This value represents the effective

maximum and is calculated according to the maximum power allowed by local regulations, any user-specified power limit, and the maximum power the device is capable of.

IEEE80211_IOC_TXPOWMAX

Return the user-specified maximum transmit power in .5 dBm units in *i_val*. The maximum setting is applied after any regulatory cap.

IEEE80211_IOC_WEP

Return the current WEP status in *i_val*. Valid values are: IEEE80211_WEP_ON (enabled for all packets sent and received), IEEE80211_WEP_OFF (disabled), and IEEE80211_WEP_MIXED (enabled for transmit and receive but also willing to receive unencrypted frames). This request is deprecated; use IEEE80211_IOC_PRIVACY and IEEE80211_IOC_UNENCRYPTED instead.

IEEE80211_IOC_WEPKEY

Return the requested WEP key via *i_data*. The key number is specified in *i_val* and may be 0-3. If the device does not support returning the WEP key or the user is not root then the key may be returned as all zeros. This request is deprecated in favor of IEEE80211_IOC_WPAKEY.

IEEE80211_IOC_WEPTXKEY

Return the number of the WEP key used for transmission in *i_val*.

IEEE80211_IOC_WME

Return whether 802.11e/WME/WMM support is enabled in *i_val*. This value will be non-zero when support is enabled and otherwise zero.

IEEE80211_IOC_WME_CWMIN

Return the WME CWmin setting (log2) for the specified Access Class (AC) in *i_val*. The AC is passed in through *i_len* together with an optional IEEE80211_WMEPARAM_BSS flag to indicate if the parameter for the BSS or the channel is desired. If WME is not supported then EINVAL will be returned.

IEEE80211_IOC_WME_CWMAX

Return the WME CWmax setting (log2) for the specified Access Class (AC) in *i_val*. See IEEE80211_IOC_WME_CWMIN above for more details.

IEEE80211_IOC_WME_AIFS

Return the WME AIFS setting for the specified Access Class (AC) in *i_val*. See IEEE80211_IOC_WME_CWMIN above for more details.

IEEE80211_IOC_WME_TXOPLIMIT

Return the WME TxOpLimit (msec) for the specified Access Class (AC) in *i_val*. See IEEE80211_IOC_WME_CWMIN above for more details.

IEEE80211_IOC_WME_ACM

Return the WME Admission Control Mechanism (ACM) setting for the specified Access Class (AC) in *i_val*. This value is meaningful only for the BSS (not channel). See IEEE80211_IOC_WME_CWMIN above for more details.

IEEE80211_IOC_WME_ACKPOLICY

Return the WME ACK Policy setting for the specified Access Class (AC) in *i_val*. When this value is zero frames will be transmitted without waiting for an Acknowledgement. This value is meaningful only for the channel (not BSS). See IEEE80211_IOC_WME_CWMIN above for more details.

IEEE80211_IOC_WPA

Return the WPA configuration in *i_val*. Valid values are 0 (WPA is not enabled), 1 (WPA1 is enabled), 2 (WPA2/802.11i is enabled), and 3 (WPA1 and WPA2/802.11i are both enabled).

IEEE80211_IOC_WPAIE

Return any WPA information element for an associated station via *i_data*. The request passed in through *i_data* identifies the MAC address of the desired station. If an RSN (802.11i) element is present it is returned; otherwise any WPA element is returned. Note this request is deprecated; use IEEE80211_IOC_WPAIE2 instead.

IEEE80211_IOC_WPAIE2

Return any WPA information elements for an associated station via *i_data*. The request passed in through *i_data* identifies the MAC address of the desired station. One or both of RSN (802.11i) and WPA elements may be returned.

IEEE80211_IOC_WPAKEY

Return the requested cryptographic key in the buffer pointed to by *i_data*. The key number is specified in *i_val* and may be 0-3. A key number of zero is used to retrieve a station's unicast cipher key when operating with WPA enabled. If the user is not root then the key data returned is all zeros.

IEEE80211_IOC_WPS

Return whether or not Wi-Fi Protected Setup (WPS) is enabled in *i_val*.

For SIOCS80211 the following values of *i_type* are valid. Note that changing a value on an interface that is running may cause the interface to be 'reset'. Resets may be handled without altering the state if

the parameter does not affect the MLME state (e.g. RTS threshold), but in some cases the interface may need to scan for a new network or clear state (including any associated stations); in that case the interface is said to be 'restarted' (it is equivalent to marking the interface down and back up). The information below identifies whether changing a value affects the state of a running interface.

IEEE80211_IOC_ADDMAC

Add an entry to the MAC address Access Control List (ACL) database using the value pointed to by *i_data*. The `wlan_acl(4)` module must be installed and enabled or `EINVAL` will be returned.

IEEE80211_IOC_AMPDU

Set whether or not AMPDU is enabled for transmit and/or receive using the value in *i_val*. This request causes a running interface operating on an HT channel to be reset. See `IEEE80211_IOC_AMPDU` above for details.

IEEE80211_IOC_AMPDU_DENSITY

Set the minimum density for bursting AMPDU frames to the value in *i_val*. This request causes a running interface to be reset. See `IEEE80211_IOC_AMPDU_DENSITY` above for details.

IEEE80211_IOC_AMPDU_LIMIT

Set the limit on the size of AMPDU frames to the value in *i_val*. This request causes a running interface to be reset. See `IEEE80211_IOC_AMPDU_LIMIT` above for details.

IEEE80211_IOC_AMSDU

Set whether or not AMSDU is enabled for transmit and/or receive using the value in *i_val*. This request causes a running interface operating on an HT channel to be reset. See `IEEE80211_IOC_AMSDU` above for details.

IEEE80211_IOC_AMSDU_LIMIT

Set the limit on the size of AMSDU frames to the value in *i_val*. This request causes a running interface to be reset. See `IEEE80211_IOC_AMSDU_LIMIT` above for details.

IEEE80211_IOC_APBRIDGE

Set whether AP bridging is enabled using the value in *i_val*. See `IEEE80211_IOC_APBRIDGE` above for details.

IEEE80211_IOC_APPIE

Set an application information element using the data pointed to by *i_data*. This request causes a running interface to be restarted if the WPA information element is changed. See `IEEE80211_IOC_APPIE` above for details.

IEEE80211_IOC_AUTHMODE

Set the current authentication mode using the value in *i_val*. This request causes a running interface to be restarted. See IEEE80211_IOC_AUTHMODE above for details. This request causes a running interface to be restarted.

IEEE80211_IOC_BEACON_INTERVAL

Set the time between Beacon frames (in TU) to the value in *i_val*. This request causes a running interface to be restarted.

IEEE80211_IOC_BGSCAN

Set whether background scanning is enabled using the value in *i_val*.

IEEE80211_IOC_BGSCAN_IDLE

Set the minimum time (in msec) a station must be idle before it will do a background scan to the value in *i_val*.

IEEE80211_IOC_BGSCAN_INTERVAL

Set the minimum time (seconds) between background scan operations to the value in *i_val*.

IEEE80211_IOC_BMISSTHRESHOLD

Set the number of consecutive missed Beacon frames before the system will attempt to roam to the value in *i_val*. This request causes a running interface to be reset.

IEEE80211_IOC_BSSID

Set the 802.11 MAC address for the desired BSS identifier according to *i_data*. This request causes a running interface to be restarted.

IEEE80211_IOC_BURST

Set whether or not packet bursting is enabled using the value in *i_val*. This request causes a running interface to be reset.

IEEE80211_IOC_CHANNEL

Set the desired/current channel to the value given by *i_val*. This request causes a running interface to immediately change to the specified channel if possible; otherwise the interface will be restarted. Note this request is deprecated; use IEEE80211_IOC_CURCHAN instead.

IEEE80211_IOC_CHANLIST

Set the list of available channels using the channel list pointed to by *i_data*. The channel list is a bit vector with bit N set to 1 if IEEE channel number N is available for use. The specified channel list is checked against the set of supported channels and any channels not supported are

silently ignored. If the intersection of the channel list and the supported channels is empty `EINVAL` is returned. Note the current channel may be marked invalid after installing a new channel list. This request causes a running interface to be restarted.

IEEE80211_IOC_COUNTERMEASURES

Set whether TKIP Countermeasures are enabled using the value in *i_val*. This request can only be used when the authentication mode is set WPA; otherwise `EOPNOTSUPP` will be returned.

IEEE80211_IOC_CURCHAN

Set the current channel using the information referenced by *i_data*. This request causes a running interface to immediately change to the specified channel if possible; otherwise the interface will be restarted.

IEEE80211_IOC_DELKEY

Delete the key specified by the information referenced by *i_data*.

IEEE80211_IOC_DELMAC

Remove an entry in the MAC address Access Control List (ACL) database using the value pointed to by *i_data*. The `wlan_acl(4)` module must be installed and enabled or `EINVAL` will be returned.

IEEE80211_IOC_DFS

Set whether or not Dynamic Frequency Selection (DFS) is enabled using the value in *i_val*. This request will fail with `EINVAL` if 802.11h support is not enabled. See `IEEE80211_IOC_DFS` above for details.

IEEE80211_IOC_DOTD

Set whether or not 802.11d support is enabled using the value in *i_val*. This request causes a running interface to be restarted. See `IEEE80211_IOC_DOTD` above for details.

IEEE80211_IOC_DOTH

Return whether 802.11h support is enabled using the value in *i_val*. See `IEEE80211_IOC_DOTH` above for details.

IEEE80211_IOC_DROPUNENCRYPTED

Set whether unencrypted packets transmit/received should be discarded using the value in *i_val*.

IEEE80211_IOC_DTIM_PERIOD

Set the period (in beacon intervals) between DTIM events to the value in *i_val*. This request causes a running interface to be restarted.

IEEE80211_IOC_DWDS

Set whether or not Dynamic WDS support is enabled using the value in *i_val*. See IEEE80211_IOC_DWDS above for details.

IEEE80211_IOC_FF

Set whether Atheros fast-frames support is enabled using the value in *i_val*. This request causes a running interface to be restarted. See IEEE80211_IOC_FF above for details.

IEEE80211_IOC_FRAGTHRESHOLD

Set the threshold (in bytes) for enabling fragmentation frames using the value in *i_val*. This request causes a running interface to be reset. See IEEE80211_IOC_FRAGTHRESHOLD above for details.

IEEE80211_IOC_GREENFIELD

Set whether or not Greenfield preamble use is enabled using the value in *i_val*. This request causes a running interface to be reset. See IEEE80211_IOC_GREENFIELD above for details.

IEEE80211_IOC_HIDESSID

Set whether SSID hiding/cloaking is enabled using the value in *i_val*. This request causes a running interface to be reset. See IEEE80211_IOC_HIDESSID above for details.

IEEE80211_IOC_HTCOMPAT

Set whether or not 802.11n compatibility support is enabled using the value in *i_val*. This request causes a running interface to be reset if operating on HT channel. See IEEE80211_IOC_HTCOMPAT above for details.

IEEE80211_IOC_HTCONF

Set automatic promotion of HT channels using the value in *i_val*. This request causes a running interface to be restarted. See IEEE80211_IOC_HTCONF above for details.

IEEE80211_IOC_HTPROTMODE

Set the technique used to protect HT frames in a mixed 802.11n network using the value in *i_val*. This request causes a running interface to be reset. See IEEE80211_IOC_HTPROTMODE above for details.

IEEE80211_IOC_HWMP_MAXHOPS

Set the maximum acceptable hop count in an HWMP path according to *i_val*. Values must be in the range [0-255].

IEEE80211_IOC_HWMP_ROOTMODE

Set the Mesh root mode operation according to *i_val*. Valid values are IEEE80211_HWMP_ROOTMODE_DISABLED (root mode is disabled), IEEE80211_HWMP_ROOTMODE_NORMAL (send broadcast Path Request frames), IEEE80211_HWMP_ROOTMODE_PROACTIVE (send broadcast Path Request frames and force replies) and IEEE80211_HWMP_ROOTMODE_RANN (send broadcast Root Announcement (RANN) frames).

IEEE80211_IOC_INACTIVITY

Set whether or not the system handles inactivity processing using the value in *i_val*. When inactivity processing is enabled the system will track stations that have not transmitted frames and periodically probe them to check if they are still present. Stations that are inactive and do not respond to probes are dropped.

IEEE80211_IOC_MACCMD

Change the state of the MAC address Access Control List (ACL) system. There are several requests supported: IEEE80211_MACCMD_POLICY_OPEN (set the current policy to disable ACL use), IEEE80211_MACCMD_POLICY_ALLOW (set the current policy to allow only addresses listed in the database), IEEE80211_MACCMD_POLICY_DENY (set the current policy to deny addresses listed in the database), IEEE80211_MACCMD_POLICY_RADIUS (set the current policy to enable use of a RADIUS backend), IEEE80211_MACCMD_FLUSH (flush all addresses from the database), and IEEE80211_MACCMD_DETACH (detach the ACL subsystem, disabling it). The wlan_acl(4) module must be installed or EINVAL will be returned.

IEEE80211_IOC_MESH_AP

Set whether or not Mesh AP support is enabled using *i_val*.

IEEE80211_IOC_MESH_FWRD

Set whether or not packet forwarding support is enabled using *i_val*.

IEEE80211_IOC_MESH_ID

Set the Mesh ID using the value pointed to by *i_data*. A Mesh ID can be up to IEEE80211_MESHID_LEN bytes long.

IEEE80211_IOC_MESH_PP_METRIC

Set the link metric protocol using the value pointed to by *i_data*.

IEEE80211_IOC_MESH_PP_PATH

Set the path selection protocol using the value pointed to by *i_data*.

IEEE80211_IOC_MESH_RTCMD

Manipulate the state of the Mesh routing tables. Several requests are supported: IEEE80211_MESH_RTCMD_FLUSH (flush the contents of the routing table), IEEE80211_MESH_RTCMD_ADD (add an entry for the MAC address specified in *i_data* and start the Peer discovery process), and IEEE80211_MESH_RTCMD_DELETE (delete the entry corresponding to the MAC address specified in *i_data*).

IEEE80211_IOC_MESH_TTL

Set the Mesh Time To Live (TTL) setting installed in packets transmitted by this mesh node using *i_val*.

IEEE80211_IOC_MLME

Explicitly control the MLME state machine for a station using the MLME request pointed to by *i_data*. There are several MLME operations supported: IEEE80211_MLME_ASSOC (request association to an access point), IEEE80211_MLME_DIASOC (diassociate the specified station), IEEE80211_MLME_DEAUTH (deauthenticate the specified station), IEEE80211_MLME_AUHorize (mark the specified station authorized to pass data frames), IEEE80211_MLME_UNAUTHORIZE (revoke the specified station's ability to pass data frames), and IEEE80211_MLME_AUTH (request authentication to an access point). Note when this facility is used for stations operating in infrastructure mode the roaming mode should be set to manual.

IEEE80211_IOC_POWERSAVE

Set the current powersaving mode to the value in *i_val*. See IEEE80211_IOC_POWERSAVE above for valid values. This request causes a running interface to be reset.

IEEE80211_IOC_POWERSAVESLEEP

Set the powersave sleep time in TU to the value in *i_val*. This request causes a running interface to be reset.

IEEE80211_IOC_PRIVACY

Set the current MLME setting for PRIVACY using the value in *i_val*. See IEEE80211_IOC_PRIVACY above for details.

IEEE80211_IOC_PROTMODE

Set the current 802.11g protection mode to the value in *i_val*. This request causes a running interface to be reset. See IEEE80211_IOC_PROTMODE above for details. This request causes a running interface to be reset.

IEEE80211_IOC_PUREG

Set whether "pure 11g" mode is enabled using the value in *i_val*. This request causes a running

interface to be restarted. See IEEE80211_IOC_PUREG above for details.

IEEE80211_IOC_PUREN

Set whether “pure 11n” mode is enabled using the value in *i_val*. This request causes a running interface to be restarted. See IEEE80211_IOC_PUREN above for details.

IEEE80211_IOC_REGDOMAIN

Set the regulatory state using the data referenced by *i_data*. This request can only be issued when all interfaces cloned from the underlying physical device are marked down; otherwise EBUSY is returned. Note the new regulatory data may invalidate any desired channel.

IEEE80211_IOC_RIFS

Set whether or not Reduced InterFrame Spacing (RIFS) is enabled using the value in *i_val*. This setting is meaningful only when operating with 802.11n on an HT channel. This request causes a running interface to be reset.

IEEE80211_IOC_ROAM

Set station roaming parameters using the data pointed to by *i_data*.

IEEE80211_IOC_ROAMING

Set the current roaming mode to the value in *i_val*. See IEEE80211_IOC_ROAMING above for details.

IEEE80211_IOC_RTSTHRESHOLD

Set the threshold (in bytes) for enabling transmission of RTS frames to the value in *i_val*. This request causes a running interface to be reset. See IEEE80211_IOC_RTSTHRESHOLD above for details.

IEEE80211_IOC_SCANVALID

Set the age (in seconds) that results from a scan operation will be considered valid. When scan results are no longer valid and they are needed (e.g. to roam) the system will initiate a scan operation to replenish the scan cache.

IEEE80211_IOC_SCAN_REQ

Request a scan operation using the parameters pointed to by *i_val*. The underlying device must be running or ENXIO will be returned. Values for *sr_duration*, *sr_mindwell*, and *sr_maxdwell* shorter than 1 clock tick are rounded up to a tick. If more SSID's are supplied than the system is capable of handling the extra ones are silently ignored. If a scan operation is already in progress the request will be (silently) ignored.

IEEE80211_IOC_SCAN_CANCEL

Cancel any pending/active scan operation.

IEEE80211_IOC_SHORTGI

Set whether or not Short Guard Interval (SGI) is enabled using the value in *i_val*. Note SGI is only used when operating on an HT (802.11n) channel. This request causes a running interface to be reset.

IEEE80211_IOC_SMPS

Set the Spatial Multiplexing Power Save (SMPS) setting to the value in *i_val*. This request causes a running interface to be reset. See IEEE80211_IOC_SMPS above for details.

IEEE80211_IOC_SSID

Set the desired SSID using the value pointed to by *i_data*. The string may be at most IEEE80211_NWID_LEN bytes. This request causes a running interface to be restarted.

IEEE80211_IOC_STA_STATS

Clear accumulated statistics for the specified station.

IEEE80211_IOC_STA_VLAN

Set the VLAN tag for the specified station using the information pointed to by *i_data*.

IEEE80211_IOC_TDMA_BINTERVAL

Set the interval between Beacon frames to the value in *i_val*. Values must be positive. This request causes a running interface to be reset.

IEEE80211_IOC_TDMA_SLOT

Set the current TDMA slot to the value in *i_val*. Values must be in the range [0-slotcnt]. Slot 0 identifies the master in the TDMA network; if it running it will immediately start sending Beacon frames.

IEEE80211_IOC_TDMA_SLOTCNT

Set the number of slots in the TDMA network to the value in *i_val*. This request causes a running interface to be reset.

IEEE80211_IOC_TDMA_SLOTLEN

Set the length of the TDMA slot assigned to each station in the network to the value in *i_val*. Slot lengths must be in the range 200 usecs to 1024 milliseconds (though values outside the range 1-200ms are unlikely to work well). This request causes a running interface to be reset.

IEEE80211_IOC_TSN

Set whether or not Transitional Security Network (TSN) is enabled using the value in *i_val*.

IEEE80211_IOC_TURBOP

Set whether Atheros Dynamic Turbo mode is enabled using the value in *i_val*. This request causes a running interface to be restarted.

IEEE80211_IOC_TXPARAMS

Set transmit parameters using the data pointed to be *i_data*. This request causes a running interface to be restarted.

IEEE80211_IOC_TXPOWER

Set the maximum transmit power limit in .5 dBm units to the value in *i_val*. This request causes a running interface to be reset.

IEEE80211_IOC_WEP

Set the current WEP mode to the value in *i_val*. See IEEE80211_IOC_WEP above for valid values. This request causes a running interface to be restarted. Note this request is deprecated; use IEEE80211_IOC_PRIVACY and IEEE80211_IOC_DROPUNENCRYPTED instead.

IEEE80211_IOC_WEPKEY

Set the WEP key indicated by *i_val* using the data pointed to by *i_data*. Note this request is deprecated; use IEEE80211_IOC_WPAKEY instead.

IEEE80211_IOC_WEPTXKEY

Set the default transmit key used for transmission to the value in *i_val*.

IEEE80211_IOC_WME

Set whether or not WME/WMM support is enabled using the value in *i_val*. This request causes a running interface to be reset.

IEEE80211_IOC_WME_ACKPOLICY

Set the WME ACK Policy for the Access Class (AC) specified in *i_len* using the value in *i_val*.

IEEE80211_IOC_WME_ACM

Set the WME Admission Control Mechanism for the Access Class (AC) specified in *i_len* using the value in *i_val*.

IEEE80211_IOC_WME_AIFS

Set the WME AIFS parameter for the Access Class (AC) specified in *i_len* using the value in

i_val.

IEEE80211_IOC_WME_CWMAX

Set the WME CWmax parameter for the Access Class (AC) specified in *i_len* using the value in *i_val*.

IEEE80211_IOC_WME_CWMIN

Set the WME CWmin parameter for the Access Class (AC) specified in *i_len* using the value in *i_val*.

IEEE80211_IOC_WME_TXOPLIMIT

Set the WME TxOpLimit parameter for the Access Class (AC) specified in *i_len* using the value in *i_val*.

IEEE80211_IOC_WPA

Set the WPA configuration using the value in *i_val*. This request causes a running interface to be reset. See IEEE80211_IOC_WPA above for details.

IEEE80211_IOC_WPAKEY

Set the requested cryptographic key using data in the buffer pointed to by *i_data*. See IEEE80211_IOC_WPAKEY for details.

IEEE80211_IOC_WPS

Set whether or not Wi-Fi Protected Setup (WPS) is enabled using the value in *i_val*.

SEE ALSO

ioctl(2), wlan(4), wlan_acl(4), wlan_xauth(4), hostapd(8), ifconfig(8), wpa_supplicant(8)