



Software 6 Management Guide

Release 6.9.2

Westermo Network Technologies AB

July 1, 2020

www.westermo.com
info@westermo.com

Version control

Document identification	Software 6 Management Guide
Authors	Westermo Network Technologies AB
Owner	Westermo Network Technologies AB
Revision hash	417e89eabfe61bb371ac19e37d63edb84049ab9f

Notice

The reproduction, distribution and utilization of this document as well as the communication of its contents to others without express authorization is prohibited. Offenders will be held liable for the payment of damages. All rights reserved in the event of the grant of a patent, utility model or design.

Contents

1	Introduction	4
1.1	Supported Products	4
1.2	Supported Features	5
1.3	Installation Country, Product Use	6
1.4	Delivery Content	6
1.4.1	Important Safety Notes	7
1.5	Information on Disposal of Old Electronic Equipment	7
1.6	License and Copyright	7
2	Installation	9
3	Quick Start	10
3.1	Starting the product for the first time	10
3.2	Setting the IP address	11
3.2.1	Setting the IP address via Web Interface	11
3.3	Troubleshooting	11
4	Administration	13
4.1	Web-Based Management (Web Interface)	13
4.2	Simple Network Management Protocol (SNMP)	14
4.2.1	Supported Commands	15
4.2.2	Availability	15
4.2.3	Example: Change and permanently save a Parameter through SNMP command line tool	15
4.2.4	Supported MIBs	16
4.3	Configuration Files	16
4.4	Command Line Interface (CLI)	17
4.5	Web Application Programming Interface (WebAPI)	19
4.5.1	File upload / download	19
4.5.2	Remote Procedure Call (RPC)	19
4.5.3	Device configuration	21
4.6	Factory Settings and Reset	22
4.6.1	Factory Reset using Factory Reset Plug	22
4.7	Technical Support File	23
4.8	Status Indication	23
4.8.1	LED Indicators	23
4.8.2	Firmware Update	24
4.8.3	Factory Reset	24
4.9	Monitoring TRAPs	24

4.10 Alarm Handling	24
4.11 System Firmware	24
4.11.1 Upgrading System Firmware	25
4.12 Technical Preview	27
4.13 Certificate Store	27
4.13.1 Certificate Handling	28
4.13.2 Web Server (HTTPS)	29
4.13.3 802.1X	29
4.13.4 OpenVPN	31
5 Services	32
5.1 System	32
5.2 Network Configuration	33
5.2.1 Interfaces and Bridges	33
5.2.2 VLAN Interfaces	33
5.2.3 MAC VLAN Interfaces	34
5.2.4 OpenVPN Interfaces	34
5.2.5 Tunnel Endpoint Interfaces	34
5.2.6 IP Addresses	37
5.2.7 Network Configuration Examples	37
5.3 Ethernet Configuration	38
5.4 Wireless Configuration	38
5.4.1 WLAN Device physical radio configuration	39
5.4.2 WLAN Interface configuration	40
5.5 Cellular Network Configuration	43
5.5.1 Interface Configuration	43
5.5.2 SIM Slot Configuration	43
5.5.3 SIM Profile Configuration	43
5.5.4 Connection Management Configuration	44
5.5.5 Cellular Router as Gateway	45
5.5.6 Cellular Network Status	45
5.6 OpenVPN Configuration	45
5.6.1 OpenVPN Configuration Example	46
5.7 Wireless Network Access Point	48
5.7.1 Configuration File Example: Access point 2.4GHz	48
5.7.2 Configuration File Example: Access point 5GHz	49
5.8 Wireless Data Rate Control	49
5.8.1 Reduced Set of Bitrates	49
5.8.2 QMRR	49
5.9 802.11s Mesh	50
5.10 Bridge Mode (4addr)	50
5.11 Layer 2 NAT Mode	51
5.12 Wireless MAC Address Overwrite	52
5.13 Wireless Security	53
5.13.1 WPA Encryption	53
5.13.2 Port-based Network Access Control (802.1X)	54
5.13.3 Ciphers	58

5.13.4 Management frame protection (MFP, 802.11w)	60
5.14 IP Routing	61
5.14.1 Static Routing	61
5.15 VLAN	61
5.15.1 Multi SSID and VLAN	61
5.16 Mobility	63
5.16.1 Fast Association	63
5.16.2 Inter-AP Roaming	64
5.16.3 Handoff Filters	65
5.16.4 Mobility Logging	65
5.16.5 Fast BSS Transition (802.11r)	68
5.17 Quality of Service (QoS)	70
5.18 Common Address Redundancy Protocol (CARP)	74
5.19 Network Link Monitor (NLM)	74
5.19.1 NLM Monitor Types	75
5.19.2 phy Monitor	75
5.19.3 icmp Monitor	75
5.19.4 wlan Monitor	75
5.19.5 NLM Configuration for CARP Failover	76
5.19.6 NLM Configuration for Backbone Monitor	77
5.20 DNS/DHCP Server	77
5.21 Service Indicators and Counters	78
5.21.1 SNMP Trap	78
5.21.2 Counters and Status	78
5.22 Logging Features	78
5.23 Wireless Link Monitor	79
5.24 Inter-Carriage Link (ICL)	79
5.24.1 Configuration of the Inter-Carriage Link Application	80
5.25 Public Wireless Network (PWN)	85
5.25.1 Hotspot	85
5.25.2 Band Steering	86
5.26 Global Navigation Satellite System	86
5.26.1 GNSS Device Configuration	87
5.26.2 NMEA Sentences	87
5.26.3 UBX Messages	88
5.27 RSTP	88
5.28 Dynamic Frequency Selection (DFS)	88
5.28.1 Wireless Standalone	89
5.28.2 Wireless Manager (NWM)	89
5.28.3 Area Frequency Management (AFM)	92
5.29 Interference Detection Function (IDF)	97
5.30 Http Report	98
5.30.1 NWM and ChannelManager Report	98
5.30.2 IDF Report	100
5.31 Firewall	103
5.31.1 Network Address Translation (NAT)	103
5.31.2 Filter	106

5.31.3 L2 IP Filter Firewall	108
6 Country Codes	109
6.1 Configuration	109
6.2 Regions for 802.11n products	109
6.2.1 Country code WORLD	109
6.2.2 Region E	110
6.2.3 Region U	113
6.3 Regions for 802.11ac products	116
6.3.1 Region E	116
6.3.2 Region U	117
7 Security Considerations	121
7.1 Physical Interfaces	121
7.2 Network Concept	121
7.2.1 Local Administrative Access	121
7.2.2 Remote Administrative Access	122
7.3 Service Restrictions	123
7.3.1 CLI	123
7.3.2 SNMP	124
7.3.3 HTTP	124
7.4 Passwords	124
7.4.1 Strength Of PSK-Passphrase	125
8 Default settings	126
9 WESTERMO-SW6-MIB	127
9.0.1 configuration	127
9.0.2 rpc	294
9.0.3 settings	299
9.0.4 hardware	308
9.0.5 software	318
10 WESTERMO-SW6-BRIDGE-MIB	347
10.0.1 rstp	347
11 WESTERMO-SW6-FIREWALL-MIB	351
11.0.1 firewall	351
12 WESTERMO-SW6-GNSS-MIB	368
12.0.1 gnss	368
13 WESTERMO-SW6-ICL-MIB	376
13.0.1 icl	376
14 WESTERMO-SW6-NWM-MIB	381
14.0.1 nwm	381

15 WESTERMO-SW6-PWN-MIB	396
15.0.1 pwn	396
16 WebAPI Detailed Specification	398
16.1 Authentication API	398
16.1.1 RPC Methods	398
16.2 Files API	399
16.2.1 Device configuration file import / export	399
16.2.2 Syslog export	400
16.2.3 System messages export	400
16.2.4 Support File export	400
16.2.5 Firmware upload and upgrade	401
16.3 Device Configuration API	402
16.3.1 RPC Methods	402
17 Message Codes	405
18 CLI Commands	422
18.1 apply - Apply all pending configuration changes	422
18.2 changes - Show a list of changed configuration parameters	422
18.3 dmesg - Print the kernel ring buffer	422
18.4 get - Show the value of a configuration parameter	422
18.5 grep - Print lines matching a pattern	423
18.6 help - Show a list of all CLI commands	423
18.7 ip - Show or manipulate network devices	424
18.8 iperf - Perform network throughput tests	424
18.9 iw - Show or manipulate wireless devices	425
18.10logread - Show system log messages	425
18.11ping - Ping network hosts	426
18.12reset - Reset configuration parameters	426
18.13revert - Revert all pending changes	427
18.14set - Set the value of a configuration parameter	427
18.15ssh - A secure shell client	427
18.16cpdump - Dump traffic on a network	427
18.17watch - Execute a program periodically	428

1 Introduction

This document describes the functionality and features of the *Software 6*. The *Software 6* is the firmware controlling the operation of the *RT 11n* and *RT 11ac* family products.

The *RT 11n* and *RT 11ac* family products are wireless communication devices for demanding industrial applications. The *RT 11n* and *RT 11ac* family devices can operate at 2.4 and 5GHz WLAN bands depending on installation country limitations.

The devices can generally operate either as Access Point or Station. The operation is compatible with commercial IEEE 802.11 WLAN devices allowing co-existence with standard WLAN devices.

Software 6 delivers a complete set of functionality including:

- layer-2 basic switching, VLAN, etc.
- layer-3 routing, firewall, etc.
- higher-level services such as DHCP, DNS, Firewall, etc.

1.1 Supported Products

This document applies to the following product variants:

- RT-220
- RT-310
- RT-320
- Ibex-RT-330
- RT-370
- Ibex-RT-610
- Ibex-RT-630

1.2 Supported Features

Feature	RT-320	RT-220	RT-370	RT-310	RT-610	RT-330	RT-630
Web-Based Management (Web Interface)	yes	yes	yes	yes	yes	yes	yes
Simple Network Management Protocol (SNMP)	yes	yes	yes	yes	yes	yes	yes
Configuration Files	yes	yes	yes	yes	yes	yes	yes
Command Line Interface (CLI)	yes	yes	yes	yes	yes	yes	yes
Web Application Programming Interface (WebAPI)	yes	yes	yes	yes	yes	yes	yes
Factory Settings and Reset	yes	yes	yes	yes	yes	yes	yes
Technical Support File	yes	yes	yes	yes	yes	yes	yes
Status Indication	yes	yes	yes	yes	yes	yes	yes
Monitoring TRAPs	yes	yes	yes	yes	yes	yes	yes
Alarm Handling	yes	yes	yes	yes	yes	yes	yes
System Firmware	yes	yes	yes	yes	yes	yes	yes
Technical Preview	yes	yes	yes	yes	yes	yes	yes
Network Configuration	yes	yes	yes	yes	yes	yes	yes
Ethernet Configuration	yes	yes	yes	yes	yes	yes	yes
Wireless Configuration	yes	yes	yes	yes ¹	yes	no	yes
Wireless Network Access Point	yes	yes	yes	yes	yes	no	yes
Wireless Data Rate Control	yes	yes	yes	yes	no	no	yes
802.11s Mesh	yes	yes	yes	no	no	no	yes
Bridge Mode (4addr)	yes	yes	yes	no	yes	no	yes
Layer 2 NAT Mode	yes	yes	yes	no	yes	no	yes
Wireless MAC Address Overwrite	yes	yes	yes	yes	yes	no	yes
Wireless Security	yes	yes	yes	yes	yes	no	yes
IP Routing	yes	yes	yes	yes	yes	yes	yes
VLAN	yes	yes	yes	yes	yes	yes	yes
Mobility	yes	yes ²	yes	no	no	yes	yes
Quality of Service (QoS)	yes	yes	yes	yes	no	no	yes
Common Address Redundancy Protocol (CARP)	yes	yes	yes	yes	yes	no	yes
Network Link Monitor (NLM)	yes	yes	yes	yes	yes	no	yes
DNS/DHCP Server	yes	yes	yes	yes	yes	yes	yes
Service Indicators and Counters	yes	yes	yes	yes	yes	yes	yes
Logging Features	yes	yes	yes	yes	yes	yes	yes
Wireless Link Monitor	yes	yes	yes	yes	no	no	yes

¹Client (STA) mode not supported

²Only Fast Association is supported

Feature	RT-320	RT-220	RT-370	RT-310	RT-610	RT-330	RT-630
Inter-Carriage Link (ICL)	yes	yes	yes	no	no	no	yes
Public Wireless Network (PWN)	no	no	no	no	yes	no	no
RSTP	yes	yes	yes	yes	yes	no	yes
Wireless Standalone	yes	yes	yes	yes	yes	no	yes
Wireless Manager (NWM)	no	no	yes	no	no	no	no
Area Frequency Management (AFM)	no	no	yes	no	no	no	no
Interference Detection Function (IDF)	no	no	yes	no	no	no	no
Http Report	no	no	yes	no	no	no	no
Firewall	yes	yes	yes	yes	yes	yes	yes
OpenVPN Configuration	yes	yes	yes	yes	yes	yes	yes
Cellular Network Configuration	no	no	no	no	no	yes	yes
Global Navigation Satellite System	no	bo	no	no	no	yes	yes

1.3 Installation Country, Product Use

Installation country regulatory limits and operating parameters are controlled by the devices software driver. The Country Code limits are valid for client (STA) and Access Point operation modes.

All devices supports ETSI and FCC based country, among others:

Country Code	Frequency Range	Notes
Europe(EU)	2400-2483.5 MHz, 5150-5350 MHz and 5470-5725 MHz	Operation according to ETSI limitations.
United States(USA)	2400-2483.5 MHz, 5150-5350 MHz, 5470-5725 MHz and 5750-5850 MHz	Operation according to FCC limitations.

Each country has its own regulatory requirements which must be met to import the products. Please refer to section 6 for more information or contact your support for more information on regulatory requirements.

1.4 Delivery Content

The products are delivered without connection cables, and any plugs, adapters etc. The delivery includes one dust cap for one Ethernet interface.

1.4.1 Important Safety Notes



Danger! Do not use damaged equipment and/or accessories such as damaged power cord.



Danger! Never try to open the device. There are no serviceable parts inside! By trying to open the device you will be exposed to a risk of death or injury.



Warning! Never unplug equipment from the electrical outlet by holding the cord only, always disconnect the cable by applying force directly to the plug.

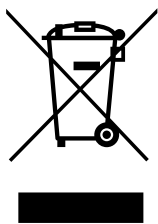


Warning! Do not operate the device in any other environmental conditions than it is designed for.



Warning! Before attaching the power cable to the device, please make sure you have antennas or terminators attached to the antenna connectors.

1.5 Information on Disposal of Old Electronic Equipment



This symbol on the product indicates that this product should not be treated as household waste when disposing it. Instead it shall be handed over to an applicable collection point for the recycling of electrical and electronic equipment. By ensuring this product is disposed correctly, you will help prevent potential negative consequences to the environment and human health, which could otherwise be caused by inappropriate disposal of this product.

1.6 License and Copyright

License and copyright for included Free/Libre Open Source Software

This product includes software developed by third parties, including Free/Libre Open Source Software (FLOSS). The specific license terms and copyright associated with the software are included in each software package respectively. Please contact your support for more information.

WESTERMO

Upon request, the applicable source code will be provided. A nominal fee may be charged to cover shipping and media. Please direct any source code request to your normal sales or support channel.

2 Installation

Installation step	Description
Fixing	Fix the products in their use environment, ensuring that the fixing environment complies with the installation environment constraints. Ensure correct system grounding based on customer's electrical installation.
Antennas	Install the antenna interfaces according to customer's requirements. Unused antenna ports shall be terminated.
Ethernet	Connect the Ethernet interfaces.
Power Feed	Connect the power cable first to the device and then to the power plug. Verify that the LED indicators shows correct power up procedure.
Configuration	Configure the device.

3 Quick Start

This section provides a simple guide to quickly get started with the device. Only the following simple configurations will be covered:

- Access Point setup
- Client (STA) setup

3.1 Starting the product for the first time



Warning! Before attaching the power cable to the device, please make sure you have antennas or terminators attached to the antenna connectors.

When booting the device for the first time, the modem will use the factory default configuration. With factory default configuration the device operates as an Access Point with layer-2 bridge, where the WLAN interface and the two Ethernet ports belong to the same bridge.

The default IP setting for the device is:

- IP address: 192.168.1.20
- Netmask: 255.255.255.0
- Default gateway: 0.0.0.0

Before you connect the modem with your LAN, you should change its IP setting according to your network topology.

3.2 Setting the IP address

The device can be configured either to use a static IP address, or to obtain a dynamic one via DHCP.

3.2.1 Setting the IP address via Web Interface

To configure the IP settings via Web Interface your PC needs to be located on the same IP subnet as the modem, i.e. the PC should be assigned an IP address on the 192.168.1.0/24 network. For example:

- PC IP address: 192.168.1.2
- PC Netmask: 255.255.255.0

Open your web browser and enter URL **http://192.168.1.20** in the browser's address field. You will be asked to enter a username and a password. Use the the factory default account settings shown below:

- Username: webadmin
- Password: admin

From the menu choose **Configuration - Advanced** entry. The current settings will be displayed. The **IP address** (appended with **Netmask** in CIDR notation) settings are under the "Network" topic, modify these to your liking. If you want to get the IP address automatically from the DHCP server change the **Protocol** to "dhcp".

You can set the **Default Gateway** under the topic "Routing". A value of "0.0.0.0" means that there is no Default Gateway in use.

To apply your changes click the "Apply" button.

Your settings will then be applied. Wait some time before accessing the Web interface from your newly set IP address.

3.3 Troubleshooting

The DC LED is dark:

The device is not powered. If you power the device using PoE (Power over Ethernet) please note that only connector X2 supports PoE.

The OPR LED is solid orange:

The device is not connected to an Access Point. If you expect connection please check your wireless configuration.

The ERR LED is solid orange:

Check the System Messages in the Web Interface under Status - View Log - System Messages. For more information on System Messages please refer to [Alarm Handling](#).

The ERR LED is solid red:

Critical hardware failure. The device must be repaired. For more information on System Messages please refer to [Alarm Handling](#).

The OPR LED is dark:

Critical hardware failure. The device must be repaired.

Unspecified problem:

- Check the System Messages in the Web Interface under Status - View Log - System Messages.
- Check the System Log (Syslog) in the Web Interface under Status - View Log - System Log.
- Download [Technical Support File](#) and send it together with the support request to your support.

4 Administration

The operation parameters in configuration let you choose the needed functionality. The configuration files are stored in the device. The devices are delivered with [Default settings](#). These settings define a set of parameters for typical device use.

Following maintenance and configuration interfaces are supported by the software:

1. **Web-based management:** Web interface offers fast access to basic functions and status information - see [Web-Based Management \(Web Interface\)](#).
2. **Simple Network Management Protocol :** is the main interface for maintenance and configuration. Access is done via remote host application - see [Simple Network Management Protocol \(SNMP\)](#).
3. **Configuration Files:** can be used to backup the configuration of a device. Later it can be used to restore the previous configuration - see [Configuration Files](#).
4. **Command Line Interface (CLI):** The CLI offers you the same scope of operation as with SNMP over Secure Shell (ssh) or Telnet - see [Command Line Interface \(CLI\)](#).
5. **Web Application Programming Interface (WebAPI):** The WebAPI offers you a programmable maintenance and configuration interface - see [Web Application Programming Interface \(WebAPI\)](#).



Important: During configuration phases write operations to the flash are performed. To avoid loss of data, it is necessary to avoid power loss during these procedures.



Important: If power is switched off during configuration phase, the device may fail to boot due to invalid configuration or it may fall back to the factory default configuration.

4.1 Web-Based Management (Web Interface)

Graphical user interface (HTTPS) can be accessed with web-browser via the IP address of the device:

- Default IP: 192.168.1.20
- Default user name: webadmin
- Default password: admin

4.2 Simple Network Management Protocol (SNMP)

The SNMP interface can be used for configuring the device and also for monitoring the device via Network Management System (NMS). The device includes an SNMP agent and a SNMP trap daemon. The SNMP agent is answering requests from the SNMP client. The SNMP trap daemon is responsible for sending events and exception conditions to the NMS.

As graphical SNMP tool we recommend iReasoning MIB browser. It is supported on many OS-platforms and it is easy to use. However, it is not free for professional use (see <http://www.ireasoning.com/mibbrowser.shtml> for details).

For command line and scripting purpose you can use Net-SNMP, which is open source and free to use (see <http://www.net-snmp.org/> for details).

SNMP exposes management data in the form of variables of the managed system. These variables describe the system configuration and status. They can be queried and changed by managing applications.

SNMP gives you many options for configuring and monitoring the device. To get access to the configuration and other parameters use the following settings for the SNMP client:

Parameter	Default
<code>cfgSnmpdVersion</code>	v2c
<code>cfgSnmpdComAdmin</code> (Read-Write Community)	admin-community

All configuration parameters of the device are defined and described in the Management Information Base (MIB). In case of the *Software 6* this is done through the [WESTERMO-SW6-MIB](#) file and others, which is part of the delivered software package.

Further, also hardware (serial number, etc.), firmware (version, etc.) and operation status (wireless,

network, etc.) can be retrieved from the device using SNMP.

4.2.1 Supported Commands

Command	Description
GET	For reading parameters
SET	For modification of parameters
TRAP	For notifications. TRAPs can only be sent if the address of the NMS is configured (includes NOTIFICATIONS, INFORM) NOTE: TRAP messages are sent using SNMP Version 2c.

4.2.2 Availability

Access	Description
Using SNMP in custom application	For most programming languages there exists an SNMP library which can be used. This enables the integration of dedicated SNMP requests into an application without the need for additional external tools.
Accessing the SNMP parameters via command line interface (CLI)	For developing purposes, the command line tools from the Net-SNMP project has been used.
Using commercial NMS	Integrate a MIB into a full size NMS, like HP OpenView, Castle Rock SNMPc.

4.2.3 Example: Change and permanently save a Parameter through SNMP command line tool

The following information must be available for SNMP tool to access the SNMP capable device:

- MIB information
- IP address of the device
- Admin Community string if SET operations are required (admin-community)

If a parameter is changed through SNMP, the value is only stored in volatile memory. To permanently store the parameter, the change must be applied. For an example see the following steps:

1. Change the parameter

```
snmpset -v 2c -c admin-community 192.168.11.238 WESTERMO-SW6-MIB::cfgSysHostname.0 s testname
```

2. Verify the change

```
snmpget -v 2c -c admin-community 192.168.11.238 WESTERMO-SW6-MIB::cfgSysHostname.0
```

3. Apply the change

```
snmpset -v 2c -c admin-community 192.168.11.238 WESTERMO-SW6-MIB::rpcCfgApply.0 i 1
```

With the [rpcCfgApply](#) command above the apply process is started. Depending on the changes this will take several seconds to finish. To observe the apply process the parameter [rpcCfgApply](#) should be polled with *snmpget* until a 0 is read. Please refer to [rpcCfgApply](#) for more information.

4.2.4 Supported MIBs

4.2.4.1 Device specific MIBs

- [WESTERMO-SW6-MIB](#)
- [WESTERMO-SW6-FIREWALL-MIB](#)
- [WESTERMO-SW6-ICL-MIB](#)
- [WESTERMO-SW6-NWM-MIB](#)
- [WESTERMO-SW6-BRIDGE-MIB](#)

4.2.4.2 Standard MIBs

Note: Not all entries in the standard MIBs are supported.

- OID root .1.0.8802.1.1.2 (LLDP-MIB)
- OID root .1.3.6.1.2.1 (MIB-2, RFC1213, HOST-RESOURCES, BRIDGE, ETHERLIKE, IF-MIB, etc.)
- OID root .1.3.6.1.4.1.2021 (UCD-SNMP-MIB)

4.3 Configuration Files

The configuration parameters of a device can be exported to or imported from a configuration file. This export or import can be done either by using an SNMP manager or simply by using the Web Interface. The exported configuration file is plain text, which allows to edit it using a simple text editor.

Since the parameter names and types used in the configuration file are the same as in the MIB database ([WESTERMO-SW6-MIB](#) and others), the documentation of each parameter can be found in the description of the MIB.

4.4 Command Line Interface (CLI)

The CLI is available either by Secure Shell (SSH) or Telnet and it makes the same scope of operation available as the SNMP interface. Additionally, it provides a selection of important maintenance tools. The `help` command offers explanations and examples for each tool.

Table 4.1 lists all commands and tools of the CLI.

Command	Short Description
<code>apply</code>	Apply all pending configuration changes
<code>changes</code>	Show a list of changed configuration parameters
<code>dmesg</code>	Print the kernel ring buffer
<code>get</code>	Show the value of a configuration parameter
<code>grep</code>	Print lines matching a pattern
<code>help</code>	Show a list of all CLI commands
<code>ip</code>	Show or manipulate network devices
<code>iperf</code>	Perform network throughput tests
<code>iw</code>	Show or manipulate wireless devices
<code>logread</code>	Show system log messages
<code>ping</code>	Ping network hosts
<code>reset</code>	Reset configuration parameters
<code>revert</code>	Revert all pending changes
<code>set</code>	Set the value of a configuration parameter
<code>ssh</code>	A secure shell client
<code>tcpdump</code>	Dump traffic on a network
<code>watch</code>	Execute a program periodically

Table 4.1: Overview of commands and tools of the CLI

The following examples illustrate how to access the CLI by using an SSH connection, use the `help` command and the configuration tools `get`, `set` and `apply`.

Suppose the CLI of a Software 6 device is accessible at 192.168.1.20 with the default username admin, use the following command to connect via SSH and enter the password when prompted.

```
$ ssh admin@192.168.1.20
admin@192.168.1.20's password:

      -----
      .'_ _ _ _ | | _ _ _ | | _ _ _ | Software 6 - Command Line Interface (CLI)
      / .' \ _ | | | | | |
      | | | | | | _ | |
      \ '._._.' \ _ | | _ / | _ | |
      '._._.' | | _ _ _ _ | | _ _ _ _
      | | _ _ _ _ | | _ _ _ _

/ #
```

When using the CLI for the first time, consider using the help command to get an overview of all supported tools, as shown in Table 4.1. To get an explanation and examples on a specific tool enter help and the name of the tool.

```
/ # help get
USAGE: get [MIB:]PARAMETER

Get the given configuration PARAMETER. If you want to get parameters which are
not part of the default MIB of the product, you must specify the MIB.

Be aware that the get command always returns the current value. This may differ
from the applied value. Use the changes command to review changed values.

EXAMPLES:
* Get the hostname:
    get cfgSysHostname.0
* Get the base MAC address of the IEEE 802.1d bridge:
    get BRIDGE-MIB::dot1dBaseBridgeAddress.0

/ #
```

The code extract below describes how configuration parameters can be changed and applied.

```
/ # get cfgSysHostname.0
Rmodem
/ # set cfgSysHostname.0 new-hostname
/ # apply
Please wait, applying a new configuration may take up to 60s ...
Apply done.
/ #
```

The CLI can be configured with the following parameters. For further information on CLI configuration, in particular on security considerations, refer to Section 7.3.1 on page 123.

Parameter	Default
cfgCliEnabled	enabled
cfgCliUsername	admin
cfgCliPassword	admin
cfgCliSshEnabled	enabled
cfgCliSshAddress	0.0.0.0:22
cfgCliTelnetEnabled	disabled
cfgCliTelnetAddress	0.0.0.0:23

4.5 Web Application Programming Interface (WebAPI)

The WebAPI is split in two main parts:

- File upload (import) / download (export) part
- Remote Procedure Call (RPC) part

For detailed information see [WebAPI Detailed Specification](#).

4.5.1 File upload / download

The file upload / download is implemented as simple http upload / download (multi-part) interface. The top URL is `/cgi-bin/luci/api/files`. Please refer to [WebAPI Detailed Specification](#) for examples.

4.5.2 Remote Procedure Call (RPC)

- The remote procedure calls are implemented as JSON RPC (version 2.0) API. The top URL is `'cgi-bin/luci/api'`.
- A JSON object with method name and parameters in a http(s) POST request sent to the device is answered with a JSON object in response. Please refer to [WebAPI Detailed Specification](#) for examples.
- For a detailed JSON-RPC 2.0 specification please see <http://www.jsonrpc.org/specification>.

4.5.2.1 Key elements

- JSON-RPC is a stateless, light-weight remote procedure call (RPC) protocol.
- An rpc call is represented by sending a Request object to a Server. The Request object has the following members:
 - `method`: Method is a string containing the name of the method to be invoked.
 - `params`: (Optional) Params is a structured value that holds the parameter values to be used during the invocation of the method.
 - `id`: Id is an identifier established by the client that must contain a String or a Number
- Response object
 - When a rpc call is made, the server is going to reply with a Response. The Response is expressed as a single JSON object with the following members:
 - `result`:
 - * The result is only present if the RPC call was successful
 - * The value of this member is determined by the method invoked
 - `error`:
 - * The error is only present if the RPC call in case of an error
 - * The value of this member is an Error Object
 - `id`:
 - * It is the same value as the value of the id member of the corresponding request object
 - * If there was an error in detecting the id in the Request object, then it will be null
- Error Object
 - The error object is an object with the following members:
 - * `code`: Code is a Number (integer) which indicates the error type that occurred
 - * `message`: Message is a String providing a short description of the error.

* data: (Optional) Data is a Primitive or Structured value that contains additional information about the error.

4.5.2.2 Error Codes

Code	Message	Description
-32700	Parse error	Invalid JSON was received by the server. An error occurred on the server while parsing the JSON text.
-32600	Invalid Request	The JSON sent is not a valid Request object.
-32601	Method not found	The method does not exist or is not available.
-32602	Invalid params	Invalid method parameter(s).
-32603	Internal error	Internal JSON-RPC error.
-32000	Error	General error code. Message and (optionally) data member will contain further information about the error.
-32001	Import error	File import error code. Message and (optionally) data member will contain further information about the error.
-32002	FW upgrade error	Firmware upgrade error code. Message and (optionally) data member will contain further information about the error.
-32003	FW upgrade running error	This denotes that a firmware upgrade is running and therefore no further WebAPI call is available.

4.5.2.3 Difference to JSON RPC specification

Following functions specified by JSON RPC are not supported:

- Notifications (methods without "id" member)
- Batch requests

4.5.3 Device configuration

There are two ways on how to configure the device using the Web API:

1. Using device configuration file import
2. Setting MIB elements

In both cases the process is as following:

1. Set configuration items (either by importing a configuration file or by setting MIB elements)

2. Apply the changes by setting MIB item [rpcCfgApply.0](#).
 - The rpc call starts the apply process, where the device checks if the configuration is valid and if so stores it in persistent memory
 - Please note that the rpc call returns immediately after the apply process has been started
3. Check if changes have been applied by polling MIB item [rpcCfgApply.0](#). Please refer to [rpcCfgApply](#) for details.

Please note that the process is similar as for when using SNMP interface or CLI.

4.6 Factory Settings and Reset

The *Software 6* is delivered with a default device configuration. A Factory Reset will reset the device configuration (including administrator password and https/802.1x certificates) to its default state ([Default settings](#)).

The following list shows the most important default configuration values which are needed to access the device. More default configuration values are listed in [Default settings](#).

Network

IP Address Mode static

IP Address 192.168.1.20, Factory IP address to access the device

IP Subnet Mask 255.255.255.0

Basic Configuration

SNMP Enabled

A Factory Reset can be issued by [rpcSysFactoryReset](#) or by using the Factory Reset Plug as described in the following.

4.6.1 Factory Reset using Factory Reset Plug

1. Power off the device
2. Plug-in the FACTORY RESET PLUG in one of the ethernet ports
3. Power on the device
4. Wait until Operation (orange) and Failure (red) LED are constantly on

5. Remove the FACTORY RESET PLUG within 15 seconds: Operation (orange) and Failure (red) LED will blink three times
6. Wait (approx. 1 minute) until device is rebooted, then you can access the device using the factory default IP

4.7 Technical Support File

The *Technical Support File* is a compressed tar file with collected system information, including log files and system configuration. It is accessible using the Web Interface. After logging in, go to *System* and then *Support*.

Download Technical Support File

Click this button to create and download the Technical Support File.

Warning: Passwords and other sensitive information may be included in the report.

The device checks during boot-up if any Kernel Log(s) are available. It will report this issue with a system message (see [Message Codes](#)). Before resetting the Kernel Log(s) you should first download the Technical Support file and send it to your support contact.

Reset Kernel Logs

This action is **irreversible** and will take approx. 20 seconds.

4.8 Status Indication

4.8.1 LED Indicators

The *RT 11n* and *RT 11ac* family products are equipped with five LED indicators:

DC	indicates whether or not power is connected to the device.
OPR	is flashing during boot up and afterwards shows a solid green when the device has booted. When wlan0 is configured in client (STA) mode, the LED shows a solid orange if not connected and solid green if connected.
ERR	indicates errors and warnings which require the intervention of the operator. When an error occurs, the Status LED show a solid red where in case of warning the LED is orange. In the error case the device has detected a serious hardware failure und must be repaired. In the warning case please check the Troubleshooting chapter of this manual.
X1/X2	show link status and activities on the respective ethernet interfaces.

4.8.2 Firmware Update

When a firmware update is in progress, the **OPR** LED is flashing orange and the **ERR** LED is flashing red at the same time until device reboot. **IMPORTANT:** One **MUST NOT** interrupt the power source until normal operation state is reached.

4.8.3 Factory Reset

The **OPR** LED is solid orange and **ERR** LED is solid red when a factory reset is detected. Both LEDs are blinking when the factory reset is in process. See also [Factory Settings and Reset](#).

4.9 Monitoring TRAPs

SNMP traps enable the device to notify the management station of significant events by way of an unsolicited SNMP message. The remote host can listen to SNMP traps in order to get notified about important events (like handoff notification) of the device. For a description how to use this feature see [SNMP Trap](#).

4.10 Alarm Handling

System warnings are indicated by the **ERR** LED (solid orange). System errors are indicated by the **ERR** LED (solid red). Warning and error messages are logged on the device and can be seen in the Web Interface (Status - View Log - System Messages). A detailed description of the single error codes can be found in [Message Codes](#).

For additional error analysis, the system log is available in the Web Interface (Status - View Log - System Log).

4.11 System Firmware

System firmware consists of two types of firmware images in a single binary: 1. bootloader image and 2. primary firmware image. The primary firmware image contains the main system software and the features described in this document. In rare cases it is required to update the bootloader, but this is handled by the system firmware update process.

4.11.1 Upgrading System Firmware

The system firmware can be upgraded via Web Interface, Web API, CLI or SNMP.



Warning! Ensure that the device is always powered during the firmware upgrade is in progress.



Warning! Although a system firmware upgrade with 'keep' configuration is supported, it is recommended to backup configuration before doing the system firmware upgrade.

The firmware file format differs between firmware version < 6.9 and ≥ 6.9 and needs to be considered:

- Firmware versions < 6.9 accepts only firmware files with extension ***.bin**
- Firmware versions ≥ 6.9 accepts only firmware files with extension ***.img**

Please contact your support if you need ***.bin** firmware files to downgrade from ≥ 6.9 to a firmware version < 6.9 or if you need ***.img** to upgrade from a firmware version < 6.9 to ≥ 6.9 .

4.11.1.1 Upgrading Firmware via Web Interface

In order to upgrade the firmware via Web Interface, your PC needs to be located in the same IP subnet as the device. When the device is using the default IP address (192.168.1.20), then the connected PC might be configured as follows:

- PC IP address: 192.168.1.1

- PC Netmask: 255.255.255.0

Open your web browser and enter the IP address of the device (default **https://192.168.1.20**) in the browser's address field. You will be asked to enter a username (default: 'webadmin') and a password (default: 'admin'). Then click on **Login** button.

Once logged in choose menu entry **System - Maintenance**. In section **Flash Firmware** you can browse your PC file system for the firmware image by clicking the **Browse...** button.

Concerning device configuration, the following three options are available:

1. **Reset to default configuration:** Device configuration will be reset to its default values.
2. **Keep the current configuration:** Device configuration will not be changed. If the new firmware image provides new configuration items, these will be initialized with their default values.
3. **Apply custom config after upgrade:** See more detailed description below.

After selecting the image file (and the custom configuration file if you have chosen **Apply custom config after upgrade**), proceed to uploading and flashing by clicking the **Flash Firmware** button. A verify page will be displayed with checksum and size information. Click **Proceed** in order to finally start the firmware upgrade process on the device.

When the **Apply custom config after upgrade** is selected, then the following steps will be processed:

1. Firmware image will be uploaded to the device.
2. Device will be upgraded using new firmware image.
3. Device configuration will be reset to new firmware image default values.
4. Provided custom configuration will then be applied. In order to assert that this custom configuration is valid for the new firmware image, it should best have been exported from a device already using the new firmware image.

4.11.1.2 Upgrading Firmware via CLI/SNMP (using TFTP server)

1. Define a valid URL which is accessible by the device. This is done by changing the firmware URL parameter `setFwFileUrl` in the settings section.
2. Optional: If you want to reset the device to factory settings, set `setFwKeepConfig` to 'reset(0)'. If you want to use a *custom config* (see paragraph above), define a valid URL in `setCfgFileUrl`.

3. Writing 'flash(2)' to `rpcFwFlash` parameter will download and validate the new firmware file. Writing 'download(1)' to `rpcFwFlash` parameter will only download and validate the new firmware file, but will not flash to the file system. Writing 'flashWithConfig(3)' to `rpcFwFlash` parameter will download and validate the new firmware file and the custom config file.
4. If the downloaded file is considered a valid firmware for this device, it will be flashed to its file system.
5. Reading `rpcFwFlash` parameter will report the status of the firmware flash process. A value of 'flash_error(-2)' denotes that the flash process failed during write of the firmware file to the file system. A return value of 'download_error(-1)' indicates an error while the firmware was downloaded or validated. A value of 'flash(2)' means that the device is writing the firmware to the file system.

4.12 Technical Preview

The Technical Preview allows access to features that are not yet officially released. It can be enabled or disabled on the **System - Support** page of the Web Interface or by using `setTechPreviewEnabled` via SNMP.

If the Technical Preview is activated additional configuration parameters are available on the **Configuration - Advanced** page.



Important: Configuration parameters provided by the Technical Preview

- are only exported if it is enabled,
- only persist after a firmware upgrade if it is enabled, and
- are subject to future changes or may be removed completely.

4.13 Certificate Store

The *Certificate Store (CS)* is responsible to handle all Certificates (CRT), Private Keys (KEY) and Certificate Revocation Lists (CRL) used by services on the device. These services are:

- Webserver (HTTPS)
- 802.1X
- OpenVPN

The files are stored in non-volatile memory. A factory reset or a firmware upgrade without *keep config* deletes all files.

All CRT, KEY and CRL files may be imported and exported in PEM or DER format.

To import an encrypted key the respective password has to be set before importing the key.

The import process validates the key, and verifies that the configured password is correct. A key file which had no password before the import, will be encrypted with the configured password. Key files are always stored in encrypted format. Thus an exported keyfile is encrypted.

After a new file has been installed, the appropriate service needs to be restarted.

4.13.1 Certificate Handling

SNMP

Files may be imported, exported or deleted by the following process:

1. Define the URL of where the file is located with [setCertFileUrl](#)
2. Select the file to be manipulated with [setCertFileSelector](#)
3. Select the file format by setting [setCertFileFormat](#). For PKCS#12 containers this setting is ignored.
4. Start the process by setting [rpcCertFile](#) to the desired action (import(1), export(2) or delete(3))
5. Observe the process progress by polling [rpcCertFile](#).

When working with encrypted key files, the process is slightly different. The password has to be set before the above flow is applicable.

An example flow for OpenVPN via SNMP is:

1. Set the key password with [cfgVpnOpenvpnKeyPassword](#)
2. Ensure the instance is not enabled [cfgNetOpenvpnEnabled](#)
3. Apply the config.
4. Follow the above instructions to import the key via SNMP.
5. After the key has been imported, configure and enable the instance.

An example flow for OpenVPN via Web Interface is:

1. Navigate to Configuration -> Advanced -> Network -> OpenVPN Interface.
2. Create an OpenVPN instance by clicking the "Add Row" button.
3. Ensure that the instance is not enabled and set the "Bridge" to -2.
4. Navigate to the VPN tab and set the "Key Password".
5. Press the apply button.
6. Follow the above instructions to import the key via SNMP.
7. After the key has been imported, configure and enable the instance.

4.13.2 Web Server (HTTPS)

The Web Server for the *Web Interfaces* uses TLS by default. Therefore the following files are required:

- HTTPS Certificate
- HTTPS Key
 - Importing an encrypted key is not supported.

4.13.3 802.1X

The following files are supported for 802.1X (WAP-EAP encryption):

- **Client Certificate (CRT):** Client x509 Certificate
- **Client Key (KEY):** Client Private Key matching the Client x509 Certificate
 - For encrypted keys set [cfgWlan802dot1xClientKeyPassword](#) before import
- **Root Certificate Authority (CA):** Authentication Server X509 Certificate
- **Subordinate Certificate Authority (CA2):** Subordinate / Intermediate Authentication Server X509 Certificate

- **Root Certificate Revocation List (CRL):** CRL issued by the Root CA
- **Subordinate Certificate Revocation List (CRL2):** CRL issued by the Subordinate / Intermediate CA
- **PKCS#12 Container:** Container file which include the CRT, KEY and maybe CA/CA2
 - For an encrypted key file set [setCrtFilePkcs12Passphrase](#) before import

Important notes:

- Only the Client Certificate and Client Key will be imported from a PKCS#12 Container.

Subordinate CA (CA2) and Subordinate CRL (CRL2) are optional. When using CA2, then the following combination of CRLs are supported:

Case	CRL	CRL2	Description
1	-	-	ok - no CRL checking
2	-	x	invalid/not supported; do not use!
3	x	-	ok - revoked Client/Server Certificates are rejected
4	x	x	ok - revoked CA2 and revoked Client/Server Certificates are rejected

Table 4.2: *Supported CRL/CRL2 combinations*

- "-" meaning CRL/CRL2 is not available, "x" meaning CRL/CRL2 is available.
- Case 2 (i.e. having CRL2 but no CRL) is not supported! In this case, CRL2 will be ignored!
- Case 3 is indirectly supported: When using a CRL and an empty CRL2 (which makes case 3 to case 4).

4.13.3.1 Certificate Refresh

The CS provides two storage locations:

1. The *current* is the actually used certificate.
2. The *next* is the new certificate installed by a refresh.

The desired storage location may be chosen during import (see [Certificate Handling](#)).

A refresh of the files may be triggered by calling [rpcCrtRefresh](#) or by rebooting the device. Refreshing the files does the following steps:

1. The *next* Certificate Bundle (see [Certificate Bundle](#)) in the CS will be checked for consistency. If found to be consistent:
2. Certificates in CS storage location *next* are moved to CS storage location *current*.
3. WLAN interfaces will be restarted.

4.13.3.2 Certificate Bundle

A *Certificate Bundle* contains at least a CA Certificate, a Client Key and a Client Certificate. In such a bundle the file in the storage location *next* will be used if available, otherwise the one from the storage location *current* will be used.

The following example illustrates how the *next* files are handled:

Certificate	<i>current</i>	<i>next</i>	use
CA	x		<i>current</i>
CA2	x	x	<i>next</i>
Client	x	x	<i>next</i>
Client Key	x	x	<i>next</i>
CA CRL	x		<i>current</i>
CA2 CRL	x	x	<i>next</i>

Table 4.3: *Certificate Bundle Example*

Therefore the Certificate Bundle for the example in table 4.3 is: CA "current", CA2 "next", Client "next", Client Key "next", CA CRL "current" and CA2 CRL "next".

4.13.4 OpenVPN

The following files are available for the OpenVPN instances:

- Client Certificate
- Client Key
 - For encrypted keys set `cfgVpnOpenvpnKeyPassword` before import
- CA Certificate
- Static Key

5 Services

5.1 System

The hostname of the the device is specified with `cfgSysHostname`. Together with `cfgSysDomain` it defines the Fully Qualified Domain Name (FQDN).

Configure the domain to which the device belongs with `cfgSysDomain`. It is by default set to `none` which means the device does not belong to any domain. The device can resolve names of other devices which are in the same domain as itself without specifying the FQDN but only the hostname. DHCP-clients which get a domain provided by DHCP-option 15 (Domain Name) will never override the value provided here, not even when it is set to `none`. Domain names provided via DHCP-option 15 will instead be added to the search domain list configurable via `cfgSysSdSearch`. Refrain from setting this to `local` as it will interfere with Multicast DNS (mDNS) operation and lead to problems. See RFC2606 (<https://www.ietf.org/rfc/rfc2606.txt>) and RFC6762 appendix G (<https://tools.ietf.org/html/rfc6762#appendix-G>).

One can specify up to 6 domains in the `cfgSysSearchdomainTable`. These will be searched when looking up a hostname. Be aware that entries which are dynamically added via DHCP-option 15 (Domain Name) and via DHCP-option 119 (Domain Search) count against this limit as well.

The `cfgSysNameserverTable` allows to specify a prioritised list of nameservers with which the device may resolve hostnames. There are 2 types of entries:

- static entries
- dynamic entries

The order of `cfgSysNameserverTable` defines the priority of the nameserver. The lower the index, the higher the priority. Entries are tried sequentially with a timeout of 3 seconds.

Static entries allow to specify a static server IP as nameserver in `cfgSysNsServer`.

Dynamic entries allow to reference a network interface, by setting `cfgSysNsDhcpInterface`, on which a DHCP-client is running. The cellular network and OpenVPN interfaces may be referenced as well, since they may provide dynamic DNS information.

Nameserver entries from DHCP-clients that are not explicitly listed are put to the end of the list.

5.2 Network Configuration

5.2.1 Interfaces and Bridges

Software 6 supports several kind of interfaces (Ethernet, WLAN, VLAN). These interfaces can be assigned to a bridge, but they do not need to be on a bridge. An interface is assigned to a bridge by the Bridge configuration parameter. For example an Ethernet interface is assigned to a bridge by [cfgNetEthBridge](#), a WLAN interface by [cfgNetWlanBridge](#) and a VLAN interface by [cfgNetVlanBridge](#). Interfaces with the same bridge index are assigned to the same bridge.

The following example shows how to assign two interfaces (eth0 and wlan0) to bridge with index 0 and one interface (eth1) which is not assigned to a bridge:

Configuration File Example: Interface configuration

```
# Config.Format = raw
# eth0 to bridge 0
WESTERMO-SW6-MIB::cfgNetEthBridge.0 = 0
# eth1 not in a bridge
WESTERMO-SW6-MIB::cfgNetEthBridge.1 = -1
# wlan0 to bridge 0
WESTERMO-SW6-MIB::cfgNetWlanBridge.0 = 0
```

Bridges with an index ≥ 100 are special bridges which forward link local traffic. This can be used for wireless links in 4addr mode which should act as a cable-replacement. Note: Such a bridge may only contain 2 interfaces!

The network interface configuration parameters are described in [cfgNetEthernetTable](#) and [cfgNetWlanTable](#).

5.2.2 VLAN Interfaces

VLAN interfaces can be created on top of Ethernet and WLAN interfaces by [cfgNetVlanParent](#) directly or the more common use case is to create VLAN interfaces assigned to a bridge by [cfgNetVlanBridge](#). Untagged frames are internally handled by using VID 0.

The following example shows the how to create a VLAN assigned to bridge with index 0 and a VLAN on top of an interface (eth0):

Configuration File Example: VLAN configuration

```
# Config.Format = raw
# VLAN interface with VID 22 assigned to bridge 0
WESTERMO-SW6-MIB::cfgNetVlanEnabled.0 = 1
```

```
WESTERMO-SW6-MIB::cfgNetVlanBridge.0 = 0
WESTERMO-SW6-MIB::cfgNetVlanVid.0 = 22
# VLAN interface with VID 33 on top of eth0
WESTERMO-SW6-MIB::cfgNetVlanEnabled.0 = 1
WESTERMO-SW6-MIB::cfgNetVlanBridge.0 = -1
WESTERMO-SW6-MIB::cfgNetVlanParent.0 = eth0
WESTERMO-SW6-MIB::cfgNetVlanVid.0 = 33
```

The VLAN configuration parameters are described in [cfgNetVlanTable](#).

5.2.3 MAC VLAN Interfaces

The MAC VLAN configuration parameters are described in [cfgNetMacVlanTable](#).

5.2.4 OpenVPN Interfaces

Each OpenVPN interface in the [cfgNetOpenvpnTable](#) is managed by a separate instance of OpenVPN. These instances are created and enabled via the [cfgVpnOpenvpnTable](#). The indices of both tables refer to the same instance, see [OpenVPN Configuration](#).

5.2.5 Tunnel Endpoint Interfaces

Tunnel Endpoint (TEP) Interfaces are defined in the [cfgNetTunnelEndPointTable](#). Tunnels enable stateless encapsulation of IP or Ethernet frames.

Supported protocols are:

Protocol	Encapsulation Overhead	Bridgeable
GRE	24 bytes	no
GRETAP	38 bytes	yes

5.2.5.1 Generic Routing Encapsulation (GRE)

The GRE protocol allows running IPv4 and IPv6 frames over IPv4.

Every inner IP frame is prepended with a GRE header and then encapsulated within an outer IPv4 frame. The resulting overhead is 24 bytes (4 bytes GRE header + 20 bytes outer IPv4 header).

GRE is often used in conjunction with IPsec. GRE is not encrypted and IPsec does not allow multicast traffic. Thus the combination of both result in an encrypted IPsec tunnel with a multicast capable GRE tunnel on top.

The example in Figure 5.1 showcases a simple scenario where the WLAN Router creates a routed tunnel to the Backbone Router. The traffic between them traverses multiple routed networks (the Wireless Network and the Routed Backbone).

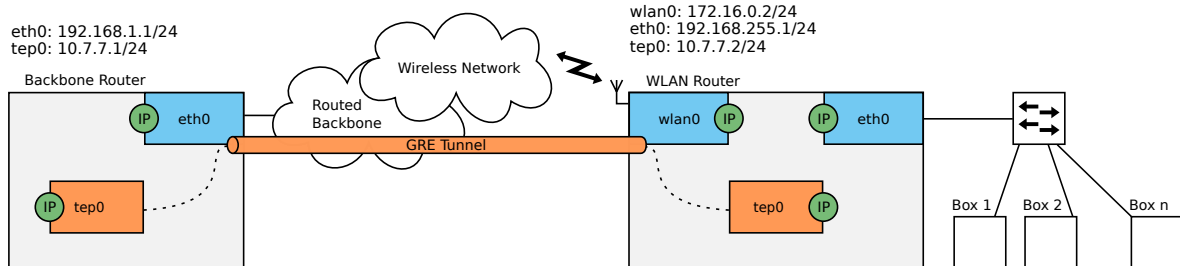


Figure 5.1: GRE in a Wireless Network

The relevant configuration items for the WLAN Router in Figure 5.1 to create a GRE tunnel, are described in the Listing 5.1:

```
WESTERMO-SW6-MIB::cfgNetTepEnabled.0 = 1
WESTERMO-SW6-MIB::cfgNetTepBridge.0 = -1
WESTERMO-SW6-MIB::cfgNetTepTunnelType.0 = 0
WESTERMO-SW6-MIB::cfgNetTepSource.0 = 0.0.0.0
WESTERMO-SW6-MIB::cfgNetTepDestination.0 = 192.168.1.1

WESTERMO-SW6-MIB::cfgNetIpEnabled.3 = 1
WESTERMO-SW6-MIB::cfgNetIpAddr.3 = 10.7.7.2/24
WESTERMO-SW6-MIB::cfgNetIpInterface.3 = tep0
```

Listing 5.1: TEP Interface Configuration for GRE

Because a TEP interface of type GRE may not be bridged, `cfgNetTepBridge` is set to -1. The source address `cfgNetTepSource` is set to 0.0.0.0, this lets the routing process decide which source address is used for the tunnel.

5.2.5.2 Generic Routing Encapsulation Tap (GRETAP)

The GRE protocol may operate in TAP mode, allowing it to run Ethernet frames over IPv4.

Every inner Ethernet frame is prepended with a GRE header and then encapsulated within an outer IPv4 frame. The resulting overhead is 38 bytes (4 bytes GRE header + 14 bytes inner Ethernet header + 20 bytes outer IPv4 header).

This allows bridging of TEP interfaces and enables usage of VLANs over the tunnel.

The scenario in Figure 5.2 showcases the WLAN Router which creates a bridged tunnel to the Backbone Router. The traffic between them traverses multiple routed networks (the Wireless Network and the Routed Backbone). The bridged devices connected to the WLAN Router are in the same subnet as tep0 of the Backbone Router.

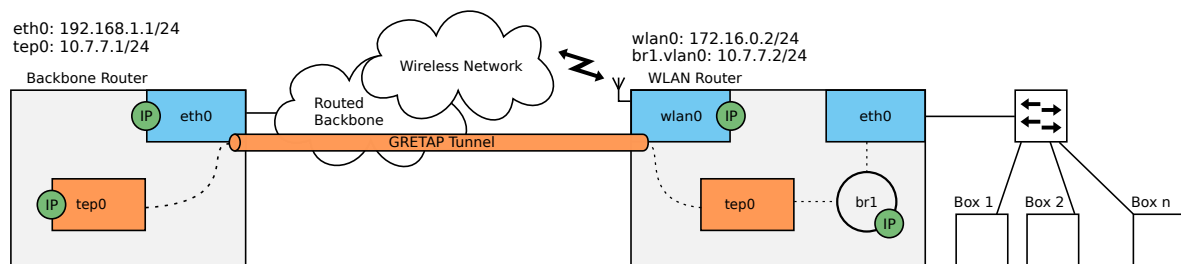


Figure 5.2: GRE-TAP in a Wireless Network

The essential configuration items for the WLAN Router in Figure 5.2 to create a GRE-TAP tunnel, are described in the Listing 5.2:

```
WESTERMO-SW6-MIB::cfgNetTepEnabled.0 = 1
WESTERMO-SW6-MIB::cfgNetTepBridge.0 = 1
WESTERMO-SW6-MIB::cfgNetTepTunnelType.0 = 1
WESTERMO-SW6-MIB::cfgNetTepSource.0 = 172.16.0.2
WESTERMO-SW6-MIB::cfgNetTepDestination.0 = 192.168.1.1

WESTERMO-SW6-MIB::cfgNetEthBridge.0 = 1

WESTERMO-SW6-MIB::cfgNetVlanEnabled.2 = 1
WESTERMO-SW6-MIB::cfgNetVlanBridge.2 = 1

WESTERMO-SW6-MIB::cfgNetIpEnabled.3 = 1
WESTERMO-SW6-MIB::cfgNetIpAddr.3 = 10.7.7.2/24
WESTERMO-SW6-MIB::cfgNetIpInterface.3 = br1.vlan0
```

Listing 5.2: TEP Interface Configuration for GRE-TAP

To bridge eth0 together with tep0 on br1, set both `cfgNetTepBridge` and `cfgNetEthBridge` to 1. To allow creation of an IP address on br1, a VLAN interface br1.vlan0 is added to br1. The source address `cfgNetTepSource` is set to 172.16.0.2, binding the source to a specific address.

5.2.6 IP Addresses

The IP configuration is separated from the interface configuration, so that there is a single way of how to assign IP addresses to interfaces. Multiple IP addresses can be assigned to the interfaces. As long as an interface is not part of a bridge, you can add an IP address directly to this (Ethernet or WLAN) interface. To assign an IP address to a bridge, one has to create a VLAN interface on that bridge. All IP addresses are configured using Classless Inter-Domain Routing (CIDR) notation (e.g 192.168.1.20/24).

The following example shows how to assign an IP address an interface (eth0) and another to VLAN 22 in bridge with index 0 (br0.vlan22):

```

Configuration File Example: IP address configuration

# Config.Format = raw
# Static IP address on eth0
WESTERMO-SW6-MIB::cfgNetIpEnabled.0 = 1
WESTERMO-SW6-MIB::cfgNetIpProto.0 = 0
WESTERMO-SW6-MIB::cfgNetIpAddr.0 = 192.168.1.22/24
WESTERMO-SW6-MIB::cfgNetIpInterface.0 = eth0
# Static IP address on br0.vlan22 (VLAN 22 in bridge with index 0)
WESTERMO-SW6-MIB::cfgNetIpEnabled.0 = 1
WESTERMO-SW6-MIB::cfgNetIpProto.0 = 0
WESTERMO-SW6-MIB::cfgNetIpAddr.0 = 192.168.2.22/24
WESTERMO-SW6-MIB::cfgNetIpInterface.0 = br0.vlan22
    
```

The IP configuration parameters are described in [cfgNetIpTable](#).

5.2.7 Network Configuration Examples

Example 1: Access Point Network Configuration

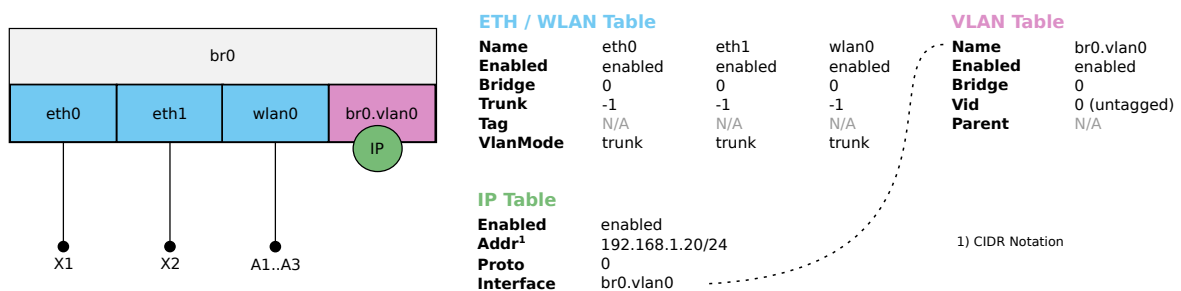


Figure 5.3: Access Point Network Configuration Example

Example 2: Client (STA) Network Configuration

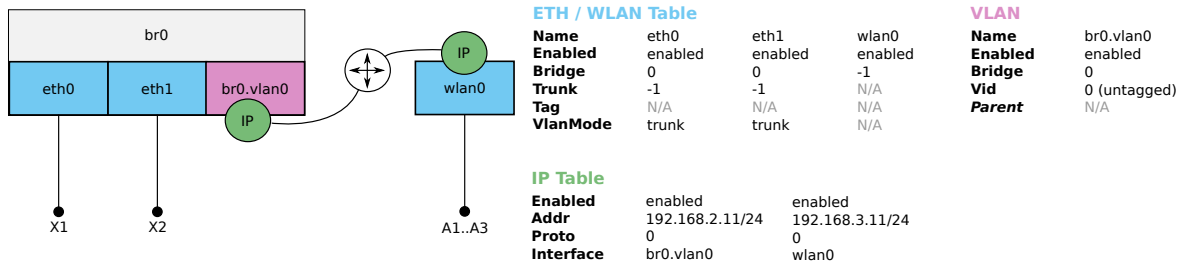


Figure 5.4: Client (STA) Network Configuration Example

Example 3: VLAN Network Configuration

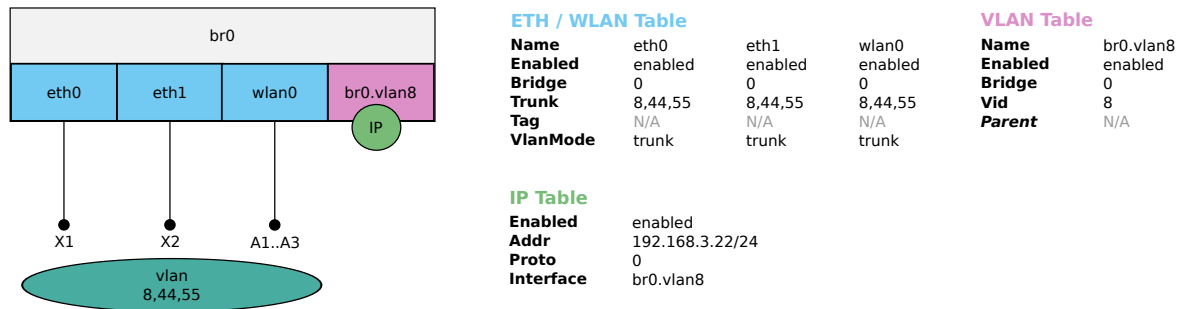


Figure 5.5: VLAN Network Configuration Example

5.3 Ethernet Configuration

All Ethernet and one WLAN port on the device are enabled and bridged by default. Ethernet interfaces auto-negotiate speed (10/100/1000 Mbit/s) and duplex mode (half/full) to the best common mode when a physical link is established.

It is possible to disable auto-negotiation of speed and duplex mode by [cfgNetEthAutoneg](#) and [cfgNetEthSpeed](#).

5.4 Wireless Configuration

The wireless configuration can be divided into three levels:

- **Network WLAN ([cfgNetWlanTable](#))** - such as bridge, VLAN mode, etc.
- **WLAN Device - physical ([cfgWlanDeviceTable](#))** - such as frequency and output power
- **WLAN Interface - logical ([cfgWlanInterfaceTable](#))** - such as operating mode, SSID, encryption and allowed bitrates, up to 16 logical WLAN interfaces can be configured

5.4.1 WLAN Device physical radio configuration

The parameters in [cfgWlanDeviceTable](#) define the physical level configuration parameters for each WLAN radio available in the product. The number of available physical radios is depending on the product variant, for example:

- RT-220, RT-310 and RT-320 include one radio: radio0 (for communication)
- RT-370 includes two radios: radio0 (for communication), radio1 (monitoring only)
- RT-610 includes two radios: radio0 (communication in 5 GHz band), radio1 (communication in 2.4 GHz or 5 GHz band)

Physical device configurations of a radio are common to all logical/virtual WLAN Interfaces created on top of it.

The most important configurations parameters are explained in the following paragraphs.

5.4.1.0.1 Bandwidth sets the Wireless Bandwidth Mode by specifying the bandwidth of the channel. Please refer to [cfgWlanDevBandwidth](#) for more information.

5.4.1.0.2 Modulation sets the modulation mode of the physical wireless radio device. Please refer to [cfgWlanDevModulation](#) for more information.

Legacy modes (a or g) are recommended and only usable in application with low throughput requirements.

5.4.1.0.3 Output Power and Antenna Gain are defining the resulting output power of the physical radio device. [cfgWlanDevPower](#) can be used to limit the combined EIRP power of all active TX antenna chains ([cfgWlanDevTxAntenna](#)) in dBm including the antenna gain ([cfgWlanDevAntennaGain](#)). The antenna gain is the value of the installed antenna in dBi. If multiple antennas with different gains are connected, the value of the antenna with the highest gain shall be configured.

Note that the maximum combined EIRP is limited by local regulations for each country. Please go to the page *Application -> Regulatory Domain Manager* in the Web Interface. (HTTP) to get more information about the maximum achievable RF output power of your device.

5.4.1.0.4 Distance sets the maximum distance ([cfgWlanDevDistance](#)) in meters a client can be away from the access point. Even though the distance is set in meters, the slot time settings change in 450m steps. In order to maximize the MAC layer efficiency it is recommended to keep this setting as low as possible.

5.4.1.0.5 Transmission Retries can be set with [cfgWlanDevShortRetry](#) and [cfgWlanDevLongRetry](#). The short retry defines the number of retransmissions of the RTS frame if there is no CTS received from the AP, whereas the long retry defines the number of retransmissions for unicast data frames not ACKed by the receiver.

In wireless environments retries on physical level are inevitable. Generally it is recommended to set lower values for UDP traffic and higher values for TCP. If the used channel is overloaded, excessive retransmissions can introduce a negative feedback-loop reducing available bandwidth even further.

5.4.1.0.6 TX and RX Antennas are configured in with [cfgWlanDevTxAntenna](#) and [cfgWlanDevRxAntenna](#). The configuration is a bitmask to enable/disable the chains. Depending on the product variant, there are either two, three or four chains available.

The number of enabled chains should match the number of antennas used.

Note, that for example with 11n modulation and with one chain enabled only, the 1-stream bitrates ([cfgWlanInterfaceBitrates](#)) MCS0 to MCS7 can be used. With two chains enabled, the 1- and 2-stream bitrates ([cfgWlanInterfaceBitrates](#)) MCS0 to MCS15 can be used.

Note, that enabling additional antennas will automatically reduce the output power of each chain as explained in Paragraph [Output Power and Antenna Gain](#).

Also note that if only one antenna is used, the used antenna must be the Antenna 1 (chain 0 i.e. 001).

5.4.2 WLAN Interface configuration

Up to 16 logical interfaces can be configured and enabled simultaneously in [cfgWlanInterfaceTable](#). This table configures the logical WLAN Interfaces which can be added on top of the physical WLAN device radios which were explained in [WLAN Device physical radio configuration](#).

The most important logical WLAN interface configuration parameters are explained in the following paragraphs.

5.4.2.0.1 Operating Mode of the WLAN interface can be set with [cfgWlanInterfaceMode](#). The supported modes are AP, STA, MONITOR and MESH.

AP is the standard Access Point / Infrastructure mode. This mode is typically used at the infrastructure side in stationary installations. One access point supports up to 200 simultaneous client connections. The AP interface is usually bridged to the Ethernet or to a VLAN directly.

STA is the station/client mode. This mode is typically used at the mobile part of the system. Stations provide a router functionality between the WLAN and Ethernet interface and therefore usually keep the bridging disabled.

The MONITOR mode allows use the interface in a fully passive monitoring mode. It is used to listen to WLAN traffic or scan for wireless interference and off-channel radar signals. Please consult the support for details, if you want to use this mode.

The MESH mode allows to connect multiple devices without the need of an Access Point. Refer to [802.11s Mesh](#) for more information.

5.4.2.0.2 Service Set Identifier (SSID) of the wireless interface is set with [cfgWlanInterfaceSsid](#). SSID is the arbitrary name of the wireless network this interface is part of.

5.4.2.0.3 Encryption can be set with [cfgWlanInterfaceEncryption](#). Six encryption modes are currently supported: open(0), psk(3), eap(6), sae(7), owe(8) and saepsk(9). It is highly recommended that at least WPA2 encryption is used. The password for the encryption can be set with [cfgWlanInterfacePassword](#). Refer to [Wireless Security](#) for more information.

5.4.2.0.4 Bitrate Limitations can be set with [cfgWlanInterfaceBitrates](#). This setting can be used to set fixed MCS index or range for 802.11n rates. Set to -1 to disable (leave on auto). It is also possible to enter multiple indices divided by a space which are then used in auto rate. This entry is only active when an n-mode is set in [cfgWlanDevModulation](#), and is ignored if g-rates or a-rates are used.

5.4.2.0.5 Enabling Wireless Multimedia Extensions (WME) for the interface is possible with [cfgWlanInterfaceWmeEnabled](#). When using legacy rates (a-rates and g-rates), this is optional. When using n-rates, this always has to be enabled. WME uses all parameters in the [cfgWlanWmeTable](#) whose [cfgWlanWmeId](#) is set to the value in [cfgWlanInterfaceWmeParameter](#). For more information about WME tables please check [Quality of Service \(QoS\)](#).

5.4.2.0.6 MAC Address Access Control List (ACL) mode for the interface can be set in [cfgWlanInterfaceMacaddrAcl](#). The available modes are:

- 0: Accept unless deny filter - accepts every MAC unless it is on the list defined in [cfgWlanAclBlackTable](#).
- 1: Deny unless accept filter - denies every MAC unless it is on the list defined in [cfgWlanAclWhiteTable](#).
- 2: Use RADIUS to accept/deny clients.

ACL can only be set if the operating mode of the interface is set to AP.

5.4.2.0.7 Scan Channels in station mode are configured with the [cfgWlanFreqTable](#) frequency lists. Each of those lists can hold up to 24 entries. Unused entries at the end of the list must be set to 0. In order to assign a frequency list to a WLAN interface, the list index is set in [cfgWlanInterfaceScanList](#). As factory default the index 0 ([cfgWlanFreqTable.0](#)) includes all 2.4Ghz center frequencies and the index 1 ([cfgWlanFreqTable.1](#)) includes all the 5Ghz center frequencies. In order to scan full country code all frequencies of the frequency list must be set to 0.

5.4.2.1 SSID Hide Feature

Service Set Identifier (SSID) specifies the name of a WLAN network. When people want to connect to a WLAN network, they normally check which WLAN networks are available in their neighbourhood, and then they choose the one they want to connect to.

If an Access Point provider does not want that random persons connect to their network, they will normally configure the Access Point so that the SSID is hidden. Then people not knowing the name of the WLAN network (i.e. its SSID) will not try to connect to this WLAN network by accident; but only people who know the name of the WLAN network will connect.

There are two MIB elements available to configure the SSID Hide Feature:

- [cfgWlanInterfaceIgnoreBroadcastSsid](#)
- [cfgWlanInterfaceUseVendorSsid](#)

Setting [cfgWlanInterfaceIgnoreBroadcastSsid](#) to enabled(1) specifies that the SSID will not be announced in the beacon of the Access Point (AP). Additionally, probe request frames that do not specify the full SSID will not be answered by this AP. As a consequence, Clients/Stations need to know the SSID to be able to connect to this WLAN network.

When [cfgWlanInterfaceIgnoreBroadcastSsid](#) is enabled, a passively scanning Client/Station (forced or because of DFS) have no way of detecting the AP it tries to find. For being able to connect to the

WLAN network in these two cases, [cfgWlanInterfaceUseVendorSsid](#) must be used:

- On an AP: set [cfgWlanInterfaceUseVendorSsid](#) to enabled(1) so that the hidden SSID is added as vendor element in the AP beacon.
- And on a STA: set [cfgWlanInterfaceUseVendorSsid](#) to enabled(1) to allow the Client/Station to use the vendor element in the AP beacon.

This parameter is supported on 802.11n products only.

5.5 Cellular Network Configuration

Cellular communication is only available on specially designed products, see also section 1.2. The configuration is divided into different sections:

- **Interface Configuration** ([cfgNetWwanEnabled](#)),
- **SIM Slot Configuration** ([cfgCellSimTable](#)),
- **SIM Profile Configuration** ([cfgCellSimProfileTable](#)), and
- **Connection Management** ([cfgCellConnectionManagement](#)).

5.5.1 Interface Configuration

The cellular network interface `wwan0`, typically labeled Wireless Wide Area Network (WWAN) at system level, is disabled by default. In order to use cellular communication, the corresponding network interface must be enabled using [cfgNetWwanEnabled](#).

5.5.2 SIM Slot Configuration

At least one SIM slot of the cellular network interface `wwan0` must be assigned to a SIM profile. Each value of the SIM slots [cfgCellSimSlot1](#) and [cfgCellSimSlot2](#) refers to the index of a SIM profile in the [cfgCellSimProfileTable](#). In case one of the slots is not used, set its reference to -1 in order to deactivate the slot. The primary SIM slot can be selected by [cfgCellSimPrimarySlot](#). First connection attempts are made with the primary SIM.

5.5.3 SIM Profile Configuration

The various parameters for a SIM card are organised in SIM profiles. The [cfgCellSimProfileTable](#) contains the respective profiles, which are then assigned to the SIM slots. If the SIM card is locked by a PIN, [cfgCellSimProfilePinEnabled](#) must be enabled to unlock it with the PIN defined in [cfgCellSimProfilePin](#). The Access Point Name (APN) [cfgCellSimProfileApn](#) is set to `auto` by default, so that the SIM card selects it by itself, the credential parameters are ignored. Defined APN and credentials configuration may be required for roaming.

Note: Roaming is enabled by default and can currently not be deactivated. Consequently, if the SIM card allows roaming, the cellular product will do so.

5.5.4 Connection Management Configuration

The Connection Management controls which SIM slot is to be used. If SIM Rotation [cfgCellConnMgmtSimRotationEnabled](#) is enabled, the Connection Management swaps from the primary to the secondary slot and vice versa as soon as the existing connection is considered lost. The connection loss detection mode is selected by [cfgCellConnMgmtMonMode](#).

- `signal(0)` for monitoring the strength and quality of the cellular signal, and
- `remoteHosts(1)` for monitoring the cellular network traffic.

The evaluation of the connection based on the `signal(0)` is hardware dependent and considered slow compared to the `remoteHosts(1)` option. The advantage of monitoring the connection at network level is that an interrupted data flow is detected much faster depending on settings of the detection algorithm. On the downside, constant network traffic is needed and the monitoring is dependent on at least one remote host, see [cfgCellConnMgmtMonRemoteTable](#).

The algorithms for detecting a connection loss can be adjusted with the monitor period [cfgCellConnMgmtMonPeriod](#) and the monitor count [cfgCellConnMgmtMonCount](#). The monitor period defines the time in seconds between two consecutive evaluations of the connection. Whereas, the monitor count defines the needed amount of consecutive failed connection tests before it is considered down. These parameters only apply for monitoring an established connection.

The loss of connection triggers the Connection Management to change the SIM slot and to set up a new connection. If no connection could be set up within a certain time, which is hardware dependent and can take up to one minute, the Connection Management changes the SIM slots connection could be established successfully. However, the monitoring of the new connection does not start until it is up and running.

5.5.4.1 Monitored Remotes

If `cfgCellConnMgmtMonMode` is set to `remoteHosts(1)`, at least one enabled entry in the `cfgCellConnMgmtMonRemoteTable` is required. The connection attempts are made one after another according to the order of the table entries. As soon as one remote host is available the cellular connection is considered up. The address of the remote host is defined by `cfgCellConnMgmtMonRemoteAddress` and `cfgCellConnMgmtMonRemoteType` determines how the availability the remote host is tested.

5.5.5 Cellular Router as Gateway

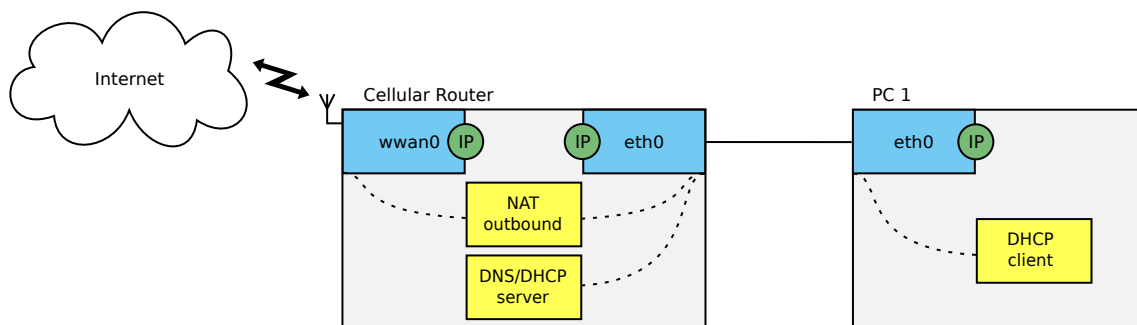


Figure 5.6: Cellular Router as Internet Gateway

The configuration example in Listing 5.3 defines a Cellular Router as a gateway with an enabled DHCP server and an outbound NAT, which is by default set up as described in section 5.31.1.2. Connect a PC with a DHCP client to one of the Ethernet interfaces of the Cellular Router. The SIM card must be inserted in slot 1. The APN is selected automatically.

```
WESTERMO-SW6-MIB::cfgSysHostname.0 = MR-Gateway
WESTERMO-SW6-MIB::cfgNetWwanEnabled.0 = INTEGER: 1
WESTERMO-SW6-MIB::cfgCellSimSlot1.0 = INTEGER: 0
WESTERMO-SW6-MIB::cfgCellSimSlot2.0 = INTEGER: -1
WESTERMO-SW6-MIB::cfgDhcpGlobalEnabled.0 = 1
```

Listing 5.3: Configuration of a Cellular Router as Gateway

Note: The parameter (`cfgCellSimSlot1`) is set to 0, which represents the index of the default SIM profile.

5.5.6 Cellular Network Status

The general status of each cellular network interface can be requested by using the entries of the [swCellTable](#). Additionally there are SNMP traps that indicate the link status of the corresponding network interface, see [Message Codes 360 and 361](#).

5.6 OpenVPN Configuration

OpenVPN may operate as a client or server. *Software 6* currently only supports the client role, see [cfgVpnOpenvpnMode](#).

OpenVPN interfaces are either of type tun or tap. For routed (tun) or bridged (tap) configurations, see [cfgVpnOpenvpnDevType](#). Both sides of the connection must use the same DevType. If the interface is of type tun, it may not be part of a bridge, see [cfgNetOpenvpnBridge](#).

5.6.1 OpenVPN Configuration Example

OpenVPN, IPsec and GRE have been introduced to supplement cellular networks in *Software 6*.

This following example depicts the usage of OpenVPN in a cellular network.

The goal is to provide private access from the Cellular Router to PC 1, see [Figure 5.7](#). An encrypted tunnel is created to travers public networks such as the Internet or private transfer networks of cellular providers.

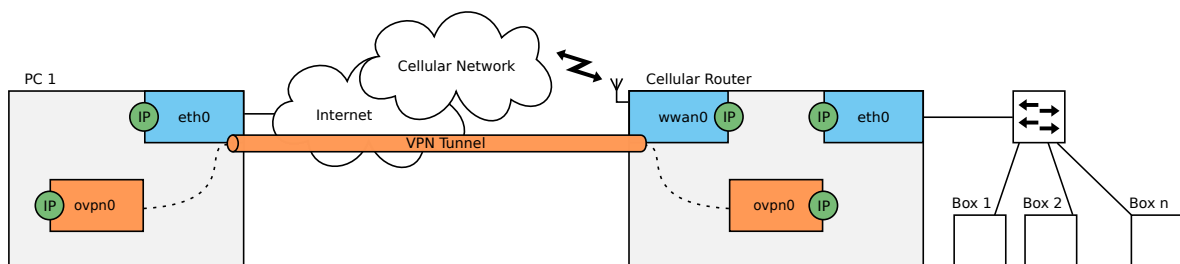


Figure 5.7: *OpenVPN in a Cellular Network*

In this example, PC 1 acts as OpenVPN server and the Cellular Router as a client. Multiple clients may connect to the server.

The focus of this example is OpenVPN. A working cellular network is assumed. Section 5.5 explains how to configure cellular networks.

A working OpenVPN instance requires:

- Installed key material
- A configured and enabled OpenVPN interface
- A configured OpenVPN instance

5.6.1.1 Installing the Key Material

The key material is provided by the administrator of the OpenVPN server. Assuming that the tunnel between PC 1 and the Cellular Router is secured by the Transport Layer Security (TLS) protocol, the key material consists of:

- The Certificate of the Certificate Authority (CA)
- The Client Certificate
- The Private Key

Section 4.13.1 describes how to install key material.

5.6.1.2 Configuring the OpenVPN Interface

The configuration of an OpenVPN interface is defined by the MIB entries of the `cfgNetOpenvpnTable`.

The required entries for the example in Figure 5.7 are in the Listing 5.4:

```
WESTERMO-SW6-MIB::cfgNetOpenvpnEnabled.0 = 1
WESTERMO-SW6-MIB::cfgNetOpenvpnBridge.0 = -1
```

Listing 5.4: *OpenVPN Interface Configuration*

The OpenVPN interface and instance are created by setting `cfgNetOpenvpnEnabled` to `enabled(1)`. The OpenVPN interface `ovpn0` in the example in Figure 5.7 is of type `tun`. The interface may not be bridged, thus `cfgNetOpenvpnBridge` is set to `-1`. The remaining entries may be left with their default values.

No IP address needs to be configured. The server provides the IP address for the client.

5.6.1.3 Configuring the OpenVPN Instance

The OpenVPN instance, similar to its corresponding interface, is defined the SNMP table [cfgVpnOpenvpnTable](#).

The required entries for the example in Figure 5.7 are in the Listing 5.5:

```
WESTERMO-SW6-MIB::cfgVpnOpenvpnMode.0 = 0
WESTERMO-SW6-MIB::cfgVpnOpenvpnDevType.0 = 0
WESTERMO-SW6-MIB::cfgVpnOpenvpnRemote.0 = 192.168.1.1
WESTERMO-SW6-MIB::cfgVpnOpenvpnRPort.0 = 1194
WESTERMO-SW6-MIB::cfgVpnOpenvpnKeyType.0 = 0
WESTERMO-SW6-MIB::cfgVpnOpenvpnKeyPassword.0 = password4key
WESTERMO-SW6-MIB::cfgVpnOpenvpnAuth.0 = SHA256
WESTERMO-SW6-MIB::cfgVpnOpenvpnCipher.0 = AES-256-CBC
WESTERMO-SW6-MIB::cfgVpnOpenvpnCompress.0 = 2
```

Listing 5.5: *OpenVPN Instance Configuration*

The Cellular Router acts as client, which is configured by using [cfgVpnOpenvpnMode](#). The interface type must be set to `tunnel(0)`, see [cfgVpnOpenvpnDevType](#). The Entries [cfgVpnOpenvpnRemote](#) and [cfgVpnOpenvpnRPort](#) define the network address of the OpenVPN server to which the client connects to. The Entry [cfgVpnOpenvpnRemote](#) accepts IPv4 addresses, as well as hostnames. For hostnames to work, a Domain Name Server (DNS) needs to be configured, see [cfgSysName-serverTable](#). Port number 1194 is the default IANA (Internet Assigned Numbers Authority) assigned port for OpenVPN.

The installed key material for the OpenVPN instance is selected by the key type, see [cfgVpnOpenvpnKeyType](#). The server uses TLS, thus the key type is `asymmetric(0)`. If the private key is encrypted, the key password must be configured using [cfgVpnOpenvpnKeyPassword](#).

Starting with OpenVPN 2.4, Negotiable Crypto Parameters (NCP) is enabled by default. It is used to automatically set a cipher and authentication algorithm. However, it is recommended to always set the values specified by the server administrator in [cfgVpnOpenvpnAuth](#) and [cfgVpnOpenvpnCipher](#), since the server may run an older version. Similar applies to the compression algorithm. In order to receive the compression settings automatically from the server, [cfgVpnOpenvpnCompress](#) may be set to `allowPush(1)`. However, an older server version will not send the expected settings, leaving the OpenVPN tunnel to fail.

The remainig entries of the OpenVPN instance configuration may be left with their default values.

5.7 Wireless Network Access Point

For an initial configuration of an Access Point which is used as public wireless network AP configure at least the following items:

- SSID ([cfgWlanfaceSsid](#)) - example: "wpwnap"
- Wireless password ([cfgWlanfacePassword](#))
- Wireless mode ([cfgWlanDevModulation](#)) - example: 2.4 GHZ = ng(10), 5 GHz = na(12)
- Wireless bandwidth ([cfgWlanDevBandwidth](#)) - example: ht20(0) or ht40Plus(1)
- Optionally set system hostname ([cfgSysHostname](#)) - example: "AP2G" or "AP5G-HT40+"

5.7.1 Configuration File Example: Access point 2.4GHz

Configuration File Example: Access point 2.4GHz

```
# Config.Format = raw
WESTERMO-SW6-MIB::cfgSysHostname.0 = AP2G
WESTERMO-SW6-MIB::cfgWlanfaceSsid.0 = wpwnap
```

5.7.2 Configuration File Example: Access point 5GHz

Configuration File Example: Access point 5GHz HT40+

```
# Config.Format = raw
WESTERMO-SW6-MIB::cfgSysHostname.0 = AP5G-HT40+
WESTERMO-SW6-MIB::cfgWlanDevModulation.0 = 12
WESTERMO-SW6-MIB::cfgWlanDevBandwidth.0 = 1
WESTERMO-SW6-MIB::cfgWlanDevFrequency.0 = 5500
WESTERMO-SW6-MIB::cfgWlanfaceScanList.0 = 1
```

For a more sophisticated public wireless network AP configuration (i.e. WiFi Hotspot) please refer to [Public Wireless Network \(PWN\)](#).

5.8 Wireless Data Rate Control

Wireless technology supports a wide range of bitrates for optimal throughput in any environment. The bitrate depends mainly on the signal level seen by the wireless receiver.

Usually the best bitrate for the actual environment is evaluated and chosen by the rate controller. For some applications, especially with fast changing environments, (e.g. mobility applications) the task to find the optimal bitrate must be optimized. *Software 6* supports fine grade optimization for bitrates.

5.8.1 Reduced Set of Bitrates

With [cfgWlanIfaceBitrates](#) and [cfgWlanDevHtCapabilities](#) the number of possible bitrates can be reduced to allow the rate controller for faster adaption to a changing environment.

5.8.2 QMRR

QMRR supports Multi Rate Retry per wireless queue (VO, VI, BK, BE) and is configured with [cfgWlanDevQmrrString](#).

QMRR logging into Syslog can be enabled by [cfgWlanGlbLinkmonitorQmrrlogging](#) and [cfgWlanGlbLinkmonitorInterval](#)

Format: QMRR|<kernel time>|<queue>|<mode>|<guard>|<rate>|<success>|<attempts>

- mode: HT20|HT40 (there can be further modes like CCK)
- guard: LGI|SGI (there can be further guards like SP)
- rate: MCS0, MCS1, ..., MCS31 (there can be further rates like 1.0M)

5.9 802.11s Mesh

802.11s Mesh is an official IEEE standard for Wireless Mesh operation. Contrary to other proprietary Mesh implementations it is open and allows interoperability with other vendors which are 802.11s compliant. It allows to connect multiple devices without the need for an Access Point. Each node is a Mesh Point and transmits its own Mesh beacons. When two Mesh Points see each others beacons they automatically associate with each other and exchange keys when configured for encryption.

To configure a Mesh Point set [cfgWlanIfaceMode](#) to 3. Mesh only supports two types of encryption (see [cfgWlanIfaceEncryption](#)): open(0) (no encryption) and sae(7).

When a Mesh Points starts up, it takes the frequency list specified by [cfgWlanIfaceAcsList](#) and scans these frequencies for the configured Mesh SSID ([cfgWlanIfaceBssid](#)). If it finds an already existing Mesh on any of the scanned frequencies, it joins it. Should it not find an already existing one, it starts beaconing a new Mesh on the frequency specified in [cfgWlanDevFrequency](#).

802.11s Mesh has 6 address fields in its header. Because of this it is possible to bridge a Mesh interface.

A Mesh can only work on a single frequency at the same time, thus operation on DFS frequencies is not recommended.

5.10 Bridge Mode (4addr)

Bridge mode or WDS (Wireless Distribution System) mode is a non-standard extension to the wireless 802.11 standard using a 4-address-format to allow transparent Ethernet bridging on the client (STA).

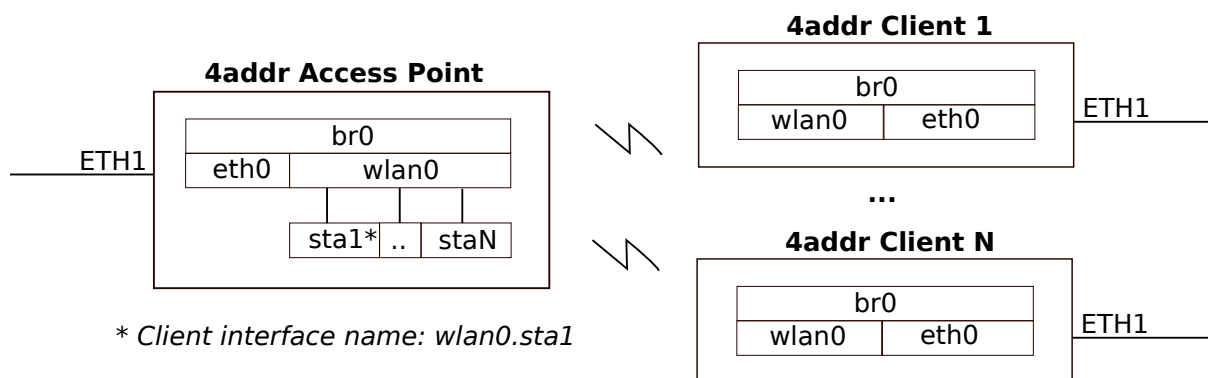


Figure 5.8: 4addr bridge mode setup

Configuration File Example: Access point

```
# Config.Format = raw
WESTERMO-SW6-MIB::cfgWlaniface4addr.0 = 1
```

Configuration File Example: STA/client

```
# Config.Format = raw
WESTERMO-SW6-MIB::cfgNetIpAddr.0 = 192.168.1.30/24
WESTERMO-SW6-MIB::cfgWlanifaceMode.0 = 1
WESTERMO-SW6-MIB::cfgWlaniface4addr.0 = 1
```



Important: [Layer 2 NAT Mode](#) instead of Bridge Mode (4addr) is recommended in combination with [Mobility](#).

5.11 Layer 2 NAT Mode

As the 4addr bridge mode (section [Bridge Mode \(4addr\)](#)), the Layer 2 NAT mode allows transparent IP bridging on the client (STA).



Important: This only works for IP traffic and not for generic L2 frames.

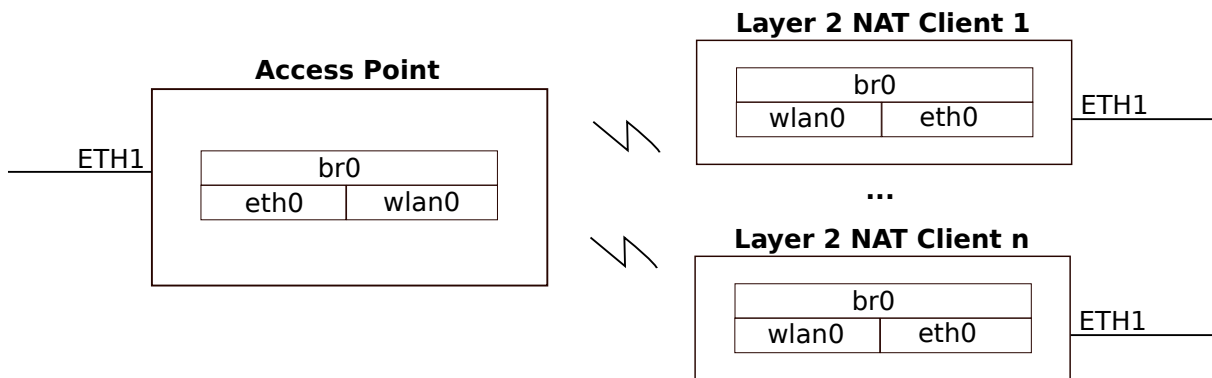


Figure 5.9: Layer 2 NAT bridge mode setup

There is no special Layer 2 NAT configuration necessary on the AP side. For the client (STA) the configuration is as following.

Configuration File Example: STA/client

```
# Config.Format = raw
WESTERMO-SW6-MIB::cfgNetIpAddr.0 = 192.168.1.30/24
WESTERMO-SW6-MIB::cfgWlanifaceMode.0 = 1
WESTERMO-SW6-MIB::cfgWlanifaceL2nat.0 = 1
```

The Layer 2 NAT mode is recommended if layer 3 (IP) transparency is required in combination with [Mobility](#) (fast handover). With Layer 2 NAT the same handover performance is achieved as with client (STA) in routing mode, which is the default mode. Whereas with Bridge Mode (4addr) the handover performance is about 10 times slower. Therefore, if Layer 3 transparency and fast handover are required, it is recommend to use Layer 2 NAT instead of 4addr bridge mode.



Important: Layer 2 NAT bridging mode cannot be used in combination with DHCP. Devices on the Ethernet side of the client (STA) cannot obtain DHCP leases.

5.12 Wireless MAC Address Overwrite

It is possible to overwrite the MAC address of the wireless interface in Access Point and client (STA) mode.

This feature might be useful in a mobility application with several Access Point along the track. The MAC Address could encode for example the index of the Access Point which makes it easier to configure static neighbour lists.

Another use case might be for applications where the MAC address of the client (STA) shall be the same as of the device attached to the modem. The user can configure the MAC address of the attached wired equipment as the source MAC address of the wireless interface. The cloned address is then used for all wireless communication.

Configuration File Example: Configure new MAC for the wireless radio

```
# Config.Format = raw
WESTERMO-SW6-MIB::cfgWlanInterfaceBssid.0 = STRING: 00:11:22:33:44:55
```

5.13 Wireless Security

5.13.1 WPA Encryption

The WPA (Wifi Protected Access) security standard is implemented according to IEEE 802.11i. It uses CCMP (Counter Mode Cipher Block Chaining Message Authentication Code Protocol) for message confidentiality and integrity. CCMP is an enhanced data cryptographic encapsulation mechanism designed for data confidentiality and based upon the Counter Mode with CBC-MAC (CCM mode) of the Advanced Encryption Standard (AES) standard.

WPA Personal uses shared passwords and Enterprise is based on certificates (TLS) and passwords when used with TTLS or PEAP.

WPA3 Personal and Enterprise are new Wi-Fi security standards developed by WifiAlliance. WPA3 Personal introduces SAE (Simultaneous Authentication of Equals), which is an authentication protocol with stronger security protections against attacks such as offline dictionary attacks, key recovery, and message forging. WPA3 Enterprise is equivalent of WPA2 with strongest ciphers used. The WPA3 Personal Transition Mode allows both SAE and WPA2 PSK. It can be used in scenarios where not all the clients support SAE.

Only WPA2/IEEE 802.11i is supported. WPA/IEEE 802.11i/D3.0 which uses TKIP encryption protocol is considered not secure and is not available.

Wi-Fi Enhanced Open introduces Opportunistic Wireless Encryption (OWE). It provides encryption and privacy on open, non-password-protected networks in areas.

The following configuration items are supported:

- [cfgWlanfaceEncryption](#) to configure the wireless encryption mode. Available modes are open(0), psk(3) for WPA2 Personal, eap(6) for WPA2/3 Enterprise, sae(7) for WPA3 Personal, owe(8) for Opportunistic Wireless Encryption and saepsk(9) for WPA3 Personal Transition Mode
- [cfgWlanfacePassword](#) to configure the WPA2 or WPA3 Personal passphrase (pre-shared key)

To further increase security, the Management Frame Protection (specified in IEEE 802.11w) can be enabled (see section [Management frame protection \(MFP, 802.11w\)](#)). MFP is mandatory for WPA3 Personal and Enterprise.

5.13.1.1 Overview of the encryption modes

Encryption mode	cfgWlanfaceEncryption	cfgWlan802dot1xCiphers
OPEN	open(0)	none
WPA2 Personal (PSK)	psk(3)	none
WPA3 Personal (SAE)	sae(7)	none
WPA2 Enterprise	eap(6)	none
WPA3 Enterprise	eap(6)	ECDHE-RSA-AES256-GCM-SHA384 ECDHE-ECDSA-AES256-GCM-SHA384 ECDHE-RSA-AES256-SHA384
Wi-Fi OWE	owe(8)	none
WPA3 Transition Mode	saepsk(9)	none

Table 5.1: *Overview of the encryption modes*

5.13.2 Port-based Network Access Control (802.1X)

IEEE 802.1X is an IEEE Standard for port-based Network Access Control (PNAC). It provides an authentication and authorization mechanism to devices wishing to attach to a LAN or WLAN.

IEEE 802.1X authentication involves three parties:

supplicant client device that wishes to attach to the LAN or WLAN (STA)

authenticator network device such as an Ethernet switch or wireless access point through which the supplicant is connected to the network (AP)

authentication server a host running software supporting the RADIUS and EAP protocols (AS)

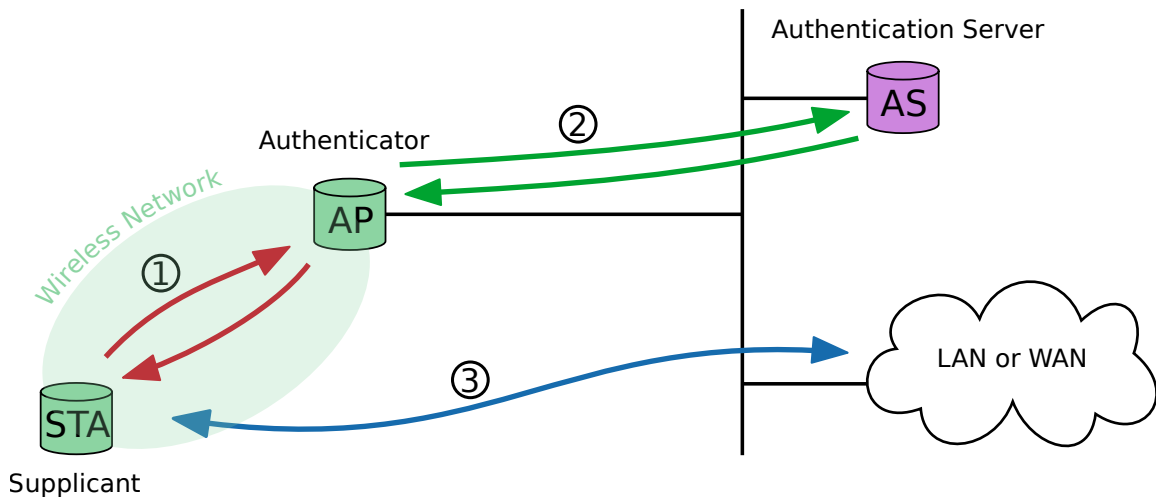


Figure 5.10: *Supplicant authentication using IEEE 802.1X*

The supplicant is not allowed to access through the authenticator to the protected side of the network ③ until the supplicant’s identity has been validated and authorized. The authenticator acts like a security guard to a protected network. With IEEE 802.1X port-based authentication, the supplicant provides credentials to the authenticator ①. Accepted credentials can be user name/password or a digital certificate. The authenticator forwards the credentials to the authentication server for verification ②. If the authentication server determines the credentials as valid, the supplicant (client device) is allowed to access resources located on the protected side of the network.

The following example shows how to configure SW6 devices as supplicant (STA) and authenticator (AP).

Requirements

- Configured authentication server¹
- PKI to generate digital certificates, valid client certificate files
- Time server to share time information between the hosts in the network

Wireless access point configuration

The following configuration example configures a device as authenticator:

```
Configuration File Example: authenticator
# Config.Format = raw
```

¹FreeRADIUS 4 is used for internal testing, find the project under <http://freeradius.org/> (February 2017)

```
# basic AP configuration
WESTERMO-SW6-MIB::cfgSysTimezone.0 = CET-1CEST,M3.5.0,M10.5.0/3
WESTERMO-SW6-MIB::cfgNetIpAddr.0 = 192.168.3.22/24
WESTERMO-SW6-MIB::cfgWlanDevModulation.0 = 12
WESTERMO-SW6-MIB::cfgWlanDevFrequency.0 = 5180
WESTERMO-SW6-MIB::cfgWlanInterfaceSsid.0 = radiustesting
WESTERMO-SW6-MIB::cfgWlanInterfaceEncryption.0 = 6
WESTERMO-SW6-MIB::cfgWlanGblCountry.0 = EU
WESTERMO-SW6-MIB::cfgNtpServer1.0 = 192.168.2.2
# IEEE 802.1X related configuration
WESTERMO-SW6-MIB::cfgWlan802dot1xAuthSrvEnabled.0 = 1
WESTERMO-SW6-MIB::cfgWlan802dot1xAuthSrvIpAddr.0 = 192.168.3.2
WESTERMO-SW6-MIB::cfgWlan802dot1xAuthSrvSharedSecret.0 = superSharedSecret
```

The authenticator does not need any certificate information. The shared secret 'superSharedSecret' has to match the authentication server client configuration.

The authenticator will forward incoming client authentication requests to the primary authentication server address (192.168.3.2). More than one authentication server address can be defined by adding more records to the authentication server configuration table.

Configuration File Example: authenticator, second authentication server address

```
# Config.Format = raw
# basic AP configuration
WESTERMO-SW6-MIB::cfgWlan802dot1xAuthSrvEnabled.1 = 1
WESTERMO-SW6-MIB::cfgWlan802dot1xAuthSrvIpAddr.1 = 192.168.3.2
WESTERMO-SW6-MIB::cfgWlan802dot1xAuthSrvSharedSecret.1 = superSharedSecret
```

If the primary authentication server is not reachable, an exponential back off is implemented:

- timeout of first attempt is 3 seconds
- after each failure the timeout is increased to maximum timeout of 120 seconds
- the authenticator gives up after 10 attempts
- after the fourth attempt, a backup server (secondary, if configured) will be attempted

If none of the configured authentication server is reachable, the supplication will not be authorized to communicate over the access point (authenticator) - no communication is possible.

Wireless station configuration

If encrypted private keys are used for the client, the pass phrase to unlock the key has to be configured before the private key upload. If no matching pass phrase is provided, the client private key file will be rejected by the device.

Use the MIB element [cfgWlan802dot1xClientKeyPassword](#) to define the pass phrase and [rpcCfgApply](#) to apply the changes.

If the private key is not locked by a pass phrase (unencrypted), this configuration parameter is ignored.

Use the web interface to upload the certificate files: *System -> Certificate Manager -> 802.1X*. Each file is verified before it is stored on the device. The mandatory information can be provided in this files:

client.crt client certificate (public key, PEM or X509 format)

client.key client private key matching the client X509 certificate (RSA)

ca.crt authentication server X509 certificate (public key, PEM or X509 format)

The client private key file may also contain the full CA certificate chain information in PEM format: it can contain client private key and CA certificate chain information (one or multiple certificates) in one single file. Make sure that in this case no separate client certificate file is provided.

Alternatively SNMP can be used to upload the mandatory certificate information to the supplicant. The MIB element [setCrtFileSelector](#) is used to define what kind of file should be uploaded and for which interface it should be used. The following file classes are defined:

10x class 100 is used for client certificate files where x is the interface index

20x class 200 is used for client private key files where x is the interface index

30x class 300 is used for CA certificate files where x is the interface index

Client certificate file upload via TFTP:

- WESTERMO-SW6-MIB::[setCrtFileSelector](#).0 100
- WESTERMO-SW6-MIB::[setCrtFileUrl](#).0 tftp://192.168.2.2/client.crt
- WESTERMO-SW6-MIB::[rpcCrtFile](#).0 1

Client private key file upload via TFTP:

- WESTERMO-SW6-MIB::[setCrtFileSelector](#).0 200
- WESTERMO-SW6-MIB::[setCrtFileUrl](#).0 tftp://192.168.2.2/client.key
- WESTERMO-SW6-MIB::[rpcCrtFile](#).0 1

CA certificate file upload via TFTP:

- WESTERMO-SW6-MIB::[setCrtFileSelector.0](#) 300
- WESTERMO-SW6-MIB::[setCrtFileUrl.0](#) tftp://192.168.2.2/ca.crt
- WESTERMO-SW6-MIB::[rpcCrtFile.0](#) 1

The following configuration example configures a device as supplicant.

Configuration File Example: supplicant

```
# Config.Format = raw
# basic STA configuration
WESTERMO-SW6-MIB::cfgSysTimezone.0 = CET-1CEST,M3.5.0,M10.5.0/3
WESTERMO-SW6-MIB::cfgNetWlanBridge.0 = -1
WESTERMO-SW6-MIB::cfgNetIpEnabled.1 = 1
WESTERMO-SW6-MIB::cfgNetIpAddr.0 = 192.168.2.11/24
WESTERMO-SW6-MIB::cfgNetIpAddr.1 = 192.168.3.11/24
WESTERMO-SW6-MIB::cfgWlanDevModulation.0 = 12
WESTERMO-SW6-MIB::cfgWlanInterfaceMode.0 = 1
WESTERMO-SW6-MIB::cfgWlanInterfaceSsid.0 = radiustesting
WESTERMO-SW6-MIB::cfgWlanInterfaceEncryption.0 = 6
WESTERMO-SW6-MIB::cfgWlanInterfaceScanList.0 = 2
WESTERMO-SW6-MIB::cfgWlanGibbCountry.0 = EU
WESTERMO-SW6-MIB::cfgNtpServer1.0 = 192.168.2.2
# IEEE 802.1X related configuration
WESTERMO-SW6-MIB::cfgWlan802dot1xClientKeyPassword.0 = superKeySecret
```

If no certificates are uploaded before applying WPA2-EAP as encryption mode ([cfgWlanInterfaceEncryption](#)), the configuration apply will fail. NTP has to be enabled on the station (time sync on the access point is optional), make sure that the station shares the same time information and time zone configuration as the authentication server. This is later important for the successful certificate validation.

After applying the described configuration on supplicant (STA) and authenticator (AP), the STA should be successfully authorized. The STA authorization status can be verified using the STA's web interface: (*Status -> Wireless Connections -> Station Dump*).

5.13.3 Ciphers

Supported authentication and encryption algorithms (cipher suites).

The TLSv1.2 cipher suites are supported and recommended to use:

ADH-AES128-GCM-SHA256	ADH-AES128-SHA256
ADH-AES256-GCM-SHA384	ADH-AES256-SHA256
AES128-CCM	AES128-CCM8
AES128-GCM-SHA256	AES128-SHA256
AES256-CCM	AES256-CCM8
AES256-GCM-SHA384	AES256-SHA256
DHE-DSS-AES128-GCM-SHA256	DHE-DSS-AES128-SHA256
DHE-DSS-AES256-GCM-SHA384	DHE-DSS-AES256-SHA256
DHE-PSK-AES128-CCM	DHE-PSK-AES128-CCM8
DHE-PSK-AES128-GCM-SHA256	DHE-PSK-AES256-CCM
DHE-PSK-AES256-CCM8	DHE-PSK-AES256-GCM-SHA384
DHE-PSK-CHACHA20-POLY1305	DHE-RSA-AES128-CCM
DHE-RSA-AES128-CCM8	DHE-RSA-AES128-GCM-SHA256
DHE-RSA-AES128-SHA256	DHE-RSA-AES256-CCM
DHE-RSA-AES256-CCM8	DHE-RSA-AES256-GCM-SHA384
DHE-RSA-AES256-SHA256	DHE-RSA-CHACHA20-POLY1305
ECDHE-ECDSA-AES128-CCM	ECDHE-ECDSA-AES128-CCM8
ECDHE-ECDSA-AES128-GCM-SHA256	ECDHE-ECDSA-AES128-SHA256
ECDHE-ECDSA-AES256-CCM	ECDHE-ECDSA-AES256-CCM8
ECDHE-ECDSA-AES256-GCM-SHA384	ECDHE-ECDSA-AES256-SHA384
ECDHE-ECDSA-CHACHA20-POLY1305	ECDHE-PSK-CHACHA20-POLY1305
ECDHE-RSA-AES128-GCM-SHA256	ECDHE-RSA-AES128-SHA256
ECDHE-RSA-AES256-GCM-SHA384	ECDHE-RSA-AES256-SHA384
ECDHE-RSA-CHACHA20-POLY1305	PSK-AES128-CCM
PSK-AES128-CCM8	PSK-AES128-GCM-SHA256
PSK-AES256-CCM	PSK-AES256-CCM8
PSK-AES256-GCM-SHA384	PSK-CHACHA20-POLY1305
RSA-PSK-AES128-GCM-SHA256	RSA-PSK-AES256-GCM-SHA384
RSA-PSK-CHACHA20-POLY1305	TLS_AES_128_GCM_SHA256
TLS_AES_256_GCM_SHA384	TLS_CHACHA20_POLY1305_SHA256

The cipher string format explained for ECDHE-RSA-AES256-GCM-SHA384:

ECDHE key exchange algorithm, Elliptic Curve Diffie-Hellman

RSA authentication algorithm, Rivest-Shamir-Adleman

AES256-GCM cipher algorithm, Advanced Encryption Standard with strength and mode (Galois Counter Mode)

SHA384 MAC or PRF algorithm, Secure Hash Algorithm with strength

The list of supported cipher suites may change over time. New cipher suites will be added, weak cipher suites will be removed. Following cipher suites are recommended and are maintained over long term:

- ECDHE-RSA-AES128-GCM-SHA256

- ECDHE-RSA-AES256-GCM-SHA384
- DHE-RSA-AES128-GCM-SHA256
- DHE-RSA-AES256-GCM-SHA384
- ECDHE-RSA-AES256-SHA384

With `cfgWlan802dot1xCiphers` the cipher suite(s) to be used can be configured according to the syntax of OpenSSL.

Examples

- Define cipher suites with RSA key exchange or authentication:
`cfgWlan802dot1xCiphers.0 = RSA`
- Define a single cipher suite:
`cfgWlan802dot1xCiphers.0 = DHE-RSA-AES256-GCM-SHA384`
- Provide an ordered list of cipher suites:
`cfgWlan802dot1xCiphers.0 = DHE-RSA-AES256-GCM-SHA384:RSA-AES-128GCM-SHA256`
- Provide a list of cipher suites which is to sort in order of encryption algorithm key length:
`cfgWlan802dot1xCiphers.0 = DHE-RSA-AES256-GCM-SHA384:RSA-AES-128GCM-SHA256:@STRENGTH`

The OpenSSL tool `ciphers` can be used as a test tool to determine the appropriate cipherlist (e.g. `openssl ciphers RSA`)

Please note the the configuration will fail to apply if the value set for `cfgWlan802dot1xCiphers` provides an empty list (no cipher suite).

5.13.4 Management frame protection (MFP, 802.11w)

The management frame protection (MFP) protects the device from denial of service attacks. By default, the management frames are not protected from being used to attack the device. The MFP 802.11w adds a field in the frame to authenticate the frame sender. If the device receives a management frame from an incorrect sender, it will discard the frame.

The following configuration items are supported:

- `cfgWlanInterface80211w` to enable the Management Frame Protection (MFP) mechanism.
- `cfgWlanInterface80211wMaxTimeout` to configure the SA query max timeout
- `cfgWlanInterface80211wRetryTimeout` to configure the SA query retry timeout

5.14 IP Routing

The devices not only provide switch functionality but are also able to route data packages.

5.14.1 Static Routing

Using static routing the devices can specify the next hop router to use to reach a given IP subnet, or add additional (directly attached) subnets to a local interface.

The example below shows a static route to the 192.168.11.0/24 subnet using 192.168.1.2 as gateway. A detailed description of all routing related SNMP commands can be found in [cfgRouting](#).

Configuration File Example: Static route

```
# Config.Format = raw
WESTERMO-SW6-MIB::cfgRouteTableEnabled.0 = 1
WESTERMO-SW6-MIB::cfgRouteTableDestinationNetwork.0 = 192.168.11.0/24
WESTERMO-SW6-MIB::cfgRouteTableGateway.0 = 192.168.1.2
WESTERMO-SW6-MIB::cfgRouteTableSource.0 = 192.168.1.20
```

5.15 VLAN

The *Software 6* has built in capability for virtual LANs (VLAN). The devices can be easily integrated into existing network environments where VLANs are in use.

The VLAN configuration is located under [cfgNetVlanTable](#).

Untagged frames are internally handled by using VID 0.

5.15.1 Multi SSID and VLAN

The devices support multiple SSIDs on a single radio. It can broadcast up to 64 wireless networks with different names (i.e SSIDs). When using Multi SSID, users could also assign different VLAN ID to different wireless network. This makes it possible to get a device work with switches which has VLAN assigned for different access level and authority.

In the following example the Ethernet interfaces (eth0 and eth1) are configured so that either eth0 or eth1 are physically connected, but not both at the same time. The device shall be accessible via Ethernet (eth0 or eth1) for administrative purposes.

The process of configuring VLANs to separate SSIDs and making these networks accessible via Ethernet is as follows:

1. Setup 1st Ethernet interface (eth0) in VLAN 8, 103 and 203 and add it to the bridge (br0)
2. Setup redundant 2nd Ethernet interface (eth1)
3. Setup WLAN physical interface: frequency, power etc.
4. Setup 1st WLAN interface (wlan0) in VLAN 103 and add it to the bridge (br0)
5. Setup 2nd WLAN interface (wlan1) in VLAN 203 and add it to the bridge (br0)
6. Setup Administrative VLAN (VID 8)
7. Setup IP for Administrative VLAN (VID 8)

An example configuration of this kind is shown in Figure 5.11 below.

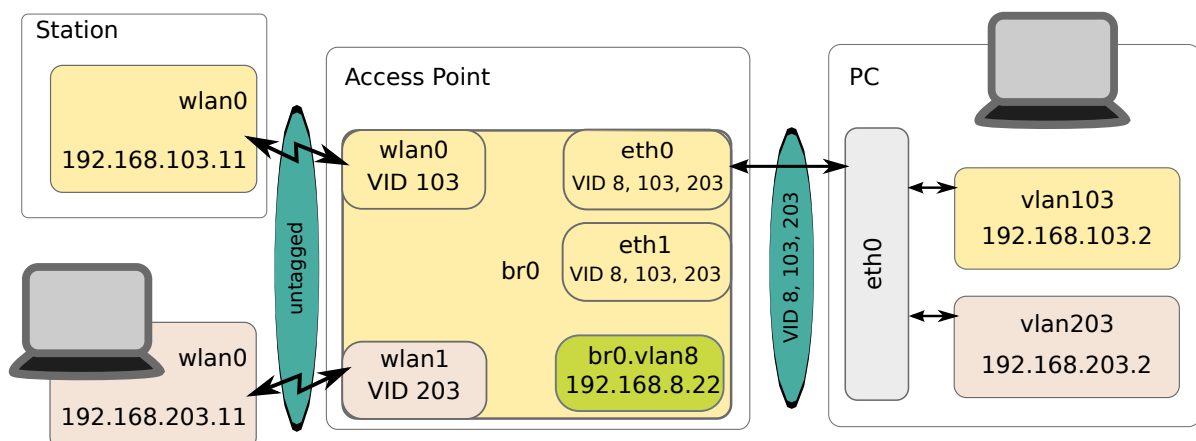


Figure 5.11: Multi SSID and VLAN example setup

Configuration File Example: Multiple SSID and VLAN

```
# Config.Format = raw
# 1. Setup 1st ethernet interfaces(eth0):
WESTERMO-SW6-MIB::cfgNetEthTrunk.0 = 8,103,203
# 2. Setup 2nd (redundant) ethernet interface (eth1):
WESTERMO-SW6-MIB::cfgNetEthTrunk.1 = 8,103,203
# 4. Setup 1st WLAN interface (wlan0, VID 103):
WESTERMO-SW6-MIB::cfgNetWlanTag.0 = 103
WESTERMO-SW6-MIB::cfgNetWlanVlanMode.0 = 1
WESTERMO-SW6-MIB::cfgWlanInterfaceSsid.0 = ssid103
```

```
# 5. Setup 2nd WLAN interface (wlan1, VID 203):
WESTERMO-SW6-MIB::cfgNetWlanEnabled.1 = 1
WESTERMO-SW6-MIB::cfgNetWlanBridge.1 = 0
WESTERMO-SW6-MIB::cfgNetWlanTag.1 = 203
WESTERMO-SW6-MIB::cfgNetWlanVlanMode.1 = 1
WESTERMO-SW6-MIB::cfgWlanInterfaceDevice.1 = 0
WESTERMO-SW6-MIB::cfgWlanInterfaceSsid.1 = ssid203
# 6. Setup Administrator VLAN (VID 8):
WESTERMO-SW6-MIB::cfgNetVlanVid.0 = 8
# 7. Setup IP for Administrative VLAN:
WESTERMO-SW6-MIB::cfgNetIpAddr.0 = 192.168.8.22/24
WESTERMO-SW6-MIB::cfgNetIpInterface.0 = br0.vlan8
```

5.16 Mobility

5.16.1 Fast Association

Fast Association bundles the optimization features which allows the client (STA) to find and connect to the best Access Point as fast as possible.

When the client (STA) is not connected it scans for available Access Point with the same SSID and then connects to the Access Point with the best RSSI level (Signal). The scan and re-connect process works as follows:

1. Loop through all frequencies defined in the [cfgWlanInterfaceScanList](#) and
 - a) tune the radio to a frequency on the [cfgWlanInterfaceScanList](#)
 - b) observe the channel for a few milliseconds. This is necessary to update the NAV (network allocation vector)
 - c) send a Probe Request frame with SSID specified
 - d) collect all Probe Responses for within a time window of a few milliseconds
2. Select from the Access Point with the same [cfgWlanInterfaceSsid](#) the one with the best RSSI level (Signal)
3. Connect to the selected Access Point

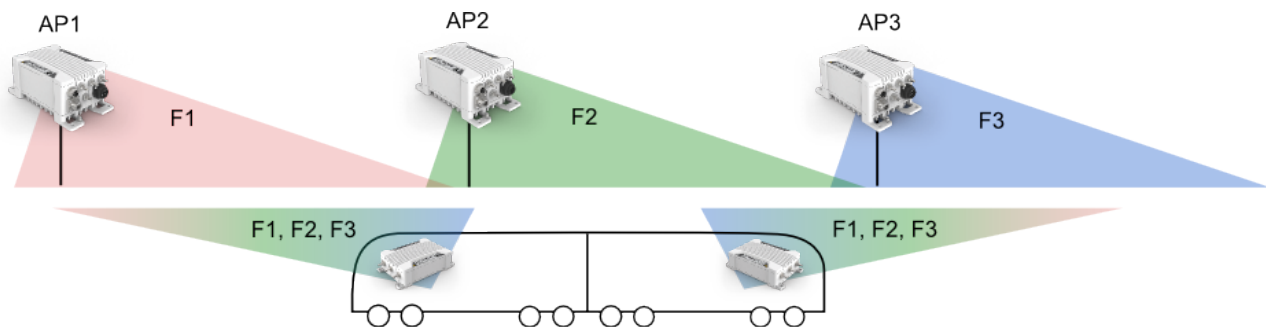
Once the client (STA) successfully connected, the client observes the Beacon frames received from its Access Point. As long as the filtered RSSI level of the Beacon frames is above a configurable threshold ([cfgWlanHoScanningLevel](#)), the client stays connected. If the RSSI level is below the

threshold or if too many Beacon frames are lost (`cfgWlanInterfaceBeaconMiss`), the client will disconnect and scan for a better Access Point.

Tip: For applications where the access points are located further apart so that no continuous coverage can be achieved, the `cfgWlanHoScanningLevel` should be set to 0 so that the client is always connected to the access point as long as there is one available.

5.16.2 Inter-AP Roaming

Inter-AP Roaming assumes continuous coverage. Meaning that on the edge of an Access Point coverage area there must be always certain overlapping with the next Access Point's coverage area. In a typical mobile to ground application, on ground (trackside or infrastructure) the device is configured in Access Point mode where the mobile device is in client (STA) mode.



Once the client is connected to an Access Point, it receives a list of neighbor Access Points as possible roaming candidates. The neighbor list contains the current operating frequency and BSSID of the neighbor Access Points. On the edge of the coverage area of the currently associated Access Point, the client initiates the scanning process to search for the next Access Point. Depending on the configured `cfgWlanHoProfile` the client scans for new Access Points while being still connected (background scanning) or it disconnects before starting to scan (foreground scanning). Which profile to use depends on the actual needs of the application. The scan and re-connect process works as follows:

1. Loop through the list of neighbor Access Points received from the Access Point, sorted by frequency
 - a) tune the radio to the current frequency
 - b) observe the channel for a few milliseconds. This is necessary to update the NAV (network allocation vector)
 - c) send a directed Probe Request frame to all neighbor BSSIDs on the current frequency
 - d) collect all Probe Responses

2. Select from the neighbor Access Points the one with the best RSSI level (Signal)
3. Connect to the selected Access Point

As a fall-back, if no neighbor list is available or no neighbour is a valid roaming candidate, the client uses the same scan and re-connect process as described in [Fast Association](#).

The client (STA) decides based on the received RSSI (Signal) and Distance of its Access Point when to roam. Either RSSI or Distance can initiate a handover independently. The RSSI is recorder for all Beacon frames received from the AP while the distance is periodically measured by ranging frames (see [cfgWlanHoDistanceMeasurementPeriod](#)). The low / high thresholds for the RSSI level and the near / far thresholds for the Distance can be configured on client and additionally on the Access Point. If the thresholds are configured on the Access Point these values are applied and used by the client for the current Access Point. This allows to tune the optimal roaming point for the coverage area of each Access Point cell separately.

The list of neighbors must be configured on each Access Point (see [cfgWlanNeighbourTable](#)) and the neighbour reports must be enabled (see [cfgWlanIfaceNeighbourReport](#)) when the so called Static Neighbour List (SNL) is configured. Inter-AP Roaming supports also Dynamic Neighbor Lists (DNL). DNL is required as soon as the operating frequency of the Access Points is not fix. This is the case if the Access Point changes its frequency due to a radar event (DFS) or due to interferences. Please refer to [Area Frequency Management \(AFM\)](#) for more information.

5.16.3 Handoff Filters

The client (STA) decides when to roam based on Beacon RSSI of its Access Point or based on Distance measurements to its Access Point. Filters are applied to both, RSSI and Distance, before comparing with the thresholds.

For Beacon RSSI the filter is configured on the client (STA) with [cfgWlanHoFilterMode](#), [cfgWlanHoFilterLongX](#) and [cfgWlanHoFilterLongY](#). In combination with the [cfgWlanIfaceBeaconInterval](#), which is configured on the Access Point, the RSSI filter characteristic can be optimally tuned for the application.

For Distance measurements the filter is configured on the STA with [cfgWlanHoDistanceFilterX](#) and [cfgWlanHoDistanceFilterY](#). In combination with the [cfgWlanHoDistanceMeasurementPeriod](#), which is configured on the STA, the Distance filter characteristic can be optimally tuned for the application.

5.16.4 Mobility Logging

For debugging purposes and verification of the system setup it is very important to have the possibility to log signal levels, distance measurements and the roaming process. To serve these needs the client

(STA) provides several handoff debug flags (see [cfgWlanDbgTable](#) and [setWlanDbgTable](#)) which allows to log received RSSI for each beacon, distance information for each ranging frame and handoff information in the Syslog (see [Logging Features](#)).

5.16.4.1 Important Messages for Handoff

Key	Format	Syslog	SNMP Trap
cfgWlanDbgHandoff	Handoff (Message Message Codes)	yes	yes
cfgWlanDbgBeaconrssi	RSSI (Message Message Codes)	yes	no
cfgWlanDbgBeaconfiltered	Filtered RSSI (Message Message Codes)	yes	no
cfgWlanDbgReports	Reports	yes	no
cfgWlanDbgRange	Distance	yes	no

5.16.4.1.1 Handoff

Format:

```
NOTICE 434 <iface> Handoff: |<reason>|<count>|<prev bssid>|<cur bssid>|<ssid  
↔ mgmt>|<ho\_time>|offchan_scan_time|
```

- **|Detection Reason Code|:**
 - 0 No reason (after restart)
 - 2 Low RSSI
 - 3 Beacon miss (connection loss)
 - 4 Low Ack
 - 5 High RSSI
 - 6 Near Distance
 - 7 Far Distance
- **|count|:** Number of handoffs since boot up
- **|prev_bssid|:** BSSID (MAC) of the previous access point
- **|cur_bssid|:** BSSID (MAC) of the current access point

- `|ssid_mgmt|`: Reserved
- `|ho_time|`: Handoff time measured between the disassociated and associated in milliseconds.
- `|offchan_scan_time|`: Handoff time measured for offchannel scanning.

5.16.4.1.2 RSSI

Format:

```
NOTICE 430 TS|<uptime>|<bssid>|RSSI_BCN|<rssi>
```

The RSSI is shown for all wireless Beacons received by the radio.

5.16.4.1.3 Filtered RSSI

Format:

```
NOTICE 431 TS|<uptime>|<bssid>|RSSI_BCN|<rssi>|<cur_filer_rssi>|<
↳ short_filter_rssi>|<long_filter_rssi>
```

The Filtered RSSI is shown only for the wireless Beacons of the Access Point the client(STA) is connected to.

5.16.4.1.4 Reports

The Reports ([cfgWlanDbgReports](#)) are dumped in Syslog (three lines for each report) and contains information about RSSI levels and other counters. The interval between the Reports depends on [cfgWlanIfaceBeaconInterval](#)

Format:

```
REP0|<uptime>|<mac>|<completed>|<sretries>|<lretries>|<xretries>
REP1|<uptime>|<mac>|<acompleted>|<aretries>|<axretries>|<expthrput>
REP2|<uptime>|<mac>|<cnt>|<seq>|<rssi>|<rssi0>|<rssi1>|<snr>|<crc>|<precrc
↳ >|<postcrc>
```

Example:

```
REP0|42351592|00:14:5a:03:49:2e|5809|82838|7185|16  
REP1|42351592|00:14:5a:03:49:2e|3930011|1372|495|34312  
REP2|42351592|00:14:5a:03:49:2e|139746|2107|32|32|0|31|0|0|0
```

Please contact support for more information about reports.

5.16.4.1.5 Distance

The Distance measurements ([cfgWlanDbgRange](#)) are dumped in Syslog and contains distance information (raw and filtered). The interval between the Reports depends on [cfgWlanHoDistanceMeasurementPeriod](#).

Format:

```
DISTANCE|<uptime>|<mac>|<raw>|<filtered>
```

Please contact support for more information about Distance measurements.

5.16.5 Fast BSS Transition (802.11r)

IEEE 802.11r is an amendment to the IEEE 802.11 standard. Fast BSS transition (FT) (802.11r) allows continuous connectivity of wireless devices in motion with fast and secure handoff. It is working for wireless devices in the same Mobility Domain (MD).

If using 802.1X, 802.11r provides a fast and still secure handoff within an MD. Whereas if using 802.1X without 802.11r, the handoff is secure but not fast.

For a simple configuration of the R0- and R1-Key Holder List (R0KH-/R1KH-list), use wildcard entries as given in the following example configuration. The wildcard entry in the R0-KH-list means that all APs of the same MD are allowed as R0-Key Holder. And the wildcard entry in the R1-KH List means, that every AP of the same MD is allowed to request an R1-Key.

5.16.5.1 AP 802.11r Configuration

When not using wildcard entries for R0KH-/R1KH-list, then each AP of an MD needs to have one entry for all of the other APs in the same MD. See [cfgWlan802dot11rR0KHTable](#) and [cfgWlan802dot11rR1KHTable](#) for some more description.

The following configuration example configures an AP for using FT, and using wildcard entry for R0- and R1-Key Holder List:

Configuration File Example: FT AP

```
# Config.Format = raw
# basic 802.11r configuration
WESTERMO-SW6-MIB::cfgWlan802dot11rEnabled.0 = 1
WESTERMO-SW6-MIB::cfgWlan802dot11rMobilityDomain.0 = a1b2
WESTERMO-SW6-MIB::cfgWlan802dot11rPmkR0Lifetime.0 = 10000
WESTERMO-SW6-MIB::cfgWlan802dot11rPmkR1KeyHolderIdentifier.0 = 000102030405 # i.e. use
    ↳ own MAC
WESTERMO-SW6-MIB::cfgWlan802dot11rR0KHParameter.0 = 0
WESTERMO-SW6-MIB::cfgWlan802dot11rR1KHParameter.0 = 0
# 802.11r R0 key holder list wildcard entry
WESTERMO-SW6-MIB::cfgWlan802dot11rR0KHId.0 = 0
WESTERMO-SW6-MIB::cfgWlan802dot11rR0KHEnabled.0 = 1
WESTERMO-SW6-MIB::cfgWlan802dot11rR0KHDestinationMac.0 = ff:ff:ff:ff:ff:ff
WESTERMO-SW6-MIB::cfgWlan802dot11rR0KHID.0 = *
WESTERMO-SW6-MIB::cfgWlan802dot11rR0KHKey.0 = 000102030405060708090a0b0c0d0e0f
# 802.11r R1 key holder list wildcard entry
WESTERMO-SW6-MIB::cfgWlan802dot11rR1KHId.0 = 0
WESTERMO-SW6-MIB::cfgWlan802dot11rR1KHEnabled.0 = 1
WESTERMO-SW6-MIB::cfgWlan802dot11rR1KHDestinationMac.0 = 00:00:00:00:00:00
WESTERMO-SW6-MIB::cfgWlan802dot11rR1KHID.0 = 00:00:00:00:00:00
WESTERMO-SW6-MIB::cfgWlan802dot11rR1KHKey.0 = 000102030405060708090a0b0c0d0e0f
```

The 256-bit R0KH-key `cfgWlan802dot11rR0KHKey` (000102030405060708090a0b0c0d0e0f in the example above) must match the 256-bit R1KH-key `cfgWlan802dot11rR1KHKey`. Please use your own key here and not the one which is given here just as an example to demonstrate the format of such a key.

PMK-R0 Key Holder ID (`cfgWlan802dot11rPmkR0KeyHolderIdentifier`) is configurable via `cfgWlan802dot1xNasId`. See also section 5.13.2.

5.16.5.2 Client (STA) 802.11r Configuration

The following configuration example configures a STA for using FT:

Configuration File Example: FT STA

```
# Config.Format = raw
WESTERMO-SW6-MIB::cfgWlanHoProfile.0 = 2 # t2gv2(2)
WESTERMO-SW6-MIB::cfgWlan802dot11rEnabled.0 = 1
```

5.16.5.3 PMK-R0 / PMK-R1 Key Lifetime

The PMK-R0 lifetime is derived from the Session-Timeout provided by the authentication server, if available, and the PMK-R1 lifetime is derived from PMK-R0 lifetime. In addition the PMK-R0 lifetime can be configured per AP with [cfgWlan802dot11rPmkR0Lifetime](#). In this case the minimum of Session-Timeout and [cfgWlan802dot11rPmkR0Lifetime](#) is used.

It is also possible to enforce daily PMK-R0 / PMK-R1 expiration on a defined time (hour and minute). Enforce PMK-R0 / PMK-R1 expiration means that all the STAs are disconnected and the key material is deleted so that the STAs need to reconnect with full authentication over the RADIUS server.

- [cfgWlan802dot11rExpirationEnabled](#) to enable forced expiration
- [cfgWlan802dot11rExpirationTime](#) to define time (hour:minute) at which PMK-R0s and PMK-R1s expiration is daily forced

5.16.5.4 PMK-R1 push

Everytime a client initiates a connection to an Mobility Domain (MD), the involved AP receives the PMK-R0 from the RADIUS server and then derives a local PMK-R1. At this time this AP may also generate PMK-R1s for every other AP in the MD and send it to the respective AP.

This can be enabled by setting [cfgWlan802dot11rPmkR1Push](#) to 1.

Pushing PMK-R1 works only for APs which have an entry in the [cfgWlan802dot11rR1KHTable](#). Since the MAC-address of the R1 is part of the derived PMK-R1, an AP which has a wildcard configured is not able to generate the PMK-R1.

However it is still desirable to work with wildcard entries in the R0KH/R1KH tables even when using push. When push is enabled and an AP can not find the required PMK-R1 when a client roams to it, it will fall back to pull. Once an R0 AP had PMK-R1 material pulled from it, it learns dynamically the R1 which did the pull. The next time this particular R0 AP tries to push PMK-R1 material it will consider this learned entry.

5.17 Quality of Service (QoS)

The *Software 6* and devices supports Wireless Multimedia Extensions (WME) based on the IEEE 802.11e standard. WME provides basic Quality of service (QoS) features to IEEE 802.11 networks. The WME settings are configured on the Access Point only. Connecting clients are informed upon association what their QoS parameters are. Note that QoS/WME does not provide guaranteed

throughput, but it is suitable for well defined applications that require QoS, such as Voice over IP (VoIP) on Wi-Fi phones (VoWLAN).

The levels of priority in Enhanced Distributed Channel Access (EDCA) are called access categories (ACs). They are used by a WMM-enabled device to control the Arbitration Inter-Frame Space (AIFS), the Contention Window Minimum (CwMin), the Contention Windows Maximum (CwMax), and the Transmit Opportunity (TXOP).

The ACs can be set with [cfgWlanWmeAc](#). The available ACs are:

- background (1)
- besteffort (2)
- video (3)
- voice (4)

All configurable parameters for ACs exist in two types: For the AP itself, and what the connecting STAs has to adapt to.

The AIFS can be set with [cfgWlanWmeAifs](#) and [cfgWlanWmeApAifs](#). A smaller AIFS increases probability of a frame getting a slot on the air to be transmitted. Higher prioritised queues should have smaller AIFS values.

The contention window (CW) can be set with [cfgWlanWmeCwMin](#), [cfgWlanWmeCwMax](#), [cfgWlanWmeApCwMin](#) and [cfgWlanWmeApCwMax](#). It has a similar function as the AIFS value, but adds some randomness between CwMin and CwMax. Queues which are expected to transmit large amounts of traffic should have a wider window with higher values to allow more randomness. Similar, queues with fewer high prioritised traffic should have a small window with low values.

To set the Transmit opportunity use [cfgWlanWmeTxOpMax](#) and [cfgWlanWmeApBurst](#). It specifies how long a given STA/AP is allowed to transmit when it has gained access to the medium.

The default values recommended by the WiFi-Alliance are:

AC	AIFS	CwMin	CwMax	TXOP
BK	7	4	10	0
BE	3	4	10	0
VI	2	3	4	94
VO	2	2	3	47

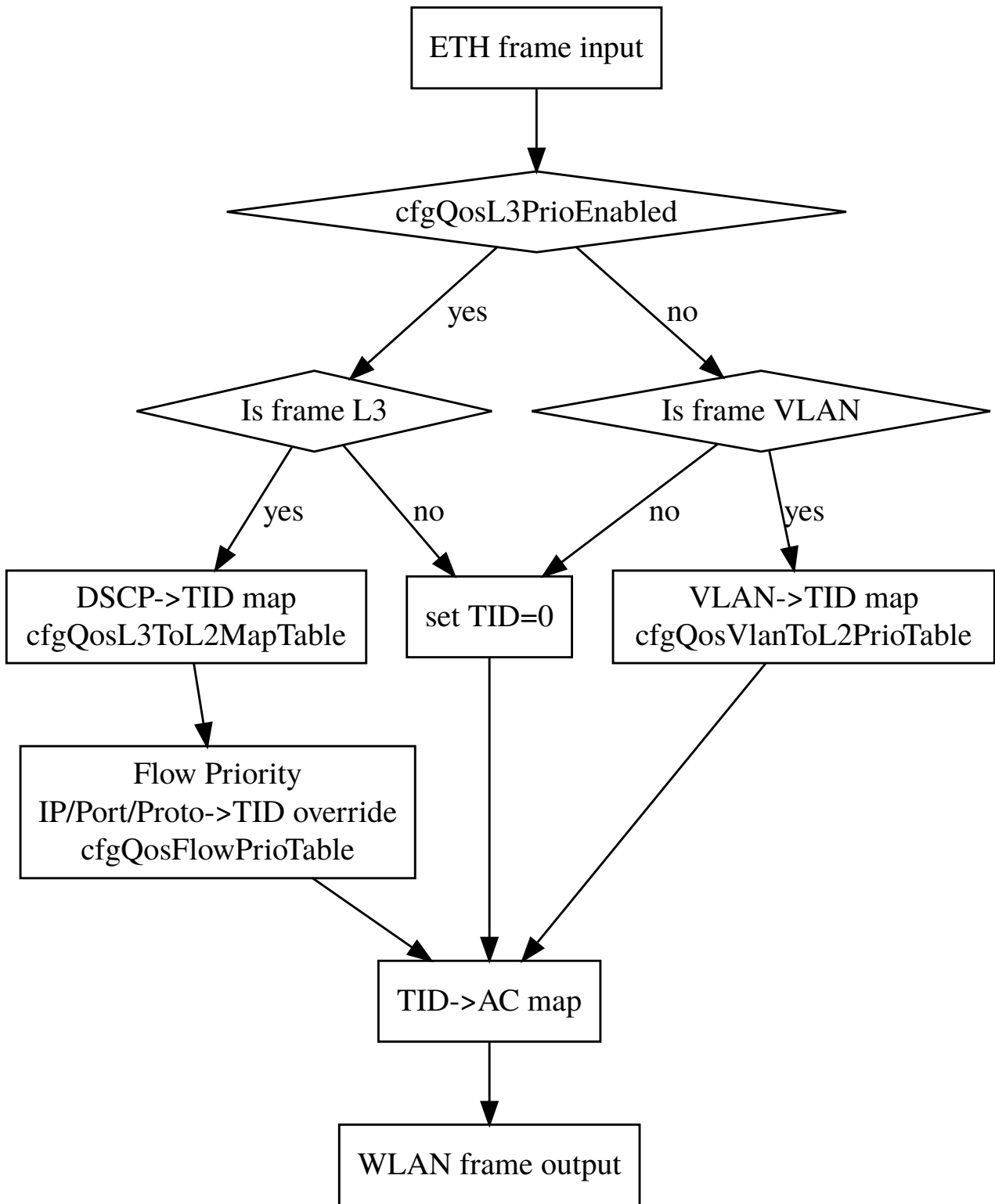
As indicated above, there are 4 hardware queues available with the names: Back Ground, Best Effort, Video and Voice.

To be able to maximise the chance of important traffic going through, it is necessary to match traffic and map it to the appropriate AC.

To achieve this, the following settings and maps are available:

1. [cfgQosL3PrioEnabled](#) - Use layer3, respectively layer2 processing.
- 2a. [cfgQosDscpToTidMapTable](#) - Map from DSCP class selector (IP TOS) to wireless priority (TID).
- 2b. [cfgQosVlanToTidMapTable](#) - Map from layer 2 priorities (802.1p) to wireless priority (TID).
3. [cfgQosIpToTidMapTable](#) - Map from IP header (Src, Dst, Proto, etc.), to wireless priority (TID).

The transmit path of frames with Qos enabled is visualised with the following diagram:



When `cfgQosL3PrioEnabled` is enabled, the DSCP header is mapped to the TID (Traffic Identifier) according to `cfgQosDscpToTidMapTable`. By default it is in a 1:1 manner. The DSCP header consists

of 6 bits. Only the upper most 3 bits, the class selector bits, are considered. The lower 3 bits, the drop probability bits, are ignored. Notice that with the default settings, the class selector 0, which is what most IP frames carry in their header without any configuration, maps to the Best Effort queue. With [cfgQoS L3PrioEnabled](#) enabled, mapping of the DSCP header to wireless queues is always performed even if a different priority is specified in the 802.1p part of an 802.1q VLAN header. After the map from DSCP to TID, it is possible to override the selected TID based on the properties of the IP header (Protocol, Source, Destination, Port, etc.).

When [cfgQoS L3PrioEnabled](#) is disabled, the 802.1p part of the 802.1q tag is mapped to the TID according to [cfgQoS VlanToTidMapTable](#). By default it is in a 1:1 manner.

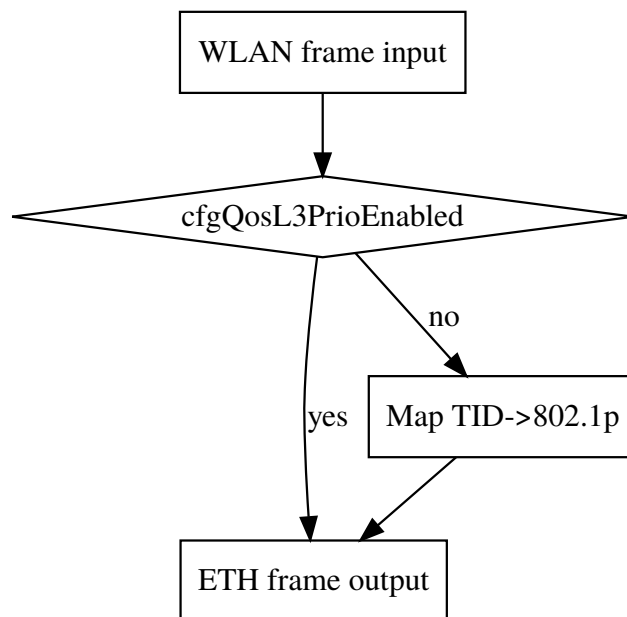
The TID is mapped to the AC classes according to the following table:

TID	AC
0	BE
1	BK
2	BK
3	BE
4	VI
5	VI
6	VO
7	VO

All frames which don't match the specified mode (e.g. non-IP frames when [cfgQoS L3PrioEnabled](#) is enabled) are mapped to the TID = 0 (Best Effort) queue.

The QoS configuration is described in [cfgQoS](#).

The receive path of frames with Qos enabled is visualised with the following diagram:



When L3 processing is disabled, the received TID will be mapped to the 802.1p field in the 802.1q tag.

5.18 Common Address Redundancy Protocol (CARP)

CARP is a networking protocol which allows multiple devices on the same LAN to share a set of IP addresses. For *Software 6* the main use case for CARP is to provide redundant gateways in a redundant mobility application.

The CARP configuration is described in [cfgNetCarpTable](#). The list of managed IP addresses per CARP instance can be configured in the [cfgNetIpTable](#) by setting the [cfgNetIpProto](#) to *carp(5)* and the correct [cfgNetIpCarpId](#)

For redundant mobility application a WLAN demote trigger can be configured using the [Network Link Monitor \(NLM\)](#). If enabled, the CARP instance demotes itself if there is no WLAN link.

5.19 Network Link Monitor (NLM)

The NLM supports several types of monitors (phy, icmp, wlan, ..) and executes so called *actions* on monitor state change from down to up and vice versa.

Supported monitors are documented in [cfgNlmMonType](#).

Supported actions are documented in [cfgNlmMonUpAction](#) and [cfgNlmMonDownAction](#).

Full NLM configuration is described in [cfgNlm](#).

Main use case is to observe the network link from the client (STA), which acts as gateway on the vehicle, to the endpoint in the ground network in a redundant mobility application with two wireless links:

- **Backbone Monitor** - On Access Point, observe the backbone network on the ground by monitoring the linkstate and/or pinging an endpoint in the ground backbone and stop WLAN operation if the endpoint(s) is down.
- **CARP Failover** - On STA, observe the link state of the WLAN interface and trigger CARP group to demote itself if the link is down or otherwise unusable.

These two features combined provides link status check from the train gateway up to the endpoint in the ground backbone.

5.19.1 NLM Monitor Types

5.19.2 phy Monitor

The phy monitor is active when [cfgNlmMonType](#) is set to 0.

This is a polling based monitor. It checks in regular intervals ([cfgNlmMonInterval](#)) the state of a list of comma-separated ethernet phys ([cfgNlmMonInterfaces](#)). E.g. "eth0, eth1". As long as at least one phy is up, the state of the monitor is up. The state will change to down when the checks fails [cfgNlmMonCount](#) times in a row.

5.19.3 icmp Monitor

The icmp monitor is active when [cfgNlmMonType](#) is set to 1.

This is a polling based monitor. It sends icmp requests (ping) in regular intervals ([cfgNlmMonInterval](#)) to the configured IP address in ([cfgNlmMonDestination](#)). The state will change to down when the checks fails [cfgNlmMonCount](#) times in a row. The state will recover as soon as at least a single icmp response (pong) is received.

5.19.4 wlan Monitor

The wlan monitor is active when `cfgNlmMonType` is set to 2.

This is an event based monitor.

The wlan monitor consists of 3 components:

- **Long Handoff Detector** - Triggers when after disassociation no authorization event is detected within the configured time in `cfgNlmMonInterval`.
- **Scan Loop Detector** - Triggers immediately on trap 415. This happens if there is no AP or only a single AP which stays below/above the Handoff thresholds. This trap is only generated when `cfgWlanHoProfile` is set to 2 or higher.
- **Handoff Loop Detector** - Triggers if there have been `cfgNlmMonCount` number of Handoff events within `cfgNlmMonCount * (cfgNlmMonInterval + cfgNlmMonScanLoopInterval)`.

The wlan monitor recovers:

- **Long Handoff Detector** - Immediately after the next successful authorization.
- **Scan Loop Detector** - After the time of the last 415 trap event + the configured `cfgNlmMonScanLoopInterval`. While down, this is rechecked regularly in `cfgNlmMonScanLoopInterval` intervals.
- **Handoff Loop Detector** - When there are less than `cfgNlmMonCount` Handoff events within the time-window `cfgNlmMonCount * (cfgNlmMonInterval + cfgNlmMonScanLoopInterval)`. While down, this is rechecked regularly in `cfgNlmMonScanLoopInterval` intervals.

5.19.5 NLM Configuration for CARP Failover

The following NLM configuration will failover when:

- A Handoff takes longer than 300 ms (`cfgNlmMonInterval`).
- 3 consecutive scans can not find a better target (Trap 415).
- 4 Handoff happen within 13200 ms ($4 * (300 + 3000)$).

Configuration File Example: NLM Configuration for CARP Demote

```
# Config.Format = raw
WESTERMO-SW6-MIB::cfgNlmGlblEnabled.0 = INTEGER: 1
```

```
WESTERMO-SW6-MIB::cfgNlmMonEnabled.0 = INTEGER: 1
WESTERMO-SW6-MIB::cfgNlmMonInterval.0 = INTEGER: 300
WESTERMO-SW6-MIB::cfgNlmMonCount.0 = INTEGER: 4
WESTERMO-SW6-MIB::cfgNlmMonType.0 = INTEGER: 2
WESTERMO-SW6-MIB::cfgNlmMonInterfaces.0 = STRING: wlan0
WESTERMO-SW6-MIB::cfgNlmMonScanLoopInterval.0 = INTEGER: 3000
# Up-/Down action for cfgNetCarpLocalInterfaceGroup.0 = 1
WESTERMO-SW6-MIB::cfgNlmMonUpAction.0 = INTEGER: 1001
WESTERMO-SW6-MIB::cfgNlmMonDownAction.0 = INTEGER: 1001
```

5.19.6 NLM Configuration for Backbone Monitor

Following NLM configuration periodically pings the endpoint with 192.168.1.1 and disables the Access Point operation on wlan0 if the endpoint does not reply.

Configuration File Example: NLM Configuration for Backbone Monitor

```
# Config.Format = raw
WESTERMO-SW6-MIB::cfgNlmGblEnabled.0 = INTEGER: 1
WESTERMO-SW6-MIB::cfgNlmMonEnabled.0 = INTEGER: 1
WESTERMO-SW6-MIB::cfgNlmMonInterval.0 = INTEGER: 1000
WESTERMO-SW6-MIB::cfgNlmMonCount.0 = INTEGER: 3
WESTERMO-SW6-MIB::cfgNlmMonType.0 = INTEGER: 1
WESTERMO-SW6-MIB::cfgNlmMonInterfaces.0 = STRING: br0.vlan0
WESTERMO-SW6-MIB::cfgNlmMonDestination.0 = IpAddress: 192.168.1.1
WESTERMO-SW6-MIB::cfgNlmMonUpAction.0 = INTEGER: 2000
WESTERMO-SW6-MIB::cfgNlmMonDownAction.0 = INTEGER: 2000
```

5.20 DNS/DHCP Server

DNS/DHCP servers can be configured on all network interface.

The DNS/DHCP configuration is described in [cfgDhcp](#)

Use [cfgDhcpScopeDhcpOptions](#) to configure arbitrary DHCP options.

Configuration File Example: DHCP server on br0.vlan0

```
# Config.Format = raw
WESTERMO-SW6-MIB::cfgDhcpGlobalEnabled.0 = 1
WESTERMO-SW6-MIB::cfgDhcpDnsmasqScopeParameter.0 = 0
WESTERMO-SW6-MIB::cfgDhcpScopeId.0 = 0,
WESTERMO-SW6-MIB::cfgDhcpScopeInterface.0 = br0.vlan0
```

```
WESTERMO-SW6-MIB::cfgDhcpScopeStart.0 = 100
WESTERMO-SW6-MIB::cfgDhcpScopeLimit.0 = 150
WESTERMO-SW6-MIB::cfgDhcpScopeGateway.0 = 192.168.1.1
```

5.21 Service Indicators and Counters

5.21.1 SNMP Trap

The WLAN modem provides notifications by means of SNMP Traps.

In order to receive SNMP traps on a trap server the trap daemon of the WLAN modem must be enabled ([cfgSnmpTrapEnabled](#)) and the trap destination IP address ([cfgSnmpTrapDest](#)) must be set to the IP address of the trap server.

The SNMP traps are defined and described in the WESTERMO-TRAP-MIB file, which is part of the delivered software package. The trap message string contains message codes in the form:

```
[ <prio> <code> ] <text message>
```

Please refer to chapter [Message Codes](#) for a complete list of all message codes.

All SNMP traps are also recorded in the Syslog (see [Logging Features](#))

5.21.2 Counters and Status

The WLAN modem provides status information and important counters defined and described by the WESTERMO-SW6-MIB.

The status information and counters are logically divided into

- **Hardware status and counters ([hardware](#))** - such as product type, serial number, revision, etc
- **Software status and counters ([software](#))** - such as firmware name, firmware revision, wireless counters etc

5.22 Logging Features

The *Software 6* supports extensive logging features into Syslog. The Syslog is available in the Web Interface / via Web API or it can be sent to a remote server (RFC 5424).

Up to four remote syslog instances can be configured. For enabling/disabling remote instances logging use [cfgLogRemoteEnabled](#). IP address and port of remote syslog instance can be configured using [cfgLogRemoteIp](#) and [cfgLogRemotePort](#). The remote syslog protocol (UDP or TCP) can be configured with [cfgLogRemoteProtocol](#)

There are three message types which are differentiated for the remote syslog:

- **standard**: This type contains all message which are not marked
- **security**: This type contains all message which are marked as security messages (LOCAL0)
- **standard**: This type contains all message which are marked as commissioning messages (LOCAL1). Please refer to [cfgWlanDbgTable](#) for more information.

For the four remote syslog servers the message types which shall be sent can be configured by [cfgLogRemoteType](#). Since the configuration of the message type is a bitfield, all combinations are possible.

Remote Syslog is essential to analyze mobility applications during commissioning (see [Mobility Logging](#)).

5.23 Wireless Link Monitor

The *Software 6* provides wireless link status message periodically to Syslog or as SNMP Trap.

In Access Point mode the link status for all connected clients is reported. In client (STA) mode the current connection to the Access Point is reported.

The feature can be enabled for each WLAN interface separately by [cfgWlanDbgLinkmonitor](#) where the reporting interval is defined by [cfgWlanGlbLinkmonitorInterval](#) globally.

Format:

```
INFO 130 interface|mac|inactive time||rx bytes|rx packets|tx bytes|tx
  ↳ packets|tx retries|tx failed||signal combined|avg signal combined|
  ↳ signal ch0|avg signal ch0|signal ch1|avg signal ch1|signal ch2|avg
  ↳ signal ch2||rx bitrate mode|rx bitrate value|tx bitrate mode|tx
  ↳ bitrate value
```

One message per connection is reported.

5.24 Inter-Carriage Link (ICL)

The Inter-Carriage Link application (ICL) offers a hands-off approach to connect and bridge carriage networks. Once ICL is enabled and operational on two carriages, the application will do the following:

- Broadcast availability to other ICL capable carriages while outside detection range.
- Automatically form a link on approach.
- Stay linked while the carriages are connected and providing the highest throughput possible.
- Cleanly disconnect on departure and switch back to broadcasting availability.

The ICL application removes the need for a cable connection and is designed to be low maintenance. Using the ICL algorithm combined with suitable antennas ensures proper linking of carriages.

5.24.1 Configuration of the Inter-Carriage Link Application

The ICL application supports two configuration modes:

- A Web Interface as a configuration wizard.
- Configuration through SNMP for full customisation.

5.24.1.1 Configuration with the Web Interface

After logging in (See [Web-Based Management \(Web Interface\)](#) how to access), the *Inter-Carriage Link* application is available in the *Applications* menu.

The first step to activate the application is to enable it. Go to *Application -> Inter-Carriage Link -> Configuration*.

You should see a page like the screenshot in figure [5.12](#).

Inter-Carriage Link - Configuration

Welcome to the Inter-Carriage Link application.

This page will guide you through the needed configuration to setup a working Inter-Carriage Link.

Enable the Inter-Carriage Link application.

The Inter-Carriage Link is disabled at the moment. To enable it please press the following button.

Attention: Be warned that every not applied configuration will be reverted!

Enable Inter-Carriage Link

Figure 5.12: Enable page

By pressing the enable button the Web Interface wizard will change a number of settings for you. You will be able to adjust various parameters before starting the ICL application.

Once enabled, you will be redirected to the configuration page as shown in figure 5.13 where you will be able to customise a list of settings.

Inter-Carriage Link - Configuration

Welcome to the Inter-Carriage Link application.

This page will guide you through the needed configuration to setup a working Inter-Carriage Link.

Settings

Network

Interface assignment eth0: Data / eth1: Admin

Admin IP 192.168.1.20/24

Admin VLAN ID 0

Wireless

Country Code EU

ICL SSID NT_ICL-

Antenna Gain (dBi) 2

Bandwidth (MHz) HT40+

Frequency (MHz) Auto 5 GHz

Inter-Carriage Link settings

Connection threshold -60 dBm (-90 - 0)

Connection delay 30 seconds (0 - 600)

Disconnection threshold -65 dBm (-90 - 0)

Disconnection delay 20 seconds (1 - 600)

Cycle time 5 seconds (2 - 60)

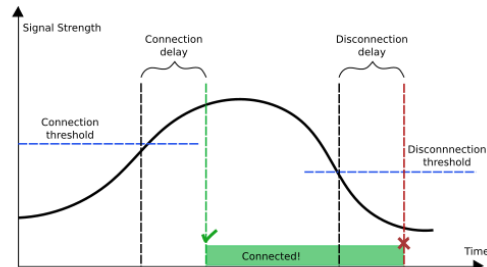
Blacklist time 600 seconds (1 - 3600)

Suspended

Frequencies (MHz) 5180 Add

Remove all invalid frequencies

5745 MHz	Remove
5785 MHz	Remove
5825 MHz	Remove



Apply

Figure 5.13: Configuration page

Interface assignment defines which interfaces will be configured as *data* and which as *admin* interface. Please make sure you choose the right interfaces. Otherwise you probably will not be able to connect to the device again.

Admin IP defines which IP address will be assigned to the *admin* interface.

Admin VLAN ID defines which VLAN id will be used for the *admin* interface.

ICL SSID defines which prefix will be used for the SSID. To established a link between two carriages this prefix must be the same on both sides.

Antenna Gain defines the antenna gain use for the Inter-Carriage link.

Bandwidth defines the preferred bandwidth. Options are HT20, HT40+ or HT40-. For more details, see [cfgWlanDevBandwidth](#).

Frequency defines the primary operating frequency and depends on the selected bandwidth. See [cfgWlanDevFrequency](#) for more details.

Frequency list contains all frequencies that will be scanned by the Inter-Carriage Link application. It also functions as a list of backup frequencies in case radar is detected on the current operating frequency. This list should include its own frequency the device operates on. Corresponding MIB entries are [cfgWlanFFreq0](#) to [cfgWlanFFreq23](#).

Connection threshold describes the minimum signal level a potential ICL partner needs to reach before it is considered a valid ICL partner. A higher value means the signal needs to be stronger and therefore a potential partner needs to be closer. [cfgIclConnectionThreshold](#) is the associated MIB entry.

Connection delay defines how long the Inter-Carriage Link algorithm should evaluate a potential ICL partner. If the connection delay is set to 0 the ICL algorithm will instantly connect to the very first potential partner circumventing most of the evaluation of the ICL algorithm. See [cfgIclConnectionDelay](#) for more details.

Disconnection threshold sets the signal level at which the link disconnection process will be started. For more details, see [cfgIclDisconnectionThreshold](#).

Disconnection delay defines how long the device should wait before scanning for a new partner once the old partner disconnected. It also defines how long a formerly connected partner tries to reconnect if the link is lost for any reason. [cfgIclDisconnectionDelay](#) is the corresponding MIB entry.

Cycle time sets the scan interval. Ideally the connection delay time is at least five times the cycle time to allow the ICL algorithm to properly evaluate a candidate. See [cfgIclCycleTime](#) for more details. Also consider the size of the Frequency list. A larger list requires more time spend scanning, thus the Cycle time should not be selected too short.

Blacklist time defines how long a blacklist entry will reside in the blacklist.

Suspended defines if the Inter-Carriage Link starts in the suspended state.

To start the Inter-Carriage Link application press *Apply*. If the application was already running, *Apply* will restart the service with the changed configuration.

Note Configuration with SNMP offers full customisation should the Web Interface not satisfy your configuration needs.

The status page as in figures 5.14, 5.15, 5.16 and 5.17 shows all important status information regarding the Inter-Carriage Link. This page will automatically refresh every two seconds to keep the status up-to-date.

Inter-Carriage Link - Status

Auto refresh every 2 seconds



My carriage

MAC:	00:07:7c:30:60:3f
Operation mode:	AP
Frequency (MHz):	5785
Status:	scanning
Operation:	<input type="button" value="Suspend"/>

Figure 5.14: Status page: Scanning for an ICL partner

Inter-Carriage Link - Status

Auto refresh every 2 seconds



My carriage

MAC:	00:07:7c:30:5f:af
Operation mode:	AP
Frequency (MHz):	5745
Status:	scanning
Operation:	<input type="button" value="Suspend"/>


Candidates

MAC	Avg. Signal (dBm)	Last Signal (dBm)
00:14:5a:03:0a:bc	-31	-31

Figure 5.15: Status page: Evaluating one or more ICL partners

Inter-Carriage Link - Status

Auto refresh every 2 seconds




My carriage	
MAC:	00:07:7c:30:5f:af
Operation mode:	STA
Frequency (MHz):	5785
Status:	connected
Operation:	<input type="button" value="Suspend"/>

Partner carriage	
MAC:	00:14:5a:03:0a:bc
Signal (dBm):	-30
<input type="button" value="Force disconnect"/>	

Inter-Carriage Link - Status

Auto refresh every 2 seconds



My carriage	
MAC:	00:07:7c:30:60:3f
Operation mode:	
Frequency (MHz):	
Status:	suspended
Operation:	<input type="button" value="Resume"/>

Figure 5.17: Status page: ICL operation suspended

Suspend

If the ICL functionality isn't needed at the moment the operation can be suspended. This can be accomplished by pressing the *Suspend* button or by setting `setIclSuspended` to 1. In this state the device will no longer broadcast availability and other partners are not able to connect.

Blacklist

Partners listed in the blacklist can't be used to establish a Inter-Carriage Link. The blacklist entries can be flushed by pressing the *Clear Blacklist* button or by setting `rpclIclClearBlacklist` to 1.

Force Disconnect

In the unlikely case the displayed connection is not the desired connection, the link can be dropped using the *Force disconnect* button or by setting `rpclclForceDisconnect` to 1. The current ICL partner will then be added to a blacklist to prevent reconnecting to the same device for the time configured with the *Blacklist time*.

5.25 Public Wireless Network (PWN)

5.25.1 Hotspot

The RT-610 product is very suitable for WiFi hotspots. It supports amongst other things dual concurrent operation in the 2.4GHz and 5GHz wireless bands, Multi-SSID mode and MU-MIMO.

With the Hotspot application as show in Figure 5.18 the configuration of a WiFi hotspot is simple.

Hotspot

Management Network

Enable Remote Mgmt
 Enable Local Mgmt
 Local Mgmt VLAN ID:
0 = No VLAN, Tagged X1
 Local Mgmt IP address:
 Default Gateway:

Network Configuration

Radio Configuration

Radio	Modulation	Bandwidth (MHz)	Frequency (MHz)	Power (dBm)	Antenna Gain (dBi)	Antenna Mask
5.0 GHz Band radio0	ac	bw80	Auto	15	6	15(1111)
2.4 GHz Band radio1	ng	bw20	2412 MHz	15	6	3(0011)

Public Network

Supported Bands: Both (2.4 GHz / 5 GHz)

SSID:

Hide SSID:

Isolate Clients:

Encryption:

VLAN ID: 0 = No VLAN, Tagged X2

Private Network

Supported Bands: Both (2.4 GHz / 5 GHz)

SSID:

Hide SSID:

Isolate Clients:

Encryption:

Encryption passphrase:

VLAN ID: 0 = No VLAN, Tagged X2

Figure 5.18: Hotspot application configuration page

After logging in (See [Web-Based Management \(Web Interface\)](#) how to access), the *Hotspot* application is available in the *Applications* menu.

5.25.2 Band Steering

Band Steering is a feature that encourages wireless clients that are capable of both 5GHz and 2.4GHz wireless bands to connect to the faster 5GHz band and leave the 2.4GHz band less-crowded for those clients who support the 2.4GHz band only. Inactive clients of a traffic overloaded 2.4GHz wireless band may be steered to the less-congested 5GHz band or vice versa. If the received signal strength indicator (RSSI) of the connected 5GHz band is very low and a stronger 2.4GHz band is available, the client will be steered to the 2.4GHz band until the 5GHz band has regained strength. Such an approach is expected to improve the wireless performance for all clients participating in the same Extended Service Set (ESS).

Since Band Steering is done within the same ESS, both bands must have the same ESS Identifier (ESSID) as well as the same security settings. That ESSID is called Matching SSID and is configurable by the MIB item [cfgWlanBsteerMatchingSsid](#). To enable Band Steering for multiple virtual access points, the matching SSID names can be defined as a comma-separated (and/or space-separated) list.

5.25.2.1 Basic Operation

Assuming that the Software 6 product is configured with the factory settings, enabling the Band Steering is fairly simple by applying the following configuration file.

Configuration File Example: Basic Band Steering

```
# Config.Format = raw
WESTERMO-SW6-MIB::cfgWlanInterfaceSsid.0 = BSteer-Enabled
WESTERMO-SW6-MIB::cfgWlanInterfaceSsid.1 = BSteer-Enabled
WESTERMO-SW6-PWN-MIB::cfgWlanBsteerEnabled.0 = 1
WESTERMO-SW6-PWN-MIB::cfgWlanBsteerMatchingSsid.0 = BSteer-Enabled
```

5.26 Global Navigation Satellite System

All products that support cellular communication come with a Global Navigation Satellite System (GNSS), see section [1.2](#). The built-in GNSS module is monitored by the service application `gpsd`. This allows the location data to be distributed across computer networks by TCP/IP.

To enable `gpsd` use the following config example:

Configuration File Example: Enable gpsd

```
# Config.Format = raw
WESTERMO-SW6-GNSS-MIB::cfgGnssGpsdEnabled.0 = 1
WESTERMO-SW6-GNSS-MIB::cfgGnssGpsdAddress.0 = 0.0.0.0:2974
```

After enabling the `gpsd`, the location data is visible on the status page of the Web Interface or available via SNMP [swGnss](#). A common way to access the data is to open a TCP connection on the defined port with a `gpsd` compatible client (`gpsmon`, `xgps` or `gpspipe` on Linux).

Note: The `gpsd` can only be bound locally or globally so the value of [cfgGnssGpsdAddress](#) must be 127.0.0.1 for local or 0.0.0.0 to bind to all addresses.

5.26.1 GNSS Device Configuration

The GNSS module is able to operate 3 satellite systems simultaneously, the following are selectable:

- GPS
- Glonass
- BeiDou
- Galileo

Detailed information on how to select a valid combination can be found in [cfgGnssDevSatelliteSystems](#). The parameter [cfgGnssDevMeasurementPeriod](#) defines the overall measurement period of the GNSS module. The effective sampling rate of each message can be set by the respective parameter [cfgGnssDevMsgsNmeaRate](#) or [cfgGnssDevMsgsUbxRate](#).

$$messageSamplingRate[Hz] = \frac{cfgGnssDevMsgsNmeaRate}{cfgGnssDevMeasurementPeriod[m.s]} \times 1000 \quad (5.1)$$

Normally, there is no need to change the sampling rate for each message. It is sufficient to select the overall measurement period and set the individual sampling rate of the messages to 1.

5.26.2 NMEA Sentences

A default setting of National Marine Electronics Association (NMEA) sentences is activated by default in order to enable a location determination when the service is switched on. All NMEA sentences

available through the GNSS module are provided and listed in the description of [cfgGnssDevMsgsNmeaType](#). In order to disable specific sentences, set the related [cfgGnssDevMsgsNmeaRate](#) to 0.

5.26.3 UBX Messages

UBX is a proprietary protocol from the company u-blox for exchanging GNSS data. All UBX messages are disabled by default. To enable a UBX message, select a message type [cfgGnssDevMsgsUbxType](#) and set the [cfgGnssDevMsgsUbxRate](#) greater than 0. To disable a message, set the [cfgGnssDevMsgsUbxRate](#) to 0. The accessible UBX messages are listed in the description of [cfgGnssDevMsgsUbxType](#).

5.27 RSTP

All RSTP related configurations are described under [cfgRstpBridge](#) and [cfgRstpPort](#).

RSTP configuration is done via the MIB [WESTERMO-SW6-BRIDGE-MIB](#).

Status information is available via Web Interface (Status - RSTP) or via the standard MIB BRIDGE-MIB (oid .1.3.6.1.2.1.17).

Please note that currently only one RSTP instance is supported and this instance is br0.

5.28 Dynamic Frequency Selection (DFS)

All devices have a basic way to enable support for DFS:

- Wireless Standalone

802.11n capable devices with a monitor card, have 2 additional ways:

- NWM
- AFM/AFC

Wireless Standalone Wireless Standalone mode can be used on a device without monitor card. This allows operation on DFS frequencies but has the downside that if radar is detected, there is an interruption of at least 1 minute.

Refer to [Wireless Standalone](#) for detailed information on operation and configuration.

NWM The NWM is intended to be used on a single AP which has a monitor card to allow seamless operation in case of radar.

Refer to [Wireless Manager \(NWM\)](#) for detailed information on operation and configuration.

AFM/AFC The AFM/AFC is intended to be used on groups of APs which have monitor cards to allow seamless operation in case of radar.

Refer to [Area Frequency Management \(AFM\)](#) for detailed information on operation and configuration.

5.28.1 Wireless Standalone

The Wireless Standalone mode is automatically active when a DFS frequency is configured but neither NWM nor AFM/AFC are active.

In Wireless Standalone mode the communication radio is used to perform the CAC to clear the frequency for operation. The communication radio is not able to do off-channel-CAC, thus when radar is detected, it has to clear a new frequency first before operation can continue. Depending on the frequency this takes at least 1 minute; when radar is detected the interruption will be at least 1 minute. The Wireless Standalone mode also does not have access to the non-volatile memory option (`cfgChMgrDfsUseNvram`). Clearing of the channel is required everytime the configuration is changed.

In case radar is detected, the radio will jump to a new frequency based on the selected ACS list (see [cfgWlanIfaceAcsList](#)). When no frequency list is selected (e.g. `cfgWlanIfaceAcsList` set to -1) all frequencies permitted by the country code are considered for selection. Please note that this may also include frequencies which are not allowed in the installed environment, e.g indoor frequencies in an outdoor installation. Please make sure to always specify and configure a correct frequency list.

Keep in mind that once the frequency has been changed, the AP will stay on the new frequency until radar is detected again. It will not automatically revert back to it's original operating frequency once the NOP time has elapsed. This also implies that if the new frequency the AP jumped to is not a DFS frequency, the AP will stay there until it is manually reconfigured.

5.28.2 Wireless Manager (NWM)

The Wireless Manager (NWM) offers advanced features for wireless access point operation

- Usable frequency list to support advanced frequency planning
- Background channel availability check (CAC) for DFS channel to support seamless operation in case of radar detection
- Available channel list in non-volatile memory for fast recovery after reboot

- Interference reports

The Wireless Manager (NWM) is only available on RT-370 devices since it requires two radios.

The first radio (antenna ports A1, A2 and A3) is the so called communication interface providing the access point functionality. The second radio (antenna port A4) is responsible to make Channel Availability Checks on channels which are not yet available. During periods where no further CACs are required, the second radio can be used to do scan the environment for interferences as described in section [5.29](#)

Antenna port A4 is used for CAC and Off-Channel-CAC. Therefore you need to assert that you have an antenna connected at antenna port A4.

The NWM can store the list of available DFS channels in a non-volatile memory. Thus, when the channel is once marked as available it will be instantly available after a device reboot. When the non-volatile memory option is enabled (see [cfgChMgrDfsUseNvram](#)), an Operator is responsible to reset the list of available DFS channel by setting [rpcNvramFreqStatesReset.0](#) to zero at installation or re-installation of the device.

The NWM depends on following sub-features:

- Channel Manager: The Channel Manager is responsible to perform CACs on usable DFS channels. Its goal is to make all DFS channels available. Further, it proposes the wireless channel to be used by the NWM.
- Scan Worker: The Scan Worker manages the second radio of the RT-370 device. On request it performs scan work jobs like CAC or wireless interference scans. The Channel Manager makes use of the Scan Worker to do the CACs on DFS channels.

5.28.2.1 Configuration with the Web Interface

The *Wireless Manager* can be configured with the Web Interface (See [Web-Based Management \(Web Interface\)](#) how to access).

The NWM is a so called application. You will find it in the *Applications* menu.

The first step to activate the application is to enable it. Go to *Application -> Wireless Manager -> Configuration*.

You should see a page like the screenshot in figure [5.19](#).

Wireless Manager - Configuration

Welcome to the Wireless Manager.

This page will guide you through the needed configuration to setup a proper working Wireless Manager.

Enable the Wireless Manager

The Wireless Manager is disabled at the moment. To enable it please press the following button.

Attention: Be warned that your configuration will be overwritten!

Enable Wireless Manager

Figure 5.19: Enable page

By pressing the enable button the following settings will be changed and the NWM started.

Setting	Value
cfgNwmEnabled.0	1
cfgChMgrEnabled.0	1
cfgWlanDevModulation.0	12
cfgWlanDevFrequency.0	5260
cfgWlanDevBandwidth.0	0
cfgWlanIfaceScanList.0	0
cfgNetWlanEnabled.0	1
cfgNetWlanEnabled.1	1
cfgWlanIfaceMode.1	2
cfgChMgrUsableFrequencyList.0	0
cfgWlanFFreq0.0	5260
cfgWlanFFreq1.0	5280
cfgWlanFFreq2.0	5300
cfgWlanFFreq3.0	5320
cfgWlanFFreq4.0	5500
cfgWlanFFreq5.0	5520
cfgWlanFFreq6.0	5540
cfgWlanFFreq7.0	5560
cfgWlanFFreq8.0	5580
cfgWlanFFreq9.0	5660
cfgWlanFFreq10.0	5700

Table 5.2: Default values for the NWM

It takes some time to enable the NWM. If the NWM could be successfully started, the configuration page as in figure 5.20 should be shown.

Wireless Manager - Configuration

Welcome to the Wireless Manager.

This page will guide you through the needed configuration to setup a proper working Wireless Manager.

Settings

Wireless Manager settings

Bandwidth (MHz)

Frequency (MHz)

Frequencies (MHz)	5180 (HT20 / HT40+)	Add
5260 MHz	<input type="checkbox"/>	Remove
5280 MHz	<input type="checkbox"/>	Remove
5300 MHz	<input type="checkbox"/>	Remove
5320 MHz	<input type="checkbox"/>	Remove
5500 MHz	<input type="checkbox"/>	Remove
5520 MHz	<input type="checkbox"/>	Remove
5540 MHz	<input type="checkbox"/>	Remove
5560 MHz	<input type="checkbox"/>	Remove
5580 MHz	<input type="checkbox"/>	Remove
5660 MHz	<input type="checkbox"/>	Remove
5680 MHz	<input type="checkbox"/>	Remove
5700 MHz	<input type="checkbox"/>	Remove

Apply

Figure 5.20: Configuration page

Bandwidth This define the preferred bandwidth. You can choose HT20, HT40+ or HT40-.

Frequency This define the preferred frequency to use. This depend on the selected bandwidth.

Frequency list The frequencies in this list are used as avoiding-possibility if a radar was detected on your preferred frequency.

In HT40+/HT40- you should always add the frequency and the extended frequency: For HT40+, 5300MHz you should add 5300MHz and 5320MHz.

Attention: If the frequency list is empty, all available frequencies will be used. The available frequencies depend on your country code.

To change the configuration and restart the services press *Apply*.

The status page as in figure 5.21 shows all important status informations about the NWM. The page will be automatically refreshed so you will always see the actual status.

Wireless Manager - Status

Wireless Manager status

Auto refresh every 2 seconds

Operation

Frequency: 5260 MHz
HT mode: HT20

Frequency status list

Frequency	HT20	HT40+	HT40-
5260 MHz			
5280 MHz			
5300 MHz			
5320 MHz			
5500 MHz			
5520 MHz			
5540 MHz			
5560 MHz			
5580 MHz			
5660 MHz			
5680 MHz			
5700 MHz			

Legend

	Frequency is available
	Frequency is not available
	Operation frequency

Figure 5.21: Status page

5.28.3 Area Frequency Management (AFM)

Area Frequency Management (AFM) enables communication between Access Points in a mobility application. AFM is mainly used in DFS enabled environments where frequency management over multiple Access Points is required to cope with radar escapes, i.e. when the frequency changes due to detection of a Radar pattern. The main tasks of the AFM is to update the neighbour lists of the Access Points, the so called Dynamic Neighbour List (DNL). The function of Dynamic Neighbour List is described in [Inter-AP Roaming](#)

5.28.3.1 Functional Principle

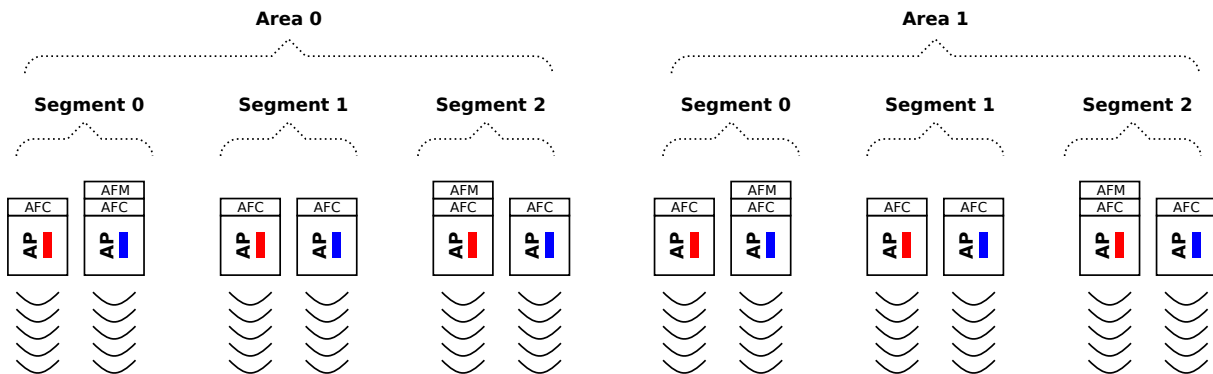


Figure 5.22: Area Frequency Management Principle

The logical structure behind Areas Frequency Management is a line. Due to performance and availability reasons the line can be split into one or more Areas, where each Area is controlled by an AFM (Area Frequency Manager) instance. The AFM instance can be run on any of the Access Point within the Area or on a device in the same IP subnet. An Area is further split into Segments, where one or more Access Points can be assigned to a Segment. The common property of a Segment is the operation frequency (i.e. the AFM assigns the same frequency to all Access Points in the Segment). Each Access Point in the Segment is controlled by an AFC (Area Frequency Client) instance. The Segments' positions within its Area are identified by their Segment index. The AFM communicates with all AFCs within his Area to exchange the required information.

An AFM also communicates with its adjacent AFMs (Area index +/- 1) to exchange information about the Segments. This allows the AFMs to generate Dynamic Neighbor Lists for its AFCs which includes information from neighbour Areas. It is also possible to define a redundant AFM instance for each Area. This AFM instance will take over if the primary AFM instance fails.

The AFM computes the Dynamic Neighbour List (DNL) for each AFC in its Area. A client (STA) connected to the Access Point can request this DNL and use for the next handover. The Access Point sends updates on DNL changes. The DNL is defined by the Neighbour Offset table for each AFC. The offset is relative to the AFC on which it is defined, i.e. the delta to the Segment index. A typical DNL is shown in figure 5.23. In this example the DNL for AP5 contains information (i.e. frequency, BSSID, etc.) from AP3 to AP7.

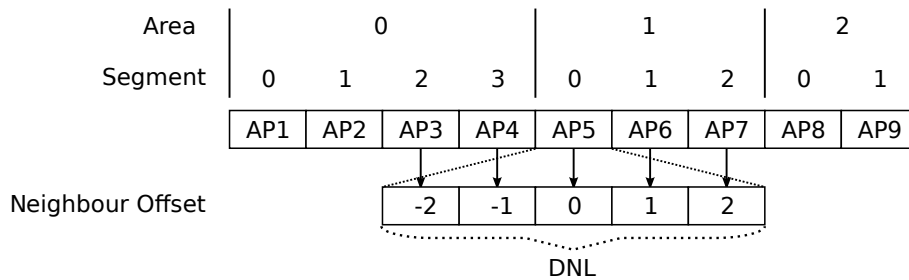


Figure 5.23: Area, Segments in relation to Neighbour Offset and DNL

The nominal frequency of a Segment is defined by the `cfgWlanDevFrequency` configured on the Access Point. If this frequency is available, it will be used as operating frequency. If an Access Point detects a Radar pattern on the operating frequency, then the AFC informs the AFM which will select another operating frequency for the Segment. The AFM selects the escape frequency based on runtime status and chosen frequency plan algorithm. Those frequencies being available at the relevant segment index are evaluated by the rule set to select the optimum escape frequency. The rule set supports preferring larger operation frequency distance between adjacent Access Points. As soon as the nominal frequency becomes available again, the AFM will set it as operating frequency again.

The first radio (antenna ports A1, A2 and A3) is the so called communication interface providing the AP functionality. The second radio (antenna port A4) is responsible to make Channel Availability Checks on channels which are not yet available. During periods where no further CACs are re-

quired, the second radio can be used to do scan the environment for interferences as described in section [Interference Detection Function \(IDF\)](#)

Antenna port A4 is used for CAC and Off-Channel-CAC. Therefore you need to assert that you have an antenna connected at antenna port A4.

5.28.3.2 Configuration

The AFM configuration is described in [cfgAfm](#), [cfgAfc](#) and [cfgChannelCleaner](#). The following configuration steps are required on the Access Points:

1. Configure WLAN in Access Point mode by [cfgWlanInterfaceMode](#).
2. Configure a common set of wireless parameters: [cfgWlanInterfaceSsid](#), [cfgWlanInterfaceEncryption](#), etc.
3. Enable the mandatory services: AFC [cfgAfcEnabled](#) and ChannelCleaner [cfgChanCleanEnabled](#).
4. Define the list of usable frequencies available for the system by [cfgChanCleanUsableFrequencyList](#).
5. Define the nominal frequency plan by assigning a nominal frequency for each Access Point by [cfgWlanDevFrequency](#).
6. Define Areas:
 - a) Select one Access Point within each area which shall run the primary AFM instance by [cfgAfmEnabled](#) and [cfgAfmPrimary](#).
 - b) Select another Access Point within each area which shall run the redundant AFM instance by [cfgAfmEnabled](#) and [cfgAfmPrimary](#). Secondary AFM is optional.
 - c) Define the Area index for primary and secondary AFM instance by [cfgAfmIndex](#).
 - d) Define the Area size for primary and secondary AFM instance by [cfgAfmAreaSize](#). The Area size defines the number of Segments within an Area
 - e) For each AFM instance define the list of primary and secondary neighbour AFMs (max 4, primary/secondary to the right and primary/secondary to the left) by [cfgAfmNeighbourTable](#).
 - f) For primary and secondary AFM define the same list of AFC to be managed by [cfgAfmAfcTable](#).

7. Within each Area define Segments:

- a) Assign the Segment (index) to each Access Point by `cfgAfcIndex`.
- b) For each AFC define its primary and secondary AFM by `cfgAfcAfmTable`.
- c) For each AFC define the relative offset of neighbour Access Points which shall appear in the Dynamic Neighbour List by `cfgAfcNeighbourOffsetTable`.

5.28.3.3 Example

Area Segment	0				1		
	0	1	2	3	0	1	2
	AP0	AP1	AP2	AP3	AP4	AP5	AP6
IP	.10	.11	.12	.13	.14	.15	.16
AFC	afc0	afc1	afc2	afc3	afc4	afc5	afc6
Primary AFM	afm0p				afm1p		
Redundant AFM		afm0r				afm1r	
Nominal Freq.	5500	5580	5700	5500	5580	5700	5500

Figure 5.24: Example AFM configuration

AFM Configuration:

Parameter	AP0	AP1	AP2	AP3	AP4	AP5	AP6
cfgAfmEnabled.0	1	1	0	0	1	1	0
cfgAfmName.0	afm0p	afm0r	-	-	afm1p	afm1r	-
cfgAfmPrimary.0	1	0	-	-	1	0	-
cfgAfmIndex.0	0	0	-	-	1	1	-
cfgAfmAreaSize.0	4	4	-	-	3	3	-
cfgAfmRedundantName.0	afm0r	afm0p	-	-	afm1r	afm1p	-
cfgAfmRedundantIp.0	.11	.10	-	-	.15	.14	-
cfgAfmNeighbourName.0	afm1p	afm1p	-	-	afm0p	afm0p	-
cfgAfmNeighbourIp.0	.14	.14	-	-	.10	.10	-
cfgAfmNeighbourName.1	afm1r	afm1r	-	-	afm0r	afm0r	-
cfgAfmNeighbourIp.1	.15	.15	-	-	.11	.11	-
cfgAfmAfcName.0	afc0	afc0	-	-	afc4	afc4	-
cfgAfmAfcIp.0	.10	.10	-	-	.14	.14	-
cfgAfmAfcName.1	afc1	afc1	-	-	afc5	afc5	-
cfgAfmAfcIp.1	.11	.11	-	-	.15	.15	-
cfgAfmAfcName.2	afc2	afc2	-	-	afc6	afc6	-
cfgAfmAfcIp.2	.12	.12	-	-	.16	.16	-
cfgAfmAfcName.3	afc3	afc3	-	-	-	-	-
cfgAfmAfcIp.3	.13	.13	-	-	-	-	-

AFC Configuration:

Parameter	AP0	AP1	AP2	AP3	AP4	AP5	AP6
cfgAfcEnabled.0	1						
cfgAfcName.0	afc0	afc1	afc2	afc3	afc4	afc5	afc6
cfgAfcIndex.0	0	1	2	3	0	1	2
cfgAfcBackupFreq.0	5180						
cfgAfcAfmName.0	afm0p	afm0p	afm0p	afm0p	afm1p	afm1p	afm1p
cfgAfcAfmIp.0	.10	.10	.10	.10	.14	.14	.14
cfgAfcAfmName.1	afm0r	afm0r	afm0r	afm0r	afm1r	afm1r	afm1r
cfgAfcAfmIp.2	.11	.11	.11	.11	.15	.15	.15
cfgAfcNeighbourOffset.0	-	-	-2	-2	-2	-2	-2
cfgAfcNeighbourOffset.1	-	-1	-1	-1	-1	-1	-1
cfgAfcNeighbourOffset.2	0	0	0	0	0	0	0
cfgAfcNeighbourOffset.3	1	1	1	1	1	1	-
cfgAfcNeighbourOffset.4	2	2	2	2	2	-	-

Common Configuration:

Parameter	AP0	AP1	AP2	AP3	AP4	AP5	AP6
cfgWlanDevModulation.0				12			
cfgWlanDevModulation.1				12			
cfgWlanDevFrequency.0	5500	5580	5700	5500	5580	5700	5500
cfgWlanIfaceMode.0				0			
cfgWlanIfaceMode.1				2			
cfgWlanIfaceNeighbourReport.0				1			
cfgWlanFFreq0.0				5180			
cfgWlanFFreq1.0				5500			
cfgWlanFFreq2.0				5520			
cfgWlanFFreq3.0				5540			
cfgWlanFFreq4.0				5560			
cfgWlanFFreq5.0				5580			
cfgWlanFFreq6.0				5660			
cfgWlanFFreq7.0				5680			
cfgWlanFFreq8.0				5700			
cfgChanCleanEnabled				1			
cfgChanCleanUsableFrequencyList				0			

5.29 Interference Detection Function (IDF)

Each wireless device (AP and STA) has the functionality to analyze the operation frequency in use. Those values can be read out at any time on any device via SNMP.

In RT-370 products, the IDF functionality supports JSON reports which can be sent to a configured URL ([cfgHttpRprtServerUrl](#)) at a configured interval ([cfgIdfInterval](#)).

IDF is (currently) only available on RT-370 devices since it uses the second radio (antenna port A4, i.e. wlan1) interface.

Please note that the second radio is also used for DFS as described in section [5.28.2](#)

Basic IDF configuration:

- Set [cfgIdfEnabled.0](#) to INTEGER: enabled(1)
- Set [cfgHttpRprtServerUrl.0](#) to STRING: http://192.168.1.1:8000/json

Configure second WLAN interface (Monitor Mode):

- Set [cfgWlanIfaceMode.1](#) to INTEGER: monitor(2) (i.e. Monitor Mode)
- Set [cfgNetWlanEnabled.1](#) to INTEGER: enabled(1)

Example IDF task configuration (scan frequency 5500, wifi data collection):

- Set `cfgIdfScanWorkFreq.0` to INTEGER: 5500
- Set `cfgIdfScanWorkAction.0` to INTEGER: wifi(4)
- Set `cfgIdfScanWorkSeconds.0` to INTEGER: 1

All actions according to the IDF task list (`cfgIdfScanWorkTable`) are processed sequentially. The whole process is repeated endlessly. The time per action can be configured (`cfgIdfScanWorkSeconds`) in seconds.

For a complete list of IDF configuration elements see also [cfgIdf](#).

5.30 Http Report

The HTTP Report interface is used by several services and provides a simple way of status reporting to a standard HTTP server.

- Protocol: HTTP POST
- Content-type: application/json
- Servers URL: Can be configured by [cfgHttpRprtServerUrl](#)

5.30.1 NWM and ChannelManager Report

For more information about the current status of the NWM and the Channel Manager (see section 5.28.2) it is possible to request HTTP reports via SNMP.

Nwm 'status' report: Set `rpcNwmHttpReport.0` to 1

```
{
  "report": "NwmStatus",
  "epoch": 1436275841,
  "data": {
    "name": "Nwm",
    "nominal_freq": { "freq": 5300, "ext_freq": 5320, "htmode": 1 },
    "opfreq": { "freq": 5300, "ext_freq": 5320, "htmode": 1 }
  }
}
```

Nwm 'frequency state' report: Set [rpcNwmHttpReport.0](#) to 2

```
{
  "report": "NwmFreqState",
  "epoch": 1436277622,
  "data": {
    "name": "Nwm",
    "freq_state_list": [
      [ 5180, 1 ],
      [ 5200, 1 ],
      [ 5220, 1 ],
      [ 5240, 1 ],
      [ 5260, 1 ],
      [ 5280, 1 ],
      [ 5300, 1 ],
      [ 5320, 1 ]
    ]
  }
}
```

Channel Manager 'frequency state' report: Set [rpcChMgrHttpReport.0](#) to 1

```
{
  "report": "ChMgrFreqState",
  "epoch": 1436277622,
  "data": {
    "freq_state_list": [
      [ 5180, 1 ],
      [ 5200, 1 ],
      [ 5220, 1 ],
      [ 5240, 1 ],
      [ 5260, 1 ],
      [ 5280, 1 ],
      [ 5300, 1 ],
      [ 5320, 1 ]
    ]
  }
}
```

Channel Manager 'channels' report: Set [rpcChMgrHttpReport.0](#) to 2

```
{
  "report": "ChMgrChannels",
  "epoch": 1436277864,
  "data": {
    "chan_nom": { "freq": 5300, "ext_freq": 5320, "htmode": 1 },
    "proposed_chan": { "freq": 5300, "ext_freq": 5320, "htmode": 1 },
    "chans_ht20": {
```

```
    "all": [ 5180, 5200, 5220, 5240, 5260, 5280, 5300, 5320 ],
    "available": [ 5180, 5200, 5220, 5240, 5260, 5280, 5300, 5320 ],
    "htmode": 0
  },
  "chans_ht40m": {
    "all": [ 5200, 5240, 5280, 5320 ],
    "available": [ 5200, 5240, 5280, 5320 ],
    "htmode": 2
  },
  "chans_ht40p": {
    "all": [ 5180, 5220, 5260, 5300 ],
    "available": [ 5180, 5220, 5260, 5300 ],
    "htmode": 1
  }
}
}
```

5.30.2 IDF Report

An IDF report contains the scan results of all results available in the scan result buffer at the end of the [cfgldfInterval](#) (i.e. at the time the IDF report is sent).

Please refer to section [5.29](#) for more information on how to configure the IDF

Elements of the IDF Report are:

- data.wlan_stats will only be present if [cfgldfScanWorkAction](#) = wifi(4)
- data.wlan_stats.domest_fstats, data.wlan_stats.alien_fstats and data.wlan_stats.alien_mac might not be sent in each report
- data.radar_reports will only be present if [cfgldfScanWorkAction](#) = radar(3)
- data.spectral_reports will only be present if [cfgldfScanWorkAction](#) = spectral(2)

General elements of the IDF Report wlan_stats element are:

- epoch: UNIX time stamp
- freq: frequency in MHz
- bandwidth: bandwidth in MHz
- window_time: window time in milliseconds

- busy_time: busy time value in milliseconds
- rx_time: rx time value in milliseconds
- tx_time: tx time value in milliseconds

Elements of the IDF Report data.wlan_stats.domest_fstats and data.wlan_stats.alien_fstats element are:

- frame_cnt: frame counter
- frame_size: minimum, average and maximum frame length counter
- frame_total: bytes counter
- rssi: minimum, average and maximum frame RSSI

Elements of the IDF Report data.wlan_stats.alien_mac (list of alien MAC addresses found) element are:

- mac: MAC address
- frame_total: frame counter
- rssi_max: maximum frame RSSI
- rssi_avg: average frame RSSI

Elements of an IDF Report radar_reports element are:

- epoch: UNIX time stamp
- freq: Frequency in MHz
- radar_detected: radar counter
- seconds: observation time

Elements of an IDF Report spectral_reports element are:

- epoch: UNIX time stamp
- freq: Frequency in MHz
- num_samples: Number of FFT data processed

- seconds: observation time
- stats: minimum, average and maximum (all in dBm) of bin0, bin1, bin3, ..., bin55

Example of an IDF Report:

```
{
  "report": "IDF",
  "epoch": 1418301089,
  "data": {
    "name": "IDF",
    "wlan_stats" : [
      {
        "epoch" : 1315522477,
        "freq" : 5500,
        "bandwidth" : 20,
        "window_time" : 60,
        "busy_time" : 15,
        "rx_time" : 12,
        "tx_time" : 0,
        "domest_fstats" : {
          "frame_cnt" : 100,
          "frame_size" : [100, 200, 300],
          "frame_total" : 20153,
          "rssi" : [ 35, 45, 50 ]
        },
        "alien_fstats" : {
          "frame_cnt" : 10,
          "frame_size" : [ 50, 300, 500 ],
          "frame_total" : 4598,
          "rssi" : [ 15, 35, 50 ]
        },
        "alien_mac" : [
          {
            "mac" : "00:00:00:00:00:01",
            "frame_total" : 2856,
            "rssi_max" : 50,
            "rssi_avg" : 50
          }
        ]
      }
    ],
    "radar_reports": [
      {
        "epoch": 1315528096,
        "freq": 5700,
        "radar_detected": 2,

```

```
    "seconds": 60
  }
],
"spectral_reports": [
  {
    "epoch": 1315522405,
    "freq": 5500,
    "num_samples": 14438,
    "seconds": 60,
    "stats": [
      [-159, -107, -91],
      [-154, -106, -92],
      [-157, -106, -91],
      ..
      [-157, -107, -92]
    ]
  }
]
}
```

5.31 Firewall

5.31.1 Network Address Translation (NAT)

The *Software 6* uses the well known *netfilter* to filter or mangle network traffic. The configuration is done by the *iptables* application. Most terms are based on these software components.

To use the firewall it has to be enabled. Otherwise no feature described below will work. To enable the feature the `cfgFwEnabled` flag has to be enabled.

In the following sections you can define multiple rules. Please keep in mind that the order of the rules is important! This means the rule with the index 2 will be processed before the rule with the index 3.

5.31.1.1 Port forward

Port forwarding can be used to forward network traffic to another destination. This is also known as *Destination Network Address Translation (DNAT)*.

To illustrate how to configure the port forward, we setup a port forward to a web server and a database server which are common use cases. The goal is to connect from Radio Modem 1 (RM1), through

the wireless link to RM2, to the web server or the database server.

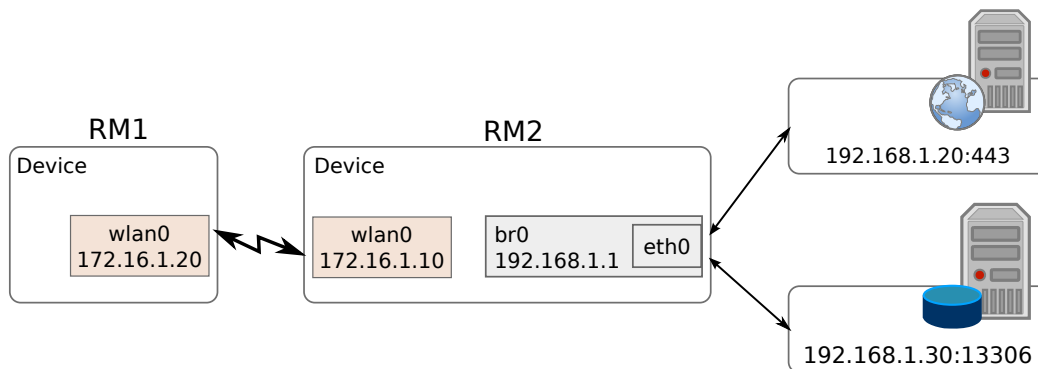


Figure 5.25: Port forward to servers behind an device

To forward the network traffic to the web server and the database server we add two new *rules* to the port forward rules table:

1. For the web server we only want to forward tcp traffic to port 443.
2. For the database server we want to forward tcp and udp traffic for the port range 2000 - 2100 to illustrate port ranges. In addition we want to accept traffic to the wlan from anywhere.

Configuration File Example: Port forward

```
# Config.Format = raw
WESTERMO-SW6-FIREWALL-MIB::cfgFwEnabled.0 = 1
# Port forward to the web server
WESTERMO-SW6-FIREWALL-MIB::cfgFwNatPrtFwdEnabled.0 = 1
WESTERMO-SW6-FIREWALL-MIB::cfgFwNatPrtFwdInterface.0 = wlan0
WESTERMO-SW6-FIREWALL-MIB::cfgFwNatPrtFwdProtocol.0 = 2
WESTERMO-SW6-FIREWALL-MIB::cfgFwNatPrtFwdDestinationAddress.0 = 172.16.1.0/24
WESTERMO-SW6-FIREWALL-MIB::cfgFwNatPrtFwdDestinationPortStart.0 = 443
WESTERMO-SW6-FIREWALL-MIB::cfgFwNatPrtFwdDestinationPortEnd.0 = -1
WESTERMO-SW6-FIREWALL-MIB::cfgFwNatPrtFwdRedirectDestinationAddress.0 = 192.168.1.20
WESTERMO-SW6-FIREWALL-MIB::cfgFwNatPrtFwdRedirectDestinationPort.0 = 443
# Port forward to the database server
WESTERMO-SW6-FIREWALL-MIB::cfgFwNatPrtFwdEnabled.1 = 1
WESTERMO-SW6-FIREWALL-MIB::cfgFwNatPrtFwdInterface.1 = wlan0
WESTERMO-SW6-FIREWALL-MIB::cfgFwNatPrtFwdProtocol.1 = 3
WESTERMO-SW6-FIREWALL-MIB::cfgFwNatPrtFwdDestinationAddress.1 = 0.0.0.0/0
WESTERMO-SW6-FIREWALL-MIB::cfgFwNatPrtFwdDestinationPortStart.1 = 2000
WESTERMO-SW6-FIREWALL-MIB::cfgFwNatPrtFwdDestinationPortEnd.1 = 2100
WESTERMO-SW6-FIREWALL-MIB::cfgFwNatPrtFwdRedirectDestinationAddress.1 = 192.168.1.20
WESTERMO-SW6-FIREWALL-MIB::cfgFwNatPrtFwdRedirectDestinationPort.1 = 13306
```

5.31.1.2 Outbound NAT

With the outbound NAT the device can control how traffic leaving the device will be translated. It's also known as Source NAT (SNAT) and used in the most home routers to rewrite the source address to the address of the WAN interface of the router so the traffic finds the way back home.

The SNAT can be done by simple masquerade, means take the address of the network interface or by defining the source address/port.

As for the port forward we use a simple example to illustrate the functionality as shown in Figure 5.26. The goal is to connect from a Laptop, through RM1 to the web interface of RM2. For this example the Laptop use RM1 as default gateway and the wlan0 interface of RM1 has a dynamic address.

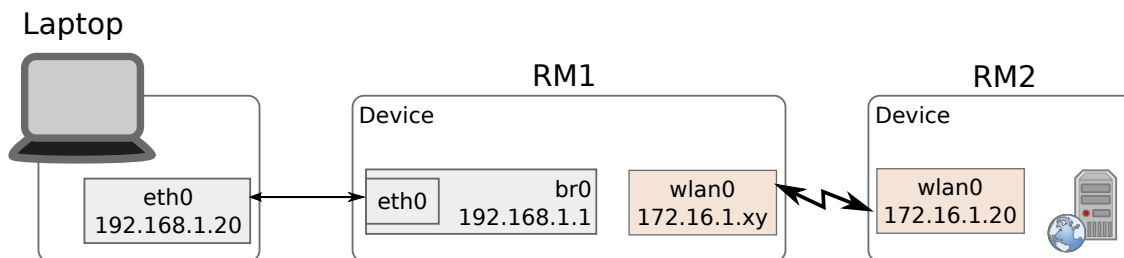


Figure 5.26: Example for outbound NAT on a device

At this point we will only describe the steps to configure the outbound NAT. Following example enables the first rule, set the output interface to *wlan0* and applies to TCP traffic only.

Configuration File Example: Outbound NAT

```
# Config.Format = raw
WESTERMO-SW6-FIREWALL-MIB::cfgFwEnabled.0 = 1
WESTERMO-SW6-FIREWALL-MIB::cfgFwNatOutEnabled.0 = 1
WESTERMO-SW6-FIREWALL-MIB::cfgFwNatOutInterface.0 = wlan0
WESTERMO-SW6-FIREWALL-MIB::cfgFwNatPrtFwdProtocol.0 = 2
```

With this configuration you should be able to connect from the laptop to 172.16.1.20, to the web interface of RM2.

For all possible configurations please see [firewall](#) in the MIB reference.

5.31.2 Filter

The iptables filter allows to create firewall rules based on interfaces, protocol, IP-addresses and ports.

There are 3 chains on which rules can be created:

- **Input**
Rules on the input chain allow/block frames which are destined to the device itself. With these rules you can e.g. block or limit access to the Web Interface or SNMP.
- **Forward**
Rules on the forward chain allow/block frames which are routed/forwarded by the device. This is useful e.g. when multiple VLANs are terminated on the device but traffic between them should not be possible. Be aware that these rules can not be applied when the frames are forwarded on L2 (switched). Use the L2 IP Filter Firewall described below for this.
- **Output**
Rules on the output chain allow/block frames which originate from the device.

The options [cfgFwFltDefaultPolicyInput](#), [cfgFwFltDefaultPolicyForward](#) and [cfgFwFltDefaultPolicyOutput](#) define what happens with frames when no rule matches.

When there are multiple conflicting rules which have overlapping match criteria, then the rule with the lowest index will be executed.

Configuration File Example:

Input filtering, Allow SNMP from 192.168.1.0/24 and block everywhere else

```
# Config.Format = raw
WESTERMO-SW6-FIREWALL-MIB::cfgFwEnabled.0 = 1

# Set default input policy to drop
WESTERMO-SW6-FIREWALL-MIB::cfgFwFltDefaultPolicyInput.0 = 0

# Allow traffic to SNMP from the subnet 192.168.1.0/24 to 192.168.1.20
WESTERMO-SW6-FIREWALL-MIB::cfgFwFltREnabled.0 = 1
WESTERMO-SW6-FIREWALL-MIB::cfgFwFltRChain.0 = 1
WESTERMO-SW6-FIREWALL-MIB::cfgFwFltRAction.0 = 1
WESTERMO-SW6-FIREWALL-MIB::cfgFwFltRInputInterface.0 = -1
WESTERMO-SW6-FIREWALL-MIB::cfgFwFltROutputInterface.0 = -1
WESTERMO-SW6-FIREWALL-MIB::cfgFwFltRProtocol.0 = 17
WESTERMO-SW6-FIREWALL-MIB::cfgFwFltRSourceAddress.0 = 192.168.1.0/24
WESTERMO-SW6-FIREWALL-MIB::cfgFwFltRSourcePortStart.0 = -1
WESTERMO-SW6-FIREWALL-MIB::cfgFwFltRSourcePortEnd.0 = -1
WESTERMO-SW6-FIREWALL-MIB::cfgFwFltRDestinationAddress.0 = 192.168.1.20/32
WESTERMO-SW6-FIREWALL-MIB::cfgFwFltRDestinationPortStart.0 = 161
WESTERMO-SW6-FIREWALL-MIB::cfgFwFltRDestinationPortEnd.0 = -1
```

Configuration File Example:

Forward filtering, Allow HTTP from everywhere and block everything else

```
# Config.Format = raw
WESTERMO-SW6-FIREWALL-MIB::cfgFwEnabled.0 = 1

# Set default forward policy to drop
WESTERMO-SW6-FIREWALL-MIB::cfgFwFltDefaultPolicyForward.0 = 0

# Allow HTTP traffic from everywhere to everywhere
WESTERMO-SW6-FIREWALL-MIB::cfgFwFltREnabled.0 = 1
WESTERMO-SW6-FIREWALL-MIB::cfgFwFltRChain.0 = 2
WESTERMO-SW6-FIREWALL-MIB::cfgFwFltRAction.0 = 1
WESTERMO-SW6-FIREWALL-MIB::cfgFwFltRInputInterface.0 = -1
WESTERMO-SW6-FIREWALL-MIB::cfgFwFltROutputInterface.0 = -1
WESTERMO-SW6-FIREWALL-MIB::cfgFwFltRProtocol.0 = 6
WESTERMO-SW6-FIREWALL-MIB::cfgFwFltRSourceAddress.0 = -1
WESTERMO-SW6-FIREWALL-MIB::cfgFwFltRSourcePortStart.0 = -1
WESTERMO-SW6-FIREWALL-MIB::cfgFwFltRSourcePortEnd.0 = -1
WESTERMO-SW6-FIREWALL-MIB::cfgFwFltRDestinationAddress.0 = -1
WESTERMO-SW6-FIREWALL-MIB::cfgFwFltRDestinationPortStart.0 = 80
WESTERMO-SW6-FIREWALL-MIB::cfgFwFltRDestinationPortEnd.0 = -1
```

Configuration File Example:

Output filtering, Block access to the tftp server at 192.168.1.3

```
# Config.Format = raw
WESTERMO-SW6-FIREWALL-MIB::cfgFwEnabled.0 = 1

# Set default output policy to accept
WESTERMO-SW6-FIREWALL-MIB::cfgFwFltDefaultPolicyOutput.0 = 1

# Block TFTP traffic to server at 192.168.1.3/32
WESTERMO-SW6-FIREWALL-MIB::cfgFwFltREnabled.0 = 1
WESTERMO-SW6-FIREWALL-MIB::cfgFwFltRChain.0 = 3
WESTERMO-SW6-FIREWALL-MIB::cfgFwFltRAction.0 = 0
WESTERMO-SW6-FIREWALL-MIB::cfgFwFltRInputInterface.0 = -1
WESTERMO-SW6-FIREWALL-MIB::cfgFwFltROutputInterface.0 = -1
WESTERMO-SW6-FIREWALL-MIB::cfgFwFltRProtocol.0 = 17
WESTERMO-SW6-FIREWALL-MIB::cfgFwFltRSourceAddress.0 = -1
WESTERMO-SW6-FIREWALL-MIB::cfgFwFltRSourcePortStart.0 = -1
WESTERMO-SW6-FIREWALL-MIB::cfgFwFltRSourcePortEnd.0 = -1
WESTERMO-SW6-FIREWALL-MIB::cfgFwFltRDestinationAddress.0 = 192.168.1.3/32
WESTERMO-SW6-FIREWALL-MIB::cfgFwFltRDestinationPortStart.0 = 69
```

```
WESTERMO-SW6-FIREWALL-MIB::cfgFwFltRDestinationPortEnd.0 = -1
```

5.31.3 L2 IP Filter Firewall

The L2 IP Filter can be used to filter IP frames on bridges. This filter will only apply on IP frames and will not touch anything else.

To make use of the L2 IP Filter, it must be enabled by [cfgFwL2IpFilterEnabled](#) and the global default action must be defined by [cfgFwL2IpFilterDefaultAction](#). Since the global default action applies to all bridges, take care to not lock yourself out when the default action is 'drop'.

Up to 64 filter rules can be configured in the [cfgFwL2IpFilterTable](#). For each filter rule the source and destination network/IP on which the rule matches, the bridge on which the rule is installed, the action (accept or drop) to perform and the priority of the rule can be configured. Please refer to [cfgFwL2IpFilterTable](#) for more information.

Following example of a L2 IP Filter rule drops IP frames with any source IP and 192.168.3.130 as destination IP on bridge 0. All other frames are processed normally.

Configuration File Example: L2 IP Filter

```
# Config.Format = raw
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFilterEnabled.0 = 1
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFilterDefaultAction.0 = 0

WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrEnabled.0 = 1
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrBridge.0 = 0
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrSource.0 = 0.0.0.0/0
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrDestination.0 = 192.168.3.130/32
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrAction.0 = 1
```

6 Country Codes

Product regulatory limits and operating parameters are controlled by product software driver with the country code settings. The country code limits are equally valid for Client (STA) and Access Point operation mode. Not all country codes are supported by all product variants and versions.

6.1 Configuration

The configuration values as follows are relevant for the system to compute the allowed output power at antenna port and radiated:

- set [cfgWlanDevAntennaGain](#) Set the antenna gain in dBi. The selected country code describes the range. In some regions the antenna type is fixed. The user cannot change the antenna gain.
- set [cfgWlanDevPower](#) to limit the output power to a defined EIRP level. The modem software will calculate a value which is at this maximum level or lower
- set [cfgWlanDevFrequency](#) to set the nominal operation frequency

6.2 Regions for 802.11n products

6.2.1 Country code WORLD

Country code 'WORLD' is the default country code. The country code shall be changed to the correct country by the user before installation.

	Min	Max
Frequencies 2.4 GHz	2412, 2417, 2422, 2427, 2432, 2437, 2442, 2447, 2452, 2457, 2462	
- Antenna Gain	0 dBi	12 dBi
- EIRP (one antenna)	6 dBm	18 dBm
- EIRP (two antennas)	9 dBm	18 dBm
- EIRP (three antennas)	11 dBm	18 dBm

6.2.2 Region E

- Max. EIRP Power are lower than regulatory power limits to respect the maximum limit at all conditions.
- Client (STA) and Access Point maximal EIRP differences at 5 GHz are depends on operation mode. Access Point mode is DFS master, client (STA) is DFS slave.
- DFS weather channels are disabled due to ETSI requirements.
- For DFS frequencies a modem in client (STA) mode will scan passive (no probe requests)

6.2.2.1 Country code EU

Country code 'EU' applies for Europe.

The country code can be set by:

- Set `cfgWlanGlblCountry` to EU

Note 1: EIRP of 27 dBm applies for 5500 to 5700 in Access Point mode (DFS master).

	Min	Max
Frequencies 2.4 GHz	2412, 2417, 2422, 2427, 2432, 2437, 2442, 2447, 2452, 2457, 2462, 2467, 2472	
- Antenna Gain	0 dBi	12 dBi
- EIRP (one antenna)	6 dBm	18 dBm
- EIRP (two antennas)	9 dBm	18 dBm
- EIRP (three antennas)	11 dBm	18 dBm
Frequencies 5 GHz	5180, 5200, 5220, 5240, 5260, 5280, 5300, 5320, 5500, 5520, 5540, 5560, 5580, 5660, 5680, 5700	
- Antenna Gain	0 dBi	15 dBi
- EIRP (one antenna)	6 dBm	21 dBm, 27 dBm (Note 1)
- EIRP (two antennas)	9 dBm	21 dBm, 27 dBm (Note 1)
- EIRP (three antennas)	11 dBm	21 dBm, 27 dBm (Note 1)
- Indoor frequencies	5180 to 5320	
- DFS (slave/master)	5260 to 5700	
Frequencies 5.8 GHz	5745, 5765, 5785, 5805, 5825, 5845, 5865	
- Antenna Gain	0 dBi	7 dBi
- EIRP (one antenna)	6 dBm	13 dBm
- EIRP (two antennas)	9 dBm	13 dBm
- EIRP (three antennas)	11 dBm	13 dBm

6.2.2.2 Country code CHINA

Country code 'CHINA' applies for China using low gain antennas in the 2.4 GHz band.

The country code can be set by:

- Set `cfgWlanGlblCountry` to CHINA

	Min	Max
Frequencies 2.4 GHz	2412, 2417, 2422, 2427, 2432, 2437, 2442, 2447, 2452, 2457, 2462, 2467, 2472	
- Antenna Gain	0 dBi	9 dBi
- EIRP (one antenna)	6 dBm	18 dBm
- EIRP (two antennas)	9 dBm	18 dBm
- EIRP (three antennas)	11 dBm	18 dBm
Frequencies 5.8 GHz	5745, 5765, 5785, 5805, 5825	
- Antenna Gain	0 dBi	14 dBi
- EIRP (one antenna)	6 dBm	31 dBm
- EIRP (two antennas)	9 dBm	31 dBm
- EIRP (three antennas)	11 dBm	31 dBm

6.2.2.3 Country code CHINA_HIGH_GAIN

Country code 'CHINA_HIGH_GAIN' applies for China using high gain antennas in the 2.4 GHz band.

The country code can be set by:

- Set `cfgWlanGlbCountry` to CHINA_HIGH_GAIN

	Min	Max
Frequencies 2.4 GHz	2412, 2417, 2422, 2427, 2432, 2437, 2442, 2447, 2452, 2457, 2462, 2467, 2472	
- Antenna Gain	10 dBi	14 dBi
- EIRP (one antenna)	6 dBm	25 dBm
- EIRP (two antennas)	9 dBm	25 dBm
- EIRP (three antennas)	11 dBm	25 dBm
Frequencies 5.8 GHz	5745, 5765, 5785, 5805, 5825	
- Antenna Gain	0 dBi	14 dBi
- EIRP (one antenna)	6 dBm	31 dBm
- EIRP (two antennas)	9 dBm	31 dBm
- EIRP (three antennas)	11 dBm	31 dBm

6.2.2.4 Country code AUS_NZL

Country code 'AUS_NZL' applies for Australia and New Zealand.

The country code can be set by:

- Set `cfgWlanGlblCountry` to AUS_NZL

	Min	Max
Frequencies 2.4 GHz	2412, 2417, 2422, 2427, 2432, 2437, 2442, 2447, 2452, 2457, 2462, 2467, 2472	
- Antenna Gain	0 dBi	12 dBi
- EIRP (one antenna)	6 dBm	34 dBm
- EIRP (two antennas)	9 dBm	34 dBm
- EIRP (three antennas)	11 dBm	34 dBm
Frequencies 5 GHz	5180, 5200, 5220, 5240, 5260, 5280, 5300, 5320, 5500, 5520, 5540, 5560, 5580, 5660, 5680, 5700	
- Antenna Gain	0 dBi	15 dBi
- EIRP (one antenna)	6 dBm	27 dBm
- EIRP (two antennas)	9 dBm	27 dBm
- EIRP (three antennas)	11 dBm	27 dBm
- Indoor frequencies	5180 to 5320	
- DFS	5260 to 5700	
Frequencies 5.8 GHz	5745, 5765, 5785, 5805, 5825	
- Antenna Gain	0 dBi	7 dBi
- EIRP (one antenna)	6 dBm	34 dBm
- EIRP (two antennas)	9 dBm	34 dBm
- EIRP (three antennas)	11 dBm	34 dBm

6.2.3 Region U

- For this region the antenna type is fixed. The user cannot change the antenna gain.
- Max. EIRP Power are lower than regulatory power limits to respect the maximum limit at all conditions.
- DFS weather channels are disabled due to FCC requirements.
- For DFS frequencies a modem in client (STA) mode will scan passive (no probe requests)

6.2.3.1 Country codes USA and CANADA

Country code 'USA' applies for the United States. Country code 'CANADA' applies for the Canada.

The country code can be set by:

- Set `cfgWlanGlblCountry` to USA or CANADA

	Min	Max
Frequencies 2.4 GHz	2412, 2417, 2422, 2427, 2432, 2437, 2442, 2447, 2452, 2457, 2462	
- Antenna Types	Refer to Antennas	
- EIRP	Refer to Antennas	
Frequencies 5 GHz	5180, 5200, 5220, 5240, 5260, 5280, 5300, 5320, 5500, 5520, 5540, 5560, 5580, 5660, 5680, 5700	
- Antenna Types	Refer to Antennas	
- EIRP	Refer to Antennas	
- Indoor frequencies	5180 to 5240	
- DFS (slave/master)	5260 to 5720	
Frequencies 5.8 GHz	5745, 5765, 5785, 5805, 5825	
- Antenna Types	Refer to Antennas	
- EIRP	Refer to Antennas	

6.2.3.2 Antennas

Only the PROFESSIONAL user can configure the attached antenna for each product by configuring one of the antennas listed in the Table 6.1. The current antenna configuration of the product is shown on the Web Interface login page. More information about EIRP can be found in the Web Interface under *Application > Regulatory Domain Manager*.

Antenna	Description
H&S SPA 2400/75/8/0/V	1 to 3 active antenna ports allowed Gain 2.4GHz = 8 dBi, Gain 5GHz = na Max. EIRP (2412 to 2462) = 22 dBm Examples: - H&S SPA 2400/75/8/0/V
Tekfun F51-N	1 to 3 active antenna ports allowed

Gain 2.4GHz = 5 dBi, Gain 5GHz = 7 dBi
Max. EIRP (2412 to 2462) = 24 dBm
Max. EIRP (5180 to 5240) = 19 dBm (USA)
Max. EIRP (5180 to 5240) = 21 dBm (CANADA)
Max. EIRP (5260 to 5320) = 28 dBm (USA)
Max. EIRP (5260 to 5320) = 21 dBm (CANADA)
Max. EIRP (5500 to 5700) = 28 dBm
Max. EIRP (5745 to 5825) = 28 dBm

Examples:

- Tekfun F51-N
- Westermo ICL 5GHz MIMO Antenna

H&S SPA 5600/40/14/0/V_2

1 or 2 active antenna ports allowed
Gain 2.4GHz = na, Gain 5GHz = 14 dBi
Max. EIRP (5180 to 5240) = 19 dBm (USA)
Max. EIRP (5180 to 5240) = 21 dBm (CANADA)
Max. EIRP (5260 to 5320) = 28 dBm (USA)
Max. EIRP (5260 to 5320) = 21 dBm (CANADA)
Max. EIRP (5500 to 5700) = 28 dBm
Max. EIRP (5745 to 5825) = 30 dBm

Examples:

- H&S SPA 5600/40/14/0/V_2

H&S SPA 5600/65/9/0/MIMO

1 to 3 active antenna ports allowed
Gain 2.4GHz = na, Gain 5GHz = 9 dBi
Max. EIRP (5180 to 5240) = 19 dBm (USA)
Max. EIRP (5180 to 5240) = 21 dBm (CANADA)
Max. EIRP (5260 to 5320) = 28 dBm (USA)
Max. EIRP (5260 to 5320) = 21 dBm (CANADA)
Max. EIRP (5500 to 5700) = 28 dBm
Max. EIRP (5745 to 5825) = 30 dBm

Examples:

- H&S SPA 5600/65/9/0/MIMO

H&S SPA 5600/45/12/10/V

1 or 2 active antenna ports allowed
Gain 2.4GHz = na, Gain 5GHz = 12 dBi
Max. EIRP (5180 to 5240) = 19 dBm (USA)
Max. EIRP (5180 to 5240) = 21 dBm (CANADA)
Max. EIRP (5260 to 5320) = 28 dBm (USA)
Max. EIRP (5260 to 5320) = 21 dBm (CANADA)
Max. EIRP (5500 to 5700) = 28 dBm
Max. EIRP (5745 to 5825) = 28 dBm

Examples:

H&S SPA 5600/40/14/0/V_2 with Neratec 105072

- H&S SPA 5600/45/12/10/V
- 1 or 2 active antenna ports allowed
- Gain 2.4GHz = na, Gain 5GHz = 13 dBi
- Max. EIRP (5745-5825) = 34 dBm

Examples:

- H&S SPA 5600/40/14/0/V_2 with Neratec 105072

Table 6.1: *Region U antennas of radio0*

6.3 Regions for 802.11ac products

6.3.1 Region E

- Max. EIRP Power are lower than regulatory power limits to respect the maximum limit at all conditions.
- Client (STA) and Access Point maximal EIRP differences at 5 GHz are depends on operation mode. Access Point mode is DFS master, client (STA) is DFS slave.
- For DFS frequencies a modem in client (STA) mode will scan passive (no probe requests)

6.3.1.1 Europe

Following country codes (<country>) are supported for Europe: 'AU', 'BE', 'BG', 'HR', 'CY', 'CZ', 'DK', 'EE', 'FI', 'FR', 'DE', 'GR', 'HU', 'IE', 'IT', 'LV', 'LT', 'LU', 'MT', 'NL', 'NO', 'PL', 'PT', 'RO', 'SK', 'SI', 'ES', 'SE', 'GB', 'IS', 'LI', 'CH', 'TR'

The country code can be set by:

- Set `cfgWlanGlblCountry` to <country>

	Min	Max
Frequencies 2.4 GHz	2412, 2417, 2422, 2427, 2432, 2437, 2442, 2447, 2452, 2457, 2462, 2467, 2472	
- Antenna Gain	0 dBi	12 dBi
- EIRP (one antenna)	6 dBm	20 dBm
- EIRP (two antennas)	9 dBm	20 dBm
- EIRP (three antennas)	11 dBm	20 dBm
- EIRP (four antennas)	12 dBm	20 dBm
Frequencies 5 GHz	5180, 5200, 5220, 5240, 5260, 5280, 5300, 5320, 5500, 5520, 5540, 5560, 5580, 5600, 5620, 5640, 5660, 5680, 5700	
- Antenna Gain	0 dBi	15 dBi
- EIRP (one antenna)	6 dBm	27 dBm
- EIRP (two antennas)	9 dBm	27 dBm
- EIRP (three antennas)	11 dBm	27 dBm
- EIRP (four antennas)	12 dBm	27 dBm
- Indoor frequencies	5180 to 5320	
- DFS (slave/master)	5260 to 5700	

6.3.2 Region U

- For this region the antenna type is fixed. The user cannot change the antenna gain.
- Max. EIRP Power are lower than regulatory power limits to respect the maximum limit at all conditions.
- DFS weather channels are disabled due to FCC requirements.
- For DFS frequencies a modem in client (STA) mode will scan passive (no probe requests)

6.3.2.1 North America

Following country codes (<country>) are supported for North America: 'US', 'CA'

The country code can be set by:

- Set `cfgWlanGlblCountry` to <country>

	Min	Max
Frequencies 2.4 GHz	2412, 2417, 2422, 2427, 2432, 2437, 2442, 2447, 2452, 2457, 2462	
- Antenna Types	Refer to Antennas	
- EIRP	Refer to Antennas	
Frequencies 5 GHz	5180, 5200, 5220, 5240, 5260, 5280, 5300, 5320, 5500, 5520, 5540, 5560, 5580, 5600, 5620, 5640, 5660, 5680, 5700, 5720	
- Antenna Types	Refer to Antennas	
- EIRP	Refer to Antennas	
- Indoor frequencies	5180 to 5240	
- DFS (slave/master)	5260 to 5720	
Frequencies 5.8 GHz	5745, 5765, 5785, 5805, 5825	
- Antenna Types	Refer to Antennas	
- EIRP	Refer to Antennas	

6.3.2.2 Antennas

Only the PROFESSIONAL user can configure the attached antenna for each the product by configuring one of the antennas listed in the Table 6.2 and 6.3, respectively. The current antenna configuration of the product is shown on the Web Interface login page. Please contact your support for more information.

Antenna	Description
5GHz 5dBi	1 to 4 active antenna ports allowed Gain 2.4GHz = na, Gain 5GHz = 5 dBi Max. EIRP (5180 to 5240) = 23 dBm (USA) Max. EIRP (5180 to 5240) = 21 dBm (CANADA) Max. EIRP (5260 to 5320) = 22 dBm (USA) Max. EIRP (5260 to 5320) = 21 dBm (CANADA) Max. EIRP (5500 to 5700) = 21 dBm Max. EIRP (5745 to 5825) = 29 dBm Examples: - H&S SENCITY®Omni-S MIMO Antenna
5GHz 4dBi	1 to 4 active antenna ports allowed Gain 2.4GHz = na, Gain 5GHz = 4 dBi Max. EIRP (5180 to 5240) = 22 dBm (USA) Max. EIRP (5180 to 5240) = 21 dBm (CANADA) Max. EIRP (5260 to 5320) = 21 dBm

Max. EIRP (5500 to 5700) = 20 dBm
 Max. EIRP (5745 to 5825) = 28 dBm

5GHz 10dBi 1 to 4 active antenna ports allowed
 Gain 2.4GHz = na, Gain 5GHz = 10 dBi
 Max. EIRP (5180 to 5240) = 23 dBm (USA)
 Max. EIRP (5180 to 5240) = 21 dBm (CANADA)
 Max. EIRP (5260 to 5320) = 27 dBm (USA)
 Max. EIRP (5260 to 5320) = 21 dBm (CANADA)
 Max. EIRP (5500 to 5700) = 26 dBm
 Max. EIRP (5745 to 5825) = 34 dBm

Examples:

- H&S SPA 5600/65/9/0/MIMO
- Antonics OmPlecs®-TOP 200 AMR MF-06 -1-
- Antonics OmPlecs®-TOP 200 AMR MF-06 -4-
- Mars MA-WO6960-DP6
- Mars MA-WO3860-MIMO

Table 6.2: Region U antennas of radio0

Antenna	Description
2.4GHz 11dBi	1 or 2 active antenna ports allowed Gain 2.4GHz = 11 dBi, Gain 5GHz = na Max. EIRP (2412 to 2462) = 29 dBm Examples: - H&S SPA 2400/75/8/0/V
5GHz 10dBi	1 or 2 active antenna ports allowed Gain 2.4GHz = na, Gain 5GHz = 10 dBi Max. EIRP (5180 to 5240) = 30 dBm (USA) Max. EIRP (5180 to 5240) = 21 dBm (CANADA) Max. EIRP (5260 to 5320) = 28 dBm (USA) Max. EIRP (5260 to 5320) = 21 dBm (CANADA) Max. EIRP (5500 to 5700) = 28 dBm Max. EIRP (5745 to 5825) = 31 dBm Examples: - H&S SPA 5600/65/9/0/MIMO
2.4GHz, 5GHz 10dBi	1 or 2 active antenna ports allowed Gain 2.4GHz = 10 dBi, Gain 5GHz = 10 dBi Max. EIRP (2412 to 2462) = 28 dBm Max. EIRP (5180 to 5240) = 30 dBm (USA) Max. EIRP (5180 to 5240) = 21 dBm (CANADA) Max. EIRP (5260 to 5320) = 28 dBm (USA)

Max. EIRP (5260 to 5320) = 21 dBm (CANADA)
 Max. EIRP (5500 to 5700) = 28 dBm
 Max. EIRP (5745 to 5825) = 31 dBm

Examples:

- Antonics OmPlecs®-TOP 200 AMR MF-06 -1-
- Antonics OmPlecs®-TOP 200 AMR MF-06 -2-
- Mars MA-WO6960-DP6
- Mars MA-WO3860-MIMO

2.4GHz, 5GHz 7dBi

1 or 2 active antenna ports allowed
 Gain 2.4GHz = 7 dBi, Gain 5GHz = 7 dBi
 Max. EIRP (2412 to 2462) = 25 dBm
 Max. EIRP (5180 to 5240) = 27 dBm (USA)
 Max. EIRP (5180 to 5240) = 21 dBm (CANADA)
 Max. EIRP (5260 to 5320) = 30 dBm (USA)
 Max. EIRP (5260 to 5320) = 21 dBm (CANADA)
 Max. EIRP (5500 to 5700) = 28 dBm
 Max. EIRP (5745 to 5825) = 28 dBm

Examples:

- Tekfun F51-N

2.4GHz, 5GHz 5dBi

1 or 2 active antenna ports allowed
 Gain 2.4GHz = 5 dBi, Gain 5GHz = 5 dBi
 Max. EIRP (2412 to 2462) = 23 dBm
 Max. EIRP (5180 to 5240) = 25 dBm (USA)
 Max. EIRP (5180 to 5240) = 21 dBm (CANADA)
 Max. EIRP (5260 to 5320) = 28 dBm (USA)
 Max. EIRP (5260 to 5320) = 21 dBm (CANADA)
 Max. EIRP (5500 to 5700) = 28 dBm
 Max. EIRP (5745 to 5825) = 26 dBm

Examples:

- H&S SENCITY®Omni-S MIMO Antenna

2.4GHz, 5GHz 2dBi

1 or 2 active antenna ports allowed
 Gain 2.4GHz = 2 dBi, Gain 5GHz = 2 dBi
 Max. EIRP (2412 to 2462) = 20 dBm
 Max. EIRP (5180 to 5240) = 22 dBm (USA)
 Max. EIRP (5180 to 5240) = 21 dBm (CANADA)
 Max. EIRP (5260 to 5320) = 25 dBm (USA)
 Max. EIRP (5260 to 5320) = 21 dBm (CANADA)
 Max. EIRP (5500 to 5700) = 25 dBm
 Max. EIRP (5745 to 5825) = 23 dBm

Table 6.3: Region U antennas of radio1

7 Security Considerations

During commissioning it is often desirable to not have the devices locked down. This helps to debug and analyze issues. However once the commissioning phase is complete the devices should be locked down and access restricted. The following chapter provides some points what to look for and which config parameters to change to reduce the chances of an incident.

7.1 Physical Interfaces

Disable ports which are not in use. This will reduce that chance that an unauthorized party gains unnoticed access to the backbone.

Configuration File Example: Disable the second port (eth1)

```
# Config.Format = raw
WESTERMO-SW6-MIB::cfgNetEthEnabled.1 = 0
```

7.2 Network Concept

7.2.1 Local Administrative Access

Contrary to disabling unused ports, it may be desirable to provide local administrative access to the device. Use cases are to allow access to the device when the backbone is lost, or someone is working physically on site of the device. In such a case keep the normally unused port enabled but on its own bridge with its own IP. This will allow local access to the administrative interface of the device, but no further access to the backbone.

Configuration File Example: Enable local administrative access on second port (eth1)

```
# Config.Format = raw
# eth0
WESTERMO-SW6-MIB::cfgNetEthEnabled.0 = 1
WESTERMO-SW6-MIB::cfgNetEthBridge.0 = 0
# eth1
```



```
WESTERMO-SW6-MIB::cfgNetEthEnabled.1 = 1
WESTERMO-SW6-MIB::cfgNetEthBridge.1 = 1
WESTERMO-SW6-MIB::cfgNetEthVlanMode.1 = 1
WESTERMO-SW6-MIB::cfgNetEthTag.1 = 0
# br1.vlan0
WESTERMO-SW6-MIB::cfgNetVlanEnabled.1 = 1
WESTERMO-SW6-MIB::cfgNetVlanBridge.1 = 1
WESTERMO-SW6-MIB::cfgNetVlanVid.1 = 0
# IP on br1.vlan0
WESTERMO-SW6-MIB::cfgNetIpEnabled.2 = 1
WESTERMO-SW6-MIB::cfgNetIpAddr.2 = 192.168.1.20/24
WESTERMO-SW6-MIB::cfgNetIpProto.2 = 0
WESTERMO-SW6-MIB::cfgNetIpInterface.2 = br1.vlan0
```

7.2.2 Remote Administrative Access

To access the administrative interface of the device from an NMS or manually it may be desirable to have an IP address on the interface towards the backbone. However this administrative access should not be provided in a way that users of the AP have access to it. A good practice is to separate user-data and administrative data via VLANs.

In the following example we configure:

- disable eth1
- eth0 in br0 as trunk-port for vlans 9 and 13
- wlan0 in br0 as access-port for vlan 9
- br0.vlan13 as access-port for vlan 13 with a local IP

Configuration File Example: Allow remote administrative access on separate VLAN

```
# Config.Format = raw
# disable second port eth1
WESTERMO-SW6-MIB::cfgNetEthEnabled.1 = 0
# eth0 in br0 as trunk with vlans 9 and 13
WESTERMO-SW6-MIB::cfgNetEthEnabled.0 = 1
WESTERMO-SW6-MIB::cfgNetEthBridge.0 = 0
WESTERMO-SW6-MIB::cfgNetEthVlanMode.0 = 0
WESTERMO-SW6-MIB::cfgNetEthTrunk.0 = 9, 13
# wlan0 in br0 as access port for vlan 9
WESTERMO-SW6-MIB::cfgNetEthEnabled.0 = 1
WESTERMO-SW6-MIB::cfgNetEthBridge.0 = 0
WESTERMO-SW6-MIB::cfgNetEthVlanMode.0 = 1
WESTERMO-SW6-MIB::cfgNetEthTag.0 = 9
```

```
# br0.vlan13
WESTERMO-SW6-MIB::cfgNetVlanEnabled.1 = 1
WESTERMO-SW6-MIB::cfgNetVlanBridge.1 = 0
WESTERMO-SW6-MIB::cfgNetVlanVid.1 = 13
# IP on br0.vlan13
WESTERMO-SW6-MIB::cfgNetIpEnabled.2 = 1
WESTERMO-SW6-MIB::cfgNetIpAddr.2 = 10.11.12.13/24
WESTERMO-SW6-MIB::cfgNetIpProto.2 = 0
WESTERMO-SW6-MIB::cfgNetIpInterface.2 = br0.vlan13
```

7.3 Service Restrictions

To reduce the attack surface on a device, all services which are not required should be disabled.

Services to disable:

- CLI - WESTERMO-SW6-MIB::cfgCliEnabled.0
- HTTP - WESTERMO-SW6-MIB::cfgHttpEnabled.0
- LLDP - WESTERMO-SW6-MIB::cfgLldpEnabled.0
- MDNS - WESTERMO-SW6-MIB::cfgMdnsEnabled.0
- DHCP-Server - WESTERMO-SW6-MIB::cfgDhcpGlobalEnabled.0

7.3.1 CLI

As a general rule, the CLI should be disabled. In case the CLI must be used for any purpose whatsoever, it is recommended to use it with an SSH connection (WESTERMO-SW6-MIB::cfgCliSshEnabled.0) rather than a Telnet connection. Note that Telnet is considered insecure and is therefore disabled by default. If the CLI is activated it is bound to 0.0.0.0:22 for SSH. This means it is listening to all addresses by default. Consider binding the CLI to a specific address. The entries WESTERMO-SW6-MIB::cfgCliSshAddress.0 and WESTERMO-SW6-MIB::cfgCliTelnetAddress.0 allow the CLI to be reached only via the addresses under which it must be accessible. Assuming that the administrative interface is accessible at 192.168.1.20, the CLI should be bound to 192.168.1.20:22 and 192.168.1.20:23 respectively. However, the configuration parameters of the CLI allow for multiple addresses and ports.

7.3.2 SNMP

The SNMP service can not be disabled since it is the main internal configuration interface of the device. It is bound by default to 0.0.0.0:161. The entry WESTERMO-SW6-MIB::[cfgSnmpdAddress.0](#) allows to bind the service only to the addresses it should be reachable on. e.g. If the administrative interface is reachable at 192.168.1.20, the SNMP server should be bound to 192.168.1.20:161. If SNMP is not used, access to it can be prevented by binding it to the localhost only. e.g. 127.0.0.1:161. The SNMP server can be bound to multiple addresses and/or ports.

7.3.3 HTTP

The HTTP server provides access to the Web Interface via http and https. By default http on port 80 redirects to https on port 443. Whenever possible, keep the http to https redirection enabled ('WESTERMO-SW6-MIB::[cfgHttpRedirectEnabled.0](#)'). The entries WESTERMO-SW6-MIB::[cfgHttpHttpAddress.0](#) and WESTERMO-SW6-MIB::[cfgHttpHttpsAddress.0](#) allow to bind the service only to the addresses it should be reachable on. e.g. If the administrative interface is reachable at 192.168.1.20, the HTTP server should be bound to 192.168.1.20:80 respectively 192.168.1.20:443. If unencrypted access to port 80 is not used, access to it can be prevented by binding WESTERMO-SW6-MIB::[cfgHttpHttpAddress.0](#) to the localhost only. e.g. 127.0.0.1:80. The HTTP server can be bound to multiple addresses and/or ports.

7.4 Passwords

All passwords should be changed from their default value. A list of the relevant config parameters:

- CLI - username: WESTERMO-SW6-MIB::[cfgCliUsername.0](#)
- CLI - password: WESTERMO-SW6-MIB::[cfgCliPassword.0](#)
- HTTP - password: WESTERMO-SW6-MIB::[cfgHttpPassword.0](#)
- SNMP - admin-community/passphrase: WESTERMO-SW6-MIB::[cfgSnmpdComAdmin.0](#)
- SNMP - maintainer-community/passphrase: WESTERMO-SW6-MIB::[cfgSnmpdComMaintainer.0](#)
- SNMP - monitor-community/passphrase: WESTERMO-SW6-MIB::[cfgSnmpdComMonitor.0](#)
- WLAN - PSK-passphrase: WESTERMO-SW6-MIB::[cfgWlanIfacePassword.X](#)

7.4.1 Strength Of PSK-Passphrase

When using a private shared key (PSK), it is strongly suggested to use a maximum sized random string of characters as PSK-passphrase. The maximum length of the passphrase is 63 characters, and should be 63 characters long. This is a countermeasure against an adversary trying to brute-force the passphrase. Ensure that the passphrase does not contain any human readable words in any language, since they are susceptible to dictionary attacks.

8 Default settings

The following table shows default value of selected configuration parameters.

Network Settings:

cfgNetIpAddr	192.168.1.20/24
cfgNetIpProto	static(0)
cfgNetIpInterface	br0.vlan0
cfgNetEthBridge	br0(0)
cfgNetWlanBridge	br0(0)
cfgNetVlanBridge	br0(0)
cfgNetVlanVid	0

WLAN physical device Settings (*RT 11n* family products):

cfgWlanGlblCountry	WORLD
cfgWlanDevModulation	ng(10)
cfgWlanDevBandwidth	ht20(0)
cfgWlanDevFrequency	2412
cfgWlanDevPower	9

WLAN logical interface Settings:

cfgWlanIfcMode	AP(0)
cfgWlanIfcSsid	Rmodem1
cfgWlanIfcEncryption	wpa2(3)
cfgWlanIfcPassword	password
cfgWlanIfcBitrates	auto(-1)
cfgWlanIfcScanList	0

Routing settings:

cfgRouteDefGateway	0.0.0.0 (disabled)
--------------------	--------------------

9 WESTERMO-SW6-MIB

9.0.1 configuration

9.0.1.1 cfgSystem

9.0.1.1.1 cfgSysHostname

The hostname of the device

Valid characters for hostnames are ASCII(7) letters from a to z, the digits from 0 to 9, and the hyphen (-). A hostname may not start with a hyphen.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1.1

9.0.1.1.2 cfgSysSearchdomainTable

Domain Search List

Configure the domain search list by adding entries in this table.

The domain search list, as well as the local domain name (see `cfgSysDomain`), is used by the resolver to create a fully qualified domain name from a relative name.

Applies to AP and STA.

<i>Status</i>	current
<i>Range</i>	0 - 5
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1.10

9.0.1.1.3 cfgSysSdSearch

Search Domain List Entry

This entry will be ignored when set to 'none'.

Example:

- example.com
- subdomain.otherdomain.org

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1.10.1.2

9.0.1.1.4 cfgSysNameserverTable

Nameserver configuration.

<i>Status</i>	current
<i>Range</i>	0 - 15
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1.11

9.0.1.1.5 cfgSysNsType

Type of the Nameserver Entry

- **none(0)**: Disables this entry
- **server(1)**: Uses the address specified by `cfgSysNsServer`
- **dhcpinterface(2)**: Uses nameservers provided by a DHCP-client referenced by `cfgSysNsDhcpInterface`

Applies to AP and STA.

<i>Enumeration</i>	none (0), server (1), dhcpinterface (2)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1.11.1.2

9.0.1.1.6 cfgSysNsServer

Nameserver Address

This parameter is only used when `cfgSysNsType` is set to **server(1)**.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	IpAddress
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1.11.1.3

9.0.1.1.7 cfgSysNsDhcpInterface

DHCP Client Interface

This parameter is only used when `cfgSysNsType` is set to **interface(2)**.

Name of an interface on which a DHCP client is running. This may be a DHCP client defined by `cfgNetIpTable` or a `wan` or `ovpn` interface which have their own means of handling DHCP.

Examples:

- wlan0
- ovpn0
- macvlan2
- wwan0
- br0.vlan7

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 16
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1.11.1.4

9.0.1.1.8 cfgSysTimezone

POSIX timezone string, defines the local time

For more strings also see http://wiki.openwrt.org/doc/uci/system#time_zones

Example:

- Europe/Zurich: CET-1CEST,M3.5.0,M10.5.0/3

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1.2

9.0.1.1.9 cfgSysDomain

Local Domain Name of the Device

Will be ignored when set to 'none'.

The local domain name, as well as the domain search list (as configured in `cfgSysSearchdomainTable`), is used by the resolver to create a fully qualified domain name from a relative name.

Example:

- `yourdomain.org`
- `subdomain.yourdomain.org`

Note:

It is recommended to not use the domain `local` because it collides with mDNS.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1.3

9.0.1.2 cfgCli

9.0.1.2.1 cfgCliEnabled

CLI feature support configuration.

Applies to AP and STA.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.100.1

9.0.1.2.2 **cfgCliUsername**

CLI username.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 31
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.100.2

9.0.1.2.3 **cfgCliPassword**

CLI password.

For SSH, a password is mandatory. Accessing the device via telnet without using a password is done by setting the password to an empty string (zero string).

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	0 - 63
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.100.3

9.0.1.2.4 **cfgCliTelnetEnabled**

CLI telnet support configuration.

Applies to AP and STA.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.100.4

9.0.1.2.5 **cfgCliSshEnabled**

CLI ssh support configuration.

Applies to AP and STA.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.100.6

9.0.1.2.6 **cfgCliTelnetAddress**

Address to which the telnet server for CLI binds.

The default is '0.0.0.0:23'. Multiple space and/or comma separated tuples are allowed.

Examples:

- 192.168.1.20:23
- 192.168.1.20:23, 192.168.2.20:8023, 172.16.32.32:10023
- 192.168.1.20:23 192.168.2.20:8023 172.16.32.32:10023

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.100.8

9.0.1.2.7 **cfgCliSshAddress**

Address to which the ssh server for CLI binds.

The default is '0.0.0.0:22'. Multiple space and/or comma separated tuples are allowed.

Examples:

- 192.168.1.20:22
- 192.168.1.20:22, 192.168.2.20:8022, 172.16.32.32:10022
- 192.168.1.20:22 192.168.2.20:8022 172.16.32.32:10022

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.100.9

9.0.1.3 cfgVpn

9.0.1.3.1 cfgVpnOpenvpn

9.0.1.3.1.1 cfgVpnOpenvpnTable

OpenVPN Instance Table

Each entry in this table represents one instance of the OpenVPN service and is in a one-to-one relation with the OpenVPN Interface whose index is identical. Also see `cfgNetOpenvpnTable`.

Applies to AP and STA.

<i>Status</i>	current
<i>Range</i>	0 - 15
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1003.1.1

9.0.1.3.1.2 cfgVpnOpenvpnDevType

Interface Type

OpenVPN is designed to work with virtual network interface either of type tunnel or tap. The interface types on both sides of an OpenVPN connection must match.

Interface types are:

- **tun(0)**: to encapsulate IPv4 or IPv6 (OSI Layer 3), or
- **tap(1)**: to encapsulate Ethernet 802.3 (OSI Layer 2).

To be able to bridge an OpenVPN interface with `cfgNetOpenvpnBridge` its type must be **tap(1)**.

Applies to AP and STA.

<i>Enumeration</i>	tun (0), tap (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1003.1.1.1.10

9.0.1.3.1.3 cfgVpnOpenvpnCustomOptions

Custom OpenVPN Options

These options are appended to the OpenVPN configuration. This allows to set options not available via other configuration items. Set to `none` when no additional options shall be added.

When setting multiple options, separate them with a semicolon ;.

The full list of all available options is at: <https://openvpn.net/community-resources/reference-manual-for-openvpn-2-4/>

Prohibited options are:

- ipchange
- route-up
- route-pre-down
- up
- down
- script-security
- cd
- chroot
- log
- client-connect
- client-disconnect
- learn-address
- auth-user-pass-verify
- tls-verify

Essentially everything which calls a script or changes files.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	4 - 4095
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1003.1.1.1.1000

9.0.1.3.1.4 cfgVpnOpenvpnKeepaliveInterval

Keep-Alive Interval

Send keep-alive packets to the remote OpenVPN instance if no packets have been sent for at least the given number of seconds.

This parameter has two intended uses: * Compatibility with stateful firewalls * To provide a basis for the remote OpenVPN instance to detect the existence of its peer

Note: If OpenVPN is in client mode, this parameter may be overridden by the server.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1003.1.1.1.11

9.0.1.3.1.5 **cfgVpnOpenvpnKeepaliveTimeout**

Keep-Alive Timeout

This parameter specifies the number of seconds that trigger a restart of the OpenVPN instance if no keep-alive or other packet has been received from the remote side, see `cfgVpnOpenvpnKeepaliveInterval`.

Note: If OpenVPN is in client mode, this parameter may be overridden by the server.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1003.1.1.1.12

9.0.1.3.1.6 **cfgVpnOpenvpnConnectRetry**

Connect Retry Interval

Number of seconds to wait between connection attempts. Repeated reconnection attempts are slowed down after 5 retries per remote by doubling the wait time after each unsuccessful attempt.

The parameter `cfgVpnOpenvpnConnectRetryLimit` specifies the maximum value of wait time in seconds at which it gets capped.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	1 - 65535
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1003.1.1.1.13

9.0.1.3.1.7 **cfgVpnOpenvpnConnectRetryLimit**

Connect Retry Interval Limit

The maximum value of wait time in seconds (see `cfgVpnOpenvpnConnectRetry`) at which it gets capped.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	1 - 65535
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1003.1.1.1.14

9.0.1.3.1.8 cfgVpnOpenvpnCompress

Compression Algorithm

Specify a compression algorithm:

- **disabled(0)**: Data compression is turned off.
- **allowPush(1)**: Data compression is turned off, but may be enabled by the server later.
- **lzo(2)**: Lempel-Ziv-Oberhumer (LZO) algorithm
- **lz4(3)**: LZ4 algorithm (faster than LZO)
- **lz4v2(4)** OpenVPN optimised version of the LZ4 algorithm

The LZO algorithm provides a slightly better compression ratio than the LZ4 compression. However, it is considerably slower and should not be used unless for backward compatibility.

Applies to AP and STA.

<i>Enumeration</i>	disabled (0), allowPush (1), lzo (2), lz4 (3), lz4v2 (4)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1003.1.1.1.15

9.0.1.3.1.9 cfgVpnOpenvpnVerb

Log Verbosity

Each log verbosity level shows all messages from the previous levels. Level 3 is recommended for a good summary of what is happening.

- **0**: No output except fatal errors
- **1 - 4**: Normal usage range
- **5**: Output R and W characters to the console for each packet read and write operation, uppercase is used for TCP/UDP packets and lowercase is used for TUN/TAP packets.
- **6 - 11**: Levels for debugging purposes

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1003.1.1.1.16

9.0.1.3.1.10 `cfgVpnOpenvpnKeyPassword`

Password to Unlock Private Key

This is only required if the key is encrypted.

Applies to STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1003.1.1.1.17

9.0.1.3.1.11 `cfgVpnOpenvpnKeyType`

Key Type

- **asymmetric(0)**: Use an asymmetric encryption with certificate, private and public keys (TLS).
- **symmetric(1)**: Use encryption with a static key.
- **combined(2)**: Use asymmetric encryption and encrypt the control channel with a static key.

Note: The key material for the asymmetric encryption is managed by the Certificate Store, see `setCrtFileSelector`.

Applies to AP and STA.

<i>Enumeration</i>	asymmetric (0), symmetric (1), combined (2)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1003.1.1.1.18

9.0.1.3.1.12 `cfgVpnOpenvpnKeyDirection`

Key Direction

This option is active when `cfgVpnOpenvpnKeyType` is set to either **symmetric(1)** or **combined(2)**.

- **omitted(-1)**: No direction is defined
- **zero(0)**: Use direction 0
- **one(1)**: Use direction 1

When the direction parameter is omitted, 2 keys are used bidirectionally: One for HMAC and the other for encryption/decryption.

With a direction specified, 4 keys are used: One per direction for HMAC and encryption.

Note: The direction parameter should always be complementary on either side of the connection, i.e. one side should use '0' and the other should use '1', or both sides should omit it altogether.

Applies to AP and STA.

<i>Enumeration</i>	omitted (-1), zero (0), one (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1003.1.1.1.19

9.0.1.3.1.13 cfgVpnOpenvpnMode

Mode

- **client(0):** OpenVPN instance is a client and connects to a server
- **server(1): NOT IMPLEMENTED**

Applies to AP and STA.

<i>Enumeration</i>	client (0), server (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1003.1.1.1.2

9.0.1.3.1.14 cfgVpnOpenvpnRemoteCertTls

Remote Certificate Verification

Verify if the Extended Key Usage field in the certificate of the remote host has the correct type.

- **disabled(0):** Do not verify the remote certificate.
- **client(1):** Check for client type.
- **server(2):** Check for server type.

Applies to AP and STA.

<i>Enumeration</i>	disabled (0), client (1), server (2)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1003.1.1.1.20

9.0.1.3.1.15 cfgVpnOpenvpnVerifyX509Name

X.509 Certificate Verification Method

The following verification methods are available:

- **disabled(0)**: No verification is done
- **name(1)**: Match the exact CN (Common Name)
- **prefix(2)**: Match the prefix of the CN
- **subject(3)**: Match the complete subject DN

The `cfgVpnOpenvpnVerifyX509String` parameter defines the string to be matched.

Applies to AP and STA.

<i>Enumeration</i>	disabled (0), name (1), prefix (2), subject (3)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	'number': '1', 'nodetype': 'namednumber'
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1003.1.1.1.21

9.0.1.3.1.16 `cfgVpnOpenvpnVerifyX509String`

X.509 Certificate Verification String

If a X.509 certificate verification method is enabled (see `cfgVpnOpenvpnVerifyX509Name`), this parameter defines the string to be compared by the verification method.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1003.1.1.1.22

9.0.1.3.1.17 `cfgVpnOpenvpnUsername`

Username

Authenticate with the server using the given username.

It is disabled when set to **none**.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1003.1.1.1.23

9.0.1.3.1.18 cfgVpnOpenvpnPassword

Password

Authenticate with the server using the given password.

It is disabled when set to **none**.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1003.1.1.1.24

9.0.1.3.1.19 cfgVpnOpenvpnLocal

Local IP Address

The OpenVPN instance binds to the given IP address only. Address 0.0.0.0 binds the OpenVPN instance to all interfaces.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	7 - 15
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1003.1.1.1.3

9.0.1.3.1.20 cfgVpnOpenvpnLPort

Local TCP/UDP Port

Specifies the TCP/UDP port number for bind. If the local port number is set to 0, OpenVPN uses a random port number.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	0 - 65535
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1003.1.1.1.4

9.0.1.3.1.21 **cfgVpnOpenvpnRemote**

Remote Host Name or IP Address

The OpenVPN client tries to connect to a server at the given remote host name or IP address.

The remote option will be omitted from the OpenVPN config file when set to 'none'. This allows to specify your own remote entries via the custom options (see `cfgVpnOpenvpnCustomOptions`).

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	6 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1003.1.1.1.5

9.0.1.3.1.22 **cfgVpnOpenvpnRPort**

Remote TCP/UDP Port

Specifies the TCP/UDP port to which the connection is created. This is the port on which the OpenVPN server is listening.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	1 - 65535
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1003.1.1.1.6

9.0.1.3.1.23 **cfgVpnOpenvpnProto**

Transport Protocol

The following transport protocols are available:

- **UDP(0)**: User Datagram Protocol
- **TCP(1)**: Transmission Control Protocol

Applies to AP and STA.

<i>Enumeration</i>	udp (0), tcp (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1003.1.1.1.7

9.0.1.3.1.24 `cfgVpnOpenvpnAuth`

Packet Authentication

Authenticate packets with a Hash-based Message Authentication Code HMAC using the given message digest algorithm.

In static-key encryption mode, the HMAC key is included in the key file. In TLS mode, the HMAC key is dynamically generated and shared between peers via the TLS control channel.

Examples:

- **SHA256**
- **SHA3-512**
- **none**: to disable HMAC packet authentication

For a full list of supported algorithms please consult the user manual or execute **openvpn --show-digests** on a device.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1003.1.1.1.8

9.0.1.3.1.25 `cfgVpnOpenvpnCipher`

Packet Encryption

Encrypt packets with the given cipher algorithm.

Examples:

- **AES-256-CBC**
- **AES-256-GCM**
- **none**: to disable packet encryption

For a full list of supported algorithms please consult the user manual or execute **openvpn --show-ciphers** on a device.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1003.1.1.1.9

9.0.1.3.2 cfgVpnIpsec

9.0.1.3.2.1 cfgVpnIpsecTable

IPsec Table

Applies to AP and STA.

<i>Status</i>	current
<i>Range</i>	0 - 15
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1003.2.1

9.0.1.3.2.2 cfgVpnIpsecType

TECHPREVIEW: IPsec Type

Applies to AP and STA.

<i>Enumeration</i>	tunnel (0), transport (1), transportProxyx (2), passthrough (3), drop (4)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1003.2.1.1.100

9.0.1.3.2.3 cfgVpnIpsecAuto

TECHPREVIEW: IPsec Auto Startup Operation

auto = ignore | add | route | start

Applies to AP and STA.

<i>Enumeration</i>	ignore (0), add (1), route (2), start (3)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1003.2.1.1.101

9.0.1.3.2.4 cfgVpnIpsecKeyExchange

TECHPREVIEW: Key Exchange Method

keyexchange = ikev1 | ikev2

Applies to AP and STA.

<i>Enumeration</i>	ikev1 (1), ikev2 (2)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1003.2.1.1.102

9.0.1.3.2.5 cfgVpnIpsecMobike

TECHPREVIEW: Enable or Disable IKEv2 MOBIKE Protocol

mobike = yes | no

Enables the IKEv2 MOBIKE protocol defined by RFC 4555. If set to no, the charon daemon will not actively propose MOBIKE as initiator and ignore the MOBIKE_SUPPORTED notify as responder.

Applies to AP and STA.

<i>Enumeration</i>	no (0), yes (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1003.2.1.1.103

9.0.1.3.2.6 cfgVpnIpsecIke

TECHPREVIEW: IKE/ISAKMP SA Encryption/Authentication Algorithms

ike =

Comma-separated list of IKE/ISAKMP SA encryption/authentication algorithms to be used, e.g. aes128-sha256-modp3072. The notation is encryption-integrity[-prf]-dhgroup. In IKEv2, multiple algorithms and proposals may be included, such as aes128-aes256-sha1-modp3072-modp2048,3des-sha1-md5-modp1024.

It is possible to configure a PRF algorithm different to that defined for integrity protection. If no PRF is configured, the algorithms defined for integrity are proposed as PRF. The prf keywords are the same as the integrity algorithms, but have a prf prefix (such as prfsha1, prfsha256 or prfaesxcbc).

Defaults to aes128-sha256-modp3072 (aes128-sha1-modp2048,3des-sha1-modp1536 before 5.4.0) for IKEv1. The daemon adds its extensive default proposal to this default or the configured value. To restrict it to the configured proposal an exclamation mark (!) can be added at the end.

Refer to IKEv1CipherSuites and IKEv2CipherSuites for a list of valid keywords.

Note: As a responder both daemons accept the first supported proposal received from the peer. In order to restrict a responder to only accept specific cipher suites, the strict flag (!, exclamation mark) can be used, e.g: aes256-sha512-modp4096!

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1003.2.1.1.104

9.0.1.3.2.7 cfgVpnIpsecEsp

TECHPREVIEW: ESP Encryption/Authentication Algorithms

esp =

Comma-separated list of ESP encryption/authentication algorithms to be used for the connection, e.g. aes128-sha256. The notation is encryption-integrity[-dhgroup][-esnmode]. For IKEv2, multiple algorithms (separated by -) of the same type can be included in a single proposal. IKEv1 only includes the first algorithm in a proposal. Only either the ah or the esp keyword may be used, AH+ESP bundles are not supported.

Defaults to aes128-sha256. The daemon adds its extensive default proposal to this default or the configured value. To restrict it to the configured proposal an exclamation mark (!) can be added at the end.

Note: As a responder, the daemon defaults to selecting the first configured proposal that's also supported by the peer. By disabling charon.prefer_configured_proposals in strongswan.conf this may be changed to selecting the first acceptable proposal sent by the peer instead. In order to restrict a responder to only accept specific cipher suites, the strict flag (!, exclamation mark) can be used, e.g: aes256-sha512-modp4096!

If dh-group is specified, CHILD_SA rekeying and initial negotiation include a separate Diffie-Hellman exchange (this also applies to IKEv1 Quick Mode). However, for IKEv2, the keys of the CHILD_SA created implicitly with the IKE_SA will always be derived from the IKE_SA's key material. So any DH group specified here will only apply when the CHILD_SA is later rekeyed or is created with a separate CREATE_CHILD_SA exchange. Therefore, a proposal mismatch might not immediately be noticed when the SA is established, but may later cause rekeying to fail.

Valid values for esnmode are esn and noesn. Specifying both negotiates extended sequence number support with the peer, the default is noesn.

Refer to IKEv1CipherSuites and IKEv2CipherSuites for a list of valid keywords.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1003.2.1.1.105

9.0.1.3.2.8 `cfgVpnIpsecIkeLifetime`

TECHPREVIEW: IKE Lifetime

`ikelifetime =`

How long the keying channel of a connection (ISAKMP or IKE SA) should last before being renegotiated.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1003.2.1.1.106

9.0.1.3.2.9 `cfgVpnIpsecLifetime`

TECHPREVIEW: Connection Instance Lifetime

`lifetime =`

How long a particular instance of a connection (a set of encryption/authentication keys for user packets) should last, from successful negotiation to expiry; acceptable values are an integer optionally followed by s (a time in seconds) or a decimal number followed by m, h, or d (a time in minutes, hours, or days respectively) (default 1h, maximum 24h). Normally, the connection is renegotiated (via the keying channel) before it expires (see `margintime`). The two ends need not exactly agree on lifetime, although if they do not, there will be some clutter of superseded connections on the end which thinks the lifetime is longer.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1003.2.1.1.107

9.0.1.3.2.10 `cfgVpnIpsecKeyingTries`

TECHPREVIEW: Keying Tries

`keyingtries = 3 | | %forever`

When set to -1 means %forever, otherwise what is set.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1003.2.1.1.108

9.0.1.3.2.11 **cfgVpnIpsecRekeyFuzz**

TECHPREVIEW: Rekey Fuzz

rekeyfuzz = 100% |

Maximum percentage by which marginbytes, marginpackets and margintime should be randomly increased to randomize rekeying intervals (important for hosts with many connections); acceptable values are an integer, which may exceed 100, followed by a '%' . The value of marginTYPE, after this random increase, must not exceed lifeTYPE (where TYPE is one of bytes, packets or time). The value 0% will suppress randomization. Relevant only locally, other end need not agree on it.

Also see Expiry and Rekey.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1003.2.1.1.109

9.0.1.3.2.12 **cfgVpnIpsecMarginTime**

TECHPREVIEW: Margin Time

margintime = 9m |

How long before connection expiry or keying-channel expiry should attempts to negotiate a replacement begin; acceptable values as for lifetime (default 9m). Relevant only locally, other end need not agree on it.

Also see Expiry and Rekey.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1003.2.1.1.110

9.0.1.3.2.13 `cfgVpnIpsecDpdAction`

TECHPREVIEW: Dead Peer Detection Protocol Usage

`dpdaction = none | clear | hold | restart`

Controls the use of the Dead Peer Detection protocol (DPD, RFC 3706) where R_U_THERE notification messages (IKEv1) or empty INFORMATIONAL messages (IKEv2) are periodically sent in order to check the liveness of the IPsec peer. The values clear, hold, and restart all activate DPD and determine the action to perform on a timeout. With clear the connection is closed with no further actions taken. hold installs a trap policy, which will catch matching traffic and tries to re-negotiate the connection on demand. restart will immediately trigger an attempt to re-negotiate the connection.

The default is none which disables the active sending of DPD messages.

Applies to AP and STA.

<i>Enumeration</i>	none (0), clear (1), hold (2), restart (3)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1003.2.1.1.111

9.0.1.3.2.14 `cfgVpnIpsecDpdDelay`

TECHPREVIEW: DPD Delay

`dpddelay = 30s |`

Defines the period time interval with which R_U_THERE messages/INFORMATIONAL exchanges are sent to the peer. These are only sent if no other traffic is received. In IKEv2, a value of 0 sends no additional INFORMATIONAL messages and uses only standard messages (such as those to rekey) to detect dead peers.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1003.2.1.1.112

9.0.1.3.2.15 `cfgVpnIpsecDpdTimeout`

TECHPREVIEW: DPD Timeout

`dpdtimeout = 150s |`

Defines the timeout interval, after which all connections to a peer are deleted in case of inactivity.

This only applies to IKEv1, in IKEv2 the default retransmission timeout applies, as every exchange is used to detect dead peers.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1003.2.1.1.113

9.0.1.3.2.16 **cfgVpnIpsecKeyPassword**

TECHPREVIEW: Password to Unlock Private Key

This is only required if the key is encrypted.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1003.2.1.1.114

9.0.1.3.2.17 **cfgVpnIpsecPassword**

TECHPREVIEW: IPsec Password

When `cfgVpnIpsecRightAuth` is set to **psk**.

A preshared secret is most conveniently represented as a sequence of characters. The sequence cannot contain newline or double-quote characters. Alternatively, preshared secrets can be represented as hexadecimal or Base64 encoded binary values. A character sequence beginning with 0x is interpreted as sequence hexadecimal digits. Similarly, a character sequence beginning with 0s is interpreted as Base64 encoded binary data.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1003.2.1.1.115

9.0.1.3.2.18 **cfgVpnIpsecLeft**

TECHPREVIEW: Left or Local

left = | | %any | %any4 | %any6 | range | subnet

The IP address of the participant's public-network interface or one of several magic values. The value %any for the local endpoint signifies an address to be filled in (by automatic keying) during negotiation. If the local peer initiates the connection setup the routing table will be queried to determine the correct local IP address. In case the local peer is responding to a connection setup then any IP address that is assigned to a local interface will be accepted. The value %any4 restricts address selection to IPv4 addresses, the value %any6 restricts address selection to IPv6 addresses.

The prefix % in front of a fully-qualified domain name or an IP address will implicitly set leftallowany=yes.

leftallowany is a modifier for left, making it behave as %any although a concrete IP address has been assigned. Recommended for dynamic IP addresses that can be resolved by DynDNS at IPsec startup or update time.

If %any is used for the remote endpoint it literally means any IP address.

If an FQDN is assigned it is resolved every time a configuration lookup is done. If DNS resolution times out, the lookup is delayed for that time.

Connections can be limited to a specific range of hosts. To do so a range (10.1.0.0-10.2.255.255) or a subnet (10.1.0.0/16) can be specified, and multiple addresses, ranges and subnets can be separated by commas. While one can freely combine these items, to initiate the connection at least one non-range/subnet is required.

Please note that with the usage of wildcards multiple connection descriptions might match a given incoming connection attempt. The most specific description is used in that case.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1003.2.1.1.2

9.0.1.3.2.19 cfgVpnIpsecRight

TECHPREVIEW: Right or Remote

right = | | %any | %any4 | %any6 | range | subnet

The IP address of the participant's public-network interface or one of several magic values. The value %any for the local endpoint signifies an address to be filled in (by automatic keying) during negotiation. If the local peer initiates the connection setup the routing table will be queried to determine the correct local IP address. In case the local peer is responding to a connection setup then any IP address that

is assigned to a local interface will be accepted. The value %any4 restricts address selection to IPv4 addresses, the value %any6 restricts address selection to IPv6 addresses.

The prefix % in front of a fully-qualified domain name or an IP address will implicitly set rightallowany=yes.

rightallowany is a modifier for right, making it behave as %any although a concrete IP address has been assigned. Recommended for dynamic IP addresses that can be resolved by DynDNS at IPsec startup or update time.

If %any is used for the remote endpoint it literally means any IP address.

If an FQDN is assigned it is resolved every time a configuration lookup is done. If DNS resolution times out, the lookup is delayed for that time.

Connections can be limited to a specific range of hosts. To do so a range (10.1.0.0-10.2.255.255) or a subnet (10.1.0.0/16) can be specified, and multiple addresses, ranges and subnets can be separated by commas. While one can freely combine these items, to initiate the connection at least one non-range/subnet is required.

Please note that with the usage of wildcards multiple connection descriptions might match a given incoming connection attempt. The most specific description is used in that case.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1003.2.1.1.3

9.0.1.3.2.20 cfgVpnIpsecLeftSubnet

TECHPREVIEW: IPsec Left Subnet

leftsubnet = [[]][. . .]

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1003.2.1.1.4

9.0.1.3.2.21 **cfgVpnIpsecRightSubnet**

TECHPREVIEW: IPsec Right Subnet

rightsubnet = [[]],...

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1003.2.1.1.5

9.0.1.3.2.22 **cfgVpnIpsecLeftId**

TECHPREVIEW: IPsec Left ID

leftid =

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1003.2.1.1.6

9.0.1.3.2.23 **cfgVpnIpsecRightId**

TECHPREVIEW: IPsec Right ID

rightid =

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1003.2.1.1.7

9.0.1.3.2.24 **cfgVpnIpsecLeftAuth**

TECHPREVIEW: IPsec Left Authentication

leftauth =

Applies to AP and STA.

<i>Enumeration</i>	pubkeyRSA (0), pubkeyECDSA (1), pubkeyBLISS (2), psk (10)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1003.2.1.1.8

9.0.1.3.2.25 cfgVpnIpsecRightAuth

TECHPREVIEW: IPsec Right Authentication

rightauth =

Applies to AP and STA.

<i>Enumeration</i>	pubkeyRSA (0), pubkeyECDSA (1), pubkeyBLISS (2), psk (10)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1003.2.1.1.9

9.0.1.3.3 cfgVpnIpsecDebug

9.0.1.3.3.1 cfgVpnIpsecDbgApp

TECHPREVIEW: app: applications other than daemons

- **silent(-1):** Absolutely silent
- **basic(0):** Very basic auditing logs, (e.g. SA up/SA down)
- **generic(1):** Generic control flow with errors, a good default to see whats going on
- **detailed(2):** More detailed debugging control flow
- **raw(3):** Including RAW data dumps in hex
- **sensitive(4):** Also include sensitive material in dumps, e.g. keys

Applies to AP and STA.

<i>Enumeration</i>	silent (-1), basic (0), generic (1), detailed (2), raw (3), sensitive (4)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1003.4.1

9.0.1.3.3.2 cfgVpnIpsecDbgImv

TECHPREVIEW: imv: Integrity Measurement Verifier

- **silent(-1)**: Absolutely silent
- **basic(0)**: Very basic auditing logs, (e.g. SA up/SA down)
- **generic(1)**: Generic control flow with errors, a good default to see whats going on
- **detailed(2)**: More detailed debugging control flow
- **raw(3)**: Including RAW data dumps in hex
- **sensitive(4)**: Also include sensitive material in dumps, e.g. keys

Applies to AP and STA.

<i>Enumeration</i>	silent (-1), basic (0), generic (1), detailed (2), raw (3), sensitive (4)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1003.4.10

9.0.1.3.3.3 cfgVpnIpsecDbgJob

TECHPREVIEW: job: Jobs queuing/processing and thread pool management

- **silent(-1)**: Absolutely silent
- **basic(0)**: Very basic auditing logs, (e.g. SA up/SA down)
- **generic(1)**: Generic control flow with errors, a good default to see whats going on
- **detailed(2)**: More detailed debugging control flow
- **raw(3)**: Including RAW data dumps in hex
- **sensitive(4)**: Also include sensitive material in dumps, e.g. keys

Applies to AP and STA.

<i>Enumeration</i>	silent (-1), basic (0), generic (1), detailed (2), raw (3), sensitive (4)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1003.4.11

9.0.1.3.3.4 cfgVpnIpsecDbgKnl

TECHPREVIEW: knl: IPsec/Networking kernel interface

- **silent(-1)**: Absolutely silent
- **basic(0)**: Very basic auditing logs, (e.g. SA up/SA down)
- **generic(1)**: Generic control flow with errors, a good default to see whats going on
- **detailed(2)**: More detailed debugging control flow
- **raw(3)**: Including RAW data dumps in hex
- **sensitive(4)**: Also include sensitive material in dumps, e.g. keys

Applies to AP and STA.

<i>Enumeration</i>	silent (-1), basic (0), generic (1), detailed (2), raw (3), sensitive (4)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1003.4.12

9.0.1.3.3.5 cfgVpnIpsecDbgLib

TECHPREVIEW: lib: libstrongwan library messages

- **silent(-1):** Absolutely silent
- **basic(0):** Very basic auditing logs, (e.g. SA up/SA down)
- **generic(1):** Generic control flow with errors, a good default to see whats going on
- **detailed(2):** More detailed debugging control flow
- **raw(3):** Including RAW data dumps in hex
- **sensitive(4):** Also include sensitive material in dumps, e.g. keys

Applies to AP and STA.

<i>Enumeration</i>	silent (-1), basic (0), generic (1), detailed (2), raw (3), sensitive (4)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1003.4.13

9.0.1.3.3.6 cfgVpnIpsecDbgMgr

TECHPREVIEW: mgr: IKE_SA manager, handling synchronization for IKE_SA access

- **silent(-1):** Absolutely silent
- **basic(0):** Very basic auditing logs, (e.g. SA up/SA down)
- **generic(1):** Generic control flow with errors, a good default to see whats going on
- **detailed(2):** More detailed debugging control flow
- **raw(3):** Including RAW data dumps in hex
- **sensitive(4):** Also include sensitive material in dumps, e.g. keys

Applies to AP and STA.

<i>Enumeration</i>	silent (-1), basic (0), generic (1), detailed (2), raw (3), sensitive (4)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1003.4.14

9.0.1.3.3.7 cfgVpnIpsecDbgNet

TECHPREVIEW: net: IKE network communication

- **silent(-1):** Absolutely silent

- **basic(0)**: Very basic auditing logs, (e.g. SA up/SA down)
- **generic(1)**: Generic control flow with errors, a good default to see whats going on
- **detailed(2)**: More detailed debugging control flow
- **raw(3)**: Including RAW data dumps in hex
- **sensitive(4)**: Also include sensitive material in dumps, e.g. keys

Applies to AP and STA.

<i>Enumeration</i>	silent (-1), basic (0), generic (1), detailed (2), raw (3), sensitive (4)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1003.4.15

9.0.1.3.3.8 cfgVpnlpsecDbgPts

TECHPREVIEW: pts: Platform Trust Service

- **silent(-1)**: Absolutely silent
- **basic(0)**: Very basic auditing logs, (e.g. SA up/SA down)
- **generic(1)**: Generic control flow with errors, a good default to see whats going on
- **detailed(2)**: More detailed debugging control flow
- **raw(3)**: Including RAW data dumps in hex
- **sensitive(4)**: Also include sensitive material in dumps, e.g. keys

Applies to AP and STA.

<i>Enumeration</i>	silent (-1), basic (0), generic (1), detailed (2), raw (3), sensitive (4)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1003.4.16

9.0.1.3.3.9 cfgVpnlpsecDbgTls

TECHPREVIEW: tls: libtls library messages

- **silent(-1)**: Absolutely silent
- **basic(0)**: Very basic auditing logs, (e.g. SA up/SA down)
- **generic(1)**: Generic control flow with errors, a good default to see whats going on
- **detailed(2)**: More detailed debugging control flow
- **raw(3)**: Including RAW data dumps in hex
- **sensitive(4)**: Also include sensitive material in dumps, e.g. keys

Applies to AP and STA.

<i>Enumeration</i>	silent (-1), basic (0), generic (1), detailed (2), raw (3), sensitive (4)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1003.4.17

9.0.1.3.3.10 cfgVpnlpsecDbgTnc

TECHPREVIEW: tnc: Trusted Network Connect

- **silent(-1):** Absolutely silent
- **basic(0):** Very basic auditing logs, (e.g. SA up/SA down)
- **generic(1):** Generic control flow with errors, a good default to see whats going on
- **detailed(2):** More detailed debugging control flow
- **raw(3):** Including RAW data dumps in hex
- **sensitive(4):** Also include sensitive material in dumps, e.g. keys

Applies to AP and STA.

<i>Enumeration</i>	silent (-1), basic (0), generic (1), detailed (2), raw (3), sensitive (4)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1003.4.18

9.0.1.3.3.11 cfgVpnlpsecDbgAsn

TECHPREVIEW: asn: Low-level encoding/decoding (ASN.1, X.509 etc.)

- **silent(-1):** Absolutely silent
- **basic(0):** Very basic auditing logs, (e.g. SA up/SA down)
- **generic(1):** Generic control flow with errors, a good default to see whats going on
- **detailed(2):** More detailed debugging control flow
- **raw(3):** Including RAW data dumps in hex
- **sensitive(4):** Also include sensitive material in dumps, e.g. keys

Applies to AP and STA.

<i>Enumeration</i>	silent (-1), basic (0), generic (1), detailed (2), raw (3), sensitive (4)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1003.4.2

9.0.1.3.3.12 cfgVpnlpsecDbgCfg

TECHPREVIEW: cfg: Configuration management and plugins

- **silent(-1):** Absolutely silent

- **basic(0)**: Very basic auditing logs, (e.g. SA up/SA down)
- **generic(1)**: Generic control flow with errors, a good default to see whats going on
- **detailed(2)**: More detailed debugging control flow
- **raw(3)**: Including RAW data dumps in hex
- **sensitive(4)**: Also include sensitive material in dumps, e.g. keys

Applies to AP and STA.

<i>Enumeration</i>	silent (-1), basic (0), generic (1), detailed (2), raw (3), sensitive (4)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1003.4.3

9.0.1.3.3.13 cfgVpnIpsecDbgChd

TECHPREVIEW: chd: CHILD_SA/IPsec SA

- **silent(-1)**: Absolutely silent
- **basic(0)**: Very basic auditing logs, (e.g. SA up/SA down)
- **generic(1)**: Generic control flow with errors, a good default to see whats going on
- **detailed(2)**: More detailed debugging control flow
- **raw(3)**: Including RAW data dumps in hex
- **sensitive(4)**: Also include sensitive material in dumps, e.g. keys

Applies to AP and STA.

<i>Enumeration</i>	silent (-1), basic (0), generic (1), detailed (2), raw (3), sensitive (4)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1003.4.4

9.0.1.3.3.14 cfgVpnIpsecDbgDmn

TECHPREVIEW: dmn: Main daemon setup/cleanup/signal handling

- **silent(-1)**: Absolutely silent
- **basic(0)**: Very basic auditing logs, (e.g. SA up/SA down)
- **generic(1)**: Generic control flow with errors, a good default to see whats going on
- **detailed(2)**: More detailed debugging control flow
- **raw(3)**: Including RAW data dumps in hex
- **sensitive(4)**: Also include sensitive material in dumps, e.g. keys

Applies to AP and STA.

<i>Enumeration</i>	silent (-1), basic (0), generic (1), detailed (2), raw (3), sensitive (4)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1003.4.5

9.0.1.3.3.15 cfgVpnIpsecDbgEnc

TECHPREVIEW: enc: Packet encoding/decoding encryption/decryption operations

- **silent(-1):** Absolutely silent
- **basic(0):** Very basic auditing logs, (e.g. SA up/SA down)
- **generic(1):** Generic control flow with errors, a good default to see whats going on
- **detailed(2):** More detailed debugging control flow
- **raw(3):** Including RAW data dumps in hex
- **sensitive(4):** Also include sensitive material in dumps, e.g. keys

Applies to AP and STA.

<i>Enumeration</i>	silent (-1), basic (0), generic (1), detailed (2), raw (3), sensitive (4)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1003.4.6

9.0.1.3.3.16 cfgVpnIpsecDbgEsp

TECHPREVIEW: esp: libipsec library messages

- **silent(-1):** Absolutely silent
- **basic(0):** Very basic auditing logs, (e.g. SA up/SA down)
- **generic(1):** Generic control flow with errors, a good default to see whats going on
- **detailed(2):** More detailed debugging control flow
- **raw(3):** Including RAW data dumps in hex
- **sensitive(4):** Also include sensitive material in dumps, e.g. keys

Applies to AP and STA.

<i>Enumeration</i>	silent (-1), basic (0), generic (1), detailed (2), raw (3), sensitive (4)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1003.4.7

9.0.1.3.3.17 cfgVpnIpsecDbgIke

TECHPREVIEW: ike: IKE_SA/ISAKMP SA

- **silent(-1):** Absolutely silent

- **basic(0)**: Very basic auditing logs, (e.g. SA up/SA down)
- **generic(1)**: Generic control flow with errors, a good default to see whats going on
- **detailed(2)**: More detailed debugging control flow
- **raw(3)**: Including RAW data dumps in hex
- **sensitive(4)**: Also include sensitive material in dumps, e.g. keys

Applies to AP and STA.

<i>Enumeration</i>	silent (-1), basic (0), generic (1), detailed (2), raw (3), sensitive (4)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1003.4.8

9.0.1.3.3.18 cfgVpnIpsecDbgImc

TECHPREVIEW: imc: Integrity Measurement Collector

- **silent(-1)**: Absolutely silent
- **basic(0)**: Very basic auditing logs, (e.g. SA up/SA down)
- **generic(1)**: Generic control flow with errors, a good default to see whats going on
- **detailed(2)**: More detailed debugging control flow
- **raw(3)**: Including RAW data dumps in hex
- **sensitive(4)**: Also include sensitive material in dumps, e.g. keys

Applies to AP and STA.

<i>Enumeration</i>	silent (-1), basic (0), generic (1), detailed (2), raw (3), sensitive (4)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.1003.4.9

9.0.1.4 cfgCellular

9.0.1.4.1 cfgCellSimTable

SIM Parameter Table

Applies to cellular products only.

<i>Status</i>	current
<i>Range</i>	0 - 0
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.101.1

9.0.1.4.2 **cfgCellSimSlot1**

SIM Slot 1

This value references the index of the desired SIM profile in the `cfgCellSimProfileTable` for the SIM card in slot 1. A value of -1 disables slot 1.

Applies to cellular products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.101.1.1.2

9.0.1.4.3 **cfgCellSimSlot2**

SIM Slot 2

This value references the index of the desired SIM profile in the `cfgCellSimProfileTable` for the SIM card in slot 2. A value of -1 disables slot 2.

Applies to cellular products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.101.1.1.3

9.0.1.4.4 **cfgCellSimPrimarySlot**

Primary SIM Slot

First connection attempts are always done with the SIM card in the primary slot. Choose either

- **slot1(1)**
- **slot2(2)**

as primary SIM slot.

Note: If only one slot is activated, this parameter has no effect.

Applies to cellular products only.

<i>Enumeration</i>	slot1 (1), slot2 (2)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.101.1.1.4

9.0.1.4.5 **cfgCellSimProfileTable**

SIM Profiles

Applies to cellular products only.

<i>Status</i>	current
<i>Range</i>	0 - 9
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.101.2

9.0.1.4.6 **cfgCellSimProfileApn**

Access Point Name

If a specific Access Point Name (APN) is required by the service provider, it can be defined using this entry, otherwise it can be left at its default value **auto**.

Note: If this entry is set to **auto** the authentication entries `cfgCellSimProfileUsername`, `cfgCellSimProfileP` and `cfgCellSimProfileAuthType` are ingored.

Applies to cellular products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.101.2.1.2

9.0.1.4.7 **cfgCellSimProfileUsername**

Username

If the service provider requiries authentication, the username can be specified by this entry. If no authentication type is selected, see `cfgCellSimProfileAuthType`, this entry is ignored.

Applies to cellular products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.101.2.1.3

9.0.1.4.8 **cfgCellSimProfilePassword**

Password

Set the password for the user defined with `cfgCellSimProfileUsername`.

Applies to cellular products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.101.2.1.4

9.0.1.4.9 `cfgCellSimProfilePinEnabled`

PIN Authentication Disabled or Enabled

Set to **enabled(1)** if PIN authentication is required for the SIM card in the corresponding slot.

Applies to cellular products only.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.101.2.1.5

9.0.1.4.10 `cfgCellSimProfilePin`

PIN of the SIM Card

The PIN is ignored, when PIN authentication is disabled, see `cfgCellSimProfilePinEnabled`.

Applies to cellular products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	4 - 4
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.101.2.1.6

9.0.1.4.11 `cfgCellSimProfileAuthType`

Authentication Type

Select the authentication type, which is required by the service provider. If no authentication is required set this entry to its default value **none(0)**, otherwise choose one of the supported authentication types:

- **pap(1)**,

- **chap(2)**, or
- **both(3)**.

Applies to cellular products only.

<i>Enumeration</i>	none (0), pap (1), chap (2), both (3)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.101.2.1.7

9.0.1.4.12 **cfgCellSimProfileRoaming**

TECHPREVIEW: Roaming Disabled or Enabled

Roaming is enabled by default. However, setting this entry to **disabled(0)** prevents the use of other available cellular networks outside the range of the home network, although the SIM card would allow it.

Note: Disable roaming is currently not supported.

Applies to cellular products only.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.101.2.1.8

9.0.1.4.13 **cfgCellConnectionManagement**

9.0.1.4.13.1 **cfgCellConnMgmtSimRotationEnabled**

SIM Rotation Disabled or Enabled

Set to **enabled(1)** for rotating between primary and secondary SIM after a connection loss.

Applies to cellular products only.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.101.3.1

9.0.1.4.13.2 **cfgCellConnMgmtMonMode**

Monitor Mode

The monitor mode defines which algorithm the connection management is using to evaluate whether a cellular connection is up. The following monitor modes are supported:

- **signal(0)**: Monitor the cellular signal (e.g. RSSI level)
- **remoteHosts(1)**: Monitor the cellular network (e.g. the availability of remote network hosts).

Adjusting the monitor algorithm can be achieved by setting the `cfgCellConnMgmtMonPeriod` and `cfgCellConnMgmtMonPeriod`.

Using the **remoteHosts(1)** mode requires at least one active remote host in the `cfgCellConnMgmtMonRemoteTab`.

Applies to cellular products only.

<i>Enumeration</i>	signal (0), remoteHosts (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.101.3.2

9.0.1.4.13.3 `cfgCellConnMgmtMonPeriod`

Monitor Period

The monitor period defines the time in seconds between two consecutive evaluations of the cellular connection.

Applies to cellular products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	1 - 86400
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.101.3.3

9.0.1.4.13.4 `cfgCellConnMgmtMonCount`

Monitor Count

The monitor count defines the needed amount of consecutive failed connection tests before a connection is considered down.

Applies to cellular products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	1 - 10
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.101.3.4

9.0.1.4.13.5 **cfgCellConnMgmtMonRemoteTable**

Monitored Remotes

All hosts listed in this table are used by the **network(1)** mode, defined in `cfgCellConnMgmtMonMode`. If the connection management uses this algorithm to evaluate the status of the connection, it tests all hosts one after the other. As soon as one host is available the cellular connection is considered up.

Applies to cellular products only.

<i>Status</i>	current
<i>Range</i>	0 - 3
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.101.3.5

9.0.1.4.13.6 **cfgCellConnMgmtMonRemoteType**

Monitoring Type

The following methods are provided to check the availability of a remote host via the cellular network:

- **disabled(0)**: Ignore this host.
- **icmp(1)**: Use ICMP to ping the host.
- **tcp(2)**: Probe a TCP/IP port of the host.

Applies to cellular products only.

<i>Enumeration</i>	disabled (0), icmp (1), tcp (2)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.101.3.5.1.2

9.0.1.4.13.7 **cfgCellConnMgmtMonRemoteAddress**

Remote Address

The address of a remote host can either be defined by an IP address or a host name. If `cfgCellConnMgmtMonRemoteType` is set to **tcp(2)**, an additional port number must be specified. The format of a valid remote address is:

<IP address|host name>[:port]

Examples:

- **icmp(1)**: 8.8.8.8
- **tcp(2)**: www.example.com:80

Applies to cellular products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.101.3.5.1.3

9.0.1.5 cfgLogging

9.0.1.5.1 cfgLogRemote

9.0.1.5.1.1 cfgLogRemoteTable

List of Syslog Destinations

<i>Status</i>	current
<i>Range</i>	0 - 3
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.11.2.1

9.0.1.5.1.2 cfgLogRemoteEnabled

Log Syslog messages to a remote server in accordance to RFC 5424.

Applies to AP and STA.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.11.2.1.1.2

9.0.1.5.1.3 cfgLogRemoteLevel

Log only messages with equal or higher priority than prio N (0-7).

Applies to AP and STA.

<i>Enumeration</i>	emergency (0), alert (1), critical (2), error (3), warning (4), notice (5), info (6), debug (7)
<i>Access</i>	readonly
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.11.2.1.1.3

9.0.1.5.1.4 **cfgLogRemoteProtocol**

Protocol to Send Log Messages

The udp(0) protocol complies with the standard syslog protocol.

Applies to AP and STA.

<i>Enumeration</i>	udp (0), tcp (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.11.2.1.1.4

9.0.1.5.1.5 **cfgLogRemotelp**

Remote IP address.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	IpAddress
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.11.2.1.1.5

9.0.1.5.1.6 **cfgLogRemotePort**

Remote Port

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.11.2.1.1.6

9.0.1.5.1.7 **cfgLogRemoteType**

TECHPREVIEW: Bitfield to Control Remote Syslog Type

- **0x00** - no remote syslog
- **0x01** - standard syslog
- **0x02** - security syslog
- **0x04** - commissioning syslog

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	0 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.11.2.1.1.7

9.0.1.6 cfgSnmp

9.0.1.6.1 cfgSnmpd

9.0.1.6.1.1 cfgSnmpdLocation

SNMP system location.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.12.1.1

9.0.1.6.1.2 cfgSnmpdCommunity

cfgSnmpdComAdmin

Password for the administrator.

This is the community or the passphrase for the user administrator depending on the cfgSnmpdVersion:

- **v2c:** community string for administrator
- **v3usm:** passphrase for authentication and privacy for user admin

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.12.1.100.1

cfgSnmpdComMaintainer

Password for the maintainer.

This is the community or the passphrase for the user maintainer depending on the cfgSnmpdVersion:

- **v2c**: community string for maintainer
- **v3usm**: passphrase for authentication and privacy for user maintainer

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.12.1.100.2

cfgSnmpdComMonitor

Password for the monitor.

This is the community or the passphrase for the user monitor depending on the cfgSnmpdVersion:

- **v2c**: community string for monitor
- **v3usm**: passphrase for authentication and privacy for user monitor

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.12.1.100.3

cfgSnmpdContact

SNMP contact.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.12.1.2

cfgSnmpdVersion

SNMP version, either **v2c(0)** or **v3usm(1)**, for the users admin, maintainer and monitor.

Please refer to the user guide for more information about the access rights of the three predefined users.

The User-based Security Model (USM), which is the default Security Module for SNMPv3, with the authentication type `cfgSnmpdAuthType` and the privacy protocol defined in `cfgSnmpdPrivType` is implemented and can be chosen by setting this value to **v3usm(1)**.

Setting **v3usm(1)** disables access to the device via SNMPv2.

SNMPv3 USM configuration:

- **User:** admin, maintainer or monitor
- **Authentication Protocol:** `cfgSnmpdAuthType`
- **Privacy Protocol:** `cfgSnmpdPrivType`

Applies to AP and STA.

<i>Enumeration</i>	v2c (0), v3usm (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.12.1.3

cfgSnmpdName

SNMP node name often used in NMS.

By convention, this is the node's fully-qualified domain name.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.12.1.4

cfgSnmpdEnabled

Enable or disable the SNMP Agent.

Applies to AP and STA.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.12.1.5

cfgSnmpdAddress

Address to which the SNMP Agent binds.

The default is 'udp:0.0.0.0:161'. When defining a configuration, default arguments may be omitted. Multiple space and/or comma separated tuples are allowed.

Supported transport protocols are:

- UDP

Examples:

- udp:192.168.1.20:161
- 192.168.1.20, 10.0.0.1:10161, udp:172.16.32.32:30161
- 192.168.1.20 10.0.0.1:10161 udp:172.16.32.32:30161

More at <http://www.net-snmp.org/docs/man/snmpd.examples.html#lbaE>

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.12.1.6

cfgSnmpdAuthType

TECHPREVIEW: Authentication Type for SNMPv3

- **sha1(2):** SHA-1
- **sha384(5):** SHA-384
- **sha512(6):** SHA-512

Applies to AP and STA.

<i>Enumeration</i>	sha1 (2), sha384 (5), sha512 (6)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.12.1.7

cfgSnmpdPrivType

TECHPREVIEW: Encryption Type for SNMPv3

- **aes128(2):** AES-128
- **aes256(4):** AES-256

Applies to AP and STA.

<i>Enumeration</i>	aes128 (2), aes256 (4)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.12.1.8

9.0.1.6.2 cfgSnmpTrap

9.0.1.6.2.1 cfgSnmpTrapEnabled

Enable SNMP Notifications.

Applies to AP and STA.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.12.10.1

9.0.1.6.2.2 cfgSnmpTrapAuthProtocol

TECHPREVIEW: SNMP Authentication Protocol

Set the authentication protocol used for authenticated SNMPv3 messages.

Note: Only used if `cfgSnmpTrapVersion` is **usm(2)**.

Applies to AP and STA.

<i>Enumeration</i>	none (0), sha1 (2), sha384 (5), sha512 (6)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.12.10.10

9.0.1.6.2.3 cfgSnmpTrapAuthPassword

TECHPREVIEW: Authentication Password

Set the authentication password used for authenticated SNMPv3 messages.

Note: Only used if `cfgSnmptTrapVersion` is **usm(2)**.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	0 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.12.10.11

9.0.1.6.2.4 `cfgSnmptTrapPrivProtocol`

TECHPREVIEW: SNMP Privacy Protocol

Note: Only used if `cfgSnmptTrapVersion` is **usm(2)**.

Applies to AP and STA.

<i>Enumeration</i>	none (0), aes128 (2), aes256 (4)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.12.10.12

9.0.1.6.2.5 `cfgSnmptTrapPrivPassword`

TECHPREVIEW: Privacy Password

Set the privacy password used for encrypted SNMPv3 messages.

Note: Only used if `cfgSnmptTrapVersion` is **usm(2)**.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	0 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.12.10.13

9.0.1.6.2.6 `cfgSnmptTrapTimeout`

TECHPREVIEW: SNMP Trap Timeout

Specifies the timeout in seconds between retries.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	1 - 2147483647
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.12.10.14

9.0.1.6.2.7 cfgSnmpTrapRetries

TECHPREVIEW: SNMP Trap Retries

Specifies the number of retries to be used for INFORM notifications.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	0 - 2147483647
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.12.10.15

9.0.1.6.2.8 cfgSnmpTrapVersion

SNMP Version

SNMP notifications can be sent in the following versions:

- **v1(0):** Version 1 (RFCs 1155-1157), this version is obsolete and will default to **v2c(1)**.
- **v2c(1):** Version 2c (RFCs 1901-1908)
- **usm(2):** Version 3 (RFCs 2571-2574)

Applies to AP and STA.

<i>Enumeration</i>	v1 (0), v2c (1), usm (2)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.12.10.2

9.0.1.6.2.9 cfgSnmpTrapCommunity

SNMP Community.

Note: Only used for SNMP versions v1(0) and v2c(1).

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.12.10.3

9.0.1.6.2.10 cfgSnmpTrapDest

IP address of the trap receiver.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	IpAddress
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.12.10.4

9.0.1.6.2.11 cfgSnmpTrapType

TECHPREVIEW: SNMP Notification Type

- **trap(1):** Send all SNMP notifications as TRAPv1 or TRAPv2 depending on the SNMP version **v1(0)** or **v2c(1)**, respectively.
- **inform(2):** Send all SNMP notifications as INFORM. Note that INFORM PDUs are not supported for SNMP version **v1(0)**.

Applies to AP and STA.

<i>Enumeration</i>	trap (1), inform (2)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.12.10.5

9.0.1.6.2.12 cfgSnmpTrapEngineId

TECHPREVIEW: SNMP Authoritative Engine ID

Set the authoritative (security) engine ID used for SNMPv3 messages, given as a hexadecimal string (optionally prefixed by '0x'). The value must be between 5 and 32 octets long. This engine ID will be discovered automatically if this parameter is set to 'auto'.

Note: Only used if `cfgSnmpTrapVersion` is **usm(2)**.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	0 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.12.10.6

9.0.1.6.2.13 cfgSnmpTrapUser

TECHPREVIEW: SNMP User / Security Name

Set the user name used for authenticated SNMPv3 messages.

Note: Only used if `cfgSnmpTrapVersion` is **usm(2)**.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	0 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.12.10.9

9.0.1.7 cfgDhcp

9.0.1.7.1 cfgDhcpGlobal

9.0.1.7.1.1 cfgDhcpGlobalEnabled

Enable DNS/DHCP Server Functionality

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.13.1.1

9.0.1.7.1.2 cfgDhcpDnsmasqTable

DNS/DHCP Server Instances

Dnsmasq serves as DNS and DHCP server. A single instance may serve multiple interfaces and scopes. It may run as DNS server only, or as DHCP server only.

<i>Status</i>	current
<i>Range</i>	0 - 9
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.13.2

9.0.1.7.1.3 **cfgDhcpDnsmasqScopeParameter**

Parameter to Set Which Scope ID to use for the DHCP Server

This is used in conjunction with the scope ID parameter `cfgDhcpScopeId`).

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	0 - 8
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.13.2.1.2

9.0.1.7.1.4 **cfgDhcpDnsmasqDnsPort**

DNS Server Port

The UDP and TCP port on which the DNS service is running.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	1 - 65535
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.13.2.1.3

9.0.1.7.1.5 **cfgDhcpDnsmasqDnsListenAddress**

DNS Server Listen Address

Multiple space and/or comma separated addresses are allowed.

When set to 'auto', this instance binds to all addresses on interfaces specified by `cfgDhcpScopeInterface` in the referenced `cfgDhcpDnsmasqScopeParameter`. When no interfaces are referenced, this instance binds to any address in the system.

When set to 'wildcard', this instance binds to '0.0.0.0' and ':::'.

Examples:

- auto
- wildcard
- 127.0.0.1, 192.168.1.20
- 127.0.0.1 192.168.1.20
- 192.168.1.20,10.0.88.1,10.0.99.1

Note: When a dnsmasq instance binds a port to wildcard, no other instance may bind the same port.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.13.2.1.4

9.0.1.7.1.6 cfgDhcpDnsmasqDnsResolveOrder

Upstream DNS Server Resolve Order

Defines in what order the upstream DNS servers shall be queried.

- **any(0):** Send queries to any of the upstream servers and try to favour servers that are known to be up.
- **strictorder(1):** Try each query with each server strictly in the order they are configured in the `cfgSysNameserverTable`.
- **all(2):** Send all queries to all available servers. The reply from the server which answers first will be returned to the original requester.

<i>Enumeration</i>	any (0), strictorder (1), all (2)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.13.2.1.5

9.0.1.7.1.7 cfgDhcpDnsmasqDnsEnabled

DNS Server Disabled or Enabled

Enable the DNS Server functionality of this instance.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.13.2.1.6

9.0.1.7.1.8 cfgDhcpDnsmasqDhcpEnabled

DHCP Server Disabled or Enabled

Enable the DHCP Server functionality of this instance.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.13.2.1.7

9.0.1.7.1.9 **cfgDhcpDnsmasqDnsStopDnsRebind**

DNS Rebind Protection

Reject and log to syslog addresses from upstream nameservers which are in the private ranges (RFC1918). This blocks an attack where a browser behind a firewall is used to probe machines on the local network.

When using split-DNS, use `cfgDhcpDnsmasqDnsRebindDomainOk` to specify domains that are allowed to resolve to private addresses.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.13.2.1.8

9.0.1.7.1.10 **cfgDhcpDnsmasqDnsRebindDomainOk**

Allowed Domains for DNS Rebind Protection

This entry is active when `cfgDhcpDnsmasqDnsStopDnsRebind` is enabled.

Enter a space and/or comma separated list of domains which are allowed to resolve to private addresses (RFC1918).

No Domains are excepted when set to `none`.

Examples:

- example.com, example.net, example.org
- yourdomain.com anotherdomain.com

****Note:**** Subdomains are included when excepting domains. e.g when `domain.net` is set, then `subdomain1.domain.net` and `subdomain2.domain.net` are excepted as well.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.13.2.1.9

9.0.1.7.1.11 cfgDhcpScopeTable

DHCP Instance Configs

<i>Status</i>	current
<i>Range</i>	0 - 7
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.13.3

9.0.1.7.1.12 cfgDhcpScopeDhcpOptions

DHCP Scope Custom Options

This config option allows to specify arbitrary DHCP options which will be sent to DHCP clients. Set to `none` when no additional options are required. Enter the DHCP option number followed by a comma followed by the arguments of the DHCP option.

Multiple DHCP options may be specified, separated by a space.

Examples:

- 15, domain.example.com
- 119, search-domain.example.com, another-search.example.com
- 121, 10.0.32.0/24, 10.0.8.6, 10.0.33.0/24, 10.0.8.7
- 15, domain.example.com 119, search-domain.example.com

For a full list of available DHCP options see: <https://www.iana.org/assignments/bootp-dhcp-parameters/bootp-dhcp-parameters.xhtml>

Note: To be able to set DHCP Option 3 and 6 via this configuration item, you first need to disable the respective direct configuration item. Set 0.0.0.0 to `cfgDhcpScopeGateway` for option 3. Set 0.0.0.0 to `cfgDhcpScopeDnsServer1` for option 6.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 1024
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.13.3.1.14

9.0.1.7.1.13 cfgDhcpScopeAutoGateway

DHCP Scope Auto Gateway

When no default gateway is configured in `cfgDhcpScopeGateway` or `cfgDhcpScopeDhcpOptions`, then this options will automatically assign the IP address of this DHCP server as default gateway.

Applies to AP and STA.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.13.3.1.15

9.0.1.7.1.14 **cfgDhcpScopeAutoDns**

DHCP Scope Auto Dns

When no DNS servers are configured in `cfgDhcpScopeDnsServer1`, `cfgDhcpScopeDnsServer2` and `cfgDhcpScopeDhcpOptions`, then this options will automatically assign the IP address of this DHCP server as DNS server.

This option has no effect, when `cfgDhcpDnsmasqDnsEnabled` of the `dnsmasq` instance referencing this scope is disabled.

Applies to AP and STA.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.13.3.1.16

9.0.1.7.1.15 **cfgDhcpScopeld**

Scope ID

This is used in conjunction with the DHCP parameter `cfgDhcpDnsmasqScopeParameter`.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	0 - 8
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.13.3.1.2

9.0.1.7.1.16 **cfgDhcpScopeInterface**

DHCP Server Listening Interface

Network interface on which the DHCP server listen for DHCP requests. The interface on which the server runs must have an address configured. The DHCP server offers lease addresses based on the assigned address, the `cfgDhcpScopeStart` offset and the `cfgDhcpScopeLimit`. If an interface has multiple addresses, then the first address in the order specified in the `cfgNetIpTable` is used.

Examples:

- eth1
- br0.vlan0
- macvlan0

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.13.3.1.3

9.0.1.7.1.17 cfgDhcpScopeStart

DHCP Scope Start Offset

Specifies the first address of the scope as an offset from the network address and can be calculated as:

network address + cfgDhcpScopeStart

The network address is derived from the first configured IP address on the interface specified by cfgDhcpScopeInterface.

Examples:

- br0.vlan0 has 192.168.1.20/24. The network address of this CIDR block is 192.168.1.0. With cfgDhcpScopeStart set to 100, the lowest address which will be assigned is 192.168.1.100.
- br0.vlan99 has 172.29.101.7/23. The network address of this CIDR block is 172.29.100.0. With cfgDhcpScopeStart set to 306, the lowest address which will be assigned is 172.29.101.50.
- br0.vlan1000 has 10.0.8.140/26. The network address of this CIDR block is 10.0.8.128. With cfgDhcpScopeStart set to 22, the lowest address which will be assigned is 10.0.8.150.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.13.3.1.4

9.0.1.7.1.18 cfgDhcpScopeLimit

DHCP Scope Limit

Limits the number of addresses which are assigned to clients. Thus this parameter defines how many addresses are in the scope. The highest address assigned to a client can be calculated as:

$\text{network address} + \text{cfgDhcpScopeStart} + \text{cfgDhcpScopeLimit} - 1$

Examples:

- br0.vlan0 has 192.168.1.20/24. The network address of this CIDR block is 192.168.1.0. With `cfgDhcpScopeStart` set to 100 and `cfgDhcpScopeLimit` set to 150, the highest address which will be assigned is 192.168.1.249.
- br0.vlan99 has 172.29.101.7/23. The network address of this CIDR block is 172.29.100.0. With `cfgDhcpScopeStart` set to 306 and `cfgDhcpScopeLimit` set to 100, the highest address which will be assigned is 172.29.101.149.
- br0.vlan1000 has 10.0.8.140/26. The network address of this CIDR block is 10.0.8.128. With `cfgDhcpScopeStart` set to 22 and `cfgDhcpScopeLimit` set to 30 the highest address which will be assigned is 10.0.8.179.

Note: When `cfgDhcpScopeLimit` is set to a value greater than the remaining size of the CIDR block, the highest address is set to the last address in the block. Essentially the size of the scope is reduced until it fits the block.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.13.3.1.5

9.0.1.7.1.19 `cfgDhcpScopeLeasetime`

DHCP Scope Lease Time

Specifies the lease time of addresses handed out to clients.

Examples:

- 12h
- 30m
- 180s

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.13.3.1.6

9.0.1.7.1.20 **cfgDhcpScopeGateway**

DHCP Scope Default Gateway (DHCP Option 3)

Specifies the default gateway address handed out to clients. A value of 0.0.0.0 means not used. IPv4 only.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	IpAddress
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.13.3.1.7

9.0.1.7.1.21 **cfgDhcpScopeDnsServer1**

DHCP Scope Primary DNS Server (DHCP Option 6)

Specifies the primary DNS server address handed out to clients. A value of 0.0.0.0 means not used. IPv4 only.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	IpAddress
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.13.3.1.8

9.0.1.7.1.22 **cfgDhcpScopeDnsServer2**

DHCP Scope Secondary DNS Server (DHCP Option 6)

Specifies the secondary DNS server address handed out to clients. If the primary DNS server is not configured, this entry will also be ignored. A value of 0.0.0.0 means not used. IPv4 only.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	IpAddress
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.13.3.1.9

9.0.1.8 cfgNtp

9.0.1.8.1 cfgNtpEnabled

Synchronize the system time with given server.

Applies to AP and STA.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.14.1

9.0.1.8.2 cfgNtpServer1

NTP server 1

If the IP is set to 0.0.0.0 the NTP client will only listen to broadcast packages.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	IpAddress
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.14.2

9.0.1.8.3 cfgNtpServer2

NTP server 2

Used as fallback if server 1 cannot be reached.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	IpAddress
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.14.3

9.0.1.9 cfgHttp

9.0.1.9.1 cfgHttpUser

Web administrator username.

Applies to AP and STA.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.15.1

9.0.1.9.2 cfgHttpPassword

Web administrator password.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	5 - 126
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.15.2

9.0.1.9.3 cfgHttpEnabled

Configure if the webserver shall be started.

When disabling the webserver, make sure you still have another way to access the device, e.g by CLI or via SNMP.

Applies to AP and STA.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.15.3

9.0.1.9.4 cfgHttpRedirectEnabled

If enabled, all access to `cfgHttpIpAddress` shall be redirected to `cfgHttpHttpsAddress`. This does not disable http.

Applies to AP and STA.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.15.4

9.0.1.9.5 cfgHttpHttpAddress

Address to which the http server binds.

The default is '0.0.0.0:80'. Multiple space and/or comma separated tuples are allowed.

Examples:

- 192.168.1.20:80
- 192.168.1.20:80, 192.168.2.20:8080, 172.16.32.32:10080
- 192.168.1.20:80 192.168.2.20:8080 172.16.32.32:10080

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.15.5

9.0.1.9.6 cfgHttpHttpsAddress

Address to which the https server binds.

The default is '0.0.0.0:443'. Multiple space and/or comma separated tuples are allowed.

Examples:

- 192.168.1.20:443
- 192.168.1.20:443, 192.168.2.20:8443, 172.16.32.32:10443
- 192.168.1.20:443 192.168.2.20:8443 172.16.32.32:10443

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.15.6

9.0.1.10 cfgLldp

9.0.1.10.1 cfgLldpEnabled

Enable LLDP.

Applies to AP and STA.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.16.1

9.0.1.10.2 cfgLldpDescription

LLDP Description

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	0 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.16.2

9.0.1.11 cfgMdns

9.0.1.11.1 cfgMdnsEnabled

Enable mDNS.

Applies to AP and STA.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.17.1

9.0.1.11.2 cfgMdnsNetwork

Space Separated List of mDNS Aware Network Interfaces

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.17.2

9.0.1.12 cfgNetwork

9.0.1.12.1 cfgNetEthernetTable

Ethernet Network Interfaces

<i>Status</i>	current
<i>Range</i>	0 - 9
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.1

9.0.1.12.2 cfgNetEthTrunk

Eth Trunk

This entry is active when `cfgNetEthVlanMode` is set to **trunk(0)** or **nativeuntagged(3)**.

It specifies which 802.1q VLANs are accepted ingress and egress on the respective port. All unspecified VLANs are dropped. Set this entry to -1 to allow all VLANs. Untagged traffic is considered as VLAN 0.

The format of this entry is a space or comma separated list. To describe ranges the character '-' can be used.

Examples:

- '0,12,24,69'
- '7 56 127'
- '0, 84, 99, 2000'
- '0, 12-17, 3000-4000'
- '0-99 101-199 201-299, 301-4094'

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.1.1.10

9.0.1.12.3 cfgNetEthTag

Eth Tag

This entry is active when `cfgNetEthVlanMode` is set to **access(1)** or **nativeuntagged(3)**. It specifies which 802.1q VLAN should be used for untagged ingress and egress traffic. Set this entry to -1 to disable it.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	-1 - 4094
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.1.1.11

9.0.1.12.4 cfgNetEthVlanMode

Eth VLAN Mode

- **trunk(0)**: A trunk port carries packets on one or more specified VLANs specified in the `cfgNetEthTrunk` entry. A packet that ingresses on a trunk port is in the VLAN specified in its 802.1q header, or VLAN 0 if the packet has no 802.1q header (untagged frame). A packet that egresses through a trunk port will have an 802.1q header if it has a nonzero VLAN ID. Frames egressing on VLAN 0 have their tag stripped (egress untagged). Any packet that ingresses on a trunk port tagged with a VLAN that the port does not trunk is dropped.
- **access(1)**: An access port carries packets on exactly one VLAN specified in `cfgNetEthTag`. Packets egressing on an access port have no 802.1q header (egress untagged). Any packet with an 802.1q header with a nonzero VLAN ID that ingresses on an access port is dropped, regardless of whether the VLAN ID in the header is the access port's VLAN ID or not.
- **nativeuntagged(3)**: A native-untagged port resembles a trunk port, with the exception that a packet without an 802.1q header (ingress untagged) is automatically in the native VLAN specified in `cfgNetEthTag`. Frames egressing in the native VLAN are automatically untagged (egress untagged).

Applies to AP and STA.

<i>Enumeration</i>	trunk (0), access (1), nativeuntagged (3)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.1.1.12

9.0.1.12.5 cfgNetEthLldpEnabled

When `cfgLldpEnabled` is enabled, this parameter controls if the interface takes part in LLDP operation.

Applies to AP and STA.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.1.1.15

9.0.1.12.6 cfgNetEthMtu

The MTU of the Ethernet Interface

The minimum value is 68. The maximum value is 9000. The default value is -1, which means that the system default value (usually 1500) will be used.

When this interface is part of a bridge, the here configured MTU affects the MTU of the bridge. The bridge will have the smallest MTU of all its bridge members.

Example:

- br0 contains the interfaces eth0, eth1 and wlan0
- eth0 is set to 1400
- eth1 is set to 1200
- wlan0 is set to -1 (default 1500)
- This will result in an MTU of 1200 for br0

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	-1 - 9000
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.1.1.16

9.0.1.12.7 cfgNetEthName

Name of the ethernet interface.

Applies to AP and STA.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.1.1.2

9.0.1.12.8 cfgNetEthEnabled

Ethernet interface disabled or enabled.

Applies to AP and STA.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.1.1.3

9.0.1.12.9 cfgNetEthBridge

Ethernet Interface Bridge Membership

If value ≥ 0 then interface is part of bridge.

- -1: none
- 0: br0
- 1: br1
- X: brX

Bridges with an index ≥ 100 are special bridges which forward link local traffic. This can be used for wireless links in 4addr mode which should act as a cable-replacement.

Note: Such a bridge may only contain 2 interfaces!

Example:

- wlan0 and eth0 in br100, with eth1 as management interface

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.1.1.7

9.0.1.12.10 cfgNetEthAutoneg

Enable or Disable Auto Negotiation of the PHY

- **forced(0):** Forces the speed and duplex defined by `cfgNetEthSpeed`. Only 10Mbit and 100Mbit rates are allowed in forced mode. 1000Mbit requires the mode to be auto.
- **auto(1):** Advertises the supported auto negotiation defined by `cfgNetEthSpeed`.

Applies to AP and STA.

<i>Enumeration</i>	forced (0), auto (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.1.1.8

9.0.1.12.11 cfgNetEthSpeed

Defines a bitmask containing the possible speed/duplex combinations.

- 0x01 (1) = 10Mbit/Half
- 0x02 (2) = 10Mbit/Full
- 0x04 (4) = 100Mbit/Half
- 0x08 (8) = 100Mbit/Full
- 0x20 (32) = 1000Mbit/Full

When `cfgNetEthSpeed` is **forced(0)** only a single bit may be active. Only 10Mbit and 100Mbit rates are allowed in forced mode. 1000Mbit requires the mode to be auto.

Examples:

- **1:** Force 10Mbit half duplex
- **8:** Force 100Mbit full duplex

When `cfgNetEthSpeed` is **auto(1)** multiple bits may be set which are used to advertise the supported speed/duplex. 1000Mbit/Half is not supported.

Examples:

- **12:** Advertise 100 Mbit, half/full duplex (4 + 8)
- **15:** Advertise 10/100 Mbit, half/full duplex (1 + 2 + 4 + 8)
- **32:** Advertise only 1000 Mbit full duplex
- **47:** Advertise all speeds (1 + 2 + 4 + 8 + 32)

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.1.1.9

9.0.1.12.12 cfgNetWwanTable

Cellular Network Interfaces

Applies to cellular products only.

<i>Status</i>	current
<i>Range</i>	0 - 0
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.11

9.0.1.12.13 **cfgNetWwanMtu**

The MTU of the Cellular Network Interface

The default value is -1, which means that the system default value (usually 1500) will be used. This value may be provided by the service provider and may change depending on the provider.

It is not recommended to set a value, unless you know what you do.

Applies to cellular products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	-1 - 65535
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.11.1.16

9.0.1.12.14 **cfgNetWwanName**

Name of the Cellular Network Interface

The name of cellular network interfaces at system level is typically derived from the term Wireless Wide Area Network (WWAN), e.g. wwan0.

Applies to cellular products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.11.1.2

9.0.1.12.15 **cfgNetWwanEnabled**

Cellular Network Interface Disabled or Enabled

Applies to cellular products only.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.11.1.3

9.0.1.12.16 cfgNetOpenvpnTable

OpenVPN Interface Table

This table is in a one-to-one relation with the `cfgVpnOpenvpnTable`, where both indices match.

<i>Status</i>	current
<i>Range</i>	0 - 15
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.12

9.0.1.12.17 cfgNetOpenvpnTrunk

OpenVPN Interface Trunk

This entry is active when `cfgNetOpenvpnVlanMode` is set to **trunk(0)** or **nativeuntagged(3)**.

It specifies which 802.1q VLANs are accepted ingress and egress on the respective port. All unspecified VLANs are dropped. Set this entry to -1 to allow all VLANs. Untagged traffic is considered as VLAN 0.

The format of this entry is a space or comma separated list. To describe ranges the character '-' can be used.

Examples:

- '0,12,24,69'
- '7 56 127'
- '0, 84, 99, 2000'
- '0, 12-17, 3000-4000'
- '0-99 101-199 201-299, 301-4094'

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.12.1.10

9.0.1.12.18 cfgNetOpenvpnTag

OpenVPN Interface Tag

This entry is active when `cfgNetOpenvpnVlanMode` is set to: **access(1)** or **native-untagged(3)**. It specifies which 802.1q VLAN should be used for untagged ingress and egress traffic. Set this entry to -1 to disable it.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	-1 - 4094
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.12.1.11

9.0.1.12.19 **cfgNetOpenvpnVlanMode**

OpenVPN Interface VLAN Mode

This entry specifies how the port should behave:

- **trunk(0):** A trunk port carries packets on one or more specified VLANs specified in the `cfgNetOpenvpnTrunk` entry. A packet that ingresses on a trunk port is in the VLAN specified in its 802.1q header, or VLAN 0 if the packet has no 802.1q header (untagged frame). A packet that egresses through a trunk port will have an 802.1q header if it has a nonzero VLAN id. Frames egressing on VLAN 0 have their tag stripped (egress untagged). Any packet that ingresses on a trunk port tagged with a VLAN that the port does not trunk is dropped.
- **access(1):** An access port carries packets on exactly one VLAN specified in the `cfgNetOpenvpnTag`. Packets egressing on an access port have no 802.1q header (egress untagged). Any packet with an 802.1q header with a nonzero VLAN id that ingresses on an access port is dropped, regardless of whether the VLAN id in the header is the access port's VLAN id.
- **nativeuntagged(3):** A native-untagged port resembles a trunk port, with the exception that a packet without an 802.1q header (ingress untagged) is automatically in the native-vlan specified in `cfgNetOpenvpnTag`. Frames egressing in the native-vlan are automatically untagged (egress untagged).

Applies to AP and STA.

<i>Enumeration</i>	trunk (0), access (1), nativeuntagged (3)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.12.1.12

9.0.1.12.20 **cfgNetOpenvpnLldpEnabled**

LLDP Port Operation

When `cfgLldpEnabled` is enabled, this parameter controls if the interface takes part in LLDP operation.

Applies to AP and STA.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.12.1.15

9.0.1.12.21 **cfgNetOpenvpnMtu**

The MTU of the OpenVPN Interface

The minimum allowed valid value is 68. The maximum allowed valid value is 65535. The default value is -1, which does not change what is set by the system (usually 1500).

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	-1 - 65535
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.12.1.16

9.0.1.12.22 **cfgNetOpenvpnName**

Name of the OpenVPN Interface

Applies to AP and STA.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.12.1.2

9.0.1.12.23 **cfgNetOpenvpnEnabled**

OpenVPN Interface Disabled or Enabled

Applies to AP and STA.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.12.1.3

9.0.1.12.24 **cfgNetOpenvpnBridge**

OpenVPN Interface Bridge Membership

If value ≥ 0 then interface is part of bridge.

- -1: none
- 0: br0
- 1: br1
- X: brX

Bridges with an index ≥ 100 are special bridges which forward link local traffic. This can be used for tunnels which act as a cable-replacement.

Note: Such a bridge may only contain 2 interfaces!

Example:

- ovpn0 and eth0 in br100, with eth1 as management interface

Note: Interfaces may only be configured in a bridge when `cfgVpnOpenvpnDevType` is set to 1 (tap).

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.12.1.7

9.0.1.12.25 `cfgNetIpsecTable`

IPsec Network Interfaces.

<i>Status</i>	current
<i>Range</i>	0 - 15
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.13

9.0.1.12.26 `cfgNetIpsecMtu`

TECHPREVIEW: IPsec Tunnel MTU

When VTI is enabled, this allows to manually set the MTU of the IPsec tunnel.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	-1 - 65515
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.13.1.16

9.0.1.12.27 **cfgNetIpsecName**

TECHPREVIEW: Name of the IPsec interface

Applies to AP and STA.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.13.1.2

9.0.1.12.28 **cfgNetIpsecEnabled**

TECHPREVIEW: IPsec Interface Disabled or Enabled

Applies to AP and STA.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.13.1.3

9.0.1.12.29 **cfgNetFlowControllerTable**

FlowController Interfaces.

<i>Status</i>	current
<i>Range</i>	0 - 15
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.14

9.0.1.12.30 **cfgNetFcTrunk**

TECHPREVIEW: Flow Controller Trunk

This entry is active when `cfgNetFcVlanMode` is set to **trunk(0)** or **nativeuntagged(3)**.

It specifies which 802.1q VLANs are accepted ingress and egress on the respective port. All unspecified VLANs are dropped. Set this entry to -1 to allow all VLANs. Untagged traffic is considered as VLAN 0.

The format of this entry is a space or comma separated list. To describe ranges the character '-' can be used.

Examples:

- '0,12,24,69'

- '7 56 127'
- '0, 84, 99, 2000'
- '0, 12-17, 3000-4000'
- '0-99 101-199 201-299, 301-4094'

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.14.1.10

9.0.1.12.31 cfgNetFcTag

TECHPREVIEW: Flow Controller Tag

This entry is active when `cfgNetFcVlanMode` is set to: **access(1)** or **native-untagged(3)**.

It specifies which 802.1q VLAN should be used for untagged ingress and egress traffic. Set this entry to -1 to disable it.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	-1 - 4094
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.14.1.11

9.0.1.12.32 cfgNetFcVlanMode

TECHPREVIEW: Flow Controller VLAN Mode

- **trunk(0):** A trunk port carries packets on one or more specified VLANs specified in the `cfgNetFcTrunk` entry. A packet that ingresses on a trunk port is in the VLAN specified in its 802.1q header, or VLAN 0 if the packet has no 802.1q header (untagged frame). A packet that egresses through a trunk port will have an 802.1q header if it has a nonzero VLAN id. Frames egressing on VLAN 0 have their tag stripped (egress untagged). Any packet that ingresses on a trunk port tagged with a VLAN that the port does not trunk is dropped.
- **access(1):** An access port carries packets on exactly one VLAN specified in the `cfgNetFcTag`. Packets egressing on an access port have no 802.1q header (egress untagged). Any packet with an 802.1q header with a nonzero VLAN id that ingresses on an access port is dropped, regardless of whether the VLAN id in the header is the access port's VLAN id.
- **nativeuntagged(3):** A native-untagged port resembles a trunk port, with the exception that a packet without an 802.1q header (ingress untagged) is automatically in the native-vlan specified

in `cfgNetFcTag`. Frames egressing in the native-vlan are automatically untagged (egress untagged).

Applies to AP and STA.

<i>Enumeration</i>	trunk (0), access (1), nativeuntagged (3)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.14.1.12

9.0.1.12.33 `cfgNetFcLldpEnabled`

TECHPREVIEW: LLDP Disabled or Enabled

When `cfgLldpEnabled` is enabled, this parameter controls if the interface takes part in LLDP operation.

Applies to AP and STA.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.14.1.15

9.0.1.12.34 `cfgNetFcFlowMode`

TECHPREVIEW: Flow Controller Mode

- **normal(0):** Behaves like a normal bridge
- **fullduplex(1):** Transmits frames on the bridge-master, and receives frames on all other members of the bridge. The bridge-master is the first interface which is added to a bridge. The order in which interfaces are added is ascending in interface number: eth, wlan, ovpn, fc.

Applies to AP and STA.

<i>Enumeration</i>	normal (0), fullduplex (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.14.1.17

9.0.1.12.35 `cfgNetFcName`

TECHPREVIEW: Name of the Flow Controller Bridge

Applies to AP and STA.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.14.1.2

9.0.1.12.36 **cfgNetFcEnabled**

TECHPREVIEW: Flow Controller Bridge Disabled or Enabled

Applies to AP and STA.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.14.1.3

9.0.1.12.37 **cfgNetFcBridge**

TECHPREVIEW: Flow Controller Bridge Membership

If value ≥ 0 then interface is part of bridge.

- -1: none
- 0: br0
- 1: br1
- X: brX

Bridges with an index ≥ 100 are special bridges which forward link local traffic. This can be used for wireless links in 4addr mode which should act as a cable-replacement.

Note: Such a bridge may only contain 2 interfaces!

Example:

- wlan0 and eth0 in br100, with eth1 as management interface

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.14.1.7

9.0.1.12.38 cfgNetTunnelEndPointTable

Tunnel Endpoint Interfaces

<i>Status</i>	current
<i>Range</i>	0 - 15
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.15

9.0.1.12.39 cfgNetTepTrunk

Tunnel Endpoint Trunk

This entry is active when `cfgNetTepVlanMode` is set to **trunk(0)** or **nativeuntagged(3)**.

It specifies which 802.1q VLANs are accepted ingress and egress on the respective port. All unspecified VLANs are dropped. Set this entry to -1 to allow all VLANs. Untagged traffic is considered as VLAN 0.

The format of this entry is a space or comma separated list. To describe ranges the character '-' can be used.

Examples:

- '0,12,24,69'
- '7 56 127'
- '0, 84, 99, 2000'
- '0, 12-17, 3000-4000'
- '0-99 101-199 201-299, 301-4094'

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.15.1.10

9.0.1.12.40 cfgNetTepTag

Tunnel Endpoint Tag

This entry is active when `cfgNetTepVlanMode` is set to: **access(1):** or **nativeuntagged(3)**. It specifies which 802.1q VLAN should be used for untagged ingress and egress traffic. Set this entry to -1 to disable it.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	-1 - 4094
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.15.1.11

9.0.1.12.41 **cfgNetTepVlanMode**

Tunnel Endpoint VLAN Mode

- **trunk(0):** A trunk port carries packets on one or more specified VLANs specified in the `cfgNetTepTrunk` entry. A packet that ingresses on a trunk port is in the VLAN specified in its 802.1q header, or VLAN 0 if the packet has no 802.1q header (untagged frame). A packet that egresses through a trunk port will have an 802.1q header if it has a nonzero VLAN id. Frames egressing on VLAN 0 have their tag stripped (egress untagged). Any packet that ingresses on a trunk port tagged with a VLAN that the port does not trunk is dropped.
- **access(1):** An access port carries packets on exactly one VLAN specified in the `cfgNetTepTag`. Packets egressing on an access port have no 802.1q header (egress untagged). Any packet with an 802.1q header with a nonzero VLAN id that ingresses on an access port is dropped, regardless of whether the VLAN id in the header is the access port's VLAN id.
- **nativeuntagged(3):** A native-untagged port resembles a trunk port, with the exception that a packet without an 802.1q header (ingress untagged) is automatically in the native-vlan specified in `cfgNetTepTag`. Frames egressing in the native-vlan are automatically untagged (egress untagged).

Applies to AP and STA.

<i>Enumeration</i>	trunk (0), access (1), nativeuntagged (3)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.15.1.12

9.0.1.12.42 **cfgNetTepLldpEnabled**

Tunnel Endpoint LLDP Operation on Port

When `cfgLldpEnabled` is enabled, this parameter controls if the interface takes part in LLDP operation.

Applies to AP and STA.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.15.1.15

9.0.1.12.43 `cfgNetTepMtu`

MTU of the Tunnel Endpoint Interface

The minimum allowed valid value is 68. The maximum allowed valid value depends on the used tunnel type specified in `cfgNetTepTunnelType`. The default value is -1, which doesn't change what is set by the system. The default value as well depends on the tunnel type specified in `cfgNetTepTunnelType`.

Type	Usual Default	Maximum
gre	1476 (1500-20-4)	65504
gretap	1462 (1500-20-4-14)	65490

The outer IP header is 20 bytes. The GRE header is 4 bytes. For `gretap` there is an additional ethernet header of 14 bytes.

The default value is derived from the MTU of the interface over which the tunnel will transmit frames based on the routing table lookup based on the content of `cfgNetTepDestination`.

When this interface is part of a bridge, the here configured MTU affects the MTU of the bridge. The bridge will have the smallest MTU of all its bridge members.

Example:

- `br0` contains the interfaces `eth0`, `eth1` and `tep0`
- `eth0` is set to -1 (default 1500)
- `eth1` is set to 9000
- `tep0` is set to 1300
- This will result in an MTU of 1300 for `br0`

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	-1 - 65504
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.15.1.16

9.0.1.12.44 `cfgNetTepTunnelType`

Tunnel Endpoint Type

- **gre(0)**: Allows to tunnel L3 frames. Can not be bridged.
- **gretap(1)**: Allows to tunnel L2 frames. Can be bridged.

Applies to AP and STA.

<i>Enumeration</i>	gre (0), gretap (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.15.1.18

9.0.1.12.45 **cfgNetTepSource**

Source Address of the Tunnel

Can be set to 0.0.0.0 to let the system select the appropriate address based on the routing table.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	7 - 15
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.15.1.19

9.0.1.12.46 **cfgNetTepName**

Name of the Tunnel Endpoint

Applies to AP and STA.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.15.1.2

9.0.1.12.47 **cfgNetTepDestination**

Destination Address of the Tunnel

Encapsulated frames are sent to this address.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	7 - 15
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.15.1.20

9.0.1.12.48 `cfgNetTepEnabled`

Tunnel Endpoint Disabled or Enabled

Applies to AP and STA.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.15.1.3

9.0.1.12.49 `cfgNetTepBridge`

Tunnel Endpoint Interface Bridge Membership

If value ≥ 0 then interface is part of bridge.

- -1: none
- 0: br0
- 1: br1
- X: brX

Bridges with an index ≥ 100 are special bridges which forward link local traffic. This can be used for wireless links in 4addr mode which should act as a cable-replacement.

Note: Such a bridge may only contain 2 interfaces!

Example:

- `tep0` and `wlan0` in `br100`, with `eth0/1` as management interface

Not all types of tunnels configured in `cfgNetTepTunnelType` are able to be bridged. Currently supported types to be bridged are:

- `gretap(1)`:

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.15.1.7

9.0.1.12.50 `cfgNetWlanTable`

WLAN Network Interfaces

<i>Status</i>	current
<i>Range</i>	0 - 63
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.2

9.0.1.12.51 cfgNetWlanTrunk

Wlan Trunk

This entry is active when `cfgNetWlanVlanMode` is set to **trunk(0)** or **nativeuntagged(3)**.

It specifies which 802.1q VLANs are accepted ingress and egress on the respective port. All unspecified VLANs are dropped. Set this entry to -1 to allow all VLANs. Untagged traffic is considered as VLAN 0.

The format of this entry is a space or comma separated list. To describe ranges the character '-' can be used.

Examples:

- '0,12,24,69'
- '7 56 127'
- '0, 84, 99, 2000'
- '0, 12-17, 3000-4000'
- '0-99 101-199 201-299, 301-4094'

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.2.1.10

9.0.1.12.52 cfgNetWlanTag

Wlan Tag

This entry is active when `cfgNetWlanVlanMode` is set to **access(1)** or **nativeuntagged(3)**. It specifies which 802.1q VLAN should be used for untagged ingress and egress traffic. Set this entry to -1 to disable it.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	-1 - 4094
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.2.1.11

9.0.1.12.53 **cfgNetWlanVlanMode**

WLAN VLAN Mode

- **trunk(0):** A trunk port carries packets on one or more specified VLANs specified in the `cfgNetWlanTrunk` entry. A packet that ingresses on a trunk port is in the VLAN specified in its 802.1q header, or VLAN 0 if the packet has no 802.1q header (untagged frame). A packet that egresses through a trunk port will have an 802.1q header if it has a nonzero VLAN ID. Frames egressing on VLAN 0 have their tag stripped (egress untagged). Any packet that ingresses on a trunk port tagged with a VLAN that the port does not trunk is dropped.
- **access(1):** An access port carries packets on exactly one VLAN specified in the `cfgNetWlanTag`. Packets egressing on an access port have no 802.1q header (egress untagged). Any packet with an 802.1q header with a nonzero VLAN ID that ingresses on an access port is dropped, regardless of whether the VLAN ID in the header is the access port's VLAN ID.
- **nativeuntagged(3):** A native-untagged port resembles a trunk port, with the exception that a packet without an 802.1q header (ingress untagged) is automatically in the native VLAN specified in `cfgNetWlanTag`. Frames egressing in the native VLAN are automatically untagged (egress untagged).

Applies to AP and STA.

<i>Enumeration</i>	trunk (0), access (1), nativeuntagged (3)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.2.1.12

9.0.1.12.54 **cfgNetWlanLldpEnabled**

When `cfgLldpEnabled` is enabled, this parameter controls if the interface takes part in LLDP operation.

Applies to AP and STA.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.2.1.15

9.0.1.12.55 **cfgNetWlanMtu**

The MTU of the Wireless Interface

The minimum value is 256. The maximum value is 2304. The default value is -1, which means that the system default value (usually 1500) will be used.

When this interface is part of a bridge, the here configured MTU affects the MTU of the bridge. The bridge will have the smallest MTU of all its bridge members.

Example:

- br0 contains the interfaces eth0, eth1 and wlan0
- eth0 is set to -1 (default 1500)
- eth1 is set to 9000
- wlan0 is set to 2000
- This will result in an MTU of 1500 for br0

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	-1 - 2304
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.2.1.16

9.0.1.12.56 **cfgNetWlanName**

Name of the wireless interface

Applies to AP and STA.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.2.1.2

9.0.1.12.57 **cfgNetWlanEnabled**

Wireless interface disabled or enabled

Applies to AP and STA.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.2.1.3

9.0.1.12.58 cfgNetWlanBridge

Wireless Interface Bridge Membership

If value ≥ 0 then interface is part of bridge.

- -1: none
- 0: br0
- 1: br1
- X: brX

Bridges with an index ≥ 100 are special bridges which forward link local traffic. This can be used for wireless links in 4addr mode which should act as a cable-replacement.

Note: Such a bridge may only contain 2 interfaces!

Example:

- wlan0 and eth0 in br100, with eth1 as management interface

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.2.1.7

9.0.1.12.59 cfgNetVlanTable

VLAN Network Interfaces

<i>Status</i>	current
<i>Range</i>	0 - 127
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.3

9.0.1.12.60 cfgNetVlanMtu

The MTU of the VLAN Interface

The minimum value is 68. The maximum value is 9000. The default value is -1, which means that the MTU is inherited from the bridge MTU.

VLAN interfaces are internal ports of a bridge. Based on the MTU of the bridge, these internal ports inherit the MTU of the bridge. The Bridge MTU is the minimum MTU of its member-interfaces. VLAN interfaces are not considered member-interfaces when deriving the bridge MTU. When setting the VLAN MTU, it is not possible to set a higher MTU, than the value inherited from the bridge. Setting

a value higher than the bridge MTU, will not result in an error, but instead set the MTU to the value inherited from the bridge.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	-1 - 9000
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.3.1.16

9.0.1.12.61 **cfgNetVlanName**

Name of the VLAN interface.

Applies to AP and STA.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.3.1.2

9.0.1.12.62 **cfgNetVlanEnabled**

VLAN interface disabled or enabled.

Applies to AP and STA.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.3.1.3

9.0.1.12.63 **cfgNetVlanBridge**

VLAN Interface Bridge Membership

If value ≥ 0 then interface is part of bridge.

- -1: none
- 0: br0
- 1: br1
- X: brX

VLAN interfaces are always of type access.

When set to -1, the VLAN interface will be created on the parent interface defined by `cfgNetVlanParent`.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.3.1.7

9.0.1.12.64 `cfgNetVlanParent`

VLAN Parent

Name of the physical parent interface on which the VLAN resides.

This entry is only active when the VLAN interface is not part of a bridge (`cfgNetVlanBridge = -1`).

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.3.1.8

9.0.1.12.65 `cfgNetVlanVid`

VLAN ID

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	0 - 4094
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.3.1.9

9.0.1.12.66 `cfgNetIpTable`

IP address configuration.

<i>Status</i>	current
<i>Range</i>	0 - 127
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.6

9.0.1.12.67 **cfgNetIpCarpId**

Id which references the CARP-instance by `cfgNetCarpIndex`.

This parameter is only active when `cfgNetIpProto` is set to **carp(5)**. Indicates that the referenced CARP instance processes this IP.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	0 - 15
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.6.1.10

9.0.1.12.68 **cfgNetIpEnabled**

IP disabled or enabled.

Applies to AP and STA.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.6.1.3

9.0.1.12.69 **cfgNetIpAddr**

The IPv4 address (using CIDR notation) of the interface specified in `cfgNetIpInterface`.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	5 - 50
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.6.1.4

9.0.1.12.70 **cfgNetIpProto**

This parameter defines which protocol is used to get the IPv4 settings for this interface.

- **static(0)**: Indicates that the address is manually configured to a specified address given by the IPv4 address parameter of this interface configuration.
- **dhcp(1)**: Indicates that an IPv4 address will be obtained by the DHCP client. In case the DHCP client is unable to get a valid IPv4 address the static IP address will be used as a fallback.
- **linkLocal(4)**: Indicates that an IPv4 link local address (an address in the range of 169.254.0.1 to 169.254.255.254, randomly chosen by the system) will be used on this interface. `cfgNetIpAddr`

is then not used for the interface. **Note:** Only one interface of the device can use a linkLocal protocol.

- **carp(5):** Indicates that an IPv4 address will be set by the CARP instance specified in `cfgNetIpCarpId`, when the CARP instance has the state MASTER.

Note: Only one interface of the device can use a link local protocol.

For wireless interfaces, the following additional modes are available:

- **dhcpForceRenew(2):** Indicates that an IPv4 address will be obtained by the DHCP client. In case the DHCP client is unable to get a valid IPv4 address the static IP address will be used as a fallback. On a STA this mode will perform a DHCP RENEW after every connection to an AP. This is useful if the device is roaming between different DHCP servers.
- **dhcpForceRelease(3):** Indicates that an IPv4 address will be obtained by the DHCP client. In case the DHCP client is unable to get a valid IPv4 address, the static IP address will be used as a fallback. On a STA this mode will perform a DHCP RELEASE followed by a DHCP DISCOVER after every connection to an AP. This is useful if the device is roaming between different DHCP servers which don't send NAK to an unknown device sending RENEW.

Applies to AP and STA.

<i>Enumeration</i>	static (0), dhcp (1), dhcpForceRenew (2), dhcpForceRelease (3), linkLocal (4), carp (5)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.6.1.6

9.0.1.12.71 `cfgNetIpInterface`

Name of the interface on which the IP resides.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.6.1.8

9.0.1.12.72 `cfgNetCarpTable`

Redundant IP Addresses with the Common Address Redundancy Protocol (CARP)

<i>Status</i>	current
<i>Range</i>	0 - 15
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.7

9.0.1.12.73 **cfgNetCarpVhid**

The Virtual Host ID

This is a unique number that is used to identify the redundancy group to other nodes in the group, and to distinguish between groups on the same network.

This must be the same on all members of the group.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.7.1.10

9.0.1.12.74 **cfgNetCarpPassword**

The CARP authentication password

This is the password which is used to encrypt the CARP frames when talking to other CARP-enabled hosts in this redundancy group.

This must be the same on all members of the group.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.7.1.11

9.0.1.12.75 **cfgNetCarpAdvbase**

Advertisement base in seconds

This parameter specifies how often to transmit advertisement frames that we're a member of the redundancy group.

This time is divided with `cfgNetCarpAdvdivider` to decrease the failovertime of the redundancy group.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.7.1.12

9.0.1.12.76 **cfgNetCarpAdvskew**

Advertisement base in 1/255th of a second

This parameter specifies how much to skew the advbase when sending CARP advertisements.

By manipulating advskew, the master of a CARP group can be chosen. The higher the number, the less often frames are transmitted and the less preferred the host will be when determining the master.

This time is divided with `cfgNetCarpAdvdivider` to decrease the failovertime of the redundancy group.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	0 - 254
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.7.1.13

9.0.1.12.77 **cfgNetCarpAdvdivider**

Divider for `cfgNetCarpAdvbase` and `cfgNetCarpAdvskew`

This parameter specifies how much the Advbase and Advskew are divided to speed the algorithm up. With a factor of 1, the Advbase and Advskew are unchanged.

Increasing the Advdivider to 10 speeds the algorithm up to allow a minimum advertise time of 100ms instead of 1s.

This must be the same on all members of the group.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	1 - 100
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.7.1.14

9.0.1.12.78 **cfgNetCarpRatio**

The dead ratio after which an existing master is considered dead

A slave device will wait:

$cfgNetCarpRatio * (cfgNetCarpAdvbase + cfgNetCarpAdvskew)$

before considering the current master as dead and attempt to become the new master. Since The locally configured values for Advbase and Advskew are used, if there is a better suited master (lower Advbase/Advskew), it will start advertising before the local device.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	1 - 100
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.7.1.15

9.0.1.12.79 **cfgNetCarpPreempt**

Preempt other masters when the local device is better

Allow hosts within a redundancy group that have a better Advbase and Advskew to preempt the current master.

Applies to AP and STA.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.7.1.16

9.0.1.12.80 **cfgNetCarpPreemptdemote**

Preempt other master when they demote themselves

Allow hosts within a redundancy group that have a better Advbase and Advskew to preempt the current master when it demotes itself (skew >= 240).

Preemptdemote only works when `cfgNetCarpPreempt` is used.

Preemptdemote also controls if a master demotes itself when another CARP-instance in the same `cfgNetCarpLocalInterfaceGroup` goes down or is demoted.

Applies to AP and STA.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.7.1.17

9.0.1.12.81 **cfgNetCarpLocalInterfaceGroup**

The local carp interface group the carp interface belongs to.

When `cfgNetCarpPreempt` or `cfgNetCarpPreemptdemote` is set to enabled, the interface will demote itself (skew = 240) when another interface within the same local group goes down or is demoted.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	0 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.7.1.18

9.0.1.12.82 **cfgNetCarpSyncInterface**

Name of the CARP control interface.

This interface is used to transmit and receive CARP advertisements.

This interface is required to have its own unique IP address.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.7.1.19

9.0.1.12.83 **cfgNetCarpMcastIp**

Multicast address on which CARP advertisements are transmitted.

The default multicast address for CARP is 224.0.0.18.

Use this parameter to specify a custom multicast destination.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	IpAddress
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.7.1.21

9.0.1.12.84 **cfgNetCarpEnabled**

CARP interface disabled or enabled

Applies to AP and STA.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.7.1.3

9.0.1.12.85 **cfgNetMacVlanTable**

MACVLAN Network Interfaces

<i>Status</i>	current
<i>Range</i>	0 - 63
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.9

9.0.1.12.86 **cfgNetMacVlanMac**

MAC address of the MACVLAN

Set 00:00:00:00:00:00 to use a random MAC address.

Format: 00:14:5a:02:10:42

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	17 - 17
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.9.1.15

9.0.1.12.87 **cfgNetMacVlanMtu**

The MTU of the MACVLAN Interface

The minimum value is 68. The maximum value is 9000. The default value is -1, which means that the MTU is inherited from the parent interface.

MACVLAN interfaces are virtual interfaces which reside on another interface as parent. Based on the MTU of the parent, these virtual interfaces inherit the MTU of the parent. Setting the MACVLAN MTU to a value higher than the MTU inherited from the parent, will result in an error.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	-1 - 9000
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.9.1.16

9.0.1.12.88 **cfgNetMacVlanName**

Name of the MACVLAN interface.

Applies to AP and STA.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.9.1.2

9.0.1.12.89 **cfgNetMacVlanEnabled**

MACVLAN interface disabled or enabled.

Applies to AP and STA.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.9.1.3

9.0.1.12.90 **cfgNetMacVlanParent**

Name of the parent interface on which the MACVLAN resides.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.2.9.1.8

9.0.1.13 cfgWireless

9.0.1.13.1 cfgWlanDeviceTable

Wireless Hardware Modules

<i>Status</i>	current
<i>Range</i>	0 - 1
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.1

9.0.1.13.2 cfgWlanDevDistance

Maximum distance in meters a client can be apart from the access point. Even though the distance is set in meters, the slot time settings change in 450m steps.

Applies to AP and STA. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	0 - 114750
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.1.1.10

9.0.1.13.3 cfgWlanDevRts

RTS/CTS Threshold

Frames equal or longer in bytes than this value require a RTS/CTS handshake.

RTS/CTS is used in hidden node situations. In 11bg and b mode, these frames are sent in DSSS modulation at 11b data rates. Otherwise (pure-g and a) OFDM rates are used.

The following settings are special:

- **-1** disable value, RTS/CTS is disabled.
- **0** minimum value, RTS/CTS is always used.
- **2346** maximum value legacy-rates, RTS/CTS is enabled for maximum sized frames.
- **65535** maximum value n-rates, RTS/CTS is enabled for maximal aggregate sized frames.

Note: It is not recommended to use RTS/CTS in AP mode.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	-1 - 65535
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.1.1.11

9.0.1.13.4 cfgWlanDevFragments

Fragmentation Threshold

Frames longer in bytes than this threshold will be fragmented.

Fragmentation can be used to reduce the number of retransmissions. The following settings are special

- **-1** disable value, fragmentation is disabled
- **256** minimum value, frames above 256 are fragmented.
- **2346** maximum value, essentially the same as disabled.

Applies to AP and STA. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	-1 - 2346
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.1.1.12

9.0.1.13.5 cfgWlanDevShortRetry

Number of times the transmission of the RTS frame will be retried if there is no CTS received from the AP.

Applies to AP and STA. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	1 - 10
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.1.1.13

9.0.1.13.6 cfgWlanDevLongRetry

Number of times the unicast data frames will be retried if there is no ACK from the receiver.

Applies to AP and STA. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	1 - 10
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.1.1.14

9.0.1.13.7 cfgWlanDevAntennaGain

Antenna gain in dBi.

If multiple antennas with different gains are connected, the value of the antenna with the highest gain shall be configured.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.1.1.15

9.0.1.13.8 cfgWlanDevTxAntenna

Number of Wireless Transmitter Antenna Ports

This is a bitmask to enable/disable the chains.

Examples:

- 1(0001) = A1 (chain 0) enabled
- 3(0011) = A1 and A2 (chain 0 and 1) enabled
- 7(0111) = A1, A2 and A3 (chain 0, 1 and 2) enabled
- 15(1111) = A1, A2, A3 and A4 (chain 0, 1, 2 and 3) enabled

Note: Number of available antennas depends on the product. Please check the data-sheet of your product.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.1.1.16

9.0.1.13.9 cfgWlanDevRxAntenna

Number of Wireless Receiver Antenna Ports

This is a bitmask to enable/disable the chains.

Examples:

- 1(0001) = A1 (chain 0) enabled
- 3(0011) = A1 and A2 chain 0 and 1) enabled
- 7(0111) = A1, A2 and A3 (chain 0, 1 and 2) enabled
- 15(1111) = A1, A2, A3 and A4 (chain 0, 1, 2 and 3) enabled

Note: Number of available antennas depends on the product. Please check the data-sheet of your product.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.1.1.17

9.0.1.13.10 **cfgWlanDevPhy**

The map between physical device and radio.

Applies to AP and STA.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.1.1.18

9.0.1.13.11 **cfgWlanDevName**

Name of the Wireless Device.

Applies to AP and STA.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.1.1.2

9.0.1.13.12 **cfgWlanDevHtCapabilities**

HT capability flags:

- [LDPC] = 1 Enable support for LDPC coding
- [SHORT-GI-20] = 32 Allow short GI for 20 MHz
- [SHORT-GI-40] = 64 Allow short GI for 40 MHz
- [TX-STBC] = 128 Enable support for TX-STBC
- [RX-STBC1] = 256 Enable support for RX-STBC1
- [DSSS_CCK-40] = 4096 Enable support for DSSS/CCK Mode in 40 MHz
- [40-INTOLERANT] = 16384 Advertise 40 MHz intolerance

Applies to AP. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	0 - 65535
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.1.1.24

9.0.1.13.13 **cfgWlanDevQmrrString**

A list of rate controller quadruples per queue.

Each quadruple consist of (mcs-rate [0-31], tries [0-15], rts_cts [0-1], sgi [0-1]) with 4 entries per queue.

16 values together are for a single queue.

The values are in the form: rate1, try1, rts_cts1, sgi1, rate2, . . . , rate4, try4, rts_cts4, sgi4.

The order of the queues is VO, VI, BE, BK.

QMRR override for a specific queue is disabled when its respective try1 value is 0.

When QMRR override is disabled, the normal minstrel or other configured overrides, are used.

Frames in the VO queue are never aggregated.

All characters other than numbers are ignored

Example:

```
[(7 1 0 0) (4 2 0 0) (2 3 0 0) (0 4 1 0)] [(7 1 0 0) (4 1 0 1) (2 1 0 1) (0 1 1 1)] [(7 1 0 0) (4 1 0 0) (0 0 0 0) (0 0 0 0)] [(7 1 0 0) (0 0 0 0) (0 0 0 0) (0 0 0 0)]
```

Applies to AP and STA. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.1.1.26

9.0.1.13.14 **cfgWlanDevModulation**

Physical Wireless Device Modulation Mode

The following modulation modes are configurable:

- **g(2)**: This modulation mode uses OFDM data rates up to 54 MBit/s in the frequency band between 2.4 and 2.4835 GHz. It supports the 802.11g standard.
- **bg(3)**: This modulation mode uses data rates up to 54 MBit/s in the frequency band between 2.4 and 2.4835 GHz. It supports the 802.11bg standard. The modulation is either DSSS for the slower rates or OFDM for the faster ones.
- **a(4)**: Mode supports data rates up to 54 MBit/s in the 5GHz frequency band and only OFDM modulation.
- **n(8)**: Mode supports data rates up to 300 MBit/s in the 2.4GHz and 5GHz frequency band and only OFDM modulation. Mode **n(8)** cannot be used as such. It has to be combined with g or a to specify which frequency band shall be used.
- **ng(10)**: For 2.4GHz
- **na(12)**: For 5GHz.
- **ac(28)**: 11ac mode for 5GHz.

Note: Some products do not support all modulations. Please check the data-sheet of your product.

Applies to AP and STA.

<i>Enumeration</i>	g (2), bg (3), a (4), ng (10), na (12), ac (28)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.1.1.4

9.0.1.13.15 cfgWlanDevBandwidth

Wireless Bandwidth Mode specifies the Bandwidth of the Channel.

- **bw20(0)**: for 20MHz wide channel.
- **bw40Plus(1)**: for 40MHz wide channel with the side channel on the top.
- **bw40Minus(2)**: for 40MHz wide channel with the side channel on the bottom.
- **bwQuarter(3)**: for 5MHz wide channel (quarter rate).
- **bwHalf(4)**: for 10MHz wide channel (half rate).
- **bw80(5)**: 80MHz wide channel (only in 11ac mode)
- **bw160(6)**: 160MHz wide channel (only in 11ac mode)
- **bw8080(7)**: 80+80MHz wide channel (only in 11ac mode)
- **bwAuto(8)**: automatic channel width (only in 11ac mode)

Note: **bw40Plus(1)** and **bw40Minus(2)** may not be usable on all channels.

Examples:

The following table shows examples of which channels may be used. The full list can be found in IEEE 802.11n Annex J. Depending on the country, not all frequencies may be available.

Band	bw40Plus(1)	bw40Minus(2)
2.4 GHz	2412 to 2452	2432 to 2472

Band	bw40Plus(1)	bw40Minus(2)
5 GHz	5180, 5220, 5260, etc.	5200, 5240, 5280, etc.

Note: Some products do not support all bandwidths. Please check the data-sheet of your product.

Applies to AP and STA.

<i>Enumeration</i>	bw20 (0), bw40Plus (1), bw40Minus (2), bwQuarter (3), bwHalf (4), bw80 (5), bw160 (6), bw8080 (7), bwAuto (8)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.1.1.5

9.0.1.13.16 **cfgWlanDevFrequency**

Wireless Frequency in MHz.

In AP mode, setting the wireless frequency to zero(0) enables the automatic channel selection (ACS) feature. This forces the AP to choose the best channel for operation.

In STA mode, the `cfgWlanIfaceScanList` is used to configure which frequencies are to be scanned.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.1.1.6

9.0.1.13.17 **cfgWlanDevPower**

TX Power Limit (EIRP)

Max. limit for wireless output power as effective isotropic radiated power (EIRP) in dBm including array gain and antenna gain.

Note: This parameter only limits the maximum output power. The effective output power might be lower (regulatory limits, rate depended limits).

EIRP (dBm) = antenna port power (dBm) + array gain (dB) + antenna gain (dBi)

- The antenna port power in dBm defines the power transmitted per antenna port (chain).

- The array gain in dB defines the gain which is achieved by the use of multiple antenna ports (chains). The number of active antenna ports (chains) is defined by `cfgWlanDevTxAntenna`. The array gain depends on number of active antenna ports (chains) as following:
 - One antenna port (chain) = 0 dB
 - Two antenna ports (chains) = 3 dB
 - Three antenna ports (chains) = 5 dB
 - Four antenna ports (chains) = 6 dB
- The antenna gain in dBi defines the gain which is achieved by the antenna. The antenna gain is configured by `cfgWlanDevAntennaGain`.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	6 - 50
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.1.1.8

9.0.1.13.18 `cfgWlan802dot1xTable`

Wireless 802dot1x

<i>Status</i>	current
<i>Range</i>	0 - 63
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.10

9.0.1.13.19 `cfgWlan802dot1xIdentity`

The identity string for EAP

Applies to STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.10.1.10

9.0.1.13.20 `cfgWlan802dot1xClientKeyPassword`

The password to unlock the private key

This is only required if the key is encrypted.

Applies to STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.10.1.17

9.0.1.13.21 **cfgWlan802dot1xTlsControlParams**

Bitfield to control TLS behaviour

- **0x1** ignore certificate validity time
- **0x2** ignore ca certificate

Applies to STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	0 - 3
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.10.1.18

9.0.1.13.22 **cfgWlan802dot1xName**

Name of the virtual wireless interface

Applies to AP and STA.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.10.1.2

9.0.1.13.23 **cfgWlan802dot1xCiphers**

OpenSSL Cipher String for the client

This is an OpenSSL specific configuration option for configuring the default cipher.

Please read the documentation for a list of all available ciphers and used syntax.

Used only if `cfgWlanIfaceEncryption = eap(6)`.

Examples:

- ECDHE-RSA-AES128-GCM-SHA256
- DHE-RSA-AES128-GCM-SHA256

- ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-SHA384

Applies to STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.10.1.21

9.0.1.13.24 **cfgWlan802dot1xOwnIpAddr**

The own IP address of the authenticator

This field is used as NAS-IP-Address RADIUS attribute. Set this to the IP address with which the authenticator will communicate with the RADIUS server.

Applies to AP.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	IpAddress
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.10.1.3

9.0.1.13.25 **cfgWlan802dot1xAuthServerParameter**

Reference ID to the radius auth server table

Uses all parameters in the `cfgWlan802dot1xAuthServerTable` which have as `cfgWlan802dot1xAuthSrvId` the value set here.

Applies to AP.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.10.1.4

9.0.1.13.26 **cfgWlan802dot1xAcctServerParameter**

Reference ID to the radius acct server table

Uses all parameters in the `cfgWlan802dot1xAcctServerTable` which have as `cfgWlan802dot1xAcctSrvId` the value set here.

Applies to AP.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.10.1.5

9.0.1.13.27 **cfgWlan802dot1xRetryPrimaryInterval**

Retry interval to return to the primary RADIUS server in seconds

RADIUS client code will automatically try to use the next server when the current server is not replying to requests. If this interval is set, primary server will be retried after configured amount of time even if the currently used secondary server is still working. Set to 0 to disable.

Applies to AP.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	0 - 86400
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.10.1.6

9.0.1.13.28 **cfgWlan802dot1xInterimAccountingInterval**

Interim accounting update interval in seconds

If this is set (larger than 0) and acct_server is configured, hostapd will send interim accounting updates every N seconds. Set to 0 to disable.

Note: If set, this overrides possible Acct-Interim-Interval attribute in Access-Accept message. Thus, this value should not be configured in hostapd.conf if RADIUS server is used to control the interim interval. This value should not be less 600 (10 minutes) and must not be less than 60 (1 minute).

Applies to AP.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	0 - 86400
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.10.1.7

9.0.1.13.29 **cfgWlan802dot1xNasId**

Optional NAS Identifier string for RADIUS messages

When used, this should be unique to the NAS within the scope of the RADIUS server. For example, a fully qualified domain name can be used here. When using IEEE 802.11r, nas_identifier must be set and must be between 1 and 48 octets long.

Applies to AP.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 48
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.10.1.8

9.0.1.13.30 **cfgWlan802dot1xEapType**

Specify the EAP type

Applies to STA.

<i>Enumeration</i>	tls (0), peap (1), ttls (2)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.10.1.9

9.0.1.13.31 **cfgWlan802dot1xAuthServerTable**

Wireless 802dot1x AuthServer

<i>Status</i>	current
<i>Range</i>	0 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.11

9.0.1.13.32 **cfgWlan802dot1xAuthSrvEnabled**

Enable this entry in the auth server list

Applies to AP.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.11.1.2

9.0.1.13.33 **cfgWlan802dot1xAuthSrvId**

ID of the authorisation server table

The configuration item `cfgWlan802dot1xAuthServerParameter` references to this ID.

Applies to AP.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.11.1.3

9.0.1.13.34 **cfgWlan802dot1xAuthSrvIpAddr**

IP of the RADIUS server against which will be authenticated

Applies to AP.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	IpAddress
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.11.1.4

9.0.1.13.35 **cfgWlan802dot1xAuthSrvPort**

Port of the RADIUS server against which will be authenticated

Applies to AP.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.11.1.5

9.0.1.13.36 **cfgWlan802dot1xAuthSrvSharedSecret**

Password to connect to the specified RADIUS server

Applies to AP.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.11.1.6

9.0.1.13.37 **cfgWlan802dot1xAcctServerTable**

Wireless 802dot1x AcctServer

<i>Status</i>	current
<i>Range</i>	0 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.12

9.0.1.13.38 `cfgWlan802dot1xAcctSrvEnabled`

Enable this entry in the acct server list.

Applies to AP.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.12.1.2

9.0.1.13.39 `cfgWlan802dot1xAcctSrvId`

ID of the accounting server table

The configuration item `cfgWlan802dot1xAcctServerParameter` refers to this ID.

Applies to AP.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.12.1.3

9.0.1.13.40 `cfgWlan802dot1xAcctSrvIpAddr`

IP of the RADIUS accounting server Set to 0.0.0.0 to disable.

Applies to AP.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	IpAddress
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.12.1.4

9.0.1.13.41 `cfgWlan802dot1xAcctSrvPort`

Port of the RADIUS accounting server

Applies to AP.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.12.1.5

9.0.1.13.42 `cfgWlan802dot1xAcctSrvSharedSecret`

Password to connect to the specified RADIUS accounting server

Applies to AP.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.12.1.6

9.0.1.13.43 `cfgWlan802dot11rTable`

Wireless 802dot11r

<i>Status</i>	current
<i>Range</i>	0 - 63
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.13

9.0.1.13.44 `cfgWlan802dot11rR0KHParameter`

Reference ID to the R0KH parameter table

Uses all parameters in the `cfgWlan802dot11rR0KHTable` which have as `cfgWlan802dot11rR0KHTblId` the value set here.

Applies to AP. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.13.1.10

9.0.1.13.45 `cfgWlan802dot11rR1KHParameter`

Reference ID to the R1KH parameter table

Uses all parameters in the `cfgWlan802dot11rR1KHTable` which have as `cfgWlan802dot11rR1KHTblId` the value set here.

Applies to AP. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.13.1.11

9.0.1.13.46 `cfgWlan802dot11rExpirationEnabled`

Enable/disable PMK-R0/-R1 expiration forcing

If set to **enabled(1)**, AP forces PMK-R0s and PMK-R1s expiration once a day (see `cfgWlan802dot11rExpiration`

Applies to AP. 802.11n products only.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.13.1.12

9.0.1.13.47 `cfgWlan802dot11rExpirationTime`

Daily PMK-R0/-R1 expiration time

Define time (hour:minute) at which PMK-R0s and PMK-R1s expiration is daily forced (if `cfgWlan802dot11rExpiration` is **enabled(1)**).

The time is referenced to the local time as define in `cfgSysTimezone`

Examples:

- 00:00 - force expiration each day at midnight
- 01:00 - force expiration each day at 01:00
- 23:05 - force expiration each day at 23:05

Applies to AP. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	5 - 5
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.13.1.13

9.0.1.13.48 `cfgWlan802dot11rVlan`

The 802.1q VLAN tag of 802.11r backbone traffic

Defines the VLAN tag with which all 802.11r management frames (ethertype 0x88b7) are transmitted on the backbone.

Setting a value of 0 disables the VLAN header and transmits the frames untagged.

The priority of all tagged frames is set to 0x7.

When this value is set to a value other than 0, make sure that the VLAN mode of the wireless interface `cfgNetWlanVlanMode` allows tagged frames. Also ensure that `cfgNetWlanTrunk` includes all VLANs or this specific VLAN.

Applies to AP. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	0 - 4094
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.13.1.14

9.0.1.13.49 `cfgWlan802dot11rName`

Name of the virtual wireless interface

Applies to AP and STA. 802.11n products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.13.1.2

9.0.1.13.50 `cfgWlan802dot11rEnabled`

Enable usage of 802.11r on this device

Applies to AP and STA. 802.11n products only.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.13.1.3

9.0.1.13.51 `cfgWlan802dot11rMobilityDomain`

Mobility Domain identifier (`dot11FTMobilityDomainID`, MDID)

MDID 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. 2-octet identifier as a hex string.

Examples:

- a1b2
- faba
- 5678

Applies to AP. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	4 - 4
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.13.1.4

9.0.1.13.52 **cfgWlan802dot11rPmkR0KeyHolderIdentifier**

PMK-R0 Key Holder identifier (dot11FTR0KeyHolderID)

Configure this in the field `cfgWlan802dot1xNasId`.

Applies to AP. 802.11n products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 48
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.13.1.5

9.0.1.13.53 **cfgWlan802dot11rPmkR0Lifetime**

PMK-RO lifetime in seconds (dot11FTR0KeyLifetime)

When Session-Timeout attribute is provided by RADIUS server, then $\min(\text{Session-Timeout}, \text{cfgWlan802dot11rPmkR0Lifetime})$ is used.

Ranges

- 0 - Infinite lifetime (disabled)
- 1..59 - Reserved, do not use
- 60..2147483647 - Allowed lifetime in seconds

Applies to AP. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	0 - 2147483647
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.13.1.6

9.0.1.13.54 **cfgWlan802dot11rPmkR1KeyHolderIdentifier**

PMK-R1 Key Holder identifier (dot11FTR0KeyHolderID)

6-octet identifier as a hex string. This may be the same as the local MAC address. Default magic number 000102030405 means use own mac address (bssid).

Format: 020102030405

Applies to AP. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	12 - 12
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.13.1.7

9.0.1.13.55 **cfgWlan802dot11rPmkR1Push**

Whether PMK-R1 push is enabled at R0KH

- **0** do not push PMK-R1 to all configured R1KHs (default).
- **1** push PMK-R1 to all configured R1KHs whenever a new PMK-R0 is derived.

Applies to AP. 802.11n products only.

<i>Enumeration</i>	donotpush (0), push (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.13.1.9

9.0.1.13.56 **cfgWlan802dot11rR0KHTable**

Wireless 802dot11r R0KH

List of R0KHs in the same Mobility Domain. This list is used to map R0KH-ID (NAS Identifier) to a destination MAC address when requesting PMK-R1 key from the R0KH that the STA used during the Initial Mobility Domain Association.

Format: <MAC address> <NAS Identifier> <128-bit key as hex string>

Examples:

```
r0kh=02:01:02:03:04:05 r0kh-1.example.com 000102030405060708090a0b0c0d0e0f
r0kh=02:01:02:03:04:06 r0kh-2.example.com 00112233445566778899aabbccddeeff
```

This may also contain a wildcard entry to transmit a request to the broadcast address instead of a unicast. This has the advantage that not all potential R0KH have to be configured. The provided key has to match the configured wildcard key in the `cfgWlan802dot11rR1KHTable`

Wildcard entry

r1kh=ff:ff:ff:ff:ff:ff * 0123456789abcdef0123456789abcdef

Applies to AP. 802.11n products only.

<i>Status</i>	current
<i>Range</i>	0 - 511
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.14

9.0.1.13.57 cfgWlan802dot11rR0KHId

ID of the R0KH table

The configuration item `cfgWlan802dot11rR0KHParameter` references to this ID.

Applies to AP. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.14.1.2

9.0.1.13.58 cfgWlan802dot11rR0KHEnabled

Enable this entry in the R0KH list

Applies to AP. 802.11n products only.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.14.1.3

9.0.1.13.59 cfgWlan802dot11rR0KHDestinationMac

MAC addresses of possible R0KHs from which PMK-R1 can be requested

Format: 02:01:02:03:04:05

Applies to AP. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	17 - 17
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.14.1.4

9.0.1.13.60 **cfgWlan802dot11rR0KHHID**

NAS Identifier of all R0KHs to map to the MAC address

See the field: `cfgWlan802dot11rPmkR0KeyHolderIdentifier`.

Applies to AP. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 48
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.14.1.5

9.0.1.13.61 **cfgWlan802dot11rR0KHKey**

Static Key of the R0KH

Connecting R1KHs need to have this key configured.

Format: 000102030405060708090a0b0c0d0e0f

Applies to AP. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	32 - 32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.14.1.6

9.0.1.13.62 **cfgWlan802dot11rR1KHTable**

Wireless 802dot11r R1KH

List of R1KHs in the same Mobility Domain This list is used to map R1KH-ID to a destination MAC address when sending PMK-R1 key from the R0KH. This is also the list of authorized R1KHs in the MD that can request PMK-R1 keys.

Format: <MAC address> <R1KH-ID> <128-bit key as hex string>

Examples:

```
r1kh=02:01:02:03:04:05 02:11:22:33:44:55 000102030405060708090a0b0c0d0e0f  
r1kh=02:01:02:03:04:06 02:11:22:33:44:66 00112233445566778899aabbccddeeff
```

This may also contain a wildcard entry allowing everyone to request a PMK-R1 from this R0KH. The provided key has to match the configured wildcard key in the `cfgWlan802dot11rR0KHTable`

Wildcard entry

```
r1kh=00:00:00:00:00:00 00:00:00:00:00:00 0123456789abcdef0123456789abcdef
```

Applies to AP. 802.11n products only.

<i>Status</i>	current
<i>Range</i>	0 - 511
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.15

9.0.1.13.63 `cfgWlan802dot11rR1KHId`

ID of the R1KH table

The configuration item `cfgWlan802dot11rR1KHParameter` references to this ID.

Applies to AP.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.15.1.2

9.0.1.13.64 `cfgWlan802dot11rR1KHEnabled`

Enable this entry in the R1KH list

Applies to AP. 802.11n products only.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.15.1.3

9.0.1.13.65 **cfgWlan802dot11rR1KHDestinationMac**

MAC addresses of R1KHs which can request PMK-R1 from the local R0KH

Format: 02:01:02:03:04:05

Applies to AP.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	17 - 17
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.15.1.4

9.0.1.13.66 **cfgWlan802dot11rR1KHHID**

PMK-R1 Key Holder identifier (dot11FTR1KeyHolderID)

6-octet identifier as a hex string to map to the MAC. This may be the same as the destination MAC. See the field `cfgWlan802dot11rPmkR1KeyHolderIdentifier`.

Format: 02:01:02:03:04:05

Applies to AP. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	17 - 17
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.15.1.5

9.0.1.13.67 **cfgWlan802dot11rR1KHKey**

Static Key of the R0KH

These keys are used wenn sending updates to R1KHs from the local R0KH. The respective key has to match the respective entry on the target in the field `cfgWlan802dot11rR0KHKey`.

Format: 000102030405060708090a0b0c0d0e0f

Applies to AP. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	32 - 32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.15.1.6

9.0.1.13.68 **cfgWlanNeighbourTable**

Hostapd Neighbour Table

<i>Status</i>	current
<i>Range</i>	0 - 511
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.16

9.0.1.13.69 **cfgWlanNeighbourId**

ID of the neighbour table

The configuration item `cfgWlanIfaceNeighbourParameter` references to this ID.

Applies to AP. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.16.1.2

9.0.1.13.70 **cfgWlanNeighbourEnabled**

Enable this entry in the list

Applies to AP. 802.11n products only.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.16.1.3

9.0.1.13.71 **cfgWlanNeighbourBSSID**

BSSID (MAC address) of the neighbour

Format: 00:14:5a:02:10:42

Applies to AP.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	17 - 17
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.16.1.4

9.0.1.13.72 **cfgWlanNeighbourFrequency**

Frequency in MHz of the neighbour

Applies to AP. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	1 - 6000
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.16.1.5

9.0.1.13.73 **cfgWlanInterfaceTable**

Wireless Virtual Interfaces

<i>Status</i>	current
<i>Range</i>	0 - 63
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.2

9.0.1.13.74 **cfgWlanifaceDtim**

Number of beacons between transmission of DTIM element

This attribute specifies the number of beacon intervals that shall elapse between transmission of beacon frames containing a TIM element whose DTIM count field is 0. This value is transmitted in the DTIM Period field of Beacon frames.

The DTIM counter is used to signal to power saving sleeping clients how long they can sleep between wakeups to get data from the AP.

Applies to AP. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.10

9.0.1.13.75 **cfgWlanifaceAplsolate**

Enable AP clients isolation

If enabled, clients connected to this AP can't communicate to each other.

Applies to AP.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.100

9.0.1.13.76 **cfgWlanifaceBitrates**

Fixed MCS index for 802.11n rates

Set to -1 to disable (leave on auto). Allows for entering multiple indices divided by a space which are then used in auto rate. This entry is only active when an n-rate is set in `cfgWlanDevModulation` (not only g-rate or only a-rate).

Applies to AP and STA. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.11

9.0.1.13.77 **cfgWlanifaceBeaconInterval**

Time in kus (1.024 ms) between the sending of beacon frames

Applies to AP. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	15 - 65535
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.12

9.0.1.13.78 **cfgWlanifaceLlcBroadcastVlan**

VLAN to send broadcast LLC frame after Handoff

A space and/or comma separated list of VLANs. This list defines to which VLANs the broadcast LLC frame is sent. The broadcast LLC frames is sent when a STA connects to an AP. This broadcast LLC frame is used to update the FDB of all switches on the backbone which are involved on the path on which data flows.

The value 0 specifies that the frame is sent untagged.

When this value is set to a value other than 0, make sure that the VLAN mode of the wireless interface `cfgNetWlanVlanMode` allows tagged frames. Also ensure that `cfgNetWlanTrunk` includes the specified VLAN(s).

Examples:

- 69
- 12, 24, 69
- 0 12 24 69

Applies to AP.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.120

9.0.1.13.79 **cfgWlanifaceWmeParameter**

Reference ID to the WME parameter table

Uses all parameters in the `cfgWlanWmeTable` which have as `cfgWlanWmeId` the value set here.

Applies to AP and STA. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.13

9.0.1.13.80 **cfgWlanifaceWmeEnabled**

Enables usage of the WME parameter table

When using legacy rates (a-rates and g-rates) this is optional. When using n-rates this has to be enabled at all times.

Applies to AP. 802.11n products only.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.14

9.0.1.13.81 **cfgWlanifaceScanList**

Index to specify a frequency list to be scanned

Is only active when `cfgWlanIfaceMode` is set to `sta(1)`. Set to **-1**, to scan the frequency defined in `cfgWlanDevFrequency`. Set to **-2**, to scan all frequencies allowed by the country code.

Applies to STA. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	-2 - 23
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.15

9.0.1.13.82 `cfgWlanIfaceIgnoreBroadcastSsid`

Hide SSID

Send empty SSID in beacons and ignore probe request frames that do not specify the full SSID, i.e., require stations to know SSID.

Applies to AP.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.16

9.0.1.13.83 `cfgWlanIfaceMacaddrAcl`

Mode of the MAC access control list

- **acceptunlessdeny(0)**: Accept unless deny filter. Accept every MAC unless it is on the list defined in `cfgWlanAclBlackTable`.
- **denyunlessaccept(1)**: Deny unless accept filter. Deny every MAC unless it is on the list defined in `cfgWlanAclWhiteTable`.
- **radius(2)**: Use RADIUS to accept/deny clients.

Applies to AP. 802.11n products only.

<i>Enumeration</i>	acceptunlessdeny (0), denyunlessaccept (1), radius (2)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.17

9.0.1.13.84 `cfgWlanIfaceMaxNumSta`

Maximum Number of Clients

802.11ac products have an upper limit of 512 clients.

Applies to AP.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	1 - 2007
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.18

9.0.1.13.85 **cfgWlanifaceBssid**

BSSID of AP

Set 00:00:00:00:00:00 to use the MAC address stored in the flash of the wireless card itself. If this is the second or more virtual AP on this card it will automatically set the locally assigned bit and add an increasing counter in the leading 0 range.

When the device is operating in STA mode the MAC address of the wireless interface can be configured.

Format: 00:14:5a:02:10:42

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	17 - 17
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.19

9.0.1.13.86 **cfgWlanifaceName**

Name of the virtual wireless interface

Applies to AP and STA.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.2

9.0.1.13.87 **cfgWlanifaceLegacyRates**

Wireless legacy data rates

- **11b:** 1, 2, 5.5, 11 Mbps
- **11a/g:** 6, 9, 12, 18, 24, 36, 48, 54 Mbps

The values are interpreted as flags:

- ****auto(0)***
- **1Mbps(1)**
- **2Mbps(2)**
- **5.5Mbps(4)**
- **6Mbps(8)**
- **9Mbps(16)**
- **11Mbps(32)**
- **12Mbps(64)**
- **18Mbps(128)**
- **24Mbps(256)**
- **36Mbps(512)**
- **48Mbps(1024)**
- ****54Mbps(2048)**

When `cfgWlanDevBandwidth` is equal 3 (quarter rates) the rate is quarter i.e. 36Mbps becomes 9Mbps. When `cfgWlanDevBandwidth` is equal 4 (half rates) the rate is halved i.e. 36Mbps becomes 18Mbps.

This entry only has an effect on clients which only can use g-rates or a-rates. For clients which support MCS-rates the entry `cfgWlanIfaceBitrates` can be used to allow specific rates.

Applies to AP and STA. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	0 - 2048
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.20

9.0.1.13.88 `cfgWlanIface4addr`

This option allows to bridge the STA side

When used on the STA, the corresponding AP has to enable this feature as well. This option may not be enabled together with `cfgWlanIfaceL2nat`.

Applies to AP and STA.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.21

9.0.1.13.89 **cfgWlanInactivityTimeout**

Allowed idle time before station is removed

If a station does not send anything in `ap_max_inactivity` seconds, an empty data frame is sent to it in order to verify whether it is still in range. If this frame is not ACKed, the station will be disassociated and then deauthenticated. This feature is used to clear the station table of old entries when the STAs move out of range.

Applies to AP. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	15 - 65535
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.23

9.0.1.13.90 **cfgWlanUseVendorSsid**

Enable Vendor Element containing the SSID

When `cfgWlanIgnoreBroadcastSsid` is enabled, a passively scanning STA (forced or because of DFS) has no way of detecting the AP it tries to find. On an AP this options adds the hidden SSID as vendor element. On a STA this options allows it to use the vendor element in the beacon.

Applies to AP and STA. 802.11n products only.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.26

9.0.1.13.91 **cfgWlanDevice**

Maps the virtual wireless interface to the radio device

Applies to AP and STA.

<i>Enumeration</i>	radio0 (0), radio1 (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.3

9.0.1.13.92 **cfgWlanMFP80211w**

Controls usage of 802.11w Management Frame Protection (MFP)

On AP, if set to **optional(1)**, MFP will be used only for clients which have it also enabled (either **optional(1)** or **required(2)**). If set to **required(2)**, only MFP enabled clients will be able to connect.

On STA, if set to **optional(1)**, MFP will be used only for Access Points which have it also enabled (either **optional(1)** or **required(2)**). If set to **required(2)**, client will connect only to MFP enabled Access Points.

Applies to AP and STA.

<i>Enumeration</i>	disabled (0), optional (1), required (2)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.30

9.0.1.13.93 **cfgWlanifaceleeee80211wMaxTimeout**

802.11w Management Frame Protection (MFP) timeout

Association SA query maximum timeout (in TU = 1.024 ms; for MFP) (maximum time to wait for a SA query response).

Applies to AP.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	1 - 4000
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.31

9.0.1.13.94 **cfgWlanifaceleeee80211wRetryTimeout**

802.11w Management Frame Protection (MFP) retry timeout

Association SA query retry timeout (in TU = 1.024 ms; for MFP) (time between two subsequent SA query requests).

Applies to AP.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	1 - 4000
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.32

9.0.1.13.95 **cfgWlanifaceMode**

Wireless Operation Mode

Allowed modes are:

- **ap(0)**: defines the interface as Access Point (AP)
- **sta(1)**: defines the interface as Station (STA)
- **monitor(2)**: defines the interface as Monitor (MON)
- **mesh(3)**: defines the interface as Mesh (MESH)

Note: Some products do not support all modes. Please check the data-sheet of your product.

Applies to AP and STA.

<i>Enumeration</i>	ap (0), sta (1), monitor (2), mesh (3)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.4

9.0.1.13.96 **cfgWlanifaceAcsList**

Index to specify a frequency list for Automated Channel Selection

Used for Automated Channel Selection (ACS) support when in AP mode. To disable set to **-1**.

Applies to AP. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.40

9.0.1.13.97 **cfgWlanifaceSsid**

The Service Set Identifier (SSID) of the wireless interface

This is the arbitrary name of the wireless network this interface is part of.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.5

9.0.1.13.98 **cfgWlanifaceEncryption**

Wireless Encryption Mode

Supported encryption modes:

- **open(0)** means open network without encryption.
- **psk(3)** means WPA-PSK security standard (IEEE 802.11i, AES-CCMP)
- **eap(6)** WPA-EAP (EAP-TLS/TTLS/PEAP)
- **sae(7)** enables SAE (WPA3-Personal)
- **owe(8)** Opportunistic Wireless Encryption (WPA3 Enhanced Open)
- **saepsk(9)** enables WPA3-Personal Transition mode (SAE + PSK)

WPA-EAP enables 802.1X support.

Applies to AP and STA.

<i>Enumeration</i>	open (0), psk (3), eap (6), sae (7), owe (8), saepsk (9)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.6

9.0.1.13.99 **cfgWlanInterfaceNeighbourReport**

Enable/disable neighbour reporting

A STA can request the neighbour table from an AP and use this information to improve its handoff decision.

Applies to AP. 802.11n products only.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.60

9.0.1.13.100 **cfgWlanInterfaceNeighbourParameter**

Reference ID to the neighbour table

Uses all neighbours in the `cfgWlanNeighbourTable` which have as `cfgWlanNeighbourId` the value set here.

Applies to AP. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.61

9.0.1.13.101 **cfgWlanIfacePassword**

Wireless password if an encryption is in use

Each character in the pass-phrase must have an encoding in the range of 32 to 126 (decimal), inclusive. (IEEE Std. 802.11i-2004, Annex H.4.1) The space character is included in this range.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	8 - 63
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.7

9.0.1.13.102 **cfgWlanIfacePassiveScanning**

Wireless Scanning Mode

If the scanning mode is set to active(0) the station will send a probe request to detect available access points if it's allowed by the country code.

If the scanning mode is set to passive(1) the station will always perform passive scanning to detect available access points.

Applies to STA. 802.11n products only.

<i>Enumeration</i>	active (0), passive (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.8

9.0.1.13.103 **cfgWlanIfaceL2nat**

This option allows to bridge the STA side

It is intended to be used in setups where the STA is doing handoff between multiple APs. There is no configuration on the APs required. This is an alternative to `cfgWlanIface4addr`. This option may not be enabled together with `cfgWlanIface4addr`.

Applies to STA.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.80

9.0.1.13.104 `cfgWlanIfaceL2natLearningMode`

Type of frames from which the L2nat IP/MAC table may be learned

When `cfgWlanIfaceL2nat` is enabled a table of the association between IP-addresses and MAC-addresses is kept. This options specifies from which type of frames this association may be learned:

- **both(0)**: MAC/IP association is learned from ARP frames and from IP frames.
- **arp(1)**: MAC/IP association is learned only from ARP frames.

Applies to STA.

<i>Enumeration</i>	both (0), arp (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.81

9.0.1.13.105 `cfgWlanIfaceL2natDefaultDestination`

Default destination for L2nat

This option defines the default MAC address to send frames to. Whenever a unicast frame is received for which no learned entry exists, or which isn't a L3 frame (e.g a custom L2 protocol), the frame will be sent to the address specified here.

This address is by default the broadcast address.

Format: `ff:ff:ff:ff:ff:ff`

When the STA is directly connected to a router, the only possible destination is the MAC of the attached router. However it is cumbersome to manually configure the MAC address for the router. When this field is set to `00:00:00:00:00:00`, it is in router-auto-learn-mode. Depending on `cfgWlanIfaceL2natLearningMode` it will automatically learn the default destination to the MAC of the attached router from the flowing frames. In this mode `cfgWlanIfaceL2natLearningMode` should be set to **arp(1)**.

Applies to STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	17 - 17
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.82

9.0.1.13.106 `cfgWlanIfaceBeaconMiss`

Number of consecutive beacons misses before the station disconnects

Applies to STA. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.9

9.0.1.13.107 **cfgWlanfaceTimeAdvertisement**

Enable AP to include local time in association response frames

This provides a mean for APs to distribute its local system time to STAs in setups where devices have no RTC and start up with an invalid system time.

If configured, AP embeds its current system time as IE in association response frames, which STAs can use to update to before they get access to trusted time sources like NTP.

Note that enabling this feature increases the size of response frames, therefore this feature shall only be activated where required and defaults to disabled.

Applies to AP. 802.11n products only.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.90

9.0.1.13.108 **cfgWlanHandoffTable**

Wireless handoff parameters

<i>Status</i>	current
<i>Range</i>	0 - 63
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.3

9.0.1.13.109 **cfgWlanHoFilterLongX**

IIR Filter Parameter x for long RSSI filter.

Applies to STA. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.3.1.10

9.0.1.13.110 **cfgWlanHoFilterLongY**

IIR Filter Parameter y for long RSSI filter.

Applies to STA. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.3.1.11

9.0.1.13.111 **cfgWlanHoScanRateLimitTime**

Time in milliseconds (4ms steps) in which a number of attempts to connect to an AP can be tried.

Applies to STA. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	4 - 65535
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.3.1.12

9.0.1.13.112 **cfgWlanHoScanRateLimitTries**

Number of attempts to connect to an AP before the AP is blacklisted and ignored. The AP is removed from the blacklist after `cfgWlanHoScanRateLimitTime` since the first attempt.

Applies to STA. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.3.1.13

9.0.1.13.113 **cfgWlanHoPassiveChanTime**

Time in milliseconds (4ms steps) we stay on a channel during passive scanning and wait for beacons.

Applies to STA. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	1 - 65535
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.3.1.16

9.0.1.13.114 **cfgWlanHoLevelLow**

Scanning level low in RSSI

When the RSSI level to the current connected AP is below this value, perform a handoff to the next AP.

This value is only active when `cfgWlanHoProfile` is set to T2Gv3. When configured on an AP, this value is advertised for clients to use. When configured on a STA, this is the default value which is used when the currently connected AP doesn't provide a different value.

Applies to AP and STA. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.3.1.18

9.0.1.13.115 **cfgWlanHoLevelHigh**

Scanning level high in RSSI

When the RSSI level to the current connected AP is higher or equal this value, perform a handoff to the next AP.

This value is only active when `cfgWlanHoProfile` is set to T2Gv3. When configured on an AP, this value is advertised for clients to use. When configured on a STA, this is the default value which is used when the currently connected AP doesn't provide a different value.

Applies to AP and STA. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.3.1.19

9.0.1.13.116 **cfgWlanHolfaceName**

Name of the virtual wireless interface

Applies to STA. 802.11n products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.3.1.2

9.0.1.13.117 `cfgWlanHoDistanceNear`

Distance level near

When the measured distance (in meters) to the current connected AP is lower or equal this value, perform a handoff to the next AP.

This value is only active when `cfgWlanHoProfile` is set to T2Gv3. When configured on an AP, this value is advertised for clients to use. When configured on a STA, this is the default value which is used when the currently connected AP doesn't provide a different value.

Note: Distance value is not in meters.

Applies to AP and STA. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	0 - 114750
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.3.1.20

9.0.1.13.118 `cfgWlanHoDistanceFar`

Distance level far

When the measured distance (in meters) to the current connected AP is greater or equal this value, perform a handoff to the next AP.

This value is only active when `cfgWlanHoProfile` is set to T2Gv3. When configured on an AP, this value is advertised for clients to use. When configured on a STA, this is the default value which is used when the currently connected AP doesn't provide a different value.

Note: Distance value is not in meters.

Applies to AP and STA. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	0 - 114750
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.3.1.21

9.0.1.13.119 `cfgWlanHoDistanceMeasurementPeriod`

Distance ranging measurement period in milliseconds.

Setting zero disables distance handoff. This value is only active when `cfgWlanHoProfile` is set to T2Gv3.

Typical ranging measurement period values are in the range from 200 ms to 1000 ms. Values lower than 100 ms are not recommended.

Applies to STA. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	0 - 100000
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.3.1.22

9.0.1.13.120 **cfgWlanHoDistanceFilterX**

IIR Filter Parameter x for distance measurement.

This value is only active when `cfgWlanHoProfile` is set to T2Gv3.

Applies to STA. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.3.1.23

9.0.1.13.121 **cfgWlanHoDistanceFilterY**

IIR Filter Parameter y for distance measurements.

This value is only active when `cfgWlanHoProfile` is set to T2Gv3.

Applies to STA. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.3.1.24

9.0.1.13.122 **cfgWlanHoProfile**

Handoff Profile

- **t2gv1(1)**: Train to Ground v1
- **t2gv2(2)**: Train to Ground v2
- **t2gv2fg(3)**: Train to Ground v2 fg scan
- **t2gv3(4)**: Train to Ground v3 (RSSI high/low, Distance near/far)

Applies to STA. 802.11n products only.

<i>Enumeration</i>	t2gv1 (1), t2gv2 (2), t2gv2fg (3), t2gv3 (4)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.3.1.3

9.0.1.13.123 **cfgWlanHoScanningLevel**

Scanning level in RSSI

When the RSSI level of the currently connected access point drops below the value configured in this parameter, the STA will scan for better access points on all frequencies specified by the scan list configured in `cfgWlanInterfaceScanList` and `cfgWlanFreqTable`.

Applies to STA. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	0 - 95
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.3.1.5

9.0.1.13.124 **cfgWlanHoBeacons**

Number of beacons which have to be received from an AP before a decision about handoff is allowed. Essentially forces the STA to stay on a given AP for before doing another handoff.

Applies to STA. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	4 - 20
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.3.1.6

9.0.1.13.125 **cfgWlanHoRecovery**

Recovery time in milliseconds after a successful handoff

During this time no further handoff will be executed.

Applies to STA. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	0 - 2000
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.3.1.7

9.0.1.13.126 **cfgWlanHoFilterMode**

RSSI filter mode.

- **short(0)**: high responsive static IIR RSSI filter.
- **long(1)**: configurable IIR RSSI filter.

Applies to STA. 802.11n products only.

<i>Enumeration</i>	short (0), long (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.3.1.9

9.0.1.13.127 **cfgWlanFreqTable**

Frequency list entry

<i>Status</i>	current
<i>Range</i>	0 - 23
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.4

9.0.1.13.128 **cfgWlanFFreq8**

Frequency in MHz, 0 is interpreted as empty

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.4.1.10

9.0.1.13.129 **cfgWlanFFreq9**

Frequency in MHz, 0 is interpreted as empty

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.4.1.11

9.0.1.13.130 **cfgWlanFFreq10**

Frequency in MHz, 0 is interpreted as empty

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.4.1.12

9.0.1.13.131 cfgWlanFFreq11

Frequency in MHz, 0 is interpreted as empty

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.4.1.13

9.0.1.13.132 cfgWlanFFreq12

Frequency in MHz, 0 is interpreted as empty

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.4.1.14

9.0.1.13.133 cfgWlanFFreq13

Frequency in MHz, 0 is interpreted as empty

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.4.1.15

9.0.1.13.134 cfgWlanFFreq14

Frequency in MHz, 0 is interpreted as empty

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.4.1.16

9.0.1.13.135 **cfgWlanFFreq15**

Frequency in MHz, 0 is interpreted as empty

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.4.1.17

9.0.1.13.136 **cfgWlanFFreq16**

Frequency in MHz, 0 is interpreted as empty

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.4.1.18

9.0.1.13.137 **cfgWlanFFreq17**

Frequency in MHz, 0 is interpreted as empty

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.4.1.19

9.0.1.13.138 **cfgWlanFFreq0**

Frequency in MHz, 0 is interpreted as empty

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.4.1.2

9.0.1.13.139 **cfgWlanFFreq18**

Frequency in MHz, 0 is interpreted as empty

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.4.1.20

9.0.1.13.140 **cfgWlanFFreq19**

Frequency in MHz, 0 is interpreted as empty

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.4.1.21

9.0.1.13.141 **cfgWlanFFreq20**

Frequency in MHz, 0 is interpreted as empty

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.4.1.22

9.0.1.13.142 **cfgWlanFFreq21**

Frequency in MHz, 0 is interpreted as empty

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.4.1.23

9.0.1.13.143 **cfgWlanFFreq22**

Frequency in MHz, 0 is interpreted as empty

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.4.1.24

9.0.1.13.144 **cfgWlanFFreq23**

Frequency in MHz, 0 is interpreted as empty

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.4.1.25

9.0.1.13.145 **cfgWlanFFreq1**

Frequency in MHz, 0 is interpreted as empty

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.4.1.3

9.0.1.13.146 **cfgWlanFFreq2**

Frequency in MHz, 0 is interpreted as empty

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.4.1.4

9.0.1.13.147 **cfgWlanFFreq3**

Frequency in MHz, 0 is interpreted as empty

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.4.1.5

9.0.1.13.148 **cfgWlanFFreq4**

Frequency in MHz, 0 is interpreted as empty

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.4.1.6

9.0.1.13.149 **cfgWlanFFreq5**

Frequency in MHz, 0 is interpreted as empty

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.4.1.7

9.0.1.13.150 **cfgWlanFFreq6**

Frequency in MHz, 0 is interpreted as empty

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.4.1.8

9.0.1.13.151 **cfgWlanFFreq7**

Frequency in MHz, 0 is interpreted as empty

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.4.1.9

9.0.1.13.152 **cfgWlanWmeTable**

Wireless Multimedia Extensions Table

Wireless Multimedia Extensions (WME) based on the IEEE 802.11e standard. It provides basic Quality of Service (QoS) features to IEEE 802.11 networks.

The levels of priority in EDCA are called access categories (ACs). The contention window (CW) can be set according to the traffic expected for each access category, with a wider window needed for categories with heavier traffic.

Applies to AP. 802.11n products only.

<i>Status</i>	current
<i>Range</i>	0 - 31
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.5

9.0.1.13.153 **cfgWlanWmeApAifs**

Arbitration inter-frame space (AIFS)

Is used on the AP itself.

Applies to AP.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	0 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.5.1.10

9.0.1.13.154 **cfgWlanWmeApBurst**

Maximum length for bursting (equivalent to TxOpLimit)

This value is in units of 32us.

Is used on the AP itself.

Applies to AP. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	0 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.5.1.11

9.0.1.13.155 **cfgWlanWmeld**

ID of the WME parameter table

The virtual wireless interface references to this ID, specified by `cfgWlanIfaceWmeParameter`.

Applies to AP. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.5.1.2

9.0.1.13.156 **cfgWlanWmeAc**

WME Access Category

The following categories are available:

- **none(0)** use driver default value of queue
- **BK - background(1)**
- **BE - besteffort(2)**
- **VI - video(3)**
- **VO - voice(4)**

Frames in the VO queue are never aggregated.

Applies to AP. 802.11n products only.

<i>Enumeration</i>	none (0), background (1), besteffort (2), video (3), voice (4)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.5.1.3

9.0.1.13.157 cfgWlanWmeCwMin

Contention window minimum in exponential form

Is used on STAs connected to this AP: Real value = $(2^n) - 1$

Applies to AP. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	0 - 12
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.5.1.4

9.0.1.13.158 cfgWlanWmeCwMax

Contention window maximum in exponential form

Is used on STAs connected to this AP: Real value = $(2^n) - 1$

Applies to AP. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	0 - 12
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.5.1.5

9.0.1.13.159 cfgWlanWmeAifs

Arbitration inter-frame space (AIFS)

Is used on STAs connected to this AP.

Applies to AP. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.5.1.6

9.0.1.13.160 **cfgWlanWmeTxOpMax**

Transmit Opportunity

A Transmit Opportunity (TXOP) is a bound time interval during which a station can send as many frames as possible (as long as the duration of the transmissions does not exceed the maximum duration of the TXOP). A value of 0 indicates that a single MSDU or MMPDU in addition to a possible RTS/CTS or CTS to itself may be transmitted at any PHY rate for each TXOP. This value is in units of 32 us.

Is used on STAs connected to this AP.

Applies to AP. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	0 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.5.1.7

9.0.1.13.161 **cfgWlanWmeApCwMin**

Contention window minimum

Allowed values: 1, 3, 7, 15, 31, 63, 127, 255, 511, 1023.

Is used on the AP itself.

Applies to AP. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	1 - 1023
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.5.1.8

9.0.1.13.162 **cfgWlanWmeApCwMax**

Contention window maximum

Allowed values: 1, 3, 7, 15, 31, 63, 127, 255, 511, 1023

cwMax has to be greater or equal cwMin.

Is used on the AP itself.

Applies to AP. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	1 - 1023
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.5.1.9

9.0.1.13.163 **cfgWlanDbgTable**

Wireless Handoff Debug Parameters

<i>Status</i>	current
<i>Range</i>	0 - 63
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.6

9.0.1.13.164 **cfgWlanDbgRatelimit**

Persistent default value to enable/disable the rate limiter messages in standard syslog.

These log messages are subject to change. DO NOT PARSE!

Applies to STA. 802.11n products only.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.6.1.10

9.0.1.13.165 **cfgWlanDbgLinkmonitor**

Periodically sends a trap containing link information of all connected devices on this interface.

Applies to both AP and STA. 802.11n products only.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.6.1.11

9.0.1.13.166 **cfgWlanDbgBeacontsf**

Persistent default value to enable/disable the Beacon RSSI messages in standard syslog. The TS field contains the internal TSF (mactime) instead of the system uptime.

These log messages are subject to change. DO NOT PARSE!

Applies to STA. 802.11n products only.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.6.1.12

9.0.1.13.167 **cfgWlanDbgRange**

Persistent default value to enable/disable the distance measurement messages in standard syslog.

These log messages are subject to change. DO NOT PARSE!

Note: Distance value is not in meters.

Applies to STA. 802.11n products only.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.6.1.13

9.0.1.13.168 **cfgWlanDbgReports**

Persistent default value to enable/disable the periodical WLAN debug data reporting in standard syslog.

These log messages are subject to change. DO NOT PARSE!

Applies to STA. 802.11n products only.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.6.1.14

9.0.1.13.169 **cfgWlanDbgIfaceName**

Name of the virtual wireless interface.

Applies to STA. 802.11n products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.6.1.2

9.0.1.13.170 **cfgWlanDbgHandoff**

Persistent default value to enable/disable the handoff trap.

These log messages are subject to change. DO NOT PARSE!

Applies to STA. 802.11n products only.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.6.1.3

9.0.1.13.171 **cfgWlanDbgScan**

Persistent default value to enable/disable the scan messages in standard syslog.

These log messages are subject to change. DO NOT PARSE!

Applies to STA. 802.11n products only.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.6.1.4

9.0.1.13.172 **cfgWlanDbgMlme**

Persistent default value to enable/disable the MLME messages in standard syslog.

These log messages are subject to change. DO NOT PARSE!

Applies to STA. 802.11n products only.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.6.1.5

9.0.1.13.173 **cfgWlanDbgEvents**

Persistent default value to enable/disable the events messages in standard syslog.

These log messages are subject to change. DO NOT PARSE!

Applies to STA. 802.11n products only.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.6.1.6

9.0.1.13.174 **cfgWlanDbgBeaconrssi**

Persistent default value to enable/disable the Beacon RSSI messages in commissioning syslog.

These log messages are subject to change. DO NOT PARSE!

Applies to STA. 802.11n products only.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.6.1.7

9.0.1.13.175 **cfgWlanDbgAckrssi**

Persistent default value to enable/disable the ACK RSSI messages in standard syslog.

These log messages are subject to change. DO NOT PARSE!

Applies to STA. 802.11n products only.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.6.1.8

9.0.1.13.176 **cfgWlanDbgBeaconfiltered**

Persistent default value to enable/disable the beacon filtered RSSI messages in commissioning syslog.

These log messages are subject to change. DO NOT PARSE!

Applies to STA. 802.11n products only.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.6.1.9

9.0.1.13.177 **cfgWlanAclWhiteTable**

Wireless MAC Access Control Whitelist

<i>Status</i>	current
<i>Range</i>	0 - 63
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.7

9.0.1.13.178 **cfgWlanAclWhiteEnabled**

Enable this entry in the ACL

Applies to AP.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.7.1.2

9.0.1.13.179 **cfgWlanAclWhiteInterface**

Name of the virtual wireless interface

The ACL is on this entry active.

Applies to AP.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.7.1.3

9.0.1.13.180 **cfgWlanAclWhiteAddr**

MAC address in the ACL

Format: 00:14:5a:02:10:42

Applies to AP.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	17 - 17
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.7.1.4

9.0.1.13.181 **cfgWlanAclWhiteMask**

Mask of the MAC address to specify ranges of MAC addresses. To be used like CIDR notation of IP addresses.

Applies to AP.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	0 - 48
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.7.1.5

9.0.1.13.182 **cfgWlanAclBlackTable**

Wireless MAC Access Control Blacklist

<i>Status</i>	current
<i>Range</i>	0 - 63
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.8

9.0.1.13.183 **cfgWlanAclBlackEnabled**

Enable this entry in the ACL

Applies to AP.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.8.1.2

9.0.1.13.184 **cfgWlanAclBlackInterface**

Name of the virtual wireless interface

The ACL is on this entry active.

Applies to AP.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.8.1.3

9.0.1.13.185 **cfgWlanAcIBlackAddr**

MAC address in the ACL

Format: 00:14:5a:02:10:42

Applies to AP.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	17 - 17
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.8.1.4

9.0.1.13.186 **cfgWlanAcIBlackMask**

Mask of the MAC address

Allows to enable the use of ranges of MAC addresses. To be used like CIDR notation of IP addresses.

Applies to AP.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	0 - 48
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.8.1.5

9.0.1.13.187 **cfgWlanGlobal**

9.0.1.13.187.1 **cfgWlanGlblCountry**

Wireless country code

Note: Refer to documentation which countries are supported for your device.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.9.1

9.0.1.13.187.2 `cfgWlanGlblLinkmonitorInterval`

LinkMonitor interval in milliseconds

A new trap is sent every milliseconds.

Note: A short interval and/or numerous connections may affect system performance negatively.

Applies to AP and STA. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	200 - 60000
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.9.2

9.0.1.13.187.3 `cfgWlanGlblLinkmonitorQmrrlogging`

LinkMonitor QMRR logging

If enabled, the QMRR statistics collected for phy0 are periodically printed to syslog at the interval set in `cfgWlanGlblLinkmonitorInterval`.

Applies to AP and STA. 802.11n products only.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.9.3

9.0.1.13.187.4 `cfgWlanGlblConnectionStatusWlanInterface`

Persistent default of the volatile setting `swDrvConStatWlanIf`

The value set here is used to initialize `swDrvConStatWlanIf`. Initialisation happens on startup or configuration change.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	4 - 17
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.3.9.4

9.0.1.14 `cfgRouting`

9.0.1.14.1 cfgRouteDefault

9.0.1.14.1.1 cfgRouteDefGateway

Default Gateway

The default gateway defines the node on an IP network that serves as a router for any other network which is not defined in the routing table.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	IpAddress
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.4.1.1

9.0.1.14.1.2 cfgRouteDefGwOverride

Override the default gateway

Override a default gateway previously received via DHCP with the value in `cfgRouteDefGateway`. If this is disabled and a default gateway already exists it will not be changed.

Applies to AP and STA.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.4.1.2

9.0.1.14.1.3 cfgRouteTable

Static Routes

<i>Status</i>	current
<i>Range</i>	0 - 265
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.4.2

9.0.1.14.1.4 cfgRouteTableEnabled

Enable/Disable this route entry

Applies to AP and STA.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.4.2.1.2

9.0.1.14.1.5 **cfgRouteTableDestinationNetwork**

Destination network in CIDR notation

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.4.2.1.3

9.0.1.14.1.6 **cfgRouteTableGateway**

Gateway to destination network

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	IpAddress
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.4.2.1.5

9.0.1.14.1.7 **cfgRouteTableSource**

Source for traffic to destination network

Optional, use only if you have multiple possible sources.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	IpAddress
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.4.2.1.6

9.0.1.14.1.8 **cfgRouteTableCarpld**

The CARP instance which brings this route up.

Has to be set to -1 when this is a normal route and should not be handled by a CARP instance. All routes which have a value 0..15 are brought up by the respective CARP instance when it becomes a

master for an IP. This allows to create routes which are routed over a CARP-Address.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	-1 - 15
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.4.2.1.8

9.0.1.14.1.9 cfgMRouteTable

Static Multicast Routes

<i>Status</i>	current
<i>Range</i>	0 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.4.3

9.0.1.14.1.10 cfgMRouteTableEnabled

Enable/Disable this multicast route entry

Applies to AP and STA.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.4.3.1.2

9.0.1.14.1.11 cfgMRouteTableInput

Input Interface

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.4.3.1.3

9.0.1.14.1.12 cfgMRouteTableSource

Unicast source address to listen to

If it is set to 0.0.0.0 all multicast traffic will be forwarded.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	IpAddress
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.4.3.1.4

9.0.1.14.1.13 cfgMRouteTableGroup

Multicast group to forward

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	IpAddress
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.4.3.1.5

9.0.1.14.1.14 cfgMRouteTableOutput

Output interface(s)

Can be a list of interface names separated by spaces.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.4.3.1.6

9.0.1.15 cfgNlm

9.0.1.15.1 cfgNlmGlobal

9.0.1.15.1.1 cfgNlmGlbEnabled

Enable the Network Link Monitor.

Applies to AP and STA.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.40.1.1

9.0.1.15.1.2 cfgNlmMonitorTable

NLM Monitor Table

<i>Status</i>	current
<i>Range</i>	0 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.40.2

9.0.1.15.1.3 cfgNlmMonUpAction

NLM 'up' state monitor action.

The action is executed on a monitor-state transition to 'up'.

Set to 0 to disable (i.e. no action).

Supported actions:

1xxx (offset: 1000, x: CARP group from 1 to 255) Un-demote CARP group defined by `cfgNetCarpLocalInterface`

20xx (offset: 2000, x: wlan interface from 0 to 15) Enable Access Point operation on the wireless interface. Note: Currently all wireless interfaces on radio0 are enabled.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	0 - 2255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.40.2.1.10

9.0.1.15.1.4 cfgNlmMonDownAction

NLM 'down' state monitor script.

The action is executed on a monitor-state transition to 'down'.

Set to 0 to disable (i.e. no action).

Supported actions:

1xxx (offset: 1000, x: CARP group from 1 to 255) Demote CARP group defined by `cfgNetCarpLocalInterface`

20xx (offset: 2000, x: wlan interface from 0 to 15) Disable Access Point operation on the wireless interface. Note: Currently all wireless interfaces on radio0 are disabled.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	0 - 2255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.40.2.1.11

9.0.1.15.1.5 **cfgNlmMonScanLoopInterval**

Scan Loop Debounce Interval in Milliseconds

If set to non-zero it will mark interface 'down' after receiving scan loop trap 415 and mark it 'up' after scan loop interval if no other 415 events have been received.

Applies to STA. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	0 - 2147483647
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.40.2.1.12

9.0.1.15.1.6 **cfgNlmMonEnabled**

Enable this entry in the monitor list.

Applies to AP and STA.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.40.2.1.2

9.0.1.15.1.7 **cfgNlmMonInterval**

Execution Interval of this Monitor in Milliseconds

In case of an event driven monitor (e.g. `cfgNlmMonType` set to **wlan(2)**), this defines a timeout.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	0 - 2147483647
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.40.2.1.3

9.0.1.15.1.8 cfgNlmMonCount

NLM monitor counter.

The number of times the measured criteria has to be down, until the monitor is reported as down.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	0 - 65535
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.40.2.1.4

9.0.1.15.1.9 cfgNlmMonType

Objects which can be monitored.

- **phy(0)** monitor checks periodically the link status of the ethernet interfaces defined by `cfgNlmMonInterface`. If at least one interface in the list specified is up the monitor is considered up.
- **icmp(1)** monitor pings periodically the destination defined by `cfgNlmMonDestination`. If the destination does reply to the ECHO request within the `cfgNlmMonInterval` the monitor is considered up.
- **wlan(2)** monitor listens to link status events of the wireless interface defined by `cfgNlmMonInterfaces`.

The wlan monitor consists of 3 components: **Long Handoff Detector** - Triggers when after dis-association no authorization event is detected within the configured time in `cfgNlmMonInterval`. **Scan Loop Detector** - Triggers immediately on trap 415. This happens if there is no AP or only a single AP which stays below/above the Handoff thresholds. This trap is only generated when `cfgWlanHoProfile` is set to 2 or higher. **Handoff Loop Detector** - Triggers if there have been `cfgNlmMonCount` number of Handoff events within `cfgNlmMonCount * (cfgNlmMonInterval + cfgNlmMonScanLoopInterval)`.

The wlan monitor recovers: **Long Handoff Detector** - Immediately after the next successful authorization. **Scan Loop Detector** - After the time of the last 415 trap event + the configured `cfgNlmMonScanLoopInterval`. When down, this is checked regularly in `cfgNlmMonScanLoopInterval` intervals. **Handoff Loop Detector** - When there are less than `cfgNlmMonCount` Handoff events within the time-window `cfgNlmMonCount * (cfgNlmMonInterval + cfgNlmMonScanLoopInterval)`. When down, this is checked regularly in `cfgNlmMonScanLoopInterval` intervals.

Note: WLAN monitor is supported for 802.11n products only.

Applies to AP and STA.

<i>Enumeration</i>	phy (0), icmp (1), wlan (2)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.40.2.1.5

9.0.1.15.1.10 cfgNlmMonInterfaces

NLM monitor interface(s).

This parameter specifies the name of the interface(s) which is/are to be monitored.

For the type **phy(0)** multiple comma separated interfaces are allowed.

For the type **wlan(2)** a single interface (802.11n products only).

All other modes ignore this parameter.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.40.2.1.6

9.0.1.15.1.11 cfgNlmMonDestination

NLM monitor destination.

This parameter specifies the IPv4 address which is to be monitored.

It will only be used when `cfgNlmMonType` is set to **icmp(1)**.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	IpAddress
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.40.2.1.7

9.0.1.16 cfgIplTables

9.0.1.17 cfgQos

9.0.1.17.1 cfgQosL3PrioEnabled

Layer 3 Prioritization

Controls IP Precedence based priority assignments.

Actual prioritization on the wireless link only occurs if `cfgQosWmeEnabled` is enabled as well.

Applies to AP and STA. 802.11n products only.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.6.1

9.0.1.17.2 cfgQosDscpToTidMapTable

DSCP to TID map

Mapping table from DSCP class selector (IP TOS) to wireless priority (TID).

Applies to AP and STA. 802.11n products only.

<i>Status</i>	current
<i>Range</i>	0 - 7
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.6.2

9.0.1.17.3 cfgQosDscpToTidMapValue

Layer 2 priorities for IP precedence values 0-7

Applies to AP and STA. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	0 - 7
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.6.2.1.2

9.0.1.17.4 cfgQosVlanToTidMapTable

802.1p to TID map

Mapping table from layer 2 priorities (802.1p) to wireless priority (TID).

<i>Status</i>	current
<i>Range</i>	0 - 7
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.6.3

9.0.1.17.5 **cfgQosVlanToTidMapValue**

Layer 2 priorities for VLAN priorities 0-7.

Applies to AP and STA. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	0 - 7
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.6.3.1.2

9.0.1.17.6 **cfgQosIpToTidMapTable**

IP header to TID map

Mapping table from IP header (Source, Destination, Protocol, Port), to wireless priority (TID).

<i>Status</i>	current
<i>Range</i>	0 - 127
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.6.4

9.0.1.17.7 **cfgQosIpToTidMapSrcNet**

Source network for ip prioritization rule

In CIDR format.

Applies to AP and STA. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	OctetString
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.6.4.1.2

9.0.1.17.8 **cfgQosIpToTidMapDestNet**

Destination network for ip prioritization rule

In CIDR format.

Applies to AP and STA. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	OctetString
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.6.4.1.3

9.0.1.17.9 **cfgQosIpToTidMapProto**

Protocol for ip prioritization rule

Allowed protocols are:

- **any(0)**
- **udp(1)**
- **tcp(2)**

Applies to AP and STA. 802.11n products only.

<i>Enumeration</i>	any (0), udp (1), tcp (2)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.6.4.1.4

9.0.1.17.10 **cfgQosIpToTidMapSrcPort**

Source port for ip prioritization rule

Use port -1 to match any port. This setting can only be used if the protocol is set to udp or tcp.

Applies to AP and STA. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	-1 - 65536
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.6.4.1.5

9.0.1.17.11 **cfgQosIpToTidMapDestPort**

Destination port for ip prioritization rule

Use port -1 to match any port. This setting can only be used if the protocol is set to udp or tcp.

Applies to AP and STA. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	-1 - 65536
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.6.4.1.6

9.0.1.17.12 cfgQosIpToTidMapPrecedence

Precedence to set for ip prioritization rule

Applies to AP and STA. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	0 - 7
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.6.4.1.7

9.0.1.17.13 cfgQosIpToTidMapEnabled

Enable/disable ip prioritisation rules

Applies to AP and STA. 802.11n products only.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.6.4.1.8

9.0.2 rpc

9.0.2.1 rpcConfiguration

9.0.2.1.1 rpcCfgRevert

In case there are any changes in the configuration section, which are not applied yet, they can be all reverted by writing **all(1)** to this parameter.

Reading this parameter will show the status of the last RPC. A value less than 0 means an error occurred. A value of 0 is returned if the revert process was successful.

Applies to AP and STA.

<i>Enumeration</i>	allError (-1), nop (0), all (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.3.1.1

9.0.2.1.2 rpcCfgApply

All changes to any parameter in the configuration section have to be applied before they come into operation. To apply all new parameters to the device, set this parameter to **all(1)**.

Reading this parameter will show the status of the apply process. A value less than 0 indicates that an error occurred during the last apply process, **nop(0)** means no operation and indicates that no apply process is in operation and no error has occurred. The return value all(1) means the apply process is still running.

Applies to AP and STA.

<i>Enumeration</i>	allError (-1), nop (0), all (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.3.1.2

9.0.2.1.3 rpcCfgFile

Export or import a configuration to or from a file respectively.

Please refer to setCfgFileUrl for more information on how to set the configuration file.

Reading this parameter will show the status of the process. A value less than 0 indicates the occurrence of an error during the last process, **nop(0)** means no operation and indicates that no process is in operation and no error has occurred. A return value greater than 0 means the process is still running.

Applies to AP and STA.

<i>Enumeration</i>	errorImport (-2), errorExport (-1), nop (0), export (1), import (2)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.3.1.4

9.0.2.2 rpcFirmware

9.0.2.2.1 rpcFwFlash

Start Download/Flash of a New Firmware

To flash a new firmware to the device, define a valid URL accessible by the device. Change the firmware URL parameter setFwFileUrl in the settings section, if needed.

- Writing **flash(2)** to this parameter will download and validate the new firmware file. if the downloaded file is recognized as a valid firmware for this device, it will be flashed to the file system of the device.

- Writing **flashWithConfig(3)** to this parameter will download the firmware and the custom config defined with setCfgFileUrl and validate the new firmware file. If the download is recognized as a valid firmware for this device, it will be written to the file system of the device. The supplied custom config will be applied after the upgrade.

Reading this parameter will return the status of the firmware flash process. A value of **flashError(-2)** indicates that the flash process failed during writing. A return value of **downloadError(-1)** indicates the occurrence of an error during download or validation of the firmware/config. A value of **flash(2)** indicates that the device is currently writing the firmware to the file system.

Applies to AP and STA.

<i>Enumeration</i>	flashError (-2), downloadError (-1), nop (0), download (1), flash (2), flashWithConfig (3)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.3.2.1

9.0.2.3 rpcSystem

9.0.2.3.1 rpcSysReboot

Reboot system after n seconds.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.3.3.1

9.0.2.3.2 rpcSysFactoryReset

Factory Reset

Perform a factory reset (i.e. reset device configuration, including administrator password and HTTPS/802.1x certificates, to its default state).

Note: You will not be able to communicate with the device until the factory reset has finished and the device has booted again.

Applies to AP and STA.

<i>Enumeration</i>	nop (0), reset (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.3.3.2

9.0.2.3.3 rpcSysErrorReset

Error Reset

Writing **reset(1)** to this parameter will reset all logged warning and errors of the system. The device LEDs will indicate normal operating state after result.

Applies to AP and STA.

<i>Enumeration</i>	nop (0), reset (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.3.3.3

9.0.2.3.4 rpcSysKernelLogReset

Reset Kernel Logs

- Writing **reset(1)** to this parameter will clear all kernel logs.
- Reading this parameter will show the status of the process. A **nop(0)** means no operation and points out that there is no process in operation. In case the return value is greater than 0 the process is still running.

Applies to AP and STA.

<i>Enumeration</i>	nop (0), reset (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.3.3.4

9.0.2.4 rpcCertificate

9.0.2.4.1 rpcCrtFile

Import or export a HTTPS/802.1X certification/key to or from a file respectively.

Please refer to setCrtFileUrl for more information on how to set the certification/key file URL.

Reading this parameter will show the status of the process. A value less than 0 indicates that an error has occurred during the last process, **nop(0)** means no operation and points out that there is no

process in operation and no error has occurred. A return value greater than 0 means the process is still running.

Applies to AP and STA.

<i>Enumeration</i>	validateerror (-4), deleteerror (-3), exporterror (-2), importerror (-1), nop (0), import (1), export (2), delete (3)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.3.4.1

9.0.2.4.2 rpcCrtRefresh

Refresh Certificates

RPC to manipulate key material in the Certificate Store (CS). Also allows to restart services to use newly installed material.

- **negative values** Represents a failed operation corresponding with its positive value.
- **nop(0)** No Operation in progress.
- **all(1)** Move all available certificates from CS Location 'next' to CS Location 'current'.
- **restartNetwork(10000)** Same as **all(1)** but additionally restart the network (to use all 'current' certificates).
- **restartHttps(10001)** Restart uhttpd to use new key/certificate for https.

Note: The RPC will fail if there are pending config changes.

Applies to STA.

<i>Enumeration</i>	errorRestartHttps (-10001), errorRestartNetwork (-10000), errorAll (-1), nop (0), all (1), restartNetwork (10000), restartHttps (10001)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.3.4.2

9.0.2.5 rpcDriver

9.0.2.5.1 rpcDrvTable

RPC driver module

<i>Status</i>	current
<i>Range</i>	0 - 1
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.3.6.1

9.0.2.5.2 rpcDrvName

Name of the Radio Device

Applies to AP.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.3.6.1.1.2

9.0.2.5.3 rpcDrvDfsSimulateRadar

Simulate Radar Detection on the Current Channel

Applies to AP.

<i>Enumeration</i>	nop (0), fire (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.3.6.1.1.5

9.0.3 settings

9.0.3.1 setConfiguration

9.0.3.1.1 setCfgFileUrl

The configuration file URL defines to or from which location the configuration file will be exported or imported. At the moment only TFTP protocol is supported.

Example:

- tftp://192.168.1.1/device.cfg

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.4.1.1

9.0.3.2 setWireless

9.0.3.2.1 setWlanDeviceTable

Wireless Hardware Modules

<i>Status</i>	current
<i>Range</i>	0 - 1
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.4.3.1

9.0.3.2.2 setWlanDevName

Name of the Wireless Device

Applies to AP and STA.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.4.3.1.1.2

9.0.3.2.3 setWlanDevRfOutput

RF Output

- **interfaceDisabled(-1)** The interface is disabled. It is not possible to change this setting.
- **rfOutputOff(0)** Set the wlan interface down, stop transmitting.
- **rfOutputOn(1)** Set the wlan interface up, start transmitting.

Applies to AP and STA. 802.11n products only.

<i>Enumeration</i>	interfaceDisabled (-1), rfOutputOff (0), rfOutputOn (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.4.3.1.1.3

9.0.3.2.4 setWlanDevFrequency

Wireless Frequency in MHz

Applies to AP and STA. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.4.3.1.1.6

9.0.3.2.5 setWlanDevPower

Wireless Output Power

Output power as effective isotropic radiated power (EIRP) in dBm including antenna gain.

Applies to AP and STA. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.4.3.1.1.8

9.0.3.2.6 setWlanDbgTable

Wireless Handoff Debug Parameters

<i>Status</i>	current
<i>Range</i>	0 - 63
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.4.3.6

9.0.3.2.7 setWlanDbgRatelimit

Volatile setting to enable/disable the rate limiter messages in standard syslog.

These log messages are subject to change. DO NOT PARSE!

Applies to STA. 802.11n products only.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.4.3.6.1.10

9.0.3.2.8 setWlanDbgBeacontsf

Volatile setting to enable/disable the Beacon RSSI messages in standard syslog. The TS field contains the internal TSF (mactime) instead of the system uptime.

These log messages are subject to change. DO NOT PARSE!

Applies to STA. 802.11n products only.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.4.3.6.1.12

9.0.3.2.9 setWlanDbgRange

Volatile setting to enable/disable the distance (range) measurement messages in standard syslog.

Note: Distance value is not in meters.

These log messages are subject to change. DO NOT PARSE!

Applies to STA. 802.11n products only.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.4.3.6.1.13

9.0.3.2.10 setWlanDbgReports

Volatile setting to enable/disable the periodical WLAN debug data reporting in standard syslog.

These log messages are subject to change. DO NOT PARSE!

Applies to STA.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.4.3.6.1.14

9.0.3.2.11 setWlanDbgIfaceName

Name of the virtual wireless interface.

Applies to AP and STA. 802.11n products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.4.3.6.1.2

9.0.3.2.12 setWlanDbgHandoff

Volatile setting to enable/disable the handoff trap.

These log messages are subject to change. DO NOT PARSE!

Applies to STA. 802.11n products only.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.4.3.6.1.3

9.0.3.2.13 setWlanDbgScan

Volatile setting to enable/disable the scan messages in standard syslog.

These log messages are subject to change. DO NOT PARSE!

Applies to STA. 802.11n products only.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.4.3.6.1.4

9.0.3.2.14 setWlanDbgMlme

Volatile setting to enable/disable the MLME messages in standard syslog.

These log messages are subject to change. DO NOT PARSE!

Applies to STA. 802.11n products only.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.4.3.6.1.5

9.0.3.2.15 setWlanDbgEvents

Volatile setting to enable/disable the events messages in standard syslog.

These log messages are subject to change. DO NOT PARSE!

Applies to STA. 802.11n products only.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.4.3.6.1.6

9.0.3.2.16 setWlanDbgBeaconrssi

Volatile setting to enable/disable the Beacon RSSI messages in commissioning syslog.

These log messages are subject to change. DO NOT PARSE!

Applies to STA. 802.11n products only.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.4.3.6.1.7

9.0.3.2.17 setWlanDbgAckrssi

Volatile setting to enable/disable the ACK RSSI messages in standard syslog.

These log messages are subject to change. DO NOT PARSE!

Applies to STA. 802.11n products only.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.4.3.6.1.8

9.0.3.2.18 setWlanDbgBeaconfiltered

Volatile setting to enable/disable the Beacon filtered RSSI messages in commissioning syslog.

These log messages are subject to change. DO NOT PARSE!

Applies to STA. 802.11n products only.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.4.3.6.1.9

9.0.3.3 setFirmware

9.0.3.3.1 setFwFileUrl

Download Firmware From This URL

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.4.5.1

9.0.3.3.2 setFwKeepConfig

Try to Import Configuration From the Previous Firmware Version

<i>Enumeration</i>	reset (0), keep (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.4.5.2

9.0.3.4 setCertificate

9.0.3.4.1 setCrtFileUrl

Download/upload URL for HTTPS/802.1x certificats and keys

The certification/key (for HTTPS/802.1X) file-URL defines the location of the certification/key file where it will be downloaded from or uploaded to. At the moment only the TFTP protocol is supported.

Example:

- tftp://192.168.1.1/uttpd.crt

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.4.6.1

9.0.3.4.2 setCrtFileSelector

Set this field to select which file should be downloaded/uploaded via the rpcCrtFile.

- **0** means legacy HTTPS. Upload/download certificate and key from the same url with a .crt and .key extension.
- **1** means HTTPS certificate
- **2** means HTTPS key
- **100** means certificate for wlan0, 101 for wlan1 and so on

- **200** means private key for wlan0, 201 for wlan1 and so on
- **300** means CA certificate for wlan0, 301 for wlan1 and so on
- **400** means CA2 certificate for wlan0, 401 for wlan1 and so on
- **500** means CRL certificate for wlan0, 501 for wlan1 and so on
- **600** means CRL2 certificate for wlan0, 601 for wlan1 and so on
- **700** means PKCS#12 container for wlan0, 701 for wlan1 and so on
- **10xxx** means upload/download the 'next' file which will be installed during next reboot or by calling `rpcCrtRefresh`. So a value of **10203** means next private key for wlan3.
- **20100** means certificate for ovpn0, 20101 for ovpn1 and so on
- **20200** means private key for ovpn0, 20201 for ovpn1 and so on
- **20300** means CA certificate for ovpn0, 20301 for ovpn1 and so on
- **20400** means static key for ovpn0, 20401 for ovpn1 and so on

To disable the CA certificate verification on 802.1X write the following string into the ca file (e.g. wlan0ca(300)):

DISABLE CA CERTIFICATE VERIFICATION. THIS WILL ALLOW A 3RD PARTY TO CAPTURE MY PASSWORD.

For static keys the `setCrtFileFormat` will be ignored so they must be in PEM format.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	0 - 90000
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.4.6.2

9.0.3.4.3 setCrtFileFormat

Set the certificate/key file format for `rpcCrtFile` actions

- **0** means the imported/exported certificate/key will be in the PEM format.
- **1** means the imported/exported certificate/key will be in the DER format.

Applies to AP and STA.

<i>Enumeration</i>	pem (0), der (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.4.6.3

9.0.3.4.4 setCrtFilePkcs12Passphrase

Set the PKCS#12 passphrase used to import with `rpcCertFile`

This passphrase will be used during the import to decrypt the container data. The `cfgWlan802dot1xClientKeyPa` will then be used to encrypt the client key on the device.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.4.6.4

9.0.3.5 setSystem

9.0.3.5.1 setSysTime

System time as epoch

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.4.7.1

9.0.3.6 setTechPreview

9.0.3.6.1 setTechPreviewEnabled

Technical Preview disabled or enabled

The Technical Preview allows access to upcoming features that are not yet officially released.

Note: This parameter is volatile and is lost after a reboot.

Applies to AP and STA.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.4.8.1

9.0.4 hardware

9.0.4.1 hwSystem

9.0.4.1.1 hwSysProduct

Product Type

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.5.1.1

9.0.4.1.2 hwSysSerial

Serial Number of the Product

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.5.1.2

9.0.4.1.3 hwSysRevision

ERP Revision of the Product

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.5.1.3

9.0.4.1.4 hwSysVersion

Version of the Product

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.5.1.4

9.0.4.2 hwBaseBoard

9.0.4.2.1 hwBbType

Product Type of the Base Board

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	0 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.5.10.1

9.0.4.2.2 hwBbSerial

Serial Number of the Base Board

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	0 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.5.10.2

9.0.4.2.3 hwBbRevision

ERP Revision of the Base Board

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.5.10.3

9.0.4.2.4 hwBbVersion

Version of the Base Board

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.5.10.4

9.0.4.2.5 hwBbPcbld

Hardware Assembly ID

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.5.10.5

9.0.4.2.6 hwBbAssemblyId

Hardware Assembly ID

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.5.10.6

9.0.4.3 hwIfaceBoard

9.0.4.3.1 hwIfaceBrdAssembled

Interface Board Present or Not

<i>Enumeration</i>	inexistent (0), present (1)
<i>Access</i>	readonly
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.5.11.1

9.0.4.3.2 hwIfaceBrdType

Product Type of the Interface Board

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	0 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.5.11.2

9.0.4.3.3 hwIfaceBrdSerial

Serial Number of the Interface Board

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	0 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.5.11.3

9.0.4.3.4 hwlfBrdRevision

ERP Revision of the Interface Board

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.5.11.4

9.0.4.3.5 hwlfBrdVersion

Version of the Interface Board

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.5.11.5

9.0.4.3.6 hwlfBrdPcbld

Hardware Assembly ID

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.5.11.6

9.0.4.3.7 hwlfBrdAssemblyld

Hardware Assembly ID

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.5.11.7

9.0.4.4 hwNetwork

9.0.4.4.1 hwNetEthernetTable

Ethernet Network Interfaces

<i>Status</i>	current
<i>Range</i>	0 - 2
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.5.2.1

9.0.4.4.2 hwNetEthName

Name of the Ethernet Interface

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.5.2.1.1.2

9.0.4.4.3 hwNetEthAssembled

Ethernet Interface Present or Not

<i>Enumeration</i>	inexistent (0), present (1)
<i>Access</i>	readonly
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.5.2.1.1.3

9.0.4.4.4 hwNetEthMacAddress

Ethernet MAC Address

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.5.2.1.1.4

9.0.4.4.5 hwNetEthOperation

Ethernet Interface Plugged or Unplugged

<i>Enumeration</i>	down (0), up (1)
<i>Access</i>	readonly
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.5.2.1.1.5

9.0.4.4.6 hwNetEthSpeed

Ethernet Speed in Mbps

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.5.2.1.1.6

9.0.4.4.7 hwNetEthHwIndex

Index of MAC-Address / Interface

The physical address of the Ethernet interface of the base board, since not all products are assembled the same way this is to describe how the wiring is done.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.5.2.1.1.7

9.0.4.5 hwSensor

9.0.4.5.1 hwSensorTable

Hardware sensor information table.

<i>Status</i>	current
<i>Range</i>	0 - 15
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.5.21.1

9.0.4.5.2 hwSensorName

Name of the Hardware Sensor

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.5.21.1.1.2

9.0.4.5.3 hwSensorUnit

Unit of the Hardware Sensor

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.5.21.1.1.3

9.0.4.5.4 hwSensorValue

Value of the Hardware Sensor

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.5.21.1.1.4

9.0.4.6 hwWireless

9.0.4.6.1 hwWlanDeviceTable

Hardware information of the wireless LAN Devices

<i>Status</i>	current
<i>Range</i>	0 - 1
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.5.3.1

9.0.4.6.2 hwWlanDevAntennaProfileId

Antenna Profile ID

Please check the user manual for antenna details.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.5.3.1.1.10

9.0.4.6.3 hwWlanDevAntennaGain

Antenna Gain in dBi

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.5.3.1.1.11

9.0.4.6.4 hwWlanDevCableLoss

Cable Loss in dB

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.5.3.1.1.12

9.0.4.6.5 hwWlanDevAssembled

Wireless Device Present or Not

<i>Enumeration</i>	inexistent (0), present (1)
<i>Access</i>	readonly
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.5.3.1.1.2

9.0.4.6.6 hwWlanDevType

Type of the Wireless Device

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.5.3.1.1.3

9.0.4.6.7 hwWlanDevSerial

Serial Number / Customer Field

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.5.3.1.1.4

9.0.4.6.8 hwWlanDevRevision

ERP Revision of the RF Board

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.5.3.1.1.5

9.0.4.6.9 hwWlanDevVersion

Version of the RF board

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.5.3.1.1.6

9.0.4.6.10 hwWlanDevPcbId

Hardware Assembly ID

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.5.3.1.1.7

9.0.4.6.11 hwWlanDevAssemblyId

Hardware Assembly ID

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.5.3.1.1.8

9.0.4.6.12 hwWlanDevMacAddress

MAC Address

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.5.3.1.1.9

9.0.4.6.13 hwWlanGlobal

9.0.4.6.13.1 hwWlanGblRegulatoryRegionId

Regulatory Region ID

Applies to AP and STA.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.5.3.2.2

9.0.4.7 hwCellular

9.0.4.7.1 hwCellAssembled

Cellular Module Assembled

<i>Enumeration</i>	inexistent (0), present (1)
<i>Access</i>	readonly
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.5.50.1

9.0.4.7.2 hwCellType

Cellular Module Type

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	0 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.5.50.2

9.0.4.7.3 hwCellSerial

Cellular Module Serial Number

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	0 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.5.50.3

9.0.4.7.4 hwCellImei

Cellular Module IMEI Number

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	0 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.5.50.4

9.0.4.8 hwGnss

9.0.4.8.1 hwGnssAssembled

GNSS Module Assembled

<i>Enumeration</i>	inexistent (0), present (1)
<i>Access</i>	readonly
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.5.51.1

9.0.4.8.2 hwGnssType

GNSS Module Type

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	0 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.5.51.2

9.0.4.8.3 hwGnssSerial

GNSS Module Serial Number

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	0 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.5.51.3

9.0.5 software

9.0.5.1 swFirmware

9.0.5.1.1 swFwName

Firmware Name

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.2.1

9.0.5.1.2 swFwVersion

Firmware Version

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.2.2

9.0.5.1.3 swFwRevision

Firmware Revision

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.2.3

9.0.5.2 swBootloader

9.0.5.2.1 swBootName

Name of the Bootloader

Applies to AP and STA.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.20.1

9.0.5.2.2 swBootVersion

Version of the Bootloader

Applies to AP and STA.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.20.2

9.0.5.2.3 swBootBuildDate

Date when the Bootloader was Built

Applies to AP and STA.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.20.3

9.0.5.3 swSystem

9.0.5.3.1 swSysRebootReason

System Reboot Reason

<i>Enumeration</i>	coldstart (0), warmstart (1), watchdog (2), oops (3), unknown (9)
<i>Access</i>	readonly
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.3.1

9.0.5.3.2 swSysMessageTable

System messages (e.g. Errors, Warnings)

<i>Status</i>	current
<i>Range</i>	0 - 63
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.3.100

9.0.5.3.3 swSysMsgPriority

Message Priority/Level

<i>Enumeration</i>	emergency (0), alert (1), critical (2), error (3), warning (4), notice (5), info (6), debug (7)
<i>Access</i>	readonly
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.3.100.1.2

9.0.5.3.4 swSysMsgCode

Message Code

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.3.100.1.3

9.0.5.3.5 swSysMsgText

Message

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.3.100.1.4

9.0.5.3.6 swSysBootStatus

The boot status indicates whether the booting sequence of the device has been completed.

<i>Enumeration</i>	done (0), booting (1)
<i>Access</i>	readonly
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.3.2

9.0.5.4 swConfiguration

9.0.5.4.1 swCfgChangesCount

Number of not yet Applied Device Configuration Changes

Applies to AP and STA.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.30.1

9.0.5.5 swOperatingSystem

9.0.5.5.1 swOsName

Operating System Name

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.4.1

9.0.5.5.2 swOsVersion

Operating System Version

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.4.2

9.0.5.5.3 swOsRevision

Operating System Revision

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.4.3

9.0.5.5.4 swOsUptime

Uptime of the Operating System

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	TimeTicks
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.4.4

9.0.5.6 swDriver

9.0.5.6.1 swDrvDfsTable

DFS Driver Statistics

<i>Status</i>	current
<i>Range</i>	0 - 1
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.1

9.0.5.6.2 swDrvDfsName

Name of The Wireless Device

Applies to AP and STA. 802.11n products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.1.1.2

9.0.5.6.3 swDrvDfsPulsesDetected

Pulses Detected by The Wireless Device

Applies to AP. 802.11n products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.1.1.3

9.0.5.6.4 swDrvDfsPulsesProcessed

Pulses Processed by The Wireless Device

Applies to AP. 802.11n products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.1.1.4

9.0.5.6.5 swDrvDfsRadarDetected

Radar Sequences Detected by The Wireless Device

Applies to AP. 802.11n products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.1.1.5

9.0.5.6.6 swDrvCntWlanMacTable

Wireless MAC-Layer Statistics

<i>Status</i>	current
<i>Range</i>	0 - 63
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.4

9.0.5.6.7 swDrvCntWlanMacRxHandlersDrop

MAC Debug Entry

Applies to AP and STA. 802.11n products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.4.1.10

9.0.5.6.8 swDrvCntWlanMacRxHandlersQueued

MAC Debug Entry

Applies to AP and STA. 802.11n products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.4.1.11

9.0.5.6.9 swDrvCntWlanMacRxHandlersDropNullfunc

MAC Debug Entry

Applies to AP and STA. 802.11n products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.4.1.12

9.0.5.6.10 swDrvCntWlanMacRxHandlersDropDefrag

MAC Debug Entry

Applies to AP and STA. 802.11n products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.4.1.13

9.0.5.6.11 swDrvCntWlanMacRxHandlersDropShort

MAC Debug Entry

Applies to AP and STA. 802.11n products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.4.1.14

9.0.5.6.12 swDrvCntWlanMacTxExpandSkbHead

MAC Debug Entry

Applies to AP and STA. 802.11n products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.4.1.15

9.0.5.6.13 swDrvCntWlanMacTxExpandSkbHeadCloned

MAC Debug Entry

Applies to AP and STA. 802.11n products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.4.1.16

9.0.5.6.14 swDrvCntWlanMacRxExpandSkbHead

MAC Debug Entry

Applies to AP and STA. 802.11n products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.4.1.17

9.0.5.6.15 swDrvCntWlanMacRxExpandSkbHead2

MAC Debug Entry

Applies to AP and STA. 802.11n products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.4.1.18

9.0.5.6.16 swDrvCntWlanMacRxHandlersFragments

MAC Debug Entry

Applies to AP and STA. 802.11n products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.4.1.19

9.0.5.6.17 swDrvCntWlanMacName

Name of The Wireless Device

Applies to AP and STA. 802.11n products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.4.1.2

9.0.5.6.18 swDrvCntWlanMacTxstatusDrop

MAC Debug Entry

Applies to AP and STA. 802.11n products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.4.1.20

9.0.5.6.19 swDrvCntWlanMacTxHandlersDrop

MAC Debug Entry

Applies to AP and STA. 802.11n products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.4.1.3

9.0.5.6.20 swDrvCntWlanMacTxHandlersQueued

MAC Debug Entry

Applies to AP and STA. 802.11n products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.4.1.4

9.0.5.6.21 swDrvCntWlanMacTxHandlersDropUnencrypted

MAC Debug Entry

Applies to AP and STA. 802.11n products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.4.1.5

9.0.5.6.22 swDrvCntWlanMacTxHandlersDropFragment

MAC Debug Entry

Applies to AP and STA. 802.11n products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.4.1.6

9.0.5.6.23 swDrvCntWlanMacTxHandlersDropWep

MAC Debug Entry

Applies to AP and STA. 802.11n products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.4.1.7

9.0.5.6.24 swDrvCntWlanMacTxHandlersDropNotAssoc

MAC Debug Entry

Applies to AP and STA. 802.11n products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.4.1.8

9.0.5.6.25 swDrvCntWlanMacTxHandlersDropUnauthPort

MAC Debug Entry

Applies to AP and STA. 802.11n products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.4.1.9

9.0.5.6.26 swDrvCntWlanWmmTable

WMM statistics

<i>Status</i>	current
<i>Range</i>	0 - 4
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.6

9.0.5.6.27 swDrvCntWlanWmmName

Name of The Queue

Applies to AP and STA. 802.11n products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.6.1.2

9.0.5.6.28 swDrvCntWlanWmmTx

Number of Frames Sent in his Queue

Applies to AP and STA. 802.11n products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.6.1.3

9.0.5.6.29 swDrvCntWlanWmmRx

Number of Frames Received in This Queue

Applies to AP and STA. 802.11n products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.6.1.4

9.0.5.6.30 swDrvCntWlanWmmShortRetries

Number of Retries for Frames Shorter Than RTS

Applies to AP and STA. 802.11n products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.6.1.5

9.0.5.6.31 swDrvCntWlanWmmLongRetries

Number of Retries for Frames Longer Than RTS

Applies to AP and STA. 802.11n products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.6.1.6

9.0.5.6.32 swDrvCntWlanWmmExceededRetries

Number of Failed Transmissions Due to Exceeding of The Retry Limit

Applies to AP and STA. 802.11n products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.6.1.7

9.0.5.6.33 swDrvConStatWlanIf

Volatile wlan interface selector for swDrvConStatTable.

Changes made here will be lost upon reconfiguration or a reboot. Use `cfgWlanGlblConnectionStatusWlanInt` to set a persistent value which is used during initialisation.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	4 - 17
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.7

9.0.5.6.34 swDrvConStatTable

Connection Status Information.

<i>Status</i>	current
<i>Range</i>	0 - 9
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.8

9.0.5.6.35 swDrvConStatTxBrType

Station TX Bitrate Type

Applies to AP and STA.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.8.1.10

9.0.5.6.36 swDrvConStatTxBrValue

Station TX Bitrate Value in Mbps

Multiplied by 10 if `swDrvConStatRxBrType` is 'legacy' or '11AC'.

Applies to AP and STA.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.8.1.11

9.0.5.6.37 `swDrvConStatTxBytes`

Station TX Bytes

Applies to AP and STA. 802.11n products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.8.1.12

9.0.5.6.38 `swDrvConStatTxPackets`

Station TX Packets

Applies to AP and STA. 802.11n products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.8.1.13

9.0.5.6.39 `swDrvConStatSigChain0`

Station Signal Chain 0 in dBm

Applies to AP and STA. 802.11n products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.8.1.14

9.0.5.6.40 `swDrvConStatSigChain1`

Station Signal Chain 1 in dBm

Applies to AP and STA. 802.11n products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.8.1.15

9.0.5.6.41 swDrvConStatSigChain2

Station Signal Chain 2 in dBm

Applies to AP and STA. 802.11n products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.8.1.16

9.0.5.6.42 swDrvConStatSigAvgChain0

Station Signal Average Chain 0 in dBm

Applies to AP and STA. 802.11n products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.8.1.17

9.0.5.6.43 swDrvConStatSigAvgChain1

Station Signal Average Chain 1 in dBm

Applies to AP and STA. 802.11n products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.8.1.18

9.0.5.6.44 swDrvConStatSigAvgChain2

Station Signal Average Chain 2 in dBm

Applies to AP and STA. 802.11n products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.8.1.19

9.0.5.6.45 swDrvConStatWlanName

WLAN Interface Name

Applies to AP and STA.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	4 - 5
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.8.1.2

9.0.5.6.46 swDrvConStatTxRetries

Station TX Retries

Applies to AP and STA.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.8.1.20

9.0.5.6.47 swDrvConStatTxFailed

Station TX Failed

Applies to AP and STA. 802.11n products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.8.1.21

9.0.5.6.48 swDrvConStatCacheNo

Station Dump Cache Access Number

The cache gets refreshed if it is older than 5 seconds.

Applies to AP and STA. 802.11n products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.8.1.22

9.0.5.6.49 swDrvConStatSigCombined

Station Signal Combined of All Active Chains in dBm

Applies to AP and STA.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.8.1.23

9.0.5.6.50 swDrvConStatSigAvgCombined

Station Signal Average Combined of All Active Chains in dBm

Applies to AP and STA.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.8.1.24

9.0.5.6.51 swDrvConStatMacName

Station MAC Address

In AP mode this is the MAC address of the connected client (STA). In client (STA) mode this is the MAC address of the AP to which the client is connected.

Applies to AP and STA.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 17
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.8.1.3

9.0.5.6.52 swDrvConStatRxBRExtra

Station RX Bitrate Extra

Applies to AP and STA.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.8.1.4

9.0.5.6.53 swDrvConStatRxBrType

Station RX Bitrate Type

Applies to AP and STA.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.8.1.5

9.0.5.6.54 swDrvConStatRxBrValue

Station RX Bitrate Value in Mbps

Multiplied by 10 if swDrvConStatRxBrType is 'legacy' or '11AC'.

Applies to AP and STA.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.8.1.6

9.0.5.6.55 swDrvConStatRxBytes

Station RX Bytes

Applies to AP and STA. 802.11n products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.8.1.7

9.0.5.6.56 swDrvConStatRxPackets

Station RX Packets

Applies to AP and STA. 802.11n products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.8.1.8

9.0.5.6.57 swDrvConStatTxBrExtra

Station TX Bitrate Extra

Applies to AP and STA.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.8.1.9

9.0.5.6.58 swDrvCntWlanTable

Wireless Dev Counters

<i>Status</i>	current
<i>Range</i>	0 - 63
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.9

9.0.5.6.59 swDrvCntWlanChannelActive

Time in ms during which the device has been on the current channel.

On STA when it's no associated the current channel is the lowest channel of selected country code.

Applies to AP and STA. 802.11n products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Counter32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.9.1.10

9.0.5.6.60 swDrvCntWlanChannelBusy

Time in ms during which the medium has been considered busy on the current channel.

Applies to AP and STA. 802.11n products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Counter32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.9.1.11

9.0.5.6.61 swDrvCntWlanChannelTransmit

Time in ms during which the device has been in transmit mode on the current channel.

Applies to AP and STA. 802.11n products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Counter32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.9.1.12

9.0.5.6.62 swDrvCntWlanChannelReceive

Time in ms during which the device has been in receive mode on the current channel.

Applies to AP and STA. 802.11n products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Counter32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.9.1.13

9.0.5.6.63 swDrvCntWlanChannelNoise

Measured channel noise in 1 dB steps.

Note: This value is not an absolute power level but the internal representation of the measured noise floor.

Applies to AP and STA. 802.11n products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.9.1.14

9.0.5.6.64 swDrvCntWlanName

Name of the wireless Dev

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.9.1.2

9.0.5.6.65 swDrvCntWlanEapAuthStartedFT

EAP sessions started through FT.

Applies to AP. 802.11n products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Counter32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.9.1.20

9.0.5.6.66 swDrvCntWlanEapAuthStartedFILS

EAP sessions started through FILS.

Applies to AP. 802.11n products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Counter32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.9.1.21

9.0.5.6.67 swDrvCntWlanEapAuthStartedPKMSA

EAP sessions started through PKMSA.

Applies to AP. 802.11n products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Counter32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.9.1.22

9.0.5.6.68 swDrvCntWlanAssocSuccess

Number of successful associations. 802.11n products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Counter32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.9.1.4

9.0.5.6.69 swDrvCntWlanAssocFailure

Number of unsuccessful associations. 802.11n products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Counter32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.9.1.5

9.0.5.6.70 swDrvCntWlanAssocFailureMaxSta

Number of times the maximum number of stations has been exceeded.

Applies to AP. 802.11n products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Counter32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.9.1.6

9.0.5.6.71 swDrvCntWlanNumAssocSta

Number of associated STA.

Applies to AP. 802.11n products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Counter32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.9.1.7

9.0.5.6.72 swDrvCntWlanEapAuthStarted

Number of EAP authentication sessions started since the AP has been started.

Applies to AP. 802.11n products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Counter32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.9.1.8

9.0.5.6.73 swDrvCntWlanEapAuthFailed

Number of EAP authentication sessions that have failed since the AP has been started.

Applies to AP. 802.11n products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Counter32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.5.9.1.9

9.0.5.7 swCellular

9.0.5.7.1 swCellTable

Cellular Status Information Table

This table is in a one-to-one relation to the `cfgNetWwanTable` where the indexes of the respective entries match.

Applies to cellular products only.

<i>Status</i>	current
<i>Range</i>	0 - 0
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.50.1

9.0.5.7.2 swCellConnectionStatus

Connection Status

- **unknown(0)**
- **disabled(1)**
- **simSwitching(2)**: Switching the SIM card.
- **netDown(3)**: Network is down.
- **resetting(4)**
- **netUp(5)**: Network is up.
- **connecting(6)**
- **connected(7)**
- **disconnected(8)**
- **invalid(9)**

Applies to cellular products only.

<i>Enumeration</i>	unknown (0), disabled (1), simSwitching (2), netDown (3), resetting (4), netUp (5), connecting (6), connected (7), disconnected (8), invalid (9)
<i>Access</i>	readonly
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.50.1.1.10

9.0.5.7.3 swCellConnectionMessage

Connection Status Message

Applies to cellular products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	0 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.50.1.1.11

9.0.5.7.4 swCellSignalType

Signal Type

unknown(0): Unknown signal type **wcdma(3)**: UMTS (3G) **lte(4)**: LTE (4G)

Applies to cellular products only.

<i>Enumeration</i>	unknown (0), wcdma (3), lte (4)
<i>Access</i>	readonly
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.50.1.1.12

9.0.5.7.5 swCellSignalRssi

Received Signal Strength Indication

Given in dB.

Applies to cellular products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.50.1.1.13

9.0.5.7.6 swCellSignalRsrq

Reference Signal Received Quality

Given in dB and only applies to signals of type **Ite(1)**.

Applies to cellular products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.50.1.1.14

9.0.5.7.7 swCellSignalRsrp

Reference Signal Received Power

Given in dBm and only applies to signals of type **Ite(1)**.

Applies to cellular products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.50.1.1.15

9.0.5.7.8 swCellSignalSinr

Signal-To-Interference-Plus-Noise Ratio

The signal-to-interference-plus-noise ratio (SINR) in dB is calculated from the value v of this entry using the following equation:

$$\text{SINR} = (v / 5) - 20$$

Example:

- **0**: SINR = -20dB
- **100**: SINR = 0dB
- **250**: SINR = 30dB

This only applies to signals of type **Ite(1)**.

Applies to cellular products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.50.1.1.16

9.0.5.7.9 swCellSignalRscp

Received Signal Code Power

Given in dBm and only applies to signals of type **wcdma(0)**.

Applies to cellular products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.50.1.1.17

9.0.5.7.10 swCellSignalEcio

Carrier to Noise Ratio

Given in dB and only applies to signals of type **wcdma(0)**.

Applies to cellular products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.50.1.1.18

9.0.5.7.11 swCellWwanName

Name of the Cellular Network Interface

This is name of the corresponding network interface.

Applies to cellular products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	4 - 5
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.50.1.1.2

9.0.5.7.12 swCellSimSlot

Active SIM Slot

- **slot1(1)**
- **slot2(2)**

Applies to cellular products only.

<i>Enumeration</i>	slot1 (1), slot2 (2)
<i>Access</i>	readonly
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.50.1.1.3

9.0.5.7.13 swCellSimStatus

SIM Status

This entry shows the status of the SIM card in the active slot:

noSim(0): No SIM card is inserted. **ready(1)**: The SIM card is ready for operation. **pinReq(2)**: Requesting the PIN of the SIM card. **pukReq(3)**: Requesting the PUK of the SIM card.

Applies to cellular products only.

<i>Enumeration</i>	noSim (0), ready (1), pinReq (2), pukReq (3)
<i>Access</i>	readonly
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.50.1.1.4

9.0.5.7.14 swCellSimPrimary

Is SIM Primary?

Is the SIM card of the primary slot used?

- **no(0)**
- **yes(1)**

Applies to cellular products only.

<i>Enumeration</i>	no (0), yes (1)
<i>Access</i>	readonly
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.50.1.1.5

9.0.5.7.15 swCellSimRoaming

Is SIM Roaming?

Is the active SIM card currently roaming?

- no(0)
- yes(1)

Applies to cellular products only.

<i>Enumeration</i>	no (0), yes (1)
<i>Access</i>	readonly
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.50.1.1.6

9.0.5.7.16 swCellServiceName

Service Provider Name

Applies to cellular products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	0 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.50.1.1.7

9.0.5.7.17 swCellServiceMcc

Mobile Country Code

Mobile Country Code (MCC) are defined by the ITU-T Recommendation E.212.

Applies to cellular products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.50.1.1.8

9.0.5.7.18 swCellServiceMnc

Mobile Network Code

Mobile Network Code (MNC) are defined by the ITU-T Recommendation E.212.

Applies to cellular products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.50.1.1.9

9.0.5.8 swRdm

9.0.5.8.1 swRdmMaxEirp

Maximal equivalent isotropically radiated power (EIRP) in dBm.

This value shows the maximal aggregated transmit power over all configured chains.

Applies to AP.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.6.1

9.0.5.8.2 swRdmMaxApp

Maximal antenna port power in dBm.

This value shows the maximal transmit power of a single chain.

Applies to AP.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.6.2

10 WESTERMO-SW6-BRIDGE-MIB

10.0.1 rstp

10.0.1.1 configuration

10.0.1.1.1 cfgRstpBridge

10.0.1.1.1.1 cfgRstpBridgeEnabled

Enable RSTP on bridge.

Applies to AP and STA.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.8.1.1.1

10.0.1.1.1.2 cfgRstpBridgePriority

The bridge's relative priority value for determining the root bridge (the upper 16 bits of the bridge-id). A bridge with the lowest bridge-id is elected the root. By default, the priority is 0x8000 (32768). This value needs to be a multiple of 4096, otherwise it's rounded to the nearest inferior one.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	0 - 61440
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.8.1.1.2

10.0.1.1.1.3 cfgRstpBridgeHelloTime

Hello message send interval

The interval in seconds between transmissions of hello messages by designated ports. With RSTP this value is 2 seconds.

Applies to AP and STA.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Range</i>	2 - 2
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.8.1.1.3

10.0.1.1.1.4 **cfgRstpBridgeForwardDelay**

The delay used by STP bridges to transition root and designated ports to forwarding. The default value is 15.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	4 - 30
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.8.1.1.4

10.0.1.1.1.5 **cfgRstpBridgeMaxAge**

The maximum age of the information transmitted by the bridge when it is the root bridge. The default value is 20.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	6 - 40
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.8.1.1.5

10.0.1.1.1.6 **cfgRstpBridgeTransmitHoldCount**

The transmit hold count used by the port transmit state machine to limit transmission rate. The default value is 6.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	1 - 10
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.8.1.1.6

10.0.1.1.2 **cfgRstpPort**

10.0.1.1.2.1 cfgRstpPortTable

RSTP port configuration table.

<i>Status</i>	current
<i>Range</i>	0 - 18
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.8.1.2.1

10.0.1.1.2.2 cfgRstpPortEnabled

Enable RSTP on port.

Applies to AP and STA.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.8.1.2.1.1.2

10.0.1.1.2.3 cfgRstpPortName

Port name to apply settings, e.g. eth0 or wlan0.

Applies to AP and STA.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.8.1.2.1.1.3

10.0.1.1.2.4 cfgRstpPortPriority

The port's relative priority value for determining the root port, in multiples of 16. By default, the port priority is 0x80 (128). Any value in the lower 4 bits is rounded off. The significant upper 4 bits become the upper 4 bits of the port-id. A port with the lowest port-id is elected as the root.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	0 - 240
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.8.1.2.1.1.4

10.0.1.1.2.5 cfgRstpPortPathCost

The port path cost. The port's contribution, when it is the root port, to the root path cost for the bridge.

By default the cost is automatically calculated from the port's speed. If -1 is defined, the port path cost is automatically calculated.

Data rate	RSTP cost (802.1W-2004, default value)
4 Mbit/s	5,000,000
10 Mbit/s	2,000,000
16 Mbit/s	1,250,000
100 Mbit/s	200,000
1 Gbit/s	20,000
2 Gbit/s	10,000
10 Gbit/s	2,000

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	-1 - 5000000
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.8.1.2.1.1.5

10.0.1.1.2.6 cfgRstpPortAutoEdge

The auto edge port parameter allows the automatic detection of edge ports.

Ports can be configured as edge ports to facilitate rapid changes to the forwarding state when connected to endpoints.

If enabled, the port will look for BPDUs; if there are none it begins forwarding packets. It is recommended to disable auto-edge for non-edge ports.

Applies to AP and STA.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.8.1.2.1.1.6

11 WESTERMO-SW6-FIREWALL-MIB

11.0.1 firewall

11.0.1.1 configuration

11.0.1.1.1 cfgFwEnabled

Firewall disabled or enabled.

Applies to AP and STA.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.1.1.1

11.0.1.1.2 cfgFwNat

11.0.1.1.2.1 cfgFwNatPortForwardTable

Firewall port forward rules table.

<i>Status</i>	current
<i>Range</i>	0 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.1.1.2.1

11.0.1.1.2.2 cfgFwNatPrtFwdDestinationPortEnd

Destination end port to redirect.

When forwarding multiple port, this value is the end of the range. Set to -1 if no range is forwarded. Can only be used with TCP, UDP or TCP/UDP.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	-1 - 65535
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.1.1.2.1.1.10

11.0.1.1.2.3 **cfgFwNatPrtFwdRedirectDestinationAddress**

Redirect traffic to this redirection destination address.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	IpAddress
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.1.1.2.1.1.11

11.0.1.1.2.4 **cfgFwNatPrtFwdRedirectDestinationPort**

Redirect traffic to this destination port.

Can only be used with TCP, UDP or TCP/UDP.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	-1 - 65535
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.1.1.2.1.1.12

11.0.1.1.2.5 **cfgFwNatPrtFwdEnabled**

Disable or enable the rule.

Applies to AP and STA.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.1.1.2.1.1.2

11.0.1.1.2.6 **cfgFwNatPrtFwdInterface**

Name of the network interface on which the rule applies.

Defines on which interface traffic is coming in. Groups of interfaces can be matched by adding the character '+' at the end. E.g. eth+ to match the interfaces eth0, eth1 and eth2. To match all interfaces

use the character '+' alone.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.1.1.2.1.1.3

11.0.1.1.2.7 **cfgFwNatPrtFwdProtocol**

Choose which IP protocol the rule matches.

Allowed protocols are:

- **any(0)**: Any ip protocol.
- **udp(1)**: Only UDP protocol.
- **tcp(2)**: Only TCP protocol.
- **udptcp(3)**: UDP and TCP protocol.

Applies to AP and STA.

<i>Enumeration</i>	any (0), udp (1), tcp (2), udptcp (3)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.1.1.2.1.1.4

11.0.1.1.2.8 **cfgFwNatPrtFwdSourceAddress**

Source address to match.

This can be a specific ip address or a range in CIDR notation. Set to 0.0.0.0/0 to match all inbound traffic. Set to 172.17.29.7/32 to match the specific IP 172.17.29.7 You can use ! to invert the sense of the rule: E.g. !192.168.0.0/24

Note: Usually you want 0.0.0.0/0.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	9 - 19
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.1.1.2.1.1.5

11.0.1.1.2.9 `cfgFwNatPrtFwdSourcePortStart`

Source start port to match.

Specify the port or start of a port range from which a connection originates. Can only be used with TCP, UDP or TCP/UDP. Leave this on -1 to disable. You can use ! to invert the sense of the rule: E.g. !80. When used in a range, the inversion applies to the range.

Note: Usually you want this disabled.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 6
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.1.1.2.1.1.6

11.0.1.1.2.10 `cfgFwNatPrtFwdSourcePortEnd`

Destination end port to match.

When matching multiple port, this value is the end of the range. Set to -1 if no range is to be matched. Can only be used with TCP, UDP or TCP/UDP.

Note: Usually you want this disabled.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	-1 - 65535
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.1.1.2.1.1.7

11.0.1.1.2.11 `cfgFwNatPrtFwdDestinationAddress`

Destination address to redirect.

This can be a specific ip address or a range in CIDR notation. Set to 0.0.0.0/0 to match all inbound traffic on the interface specified in `cfgFwNatPrtFwdInterface`. You can use ! to invert the sense of the rule: E.g. !192.168.0.0/24. When using static IPs set this to the configured address of the respective interface or alias you want to forward.

Be aware, that setting 0.0.0.0/0 will redirect everything arriving on the configured interface, even if not sent to the device itself.

Note: Leave this on 0.0.0.0/0 when using DHCP.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	9 - 19
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.1.1.2.1.1.8

11.0.1.1.2.12 **cfgFwNatPrtFwdDestinationPortStart**

Destination start port to redirect.

Specify the port or start of a port range for the destination. You can use ! to invert the sense of the rule: E.g. !80. When used in a range, the inversion applies to the range. Can only be used with TCP, UDP or TCP/UDP.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 20
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.1.1.2.1.1.9

11.0.1.1.2.13 **cfgFwNatOutboundTable**

Firewall outbound NAT rules table.

<i>Status</i>	current
<i>Range</i>	0 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.1.1.2.2

11.0.1.1.2.14 **cfgFwNatOutDestinationPortEnd**

Destination end port to match.

When forwarding multiple port, this value is the end of the range. Set to -1 if no range is forwarded. Can only be used with TCP, UDP or TCP/UDP.

Note: Usually you want this disabled.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	-1 - 65535
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.1.1.2.2.1.10

11.0.1.1.2.15 **cfgFwNatOutSourceRewriteAddress**

Set the address with which outbound traffic shall be rewritten.

In case you are using DHCP leave this on 0.0.0.0.

Note: If you are not rewriting the source to a specific aliases you can set this to 0.0.0.0 to automatically rewrite to the configured primary (first) address of the interface.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	IpAddress
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.1.1.2.2.1.11

11.0.1.1.2.16 **cfgFwNatOutSourceRewritePort**

Set the source-port with which outbound traffic shall be rewritten.

Can only be used with TCP, UDP or TCP/UDP. Set to -1 to disable source port rewrite.

Note: Usually you want this disabled.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	-1 - 65535
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.1.1.2.2.1.12

11.0.1.1.2.17 **cfgFwNatOutEnabled**

Disable or enable the rule.

Applies to AP and STA.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.1.1.2.2.1.2

11.0.1.1.2.18 **cfgFwNatOutInterface**

Name of the network interface on which the rule applies.

Matches traffic leaving on this interface. Needs to be set to an interface name if you are using DHCP. Set to -1 if you don't know on which interface traffic will be leaving. Match the traffic with `cfgFwNatOutDestinationAddress` instead. You can use `!` to invert the sense of the rule. E.g. `!wlan0`.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.1.1.2.2.1.3

11.0.1.1.2.19 **cfgFwNatOutProtocol**

Choose which IP protocol the rule matches.

Allowed protocols are:

- **any(0)**: Any ip protocol.
- **udp(1)**: Only UDP protocol.
- **tcp(2)**: Only TCP protocol.
- **udptcp(3)**: UDP and TCP protocol.

Applies to AP and STA.

<i>Enumeration</i>	any (0), udp (1), tcp (2), udptcp (3)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.1.1.2.2.1.4

11.0.1.1.2.20 **cfgFwNatOutSourceAddress**

Source address to match.

This can be a specific ip address or a range in CIDR notation. Set to `0.0.0.0/0` to match all inbound traffic. Set to `172.17.29.7/32` to match the specific IP `172.17.29.7`. You can use `!` to invert the sense of the rule: E.g. `!192.168.0.0/24`.

Note: Usually you want `0.0.0.0/0`.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	9 - 19
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.1.1.2.2.1.5

11.0.1.1.2.21 **cfgFwNatOutSourcePortStart**

Source start port to match.

Specify the port or start of a port range from which a connection originates. Can only be used with TCP, UDP or TCP/UDP. Leave this on -1 to disable. You can use ! to invert the sense of the rule: E.g. !80. When used in a range, the inversion applies to the range.

Note: Usually you want this disabled.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 6
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.1.1.2.2.1.6

11.0.1.1.2.22 **cfgFwNatOutSourcePortEnd**

Destination end port to match.

When matching multiple port, this value is the end of the range. Set to -1 if no range is to be matched. Can only be used with TCP, UDP or TCP/UDP.

Note: Usually you want this disabled.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	-1 - 65535
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.1.1.2.2.1.7

11.0.1.1.2.23 **cfgFwNatOutDestinationAddress**

Destination address to match.

This can be a specific ip address or a range in CIDR notation. Set to 0.0.0.0/0 to match all outbound traffic on the interface specified in `cfgFwNatOutInterface`. You can use ! to invert the sense of the

rule: E.g. !192.168.0.0/24.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	9 - 19
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.1.1.2.2.1.8

11.0.1.1.2.24 cfgFwNatOutDestinationPortStart

Destination start port to match.

Specify the port or start of a port range for the destination. Can only be used with TCP, UDP or TCP/UDP. You can use ! to invert the sense of the rule: E.g. !80. When used in a range, the inversion applies to the range.

Note: Usually you want this disabled.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 6
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.1.1.2.2.1.9

11.0.1.1.3 cfgFwL2IpFilter

11.0.1.1.3.1 cfgFwL2IpFilterEnabled

Globally enable or disable the L2 IP Filter option of all Bridges.

This filter will only apply on IP frames and will not touch anything else.

Applies to AP and STA.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.1.1.3.1

11.0.1.1.3.2 **cfgFwL2IpFilterDefaultAction**

Set the default action of all bridges when filtering is enabled.

Take care to not lock yourself out when the default action is 'drop'.

Applies to AP and STA.

<i>Enumeration</i>	accept (0), drop (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.1.1.3.2

11.0.1.1.3.3 **cfgFwL2IpFilterTable**

L2 IP Filter

<i>Status</i>	current
<i>Range</i>	0 - 63
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.1.1.3.3

11.0.1.1.3.4 **cfgFwL2IpFltrEnabled**

Rule disabled or enabled.

Applies to AP and STA.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.1.1.3.3.1.2

11.0.1.1.3.5 **cfgFwL2IpFltrBridge**

Bridge on which the rule will be applied.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	-1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.1.1.3.3.1.3

11.0.1.1.3.6 **cfgFwL2IpFltrAction**

Action to perform.

Applies to AP and STA.

<i>Enumeration</i>	accept (0), drop (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.1.1.3.3.1.4

11.0.1.1.3.7 **cfgFwL2IpFiltrPriority**

Filter priority

When multiple rules match, the rule with the highest priority will be applied.

NOTE: When multiple matching rules with the same priority exist, the rule which was first created will be used. This may lead to unexpected behaviour.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.1.1.3.3.1.5

11.0.1.1.3.8 **cfgFwL2IpFiltrSource**

Filter source

The source network/IP on which the rule matches (CIDR notation).

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	9 - 19
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.1.1.3.3.1.6

11.0.1.1.3.9 **cfgFwL2IpFiltrDestination**

Filter destination

The destination network/IP on which the rule matches (CIDR notation).

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	9 - 19
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.1.1.3.3.1.7

11.0.1.1.4 cfgFwFilter

11.0.1.1.4.1 cfgFwFitDefaultPolicyInput

The default filter policy on the input path.

Applies to AP and STA.

<i>Enumeration</i>	drop (0), accept (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.1.1.4.1

11.0.1.1.4.2 cfgFwFilterRulesTable

Firewall filter rules table.

<i>Status</i>	current
<i>Range</i>	0 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.1.1.4.10

11.0.1.1.4.3 cfgFwFitRSourcePortEnd

Source end port to match.

When matching multiple ports, this value is the end of the range. Can only be used with tcp or udp.

Set to -1 when no range is to be matched.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	-1 - 65535
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.1.1.4.10.1.10

11.0.1.1.4.4 cfgFwFitRDestinationAddress

Destination address to match.

This can be a specific ip address or a range in CIDR notation. Set to 0.0.0.0/0 to match all destinations. Set to 172.17.29.7/32 to match the specific IP 172.17.29.7. You can use ! to invert the sense of the rule: E.g. !192.168.0.0/24

Set to -1 to not use this parameter.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 20
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.1.1.4.10.1.11

11.0.1.1.4.5 cfgFwFitRDestinationPortStart

Destination start port to match.

Specify the port or start of a port range to which a connection is going. Can only be used with tcp or udp. You can use ! to invert the sense of the rule: E.g. !80. When used in a range, the inversion applies to the range.

Set to -1 to not use this parameter.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 20
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.1.1.4.10.1.12

11.0.1.1.4.6 cfgFwFitRDestinationPortEnd

Destination end port to match.

When matching multiple ports, this value is the end of the range. Can only be used with tcp or udp.

Set to -1 when no range is to be matched.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	-1 - 65535
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.1.1.4.10.1.13

11.0.1.1.4.7 **cfgFwFitREnabled**

Disable or enable the rule.

Applies to AP and STA.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.1.1.4.10.1.2

11.0.1.1.4.8 **cfgFwFitRChain**

Action to be performed.

Applies to AP and STA.

<i>Enumeration</i>	none (0), input (1), forward (2), output (3)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.1.1.4.10.1.3

11.0.1.1.4.9 **cfgFwFitRAction**

Action to be performed.

Applies to AP and STA.

<i>Enumeration</i>	drop (0), accept (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.1.1.4.10.1.4

11.0.1.1.4.10 **cfgFwFitRInputInterface**

Name of the input interface to match.

Groups of interfaces can be matched by adding the character '+' at the end. E.g. eth+ to match the interfaces eth0, eth1 and eth2. To match all interfaces use the character '+' alone. Set to -1 to not use this parameter.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 16
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.1.1.4.10.1.5

11.0.1.1.4.11 **cfgFwFitROutputInterface**

Name of the output interface to match.

Groups of interfaces can be matched by adding the character '+' at the end. E.g. eth+ to match the interfaces eth0, eth1 and eth2. To match all interfaces use the character '+' alone.

Set to -1 to not use this parameter.

This parameter is ignored for rules on the input chain.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 16
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.1.1.4.10.1.6

11.0.1.1.4.12 **cfgFwFitRProtocol**

Choose which IP protocol the rule matches.

For a list of the currently existing protocols see: https://en.wikipedia.org/wiki/List_of_IP_protocol_numbers
Some examples are:

- **any(0)**: Match any ip protocol
- **icmp(1)**
- **igmp(2)**
- **tcp(6)**
- **udp(17)**
- **gre(47)**
- **esp(50)**
- **ah(51)**
- **ospf(89)**
- **vrrp / carp(112)**
- **l2tp(115)**

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	0 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.1.1.4.10.1.7

11.0.1.1.4.13 **cfgFwFitRSourceAddress**

Source address to match.

This can be a specific ip address or a range in CIDR notation. Set to 0.0.0.0/0 to match all sources. Set to 172.17.29.7/32 to match the specific IP 172.17.29.7. You can use ! to invert the sense of the rule: E.g. !192.168.0.0/24

Set to -1 to not use this parameter.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 20
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.1.1.4.10.1.8

11.0.1.1.4.14 **cfgFwFitRSourcePortStart**

Source start port to match.

Specify the port or start of a port range from which a connection originates. Can only be used with tcp or udp. You can use ! to invert the sense of the rule: E.g. !80. When used in a range, the inversion applies to the range.

Set to -1 to not use this parameter.

Applies to AP and STA.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 20
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.1.1.4.10.1.9

11.0.1.1.4.15 **cfgFwFitDefaultPolicyForward**

The default filter policy on the forward path.

Applies to AP and STA.

<i>Enumeration</i>	drop (0), accept (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.1.1.4.2

11.0.1.1.4.16 **cfgFwFitDefaultPolicyOutput**

The default filter policy on the output path.

Applies to AP and STA.

<i>Enumeration</i>	drop (0), accept (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.1.1.4.3

12 WESTERMO-SW6-GNSS-MIB

12.0.1 gnss

12.0.1.1 configuration

12.0.1.1.1 cfgGnss

12.0.1.1.1.1 cfgGnssGpsd

cfgGnssGpsdEnabled

GPS Service Daemon Disabled or Enabled

If the GPS service daemon `gpsd` is **enabled(1)**, it makes location data available through TCP/IP.

Applies to cellular products only.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.10.1.1.1.1

cfgGnssGpsdAddress

Bind Address

The `gpsd` can only bind to the localhost or to all addresses. This means, that it will listen to all addresses, when this address is set to a value other than `127.0.0.1:<port>`.

The default bind address is `0.0.0.0:2974`.

Examples:

- **0.0.0.0:2974**
- **127.0.0.1:1234**

Applies to cellular products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.10.1.1.1.2

12.0.1.1.1.2 cfgGnssDevice

cfgGnssDevSatelliteSystems

Satellite Systems

This entry allows the selection of different satellite systems, whereby the following combinations are possible:

- **gps(1)**: GPS
- **glonass(2)**: Glonass
- **gpsGlo(3)**: GPS and Glonass
- **beidou(4)**: BeiDou
- **gpsBei(5)**: GPS and BeiDou
- **gloBei(6)**: Glonass and BeiDou
- **galileo(8)**: Galileo
- **gpsGal(9)**: GPS and Galileo
- **gloGal(10)**: Glonass and Galileo
- **gpsGloGal(11)**: GPS, Glonass and Galileo
- **beiGal(12)**: BeiDou and Galileo
- **gpsBeiGal(13)**: GPS, BeiDou and Glonass

This entry can also be interpreted as bitmask, with each bit reflecting a satellite system as follows:

- Bit 1: GPS
- Bit 2: Glonass
- Bit 3: BeiDou
- Bit 4: Galileo

However, only the combinations of the enumeration defined above are valid.

Applies to cellular products only.

<i>Enumeration</i>	gps (1), glonass (2), gpsGlo (3), beidou (4), gpsBei (5), gloBei (6), galileo (8), gpsGal (9), gloGal (10), gpsGloGal (11), beiGal (12), gpsBeiGal (13)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.10.1.1.2.1

cfgGnssDevMeasurementPeriod

Measurement Period

The measurement period T defines the time interval in milliseconds between two consecutive localisations. The minimal time is limited to 50 ms. Certain GNSS hardware may silently adjust the measurement period to even higher values.

The sampling rate f is calculated as follows:

$$f = 1000 / T$$

Examples:

- **10000**: $f = 0.1$ Hz
- **1000**: $f = 1$ Hz
- **100**: $f = 10$ Hz

Applies to cellular products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.10.1.1.2.2

cfgGnssDevMessages

cfgGnssDevMsgsNmeaTable

NMEA Sentences

The National Marine Electronics Association (NMEA) defined in the NMEA-0183 standard how data are transmitted in a sentences from one talker to multiple listeners.

Applies to cellular products only.

<i>Status</i>	current
<i>Range</i>	0 - 12
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.10.1.1.2.3.1

cfgGnssDevMsgsNmeaType

NMEA Sentence Type

The following NMEA sentence types are supported:

- GxGGA (61440)
- GxGLL (61441)
- GxGSA (61442)
- GxGSV (61443)
- GxRMC (61444)
- GxVTG (61445)
- GxGRS (61446)
- GxGST (61447)
- GxZDA (61448)
- GxGBS (61449)
- GxDTM (61450)
- GxGNS (61453)
- GxVLW (61455)

Applies to cellular products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.10.1.1.2.3.1.1.2

cfgGnssDevMsgsNmeaRate

Sentence Rate

The sentence rate r multiplies the measurement period T specified in `cfgGnssDevMeasurementPeriod` for the corresponding NMEA sentence type. It allows to define individual sampling rates for each sentence type or disabling the them by setting their rate to 0.

The individual sampling rate f is calculated as follows:

$$f = 1000 / (r * T)$$

Examples:

- **1** ($T = 1000$ ms): $f = 1$ Hz
- **2** ($T = 1000$ ms): $f = 0.5$ Hz
- **5** ($T = 100$ ms): $f = 2$ Hz

Applies to cellular products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.10.1.1.2.3.1.1.3

cfgGnssDevMsgsUbxTable

UBX Messages

UBX is a proprietary protocol from the company u-blox for exchanging GNSS data.

Applies to cellular products only.

<i>Status</i>	current
<i>Range</i>	0 - 127
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.10.1.1.2.3.2

cfgGnssDevMsgsUbxType

UBX Message Type

The following UBX message types are supported:

- NAV-POSECEF (257)
- NAV-STATUS (259)
- NAV-PVT (263)
- NAV-VELECEF(273)
- NAV-TIMEUTC (289)
- NAV-SAT (309)

Applies to cellular products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	0 - 65535
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.10.1.1.2.3.2.1.2

cfgGnssDevMsgsUbxRate

Message Rate

The message rate r multiplies the measurement period T specified in `cfgGnssDevMeasurementPeriod` for the corresponding UBX message type. It allows to define individual sampling rates for each message type or disabling the them by setting their rate to 0.

The individual sampling rate f is calculated as follows:

$$f = 1000 / (r * T)$$

Examples:

- 1 (T = 1000 ms): f = 1 Hz
- 2 (T = 1000 ms): f = 0.5 Hz
- 5 (T = 100 ms): f = 2 Hz

Applies to cellular products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.10.1.1.2.3.2.1.3

12.0.1.2 software

12.0.1.2.1 swGnss

12.0.1.2.1.1 swGnssTime

GNSS Time

The time is given in the Coordinated Universal Time (UTC) according to the ISO 8601 standard.

Example:

- 2020-06-19T10:55:50.000Z

Applies to cellular products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.10.2.1.1

12.0.1.2.1.2 swGnssMode

GNSS Navigation Mode

Possible navigation modes are:

- **na(0)**: Not available

- **none(1)**: None
- **twoD(2)**: 2D
- **threeD(3)**: 3D

Applies to cellular products only.

<i>Enumeration</i>	na (0), none (1), twoD (2), threeD (3)
<i>Access</i>	readonly
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.10.2.1.2

12.0.1.2.1.3 swGnssLon

GNSS Longitude

Longitude of the current location in decimal degrees (DD).

Example:

- 8.825446333

Applies to cellular products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.10.2.1.3

12.0.1.2.1.4 swGnssLat

GNSS Latitude

Latitude of the current location in decimal degrees (DD).

- 47.268953167

Applies to cellular products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.10.2.1.4



12.0.1.2.1.5 swGnssAlt
GNSS Altitude

Altitude of the current location in meters.

Applies to cellular products only.

<i>Access</i>	readonly
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.10.2.1.5

13 WESTERMO-SW6-ICL-MIB

13.0.1 icl

13.0.1.1 configuration

13.0.1.1.1 cfgIcl

13.0.1.1.1.1 cfgIclEnabled

Enable Inter-Carriage Link application.

Applies to AP. 802.11n products only.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.5.1.1.1

13.0.1.1.1.2 cfgIclConnectionDelay

Connection delay after a potential ICL partner was first detected.

This value in conjunction with cfgIclCycleTime defines how extensively a potential ICL partner is monitored and analyzed before a connection is established.

Applies to AP. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	0 - 600
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.5.1.1.2

13.0.1.1.1.3 cfgIclConnectionThreshold

This value defines the minimum signal level necessary for the ICL application to start evaluating a potential ICL partner.

Applies to AP. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	-90 - 0
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.5.1.1.3

13.0.1.1.1.4 **cfgIclDisconnectionDelay**

Disconnection delay in seconds defines how quickly a connected ICL pair resets to scanning mode after after the current ICL partner reaches a low signal level or gets disconnected.

Applies to AP. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	1 - 600
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.5.1.1.4

13.0.1.1.1.5 **cfgIclDisconnectionThreshold**

This value defines the minimum signal level necessary for a ICL pair to stay connected. If the signal level drops below this level for longer than in `cfgIclDisconnectionDelay` specified, the ICL application will revert the device do access point and resume scans for a new partner.

Applies to AP. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	-90 - 0
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.5.1.1.5

13.0.1.1.1.6 **cfgIclInterfaceName**

This value describes the interface the ICL Application will use for its services.

Applies to AP. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.5.1.1.6

13.0.1.1.1.7 cfgIclCycleTime

Interval of background scans in seconds.

This value in conjunction with cfgIclConnectionDelay defines how extensively a potential ICL partner is monitored and analyzed before a connection is established.

Applies to AP. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	2 - 60
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.5.1.1.7

13.0.1.1.1.8 cfgIclBlacklistTime

Duration of blacklisting in seconds.

Applies to AP. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	1 - 3600
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.5.1.1.8

13.0.1.1.1.9 cfgIclSuspended

Initial state of ICL when it starts up.

Applies to AP. 802.11n products only.

<i>Enumeration</i>	resumed (0), suspended (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.5.1.1.9

13.0.1.2 rpc

13.0.1.2.1 rpclcl

13.0.1.2.1.1 rpclclForceDisconnect

Force the device to disconnect from the current ICL partner and resume background scanning for a new partner.

<i>Enumeration</i>	nop (0), disconnect (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.5.3.1.1

13.0.1.2.1.2 rpclclClearBlacklist

Clear all currently blacklisted entries.

<i>Enumeration</i>	nop (0), clear (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.5.3.1.2

13.0.1.3 settings

13.0.1.3.1 setlcl

13.0.1.3.1.1 setlclSuspended

Suspend or resume ICL operation.

When suspended, ICL brings down the wireless interface and pauses operation. It remains silent until it is resumed.

<i>Enumeration</i>	resumed (0), suspended (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.5.4.1.1

13.0.1.4 software

13.0.1.4.1 swlcl

13.0.1.4.1.1 swlclStatus

Current status of ICL application.

- **scanning(1):** Scanning indicates Access Point mode with background scanning activated.
- **connected(2):** Connected indicates a connection with an ICL partner is established and background scanning is disabled.
- **suspended(3):** Suspended indicates that ICL is currently suspended.

<i>Enumeration</i>	disabled (0), scanning (1), connected (2), suspended (3)
<i>Access</i>	readonly
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.5.6.1.1

14 WESTERMO-SW6-NWM-MIB

14.0.1 nwm

14.0.1.1 configuration

14.0.1.1.1 cfgHttpReport

14.0.1.1.1.1 cfgHttpRprtServerUrl

URL of the remote server where DFS / IDF reports are sent to.

Format: http://<ipv4>:<port>/<path>

Applies to AP. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.3.1.1.1

14.0.1.1.2 cfgChannelManager

14.0.1.1.2.1 cfgChMgrEnabled

Enable the ChannelManager.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.3.1.3.1

14.0.1.1.2.2 cfgChMgrUsableFrequencyList

Configure which frequency list is to be used as 'list of usable frequencies' by the ChannelManager. A value of -1 means no list defined using all frequencies available in current country.

Applies to AP. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	-1 - 23
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.3.1.3.2

14.0.1.1.2.3 **cfgChMgrDfsUseNvram**

When 'enabled' the DFS states will be load / stored from / to the non-volatile ram device.

Applies to AP. 802.11n products only.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.3.1.3.3

14.0.1.1.3 **cfgNwm**

14.0.1.1.3.1 **cfgNwmEnabled**

Enable the Wireless Manager.

Applies to AP. 802.11n products only.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.3.1.4.1

14.0.1.1.4 **cfgldf**

14.0.1.1.4.1 **cfgldfEnabled**

Enable Interference Detection Function.

IDF is only supported for devices with two radios.

Applies to AP. 802.11n products only.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.3.1.5.1

14.0.1.1.4.2 **cfgldfScanWorkTable**

Table of scan works items.

Applies to AP. 802.11n products only.

<i>Status</i>	current
<i>Range</i>	0 - 31
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.3.1.5.10

14.0.1.1.4.3 **cfgldfScanWorkFreq**

Center frequency in MHz of the channel to scan.

Set center frequency to 0 to disable this scan work item and all following items.

Applies to AP. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	0 - 6100
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.3.1.5.10.1.2

14.0.1.1.4.4 **cfgldfScanWorkAction**

Scan Work Action of the interference function.

The following scan work actions are available:

none(0)

Scan work item and all following items of the table are ignored.

spectral(2)

Spectral data are collected from Antenna port 3 and spectral statistics is generated at the end of the scan work item. During a Spectral Scan Work the raw spectral data can be retrieved from the AP. Please refer to the Software User Manual for more information.

radar(3)

Runs the radar detection engine and counts all radar sequences detected by the monitor wireless device on Antenna port 3.

wifi(4)

Wifi data are collected from Antenna port 3 and wifi statistics is generated at the end of the scan work item.

Applies to AP. 802.11n products only.

<i>Enumeration</i>	none (0), spectral (2), radar (3), wifi (4)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.3.1.5.10.1.3

14.0.1.1.4.5 cfgldfScanWorkSeconds

Duration of the Scan Work Item in seconds.

At the end of the Scan Work a JSON formatted Report is generated and buffered. Set scan work time to 0 to disable this scan work item and all following items.

Applies to AP. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	0 - 86400
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.3.1.5.10.1.4

14.0.1.1.4.6 cfgldfInterval

Report interval in seconds.

This value defines the interval when all available Reports are sent to the URL defined in cfgldfHttpReportUrl.

Applies to AP. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	1 - 86400
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.3.1.5.3

14.0.1.1.4.7 cfgldfName

IDF Name.

Can be used to set an unique identifier for the reports.

Applies to AP. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.3.1.5.4

14.0.1.1.4.8 **cfgldfTrigger**

cfgldfTrigRadarCntTh

IDF trigger radar_count.

Defines the number of radar events to trigger a report.

Applies to AP. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.3.1.5.5.1

cfgldfTrigChanLoadTh

IDF trigger channel_load.

Defines the channel load ratio in percent to trigger a report.

Applies to AP. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	0 - 100
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.3.1.5.5.2

cfgldfTrigAlienLoadTh

IDF trigger alien_load.

Defines the ratio of alien channel load in percent to trigger a report.

Applies to AP. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	0 - 100
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.3.1.5.5.3

cfgIdfTrigDomLoadTh

IDF trigger domestic_load.

Defines the ratio of domestic channel load in percent to trigger a report.

Applies to AP. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	0 - 100
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.3.1.5.5.4

cfgIdfTrigAlienMaxRssiTh

IDF trigger alien_avg_rssi.

Defines the average RSSI of alien traffic to trigger a report.

Applies to AP. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	0 - 127
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.3.1.5.5.5

cfgIdfTrigDomMaxRssiTh

IDF trigger domestic_avg_rssi.

Defines the average RSSI of domestic traffic to trigger a report.

Applies to AP. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	0 - 127
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.3.1.5.5.6

14.0.1.1.5 cfgChannelCleaner

14.0.1.1.5.1 **cfgChanCleanEnabled**

Enable ChannelCleaner

Applies to AP. 802.11n products only.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.3.1.6.1

14.0.1.1.5.2 **cfgChanCleanUsableFrequencyList**

Configure which frequency list *cfgWlanFIndex* is to be used as 'list of usable frequencies' by ChannelCleaner.

Applies to AP. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	-1 - 23
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.3.1.6.10

14.0.1.1.5.3 **cfgChanCleanDfsUseNvram**

When 'enabled' the DFS states will be load/stored from/to the non-volatile ram device.

Applies to AP. 802.11n products only.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.3.1.6.9

14.0.1.1.6 **cfgAfm**

14.0.1.1.6.1 **cfgAfmEnabled**

Enable the Area Frequency Manager on the local device.

Applies to AP. 802.11n products only.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.3.1.7.1

14.0.1.1.6.2 cfgAfmRedundant

cfgAfmRedundantIp

IP address of the redundant Area Frequency Manager.

Set to '0.0.0.0' to disable.

Applies to AP. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	IpAddress
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.3.1.7.10.1

cfgAfmRedundantName

Name of the redundant Area Frequency Manager.

Applies to AP. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.3.1.7.10.2

cfgAfmName

Name of the Area Frequency Manager on the local device.

Applies to AP. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.3.1.7.2

cfgAfmNeighbourTable

A table of the two neighbour (adjacent) Area Frequency Manager.

Currently only up to two neighbour AFMs are supported.

The area index of the neighbour AFM must be less or greater than the index of this area.

Applies to AP. 802.11n products only.

<i>Status</i>	current
<i>Range</i>	0 - 4
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.3.1.7.20

cfgAfmNeighbourIp

IP address of the neighbour (adjacent) Area Frequency Manager.

Applies to AP. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	IpAddress
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.3.1.7.20.1.2

cfgAfmNeighbourName

Name of the neighbour (adjacent) Area Frequency Manager.

Applies to AP. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.3.1.7.20.1.3

cfgAfmIndex

Index of the Area which the local running Area Frequency Manager manages.

Applies to AP. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	Integer32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.3.1.7.3

cfgAfmAfcTable

A table of all Area Frequency Clients controlled by the Area Frequency Manager running on this device.

Applies to AP. 802.11n products only.

<i>Status</i>	current
<i>Range</i>	0 - 63
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.3.1.7.30

cfgAfmAfcName

Name of the Area Frequency Client to control.

Applies to AP. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.3.1.7.30.1.2

cfgAfmAfcIp

IP address of the Area Frequency Client to control.

Applies to AP. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	IpAddress
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.3.1.7.30.1.3

cfgAfmAreaSize

Number of segments controlled by this AFM.

Applies to AP. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	1 - 32
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.3.1.7.5

cfgAfmPrimary

Enable to set this AFM as primary in this area; set disable for stand-by.

Applies to AP. 802.11n products only.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.3.1.7.6

cfgAfmReportEnabled

When enabled(1) the AFM sends reports to the configured server URL (see `cfgIldfServerUrl`).

Please refer to Software 6 Interface Description for more information.

Applies to AP. 802.11n products only.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.3.1.7.7

cfgAfmVisibility

Visibility of segments

The visibility must be greater or equal as the absolute segment offset configured in any `cfgAfcNeighbourOffset` within the line.

`cfgAfmVisibility >= abs(cfgAfcNeighbourOffset)`

Applies to AP. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	1 - 1024
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.3.1.7.8

14.0.1.1.7 cfgAfc

14.0.1.1.7.1 cfgAfcEnabled

Enable the Area Frequency Client on the local device.

Applies to AP. 802.11n products only.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.3.1.8.1

14.0.1.1.7.2 **cfgAfcAfmTable**

A table of Area Frequency Managers which are allowed to control this Area Frequency Client (AFC).

Applies to AP. 802.11n products only.

<i>Status</i>	current
<i>Range</i>	0 - 4
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.3.1.8.10

14.0.1.1.7.3 **cfgAfcAfmName**

Name of the connecting Area Frequency Manager.

Applies to AP. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.3.1.8.10.1.2

14.0.1.1.7.4 **cfgAfcAfmIp**

IP address of the connecting Area Frequency Manager.

Applies to AP. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	IpAddress
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.3.1.8.10.1.3

14.0.1.1.7.5 **cfgAfcName**

Name of the Area Frequency Client on the local device.

Applies to AP. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.3.1.8.2

14.0.1.1.7.6 **cfgAfcIndex**

Index of the Area Frequency Client within the local Area.

Applies to AP. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	0 - 31
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.3.1.8.3

14.0.1.1.7.7 **cfgAfcNeighbourOffsetTable**

Table of up to 24 segment offsets relevant for neighbour list.

Applies to AP. 802.11n products only.

<i>Status</i>	current
<i>Range</i>	0 - 23
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.3.1.8.40

14.0.1.1.7.8 **cfgAfcNeighbourOffset**

Segment offset to be considered for neighbour list.

Applies to AP. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	-15 - 15
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.3.1.8.40.1.2

14.0.1.1.7.9 **cfgAfcBackupFreq**

Backup frequency which the Area Frequency Client will use if the Area Frequency Manager is not present.

Applies to AP. 802.11n products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Range</i>	0 - 6100
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.3.1.8.5

14.0.1.1.7.10 **cfgAfcReportEnabled**

When enabled(1) the AFC sends reports to the configured server URL (see `cfgHttpRprtServerUrl`).

Please refer to Software 6 Interface Description for more information.

Applies to AP. 802.11n products only.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.3.1.8.6

14.0.1.2 **rpc**

14.0.1.2.1 **rpcChannelManager**

14.0.1.2.1.1 **rpcChMgrHttpReport**

Requests a HTTP report from the Channel Manager.

Applies to AP. 802.11n products only.

<i>Enumeration</i>	nop (0), freqstate (1), channels (2)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.3.3.1.1

14.0.1.2.2 **rpcNwm**

14.0.1.2.2.1 **rpcNwmHttpReport**

Requests a HTTP report from the Wireless Manager.

Applies to AP. 802.11n products only.

<i>Enumeration</i>	nop (0), status (1), freqstate (2)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.3.3.2.1

14.0.1.2.3 rpcNvram

14.0.1.2.3.1 rpcNvramFreqStatesReset

Resets the state of DFS frequencies in the Available Channel List which is stored in non-volatile memory.

The device performs a reboot after resetting the state of the frequencies. The behaviour is similar to a factory reset but does only reset the Available Channel List.

Applies to AP. 802.11n products only.

<i>Enumeration</i>	reset (0)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.3.3.3.1

15 WESTERMO-SW6-PWN-MIB

15.0.1 pwn

15.0.1.1 configuration

15.0.1.1.1 cfgWireless

15.0.1.1.1.1 cfgWlanBandsteering

cfgWlanBsteerEnabled

Enable Band Steering.

Band steering is a technique used in dual-band (2G4 and 5G) wireless setups to encourage dual-band STAs to use the less-congested band.

Applies to AP. 802.11ac products only.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>Status</i>	current
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.9.1.1.1.1

cfgWlanBsteerMatchingSsid

List of Matching SSIDs.

This is a comma and/or space separated list of SSIDs. For each SSID, there shall be a matching SSID in the 2G4 and 5G band so that the band steering process is able to work and decide which STA shall be connected in which band.

Applies to AP. 802.11ac products only.

<i>Access</i>	readwrite
<i>Status</i>	current
<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.9.1.1.1.2

16 WebAPI Detailed Specification

- This appendix specifies the WebAPI interface.
- Top level URL for the API is `/cgi-bin/luci/api`

16.1 Authentication API

- Used for authentication (login / logout).
- Implemented as JSON RPC version 2.0 API with `/cgi-bin/luci/api/auth` as its top level URL.
- Further requests have to include the token as cookie with name "sysauth".

16.1.1 RPC Methods

16.1.1.1 Login

- method: `login(username, password)`

Example POST request

```
{
  "id": 1, "jsonrpc": "2.0",
  "method": "login",
  "params": {
    "username": "webadmin",
    "password": "10SoaL_98"
  }
}
```

Example response

```
{
  "id": 1, "jsonrpc": "2.0",
  "result": { "token": "4855d491f8dbde3628edc4e5be4861fe" }
}
```

16.1.1.2 Logout

- method: logout()

Example POST request:

```
{
  "id": 1, "jsonrpc": "2.0",
  "method": "logout"
}
```

Example response

```
{
  "id": 1, "jsonrpc": "2.0",
  "result": { "message": "Logout successful" }
}
```

16.2 Files API

- Top level URL for the Files API is /cgi-bin/luci/api/files
- Using multi part form protocol (i.e. Content-Type: multipart/form-data;)
- Additional form data might be supported
- When a system upgrade is running, any call to the Files API will result in an error -32003
 - Example: {"error": {"code": -32003, "message": "System upgrade running"}}

16.2.1 Device configuration file import / export

- URL: /cgi-bin/luci/api/files/config

- HTTP GET request for configuration file export
- Multi part HTTP POST request for configuration file import
 - Name field for configuration file part is `config`.
 - Optional form value `reset` (default: `false`). If `reset` is `true`, then the device configuration will be reset before importing.
 - if the import is successful, the answer will be a JSON object with the member "result" which is itself an object with the member "nof_changes" (number of changes).
 - Example: `{"result": {"nof_changes": "1"}}`
- if the import fails, the answer will be a JSON object with the member "error" which is itself an object with the members "code" (being -32001) and "message" (e.g. "Import failed")
 - Example: `"error": {"code": -32001, "message": "Import failed"}`

16.2.2 Syslog export

- URL: `/cgi-bin/luci/api/files/syslog`
- HTTP GET request for syslog export

16.2.3 System messages export

- URL: `/cgi-bin/luci/api/files/sys_messages`
- HTTP GET request for system message export

16.2.4 Support File export

- URL: `/cgi-bin/luci/api/files/support`
- HTTP GET request for support file export

16.2.5 Firmware upload and upgrade

- URL: `/cgi-bin/luci/api/files/firmware`
- Multi part HTTP POST request for firmware image upload and upgrade
 - Name fields for file parts are:
 - * `image` for the firmware image file
 - * `custom-config` for the custom configuration file
 - Optional form value `config_mode`
 - * `reset`: Reset to default configuration
 - * `keep`: Keep current configuration (this is the default)
 - * `custom`: Apply custom configuration after upgrade
 - Additional custom configuration file must be provided
 - if the uploaded firmware package is valid and the firmware upgrade is successfully started, the answer will be a JSON object with the member "result" which is itself an object with the member "message" (e.g. `{"result": {"message": "System upgrade started"}}`). At end of upgrade, the device will reboot. To see if upgrade is done: Poll [swOsUptime.0](#)
 - * It will count up until reboot,
 - * then timeout during reboot ("connection timeout, no route to host")
 - * and finally request new login (http status 403 "Forbidden") after reboot.
 - if the firmware upload and upgrade fails, the answer will be a JSON object with the member "error" which is itself an object with the members "code" (being -32002) and "message" (e.g. "Invalid firmware package")
 - * Example: `{"error": {"code": -32002, "message": "Invalid firmware package"}}`

16.3 Device Configuration API

- Implemented as JSON RPC version 2.0 API with `/cgi-bin/luci/api/mib` as it's top level URL.
- Same well-known interface as used by the SNMP and CLI API.

16.3.1 RPC Methods

16.3.1.1 RPC Methods Errors

Number	Category	Message
-1	General Error	E.g. "general error" or "set failed"
-2	Not Available	E.g. "element not available"

16.3.1.2 Get MIB items / device configuration

- method: `get(Array of ConfigurationItemKey)`
 - `ConfigurationItem`: Supported are all items which are defined in the available MIBs of the device.
- Note:
 - if `"params"` is not given, then return list of all available configuration items and it's values.

Example POST request

```
{
  "id": 1, "jsonrpc": "2.0",
  "method": "get",
  "params": {
    "items": [
      {"key": "WESTERMO-SW6-MIB::cfgSysHostname.0"},
      {"key": "WESTERMO-SW6-MIB::cfgSysTimezone.0"},
      {"key": "WESTERMO-SW6-MIB::swCfgChangesCount.0"},
      {"key": "WESTERMO-SW6-MIB::cfgSysHostname"}
    ]
  }
}
```

Example response

```
{
  "id": 1, "jsonrpc": "2.0",
  "result": {
    "items": [
      {"key": "WESTERMO-SW6-MIB::cfgSysHostname.0", "value": "Rmodem", "type": "
        ↳ string"},
      {"key": "WESTERMO-SW6-MIB::cfgSysTimezone.0", "value": "UTC", "type": "
        ↳ string"},
      {"key": "WESTERMO-SW6-MIB::swCfgChangesCount.0", "value": 0, "type": "
        ↳ int"}
    ],
    "errors": [
      {"key": "WESTERMO-SW6-MIB::cfgSysHostname", "error_no": -1, "error_msg": "
        ↳ general error"}
    ]
  }
}
```

16.3.1.3 Set MIB items

- method: set(Array of ConfigurationItem)
 - ConfigurationItem: Supported are all items which are defined in the available MIBs of the device.
- result:
 - items: Array of ConfigurationItemValues which have been set successfully.
 - errors: Array of ConfigurationItemValues with invalid keys (this element is only present if invalid keys have been detected).
- **Note:** Changes will not be applied as long as you do not set rpcCfgApply.0 itself.

Example POST request

```
{
  "id": 1, "jsonrpc": "2.0",
  "method": "set",
  "params": {
    "items": [
      {"key": "WESTERMO-SW6-MIB::cfgSysHostname.0", "value": "RM1"},
      {"key": "WESTERMO-SW6-MIB::cfgSysHostname", "value": "foo"},
    ]
  }
}
```

```
    {"key": "WESTERMO-SW6-MIB::cfgLogFileEnabled.0", "value": 1}
  ]
}
```

Example response

```
{
  "id": 1, "jsonrpc": "2.0",
  "result": {
    "items": [
      {"key": "WESTERMO-SW6-MIB::cfgSysHostname.0", "value": "RM1"},
      {"key": "WESTERMO-SW6-MIB::cfgLogFileEnabled.0", "value": 1}
    ],
    "errors": [
      {"key": "WESTERMO-SW6-MIB::cfgSysHostname", "value": "foo", "error_no":
        ↪ -1, "error_msg": "set failed"}
    ]
  }
}
```

17 Message Codes

Variable text, inserted at the time the message is created, is displayed using the place holder '<val>'.

[0]

```
INFO 0 <val>
```

Value is either 'Reset errors by WebGUI' or 'Manual Error Reset'.

[100]

```
ERROR 100 SYS_MON: Voltage sensor '<val>' is out of range <val> / [<val>, <val>  
↔ >]
```

This trap is sent when supply voltage or one of the internal voltages are outside specified limits which are hardware dependent.

[101]

```
ERROR 101 SYS_MON: Temperature sensor '<val>' is out of range <val> / [<val>,  
↔ <val>]
```

This trap is sent when internally measured temperature is outside specified limits which are hardware dependent.

[105]

```
CRITICAL 105 SYS_MON: Failure by reading value <val>. This value isn't  
↔ monitored anymore
```

This trap is sent when internally a value couldn't read.

[130]

```
INFO 130 <val>|<val>|<val>||<val>|<val>|<val>|<val>|<val>|<val>||<val>|<val>|<  
↔ val>|<val>|<val>|<val>|<val>|<val>||<val>|<val>|<val>|<val>
```

This message is sent periodically containing current 'iw <iface> station dump'. Format is 'interface|mac|inactive time|rx bytes|rx packets|tx bytes|tx packets|tx retries|tx failed||signal combined|avg

signal combined|signal ch0|avg signal ch0|signal ch1|avg signal ch1|signal ch2|avg signal ch2||rx bitrate mode|rx bitrate value|tx bitrate mode|tx bitrate value'.

[131]

```
INFO 131 <val>|<val>|<val>|<val>|<val>|<val>|<val>|<val>|<val>|<val>|<val>
```

This message is sent periodically containing counters of the wireless interface. Format: 'NofStations|TxPackets|RetryCount|6Mbps|9Mbps|..|54Mbps'.

[200]

```
NOTICE 200 Device has restarted because of <val>
```

This message is sent at boot-up process. RESET CAUSE can be either coldstart, warmstart, watchdog and oops.

[201]

```
NOTICE 201 System startup
```

This message is sent when the system starts up.

[202]

```
NOTICE 202 Firmware update started
```

This message is sent when a firmware update is initiated.

[203]

```
NOTICE 203 System reboot
```

This message is sent when a system reboot is issued.

[205]

```
NOTICE 205 Factory Reset confirmed, System configuration changed
```

This message is sent whenever the system configuration is changed.

[206]

```
NOTICE 206 Reserved
```

[207]

WARNING 207 Invalid upgrade image for this platform.

[208]

ERROR 208 Corrupt firmware package!

[209]

ERROR 209 Transfer firmware to device failed!

[210]

WARNING 210 Decryption of the upgrade image failed!

[220]

NOTICE 220 Start writing device identification memory!

[221]

NOTICE 221 Device identification memory successfully updated!

[222]

CRITICAL 222 Failed to update device identification memory!

[300]

INFO 300 NTP: time synchronization failed!

This message is generated when ntp client is configured in unicast mode and it failed to connect to NTP server.

[310]

NOTICE 310 Reserved

[320]

ERROR 320 BIST: Daemon '<val>' isn't running, recover it

This message is generated when process with name process is not running and has to be restarted by the bist

[321]

```
ERROR 321 BIST: Daemon watchdog is not running - force restart
```

This message is generated when watchdog process is not running. System reboots afterwards.

[322]

```
CRITICAL 322 Critical hardware failure: <val>
```

This message is generated when a critical hardware failure was detected during boot up. Please read the user manual about the "Support File".

[323]

```
ERROR 323 Kernel log(s) found
```

This message is generated when a kernel log(s) was found during boot up. Please read the user manual about the "Technical Support File".

[330]

```
INFO 330 BBMON: Backbone is down.
```

This message is generated when the connection to the backbone is detected as lost.

[331]

```
INFO 331 BBMON: Backbone is up.
```

This message is generated when the connection to the backbone is regained.

[332]

```
INFO 332 BBMON: icmp target <val> is down.
```

This message is generated when the icmp target on the backbone is detected as down.

[333]

```
INFO 333 BBMON: icmp target <val> is up.
```

This message is generated when the icmp target on the backbone is detected as up.

[350]

```
INFO 350 ICL: connection established (mode: <val>)
```

This message is generated when ICL establishes a connection to a partner. mode=AP/STA

[351]

```
INFO 351 ICL: connection lost (mode: <val>)
```

This message is generated when ICL lost connection to the partner. mode=AP/STA

[360]

```
INFO 360 CELLULAR: Link is up (iface: '<val>', slot: <val>)
```

Arguments are <iface>, the interface name; and <slot>, the number of the active SIM slot.

[361]

```
INFO 361 CELLULAR: Link is down (iface: '<val>', slot <val>, reason: '<val>')
```

Arguments are <iface>, the interface name; <slot>, the number of the active SIM slot; and a <reason> (e.g. 'low RSSI').

[400]

```
INFO 400 <val>: Station is associated
```

This message is used to trigger led status

[401]

```
INFO 401 <val>: Station is disassociated, Reason: <val>
```

This message is used to trigger led status

[402]

```
INFO 402 <val>: Station is authorized
```

This message is used to trigger various daemons.

[403]

```
NOTICE 403 WLAN: Authentication failure
```

[404]

```
NOTICE 404 <val>: Association failure
```


[405]

NOTICE 405 Reserved

[406]

NOTICE 406 WLAN: Max number of station exceeded

[407]

ERROR 407 Reserved

[408]

INFO 408 WLAN: Failed to connect to <val>: <val>

Format: WLAN: Failed to connect to <mac>: [timed out|(code) error message]

[413]

WARNING 413 Switched to secondary SSID |<val>|

[415]

NOTICE 415 WLAN: Station couldn't find better AP after <val> consecutive scans

[421]

NOTICE 421 <val>: CSA <val> => <val>

Format: '<iface>: <old freq> => <new freq>'

[430]

NOTICE 430 TS|<val>|<val>|RSSI_BCN|<val>

Format: 'TS|<timestamp>|<bssid>|RSSI BCN|<rssi>'

[431]

NOTICE 431 TS|<val>|<val>|RSSI_BCN|<val>|<val>|<val>|<val>

Format: 'TS|<timestamp>|<bssid>|RSSI BCN|<raw rssi>|<cur filer rssi>|<short filter rssi>|<long filter rssi>'

[432]

NOTICE 432 Reserved

[433]

INFO 433 WLAN: Low RSSI event: <val>

This event is used to trigger various daemons.

[434]

NOTICE 434 <val>: Handoff:|<val>|<val>|<val>|<val>|<val>|<val>|<val>|

Format: '<iface>: Handoff:|<reason>|<count>|<prev bssid>|<cur bssid>|<ssid mgmt>|<ho time>|<offchan scan time>|'

[435]

ERROR 435 Radius <val> server <val>:<val> not available (reason <val>)

AP: Radius server not available/invalid shared secret/connection rejected. Format: Radius <authentication, accounting> server <ip>:<port> not available (reason: <not available>, <shared secret>, <rejected>)

[436]

NOTICE 436 (T)TLS Authentication started

STA: Full (T)TLS authentication has been started

[437]

ERROR 437 Certificate <val> is <val>

STA: T(TLS) certificate errors. Format: Certificate <ca, client> is <wrong, missing, expired>

[438]

ERROR 438 TTLS: Invalid username and/or password

STA: TTLS Username and/or password is not recognized by radius server

[440]

WARNING 440 EAP: Auth. session failure: EAP ID: <val>, reason: <val>

EAP ID: RADIUS Server IP address; reason: Failure reason code

Failure reason codes: initialize(1), disconnected(2), connecting(3), authenticating(4), authenticated(5), aborting(6), held(7), forceAuth(8), forceUnauth(9), restart(10)

[441]

```
WARNING 441 RADIUS: Access-Reject: RADIUS Server: <val>
```

Access reject for RADIUS Server IP address

[442]

```
WARNING 442 RADIUS: Response Auth. failed: RADIUS Server: <val>
```

Response Auth. failure for RADIUS Server IP address

[443]

```
INFO 443 RADIUS: Switching to <val> server
```

Switching to either 'PRIMARY' or 'SECONDARY'

[444]

```
INFO 444 Certmgmt: <val> expiration: Subject: <val>; Issuer: <val>; Serial: <
↳ val>; Exp.date: <val>
```

Expiration alert for CA, CA2 and client CRT

[445]

```
INFO 445 Certmgmt: <val> expiration: Issuer: <val>; Exp.date: <val>
```

Expiration alert for CRL and CRL2

[446]

```
INFO 446 SCEP: CA certificate retrieval: <val>
```

s: 'SUCCESS' or 'FAILURE (err: <code>)'

[447]

```
INFO 447 SCEP: Client certificate enrollment: <val>
```

s: 'SUCCESS' or 'FAILURE (err: <code>)'

[448]

INFO 448 SCEP: Client certificate re-enrollment: <val>

s: 'SUCCESS' or 'FAILURE (err: <code>)'

[449]

INFO 449 Certmgmt: <val> download: <val>

1st s: Data type, CA CRL or CA2 CRL; 2nd s: either 'SUCCESS' or 'FAILURE'

[450]

NOTICE 450 Forced deauthentication of all stations

[451]

WARNING 451 Noisefloor above limit

[452]

WARNING 452 Noisefloor below limit

[500]

CRITICAL 500 CONFIG: Unable to connect to IPC system (ubus)!

[501]

CRITICAL 501 CONFIG: Unable to read from UCI!

[510]

ERROR 510 CONFIG: Invalid configuration, reverting to previous configuration!

[511]

ERROR 511 CONFIG: Unable to save new configuration!

[512]

CRITICAL 512 CONFIG: Unable to apply previous configuration!

[513]

WARNING 513 CONFIG: Unable to set config parameter: <val>

Warning for configuration import.

[514]

ERROR 514 CONFIG: <val>

Error detected during configuration validation.

[515]

ERROR 515 CONFIG: Invalid WLAN operation mode: <val>

Some devices may not support all operation modes. Please check the datasheet of the device.

[516]

WARNING 516 Country code <val> not supported on this product/revision!

Use this message when country code is not supported by this product/revision

[517]

ERROR 517 <val> is not supported on this product!

Some devices may not support all functions. Please check the manual of the device.

[530]

ERROR 530 PENDING CHANGES: <val>

A firmware upgrade is not allowed if there are pending changes in the configuration.

[531]

ERROR 531 System upgrade to a major version other than 6 is not supported.

[532]

ERROR 532 'Keep config' on a system downgrade is not supported.

[533]

CRITICAL 533 Config manipulation for an upgrade failed. This is not
↳ correctable situation. Do a factory reset.

[540]

ERROR 540 Failed to verify key or certificate file.

[580]

ERROR 580 CONFIG FILE: Transfer failed!

[581]

WARNING 581 CONFIG FILE: <val>

Configuration file import warning: Config version, Product, Firmware Name or Version mismatch.

[582]

ERROR 582 CONFIG FILE: Unable to read or parse!

[583]

ERROR 583 CONFIG FILE: Incorrect encryption key!

[585]

ERROR 585 Certificate import/export failed!

[600]

ERROR 600 AFC <val>: AFM did not declare after <val> seconds

[601]

ERROR 601 AFM <val>: adjacent AFM did not declare after <val> seconds

Not used, since role of AFM is unknown at startup

[602]

ERROR 602 AFM <val>: redundant AFM did not declare after <val> seconds

Not used, since role of AFM is unknown at startup

[603]

ERROR 603 AFC <val>: invalid AFM <val> tried to declare

[604]

```
ERROR 604 AFM <val>: could not connect to AFC <val>
```

[605]

```
ERROR 605 <val>: could not send report to <val>
```

[630]

```
INFO 630 AFM <val>: AFC <val> set to operational frequency <val> with penalty  
↔ <val>
```

[631]

```
INFO 631 AFM <val>: becoming active for area <val>
```

[632]

```
ERROR 632 AFM <val>: invalid AFM configured: <val>
```

This unifies all AFM/RAFM/AAFM configuration errors

[650]

```
INFO 650 RADIUS: <val> authentication server is up (server <val>:<val>)
```

This message is sent if the Authenticator declares a RADIUS server as up

[651]

```
INFO 651 RADIUS: <val> authentication server is down (server <val>:<val>)
```

This message is sent if the Authenticator declares a RADIUS server as down

[700]

```
ERROR 700 NET: Configuration failed!
```

This message is sent if the network couldn't be set up. Possible reason are wrong protocol, missing or invalid netmask or ip address.

[701]

```
WARNING 701 NET: Configuration failed, but try to continue anyway.
```

This message is sent if the network couldn't be set up. Possible reason are the interface we try to configure does not exist.

[710]

```
ERROR 710 NET: Unable to set the default gateway!
```

This message is sent if the default gateway couldn't be set properly. This happens if the destination can not be reached, or no matching subnet exist.

[711]

```
WARNING 711 NET: Unable to set the default gateway!
```

This message is sent if the default gateway couldn't be set properly. This can happen if the default gateway is already set by DHCP.

[712]

```
ERROR 712 NET: Unable to set a static route!
```

This message is sent when an static route couldn't be set.

[713]

```
WARNING 713 NET: Unable to set a static route, but apply process continued.
```

This message is sent when an static route couldn't be set.

[720]

```
ERROR 720 NET: Wireless configuration failed!
```

This message is sent when the configuration manager is not able to set up a wireless interface.

[721]

```
ERROR 721 NET: A wireless interface can not be bridged to an eth without 4addr  
↔ mode or l2nat!
```

This message is sent when the configuration manager is not able to set up a wireless interface because of missing 4addr mode or l2nat.

[722]

```
ERROR 722 NET: 4addr mode and Layer 2 NAT can not be enabled at the same time!
```


This message is sent when the configuration manager is not able to set up a wireless interface because 4addr mode and L2 NAT is enabled at the same time.

[730]

```
ERROR 730 NET: Creation of IP address failed! Probably the IP/netmask is  
↳ invalid.
```

This message is sent if an ip couldn't be set up. Possible reason are missing or invalid netmask or ip address.

[731]

```
WARNING 731 NET: The interface <val> to create the IP address on doesn't exist.  
↳
```

This message is sent if the parent interface for an ip doesn't exist.

[732]

```
WARNING 732 NET: The interface <val> is not a valid interface to create an IP  
↳ address on.
```

This message is sent if the parent interface for an ip is not valid (e.g. it's bridged).

[740]

```
ERROR 740 NET: Creation of VLAN failed!
```

This message is sent if the creation of a vlan failed.

[741]

```
WARNING 741 NET: Parent interface missing for VLAN.
```

This message is sent if the creation of a vlan failed because the parent doesn't exist.

[742]

```
ERROR 742 NET: Creation of MacVLAN <val> failed!
```

This message is sent if the creation of a macvlan failed.

[743]

```
ERROR 743 NET: Setting MAC-address <val> on Interface <val> failed!
```

This message is sent if a MAC address for an interface could not be set.

[750]

```
WARNING 750 NET: Adding a QoS rule failed.
```

[760]

```
WARNING 760 NET: Bridges which forward link local traffic can not have more  
↳ than 2 interfaces (bridge index >=100)
```

[770]

```
ERROR 770 NET: Error during network init: <val>
```

This message is sent when an error occurs during network initialisation. It contains additional information regarding what went wrong.

[800]

```
INFO 800 DFS: Starting CAC on <val> MHz.
```

This message is sent when a CAC or Off-Channel CAC is started.

[801]

```
INFO 801 DFS: Radar found on <val> MHz.
```

This message is sent when a radar pattern during In-Service Monitoring, CAC or Off-Channel CAC is detected.

[802]

```
INFO 802 DFS: Channel on <val> MHz becomes Available.
```

This message is sent when a channel on the given frequency becomes Available after a CAC or Off-Channel CAC.

[803]

```
INFO 803 DFS: Channel on <val> MHz becomes Usable again.
```

This message is sent when a channel on the given frequency becomes Usable after the NOP time.

[804]

```
INFO 804 DFS: Starting In-Service Monitoring on <val> MHz.
```

This message is sent when the In-Service Monitoring for the Operating Channel on the given frequency.

[805]

```
INFO 805 DFS: All initial CACs done.
```

This message is sent when all DFS frequencies have passed the initial CAC.

[810]

```
ERROR 810 Cellular interface disabled, reason: <val>
```

This message is sent when the cellular interface was disabled due to HW reasons or wrong PIN configuration.

[900]

```
INFO 900 CARP: instance <val> is now a BACKUP
```

This message is sent when a CARP instance became a backup.

[901]

```
INFO 901 CARP: instance <val> is now a MASTER
```

This message is sent when a CARP instance became a master.

[910]

```
ERROR 910 CARP: The syncinterface <val> does not have an unique IP.
```

This message is sent when an error occurs during carp-init.

[911]

```
ERROR 911 CARP: No valid ipaddr for the primary IP of the carp instance <val>  
↔ found.
```

This message is sent when an error occurs during carp-init.

[2700]

```
INFO 2700 IF-MIB: linkUp (<val>, <val>, <val>)
```

IF-MIB::linkUp: Provide ifIndex, ifAdminStatus and ifOperStatus

[2701]

INFO 2701 IF-MIB: linkDown (<val>, <val>, <val>)

IF-MIB::linkDown: Provide ifIndex, ifAdminStatus and ifOperStatus

[2710]

INFO 2710 BRIDGE-MIB: newRoot (<val>)

BRIDGE-MIB::newRoot: Provide ifIndex or 0

[2711]

INFO 2711 BRIDGE-MIB: topologyChange (<val>)

BRIDGE-MIB::topologyChange: Provide ifIndex or 0

18 CLI Commands

18.1 apply - Apply all pending configuration changes

USAGE: apply

The apply command applies all pending configuration changes. To show a list of all depending changes use the changes command.

EXAMPLES:

- * Apply all pending changes:
 apply

18.2 changes - Show a list of changed configuration parameters

USAGE: changes

The changes command shows a list of changed configuration parameters. All listed parameters are still pending, to apply these parameters use the apply command.

18.3 dmesg - Print the kernel ring buffer

USAGE: dmesg

The dmesg tool is used to examine the bootup messages of the kernel.

EXAMPLES:

- * Print the bootup messages:
 dmesg

18.4 get - Show the value of a configuration parameter

USAGE: `get [MIB:]PARAMETER`

Get the given configuration PARAMETER. If you want to get parameters which are not part of the default MIB of the product, you must specify the MIB.

Be aware that the `get` command always returns the current value. This may differ from the applied value. Use the `changes` command to review changed values.

EXAMPLES:

- * Get the hostname:
`get cfgSysHostname.0`
- * Get the base MAC address of the IEEE 802.1d bridge:
`get BRIDGE-MIB::dot1dBaseBridgeAddress.0`

18.5 `grep` - Print lines matching a pattern

USAGE: `grep [OPTIONS] PATTERN [FILE...]`

`Grep` searches the named input FILES for lines containing a match to the given PATTERN. If no files are specified, or if the file "-" is given, `grep` searches standard input. By default, `grep` prints the matching lines.

OPTIONS:

- i Ignore case
- v Select non-matching lines
- h Show a more detailed help text.

EXAMPLES:

- * Filter the output of `logread` for handoff messages while ignoring upper and lower case:
`logread -f | grep -i handoff`
- * Show all system log messages except for those including the keyword `RSSI`:
`logread -f | grep -v RSSI`

18.6 `help` - Show a list of all CLI commands

USAGE: `help [COMMAND]`

The `help` command show a list of all Command Line Interface (CLI) commands. If a COMMAND is specified it shows a more detailed help text of the command.

EXAMPLES:

- * Show all commands:

```
help
* Show the help text of iperf:
help iperf
```

18.7 ip - Show or manipulate network devices

USAGE: ip [OPTIONS] OBJECT {COMMAND | help}

The ip tool allows the user to run COMMANDs on different network OBJECTs. The set of possible actions depends on the object type. As a general rule, it is possible to add, delete and show (or list) objects. The help command is available for all objects. It prints out a list of available commands and argument syntax conventions.

The most important OBJECTs are: address, link, route, neigh and monitor.

OPTIONS:

-br[ief] Print only basic information in a tabular format.

EXAMPLES:

- * Show more detailed help of the route object:
ip route help
- * Show a brief overview of all network addresses:
ip -br address
- * Show a brief overview of all network links:
ip -br link
- * Show the neighbour table (ARP table):
ip neigh
- * Show the routing table:
ip route

18.8 iperf - Perform network throughput tests

USAGE: iperf [-u] [-s | -c SERVER] [OPTIONS]

Iperf is a tool for performing network throughput measurements. It can test either TCP or UDP throughput. To perform an iperf test the user must establish both a server and a client.

OPTIONS:

- u Use UDP rather than TCP.
- s Run in server mode.
- c SERVER

```
Run in client mode, connecting to SERVER.
-i N      Pause N seconds between periodic bandwidth reports.
Client specific options:
-b n[KM]  Set target bandwidth to n bits/s (default 1Mbit/s). This option
          requires UDP mode (-u).
-t N      Time in seconds to transmit for (default 10 secs).
```

EXAMPLES:

```
* Start a server to discard traffic:
  iperf -s
* Start a client connecting to 192.168.1.1 to generate traffic for 5 seconds:
  iperf -c 192.168.1.1 -t 5
* Start an UDP server on port 6001 the report every second:
  iperf -u -s -p 6001 -i 1
* Start an UDP client connecting to 192.168.1.20 on port 6001 with 100Mbit/s:
  iperf -c 192.168.1.20 -p 6001 -u -i 1 -b 100M
```

18.9 iw - Show or manipulate wireless devices

```
USAGE: iw [OPTIONS] {help [COMMAND] | {dev | phy | reg } COMMAND }
```

The iw tool allows the user to run COMMANDs on different wireless objects (dev, phy, reg). The set of possible actions depends on the object type. The help command will print all supported commands.

OPTIONS:

```
--debug    Enable netlink debugging
--version  Show the version of iw
```

EXAMPLES:

```
* Show a help text for the event command:
  iw help event
* List all wireless interfaces:
  iw dev
* List all physical interfaces:
  iw phy
* Show all connected clients to wlan0:
  iw wlan0 station dump
* Show wireless events and their relative timestamp:
  iw event -
```

18.10 logread - Show system log messages

USAGE: logread [OPTIONS]

Logread shows system log messages from the internal buffer.

OPTIONS:

-f Follow the log messages.
-t Add an extra timestamp.
-h Show a more detailed help text.

EXAMPLES:

* Show the system log messages (syslog) as they is created:
logread -f
* Show the syslog messages with their timestamp information:
logread -t

18.11 ping - Ping network hosts

USAGE: ping [OPTIONS] HOST

Send ICMP ECHO_REQUEST packets to network hosts.

OPTIONS:

-h Show a more detailed help text.

EXAMPLES:

* Ping 192.168.1.1 until the ping command is terminated by pressing CTRL-C:
ping 192.168.1.1

18.12 reset - Reset configuration parameters

USAGE: reset

The reset command resets all configuration parameters to their default values. The changes made by reset might be reviewed using the changes command.

EXAMPLES:

* Reset all configuration parameters:
reset

18.13 revert - Revert all pending changes

USAGE: revert

All pending configuration changes can be undone by using the revert command. This is not valid for already applied changes.

EXAMPLES:

```
* Revert all pending configuration changes:
    revert
```

18.14 set - Set the value of a configuration parameter

USAGE: set [MIB:]PARAMETER VALUE

Change the VALUE of a given configuration PARAMETER. If you want to set parameters which are not part of the default MIB of the product, you must specify the MIB.

Note the changes only take effect after the apply command is issued.

EXAMPLES:

```
* Set a new hostname:
    set cfgSysHostname.0 new-hostname
```

18.15 ssh - A secure shell client

USAGE: ssh [OPTIONS] [USER@]HOST[/PORT] [COMMAND]

The ssh client allows to connect to a remote ssh server.

EXAMPLES:

```
* Connect as user root to 192.168.1.20:
    ssh root@192.168.1.20
```

18.16 tcpdump - Dump traffic on a network

USAGE: tcpdump [-n] [-i IFACE] [EXPRESSION]

Tcpdump prints out a description of the contents of packets on a network

interface that match the boolean EXPRESSION; the description is preceded by a time stamp, printed, by default, as hours, minutes, seconds, and fractions of a second since midnight.

OPTIONS:

- n Don't convert addresses to names.
- i Listen on interface. If unspecified, tcpdump searches the system interface list for the lowest numbered, configured up interface.
- h Show a more detailed help text.

EXAMPLES:

- * Do not resolve names and print all traffic on interface eth0:
tcpdump -n -i eth0
- * Show only ICMP packets on eth0:
tcpdump -n -i eth0
- * Show all packets on eth0 except for port 22 (SSH):
tcpdump -i eth0 -n not port 22

18.17 watch - Execute a program periodically

USAGE: watch [-n SEC] COMMAND

The watch tool runs a COMMAND repeatedly, displaying its output. This allows to watch the program output change over time. By default, the command is run every 2 seconds. This interval may be changes by the -n option.

OPTIONS:

- n Repetition interval in seconds (default 2)

EXAMPLES:

- * Check every second for connected stations:
watch -n 1 iw wlan0 station dump