

brctl

Key	Description	Value Type	Default
brctl	EMI brctl configuration module root key.	N/A	N/A
brctl.*	Bridge interface name. Specify bridge interface name in the key wildcard.	N/A	N/A
brctl.*.brage	Bridge ageing time. The ageing time is the number of seconds a MAC address will be kept in the forwarding database (fdb) after receiving a packet from this MAC address. The entries in the forwarding database are periodically timed out to ensure that old ones do not persist in the database.	frac[10,10000,1]	300
brctl.*.brfd	Bridge forwarding delay time. Forwarding delay time is used by STP, RSTP and MSTP and is the time spent in each of the listening and learning states before the forwarding state is entered. This delay is so that when a new bridge comes onto a busy network it looks at some traffic before participating.	frac[4,30,1]	15
brctl.*.brhello	Bridge hello time. Set the interval between periodic transmissions of Configuration Messages by Designated Ports in seconds. A hello packet is sent out by the Root Bridge and the Designated Bridges. Hello packets are used to communicate information about the topology throughout the entire Bridged Local Area Network. This value is fixed to 2.0 in RSTP (Rapid Spanning Tree Protocol). Compatibility range 1.0-2.0.	frac[1,2,1]	2
brctl.*.brmaxage	Bridge maximum message age. Set the maximum age of the information transmitted by the Bridge when it is the Root Bridge in seconds.	frac[6,40,1]	20
brctl.*.brprio	Bridge priority. Set Bridge priority in the range of [0-65535]. The bridge with the lowest priority will be elected 'root bridge' in STP protocol.	int[0,65535]	32768
brctl.*.enabled	Bridge enabled. Specify if the bridge is enabled or disabled.	boolean	FALSE
brctl.*.if	Bridge ports configuration.	N/A	N/A
brctl.*.if.*	Bridge port name. Specify bridge port name in the key wildcard. The bridge port name corresponds to a physical port of the bridge e.g. eth0.	N/A	N/A
brctl.*.if.*.enabled	Bridge port enabled. Specify if the bridge port is enabled or disabled.	boolean	FALSE
brctl.*.if.*.pathcost	Bridge port path cost. Bridge port path cost is recommended to be set according to the link speed of the port. See Table 17-3, IEEE802.1D-2004. STP will try to avoid making connection using expensive path cost. It is however, for some paths whose speed is determined by auto-negotiation on the fly, this parameter will not be able to reflect the actual 'cost' of the path.	int[0,65535]	100
brctl.*.if.*.portprio	Bridge port priority. Specify the bridge port priority in the range of [0-255]. This metric is used in the designated port and root port selection algorithms. For multiple ports with the same cost there is also a priority	int[0,255]	
brctl.*.stp	Spanning tree protocol enabled. Specify if spanning tree protocol is enabled or disabled.	boolean	FALSE

dropbear

Key	Description	Value Type	Default
dropbear	EMI dropbear configuration module root key.	N/A	N/A
dropbear.enabled	SSH server enabled. Specify if the SSH server is enabled or disabled.	boolean	FALSE
dropbear.host_key_dss	DSS host key. Specify the DSS host key with base64 encoding. If the dropbear.enabled is set to true and DSS host key is not specified in here, or if the DSS host key is invalid, the DSS host key will be autogenerated.	ascii_text	not set
dropbear.host_key_rsa	RSA host key. Specify the RSA host key with base64 encoding. If the dropbear.enabled is set to true and RSA host key is not specified in here, or if the RSA host key is invalid, the RSA host key will be autogenerated.	ascii_text	not set
dropbear.port	SSH server port. Specify the TCP port which dropbear will listen for incoming SSH connections.	int[1,65535]	22

general

Key	Description	Value Type	Default
general	EMI general configuration module root key.	N/A	N/A
general.contact	Contact information	N/A	N/A
general.contact.city	City	ascii_text	not set
general.contact.country_code	Country code as ISO 3166-1-alpha-2 code	char[2]	not set
general.contact.email	E-mail address	ascii_text	not set
general.contact.name	Name	ascii_text	not set
general.contact.phone	Phone number	ascii_text	not set
general.contact.post_code	Postal code	ascii_text	not set
general.contact.state	State	ascii_text	not set

general.contact.street_address	Stree address	ascii_text	not set
general.contact.www	WWW address	ascii_text	not set
general.location	Location information of the device	N/A	N/A
general.location.city	City	ascii_text	not set
general.location.country_code	Country code as ISO 3166-1-alpha-2 code	char[2]	not set
general.location.latitude	Latitude in decimal degrees. North is positive and South is negative.	frac[-90,90,8]	not set
general.location.longitude	Longitude in decimal degrees. East is positive and West is negative.	frac[-180,180,8]	not set
general.location.name	Contact name	ascii_text	not set
general.location.post_code	Postal code	ascii_text	not set
general.location.state	State	ascii_text	not set
general.location.street_address	Stree address	ascii_text	not set

hostapd

Key	Description	Value Type	Default
hostapd	EMI hostapd configuration module root key.	N/A	N/A
hostapd.*	Hostapd instance number. Specify the hostapd instance.	N/A	N/A
hostapd.*.bridge	Notifies hostapd if the interface is included in a bridge [bridge name e.g. br0]. Default: not set.	if[any]	not set
hostapd.*.driver	Driver name. Specify the name of the wireless driver is use for the wireless interface specified in this instance.	enum[hostap wired madwifi prism54 test none nl80211 bsd]	hostap
hostapd.*.enabled	Hostapd enabled. Specify if this instance of hostapd is enabled or disabled.	boolean	FALSE
hostapd.*.ieee8021x	IEEE 802.1X	N/A	N/A
hostapd.*.ieee8021x.acct_server_addr	RADIUS accounting server IP address.	ip	not set
hostapd.*.ieee8021x.acct_server_port	RADIUS accounting server port number.	int[1,65535]	not set
hostapd.*.ieee8021x.acct_server_shared_secret	RADIUS accounting server shared secret [non-empty string].	ascii_text	not set
hostapd.*.ieee8021x.auth_server_addr	RADIUS authentication server IP address. Default: not set.	ip	not set
hostapd.*.ieee8021x.auth_server_port	RADIUS authentication server port number.	int[1,65535]	not set
hostapd.*.ieee8021x.auth_server_shared_secret	RADIUS authentication server shared secret [non-empty string].	ascii_text	not set
hostapd.*.ieee8021x.check_crl	Enable certificate revocation list (CRL) verification. A certificate revocation list (CRL) is a list of certificates (or more specifically, a list of serial numbers for certificates) that have been revoked or are no longer valid, and therefore should not be relied upon. CRL typically applies to public key infrastructure. Options: * none - do not verify CRLs, * user - check the CRL of the user certificate, * all - check all CRLs in the certificate path	enum[none user all]	0
hostapd.*.ieee8021x.dynamic_vlan	Dynamic VLAN mode; allow RADIUS authentication server to decide which VLAN is used for the stations. Dynamic VLAN assignment places a wireless user into a specific VLAN based on the credentials supplied by the user. This task of assigning users to a specific VLAN is handled by a RADIUS authentication server. This can be used, for example, to allow the wireless host to remain on the same VLAN as it moves within a campus network. Options: * none - disabled (default), * optional - use default interface if RADIUS server does not include VLAN ID, * required - reject authentication if RADIUS server does not include VLAN ID.	enum[none optional required]	none
hostapd.*.ieee8021x.eap_fast_a_id	EAP-FAST authority identity (A-ID). The A-ID should be unique across all issuing servers (32 hex digits). Default: not set.The authority identity (A-ID) is used to provide the peer a hint of the server's identity that may be useful in helping the peer to select the appropriate credential to use. For details, please refer to RFC4851.	hex[32]	not set
hostapd.*.ieee8021x.eap_fast_a_id_info	EAP-FAST authority identifier information (A-ID-Info), a user-friendly name for the A-ID [string]. Default: not set.	ascii_text	not set
hostapd.*.ieee8021x.eap_fast_prov	Enable/disable different EAP-FAST provisioning modes. * none - provisioning disabled, * anonymous - only anonymous provisioning allowed, * authenticated - only authenticated provisioning allowed, * both - both provisioning modes allowed.	enum[none anonymous authenticated both]	both
hostapd.*.ieee8021x.eap_message	Optional displayable message sent with EAP Request-Identity. A '\0' can be used to separate network info (see RFC 4284). e.g.: hello\0networkid=netw,nasid=foo,portid=0,NAIRealms=example.com	ascii_text	not set
hostapd.*.ieee8021x.eap_reauth_period	EAP reauthentication period in seconds, 0 = disable reauthentication.	int[0,65535]	3600
hostapd.*.ieee8021x.eap_server	Use integrated EAP server instead of external RADIUS authentication server.	boolean	FALSE

hostapd.*.ieee8021x.eap_sim_aka_result_ind	EAP-SIM (Subscriber Identity Module) and EAP-AKA (UMTS Authentication and Key Agreement) protected success/failure indication using AT_RESULT_IND. Please refer to RFC4186 and RFC4187 for the detailed explanation of AT_RESULT_IND.	boolean	FALSE
hostapd.*.ieee8021x.eap_sim_db	Configuration data for EAP-SIM database/authentication gateway interface. Default: not set. e.g. unix:/tmp/hlr_auc_gw.sock	ascii_text	not set
hostapd.*.ieee8021x.eapol_key_index_workaround	EAPOL-Key index workaround (set bit7) for WinXP Supplicant (needed only if only broadcast keys are used).	boolean	FALSE
hostapd.*.ieee8021x.eapol_version	IEEE 802.1X/EAPOL version (version 1 can be set for interoperability reasons). Default 2.	int[1,2]	2
hostapd.*.ieee8021x.ieee8021x	Require IEEE 802.1X authorization.	boolean	FALSE
hostapd.*.ieee8021x.nas_identifier	Optional NAS-identifier string for RADIUS messages, this should be unique to the NAS within the scope of the RADIUS server e.g. a fully qualified domain name. When using IEEE 802.11r it must be set and must be between 1 and 48 octets long.	ascii_text	not set
hostapd.*.ieee8021x.own_ip_addr	The own IP address of the access point (used as NAS-IP-Address).	ip	not set
hostapd.*.ieee8021x.pac_key_lifetime	EAP-FAST PAC-Key lifetime in seconds (hard limit). Default 604800 (one week). PAC stands for Protected Access Credential, it is used for establishing TLS tunnel between the client and the RADIUS server.	int[1,2147483647]	604800
hostapd.*.ieee8021x.pac_key_refresh_time	EAP-FAST PAC-Key refresh time in seconds (soft limit on remaining hard limit). A new PAC-Key will be generated when this number of seconds (or fewer) of the lifetime remains. Default 86400 (one day).	int[1,2147483647]	86400
hostapd.*.ieee8021x.pac_opaque_encr_key	Encryption key for EAP-FAST PAC-Opaque values. This key must be a secret, random value (32 hex digits).	hex[32]	not set
hostapd.*.ieee8021x.private_key_passwd	Passphrase for private key for integrated EAP server.	ascii_text	not set
hostapd.*.ieee8021x.radius_acct_interim_interval	Interim accounting update interval (in seconds) Valid values 60-86400, 0 means not set.	int[60,86400]	0
hostapd.*.ieee8021x.radius_retry_primary_interval	Retry interval for trying to return to the primary RADIUS server (in seconds).	int[1,2147483647]	not set
hostapd.*.ieee8021x.radius_server_auth_port	The UDP port number for the RADIUS authentication server [1-65535].	int[1,65535]	1812
hostapd.*.ieee8021x.radius_server_clients	RADIUS clients configuration for the RADIUS server. If this key is missing RADIUS server is disabled.	N/A	N/A
hostapd.*.ieee8021x.radius_server_clients.*	Line entry. Keyname is line # (0-65535)	N/A	N/A
hostapd.*.ieee8021x.radius_server_clients.*.data	[Network_address(CIDR) secret_passphrase] pair	ascii_text	not set
hostapd.*.ieee8021x.radius_server_ipv6	Use IPv6 with RADIUS server (IPv4 will also be supported using IPv6 API).	boolean	FALSE
hostapd.*.ieee8021x.tnc	Enable Trusted Network Connect (TNC), if enabled, TNC validation will be required before the peer is allowed to connect (used only with EAP-TLS and EAP-FAST, for other EAP methods the peer will be allowed to connect without TNC).	boolean	FALSE
hostapd.*.ieee8021x.use_pae_group_addr	Use PAE group address (01:80:c2:00:00:03) instead of individual target address when sending EAPOL frames with driver=wired.	boolean	FALSE
hostapd.*.ieee8021x.vlan_file	VLAN interface list for dynamic VLAN mode. Maps VLAN ID from the RADIUS server to a network interface.	N/A	N/A
hostapd.*.ieee8021x.vlan_file.*	Line entry. Keyname is line # (0-65535)	N/A	N/A
hostapd.*.ieee8021x.vlan_file.*.data	[VLAN_ID interface_name] pair.	ascii_text	not set
hostapd.*.ieee8021x.vlan_tagged_interface	Interface where 802.1q tagged packets should appear when a RADIUS server is used to determine which VLAN a station is on.	if[any]	not set
hostapd.*.ieee8021x.wep_key_len_broadcast	WEP key length for default/broadcast keys. * 0 = disable rekeying, * 5 = 40-bit WEP (also known as 64-bit WEP with 40 secret bits), * 13 = 104-bit WEP (also known as 128-bit WEP with 104 secret bits)	enum[0 5 13]	0
hostapd.*.ieee8021x.wep_key_len_unicast	WEP key length for individual/unicast keys. * 0 = disable rekeying, * 5 = 40-bit WEP (also known as 64-bit WEP with 40 secret bits), * 13 = 104-bit WEP (also known as 128-bit WEP with 104 secret bits).	enum[0 5 13]	0
hostapd.*.ieee8021x.wep_rekey_period	WEP rekeying period in seconds. 0 = do not rekey.	int[0,65535]	300
hostapd.*.interface	Wireless interface name. Specify the physical wireless interface name that will be managed by this hostapd instance.	if[any]	not set
hostapd.*.logging	Logging.	N/A	N/A
hostapd.*.logging.logger_syslog	Event logger to syslog. Module bitfield. Default -1. * bit 0 (1) = IEEE 802.11 * bit 1 (2) = IEEE 802.1X * bit 2 (4) = RADIUS * bit 3 (8) = WPA * bit 4 (16) = driver interface * bit 5 (32) = IAPP * bit 6 (64) = MLME * all bits (-1) = all	int[-1,255]	-1

hostapd.*.logging.logger_syslog_level	Event logger to syslog. Logging level [0-4]. * 0 = verbose debugging * 1 = debugging * 2 = informational messages * 3 = notification * 4 = warning	int[0,4]	2
hostapd.*.wlan	WLAN - IEEE 802.11	N/A	N/A
hostapd.*.wlan.accept_mac_file	MAC accept list. Operate in conjunction with hostapd.*.wlan.macaddr_acl.	N/A	N/A
hostapd.*.wlan.accept_mac_file.*	Line entry. Keyname is line # (0-65535)	N/A	N/A
hostapd.*.wlan.accept_mac_file.*.data	MAC address and optional VLAN ID if dynamic_vlan is enabled.	ascii_text	not set
hostapd.*.wlan.ap_max_inactivity	Station inactivity timeout in seconds (if a station does not respond to an empty frame sent after this timeout, it will be disassociated and deauthenticated).	int[0,65535]	300
hostapd.*.wlan.ap_table_expiration_time	Number of seconds of no frames received after which entries may be deleted from the AP table.	int[0,2147483647]	60
hostapd.*.wlan.ap_table_max_size	Maximum number of entries kept in AP table.	int[0,65535]	255
hostapd.*.wlan.auth_algs	Authentication algorithms. * none - No authentication algorithms enabled * open - Open System authentication enabled * shared - Shared Key authentication enabled * both - Open System and Shared Key Authentication enabled	enum[none open shared both]	none
hostapd.*.wlan.basic_rates	Rates that are included in the basic rate set.	N/A	N/A
hostapd.*.wlan.basic_rates.10	1 Mbps rate enabled.	boolean	FALSE
hostapd.*.wlan.basic_rates.110	11 Mbps rate enabled.	boolean	FALSE
hostapd.*.wlan.basic_rates.120	12 Mbps rate enabled.	boolean	FALSE
hostapd.*.wlan.basic_rates.180	18 Mbps rate enabled.	boolean	FALSE
hostapd.*.wlan.basic_rates.20	2 Mbps rate enabled.	boolean	FALSE
hostapd.*.wlan.basic_rates.240	24 Mbps rate enabled.	boolean	FALSE
hostapd.*.wlan.basic_rates.360	36 Mbps rate enabled.	boolean	FALSE
hostapd.*.wlan.basic_rates.480	48 Mbps rate enabled.	boolean	FALSE
hostapd.*.wlan.basic_rates.540	54 Mbps rate enabled.	boolean	FALSE
hostapd.*.wlan.basic_rates.55	5.5 Mbps rate enabled.	boolean	FALSE
hostapd.*.wlan.basic_rates.60	6 Mbps rate enabled.	boolean	FALSE
hostapd.*.wlan.basic_rates.90	9 Mbps rate enabled.	boolean	FALSE
hostapd.*.wlan.beacon_int	Beacon interval in kus (1.024 ms). High beacon interval may help to save energy for the wireless device. Low beacon interval can shorten the time for connecting the wireless device to the network.	int[15,65535]	100
hostapd.*.wlan.bridge_packets	Enable/disable internal bridge for packets between associated stations. Default: do not control from hostapd	boolean	FALSE
hostapd.*.wlan.ca_cert	CA (Certificate Authority) certificate for EAP-TLS/PEAP/TTLS in PEM or DER format.	ascii_text	
hostapd.*.wlan.channel	Channel number. 0 = not set. Does not work with some drivers (e.g. madwifi). Sets the channel for hostapd to operate on. Must be a channel supported by the mode set in hw_mode, as well as allowed by your countries Wireless Regulatory rules. Channel should be chosen so that it has the minimum overlap with other APs or other networks in your area. 802.11 channels are 20mhz (4 channels) wide in total, or 10mhz (2 channels) wide on each side. This means that an access point on channel 3 will interfere with an access point on channel 1 or channel 5. Use this to pick a channel. Most consumer APs default to channel 6, so you can use channel 1 or channel 11 in most cases for the best results. Also note that the channels available to you depend heavily entirely on the local regulatory rules.	int[0,200]	0
hostapd.*.wlan.country_code	Country code (ISO/IEC 3166-1). Used to set regulatory domain. [2 letter code].	ascii_char[2]	US
hostapd.*.wlan.deny_mac_file	MAC deny list. Operate in conjunction with hostapd.*.wlan.macaddr_acl.	N/A	N/A
hostapd.*.wlan.deny_mac_file.*	Line entry. Keyname is line # (0-65535)	N/A	N/A
hostapd.*.wlan.deny_mac_file.*.data	MAC address.	mac	not set
hostapd.*.wlan.dh_file	DH/DSA parameters file (in PEM format). This is an optional configuration file for setting parameters for an ephemeral DH key exchange and is required if anonymous EAP-FAST is used	ascii_text	
hostapd.*.wlan.dtim_period	DTIM (delivery traffic information message) period: number of beacons between DTIMs.	int[1,255]	2
hostapd.*.wlan.eap_user_file	EAP server user database	N/A	N/A
hostapd.*.wlan.eap_user_file.*	Line entry. Keyname is line # (0-65535)	N/A	N/A
hostapd.*.wlan.eap_user_file.*.data	Database entry for integrated EAP server.	ascii_text	not set
hostapd.*.wlan.fragm_threshold	Fragmentation threshold (2346 = disabled).	int[256,2346]	2346
hostapd.*.wlan.ft	Fast BSS Transition (FT) - IEEE 802.11r-2008.	N/A	N/A
hostapd.*.wlan.ft.mobility_domain	Mobility Domain identifier (MDID). It is used to indicate a group of APs (within an ESS, i.e., sharing the same SSID) between which a STA can use Fast BSS Transition	hex[4]	not set
hostapd.*.wlan.ft.pmk_r1_push	Enable PMK-R1 push at R0KH.	boolean	FALSE
hostapd.*.wlan.ft.r0_key_lifetime	Default lifetime of the PMK-RO in minutes.	int[1,65535]	0

hostapd.*.wlan.ft.r0kh	List of R0KHs in the same Mobility Domain, format: <MAC address> <NAS Identifier> <128-bit key as hex string>. e.g. 02:01:02:03:04:05 r0kh-1.example.com 000102030405060708090a0b0c0d0e0f	ascii_text	not set
hostapd.*.wlan.ft.r1_key_holder	PMK-R1 Key H.older identifier (12 hex digits)	hex[12]	not set
hostapd.*.wlan.ft.r1kh	List of R1KHs in the same Mobility Domain, format: <MAC address> <R0KH-ID> <128-bit key as hex string>. e.g. 02:01:02:03:04:05 02:11:22:33:44:55 000102030405060708090a0b0c0d0e0f	ascii_text	not set
hostapd.*.wlan.ft.reassociation_deadline	Reassociation deadline in time units (TUs = 1.024 ms).	int[1000,65535]	not set
hostapd.*.wlan.hw_mode	IEEE 802.11 operation mode [a,b,g].	enum[a b g]	b
hostapd.*.wlan.iapp_interface	Interface to be used for IAPP broadcast packets	if[any]	not set
hostapd.*.wlan.ieee80211d	Enable IEEE 802.11d. This advertises the country_code and the set of allowed channels and transmit power levels based on the regulatory limits.	boolean	FALSE
hostapd.*.wlan.ieee80211n	Enable IEEE 802.11n (HT).	boolean	FALSE
hostapd.*.wlan.ignore_broadcast_ssid	Send empty SSID in beacons and ignore probe request frames that do not specify full SSID. * none - disabled, * empty - send empty (length=0) SSID in beacon, * clear - clear SSID (ASCII 0), but keep the original length	enum[none empty clear]	none
hostapd.*.wlan.macaddr_acl	Access Control List. Station MAC address -based authentication * accept - accept unless in deny list, * deny - deny unless in accept list, * radius - use external RADIUS server (accept/deny lists are searched first).	enum[access deny radius]	accept
hostapd.*.wlan.max_listen_interval	Maximum allowed Listen Interval (how many Beacon periods STAs are allowed to remain asleep).	int[0,65535]	65535
hostapd.*.wlan.max_num_sta	Maximum number of stations allowed in station table.	int[0,2007]	2007
hostapd.*.wlan.passive_scan_interval	Scan different channels every N seconds, 0 = disabled.	int[0,2147483647]	0
hostapd.*.wlan.passive_scan_listen	Listen N micro seconds on each channel when doing passive scanning, 0 disabled.	int[0,2147483647]	0
hostapd.*.wlan.passive_scan_mode	Passive scanning mode. * all - scan all supported modes (802.11a/b/g/Turbo), * current - scan only the mode that is currently used for normal operations	enum[all current]	all
hostapd.*.wlan.preamble	Use short preamble for frames sent at 2 Mbps, 5.5 Mbps, and 11 Mbps to improve network performance.	boolean	FALSE
hostapd.*.wlan.private_key	Private key matching with the server certificate for EAP-TLS/PEAP/TTLS [PEM or DER format]	ascii_text	
hostapd.*.wlan.rts_threshold	RTS/CTS threshold (2347 = disabled).	int[0,2347]	2347
hostapd.*.wlan.server_cert	Server certificate for EAP-TLS/PEAP/TTLS [PEM or DER format].	ascii_text	not set
hostapd.*.wlan.ssid	SSID (Service Set Identifier) to be used in IEEE 802.11 management frames. The SSID can be up to 32 characters long. As the SSID is a name that may be displayed to users, it normally consists of displayable ASCII characters. It is what show up the scan result. Some vendors refer SSID as network name.	ascii_text	not set
hostapd.*.wlan.supported_rates	Restrict supported rates.	N/A	N/A
hostapd.*.wlan.supported_rates.10	1 Mbps rate enabled.	boolean	FALSE
hostapd.*.wlan.supported_rates.110	11 Mbps rate enabled.	boolean	FALSE
hostapd.*.wlan.supported_rates.120	12 Mbps rate enabled.	boolean	FALSE
hostapd.*.wlan.supported_rates.180	18 Mbps rate enabled.	boolean	FALSE
hostapd.*.wlan.supported_rates.20	2 Mbps rate enabled.	boolean	FALSE
hostapd.*.wlan.supported_rates.240	24 Mbps rate enabled.	boolean	FALSE
hostapd.*.wlan.supported_rates.360	36 Mbps rate enabled.	boolean	FALSE
hostapd.*.wlan.supported_rates.480	48 Mbps rate enabled.	boolean	FALSE
hostapd.*.wlan.supported_rates.540	54 Mbps rate enabled.	boolean	FALSE
hostapd.*.wlan.supported_rates.55	5.5 Mbps rate enabled.	boolean	FALSE
hostapd.*.wlan.supported_rates.60	6 Mbps rate enabled.	boolean	FALSE
hostapd.*.wlan.supported_rates.90	9 Mbps rate enabled.	boolean	FALSE
hostapd.*.wlan.wep	WEP Configuration	N/A	N/A
hostapd.*.wlan.wep.wep_default_key	The WEP key number to use when transmitting [0-3]. Default: not set.	int[0,3]	not set
hostapd.*.wlan.wep.wep_key0	WEP key 0. For 128-bit WEP, it is 26 hexadecimal characters (0-9, A-F). For 64-bit WEP, it is 10 hexadecimal characters.	ascii_text	not set
hostapd.*.wlan.wep.wep_key1	WEP key 1. For 128-bit WEP, it is 26 hexadecimal characters (0-9, A-F). For 64-bit WEP, it is 10 hexadecimal characters.	ascii_text	not set
hostapd.*.wlan.wep.wep_key2	WEP key 2. For 128-bit WEP, it is 26 hexadecimal characters (0-9, A-F). For 64-bit WEP, it is 10 hexadecimal characters.	ascii_text	not set
hostapd.*.wlan.wep.wep_key3	WEP key 3. For 128-bit WEP, it is 26 hexadecimal characters (0-9, A-F). For 64-bit WEP, it is 10 hexadecimal characters.	ascii_text	not set
hostapd.*.wlan.wme	WME Configuration. Wireless Multimedia Extensions (WME) is also known as Wi-Fi Multimedia (WMM). It helps to provide QoS (Quality of Service) to 802.11 networks by prioritizing traffic according to access categories (AC): Voice (VO), Video (Vi), Best Effort (BE), Background (BG).	N/A	N/A

hostapd.*.wlan.wme.tx_queue_after_beacon_aifs	TX queue After Beacon AIFS. 802.11a/b/g default: 2. Range 0-255.	ascii_text	hw_default
hostapd.*.wlan.wme.tx_queue_after_beacon_burst	TX queue After Beacon Burst. Maximum length for bursting in ms with up to 0.1 ms precision. 802.11a/b/g default: 0.	ascii_text	hw_default
hostapd.*.wlan.wme.tx_queue_after_beacon_cwmax	TX queue After Beacon CWmax. 802.11a/b/g default: 1023.	enum[1 3 7 15 31 63 127 255 511 1023 hw_default]	hw_default
hostapd.*.wlan.wme.tx_queue_after_beacon_cwmin	TX queue After Beacon CWmin. 802.11a/b/g default: 15.	enum[1 3 7 15 31 63 127 255 511 1023 hw_default]	hw_default
hostapd.*.wlan.wme.tx_queue_beacon_aifs	TX queue Beacon AIFS. 802.11a/b/g default: 2. Range 0-255.	ascii_text	hw_default
hostapd.*.wlan.wme.tx_queue_beacon_burst	TX queue Beacon Burst. Maximum length for bursting in ms with up to 0.1 ms precision. 802.11a/b/g default: 1.5.	ascii_text	hw_default
hostapd.*.wlan.wme.tx_queue_beacon_cwmax	TX queue Beacon CWmax. 802.11a/b/g default: 7.	enum[1 3 7 15 31 63 127 255 511 1023 hw_default]	hw_default
hostapd.*.wlan.wme.tx_queue_beacon_cwmin	TX queue Beacon CWmin. 802.11a/b/g default: 3.	enum[1 3 7 15 31 63 127 255 511 1023 hw_default]	hw_default
hostapd.*.wlan.wme.tx_queue_data0_aifs	TX queue Data0 (highest priority/AC_VO) AIFS [0-255]. 802.11a/b/g default: 1. Range 0-255.	ascii_text	hw_default
hostapd.*.wlan.wme.tx_queue_data0_burst	TX queue Data0 (highest priority/AC_VO) Burst. Maximum length for bursting in ms with up to 0.1 ms precision. 802.11a/g default: 1.5, 802.11b: 3.3.	ascii_text	hw_default
hostapd.*.wlan.wme.tx_queue_data0_cwmax	TX queue Data0 (highest priority/AC_VO) CWmax [1, 3, 7, 15, 31, 63, 127, 255, 511, 1023]. 802.11a/g default: 7, 802.11b: 15.	enum[1 3 7 15 31 63 127 255 511 1023 hw_default]	hw_default
hostapd.*.wlan.wme.tx_queue_data0_cwmin	TX queue Data0 (highest priority/AC_VO) CWmin [1, 3, 7, 15, 31, 63, 127, 255, 511, 1023]. 802.11a/g default: 3, 802.11b: 7.	enum[1 3 7 15 31 63 127 255 511 1023 hw_default]	hw_default
hostapd.*.wlan.wme.tx_queue_data1_aifs	TX queue Data1 (high priority/AC_VI) AIFS. 802.11a/b/g default: 1. Range 0-255.	ascii_text	hw_default
hostapd.*.wlan.wme.tx_queue_data1_burst	TX queue Data1 (high priority/AC_VI) Burst. Maximum length for bursting in ms with up to 0.1 ms precision. 802.11a/g default: 3.0, 802.11b: 6.0.	ascii_text	hw_default
hostapd.*.wlan.wme.tx_queue_data1_cwmax	TX queue Data1 (high priority/AC_VI) CWmax. 802.11a/g default: 15, 802.11b: 31.	enum[1 3 7 15 31 63 127 255 511 1023 hw_default]	hw_default
hostapd.*.wlan.wme.tx_queue_data1_cwmin	TX queue Data1 (high priority/AC_VI) CWmin. 802.11a/g default: 7, 802.11b: 15.	enum[1 3 7 15 31 63 127 255 511 1023 hw_default]	hw_default
hostapd.*.wlan.wme.tx_queue_data2_aifs	TX queue Data2 (normal priority/AC_BE) AIFS. 802.11a/b/g default: 3. Range 0-255.	ascii_text	hw_default
hostapd.*.wlan.wme.tx_queue_data2_burst	TX queue Data2 (normal priority/AC_BE) Burst. Maximum length for bursting in ms with up to 0.1 ms precision. 802.11a/b/g default: 0.	ascii_text	hw_default
hostapd.*.wlan.wme.tx_queue_data2_cwmax	TX queue Data2 (normal priority/AC_BE) CWmax. 802.11a/g default: 63, 802.11b: 127.	enum[1 3 7 15 31 63 127 255 511 1023 hw_default]	hw_default
hostapd.*.wlan.wme.tx_queue_data2_cwmin	TX queue Data2 (normal priority/AC_BE) CWmin. 802.11a/g default: 15, 802.11b: 31.	enum[1 3 7 15 31 63 127 255 511 1023 hw_default]	hw_default
hostapd.*.wlan.wme.tx_queue_data3_aifs	TX queue Data3 (low priority/AC_BK) AIFS. 802.11a/b/g default: 7. Range 0-255.	ascii_text	hw_default
hostapd.*.wlan.wme.tx_queue_data3_burst	TX queue Data3 (low priority/AC_BK) Burst. Maximum length for bursting in ms with up to 0.1 ms precision. 802.11a/b/g default: 0.	ascii_text	hw_default

hostapd.*.wlan.wme.tx_queue_data3_cwmax	TX queue Data3 (low priority/AC_BK) CWmax. 802.11a/b/g default: 1023.	enum[1 3 7 15 31 63 127 255 511 1023 hw_default]	hw_default
hostapd.*.wlan.wme.tx_queue_data3_cwmin	TX queue Data3 (low priority/AC_BK) CWmin. 802.11a/g default: 15, 802.11b: 31.	enum[1 3 7 15 31 63 127 255 511 1023 hw_default]	hw_default
hostapd.*.wlan.wme.wme_ac_be_acm	WME AC Best Effort Admission Control Mandatory.	boolean	FALSE
hostapd.*.wlan.wme.wme_ac_be_aifs	WME AC Best Effort Arbitration Inter-Frame Space.	int[1,255]	3
hostapd.*.wlan.wme.wme_ac_be_cwmax	WME AC Best Effort Contention Window Maximum, in exponent form, actual value used is (2^n)-1. Default 10 for 802.11a/g, 7 for 802.11b.	int[0,12]	10
hostapd.*.wlan.wme.wme_ac_be_cwmin	WME AC Best Effort Contention Window Minimum, in exponent form, actual value used is (2^n)-1. Default 4 for 802.11a/g, 5 for 802.11b.	int[0,12]	4
hostapd.*.wlan.wme.wme_ac_be_txop_limit	WME AC Best Effort Transmission Opportunity limit (in units of 32us).	int[0,65535]	0
hostapd.*.wlan.wme.wme_ac_bk_acm	WME AC Background Admission Control Mandatory.	boolean	FALSE
hostapd.*.wlan.wme.wme_ac_bk_aifs	WME AC Background Arbitration Inter-Frame Space.	int[1,255]	7
hostapd.*.wlan.wme.wme_ac_bk_cwmax	WME AC Background Contention Window Maximum, in exponent form, actual value used is (2^n)-1.	int[0,12]	10
hostapd.*.wlan.wme.wme_ac_bk_cwmin	WME AC Background Contention Window Minimum, in exponent form, actual value used is (2^n)-1. Default 4 for 802.11a/g, 5 for 802.11b.	int[0,12]	4
hostapd.*.wlan.wme.wme_ac_bk_txop_limit	WME AC Background Transmission Opportunity limit (in units of 32us).	int[0,65535]	0
hostapd.*.wlan.wme.wme_ac_vi_acm	WME AC Video Admission Control Mandatory.	boolean	TRUE
hostapd.*.wlan.wme.wme_ac_vi_aifs	WME AC Video Arbitration Inter-Frame Space.	int[1,255]	2
hostapd.*.wlan.wme.wme_ac_vi_cwmax	WME AC Video Contention Window Maximum, in exponent form, actual value used is (2^n)-1. Default 4 for 802.11a/g, 5 for 802.11b.	int[0,12]	4
hostapd.*.wlan.wme.wme_ac_vi_cwmin	WME AC Video Contention Window Minimum, in exponent form, actual value used is (2^n)-1. Default 3 for 802.11a/g, 4 for 802.11b.	int[0,12]	3
hostapd.*.wlan.wme.wme_ac_vi_txop_limit	WME AC Video Transmission Opportunity limit (in units of 32us). Default 94 for 802.11a/g, 188 for 802.11b.	int[0,65535]	94
hostapd.*.wlan.wme.wme_ac_vo_acm	WME AC Voice Admission Control Mandatory.	boolean	TRUE
hostapd.*.wlan.wme.wme_ac_vo_aifs	WME AC Voice Arbitration Inter-Frame Space.	int[1,255]	2
hostapd.*.wlan.wme.wme_ac_vo_cwmax	WME AC Voice Contention Window Maximum [0-12], in exponent form, actual value used is (2^n)-1. Default 3 for 802.11a/g, 4 for 802.11b.	int[0,12]	3
hostapd.*.wlan.wme.wme_ac_vo_cwmin	WME AC Voice Contention Window Minimum [0-12], in exponent form, actual value used is (2^n)-1. Default 2 for 802.11a/g, 3 for 802.11b.	int[0,12]	2
hostapd.*.wlan.wme.wme_ac_vo_txop_limit	WME AC Voice Transmission Opportunity limit (in units of 32us).	int[0,65535]	47
hostapd.*.wlan.wme.wme_enabled	Send Wireless Multimedia Extensions (WME) aka Wi-Fi Multimedia (WMM) parameters to clients.	boolean	FALSE
hostapd.*.wlan.wpa	WPA and WPA2 Configuration	N/A	N/A
hostapd.*.wlan.wpa.assoc_ping_attempts	Maximum number of association pings.	int[1,255]	3
hostapd.*.wlan.wpa.assoc_ping_timeout	Association ping timeout in time units (TU) of 1.024 ms; for management frame protection (MFP).	int[1,2147483647]	1000
hostapd.*.wlan.wpa.ieee80211w	Enable management frame protection (MFP). * disabled, * optional, * required	enum[none optional required]	none
hostapd.*.wlan.wpa.okc	Opportunistic Key Caching: allows PMK cache to be shared opportunistically among configured interfaces and BSSes. Opportunistic Key Caching is also called Proactive Key Caching used for speeding up wireless station roaming from one AP to another. It is done sharing PMK among multiple APs, so as to save time on authentication during hand-over.	boolean	FALSE
hostapd.*.wlan.wpa.peerkey	Allow PeerKey negotiation for direct links (IEEE 802.11e) between stations.	boolean	FALSE
hostapd.*.wlan.wpa.rsn_pairwise	Cipher suites for RSN/WPA2. Default same as wpa_pairwise. Robust Security Network (RSN) is an element of 802.11i authentication and encryption algorithms to be used for communications between wireless APs and wireless clients.	N/A	N/A
hostapd.*.wlan.wpa.rsn_pairwise.ccmp	CCMP (Counter Mode with Cipher Block Chaining Message Authentication Code Protocol). It is an encryption mode for RSN. Either CCMP or TKIP should be enabled, they should not be both enabled in the same time.	boolean	FALSE
hostapd.*.wlan.wpa.rsn_pairwise.tkip	TKIP (Temporal Key Integrity Protocol). It is an encryption mode for RSN. Either CCMP or TKIP should be enabled, they should not be both enabled in the same time.	boolean	TRUE
hostapd.*.wlan.wpa.rsn_preauth	Enable IEEE 802.11i/RSN/WPA2 pre-authentication.	boolean	FALSE
hostapd.*.wlan.wpa.rsn_preauth_interfaces	Interfaces from which pre-authentication frames are accepted.	N/A	N/A
hostapd.*.wlan.wpa.rsn_preauth_interfaces.*	Line entry. Keyname is line # (0-65535)	N/A	N/A

hostapd.*.wlan.wpa.rsn_preauth_interfaces.*.data	Interface used for connections to other APs (usually wired or WDS links).	if[any]	not set
hostapd.*.wlan.wpa.wpa	Enable WPA (bit field). * none No WPA enabled * wpa WPA enabled, * wpa2 IEEE 802.11i/RSN (WPA2) enabled * both Both WPA and WPA2 enabled	enum[none wpa wpa2 both]	none
hostapd.*.wlan.wpa.wpa_gmk_rekey	Time interval for rekeying GMK (master key used internally to generate GTKs (in seconds).	int[1,2147483647]	86400
hostapd.*.wlan.wpa.wpa_group_rekey	Time interval for rekeying GTK broadcast/multicast encryption keys (in seconds).	int[1,2147483647]	600
hostapd.*.wlan.wpa.wpa_key_mgmt	Key management algorithms.	N/A	N/A
hostapd.*.wlan.wpa.wpa_key_mgmt.wpa-eap	WPA-EAP	boolean	FALSE
hostapd.*.wlan.wpa.wpa_key_mgmt.wpa-eap-sha256	WPA-EAP-SHA256	boolean	FALSE
hostapd.*.wlan.wpa.wpa_key_mgmt.wpa-psk	WPA-PSK	boolean	FALSE
hostapd.*.wlan.wpa.wpa_key_mgmt.wpa-psk-sha256	WPA-PSK-SHA256	boolean	FALSE
hostapd.*.wlan.wpa.wpa_pairwise	Cipher suites (encryption algorithms) for pairwise keys (unicast packets).	N/A	N/A
hostapd.*.wlan.wpa.wpa_pairwise.ccmp	CCMP (Counter Mode with Cipher Block Chaining Message Authentication Code Protocol). It is an encryption mode for WPA. Either CCMP or TKIP should be enabled, they should not be both enabled in the same time.	boolean	FALSE
hostapd.*.wlan.wpa.wpa_pairwise.tkip	TKIP (Temporal Key Integrity Protocol). It is an encryption mode for WPA. Either CCMP or TKIP should be enabled, they should not be both enabled in the same time.	boolean	TRUE
hostapd.*.wlan.wpa.wpa_passphrase	ASCII passphrase that will be converted to PSK [length 8-63]. Default none.	ascii_char[8-63]	none
hostapd.*.wlan.wpa.wpa_psk	WPA pre-shared key (256 bits long i.e. 64 hex digits)	hex[64]	not set
hostapd.*.wlan.wpa.wpa_psk_file	Optional list of WPA PSKs for each MAC	N/A	N/A
hostapd.*.wlan.wpa.wpa_psk_file.*	Line entry. Keyname is line # (0-65535)	N/A	N/A
hostapd.*.wlan.wpa.wpa_psk_file.*.data	[MAC WPA_PSK] pair, where WPA_PSK can be 8-63 characters long ASCII string or 64 hex digits.	ascii_text	not set
hostapd.*.wlan.wpa.wpa_ptk_rekey	Maximum lifetime for PTK in seconds. Default 0 (no rekeying).	int[0,2147483647]	0
hostapd.*.wlan.wpa.wpa_strict_rekey	Rekey GTK when any STA that possesses the current GTK is leaving the BSS.	boolean	FALSE

hostname

Key	Description	Value Type	Default
hostname	EMI hostname configuration module root key.	N/A	N/A
hostname.name	Network host name. If the hostname is not set, system default hostname is in use.	ascii_text	(none)

httpd

Key	Description	Value Type	Default
httpd	EMI httpd configuration module root key.	N/A	N/A
httpd.*	HTTP server instance. Keyname is instance number.	N/A	N/A
httpd.*.allow	Whitelist setting. To specify which hosts can access the server according to IP address or a range of IP addresses.	N/A	N/A
httpd.*.allow.*	Whitelist rule instance.	N/A	N/A
httpd.*.allow.*.ip	Allow connections from this IP address for a particular instance of allow (whitelist) rule	ip4[addr]	not set
httpd.*.allow.*.mask	Set Subnet mask for a particular instance of allow (whitelist) rule.	int[0,32]	not set
httpd.*.default_index	Display this file when a directory is requested. The URL is relative to the requested directory in the server e.g. "index.html".	ascii_text	not set
httpd.*.deny	Blacklist setting. To specify which hosts cannot access the server according to IP address or a range of IP addresses.	N/A	N/A
httpd.*.deny.*	Blacklist rule instance.	N/A	N/A
httpd.*.deny.*.ip	Deny connections from this IP address for a particular instance of deny (blacklist) rule.	ip4[addr]	not set
httpd.*.deny.*.mask	Set Subnet mask for a particular instance of deny (blacklist) rule.	int[0,32]	not set
httpd.*.deny_by_default	Deny all connections by default	boolean	FALSE
httpd.*.enable_php	Use php binary to interpret *.php requests	boolean	FALSE
httpd.*.enabled	Enable HTTP server instance	boolean	FALSE
httpd.*.error_page.*	Specify error page for HTTP error <code>, url format.	ascii_text	
httpd.*.home	Home directory	ascii_text	
httpd.*.ip	IP to bind to. The IP address that the particular httpd instance will bind to.	ip	0.0.0.0
httpd.*.port	Port to bind to. The port number that the particular httpd instance will bind to.	int[0,65535]	80
httpd.*.protect	Password protection	N/A	N/A
httpd.*.protect.*	Password protection rule	N/A	N/A
httpd.*.protect.*.prefix	Password protected URL prefix	ascii_text	
httpd.*.protect.*.user	List of allowed users	N/A	N/A

httpd.*.protect.*.user.*	Allowed user login credentials	N/A	N/A
httpd.*.protect.*.user.*.name	Allowed username	ascii_text	
httpd.*.protect.*.user.*.pass	Corresponding password	ascii_text	
httpd.*.proxy	List of reverse proxy definitions	N/A	N/A
httpd.*.proxy.*	Reverse proxy rule. The proxy rules control whether or not a proxy connection or an email is actually relayed to its intended destination.	N/A	N/A
httpd.*.proxy.*.host	New host . It specifies a hostname of the http proxy server to relay.	ascii_text	
httpd.*.proxy.*.newpath	New path prefix. It specifies the path of the http proxy server to relay.	ascii_text	
httpd.*.proxy.*.port	New port. It specifies a port number of the http proxy server to relay.	int[0,65535]	
httpd.*.proxy.*.url	URL prefix this rule applies to. When this URL is requested to the proxy, the proxy will forward the request to http://[host][:port]/newpath	ascii_text	
httpd.*.realm	Authentication realm for basic authentication. A Realm is a "database" of usernames and passwords that identify valid users of a web application (or set of web applications), plus an enumeration of the list of roles associated with each valid user.	ascii_text	
httpd.*.redirect_host	Redirect target host	ascii_text	
httpd.*.redirect_path	Redirect target path	ascii_text	

ip

Key	Description	Value Type	Default
ip	EMI ip configuration module root key.	N/A	N/A
ip.addr	IP Protocol Settings.	N/A	N/A
ip.addr.*	Networking interface. Keyname is interface name.	N/A	N/A
ip.addr.*.*	Address. Keyname is address id (text or number).	N/A	N/A
ip.addr.*.*.label	Address alias name. Each address may be tagged with a label string. In order to preserve compatibility with Linux-2.0 net aliases, this string must coincide with the name of the device or must be prefixed with the device name followed by colon.	ascii_text	
ip.addr.*.*.local	Local IP address	ip	
ip.addr.*.*.peer	Peer IP address. It is the address of the remote endpoint for point-to-point interfaces.	ip	
ip.addr.*.*.prefixlen	Netmask length	int[0,32]	
ip.link	ARP (Address Resolution Protocol) Protocol and MAC address related settings. ARP is mainly used for translating IP addresses to Ethernet Mac addresses.	N/A	N/A
ip.link.*	Interface link settings. Keyname is interface name.	N/A	N/A
ip.link.*.address	Link level address, i.e. MAC address or Ethernet address.	mac	
ip.link.*.allmulti	Receive all multicast frames (boolean)	boolean	FALSE
ip.link.*.broadcast	Link level broadcast address (MAC). A broadcast address is a MAC address that allows information to be sent to all machines on a given subnet rather than a specific machine. The typical value is FF:FF:FF:FF:FF:FF	mac	
ip.link.*.mtu	Maximum transmission unit. It refers to the size (in bytes) of the largest Ethernet frame that the Ethernet can pass onwards. The typical MTU value for Ethernet is 1500	int[0,65535]	
ip.link.*.multicast	Enable multicasts on the interface	boolean	FALSE
ip.link.*.noarp	Disable ARP on interface (boolean, default false)	boolean	FALSE
ip.link.*.txqlen	Transmission queue length. Fast interface (e.g. 1G Ethernet) typically need longer queue for example 1000. For slow interface, the queue is typically set to 100. Longer queue will take up more memory. Transmission latency will be high if a slow interface has a tx long queue.	int[0,65535]	0
ip.link.*.up	Interface activity (boolean)	boolean	FALSE
ip.neigh	Neighbor list related settings. Neighbour is ARP cache which stores the mapping between IP addresses and MAC addresses that are used recently. Disabling ARP will require static neighbor table mappings for all hosts wishing to exchange packets across the Ethernet.	N/A	N/A
ip.neigh.*	Networking interface. Keyname is interface name. e.g. ip.neigh.eth1	N/A	N/A
ip.neigh.*.*	Neighbor network address (IPv4 or IPv6) to link level address mapping. Keyname is MAC address.	ip	

ip.neigh.*.*.nud	<p>NUD (Neighbour Unreachability Detection) state</p> <ul style="list-style-type: none"> * permanent - the neighbour entry is valid forever and can be only be removed administratively * reachable - the neighbour entry is valid until the reachability timeout expires * noarp - the neighbour entry is valid. No attempts to validate this entry will be made but it can be removed when its lifetime expires * stale - the neighbour entry is valid, but is probably already unreachable, so the kernel will try to check it at the first transmission * incomplete - the neighbour is in the process of resolution * delay - a packet has been sent to the stale neighbour and the kernel is waiting for confirmation * probe - the delay timer expired but no confirmation was received. The kernel has started to probe the neighbour with ARP/NDISC messages * failed - resolution has failed 	enum[permanent reachable noarp none stale incomplete delay probe failed]	
ip.route	Routing Settings. IP route table setting.	N/A	N/A
ip.route.*	Kernel route table id. Keyname is index (0-65535).	N/A	N/A
ip.route.*.*	Kernel route table entry. Keyname is index (0-65535).	N/A	N/A
ip.route.*.*.dev	Route output interface e.g. eth2	if[any]	
ip.route.*.*.dst	Route destination address, or 'def4' or 'def6'; mandatory <ul style="list-style-type: none"> * def4: default route for IPv4 * def6: default route for IPv6 	ip enum[def4 def6]	
ip.route.*.*.gw	Route gateway address	ip	
ip.route.*.*.priority	Route metric, priority value of the route. Lower the value, higher the priority.	int[0,4294967295]	
ip.route.*.*.tos	Route type of service. The packet to be routed is checked against its TOS and tried to be put to a route that has a matching TOS	int[0,255]	
ip.rule	IP marking and matching rules. This is also known as Routing Policy Database (RPDB). The kernel will first try to match the IP packet with a rule and then forward to the associated routing table to complete the routing.	N/A	N/A
ip.rule.*	Rule priority, keyname is an integer for priority.	N/A	N/A
ip.rule.*.action	<p>Action upon match. The text should be: table N goto N nop blackhole unreachable prohibit</p> <ul style="list-style-type: none"> * table N - look up in specified route table N e.g. ip.route.N * goto N - jump to another rule N * nop - no operation, ignore this rule * blackhole - the destinations are unreachable. Packets are discarded silently * unreachable - these destinations are unreachable. Packets are discarded and the ICMP message host unreachable is generated * prohibit - these destinations are unreachable. Packets are discarded and the ICMP message communication administratively prohibited is generated 	ascii_text	
ip.rule.*.dst	Destination prefix. The destination address of packet must match this IP prefix.	ip[cidr]	
ip.rule.*.fwmark	Firewall mark. It is a mark carried through the OS kernel in the data structure representing the packet. It is used for packet routing. It is optional. If not defined, fwmark will not be compared for route selection.	hex[n]	
ip.rule.*.fwmask	firewall mask	hex[n]	
ip.rule.*.iif	Source interface name. This particular rule will apply only to packets received on this interface.	if[any]	
ip.rule.*.src	Source prefix	ip[cidr]	
ip.rule.*.tos	Type of service. It is a flag in the header of an IP packet which is sometimes honored by upstream routers. Some routers on the Internet respect the ToS flag and others do not, however, the ToS flag can be used as part of the decision about where to route a given packet.	int[0,255]	

iptables

Key	Description	Value Type	Default
iptables	EMI iptables configuration module root key.	N/A	N/A
iptables.*	Rule entry. Keyname is line # (1-65535).	N/A	N/A
iptables.*.chain	Chain name. Can be one of the built-in chains or user-defined.		
iptables.*.destination	Can be either a network name, a hostname (names are not recommended), a network IP address (with /mask), or a plain IP address.		
iptables.*.enabled	Whether rule is in use or not.		
iptables.*.fragment	Rule only refers to second and further fragments of fragmented packets.		

iptables.*.goto	The user-defined chain where the processing should continue. Return will not continue processing in this chain.		
iptables.*.in-interface	Name of an interface via which a packet was received (only for packets entering the INPUT, FORWARD and PREROUTING chains).		
iptables.*.jump	Target of the rule. Can be user-defined chain, builtin target or an extension.		
iptables.*.match	Matching module name.		
iptables.*.out-interface	Name of an interface via which a packet is going to be sent (for packets entering the FORWARD, OUTPUT and POSTROUTING chains).		
iptables.*.policy	Key value is default policy for specified chain. Only built-in (non-user-defined) chains can have policies.		
iptables.*.protocol	The protocol of the rule or of the packet to check.		
iptables.*.rule_number	Rule position inside its chain.		
iptables.*.source	Can be either a network name, a hostname (names are not recommended), a network IP address (with /mask), or a plain IP address.		
iptables.*.table	Table name.		
iptables.*.target	Target specific options.	N/A	N/A
iptables.*.target.snat	Specifies that the source address of the packet should be modified and rules should cease being examined. Only valid in the NAT table, in the POSTROUTING chain.		
iptables.*.target.snat.random	Randomize port mappings (kernel >= 2.6.21)		
iptables.*.target.snat.to-source	New source IP address, an inclusive range of IP addresses, and optionally, a port range (valid only for TCP or UDP protocols).		
iptables.*.tcp	Options that can be used only by TCP protocol, i.e. protocol=tcp.		
iptables.*.tcp.destination-port	Destination port or port range. This can either be a service name or a port number.		
iptables.*.tcp.source-port	Source port or port range. This can either be a service name or a port number.		
iptables.*.tcp.syn	Only match TCP packets with the SYN bit set and the ACK,RST and FIN bits cleared. Such packets are used to request TCP connection initiation.		
iptables.*.tcp.tcp-flags	Match when the TCP flags are as specified.		
iptables.*.tcp.tcp-option	Match if TCP option set		
iptables.chains	User-defined chains.	N/A	N/A
iptables.chains.*	Chain entry. Keyname is line # (1-65535).	N/A	N/A
iptables.chains.*.enabled	Whether chain is in use or not.		
iptables.chains.*.name	Chain name.		
iptables.chains.*.table	Table to which this chain belongs.		

iwconfig

Key	Description	Value Type	Default
iwconfig	EMI iwconfig configuration module root key.	N/A	N/A
iwconfig.*	Wireless settings for an interface. Keyname is wireless interface name (physical or virtual). e.g. iwconfig.eth0.ap = 00:60:1D:01:23:45	N/A	N/A
iwconfig.*.ap	Select AP. Value can either be the MAC address of the AP, or auto any off. * set MAC of AP: force the card to register to the Access Point given by the address * auto or any: force the card to reassociate with the currently best AP * off: re-enable automatic mode without changing the current AP	ascii_text	
iwconfig.*.chan	Wireless operation channel. This is the channel number typically starts at 1. For 802.11b/g, the maximum channel number is 13.	int[0,65535]	
iwconfig.*.ssid	ESSID name. Value can be any string (including space, 'off', 'any' and 'on', they all considered as ESSID, not an option). Max length 255. The ESSID is used to identify cells which are part of the same virtual network. As opposed to the AP Address or NWID which define a single cell, the ESSID defines a group of cells connected via repeaters or infrastructure, where the user may roam transparently. e.g. iwconfig.eth0.ssid = "My Network"	ascii_text	
iwconfig.*.ssidswitch	Switch ESSID off(any) on. Option off and option any are identical for this key. off or any: disable ESSID checking, stations may select any ESSID (i.e. ESSID promiscuous) on: enable ESSID checking	enum[off any on]	on

iwconfig.*.frag	Fragmentation threshold. It could be either a number (0-2346), 'auto', 'fixed' or 'off'. Applicable values depend on the radio driver. Fragmentation allows to split an IP packet in a burst of smaller fragments transmitted on the medium. In most cases this adds overhead, but in a very noisy environment this reduces the error penalty and allow packets to get through interference bursts. This parameter sets the maximum fragment size which is always lower than the maximum packet size.	ascii_text	off
iwconfig.*.freq	Frequency in N.NNN [k M G] e.g. "2.46G" for 2.46GHz frequency.	ascii_text	
iwconfig.*.mode	Wi-Fi operation mode. * IBSS - Ad-Hoc mode * AP - Access Point (Master mode) * STA - Managed mode (client, station mode) * repeater - Node forwards packets between other wireless nodes * secondary - The node acts as a backup master/repeater * monitor - Node is not associated to any cell and passively monitors all packets on the frequency * auto - Automatic selection, depending on wireless driver	enum[IBSS AP STA repeater secondary monitor auto]	STA
iwconfig.*.modulation	Specific set of modulations, e.g. 11g 11a CCK OFDMg ...; list of available modulations depend on the HW/driver combination and can be retrieved with mcstat iwlist.modulation.	ascii_text	
iwconfig.*.nick	Any string. Nickname for 802.11. Not supported by all drivers. Supported by some diagnostics software but is not used by any other protocols.	ascii_text	
iwconfig.*.nwid	Legacy network ID setting. Used by old non-80211 hardware only. ESSID is used in modern hardware instead. Any string or 'on' or 'off'. Use 'off' to disable network ID checking.	ascii_text	
iwconfig.*.pw	Transmission power setting.	N/A	N/A
iwconfig.*.pw.enabled	Power management enabled.	boolean	FALSE
iwconfig.*.pw.maxperiod	Set the maximum period between wake ups. By default, value is a number, in second. Use suffix m or u after the number for milliseconds or microseconds. E.g. 200m, 300u, etc.	ascii_text	
iwconfig.*.pw.maxtimeout	Set the maximum timeout before going back to sleep. By default, value is a number, in second. Use suffix m or u after the number for milliseconds or microseconds.	ascii_text	
iwconfig.*.pw.minperiod	Set the minimum period between wake ups. By default, value is a number, in second. Use suffix m or u after the number for milliseconds or microseconds.	ascii_text	
iwconfig.*.pw.mintimeout	Set the minimum timeout before going back to sleep. By default, value is a number, in second. Use suffix m or u after the number for milliseconds or microseconds.	ascii_text	
iwconfig.*.pw.mode	Set the power management mode to: * all - receive all packets * unicast - receive unicast packets only, discard multicast and broadcast * multicast - receive multicast and broadcast only, discard unicast packets	enum[all unicast multicast]	all
iwconfig.*.pw.period	Set the period between wake ups. By default, value is a number, in second. Use suffix m or u after the number for milliseconds or microseconds. E.g. 200m, 300u, etc.	ascii_text	
iwconfig.*.pw.timeout	Set the timeout before going back to sleep. By default, value is a number, in second. Use suffix m or u after the number for milliseconds or microseconds. E.g. 200m, 300u, etc.	ascii_text	
iwconfig.*.rate	Bitrate used on 802.11, it is a number in bit per second. You may append suffix k, M or G to the value, e.g. NNN[k M G]. 'auto' or 'fixed' can also be used here. * auto - switch to automatic bit-rate mode, fallback to lower rate on noisy channels (default) * fixed - switch to fixed setting	ascii_text	auto
iwconfig.*.retry	Retry mechanism setting.	N/A	N/A
iwconfig.*.retry.lifetime	Set the maximum length of time the MAC should retry. By default, value is a number, in second. Use suffix m or u after the number for milliseconds or microseconds.	ascii_text	
iwconfig.*.retry.limit	Set the maximum number of retries.	int[0,65535]	
iwconfig.*.rts	Size of the smallest packet for RTS/CTS protocol to be used when sending. Value in bytes or 'off', 'auto', 'fixed'. Functionality of 'auto' and 'fixed' are driver-specific.	ascii_text	
iwconfig.*.sensitivity	Sensitivity threshold. It defines how sensitive the radio is to poor operating conditions. The actual functionality is hardware and driver - specific. The parameter can control the handover/roaming threshold or lowest signal level the station remains associated with current access point. Some radio units may control this value automatically.	int[0,65535]	
iwconfig.*.txpower	Transmission power. Power can be set in dBm (plain integer) or Watts (nmW, for example 15mW). 'off' can be used to disable the radio. 'auto' or 'fixed' disables power control (driver specific feature).	ascii_text	

iwconfig.*.wep	Wireless Equivalent Privacy configuration options. Wired Equivalent Privacy (WEP) is the encryption algorithm built into the 802.11 (Wi-Fi) standard. WEP encryption uses the RC4 stream cipher with 40 or 104 bit keys and a 24 bit initialization vector.	N/A	N/A
iwconfig.*.wep.current	WEP key in use. User could select either key1, key2, key3 or key4 as the active key for WEP.	enum[1 2 3 4]	1
iwconfig.*.wep.key1	wep key 1, hex digits as XXXXXXXXXXXX or XXXX-XXXX-XXXX (Max 32 HEX digits, leading zeros can be removed to make key shorter).	hex[0,FFFFFFFFFFFFFFFF]	
iwconfig.*.wep.key2	wep key 2, hex digits as XXXXXXXXXXXX or XXXX-XXXX-XXXX (Max 32 HEX digits, leading zeros can be removed to make key shorter).	hex[0,FFFFFFFFFFFFFFFF]	
iwconfig.*.wep.key3	wep key 3, hex digits as XXXXXXXXXXXX or XXXX-XXXX-XXXX (Max 32 HEX digits, leading zeros can be removed to make key shorter).	hex[0,FFFFFFFFFFFFFFFF]	
iwconfig.*.wep.key4	wep key 4, hex digits as XXXXXXXXXXXX or XXXX-XXXX-XXXX (32 HEX digits, leading zeros can be removed to make key shorter).	hex[0,FFFFFFFFFFFFFFFF]	
iwconfig.*.wep.mode	WEP enabled.	boolean	FALSE

iwpriv

Key	Description	Value Type	Default
iwpriv	EMI iwpriv configuration module root key.	N/A	N/A
iwpriv.*	Settings for wireless interface. Keyname is interface name.	N/A	N/A
iwpriv.*.*	Priority index. Keyname is an integer for index (or sequence), the command with smaller index will be executed earlier.	N/A	N/A
iwpriv.*.*.arg	Iwpriv argument.	ascii_text	
iwpriv.*.*.com	Iwpriv command.	ascii_text	
iwpriv.*.*.ioctl	IOCTL (I/O Control) number.	hex[0,FFFF]	

madwifi

Key	Description	Value Type	Default
madwifi	EMI madwifi configuration module root key.	N/A	N/A
madwifi.bstuck	Stuck beacon threshold	int[0,65535]	10
madwifi.country	Country code. Strip leading zeros, start with non-zero. Reference: http://www.nw.com/zone/iso-country-codes	int[0,999]	0
madwifi.enabled	Madwifi driver enabled. Set it false or delete the key will unload the madwifi driver.	boolean	FALSE
madwifi.maxvaps	Maximum number of Virtual Access Points	int[2,64]	4
madwifi.outdoor	Outdoor frequencies in use. 'False' for indoor use.	boolean	FALSE
madwifi.ratectl	Rate control algorithm * amrr Uses binary exponential backoff to avoid attempting to increase the bit-rate too often. It was conceived for high latency systems. * minstrel: a mac80211 rate control algorithm ported over from MadWifi which supports multiple rate retries and claimed to be one of the best rate control algorithm. See detailed doc: http://wireless.kernel.org/en/developers/Documentation/mac80211/RateControl/minstrel * onoe: A simple, robust algorithm that decreases the bit-rate when packets, on average, need at least 1 retry. Increases the bit-rate when less than 10% of packets require a retry. It performs very well in indoor 802.11b environments. * sample: chooses the bit-rate it predicts will provide the most throughput based on estimates of the expected per-packet transmission time each bit-rate. Sample Rate Control periodically sends packets at bit rates other than the current one to estimate when another bit-rate will provide better performance.	enum[amrr minstrel onoe sample]	sample
madwifi.rfkill	Enable/disable RF KILL capability. RF (Radio Frequency) kill: which allows all wireless signals to be stopped (e.g., so one can turn them off on an airplane)	boolean	FALSE
madwifi.tpc	Per-packet transmit power control (TPC) capability	boolean	FALSE
madwifi.xchan	Enable/disable extended channel mode	boolean	TRUE

mesh

Key	Description	Value Type	Default
mesh	EMI mesh configuration module root key.	N/A	N/A
mesh.acl	Local access list.	N/A	N/A
mesh.acl.*	Local access list entry. Keyname is MAC address.	N/A	N/A

mesh.acl.*.enabled	Entry is in list [true/false].	boolean	FALSE
mesh.acl.type	Local access list type * black Blacklist will prevent the MAC addresses on the ACL to connect. * white Whitelist will allow only the MAC addresses on the ACL to connect.	enum[black white]	black
mesh.aodvcostdelay	Cost delay for AODV routing [true/false].	boolean	FALSE
mesh.aodvrefreshinterval	AODV route refresh interval [0, 5-3600000ms]. 0 is off.	int[0,5,360000]	15000
mesh.defaultauth	Default authentication type. * Closed No connections allowed * Open Any connections allowed * EAP (Not supported in this version) * PSK (Not supported in this version) * SRP SRP authentication See MeshDriver User Guide for details.	enum[Closed Open EAP PSK SRP]	Open
mesh.enabled	Mesh Driver enabled. Set false or delete the key will remove Mesh Driver.	boolean	FALSE
mesh.helloint	Hello interval in milliseconds.	int[0,65535]	250
mesh.iflist	Network interface list on which meshdriver is loaded	N/A	N/A
mesh.iflist.*	Keyname is interface name. Value 'true' means it is included in iflist.	boolean	FALSE
mesh.linktimeout	Time of inactivity until a link becomes disconnected [1-3600000]ms.	int[1,3600000]	4000
mesh.meshid	Mesh network id.	ascii_text	MYMESH
mesh.nodename	Mesh node's name.	ascii_text	same as hostname
mesh.ports	Port specific settings.	N/A	N/A
mesh.ports.*	Settings for a port. Keyname is port index (1 to 100).	N/A	N/A
mesh.ports.*.flush_timeout	Interval for flushing frame buffers of mesh links in milliseconds	int[0,5000]	0
mesh.ports.*.mesh_auth	Authentication type for mesh links. * Closed No connections allowed * Open Any connections allowed * EAP (Not supported in this version) * PSK (Not supported in this version) * SRP SRP authentication See MeshDriver User Guide for details.	enum[Default Closed Open EAP PSK SRP]	Default
mesh.ports.*.virtual_auth	Authentication type for non-mesh links. * Closed No connections allowed * Open Any connections allowed * EAP (Not supported in this version) * PSK (Not supported in this version) * SRP SRP authentication See MeshDriver User Guide for details.	enum[Default Closed Open]	Default
mesh.psk	Preshared key for SRP authentication.	ascii_text	[none]

ntpclient

Key	Description	Value Type	Default
ntpclient	EMI ntpclient configuration module root key.	N/A	N/A
ntpclient.count	Maximum number of NTP queries (0 = no limit).	int[0,65535]	1
ntpclient.enabled	Enable setting the system time with NTP.	boolean	FALSE
ntpclient.goodness	Stop after getting a result more accurate than this value (in microseconds).	int[0,65535]	0
ntpclient.host	NTP server address.(e.g. "ntp.research.gov")	ascii_text	
ntpclient.interval	Interval between NTP queries (in seconds).	int[15,65535]	600
ntpclient.mindelay	Minimum packet delay for transaction (in microseconds).	int[0,65535]	800
ntpclient.port	Local NTP client UDP port (0 = any available).	int[0,65535]	0
ntpclient.trust	Trust network and server (disable RFC-4330 recommended cross-checks).	boolean	FALSE

snmpd

Key	Description	Value Type	Default
snmpd	EMI snmpd configuration module root key.	N/A	N/A
snmpd.access	Access configuration: to grant access to the view.	N/A	N/A
snmpd.access.*	Group name. Keyname is one of the group names, which is defined in snmpd.group keys	N/A	N/A
snmpd.access.*.context	SNMP context is a collection of MIB objects, often associated with an entity.	ascii_text	"
snmpd.access.*.match	Specifies how context should be matched against the context of the incoming request. * exact: match exactly * prefix: match prefix	enum[exact prefix]	exact
snmpd.access.*.notify	Specifies the view (name) to be used for notify	ascii_text	none
snmpd.access.*.read	Specifies the view (name) to be used for read	ascii_text	all

snmpd.access.*.seclvel	Security level (levels of authentication): * noauth: no authentication level * auth: authentication level * priv: privacy level	enum[noauth auth priv]	noauth
snmpd.access.*.secmod	Security model * v1: SNMPv1, described in RFC 1157 * v2c: SNMPv2c, described in RFC 1902 * usm: User-Based Security Model * any: any model above	enum[any v1 v2c usm]	any
snmpd.access.*.write	Specifies the view (name) to be used for write. The view name is specified in snmpd.view.	ascii_text	none
snmpd.com2sec	Map community to security name. An SNMP community is the group that devices and management stations running SNMP belong to. It helps define where information is sent. The community name is used to identify the group. A SNMP device or agent may belong to more than one SNMP community. It will not respond to requests from management stations that do not belong to one of its communities.	N/A	N/A
snmpd.com2sec.*	Index for com2sec. Keyname is an integer as index.	N/A	N/A
snmpd.com2sec.*.comm	Community name, such as public, private, etc. Only requests from computers in the list of community names will be accepted. "public" means the SNMP agent will respond to any host (management station).	ascii_text	public
snmpd.com2sec.*.secname	Security name, e.g. ro (read only), rw (read & write), etc.	ascii_text	ro
snmpd.com2sec.*.source	Specifies the source address where the request is coming from. Only requests from hosts on the list of IP addresses are accepted.	ascii_text	default
snmpd.enabled	snmpd enabled	boolean	FALSE
snmpd.group	Map group to security name	N/A	N/A
snmpd.group.*	Index for group. Keyname is an integer as index.	N/A	N/A
snmpd.group.*.gname	Group name, e.g. public, private	ascii_text	public
snmpd.group.*.secmod	Security model * v1: SNMPv1, described in RFC 1157 * v2c: SNMPv2c, described in RFC 1902 * usm: User-Based Security Model	enum[v1 v2c usm]	v1
snmpd.group.*.secname	Security name, e.g. ro (read only), rw (read write), etc.	ascii_text	ro
snmpd.view	view, to map with group. An SNMP view filters objects from the entire MIB and defines a subset of MIB objects.	N/A	N/A
snmpd.view.*	View name. Keyname is the name of the view. It is a label for the view record that you are updating or creating. The name is used to reference the record.	N/A	N/A
snmpd.view.*.mask	Mask, this is optional.	ip4[netmask]	
snmpd.view.*.subtree	Subtree. It could be any subtree of what's available through snmpd, for example the System group ("System").	ascii_text	0,1
snmpd.view.*.type	Type of view * included: subtree view is included * excluded: subtree view is excluded	enum[include excluded]	included

sysctl

Key	Description	Value Type	Default
sysctl	EMI sysctl configuration module root key.	N/A	N/A
sysctl.*	Keyname is the sysctl key string. Quotation marks MUST be used for the key if it contains dot(s).	ascii_text	

syslog

Key	Description	Value Type	Default
syslog	EMI syslog configuration module root key.	N/A	N/A
syslog.enabled	Enable system logging.	boolean	FALSE
syslog.keep	Number of rotated logs to keep [0-99]. Default: 1.	int[0,99]	1
syslog.level	Logging level 1-8 * 1 action must be taken immediately * 2 critical conditions * 3 error conditions * 4 warning conditions * 5 normal but significant condition * 6 informational * 7 debug-level messages only * 8 dummy level	int[1,8]	1
syslog.maxsize	Maximum size in KB before rotate (0 = no limit). Default: 200.	int[0,4294967295]	200
syslog.remote	Logging to remote host settings.	N/A	N/A

syslog.remote.enabled	Enable logging to remote host.	boolean	FALSE
syslog.remote.host	Host for remote logging.	ascii_text	
syslog.remote.local	Store logs locally also when remote logging is enabled.	boolean	FALSE
syslog.remote.port	Port for remote logging [0-65535].	int[0,65535]	514
syslog.small	Enable smaller logging output.	boolean	FALSE

udhcpd

Key	Description	Value Type	Default
udhcpd	EMI udhcpd configuration module root key.	N/A	N/A
udhcpd.*	Settings for an interface. Keyname is the interface name.	N/A	N/A
udhcpd.*.enabled	DHCP client running true/false.	boolean	FALSE
udhcpd.*.release	Release IP on quit [true/false].	boolean	FALSE
udhcpd.*.retries	Send up to this number of request packets.	int[1,10]	3
udhcpd.*.timeout	Try to get a lease for this number of seconds.	int[1,60]	3

udhcpd

Key	Description	Value Type	Default
udhcpd	EMI udhcpd configuration module root key.	N/A	N/A
udhcpd.*	Udhcpd Index. Each index implements an udhcpd instance.	N/A	N/A
udhcpd.*.auto_time	The time period (in seconds) at which udhcpd will write out a dhcpd.leases file. If this is 0, udhcpd will never automatically write a lease file. Default is 2 hours.	int[0,4294967295]	7200
udhcpd.*.bootsize	The size of the boot image for the client.	int[0,65535]	
udhcpd.*.broadcast	Specifies broadcast address for the interface	ip4[addr]	
udhcpd.*.conflict_time	The amount of time (in seconds) that an IP will be reserved (leased) for if an ARP conflict occurs. Default is 1 hour.	int[0,4294967295]	3600
udhcpd.*.cookiesrv	Cookie server (One or more IPv4 addresses). The cookie server option specifies cookie servers IP addresses available to the client. E.g. udhcpd.1,*cookiesrv="192.168.10.1 192.168.10.20"	ascii_text	
udhcpd.*.decline_time	The amount of time (in seconds) that an IP will be reserved (leased) for if a DHCP decline message is received. Default is 1 hour.	int[0,4294967295]	3600
udhcpd.*.dns	DNS server definition list. This key specifies a list of Domain Name System name servers available to the client.	N/A	N/A
udhcpd.*.dns.*	DNS server, keyname is the index. Recommended to be a number starting from 1.	ip4[addr]	
udhcpd.*.domain	Domain name. This option specifies the domain name that client should use when resolving hostnames via the Domain Name System.	ascii_text	
udhcpd.*.enabled	The udhcpd (instance) is enabled or not.	boolean	FALSE
udhcpd.*.end	The end of the IP lease block	ip4[addr]	192.168.0.254
udhcpd.*.interface	The interface name, on which udhcpd will run. If it is not set, eth0 is taken into use.	if[any]	eth0
udhcpd.*.ipttl	IP's time to live (TTL). This option specifies the default time-to-live that the client should use on outgoing datagrams. The TTL is specified as an octet with a value between 1 and 255.	int[0,255]	
udhcpd.*.lease	Lease time in seconds. Default is 10 days.	int[0,4294967295]	864000
udhcpd.*.logsrv	Log server (One or more IPv4 addresses). The log server option specifies a list of MIT-LCS UDP log servers available to the client.	ascii_text	
udhcpd.*.lprsrv	LPR server (One or more IPv4 addresses). The LPR server option specifies a list of RFC 1179 line printer servers available to the client.	ascii_text	
udhcpd.*.max_leases	The maximum number of leases	int[0,4294967295]	254
udhcpd.*.min_lease	Minimum lease time (seconds). If a lease to be given is below this value, the min time is instead used.	int[0,4294967295]	60
udhcpd.*.mtu	Specifies MTU size for the receiving interface.	int[0,65535]	
udhcpd.*.namesrv	Name server (One or more IPv4 addresses). The name server option specifies a list of name servers available to the client.	ascii_text	
udhcpd.*.ntpsrv	Network Time Protocol Server. This option specifies IP address indicating NTP servers available to the client.	ip4[addr]	
udhcpd.*.offer_time	How long an offered address is reserved (leased) in seconds.	int[0,4294967295]	60
udhcpd.*.remaining	If remaining is true (default), udhcpd will store the time remaining for each lease in the udhcpd leases file. This is for embedded systems that cannot keep time between reboots. If you set remaining to no, the absolute time that the lease expires at will be stored in the dhcpd.leases file.	boolean	TRUE
udhcpd.*.rootpath	Specifies the path-name that contains the client's root disk	ascii_text	
udhcpd.*.router	Specify the client's gateway.	ip4[addr]	
udhcpd.*.siaddr	IP address of the server to use in the next step of the client's bootstrap process. This is a bootp option, check RFC 951 for details.	ip4[addr]	0.0.0.0

udhcpd.*.sname	Optional server host name. This is bootp option, check RFC 951 for details.	ascii_text	
udhcpd.*.start	The start of the IP lease block	ip4[addr]	192.168.0.20
udhcpd.*.static_lease	Forced IP-MAC Mapping	N/A	N/A
udhcpd.*.static_lease.*	IP-MAC Mapping. Keyname is MAC address(MUST have : between hex numbers), value is the IP address to be assigned to the MAC.	ip4[addr]	
udhcpd.*.subnet	Client's subnet mask (defined in RFC 950)	ip4[netmask]	
udhcpd.*.swapsrv	Swap server (One or more IPv4 addresses). This specifies the IP address of the client's swap server.	ip4[addr]	
udhcpd.*.tftp	Specifies the TFTP server name. This option is used to identify a TFTP (Trivial File Transfer Protocol) server.	ascii_text	
udhcpd.*.timesrv	Time server (One or more IPv4 addresses). The time server option specifies RFC 868 time servers available to the client.	ascii_text	
udhcpd.*.timezone	Specify the timezone. Check RFC 4833 for details.	ascii_text	
udhcpd.*.wins	Windows Internet Name Service (WINS) sever IP. This option specifies RFC 1001/1002 NBNS (NetBIOS name server) name server. NetBIOS Name Service is currently more commonly referred to as WINS.	ip4[addr]	

user

Key	Description	Value Type	Default
user	EMI user configuration module root key.	N/A	N/A
user.*	Keyname is user name	N/A	N/A
user.*.comment	Comment text	ascii_text	not set
user.*.gid	Group ID	int[0,65535]	0
user.*.home	Home directory	ascii_text	/
user.*.pass	MD5 hashed password	ascii_char[31]	not set
user.*.shell	Shell in use. It must be /bin/ash now	enum[/bin/ash]	/bin/ash
user.*.uid	User ID	int[0,65535]	0

webui

Key	Description	Value Type	Default
webui	EMI webui configuration module root key.	N/A	N/A
webui.*	User Defined Keys	ascii_text	
webui.**	User Defined Keys	ascii_text	
webui.**.*	User Defined Keys	ascii_text	

wlanconfig

Key	Description	Value Type	Default
wlanconfig	EMI wlanconfig configuration module root key.	N/A	N/A
wlanconfig.*	Settings for physical interface. Keyname is physical interface name.	N/A	N/A
wlanconfig.**	Virtual interface for the physical interface. Keyname is virtual interface name.	N/A	N/A
wlanconfig.**.bssid	If enabled then VAP MAC will be different from underlying device MAC. If VAP is not enabled, the BSSID is the MAC address of the wireless access point. BSSID uniquely identifies each BSS (the SSID however, can be used in multiple, possibly overlapping, BSSs). In an infrastructure BSS, the BSSID is the MAC address of the wireless access point (WAP).	boolean	FALSE
wlanconfig.**.enabled	VAP enabled	boolean	FALSE
wlanconfig.**.wlanmode	wlan mode * STA - Managed mode (client, station mode) * IBSS - Ad-Hoc mode * AP - Access Point (Master mode) * monitor - Node is not associated to any cell and passively monitors all packets on the frequency * WDS - WDS (wireless distribution system) mode * ahdemo - ad-hoc demo (aka pseudo IBSS) mode	enum[STA IBSS AP monitor WDS ahdemo]	STA