IbexOS Management Guide

Release 6.12.1

Westermo Network Technologies AB

August 12, 2026

www.westermo.com
info@westermo.com

Version control

Document identification	IbexOS Management Guide
Authors	Westermo Network Technologies AB
Owner	Westermo Network Technologies AB
Revision hash	d747d31b44fffb8d83b6464ef4a496d277f377da

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.

© 2026 Westermo Network Technologies AB

Contents

1 Introduction **12**
- 1.1 Supported Products . . . 12
- 1.2 No Longer Supported Products . . . 13
- 1.3 Connector Labels . . . 14
- 1.4 Supported Features . . . 16
- 1.5 Installation Country, Product Use . . . 17
- 1.6 Delivery Content . . . 18
  - 1.6.1 Important Safety Notes . . . 18
- 1.7 Information on Disposal of Old Electronic Equipment . . . 19
- 1.8 License and Copyright . . . 19

2 Installation **20**

3 Quick Start **21**
- 3.1 Starting the product for the first time . . . 21
- 3.2 Discovery protocols . . . 21
- 3.3 Setting the IP address . . . 22
  - 3.3.1 Setting the IP address via Web Interface . . . 22
- 3.4 Usage of Examples . . . 23

4 Administration **24**
- 4.1 Web-Based Management (Web Interface) . . . 25
  - 4.1.1 Configuration - Quick Setup . . . 25
- 4.2 Simple Network Management Protocol (SNMP) . . . 26
  - 4.2.1 Supported Commands . . . 27
  - 4.2.2 Availability . . . 27
  - 4.2.3 Example: Change and permanently save a Parameter through SNMP command line tool . . . 27
  - 4.2.4 Supported MIBs . . . 28
- 4.3 Configuration Files . . . 29
- 4.4 Command Line Interface (CLI) . . . 30
  - 4.4.1 Configuration Mode . . . 32
  - 4.4.2 Export and Import Configurations with the CLI . . . 33
- 4.5 Web Application Programming Interface (WebAPI) . . . 33
  - 4.5.1 File upload / download . . . 33
  - 4.5.2 Remote Procedure Call (RPC) . . . 34
  - 4.5.3 Device configuration . . . 36
- 4.6 Factory Settings and Reset . . . 36
  - 4.6.1 Factory Reset using Factory Reset Plug . . . 37

4.7 Support . . . 37
4.7.1 Technical Support File . . . 37
4.7.2 Kernel Logs . . . 38
4.7.3 Technical Preview . . . 38
4.8 Status Indication (LED) . . . 39
4.8.1 Ibex 802.11n and cellular products . . . 39
4.8.2 Ibex 802.11ax and 802.11be products . . . 40
4.9 Monitoring TRAPs . . . 41
4.10 Alarm Handling . . . 41
4.11 System Firmware . . . 41
4.11.1 Upgrading System Firmware . . . 41
4.12 Cellular Module Firmware . . . 44
4.12.1 Upgrading Cellular Module Firmware - Local . . . 44
4.12.2 Upgrading Cellular Module Firmware - Remote (FOTA) . . . 45
4.13 Certificate Store . . . 47
4.13.1 Certificate Handling . . . 48
4.13.2 Services using Certificate Store . . . 52
4.13.3 Ciphers . . . 55
4.13.4 Monitoring/Jobs . . . 57
4.14 System . . . 59
4.15 Network Configuration . . . 60
4.15.1 Interfaces and Bridges . . . 60
4.15.2 VLAN Interfaces . . . 60
4.15.3 MAC VLAN Interfaces . . . 61
4.15.4 IP Addresses . . . 61
4.15.5 Network Configuration Examples . . . 61
4.16 Ethernet Configuration . . . 64
4.17 Wireless Configuration . . . 64
4.17.1 WLAN Device physical radio configuration . . . 65
4.17.2 WLAN Interface configuration . . . 66
4.18 Cellular Network Configuration . . . 69
4.18.1 Interface Configuration . . . 69
4.18.2 Device Configuration . . . 69
4.18.3 SIM Slot Configuration . . . 71
4.18.4 Default Bearer Configuration . . . 71
4.18.5 Connection Management Configuration . . . 74
4.18.6 Cellular Network Status . . . 75
4.19 Tunnel Endpoint Configuration . . . 75
4.19.1 Interface Configuration . . . 75
4.19.2 Generic Routing Encapsulation (GRE) . . . 75
4.19.3 Generic Routing Encapsulation Tap (GRETAP) . . . 77
4.19.4 Virtual Extensible LAN (VXLAN) . . . 78
4.20 IPsec Configuration . . . 79
4.20.1 Interface Configuration . . . 79
4.20.2 Introduction to IPsec VPNs . . . 80
4.20.3 Authenticated Keying using Internet Key Exchange . . . 81
4.20.4 Dead Peer Detection . . . 82

4.20.5 Route Based VPNs . . . 82
4.20.6 IPsec Configuration Example . . . 82
4.21 OpenVPN Configuration . . . 89
4.21.1 Interface Configuration . . . 89
4.21.2 OpenVPN Configuration Example . . . 89
4.22 Wireguard Configuration . . . 92
4.22.1 Interface Configuration . . . 92
4.22.2 Wireguard Configuration Example . . . 92
4.23 Wireless Network Access Point . . . 96
4.23.1 Configuration File Example: Access point 2.4GHz . . . 96
4.23.2 Configuration File Example: Access point 5GHz . . . 97
4.24 Wireless Data Rate Control . . . 97
4.24.1 Reduced Set of Bitrates . . . 97
4.24.2 QMRR . . . 97
4.25 802.11s Mesh . . . 98
4.26 Bridge Mode (4addr) . . . 98
4.27 Layer 2 NAT Mode . . . 99
4.28 Wireless MAC Address Overwrite . . . 100
4.29 Wireless Network Optimization . . . 100
4.29.1 Multicast to Unicast Conversion . . . 100
4.30 Wireless Security . . . 101
4.30.1 WPA Encryption . . . 101
4.30.2 Port-based Network Access Control (802.1X) . . . 102
4.30.3 RadSec proxy . . . 106
4.30.4 Management frame protection (MFP, 802.11w) . . . 108
4.30.5 Time Synchronization over WLAN . . . 108
4.31 Dynamic VLAN . . . 109
4.32 Wired 802.1X Authentication . . . 110
4.32.1 Operation . . . 110
4.32.2 Multi Radius Auth-Server Operation . . . 111
4.32.3 Configuration . . . 111
4.33 IP Routing . . . 112
4.33.1 Simple Default Gateway . . . 112
4.33.2 Static Routing Table . . . 112
4.33.3 DHCP Routing Table . . . 113
4.33.4 Equal Cost Multipath Routing (ECMP) . . . 114
4.33.5 Policy Routing . . . 115
4.33.6 Static Multicast Routing . . . 116
4.33.7 Multi WAN and Failover Examples . . . 117
4.34 VLAN . . . 123
4.34.1 Multi SSID and VLAN . . . 123
4.35 Mobility . . . 125
4.35.1 Fast Association . . . 125
4.35.2 Inter-AP Roaming . . . 125
4.35.3 Handoff Filters . . . 127
4.35.4 Mobility Logging . . . 127
4.35.5 Fast BSS Transition (802.11r) . . . 129

4.36 Quality of Service (QoS) . . . 131
4.37 Common Address Redundancy Protocol (CARP) . . . 135
4.38 Network Link Monitor (NLM) . . . 135
4.38.1 NLM Monitor Types . . . 136
4.38.2 NLM Monitor Actions . . . 138
4.38.3 NLM Configuration for CARP Failover . . . 140
4.38.4 NLM Configuration for Backbone Monitor . . . 140
4.39 DNS/DHCP Server . . . 140
4.39.1 DNS Server . . . 141
4.39.2 DHCP Server . . . 142
4.39.3 DHCP Relay . . . 142
4.40 NTP Client/Server . . . 143
4.41 Simple Service Discovery Protocol . . . 144
4.42 Service Indicators and Counters . . . 144
4.42.1 SNMP Trap . . . 144
4.42.2 Counters and Status . . . 145
4.43 Logging Features . . . 145
4.44 Wireless Link Monitor . . . 146
4.45 Inter-Carriage Link (ICL) . . . 146
4.45.1 Configuration of the Inter-Carriage Link Application . . . 147
4.46 Global Navigation Satellite System . . . 152
4.46.1 GNSS Device Configuration . . . 152
4.46.2 NMEA Sentences . . . 153
4.46.3 UBX Messages . . . 153
4.46.4 GPSD Protocol . . . 153
4.47 Public Wireless Network (PWN) . . . 155
4.47.1 Client Steering . . . 155
4.47.2 Airtime Fairness . . . 160
4.47.3 Passpoint and OpenRoaming . . . 161
4.48 RSTP . . . 162
4.49 Dynamic Frequency Selection (DFS) . . . 163
4.49.1 Wireless Standalone . . . 163
4.49.2 Wireless Manager (NWM) . . . 164
4.49.3 Area Frequency Management (AFM) . . . 167
4.50 Interference Detection Function (IDF) . . . 172
4.51 Http Report . . . 173
4.51.1 NWM and ChannelManager Report . . . 173
4.51.2 IDF Report . . . 175
4.52 Firewall . . . 178
4.52.1 L3 Network Address Translation (NAT) . . . 178
4.52.2 L3 Filter . . . 181
4.52.3 L3 Mangle . . . 184
4.52.4 L2 Filter . . . 185
4.52.5 L2 Mangle . . . 190

5 Troubleshooting **191**
5.1 LED Status . . . 191
5.1.1 Ibex 802.11n and cellular products . . . 191
5.1.2 Ibex 802.11ax and 802.11be products . . . 192
5.1.3 Unspecified status . . . 192
5.2 Cellular Connection . . . 192
5.2.1 Peripherals . . . 193
5.2.2 Status . . . 193
5.2.3 Signals . . . 194
5.2.4 Network Information . . . 195
5.2.5 Advanced Analysis Tools . . . 195
5.2.6 Performance & Connection Loss . . . 196

6 Country Codes **197**
6.1 Configuration . . . 197
6.2 Regions for 802.11n products . . . 197
6.2.1 Country code WORLD . . . 197
6.2.2 Region E . . . 198
6.2.3 Region U . . . 204
6.3 Regions for 802.11ax products . . . 207
6.3.1 Region E . . . 207
6.4 Regions for 802.11be products . . . 208
6.4.1 Region E . . . 208

7 Security Considerations **210**
7.1 Physical Interfaces . . . 210
7.2 Network Concept . . . 210
7.2.1 Local Administrative Access . . . 210
7.2.2 Remote Administrative Access . . . 211
7.3 Service Restrictions . . . 212
7.3.1 CLI . . . 213
7.3.2 SNMP . . . 213
7.3.3 HTTP . . . 213
7.4 Passwords . . . 213
7.4.1 Strength Of PSK-Passphrase . . . 214
7.5 Firewall . . . 214
7.6 Logging . . . 215
7.6.1 Syslog . . . 215
7.6.2 SNMP Traps . . . 216

8 WESTERMO-SW6-MIB **217**
8.1 configuration . . . 217
8.1.1 cfgSystem . . . 217
8.1.2 cfgCli . . . 222
8.1.3 cfgCertificate . . . 225
8.1.4 cfgScep . . . 230
8.1.5 cfgVpn . . . 236

8.1.6 cfgLdap . . . 279
8.1.7 cfgRadsec . . . 285
8.1.8 cfgCellular . . . 291
8.1.9 cfgStunnel . . . 310
8.1.10 cfgLogging . . . 314
8.1.11 cfgSnmp . . . 318
8.1.12 cfgDhcp . . . 326
8.1.13 cfgNtp . . . 345
8.1.14 cfgHttp . . . 348
8.1.15 cfgLldp . . . 353
8.1.16 cfgMdns . . . 353
8.1.17 cfgSsdp . . . 354
8.1.18 cfgNetwork . . . 354
8.1.19 cfgWireless . . . 393
8.1.20 cfgRouting . . . 469
8.1.21 cfgNlm . . . 484
8.1.22 cfgQos . . . 492
8.2 rpc . . . 499
8.2.1 rpcConfiguration . . . 499
8.2.2 rpcScep . . . 501
8.2.3 rpcCellular . . . 502
8.2.4 rpcPartitions . . . 504
8.2.5 rpcFirmware . . . 504
8.2.6 rpcSystem . . . 505
8.2.7 rpcCertificate . . . 506
8.2.8 rpcDriver . . . 508
8.3 settings . . . 509
8.3.1 setConfiguration . . . 509
8.3.2 setWireless . . . 511
8.3.3 setFirmware . . . 517
8.3.4 setCellular . . . 517
8.3.5 setCertificate . . . 519
8.3.6 setSystem . . . 522
8.3.7 setTechPreview . . . 523
8.3.8 setTlsClient . . . 523
8.4 hardware . . . 525
8.4.1 hwSystem . . . 525
8.4.2 hwBaseBoard . . . 526
8.4.3 hwIfaceBoard . . . 527
8.4.4 hwNetwork . . . 529
8.4.5 hwSensor . . . 530
8.4.6 hwWireless . . . 532
8.4.7 hwCellular . . . 535
8.4.8 hwGnss . . . 537
8.4.9 hwPowerSupply . . . 537
8.5 software . . . 538
8.5.1 swFirmware . . . 538

8.5.2 swBootloader . . . 540
8.5.3 swSystem . . . 540
8.5.4 swConfiguration . . . 543
8.5.5 swOperatingSystem . . . 543
8.5.6 swDriver . . . 546
8.5.7 swCellular . . . 566
8.5.8 swNlm . . . 598
8.5.9 swRdm . . . 598
8.5.10 swCertificate . . . 599

9 NERATEC-SW6-BLUETOOTH-MIB 604
9.1 bluetooth . . . 604
9.1.1 configuration . . . 604
9.1.2 software . . . 605

10 WESTERMO-SW6-BRIDGE-MIB 607
10.1 rstp . . . 607
10.1.1 configuration . . . 607

11 WESTERMO-SW6-FIREWALL-MIB 611
11.1 firewall . . . 611
11.1.1 configuration . . . 611

12 WESTERMO-SW6-GNSS-MIB 655
12.1 gnss . . . 655
12.1.1 configuration . . . 655
12.1.2 software . . . 662

13 WESTERMO-SW6-ICL-MIB 665
13.1 icl . . . 665
13.1.1 configuration . . . 665
13.1.2 rpc . . . 667
13.1.3 settings . . . 668
13.1.4 software . . . 668

14 WESTERMO-SW6-NWM-MIB 669
14.1 nwm . . . 669
14.1.1 configuration . . . 669
14.1.2 rpc . . . 684

15 WESTERMO-SW6-PWN-MIB 686
15.1 pwn . . . 686
15.1.1 configuration . . . 686

16 WebAPI Detailed Specification 701
16.1 Authentication API . . . 701
16.1.1 RPC Methods . . . 701
16.2 Files API . . . 704
16.2.1 Device configuration file import / export . . . 705

16.2.2 Syslog export . . . 706
16.2.3 System Messages export . . . 706
16.2.4 Security Log export . . . 706
16.2.5 Support File export . . . 706
16.2.6 Cellular Support File export . . . 706
16.2.7 Firmware upload and upgrade . . . 707
16.3 Device Configuration API . . . 708
16.3.1 RPC Methods . . . 708
16.4 Status API . . . 710
16.4.1 RPC Methods . . . 710
16.5 Using WebAPI with curl . . . 712
16.5.1 Authentication . . . 712
16.5.2 Files . . . 713
16.5.3 Device Configuration . . . 713
16.5.4 Status . . . 714

17 Message Codes **716**
17.1 Security Log Messages . . . 716
17.2 Standard Log Messages . . . 738
17.3 Commissioning Log Messages . . . 768

18 WLAN Reason Codes **774**
18.1 Wireless . . . 774
18.1.1 Proprietary Reason Codes . . . 774
18.1.2 IEEE Reason Codes . . . 775

19 CLI Commands **777**
19.1 apply - Apply all pending configuration changes . . . 777
19.2 changes - Show a list of changed configuration parameters . . . 777
19.3 configure - Manage the global configuration . . . 777
19.4 date - Display the current system time . . . 778
19.5 dmesg - Print the kernel ring buffer . . . 778
19.6 factory - Reset the system to its factory settings . . . 778
19.7 get - Show the value of a configuration parameter . . . 779
19.8 gpsmon - GPS packet monitor and control utility . . . 779
19.9 grep - Print lines matching a pattern . . . 779
19.10help - Show a list of all CLI commands . . . 780
19.11ip - Show or manipulate network devices . . . 780
19.12perf - Perform network throughput tests . . . 781
19.13psec - Invoke IPsec utilities . . . 781
19.14w - Show or manipulate wireless devices . . . 782
19.15ogread - Show system log messages . . . 782
19.16ovs-appctl - OpenVswitch Application Control . . . 783
19.17ovs-dpctl - OpenVswitch DataPath Control . . . 783
19.18ovs-ofctl - OpenVswitch OpenFlow Control . . . 784
19.19ovs-vsctl - OpenVswitch VirtualSwitch Control . . . 784
19.20ping - Ping network hosts . . . 784

19.21 ps - Show current processes . . . 785
19.22 qlog - Dump cellular log to TCP/FTP server . . . 785
19.23 reboot - Reboot the system . . . 786
19.24 reset - Reset configuration parameters . . . 786
19.25 revert - Revert all pending changes . . . 786
19.26 session-manager - List or destroy active CLI, WebInterface and WebAPI sessions . . . 786
19.27 set - Set the value of a configuration parameter . . . 787
19.28 sim - SIM Tool for PIN and PUK . . . 787
19.29 ssh - A secure shell client . . . 788
19.30 status - Show system status . . . 788
19.31 support - Display support information . . . 788
19.32 tcpdump - Dump traffic on a network . . . 789
19.33 traceroute - Traceroute network hosts . . . 789
19.34 upgrade - Upgrade firmware with optional configuration file import . . . 790
19.35 upgrade-cellular - Upgrade cellular firmware via FOTA (Firmware Over The Air) . . . 791
19.36 watch - Execute a program periodically . . . 791
19.37 wg - Wireguard management tool . . . 791

20 Deterministic Interface Index **793**

21 Application Notes **794**
21.1 L2 Tunnel Over Multiple EPS Bearer . . . 794
21.1.1 Routing Table per wwan . . . 795
21.1.2 L2 Mangling . . . 795
21.1.3 L2 Tunnel . . . 796
21.1.4 L3 Mangling . . . 797
21.1.5 Policy Routing on Mark . . . 798
21.1.6 Remote configuration . . . 798

1 Introduction

This document describes the functionality and features of the *IbexOS*. The *IbexOS* is the firmware controlling the operation of the Ibex family products.

The Ibex family products are wireless communication devices for demanding industrial applications. The Ibex family devices can operate at 2.4, 5 and 6GHz WLAN bands or 2G, 3G, 4G and 5G cellular depending on installation country limitations and specific product capabilities.

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

IbexOS 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:

			Cellular	
			4G (LTE)	**5G**
			Ibex-RT-330	Ibex-RT-330-5G
WLAN	**802.11n (Wi-Fi 4)**	Ibex-RT-220 Ibex-RT-320 Ibex-RT-370 Ibex-RT-280	Ibex-RT-630	Ibex-RT-630-5G
	802.11ax (Wi-Fi 6)	Ibex-1310 Ibex-1510 Ibex-3510		
	802.11be (Wi-Fi 7)	Ibex-1520 Ibex-3520		

1.2 No Longer Supported Products

The following products are no longer supported with version 6.12.1. Please use the shown version in the table below. It will continue to receive security updates but no new features.

			Latest Supported Version
WLAN	**802.11ac (Wi-Fi 5)**	Ibex-RT-610	6.11.x

1.3 Connector Labels

The following tables show the relation between labels found next to the connectors and names used within the firmware and its user interfaces.

Device	Connector Label	Description	Firmware Name	LED
RT-220	X1	Ethernet 0	eth0	X1
	X2	Ethernet 1	eth1	X2
	A1, A2	Radio 0	radio0 (wlan0)	
RT-320	ETH1	Ethernet 0	eth0	X1
	ETH2	Ethernet 1	eth1	X2
	A1, A2, A3	Radio 0	radio0 (wlan0)	
RT-370	ETH0	Ethernet 0	eth0	X1
	ETH1	Ethernet 1	eth1	X2
	A1, A2, A3	Radio 0	radio0 (wlan0)	
	A4	Radio 1	radio1 (wlan1)	
RT-280	X1	Ethernet 1 (fiber optical)	eth1	X1
	X2	Ethernet 2 (fiber optical)	eth2	X2
	X3	Ethernet 0	eth0	(No LED)
	A1, A2, A3	Radio 0	radio0 (wlan0)	
RT-630	X1	Ethernet 0	eth0	X1
	X2	Ethernet 1	eth1	X2
	A1, A2, A3	Radio 0	radio0 (wlan0)	
	A4, A5	LTE	cellular0 (wwan0)	
	A6	GNSS		
RT-330	X1	Ethernet 0	eth0	X1
	X2	Ethernet 1	eth1	X2
	A1, A2	LTE	cellular0 (wwan0)	
	A3	GNSS		
RT-630-5G	X1	Ethernet 0	eth0	X1
	X2	Ethernet 1	eth1	X2
	A6, A7	Radio 0	radio0 (wlan0)	
	A1, A2, A3, A4	5G	cellular0 (wwan0)	
	A5	GNSS		
RT-330-5G	X1	Ethernet 0	eth0	X1
	X2	Ethernet 1	eth1	X2
	A1, A2, A3, A4	5G	cellular0 (wwan0)	
	A5	GNSS		

Table 1.1: *Connector Labels, Firmware Names and LEDs*

Device	Connector Label	Description	Firmware Name	LED
Ibex-1310	X1	Ethernet 0	eth0	X1
	X2	Ethernet 1	eth1	X2
	A1, A2	Radio 0	radio0 (wlan0)	
	A1, A2	Radio 1	radio1 (wlan1)	
Ibex-1510	X1	Ethernet 0	eth0	X1
	X2	Ethernet 1	eth1	X2
	A1, A2	Radio 0	radio0 (wlan0)	
	A1, A2	Radio 1	radio1 (wlan1)	
Ibex-3510	X1	Ethernet 0	eth0	X1
	X2	Ethernet 1	eth1	X2
	A1, A2	Radio 0	radio0 (wlan0)	
	A1, A2	Radio 1	radio1 (wlan1)	
	B1, B2, B3, B4	Radio 2	radio2 (wlan2)	
Ibex-1520	X1	Ethernet 0	eth0	X1
	X2	Ethernet 1	eth1	X2
	A1, A2	Radio 0	radio0 (wlan0)	
	A1, A2	Radio 1	radio1 (wlan1)	
	BLE	Bluetooth		
Ibex-3520	X1	Ethernet 0	eth0	X1
	X2	Ethernet 1	eth1	X2
	A1, A2	Radio 0	radio0 (wlan0)	
	A1, A2	Radio 1	radio1 (wlan1)	
	B1, B2	Radio 2	radio2 (wlan2)	
	BLE	Bluetooth		
	GNSS	GNSS		

Table 1.2: *Connector Labels, Firmware Names and LEDs (cont.)*

1.4 Supported Features

Feature	RT-320	RT-220	RT-370, RT-280	Ibex-1310, Ibex-1510, Ibex-3510	Ibex-1520, Ibex-3520	RT-330, RT-330-5G	RT-630, RT-630-5G
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
Support	yes	yes	yes	yes	yes	yes	yes
Status Indication (LED)	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
Network Configuration	yes	yes	yes	yes	yes	yes	yes
Ethernet Configuration	yes	yes	yes	yes	yes	yes	yes
Wireless Configuration	yes	yes	yes	yes	yes	no	yes
Cellular Network Configuration	no	no	no	no	no	yes	yes
Tunnel Endpoint Configuration	yes	yes	yes	yes	yes	yes	yes
IPsec Configuration	yes	yes	yes	yes	yes	yes	yes
OpenVPN Configuration	yes	yes	yes	yes	yes	yes	yes
Wireguard Configuration	yes	yes	yes	yes	yes	yes	yes
Wireless Network Access Point	yes	yes	yes	yes	yes	no	yes
Wireless Data Rate Control	yes	yes	yes	no	no	no	yes
802.11s Mesh	yes	yes	yes	no	no	no	yes
Bridge Mode (4addr)	yes	yes	yes	yes	yes	no	yes
Layer 2 NAT Mode	yes	yes	yes	yes	yes	no	yes
Wireless MAC Address Overwrite	yes	yes	yes	yes	yes	no	yes
Wireless Security	yes	yes	yes	yes	yes	no	yes
RadSec proxy	yes	yes	yes	yes	yes	no	yes
Dynamic VLAN	yes	yes	yes	yes	yes	no	yes
Wired 802.1X Authentication	yes	yes	yes[2]	yes	yes	yes	yes
IP Routing	yes	yes	yes	yes	yes	yes	yes
VLAN	yes	yes	yes	yes	yes	yes	yes

Feature	RT-320	RT-220	RT-370, RT-280	Ibex-1310, Ibex-1510, Ibex-3510	Ibex-1520, Ibex-3520	RT-330, RT-330-5G	RT-630, RT-630-5G
Mobility	yes	yes[1]	yes	no	no	yes	yes
Quality of Service (QoS)	yes	yes	yes	no	no	no	yes
Common Address Redundancy Protocol (CARP)	yes	yes	yes	yes	yes	yes	yes
Network Link Monitor (NLM)	yes	yes	yes	yes	yes	no	yes
DNS/DHCP Server	yes	yes	yes	yes	yes	yes	yes
NTP Client/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	no	no	no	yes
Inter-Carriage Link (ICL)	yes	yes	yes	yes	yes	no	yes
Public Wireless Network (PWN)	no	no	no	yes	yes	no	no
Passpoint and OpenRoaming	no	no	no	yes	yes	no	no
RSTP	yes	yes	yes	yes	yes	yes	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
Global Navigation Satellite System	no	no	no	no	yes[3]	yes	yes
Multicast to Unicast Conversion	yes	yes	yes	no	no	yes	yes

1.5 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:

[1] Only Fast Association is supported
[2] RT-280 supports 802.1X only on X3
[3] GNSS is only supported by Ibex-3520.

Country Code	Frequency Range	Notes
Europe(EU)	2400-2483.5 MHz, 5150-5350 MHz, 5470-5725 MHz, 5725-5875 MHz and 5935-6415 MHz	Operation according to ETSI limitations.
United States(USA)	2400-2483.5 MHz, 5150-5350 MHz, 5470-5725 MHz, 5750-5850 MHz and 5935-7115 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.

1.6 Delivery Content

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

1.6.1 Important Safety Notes

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

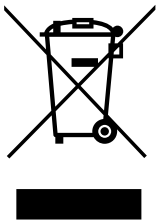
Danger! Never try to open the device. There are no serviceable parts inside! By trying to open the device you will be exposed to a risk of death or injury.

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.

1.7 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.

1.8 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.

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 get started quickly with the device.

3.1 Starting the product for the first time

Warning! Before connecting the power cable to the device, make sure antennas or terminators have been attached to the antenna connectors.

When booting the device for the first time, the factory default configuration is used. With this configuration, the device is prepared to operate as an Access Point with a layer-2 bridge. The WLAN interface (which is disabled) and the two Ethernet ports belong to the same bridge.

The default IP settings for the device are:

- IP address: 192.168.1.20
- Netmask: 255.255.255.0
- DHCP client: Enabled
- Default gateway: 0.0.0.0

Before connecting the device to the LAN, make sure to change its IP settings according to the network topology.

3.2 Discovery protocols

Various protocols are enabled by default for helping discover the connected products:

- **SSDP:** The `Windows File Explorer` lists connected devices in the `Network` section. Clicking on the device opens the Web Interface. No need for setting up an IP address.
- **MDNS:** Provides a link local address that is useful for SSDP or other device discovery tools.
- **LLDP:** Helps to identify the device when it is connected to a switch port.

3.3 Setting the IP address

The device can be configured to use static IP addresses, obtain a dynamic addresses via DHCP or automatically assign link local addresses.

3.3.1 Setting the IP address via Web Interface

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

- PC IP address: 192.168.1.2
- PC Netmask: 255.255.255.0

Open the web browser and enter **https://192.168.1.20** in the address field. When the login page requests username and password, enter the factory default credentials for the administrator:

- Username: 'admin' (or 'webadmin' as alias)
- Password: 'admin'

After successful login the "First Time Setup" page is displayed, asking for the country code of the current location. Select the country code and proceed by clicking on the "Save" button.

Wait some time before being redirected to **Configuration - Quick Setup** (see Configuration - Quick Setup for more details) where the current settings are displayed. In section System change the **Hostname** and **IP Address** (in CIDR notation) as needed. Or set the **Default Gateway**, where a value of "0.0.0.0" means that there is no Default Gateway in use.

Click the Apply button to apply the settings.

Wait some time before accessing the Web interface from the above set **IP address**.

3.4 Usage of Examples

There are configuration examples like the following one for different parts/use cases in this user manual.

```
WESTERMO-SW6-MIB::cfgSysHostname.0 = ap1
```

Listing 3.1: *CLI configuration example*

These examples are ment to be used in the CLI as following:

```
admin@Rmodem:/$ set WESTERMO-SW6-MIB::cfgSysHostname.0 = ap1
```

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 a defined set of default values.

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 and password (see local users below)

The following local users are predefined to access the Web Interface:

Username	Default Password	Role
admin (or 'webadmin' as alias)	admin	admin
monitor	monitor	monitor

Table 4.1: *Local Users*

Users with role admin have the most access rights. And users with role monitor have very restricted access rights and cannot do any change on the device.

Use the **System -> Administration** page to change the password of the logged in Local User.

Login with an LDAP user is possible if LDAP access is properly configured (cfgLdapEnabled is enabled(1), etc.) and the LDAP server (cfgLdapUrl1 or cfgLdapUrl2) is reachable. See LDAP Client for how to configure LDAP access.

Use cfgHttpSessionLimit to restrict the maximum number of connected users.

Anti brute-force mechanism is activated when a login fails (e.g. due to a wrong password) several times within a short period of time. Then login will be blocked for some time.

4.1.1 Configuration - Quick Setup

Quick Setup allows for quick and easy configuration of the device, for example to use the device as Access Point.

Configuration of the following parameters is possible:

- System parameters (like Hostname and IP address)
- for WLAN devices only
  - WLAN country code

  - WLAN interfaces

- for LTE devices only
  - Enable/disable DNS/DHCP
  - Enable/disable GNSS
  - Cellular interface

If some advanced configuration was set which cannot be handled by Quick Setup, a warning message will be displayed describing the options on how to proceed. Normally it is recommended to not use Quick Setup in these cases.

Use the **Configuration - Advanced** page for configuring more advanced features.

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
cfgSnmpdVersion	v2c
cfgSnmpdComAdmin (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 *IbexOS* 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-BRIDGE-MIB
- WESTERMO-SW6-FIREWALL-MIB
- WESTERMO-SW6-GNSS-MIB
- WESTERMO-SW6-ICL-MIB
- WESTERMO-SW6-NWM-MIB
- WESTERMO-SW6-PWN-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

All relevant parameters to operate a device in a clearly defined manner can be exported or imported by means of a configuration file. This can be achieved by using one of the following options:

- The Web-Interface provides functionality for exporting and importing configuration files on the **System - Maintenance** page.
- An SNMP manager can be used to trigger a transfer of a configuration file from or to a specifed URL, see setCfgFileUrl, setCfgFileFormat, setCfgFileType, setCfgFilePassword and rpcCfgFile.
- In the configuration mode of the CLI, the `export` and `import` commands can be used to export and import configurations respectively, refer to Section 4.4.2.
- The WebAPI provides special endpoints in order to transfer configuration files from or to the device. Section 16.2.1 describes how to use these endpoints.

Two configuration file formats (setCfgFileFormat) are available:

1. The SNMP-styled file is a flat list of the configuration parameters in an SNMP-based format. Because the parameter names are identical to ones used in the MIBs (e.g. WESTERMO-SW6-MIB), the documentation of each parameter can be found there.
2. The CLI-styled file reflects the parameters of the device in a hierarchical structure in the way they are used in the CLI.

Since both configuration files are formatted in plain text, it is possible to modify them with a simple text editor.

Configuration files can be encrypted/decrypted by providing an encryption or decryption password (setCfgFilePassword) when exporting or importing a configuration file.

openssl enc command is used for configuration file encryption/decryption:

- For file encryption, the following openssl command is used: openssl enc -aes-256-cbc -pbkdf2 -iter 2048 -in <config_file> -pass pass:<password>
- For file decryption, the following openssl command is used: openssl enc -d -aes-256-cbc -iter 2048 -in <config_file> -pass pass:<password>

4.4 Command Line Interface (CLI)

The CLI is available either through Secure Shell (SSH) or Telnet. It can be configured with the parameters listed in Table 4.2. For further information on CLI configuration, in particular on security considerations, refer to Section 7.3.1 on page 213.

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

Table 4.2: *CLI Parameters and Their Default Values*

The CLI is basically divided into two modes: the execution and the configuration mode. Right after a successful login to the CLI the execution mode is active. In this mode it provides a selection of important maintenance tools. The `help` command offers explanations and examples for each tool.

Table 4.3 lists all commands and tools of the CLI.

Command	Short Description
apply	Apply all pending configuration changes
changes	Show a list of changed configuration parameters
configure	Manage the global configuration
date	Display the current system time
dmesg	Print the kernel ring buffer
factory	Reset the system to its factory settings
get	Show the value of a configuration parameter
gpsmon	GPS packet monitor and control utility
grep	Print lines matching a pattern
help	Show a list of all CLI commands
ip	Show or manipulate network devices
iperf	Perform network throughput tests
ipsec	Invoke IPsec utilities
iw	Show or manipulate wireless devices
logread	Show system log messages
ovs-appctl	OpenVswitch Application Control
ovs-dpctl	OpenVswitch DataPath Control
ovs-ofctl	OpenVswitch OpenFlow Control
ovs-vsctl	OpenVswitch VirtualSwitch Control
ping	Ping network hosts
ps	Show current processes
qlog	Dump cellular log to TCP/FTP server
reboot	Reboot the system
reset	Reset configuration parameters
revert	Revert all pending changes
session-manager	List or destroy active CLI, WebInterface and WebAPI sessions
set	Set the value of a configuration parameter
sim	SIM Tool for PIN and PUK
ssh	A secure shell client
status	Show system status
support	Display support information
tcpdump	Dump traffic on a network
traceroute	Traceroute network hosts
upgrade	Upgrade firmware with optional configuration file import
upgrade-cellular	Upgrade cellular firmware via FOTA (Firmware Over The Air)
watch	Execute a program periodically
wg	Wireguard management tool

Table 4.3: *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, enter the configuration mode and change the current configuration.

Suppose the CLI of a IbexOS device is accessible at 192.168.1.20 with the user name `myself`, use the following command to connect via SSH and enter the password when prompted. The default user name and password can be found in Table 4.2.

```
$ ssh myself@192.168.1.20
myself@192.168.1.20's password:
     ______    _____      _____
   .' ___  |  |_   _|    |_   _|   IbexOS - Command Line Interface (CLI)
  / .'   \_|    | |        | |
  | |           | |   _    | |     Enter 'help' to get assistance on commands
  \ '.___.'\   _| |__/ |  _| |_    and tools. In order to leave the CLI,
   '.____ .'  |________| |_____|   enter 'exit'.

myself@Rmodem:/$
```

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 4.3. To get an explanation and examples on a specific tool enter `help` and the name of the tool.

```
$ help ping
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

$
```

4.4.1 Configuration Mode

The code extract below describes how to enter the configuration mode, change parameters, show the made changes and how to apply them.

```
$ configure
Rmodem(config)> system
Rmodem(config-system)> hostname
Rmodem
Rmodem(config-system)> hostname newhost
Rmodem(config-system)> timezone CET-1CEST,M3.5.0,M10.5.0/3
Rmodem(config-system)> changes
system.hostname: 'Rmodem' => 'newhost'
system.timezone: 'UTC' => 'CET-1CEST,M3.5.0,M10.5.0/3'
```

```
Rmodem(config-system)> apply
Please wait, applying a new configuration may take up to 90s ...
Apply done.
newhost(config-system)> quit
$
```

4.4.2 Export and Import Configurations with the CLI

The following example shows how configurations can be imported from or exported to a TFTP server.

```
$ configure
Rmodem(config)> export tftp://192.168.1.2/backup.cfg
transferred 13766 bytes to 'tftp://192.168.1.2/backup.cfg'.
Rmodem(config)> import tftp://192.168.1.2/mydevice.cfg
transferred 13769 bytes from 'tftp://192.168.1.2/mydevice.cfg'.
Applying the imported configuration may take up to 90s ...
Successfully imported new configuration.
mydevice(config)> exit
$
```

Note: The CLI only supports exporting or importing the CLI-styled configuration file format.

4.5 Web Application Programming Interface (WebAPI)

The WebAPI shares the user management with Web-Based Management (Web Interface).

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 follows:

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 *IbexOS* 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.

The following list shows the most important default configuration values which are needed to access the device.

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
Enabled Services	
DHCP client)	
Network Management (SNMP)	
Service Discovery (SSDP)	
Webserver (HTTP)	
Name Service (uDNS)	
Link Layer Discovery (LLDP)	

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 LED(s) are constantly on orange/red
5. Remove the FACTORY RESET PLUG within 15 seconds: The LED(s) 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 Support

4.7.1 Technical Support File

The *Technical Support File* is a compressed tar file with collected system information, including log files and system configuration, for analysis of problems. Adding the *Technical Support File* will help the support team to better understand the problem for a support request.

The *Technical Support File* is accessible using the Web Interface. After logging in, go to **System - Support** and click *Download Technical Support File* to create and download the *Technical Support File*.

It is recommended to enable encryption of the *Technical Support File*: Set setSysSfEncryptionEnabled to enabled(1) and set setSysSfEncryptionPassword to an encryption password of your choice.

Admin users can change the *Technical Support File Encryption Settings* by clicking button **Edit Technical Support File Encryption Settings**.

For Cellular devices a *Cellular Support File* can be created and downloaded for more in-deep analysis. Please contact your support contact for more details.

4.7.2 Kernel Logs

It might happen that the Kernel faces a problem which can only be resolved by a reboot of the device. In such a case, the Kernel will store a Kernel Log (crash log) in non-volatile memory for later analysis. The device checks during boot-up if any Kernel Log(s) are available. It will report this issue with a system message (see Message Code: 323).

Kernel Log(s) needs to be reset/removed by a user in the Web Interface under **System - Support**.

Kernel Log(s) are included in the *Technical Support File*. Before resetting the Kernel Log(s) one should first download the *Technical Support File* and send it to the support contact.

4.7.3 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.

4.8 Status Indication (LED)

The Ibex products are equipped with LED indicators. The corresponding technology assignment is listed in Supported Products.

4.8.1 Ibex 802.11n and cellular products

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.

4.8.1.1 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.

4.8.1.2 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.

4.8.1.3 System warnings

Indicated by the **ERR** LED lighting up solid orange.

4.8.1.4 System errors

Indicated by the **ERR** LED lighting up solid red.

4.8.2 Ibex 802.11ax and 802.11be products

DC	indicates whether or not external DC power is connected to the dual input port. The **DC** LED lights up green if both pairs are active, red if only one pair is supplying power. It stays dark if the product is powered with PoE.
ON	is flashing during boot up and afterwards shows a solid green when the device has booted. It indicates errors and warnings which require the intervention of the operator. When an error occurs, the **ON** LED shows 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. There is no color indication for the connection speed.

4.8.2.1 Firmware Update

When a firmware update is in progress, the **ON** LED is flashing orange until device reboot.

IMPORTANT: One MUST NOT interrupt the power source until normal operation state is reached.

4.8.2.2 Factory Reset

The **ON** LED is solid orange when a factory reset is detected. The LED is blinking orange when the factory reset is in process. See also Factory Settings and Reset.

4.8.2.3 System warnings

Indicated by the **ON** LED lighting up solid orange.

4.8.2.4 System errors

Indicated by the **ON** LED lighting up solid red.

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

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 ***.img** firmware files to downgrade from >= **6.9** to a firmware version < **6.9** or if you need ***.bin** 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. Provide administrator credentials (username: 'admin', default password: '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 SNMP

The following steps can be done either by using a TFTP or HTTP/HTTPS server. This server must be reachable from the device.

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 '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.

4.11.1.3 Upgrading Firmware via CLI

Please refer to the CLI command `upgrade`, see chapter CLI Commands. The update process is the same as the SNMP approach and requires either a TFTP or HTTP/HTTPS server from which the firmware image and the optional configuration file can be downloaded.

4.11.1.4 DT70 - Failsafe Firmware Upgrade

DT70 Devices have two partitions available for firmware upgrade.

Use Web Interface **Status -> Summary** or CLI command (see chapter CLI Commands) `status summary` to see which partition is actively used by the device.

FW Partition Set is in one of three different states:

1. Primary (0): Actively running FW is in primary partition.
2. Secondary (1): Actively running FW is in secondary partition.
3. De-synchronised partitions!: Neither primary nor secondary partition are in a good state!

State 3 (De-synchronised partitions!) is an error case that should never happen.

Choosing the partition for the Fimware is done automatically by the system.

Normal Firmware Upgrade process is a follows:

1. Active Firmware is in primary partition.
2. Write new Firmware to secondary partition.
3. Switch active partition to secondary partition.
4. Write same new Firmeware to primary partition.
5. Switch active partition to primary partition.
6. Done.

This process ensures, that a well running Firmware partition is always available, even if a device is powered down while writing the Firmware to a new partition.

If, after successful Firmware upgrade, **FW Partition Set** is Secondary (2), something went wrong when writing primary partition. The device is still working fine with the updated firmware from secondary partition. But it is recommended to once again upgrade the firmware.

4.12 Cellular Module Firmware

Only available for **LTE and 5G** devices.

4.12.1 Upgrading Cellular Module Firmware - Local

Cellular Module Firmware can be upgraded from local host network by defining the URL to a local TFTP/HTTP server with the parameter setCellFwUrl and starting the upgrade with rpcCellFwUpgrade. Please contact support to obtain the required upgrade packages.

Note: The cellular interface is disconnected from the network during the upgrade. Local access is required for streaming the firmware package!

4.12.2 Upgrading Cellular Module Firmware - Remote (FOTA)

The firmware of the mobile radio module can be updated over the air (FOTA). Remote access to the network via the mobile radio interface is mandatory. By default, all required update packages are provided by a Westermo web service (default URL). It is possible to set up your own web service in the private infrastructure if Internet access is not available. The upgrade can be started via the web interface, web API, CLI or SNMP.

Note: Ensure that the device has a SIM card with an active mobile service provider and reliable cellular signal coverage.

4.12.2.1 Upgrading Cellular Module Firmware via Web Interface

Login to the Web Interface of the device.

Once logged in choose menu entry **System - Cellular** to see the Cellular Manager page.

Items in section "Cellular Firmware Upgrade Over The Air (FOTA)" are as follows.

The fields **Update availability** and **Version** show if Cellular Module Firmware can be upgraded or if the preferred version is already installed.

The **APN** is automatically defined by the first WWAN interface `wwan0`. If set to `auto`, the cellular provider default APN will be used for accessing the upgrade servers.

The input field **URL** can be used to overwrite the default path to the public Westermo web service. This is required if the device is operated in a private network without Internet access. Please contact support for further instructions.

Note: Be careful overwriting the URL; the update server must be reachable. It can take several minutes, even hours, for the device to recover if an inaccessible URL is defined.

Click on the **Start FOTA Upgrade** button to download the upgrade file and start the Cellular Module Firmware upgrade.

If the device has a local connection, the progress of the upgrade is displayed in the status field. If there is a remote connection, the device cannot be reached during the upgrade and the status is not updated.

Field **Status** displays the FOTA upgrade status with the following typical values:

1. **FOTA inactive**: Initial FOTA status. No upgrade running.
2. **FOTA downloading**: Downloading Cellular Module Firmware upgrade package.
3. **FOTA updating: <percent>**: Cellular Module Firwmare upgrade in progress.
4. **FOTA completed**: After successful upgrade, just before rebooting the device.

Field **Last Error** displays the last FOTA upgrade error text with the following typical values:

1. **none**: Initial state, no error detected.
2. **download failed: <reason code>**: Failed to download the upgrade package.
3. **update failed: <reason code>**: Failed to upgrade the Cellular Module Firmware.

Reason Codes:

- **701**: Download not completed.
- **702**: URL length exceeds limit.
- **801**: Server not reachable.
- **5xx**: Unexptected update failure, please contact support.

4.12.2.2 Upgrading Cellular Module Firmware via SNMP

Check the hwCellFwVersionPreferred to see if the preferred Cellular Module Firmware version is in use (’true(1)’) or not (’false(0)’).

The following steps need a working cellular internet connection.

Initiate Cellular Module Firmware update and control it’s success.

1. Define a valid URL which is accessible by the device. This is done by changing the firmware URL parameter setCellFotaUrl in the settings section.
2. Writing ’upgrade(1)’ to rpcCellFotaUpgrade parameter will download and validate the new firmware file.
3. If the downloaded file is considered a valid firmware for this device, the Cellular Module Firmware udate process will start.

4. Reading rpcFwFlash parameter will report the status of the firmware flash process. A value of 'error(-1)' denotes that download or update process failed. A value of 'nop(0)' means that the Cellular Module Firmware update process is done.

Syslog and TRAPs will show upgrade success and error state.

The following TRAPs are available:

- Message Code: 825
- Message Code: 826
- Message Code: 827

After successful download and update of the Cellular Module Firmware, the device will automatically reboot.

4.12.2.3 Upgrading Cellular Module Firmware via CLI

Please refer to the CLI command `upgrade-cellular`, see chapter CLI Commands. The update process is similar to the SNMP approach and requires a working cellular internet connection from where the Cellular Module Firmware update package can be downloaded.

4.13 Certificate Store

The *Certificate Store (CS)* is responsible to handle all Certificates (CERT), Private Keys (KEY), Certificate Revocation Lists (CRL), Public Keys (PUBKEY) and Static Keys used by services on the device.

These services are:

- Webserver (HTTPS)
- 802.1X
- OpenVPN
- IPsec
- TLS Client
- LDAP

- Secure Syslog

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

Most 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.

Services reference certificates in CS via the identification number (CS Reference ID). The id of a file can be set during import to use either an auto generated id (next free id) or a well defined id choosen by the user. Please make sure the right id is set because any existing file will be overwritten.

The import process verify the files before they get stored in the CS.

Key files are always stored in encrypted format.

After a new file has been installed, the appropriate service needs to be restarted. This can be done by using the *Services* page, (see Web Interface), call the RPC through SNMP (see rpcCfgApply or by setup *Daily Refresh* (see Daily Refresh).

Remove a certificate from the CS is only possible if it's not used by a service.

4.13.1 Certificate Handling

4.13.1.1 Web Interface

The certificates in the CS can be managed through *Web Interface*. The *Certificate Management* page can be found in *System -> Certificate Management*.

The *Certificates* page can be used to upload certificate files, to see information about the installed certificate, download available certificates, edit the label of a certificate or to remove an existing certificate.

On the *Certificates* page it's possible to generate a CSR. During this process a new private key will be generated. This key will then be used to create a new CSR which can then be sent to a CA to get a signed certificate back. This signed certificate can then be installed on the device and used by the desired service.

The *Services* page show an overview of the services which are currently using a certificate and also information about the used certificate. The information for the same certificate (same id) on the *Services* page and the one on the *Certificates* page may differ. This happen if a new certificate was upload to a used id, but the service was not restarted yet. In this case the user gets notified on the *Services* page and the services can be restarted by using the *Apply* button.

4.13.1.2 SNMP

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

1. Define the CS type: setCrtFileUrl.
2. For import and export:
    - Define the URL of where the file is located: setCrtFileUrl.
    - Define the CS file format: setCrtFileFormat. For PKCS#12 containers this setting is ignored.
3. Define the CS identification number (ID): setCrtFileId.
4. For import of encrypted KEY and PKCS#12 container files:
    - Define the passphrase: setCrtFilePassphrase.
5. Start the process by setting rpcCrtFile to the desired action (import(1), export(2) or delete(3)).
6. Observe the process progress by polling rpcCrtFile.

4.13.1.3 Simple Certificate Enrollment Protocol (SCEP)

The following SCEP transactions are supported:

1. CA certificate retrieval
    a) Device is sending getca request to the configured SCEP server (cfgScepServerUrl).
    b) SCEP server will then provide a CA certificate which is going to be stored in CS.
    c) This CA certificate is then available for use by any service.
2. Certificate enrollment
    a) Set cfgScepCaId to define which CA certificate to use for SCEP enrollment.
    b) Device is generating a new private key (stored in CS with automatically chosen CS Reference ID.
    c) Device is sending a CSR to the configured SCEP server (cfgScepServerUrl).

d) Above generated private key and signed certificate from SCEP server and will then be stored in CS.

3. WLAN interface certificate renewal

a) Set cfgScepCertId to define which certificate to renew.

b) When the device is configured for automatic renewal (cfgScepAutoRenewEnabled) and cfgScepCertId is configured, then: when time to renewal of the device certificate is due (cfgScepAutoRenewTimeBeforeExpire):

i. Device is generating a new private key.

ii. Device is generating a new CSR (using active still valid device certificate and active still valid private key) and

iii. then sending this CSR to the configured SCEP server (cfgScepServerUrl).

iv. Signed certificate from SCEP server will then be stored in CS with ID cfgScepCertId in state *prepared*.

c) Apply the new certificates (see Web Interface, after install an new certificate).

4.13.1.4 SCEP Configuration

Normally a device needs to be configured once before using any SCEP transactions.

Basic configuration:

- cfgScepServerUrl.0 SCEP server URL (e.g. "http://<SCEP Server IP address>/cert/scep")

Configuration of enrollment:

- Configuration of Certificate Signing Request (CSR)
  - cfgScepCsrC CSR country name (2 letter code) (C)
  - cfgScepCsrST CSR state name (ST)
  - cfgScepCsrL CSR locality name (L)
  - cfgScepCsrO CSR organization name (O)
  - cfgScepCsrOU CSR organizational unit name (OU)

  - cfgScepCsrCN CSR common name (CN)

- cfgScepCaIdentifier SCEP CA Identifier
- cfgScepChallengePassword SCEP challenge password
- cfgScepPollingInterval SCEP polling interval (in seconds)
- cfgScepPollingMaxTries SCEP polling maximum number of tries
- cfgScepCaId CS Reference ID for CA certificate (used for enroll and renewal)
- Configuration of renewal (re-enrollment):
  - cfgScepCertId CS Reference ID for certificate to renew
  - Additional configuration for automatic renewal:
    * cfgScepAutoRenewEnabled Activate (enabled(1)) or deactivate (disabled(0)) SCEP automatic enrollment
    * cfgScepAutoRenewTimeBeforeExpire Define how many days before certificate expiration automatic re-enrollment shall be triggered
    * cfgScepAutoRenewRetryPeriod SCEP re-enrollment retry interval (in minutes)

4.13.1.5 SCEP Usage

After SCEP configuration is applied successfully (see above), the following RPCs are available for controlling SCEP transactions:

- Write value to rpcScep-item:
  - rpcScepGetCaCrt Write start(1) to start a GetCACert transaction
  - rpcScepEnroll Write start(1) to start a SCEP enrollment transaction
  - rpcScepEnroll Write reenroll(3) to start a SCEP renewal transaction
- Read value from rpcScep-item:
  - rpcScepGetCaCrt A value of done(0) denotes that the SCEP GetCACert transaction finished successfully. If read value is lower then 0, then there was an error in the SCEP GetCACert transaction. A value greater then 0 denotes that the corresponding RPC is still running (e.g. getca still in progress).

  - rpcScepEnroll A value of done(0) denotes that the SCEP Enroll transaction finished successfully. If read value is lower then 0, then there was an error in the SCEP entroll transaction. A value greater then 0 denotes that the corresponding RPC is still running (e.g. enroll still in progress). It is good practice to check the return value periodically until the process is done.

4.13.2 Services using Certificate Store

4.13.2.1 Web Server (HTTPS)

The Web Server for the *Web Interface* uses TLS by default. Therefore the following Certificate Store (CS) files are required:

- TLS Server Private Key (of type KEY).
- TLS Server Certificate (of type CERT). If using intermediate certificate(s) they should be part of the same file. Set the CS Reference ID with cfgHttpTlsServerCertId. The TLS Server automatically figures out which Private Key (KEY) belongs to this Server Certificate.

4.13.2.2 802.1X

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

- Client Certificate (CERT): Client x509 Certificate. Set CS Reference ID with cfgWlan802dot1xClientCertId.
- Client Key (KEY): Client Private Key matching the Client x509 Certificate.
- Certificate Authorities (CRTs): Authentication Server X509 Certificate(s). Set CS Reference IDs with cfgWlan802dot1xCaIds.
- Certificate Revocation List(s) (CRLs): CRL issued by the CAs. There can be at most one CRL per CA certificate. Set cfgWlan802dot1xTlsControlParams to 0x4 when not using any CRLs. When using multiple CAs, then one CRL per CA must be available (might be an empty CRL).

4.13.2.3 OpenVPN

The following files are available for the OpenVPN instances:

- Client Certificate. Set CS Reference ID with cfgVpnOpenvpnCertId.

- Client Key.
- CA Certificate(s). Set CS Reference ID with cfgVpnOpenvpnCaIds.
- Static Key. Set CS Reference ID with cfgVpnOpenvpnStaticKeyId.

4.13.2.4 IPsec VPN

The following files are available for the IPsec instances:

- CA Certificate(s). Set CS Reference ID with cfgVpnIpsecCaIds.
- Left Certificate. Set CS Reference ID with cfgVpnIpsecLeftCertId.
- Right Certificate. Set CS Reference ID with cfgVpnIpsecRightCertId.
- Left Signature Key: The left participant's public key. Set CS Reference ID with cfgVpnIpsecLeftSigkeyId.
- Right Signature Key: The right participant's public key. Set CS Reference ID with cfgVpnIpsecRightSigkeyId.
- Left Client Key: The left participant's private key. Set CS Reference ID with cfgVpnIpsecLeftKeyId.

4.13.2.5 TLS Client

The following CS certificates are available for the TLS Client:

- TLS CA Certificates(s). Set CS Reference ID(s) with setTlsCltCaIds.
- TLS CRLs (CRL that belongs to CA). Set setTlsCltTlsControlParams to 0x4 for using CA Certificates but not using CRLs. Use setTlsCltCrlExpiryExtension to extend the validity of CRLs.

TLS certificates normally contain the server name, not the IP address. But the subject alternative name field (subjectAltName) in the certificate can be used to include the IP address of the server. This then allows a successful secure connection using an IP address. The TLS CA Certificate must therefore contain a subjectAltName with the IP address of the HTTPS Server to which the TLS Client connects.

Use setTlsCltTlsCiphers to specify which ciphers to use for the TLS Client. See Ciphers and the OpenSSL documentation for a list of available ciphers and used syntax.

4.13.2.6 LDAP Client

Configure LDAP parameters in order to be able to use LDAP users for Web-Based Management (Web Interface).

Configure LDAP parameters in order

- Set cfgLdapEnabled.0 to enabled(1) to allow LDAP login
- Set cfgLdapUrl1.0 and optionally cfgLdapUrl2.0
- Set cfgLdapCaIds.0 to reference CAs to use for LDAPS
- Set cfgLdapAccessDn.0 to the Distinguished Name to bind for search for USER_DN
- Set cfgLdapAccessPassword.0 as password for USER_DN search
- Set cfgLdapAccessFilter.0 as filter for USER_DN search where %USER% is replaced with login username
- Set cfgLdapUserBaseDn.0 as the starting point for the ROLE_DN search
- Set cfgLdapUserRoleAttribute.0 to define which attribute (e.g. "memberOf") is used for the ROLE_DN
- Set cfgLdapAdminRoleDn.0 to defined ROLE_DN for admin role
- Set cfgLdapMonitorRoleDn.0 to defined ROLE_DN for monitor role
- Set cfgLdapRequestTimeout.0 to define how long to wait for LDAP search response

LDAP implementation is using two ldap searches and then tries to match the retrieved USER_ROLE with one of the configured Role DNs:

- Retrieve User's Distinguished Name (USER_DN)
  - binding with cfgLdapAccessDn and cfgLdapAccessPassword
  - and filtering with cfgLdapAccessFilter (which must include login username)
- Retrieve User Role (USER_ROLE)
  - binding with USER_DN (from first LDAP search) and login password
  - and filtering with cfgLdapUserRoleAttribute (e.g. "memberOf")

- Try to match with one of the three role DNs
  - if matches with cfgLdapAdminRoleDn, role is admin
  - if matches with cfgLdapMonitorRoleDn, role is monitor
  - else: login is denied

4.13.3 Ciphers

Supported authentication and encryption algorithms (ciphers and ciphersuites).

The TLS layer is provided by the openssl library, which supports TLSv1.2 and TLS1.3. Support for older TLS versions is explicitly disabled, since they are not recommended for usage.

Up to TLSv1.2, openssl uses ciphers, starting from TLSv1.3 it changed to ciphersuites.

Following cipher suites are supported:

AES128-GCM-SHA256	AES128-SHA
AES128-SHA256	AES256-GCM-SHA384
AES256-SHA	AES256-SHA256
DHE-RSA-AES128-GCM-SHA256	DHE-RSA-AES128-SHA
DHE-RSA-AES128-SHA256	DHE-RSA-AES256-GCM-SHA384
DHE-RSA-AES256-SHA	DHE-RSA-AES256-SHA256
DHE-RSA-CHACHA20-POLY1305	ECDHE-ECDSA-AES128-GCM-SHA256
ECDHE-ECDSA-AES128-SHA	ECDHE-ECDSA-AES128-SHA256
ECDHE-ECDSA-AES256-GCM-SHA384	ECDHE-ECDSA-AES256-SHA
ECDHE-ECDSA-AES256-SHA384	ECDHE-ECDSA-CHACHA20-POLY1305
ECDHE-RSA-AES128-GCM-SHA256	ECDHE-RSA-AES128-SHA
ECDHE-RSA-AES128-SHA256	ECDHE-RSA-AES256-GCM-SHA384
ECDHE-RSA-AES256-SHA	ECDHE-RSA-AES256-SHA384
ECDHE-RSA-CHACHA20-POLY1305	

Whereas for TLSv1.3 support is provided with these ciphersuites:

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

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

Permitted cipher suites for WPA3-Enterprise mode:

- ECDHE-ECDSA-AES256-GCM-SHA384
  ECDHE and ECDSA using the 384-bit prime modulus curve P-384
- ECDHE-RSA-AES256-GCM-SHA384
  ECDHE using the 384-bit prime modulus curve P-384
  RSA $\geq$ 3072-bit modulus
- DHE-RSA-AES256-GCM-SHA384
  RSA $\geq$ 3072-bit modulus
  DHE $\geq$ 3072-bit modulus

The certificate issuer is responsible to use appropriate key length for WPA3 - Enterprise as listed above.

4.13.4 Monitoring/Jobs

4.13.4.1 Certificate Expiration Monitoring

Expiration alert monitors can be defined for certificates and CRLs. Such a monitor will alert by a Trap/Syslog Message Code: 444 (for certificates) or Message Code: 445 (for CRLs) a configurable amount of days before certificate/CRL expiration (cfgCrtMonTimeBeforeExpire). The Trap/Syslog will be repeated every cfgCrtMonRepeatPeriod hours.

For certificates the field 'enddate' is used to decide when the certificate expires. For CRLs the field 'nextupdate' is used to decide when the CRL expires.

The following configuration parameters are available:

1. cfgCrtMonEnabled
2. Certificate monitoring:
    a) cfgCrtMonId - Certificate ID in the Certificate Store
    b) cfgCrtMonType - set to cert(2)
    c) cfgCrtMonTimeBeforeExpire - amount of days before certificate expiration
    d) cfgCrtMonRepeatPeriod - repeat period in hours
3. CRL monitoring:
    a) cfgCrtMonId - Certificate ID of the associated CA certificate in the Certificate Store. Please note that the CA ID needs to be referenced (*not* the CRL ID).
    b) cfgCrtMonType - set to crl(1)
    c) cfgCrtMonTimeBeforeExpire - amount of days before certificate/CRL expiration
    d) cfgCrtMonRepeatPeriod - repeat period in hours

4.13.4.2 Daily Refresh

New certificates which are installed to the device are not used immediately. The services are using a runtime copy of the certificate so the related services need to be restarted.

This can be done automatically once per day by configure the *Daily Refresh*. To enable this refresh the cfgCrtGlblDailyRefreshEnabled needs to be set to **enabled(1)** and the point in time can be set

with cfgCrtGlblDailyRefreshTime. If this time is reached the services gets restarted if there is a new certificate for the particular service.

4.13.4.3 Automatic CRL Download

CRL download is carried out through HTTP download.

The triggering of automatic CRL download can be configured number of days before CRL expiration: cfgCrtCrlTimeBeforeExpire). If the automatic download of a CRL fails, the download is retried after the configured amount of minutes (cfgCrtCrlRetryPeriod).

Configuration of automatic CRL download:

1. cfgCrtCrlEnabled
2. cfgCrtCrlCaId: Reference to the CA ID in the certificate store. The received CRL will be stored to the CRL ID of the CRL associated to the configured CA ID.
3. cfgCrtCrlUrl: Define URL for CRL download
4. cfgCrtCrlTimeBeforeExpire
5. cfgCrtCrlRetryPeriod

The CRLs downloaded from HTTP server are stored in the Certificate Store. To use the new CRLs the related services needs to be restarted (see Web Interface, after install an new certificate).

kMibNameRefrpcCrtCrlGet is provided for manually starting CRL download (using cfgCrtCrlUrl and cfgCrtCrlCaId).

4.14 System

The hostname of 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 either in parallel or sequentially which may be selected with cfgSysNameserverOrder. When querying sequentially, the timeout is 2 seconds. When querying in parallel the timeout is 5 seconds. Either way, 2 attempts are performed.

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.

cfgSysNsType may be configured to ignoreinterface(3), which prevents nameservers received via DHCP to be used.

A maximum of 6 entries are queried. When more entries are defined, the additional ones are ignored.

4.15 Network Configuration

4.15.1 Interfaces and Bridges

IbexOS 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:

```
# 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
```

Listing 4.1: *Interface configuration*

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.

4.15.2 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):

```
# 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
```

Listing 4.2: *VLAN configuration*

The VLAN configuration parameters are described in cfgNetVlanTable.

4.15.3 MAC VLAN Interfaces

The MAC VLAN configuration parameters are described in cfgNetMacVlanTable.

4.15.4 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):

```
# 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
```

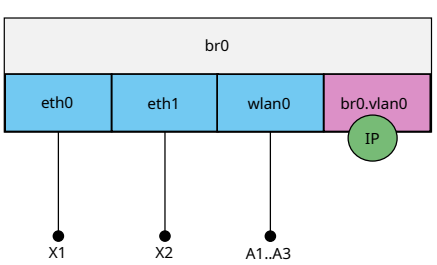
Listing 4.3: *IP address configuration*

The IP configuration parameters are described in cfgNetIpTable.

4.15.5 Network Configuration Examples

Example 1: Access Point Network Configuration

The default configuration of the access point is in bridged mode, ethernet and wireless interfaces are located in the same bridge `br0`. To assign a static IP address to the bridge, a VLAN on the bridge `br0.vlan0` must be defined and entered into the IP table. The VLAN ID 0 represents untagged traffic. All interfaces in the same bridge share the same IP address. The trunk mode of the ethernet and wireless interfaces is set to -1 to allow all VLAN tagged and untagged traffic.



ETH / WLAN Table

Name	eth0	eth1	wlan0
Enabled	enabled	enabled	enabled
Bridge	0	0	0
Trunk	-1	-1	-1
Tag	N/A	N/A	N/A
VlanMode	trunk	trunk	trunk

VLAN Table

Name	br0.vlan0
Enabled	enabled
Bridge	0
Vid	0 (untagged)
Parent	N/A

IP Table

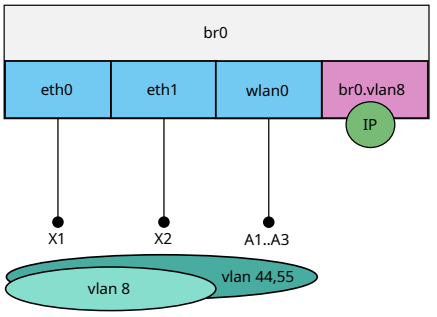
Enabled	enabled
Addr[1]	192.168.1.20/24
Proto	0
Interface	br0.vlan0

1) CIDR Notation

Figure 4.1: *Access Point Network Configuration Example*

Example 2: Access Point VLAN Network Configuration (Trunk Mode)

The bridge configuration is the same as in Example 1. In addition, the trunk of the bridged ethernet interfaces is limited to the VLAN ID's 8, 44 and 55. To assign a static IP address to a single VLAN, define the VLAN interface `br0.vlan8` (e.g. for management purposes). To restrict management access through an interface, VLAN ID 8 can be removed from the trunk. Removing VLAN ID 8 from `wlan0` allows access for local management only. The data VLAN's 44 and 55 are bridged between ethernet and wireless.



ETH / WLAN Table

Name	eth0	eth1	wlan0
Enabled	enabled	enabled	enabled
Bridge	0	0	0
Trunk	8,44,55	8,44,55	44,55
Tag	N/A	N/A	N/A
VlanMode	trunk	trunk	trunk

VLAN Table

Name	br0.vlan8
Enabled	enabled
Bridge	0
Vid	8
Parent	N/A

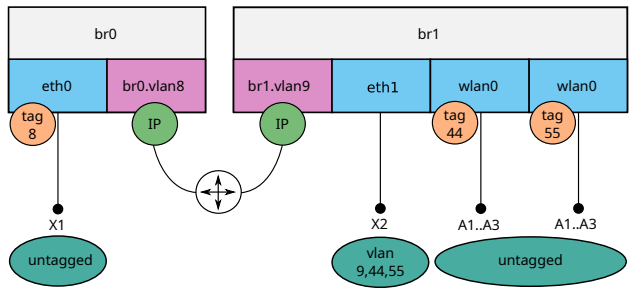
IP Table

Enabled	enabled
Addr	192.168.3.22/24
Proto	0
Interface	br0.vlan8

Figure 4.2: *Access Point VLAN Network Configuration Example (Trunk Mode)*

Example 3: Access Point VLAN Network Configuration (Access Mode)

This example refers to the network configuration of the hotspot application, please see Public Wireless Network (PWN). The Ethernet interfaces are separated for local and remote management and also for data traffic. The interface `eth0` is in VLAN access mode, only untagged ingressing frames are accepted and will be tagged with VLAN ID 8. Egressing frames will have no VLAN header and are untagged. The same applies for the wireless interfaces `wlan0` and `wlan1` where ingressing traffic is tagged with VLAN ID 44 respectively 55. The ethernet interface `eth1` on `br1` is in VLAN trunk mode and limited to the VLAN ID's 9, 44 and 55.



ETH / WLAN Table

Name	eth0	eth1	wlan0	wlan1
Enabled	enabled	enabled	enabled	enabled
Bridge	0	1	1	1
Trunk	N/A	9,44,55	N/A	N/A
Tag	8	N/A	44	55
VlanMode	access	trunk	access	access

VLAN Table

Name	br0.vlan8	br1.vlan9
Enabled	enabled	enabled
Bridge	0	1
Vid	8	9
Parent	N/A	N/A

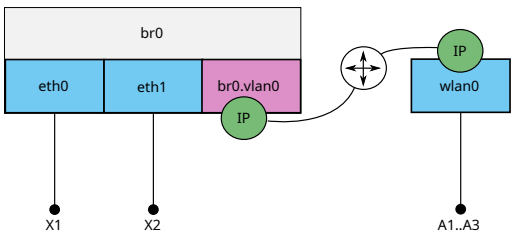
IP Table

Enabled	enabled	enabled
Addr	192.168.1.20/24	192.168.100.20/24
Proto	0	0
Interface	br0.vlan8	br1.vlan9

Figure 4.3: *Access Point VLAN Network Configuration Example (Access Mode)*

Example 4: Client (STA) Network Configuration 1

In wireless client mode, the ethernet interfaces share the same bridge `br0`. A static IP address is assigned to this bridge with the untagged VLAN `br0.vlan0`. A wireless interface in normal mode cannot be assigned to the same bridge as the ethernet interfaces, the value must be set to -1. The IP address of the wireless interface is assigned directly to `wlan0`. To communicate directly with the AP, the wireless interfaces must share the same subnet. The trunk mode of the ethernet interfaces is set to -1 to allow all VLAN tagged and untagged traffic.



ETH / WLAN Table

Name	eth0	eth1	wlan0
Enabled	enabled	enabled	enabled
Bridge	0	0	-1
Trunk	-1	-1	N/A
Tag	N/A	N/A	N/A
VlanMode	trunk	trunk	N/A

VLAN

Name	br0.vlan0
Enabled	enabled
Bridge	0
Vid	0 (untagged)
Parent	N/A

IP Table

Enabled	enabled	enabled
Addr	192.168.2.11/24	192.168.3.11/24
Proto	0	0
Interface	br0.vlan0	wlan0

Figure 4.4: *Client (STA) Network Configuration Example 1*

Example 5: Client (STA) Network Configuration 2

To split the local and remote management IP addresses to different VLAN ID's, the wireless interface must also be added to a bridge `br1`. In this example, local management access is ensured over VLAN ID 8, remote management access over VLAN ID 9. The management VLAN ID is added to the trunks according to the interfaces; the IP addresses are assigned to `br0.vlan8` and `br1.vlan9`. The data VLAN with the ID 44 is routed between the ethernet and wireless interfaces. Different IP addresses are assigned on `br0.vlan44` and `br1.vlan44` for local and remote data access. All untagged traffic is discarded.

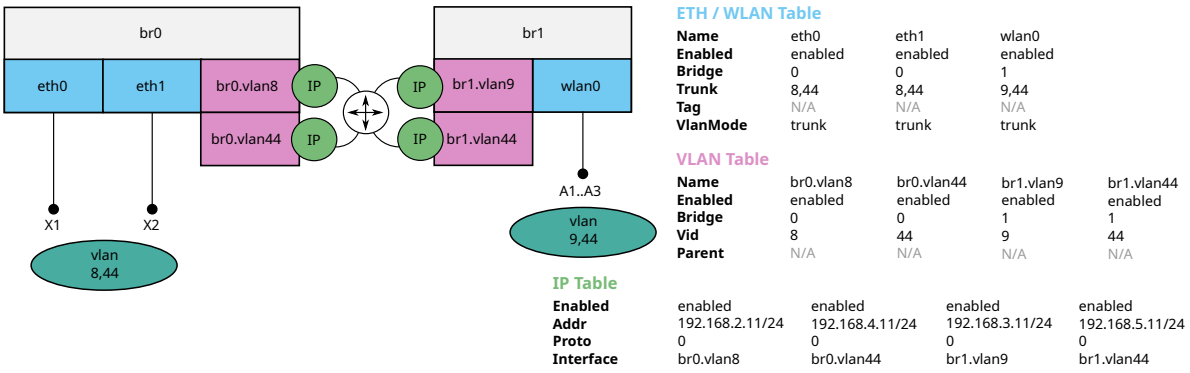


Figure 4.5: *Client (STA) Network Configuration Example 2*

4.16 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.

4.17 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

4.17.1 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:

Device	2.4 GHz	5 GHz	6 GHz	Monitor
Ibex-RT-220	radio0	radio0		
Ibex-RT-320	radio0	radio0		
Ibex-RT-280	radio0	radio0		radio1
Ibex-RT-370	radio0	radio0		radio1
Ibex-RT-630	radio0	radio0		
Ibex-RT-630-5G	radio0	radio0		
Ibex-1310	radio1	radio0		
Ibex-1510	radio1	radio0		
Ibex-3510	radio1	radio0	radio2	
Ibex-1520	radio0	radio1		
Ibex-3520	radio0	radio1	radio2	

Table 4.4: *WLAN radio frequency allocation*

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.

4.17.1.0.1 Bandwidth sets the Wireless Bandwidth Mode by specifying the bandwidth of the channel. Please refer to cfgWlanDevBandwidth for more information.

4.17.1.0.2 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.

4.17.1.0.3 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.

4.17.1.0.4 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.

4.17.1.0.5 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.

4.17.1.0.6 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 (cfgWlanIfaceBitrates) MCS0 to MCS7 can be used. With two chains enabled, the 1- and 2-stream bitrates (cfgWlanIfaceBitrates) 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).

4.17.2 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.

4.17.2.0.1 Operating Mode of the WLAN interface can be set with cfgWlanIfaceMode. 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.

4.17.2.0.2 Service Set Identifier (SSID) of the wireless interface is set with cfgWlanIfaceSsid. SSID is the arbitrary name of the wireless network this interface is part of.

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

4.17.2.0.4 Bitrate Limitations can be set with cfgWlanIfaceBitrates. 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.

4.17.2.0.5 Enabling Wireless Multimedia Extensions (WME) for the interface is possible with cfgWlanIfaceWmeEnabled. 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 cfgWlanIfaceWmeParameter. For more information about WME tables please check Quality of Service (QoS).

4.17.2.0.6 MAC Address Access Control List (ACL) mode for the interface can be set in cfgWlanIfaceMacaddrAcl. 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.

4.17.2.0.7 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 cfgWlanIfaceScanList. 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.

4.17.2.1 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:

- cfgWlanIfaceIgnoreBroadcastSsid
- cfgWlanIfaceUseVendorSsid

Setting cfgWlanIfaceIgnoreBroadcastSsid 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 cfgWlanIfaceIgnoreBroadcastSsid 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, cfgWlanIfaceUseVendorSsid must be used:

- On an AP: set cfgWlanIfaceUseVendorSsid to enabled(1) so that the hidden SSID is added as vendor element in the AP beacon.
- And on a STA: set cfgWlanIfaceUseVendorSsid 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.

4.18 Cellular Network Configuration

Cellular communication is only available on specially designed products, see also section 1.4. The configuration is divided into different sections:

- **Interface Configuration** (cfgNetWwanTable),
- **Device Configuration** (cfgCellDeviceTable),
- **SIM Slot Configuration** (cfgCellSimSlotTable),
- **Default Bearer Configuration** (cfgCellDefaultBearerTable), and
- **Connection Management** (cfgCellConnectionManagement).

4.18.1 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. Several cellular interfaces can be created. The assignment of the SIM slots and the configuration of the default bearer profiles are described in the following chapters.

4.18.2 Device Configuration

This table contains the global configuration parameters of the cellular modems.

4.18.2.1 Search Mode

The parameter cfgCellDeviceSearchMode allows to limit the cellular technologies. By default, all available technologies, depending on the cellular module, are enabled. To limit the connection to a single or multiple technologies, an integer value according to the bitmask settings can be defined, e.g.:

- **LTE only**: 0x02 = 2
- **5G only**: 0x04 = 4
- **LTE and 5G**: 0x02 + 0x04 = 6

4.18.2.2 Operator Selection

Connection attempts can be limited to a single service provider. This is useful if roaming is to be prohibited and the device must operate close to the country border. It is possible that the modem will attempt to connect to a roaming cell if it has better signal levels. If cfgCellDefaultBearerRoaming is disabled, the connection will be terminated and reestablished with the local service provider. This takes extra time. If cfgCellDeviceOperatorMode is set to `fix(1)`, the modem will only access the service provider network which is defined with the parameter cfgCellDeviceOperatorSpn. The SPN (Service Provider Name) is specified in swCellServiceName, or when roaming, the home network name can be determined using the information in swCellServiceMcc and swCellServiceMnc.

Note: If the service provider name is not defined or is incorrect, the modem cannot connect to the mobile network and remote access is lost.

4.18.2.3 Band Configuration

The cellular bands can be limited for each technology, e.g. cfgCellDeviceBandsLte. The required bands can be specified in a space and/or comma separated list. This is useful if the service provider offers several frequency bands, but not all frequencies guarantee good coverage. This may be the case if a private APN is used that is not supported on all frequency bands and the service provider also offers additional frequencies for public use.

Note: The available bands depend on the cellular module, please ask support for more information.

4.18.2.4 5G Configuration

The 5G mode can be enforced to Stand-Alone (SA) or Non-Stand-Alone (NSA): cfgCellDevice5gMode

4.18.3 SIM Slot Configuration

Each physical SIM slot is listed in the cfgCellSimSlotTable. By default, all SIM slots are enabled. SIM slots without SIM card can be disabled. In order to use SIM cards with actived SIM lock, cfgCellSimSlotPinEnabled must be enabled and a valid 4-digit cfgCellSimSlotPin must be entered. If more then one SIM slot is enabled, initial connection attempts will be made with the higher prioritized SIM slot. The SIM slot with the smaller value of cfgCellSimSlotPriority has the higher priority. If both SIM slots have the same priority, the SIM slot with the smaller index is preferred.

If the SIM card cannot be unlocked, increasing the unlock timeout cfgCellSimSlotUnlockTimeout will enable support for older and slower SIM cards. This value affects the startup speed and the performance of the SIM rotation, keep it low if possible. Changes become effective after restarting the device.

4.18.4 Default Bearer Configuration

Adding a new cellular interface in cfgNetWwanEnabled creates a default bearer entry for that interface. Each default bearer must be assigned to one or more SIM slots in the field cfgCellDefaultBearerSim-Slots. The SIM slot is identified by the name in cfgCellSimSlotName. If more than one SIM slot is to be added, the list of SIM slot names must be separated by spaces and/or commas.

The Access Point Name (APN) cfgCellDefaultBearerApn is set to `auto` by default. The cellular interface will establish a connection to the default profile of the cellular provider's network if this is supported. In case of connection problems or when using multiple default bearer interfaces on the same SIM slot, it is recommended to set the APN according to the specifications of the cellular provider. SIM cards that enforce authentication can be configured in cfgCellDefaultBearerUsername and cfgCellDefaultBearerPassword. Setting the authentication type cfgCellDefaultBearerAuthType is mandatory in this case.

Note: The APN value `auto` is only allowed on interface `wwan0`. If more than one WWAN interface is activated or SIM cards from different providers are used, it is mandatory that specific APN's are set.
Node: Roaming is enabled by default and can be deactivated using cfgCellDefaultBearerRoaming. Activated roaming may incur additional costs!

4.18.4.1 Cellular Router as Gateway

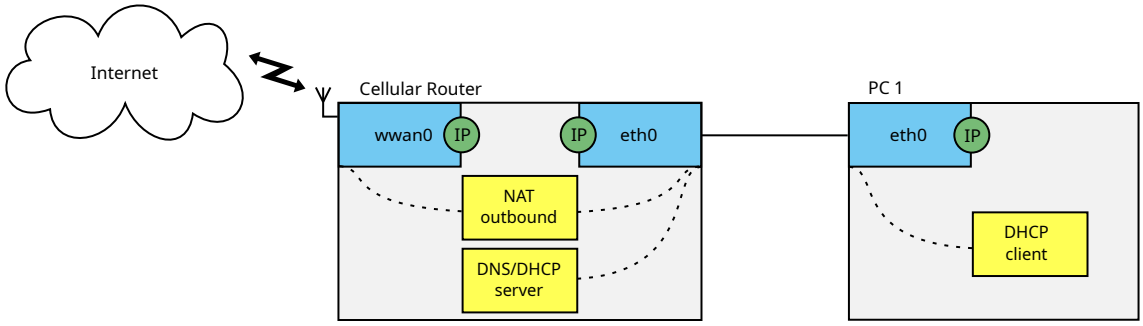


Figure 4.6: *Cellular Router as Internet Gateway*

The configuration example in Listing 4.4 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 4.52.1.2. Connect a PC with a DHCP client to one of the Ethernet interfaces of the Cellular Router. The SIM card must be inserted in SIM slot 1. The APN is selected automatically.

```
WESTERMO-SW6-MIB::cfgSysHostname.0 = MR-Gateway
WESTERMO-SW6-MIB::cfgNetWwanEnabled.0 = 1
WESTERMO-SW6-MIB::cfgCellSimSlotEnabled.0 = 1
WESTERMO-SW6-MIB::cfgDhcpGlobalEnabled.0 = 1
```

Listing 4.4: *Configuration of a Cellular Router as Gateway*

Outbound NAT (SNAT) is enabled by default for all cellular interfaces `wwanX`.

```
WESTERMO-SW6-MIB::cfgFwNatOutEnabled.0 = 1
WESTERMO-SW6-MIB::cfgFwNatOutInterface.0 = wwan+
```

Listing 4.5: *Default Configuration of Outbound NAT on a Cellular Router as Gateway*

Note: By default, the firewall is enabled to perform NAT and stateful filtering, but no default rules are defined. Please refer to chapter Firewall to prevent leaking traffic to the internet.

4.18.4.2 Multi Bearer Support for QoS

Cellular interfaces can be routed and filtered and support NAT. To enable QoS support for different default bearers of the same cellular provider, multiple cellular interfaces can be created in cfgNetWwanTable. The assigned provider network is identified via the cfgCellDefaultBearerApn. The service type of an APN and its QoS prioritization is defined in the provider specifications. The mobile router inherits the settings per interface according to the cellular base station settings.

The following example in Listing 4.6 creates two dedicated interfaces with interface based routing.

```
WESTERMO-SW6-MIB::cfgSysHostname.0 = MR-QoS
WESTERMO-SW6-MIB::cfgNetWwanEnabled.0 = 1
WESTERMO-SW6-MIB::cfgNetWwanEnabled.1 = 1
WESTERMO-SW6-MIB::cfgCellSimSlotEnabled.0 = 1
WESTERMO-SW6-MIB::cfgCellDefaultBearerSimSlots.0 = slot1
WESTERMO-SW6-MIB::cfgCellDefaultBearerSimSlots.1 = slot1
WESTERMO-SW6-MIB::cfgCellDefaultBearerApn.0 = apn_1
WESTERMO-SW6-MIB::cfgCellDefaultBearerApn.1 = apn_2
```

Listing 4.6: *Configuration of a Cellular Router with multiple Default Bearers*

4.18.4.3 Multiple SIM Cards Configuration Examples

This example is the configuration of a cellular router with two identical SIM cards inserted. The same default bearer settings can be used for both. SIM rotation must be enabled in the connection management configuration and the connection manager will switch between `slot1` and `slot2` starting with the higher priority.

```
WESTERMO-SW6-MIB::cfgCellSimSlotEnabled.0 = 1
WESTERMO-SW6-MIB::cfgCellSimSlotEnabled.1 = 1
WESTERMO-SW6-MIB::cfgCellSimSlotPriority.0 = 0
WESTERMO-SW6-MIB::cfgCellSimSlotPriority.1 = 100
WESTERMO-SW6-MIB::cfgCellDefaultBearerSimSlots.0 = slot1, slot2
WESTERMO-SW6-MIB::cfgCellDefaultBearerApn.0 = auto
WESTERMO-SW6-MIB::cfgCellConnMgmtSimRotationEnabled.0 = 1
```

Listing 4.7: *Configuration of a Cellular Router with identical SIM Cards as Backup*

The next example shows the configuration entries for a cellular router configuration with two different SIM cards. It is assumed that both SIM cards are from another country and do not allow roaming. When the router is moved across a border, the current SIM card will lose connection and the connection manager switches to the secondary SIM card. If the second SIM card is now in its country of origin, it will establish a connection.

```
WESTERMO-SW6-MIB::cfgCellSimSlotEnabled.0 = 1
WESTERMO-SW6-MIB::cfgCellSimSlotEnabled.1 = 1
WESTERMO-SW6-MIB::cfgCellSimSlotPriority.0 = 0
WESTERMO-SW6-MIB::cfgCellSimSlotPriority.1 = 100
WESTERMO-SW6-MIB::cfgCellDefaultBearerSimSlots.0 = slot1
WESTERMO-SW6-MIB::cfgCellDefaultBearerSimSlots.1 = slot2
WESTERMO-SW6-MIB::cfgCellDefaultBearerApn.0 = apn_1
WESTERMO-SW6-MIB::cfgCellDefaultBearerApn.1 = apn_2
WESTERMO-SW6-MIB::cfgCellDefaultBearerRoaming.0 = 0
WESTERMO-SW6-MIB::cfgCellDefaultBearerRoaming.1 = 0
WESTERMO-SW6-MIB::cfgCellConnMgmtSimRotationEnabled.0 = 1
```

Listing 4.8: *Configuration of a Cellular Router with differenct SIM Cards to avoid Roaming Costs*

4.18.5 Connection Management Configuration

The Connection Management controls which SIM slot is to be used. If SIM Rotation cfgCellConnMgmtSimRotationEnabled is enabled, the Connection Management toggles between the enabled SIM slots soon as the existing connection is considered lost. The connection loss detection mode is selected by cfgCellConnMgmtMonMode.

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

Note: Connection loss detection also operates when cfgCellConnMgmtSimRotationEnabled is disabled. The connection is re-established with the same SIM slot.

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 established within a certain time, which is hardware dependent and can take up to one minute, the Connection Management changes the SIM slots until connection could be established successfully. However, the monitoring of the new connection does not start until it is up and running.

Note: The SIM Slot can be selected manually during runtime by the setting setCellSimSlot. It is necessary to know the current SIM slot to be able to switch to the next one.

4.18.5.1 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.

4.18.6 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 Code: 360 and Message Code: 361.

4.19 Tunnel Endpoint Configuration

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

Supported protocols are:

Protocol	Encypsulation Overhead	Bridgeable
GRE	24 bytes	no
GRETAP	38 bytes	yes
VXLAN	50 bytes	yes

4.19.1 Interface Configuration

Each Tunnel Endpoint interface in the table cfgNetTunnelEndPointTable is managed by a separate tunnel instance. These instances are created and enabled via the table cfgVpnTunnelEndPointTable. The indices of both tables refer to the same instance.

cfgNetTunnelEndPointTable is used to configure generic network related parameters like the MTU of the interface, bridge assignment, or vlan.

cfgVpnTunnelEndPointTable is used to configure tunnel specific parameters, like the receive/transmit key or address of the remote peer.

4.19.2 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 conjuction 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 4.7 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).

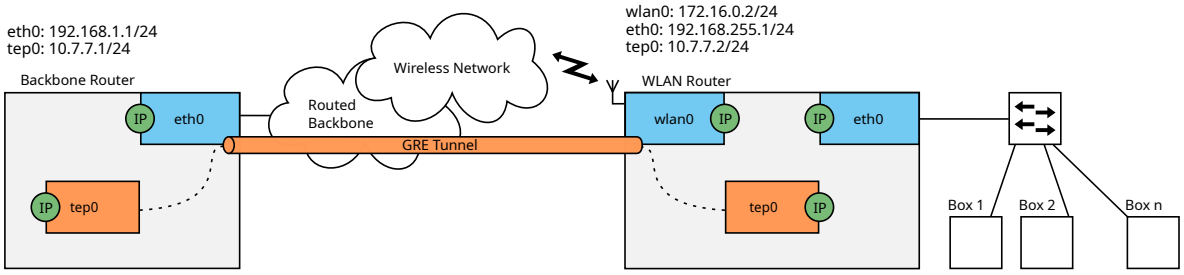


Figure 4.7: *GRE in a Wireless Network*

The relevant configuration items for the WLAN Router in Figure 4.7 to create a GRE tunnel, are described in the Listing 4.9:

```
WESTERMO-SW6-MIB::cfgNetTepEnabled.0 = 1
WESTERMO-SW6-MIB::cfgNetTepBridge.0 = -1
WESTERMO-SW6-MIB::cfgVpnTepTunnelType.0 = 0
WESTERMO-SW6-MIB::cfgVpnTepSource.0 = 0.0.0.0
WESTERMO-SW6-MIB::cfgVpnTepDestination.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 4.9: *TEP Interface Configuration for GRE*

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

cfgVpnTepTos may be used to specify the TOS of the GRE frames. When it is set to `inherit` the TOS value of the outer IP header is used. The key ID's cfgVpnTepRxKeyId and cfgVpnTepTxKeyId identify the header frame to allow parallel operation of multiple tunnels. Therefore, the RX and TX key ID's of the two endpoints must match. The same applies to GRETAP.

4.19.3 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 4.8 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.

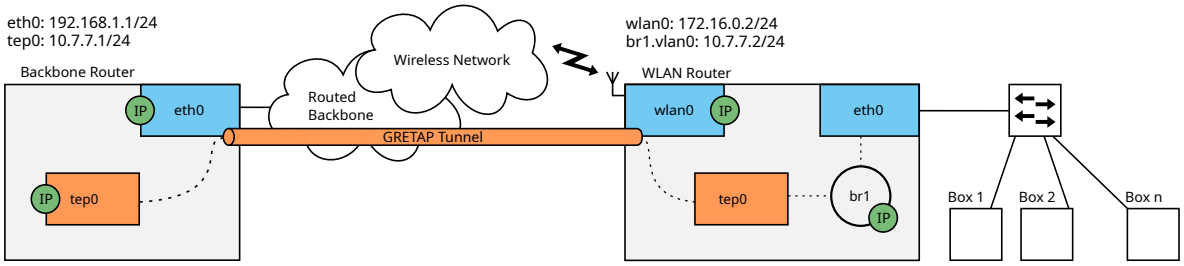


Figure 4.8: *GRETAP in a Wireless Network*

The essential configuration items for the WLAN Router in Figure 4.8 to create a GRETAP tunnel, are described in the Listing 4.10:

```
WESTERMO-SW6-MIB::cfgNetTepEnabled.0 = 1
WESTERMO-SW6-MIB::cfgNetTepBridge.0 = 1
WESTERMO-SW6-MIB::cfgVpnTepTunnelType.0 = 1
WESTERMO-SW6-MIB::cfgVpnTepSource.0 = 172.16.0.2
WESTERMO-SW6-MIB::cfgVpnTepDestination.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 4.10: *TEP Interface Configuration for GRETAP*

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 cfgVpnTepSource is set to `172.16.0.2`, binding the source to a specific address.

4.19.4 Virtual Extensible LAN (VXLAN)

Virtual Extensible LAN (VXLAN) is a network virtualization technology that attempts to address the scalability problems of large deployments. It uses a VLAN-like encapsulation technique to encapsulate layer 2 Ethernet frames within layer 4 UDP datagrams.

VXLAN is an evolution of efforts to standardize on an overlay encapsulation protocol. It provides functionality similar to GRE/GRETAP (tunnel) and VLAN (network overlay). Compared to VLAN which provides limited number of layer-2 VLANs (typically using 12-bit VLAN ID), VXLAN increases scalability up to 16 million logical networks (with 24-bit VNID) and allows for layer-2 adjacency across IP networks. Multicast or unicast with head-end replication (HER) is used to flood Broadcast, unknown-unicast and multicast traffic.

IbexOS only implements the tunnel (unicast) aspect of VXLAN. The network overlay functionality (multicast) is not supported at this time. Please contact our support if you require the network overlay functionality of VXLAN.

Every inner Ethernet frame is prepended with a VXLAN header and then encapsulated within an outer IPv4 UDP frame. The resulting overhead is 50 bytes (8 bytes VXLAN header + 14 bytes inner Ethernet header + 20 bytes outer IPv4 header + 8 bytes outer UDP header).

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

The scenario in Figure 4.9 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.

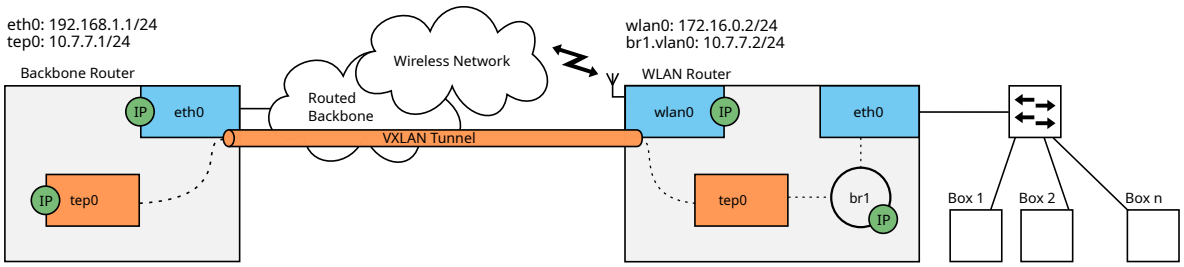


Figure 4.9: *VXLAN in a Wireless Network*

The essential configuration items for the WLAN Router in Figure 4.9 to create a VXLAN tunnel, are

described in the Listing 4.11:

```
WESTERMO-SW6-MIB::cfgNetTepEnabled.0 = 1
WESTERMO-SW6-MIB::cfgNetTepBridge.0 = 1
WESTERMO-SW6-MIB::cfgVpnTepTunnelType.0 = 2
WESTERMO-SW6-MIB::cfgVpnTepSource.0 = 172.16.0.2
WESTERMO-SW6-MIB::cfgVpnTepDestination.0 = 192.168.1.1
WESTERMO-SW6-MIB::cfgVpnTepVnid.0 = 100
WESTERMO-SW6-MIB::cfgVpnTepDestinationPort.0 = 4789

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 4.11: *TEP Interface Configuration for VXLAN*

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 cfgVpnTepSource is set to `172.16.0.2`, binding the source to a specific address. cfgVpnTepVnid defines the Virtual Network ID of VXLAN. The resulting encapsulated frames are send to the destination in cfgVpnTepDestination as UDP frames on the port 4789 as defined in cfgVpnTepDestinationPort.

4.20 IPsec Configuration

IbexOS provides virtual private network (VPN) support via IPsec VPN. The device can act as a VPN gateway in mainly in NETWORK-NETWORK scenarios.

IPsec VPN tunnels can be used to securely connect hosts and networks over the Internet. IPsec offers high level security and best performance over all VPNs due to HW acceleration.

4.20.1 Interface Configuration

Each IPsec interface in the table cfgNetIpsecTable is managed by a separate instance of IPsec. These instances are created and enabled via the table cfgVpnIpsecTable. The indices of both tables refer to the same instance.

cfgNetIpsecTable is used to configure generic network related parameters like the MTU of the interface.

cfgVpnIpsecTable is used to configure IPsec specific parameters, like the used encryption or address of remote peers.

4.20.2 Introduction to IPsec VPNs

A common use case for IPsec VPNs is to connect two networks via a secure tunnel over the Internet. We refer to this scenario as NETWORK-NETWORK VPNs. It is accomplished by having two VPN gateways, one at each site, negotiate and establish a secure tunnel, and to forward all traffic between the two networks through this tunnel. Creating VPN tunnels establishes a secure overlay network on top of the regular Internet connections.

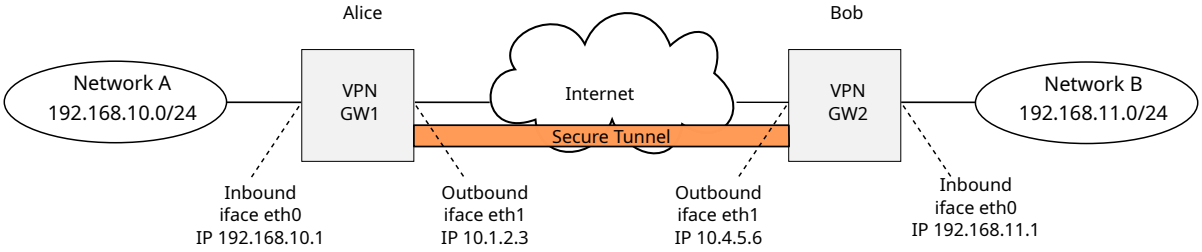


Figure 4.10: *IPsec VPN Sample Topology*

Fig. 4.10 shows a sample IPsec VPN topology used to help describe IPsec terminology and to illustrate some IPsec VPN configurations (see 4.20.6).

Explanation of some VPN related terminology:

- **Peers**: The two VPN gateways (Alice and Bob) are referred to as IPsec peers. The peers constitute the end-points of the secure tunnel. One of the peers will take the role of tunnel initiator and the other takes the responder role.

- **Initiator and Responder**: The VPN initiator is the peer that is responsible for initiating the tunnel establishment by contacting the other peer - the responder. A device is able to act both as responder and as initiator. Use cfgVpnIpsecAuto = "start" to configure a device as VPN initiator. In some cases, both peers can be configured as initiator at the same time.

- **NAT-traversal, Peer IP addresses**: In order to act as a responder, Alice must be assigned a public (routable) IP address on its interface towards the Internet. Thus, Alice generally cannot be located behind a NAT gateway, since the initiator (Bob) would not be able to initiate the tunnel. Bob will need to know Alice's IP address (or domain name) in order to know where to send the tunnel establishment messages. If Alice is assigned a fixed IP address, Bob can choose between using Alice's IP address or her domain name. But if Alice gets her address dynamically (e.g., via DHCP), Bob should use her domain name to establish the contact. The initiator (Bob) does not need to be assigned a public IP address. Bob is able to establish the tunnel even if he is located behind a NAT gateway. Furthermore, it is not mandatory for Alice to

know Bob's IP address beforehand. It is possible to configure the VPN tunnel such that Bob could connect to the Internet at various locations and still be able to establish the VPN tunnel. This is commonly referred to as Bob being a road warrior. The IKEv2 protocol includes NAT traversal (NAT-T) in the core standard. There is no need for any special configuration. If a NAT situation is detected, the client switches to UDP port 4500 (if it used port 500 initially) to send IKE_AUTH requests and UDP encapsulation will be activated for IPsec SAs.

- **Local and Remote Subnet**: Each peer will define what traffic should be allowed to pass through the established tunnel. Each peer will define the local and remote subnet, and all traffic between these subnets is sent securely through the tunnel. To secure all traffic between networks "A" and "B", Alice would define 192.168.10.0/24 as local subnet, and 192.168.11.0/24 as remote subnet in the tunnel configuration. Bob would do the opposite, i.e., define 192.168.11.0/24 as local subnet, and 192.168.10.0/24 as remote subnet. More advanced settings for the local and remote subnet parameters are possible, e.g., it is possible to configure the tunnel so that all traffic from Network B is sent through the tunnel (i.e., not only the traffic heading for Network A).

- **Outbound Interface**: The outbound interface denotes the interface, and implicitly the IP address, a VPN gateway uses to tunnel the traffic through, and to communicate with its peer. In fig. 4.10 Alice's outbound interface would be her interface towards the Internet (and the same goes for Bob). By default, the outbound interface is set to the interface leading to the default gateway.

4.20.3 Authenticated Keying using Internet Key Exchange

As part of the IPsec VPN tunnel establishment, Alice and Bob will use the IKE (Internet Key Exchange) protocol to authenticate each other and create necessary session keys to protect the data traffic. *IbexOS* supports IKE version 1 (IKEv1) and IKE version 2 (IKEv2). It supports IKE with authentication through pre-shared keys (PSK) or certificates (RSA signature keys using X.509 certificates). In the IKE handshake, Alice and Bob identify themselves and use their configured PSK or certificates to authenticate each other. When configuring an IPsec tunnel, the identities of the peers should be defined in cfgVpnIpsecLeftId and cfgVpnIpsecRightId.

See https://wiki.strongswan.org/projects/strongswan/wiki/IdentityParsing for a description of types and binary encoding of identity strings.

It is recommended to use IKEv2 (cfgVpnIpsecKeyExchange = ikev2(2)) whenever possible.

Limitations when using IKEv1:

- Using IKEv1 aggressive Mode with pre-shared keys (PSK) is the least secure option. In this particular scenario, it is possible for an attacker to gather all necessary information in order to mount an off-line dictionary (brute force) attack on the PSK. Therefore: IKEv1 agressive mode with PSK authentication is **not** recommended.

- Due to limitations in IKEv1, main mode with PSK can only be used with IP address (or %any)

as identity (cfgVpnIpsecLeftId and cfgVpnIpsecRightId).

Both for the IKE (cfgVpnIpsecIke) and ESP (cfgVpnIpsecEsp) handshakes the user can specify which cryptographic protocols to use.

Perfect Forward Secrecy (PFS) is a must have for every IPsec tunnel. Using PFS will make the endpoints negotiate a new key for each IKE and/or IPsec Security Association (SA) upon rekey or reauthentication event. This protects the confidentiality of the traffic, if the IKE shared secret is leaked. Note, that the keys of the first SA of a new IKEv2 connection are derived from the IKE shared secret. However, subsequent SAs will use new keys if PFS is used. PFS is enabled by appending a DH group to the ESP cipher settings (cfgVpnIpsecEsp).

4.20.4 Dead Peer Detection

The connectivity through an established IPsec tunnel may be broken unexpectedly, for example one of the peers go down or is disconnected, or if some kind of routing, NAT or firewall problem occurs on the path between them. Dead Peer Detection (DPD) can be used to discover and manage such situations. In DPD the peers exchange keep-alive messages to monitor if the remote peer is still reachable. If a peer determines connectivity to be broken, appropriate actions should be taken. Use cfgVpnIpsecDpdAction, cfgVpnIpsecDpdDelay and cfgVpnIpsecDpdTimeout to configure DPD.

4.20.5 Route Based VPNs

Generally IPsec processing is based on policies. After regular route lookups are done, the OS kernel consults its Security Policy Database (SPD) for a matching policy and if one is found that is associated with an IPsec Security Association (SA), the packet is processed (e.g. encrypted and sent as ESP packet).

VirtualTunnelInterface (VTI) devices act like a wrapper around existing IPsec policies. This means you cannot just route arbitrary packets to a VTI device to get them tunneled, the established IPsec policies have to match too. However, you can negotiate 0.0.0.0/0 traffic selectors on both ends to allow tunneling anything that's routed via VTI device.

Enable VTI for Route Based IPsec VPN by setting cfgVpnIpsecGlblVirtualTunnelInterface = enabled(1) and by configuring routes (cfgRouteTableEnabled, cfgRouteTableDestinationNetwork, cfgRouteTable-Source, cfgRouteTableInterface).

4.20.6 IPsec Configuration Example

Fig. 4.10 shows a sample IPsec VPN topology used to illustrate IPsec VPN configurations.

We have two VPN gateways, Alice (VPN GW1) and Bob (VPN GW2), which are used to establish a secure VPN tunnel between the central office network (192.168.10.0/24) and the branch office network (192.168.11.0/24).

4.20.6.0.1 Example of using IPsec VPN with PSK

This section illustrates configuration parameters for configuring IPsec VPNs using IKE authentication with pre-shared key (PSKs).

Configuration of the two gateways then looks like this:

```
# Static IP address for inbound iface
WESTERMO-SW6-MIB::cfgNetIpEnabled.0 = 1
WESTERMO-SW6-MIB::cfgNetIpProto.0 = 0
WESTERMO-SW6-MIB::cfgNetIpAddr.0 = 192.168.10.1/24
WESTERMO-SW6-MIB::cfgNetIpInterface.0 = eth0
# IPsec alias for outbound iface
WESTERMO-SW6-MIB::cfgNetIpEnabled.3 = 1
WESTERMO-SW6-MIB::cfgNetIpProto.3 = 0
WESTERMO-SW6-MIB::cfgNetIpAddr.3 = "10.1.2.3"
WESTERMO-SW6-MIB::cfgNetIpInterface.3 = eth1
# IPsec tunnel configuration
WESTERMO-SW6-MIB::cfgNetIpsecEnabled.0 = 1
WESTERMO-SW6-MIB::cfgVpnIpsecType.0 = 0
WESTERMO-SW6-MIB::cfgVpnIpsecAuto.0 = 1
WESTERMO-SW6-MIB::cfgVpnIpsecLeftAuth.0 = psk
WESTERMO-SW6-MIB::cfgVpnIpsecRightAuth.0 = psk
WESTERMO-SW6-MIB::cfgVpnIpsecPassword.0 = TopSecretPw1!
WESTERMO-SW6-MIB::cfgVpnIpsecLeft.0 = 10.1.2.3
WESTERMO-SW6-MIB::cfgVpnIpsecLeftId.0 = GW1
WESTERMO-SW6-MIB::cfgVpnIpsecLeftSubnet.0 = 192.168.10.0/24
WESTERMO-SW6-MIB::cfgVpnIpsecRight.0 = 10.4.5.6
WESTERMO-SW6-MIB::cfgVpnIpsecRightId.0 = GW2
WESTERMO-SW6-MIB::cfgVpnIpsecRightSubnet.0 = 192.168.11.0/24
WESTERMO-SW6-MIB::cfgVpnIpsecKeyExchange.0 = 2
WESTERMO-SW6-MIB::cfgVpnIpsecIke.0 = aes128-sha1-modp1024!
WESTERMO-SW6-MIB::cfgVpnIpsecEsp.0 = aes128-sha1-modp1024!
```

Listing 4.12: *VPN GW1: IPsec PSK Authentication*

```
# Static IP address for inbound iface
WESTERMO-SW6-MIB::cfgNetIpEnabled.0 = 1
WESTERMO-SW6-MIB::cfgNetIpProto.0 = 0
WESTERMO-SW6-MIB::cfgNetIpAddr.0 = 192.168.11.1/24
WESTERMO-SW6-MIB::cfgNetIpInterface.0 = eth0
# IPsec alias for outbound iface
WESTERMO-SW6-MIB::cfgNetIpEnabled.3 = 1
WESTERMO-SW6-MIB::cfgNetIpInterface.3 = eth1
```

```
WESTERMO-SW6-MIB::cfgNetIpProto.3 = 0
WESTERMO-SW6-MIB::cfgNetIpAddr.3 = 10.4.5.6
# IPsec tunnel configuration
WESTERMO-SW6-MIB::cfgNetIpsecEnabled.0 = 1
WESTERMO-SW6-MIB::cfgVpnIpsecType.0 = 0
WESTERMO-SW6-MIB::cfgVpnIpsecAuto.0 = 1
WESTERMO-SW6-MIB::cfgVpnIpsecLeftAuth.0 = psk
WESTERMO-SW6-MIB::cfgVpnIpsecRightAuth.0 = psk
WESTERMO-SW6-MIB::cfgVpnIpsecPassword.0 = TopSecretPw1!
WESTERMO-SW6-MIB::cfgVpnIpsecLeft.0 = 10.4.5.6
WESTERMO-SW6-MIB::cfgVpnIpsecLeftId.0 = GW2
WESTERMO-SW6-MIB::cfgVpnIpsecLeftSubnet.0 = 192.168.11.0/24
WESTERMO-SW6-MIB::cfgVpnIpsecRight.0 = 10.1.2.3
WESTERMO-SW6-MIB::cfgVpnIpsecRightId.0 = GW1
WESTERMO-SW6-MIB::cfgVpnIpsecRightSubnet.0 = 192.168.10.0/24
WESTERMO-SW6-MIB::cfgVpnIpsecKeyExchange.0 = 2
WESTERMO-SW6-MIB::cfgVpnIpsecIke.0 = aes128-sha1-modp1024!
WESTERMO-SW6-MIB::cfgVpnIpsecEsp.0 = aes128-sha1-modp1024!
```

Listing 4.13: *VPN GW2: IPsec PSK Authentication*

4.20.6.0.2 Example of using IPsec VPN with Public Key

This section illustrates configuration parameters for configuring IPsec VPNs using IKEv2 authentication with Public Key Signature Authentication.

Configuration of the two gateways then looks like this:

Upload following certificates to Gateway 1:

1. Alice's public key
2. Alice's private key
3. Bob's public key

Upload following certificates to Gateway 2:

1. Bob's public key
2. Bob's private key
3. Alice's public key

```
# Upload following certificates:
# Static IP address for inbound iface
```

```
WESTERMO-SW6-MIB::cfgNetIpEnabled.0 = 1
WESTERMO-SW6-MIB::cfgNetIpProto.0 = 0
WESTERMO-SW6-MIB::cfgNetIpAddr.0 = 192.168.10.1/24
WESTERMO-SW6-MIB::cfgNetIpInterface.0 = eth0
# IPsec alias for outbound iface
WESTERMO-SW6-MIB::cfgNetIpEnabled.3 = 1
WESTERMO-SW6-MIB::cfgNetIpProto.3 = 0
WESTERMO-SW6-MIB::cfgNetIpAddr.3 = 10.1.2.3
WESTERMO-SW6-MIB::cfgNetIpInterface.3 = eth1
# IPsec tunnel configuration
WESTERMO-SW6-MIB::cfgNetIpsecEnabled.0 = 1
WESTERMO-SW6-MIB::cfgVpnIpsecType.0 = 0
WESTERMO-SW6-MIB::cfgVpnIpsecAuto.0 = 1
WESTERMO-SW6-MIB::cfgVpnIpsecLeftAuth.0 = pubkey
WESTERMO-SW6-MIB::cfgVpnIpsecRightAuth.0 = pubkey
WESTERMO-SW6-MIB::cfgVpnIpsecLeft.0 = 10.1.2.3
WESTERMO-SW6-MIB::cfgVpnIpsecLeftId.0 = C=CH, O=Westermo, CN=Alice
WESTERMO-SW6-MIB::cfgVpnIpsecLeftSubnet.0 = 192.168.10.0/24
WESTERMO-SW6-MIB::cfgVpnIpsecRight.0 = 10.4.5.6
WESTERMO-SW6-MIB::cfgVpnIpsecRightId.0 = C=CH, O=Westermo, CN=Bob
WESTERMO-SW6-MIB::cfgVpnIpsecRightSubnet.0 = 192.168.11.0/24
WESTERMO-SW6-MIB::cfgVpnIpsecKeyExchange.0 = 2
WESTERMO-SW6-MIB::cfgVpnIpsecIke.0 = aes128-sha1-modp1024!
WESTERMO-SW6-MIB::cfgVpnIpsecEsp.0 = aes128-sha1-modp1024!
```

Listing 4.14: *VPN GW1: IPsec Public Key Authentication without CA*

```
# Static IP address for inbound iface
WESTERMO-SW6-MIB::cfgNetIpEnabled.0 = 1
WESTERMO-SW6-MIB::cfgNetIpProto.0 = 0
WESTERMO-SW6-MIB::cfgNetIpAddr.0 = 192.168.11.1/24
WESTERMO-SW6-MIB::cfgNetIpInterface.0 = eth0
# IPsec alias for outbound iface
WESTERMO-SW6-MIB::cfgNetIpEnabled.3 = 1
WESTERMO-SW6-MIB::cfgNetIpInterface.3 = eth1
WESTERMO-SW6-MIB::cfgNetIpProto.3 = 0
WESTERMO-SW6-MIB::cfgNetIpAddr.3 = 10.4.5.6
# IPsec tunnel configuration
WESTERMO-SW6-MIB::cfgNetIpsecEnabled.0 = 1
WESTERMO-SW6-MIB::cfgVpnIpsecType.0 = 0
WESTERMO-SW6-MIB::cfgVpnIpsecAuto.0 = 1
WESTERMO-SW6-MIB::cfgVpnIpsecLeftAuth.0 = pubkey
WESTERMO-SW6-MIB::cfgVpnIpsecRightAuth.0 = pubkey
WESTERMO-SW6-MIB::cfgVpnIpsecLeft.0 = 10.4.5.6
WESTERMO-SW6-MIB::cfgVpnIpsecLeftId.0 = C=CH, O=Westermo, CN=Bob
WESTERMO-SW6-MIB::cfgVpnIpsecLeftSubnet.0 = 192.168.11.0/24
WESTERMO-SW6-MIB::cfgVpnIpsecRight.0 = 10.1.2.3
WESTERMO-SW6-MIB::cfgVpnIpsecRightId.0 = C=CH, O=Westermo, CN=Alice
WESTERMO-SW6-MIB::cfgVpnIpsecRightSubnet.0 = 192.168.10.0/24
WESTERMO-SW6-MIB::cfgVpnIpsecKeyExchange.0 = 2
WESTERMO-SW6-MIB::cfgVpnIpsecIke.0 = aes128-sha1-modp1024!
```

```
WESTERMO-SW6-MIB::cfgVpnIpsecEsp.0 = aes128-sha1-modp1024!
```

Listing 4.15: *VPN GW2: IPsec Public Key Authentication without CA*

4.20.6.0.3 Example of using IPsec VPN with CA Certificate

This section illustrates configuration parameters for configuring IPsec VPNs using IKEv2 authentication with CA Authentication.

Configuration of the two gateways then looks like this:

Upload following certificates to Gateway 1:

1. CA Certificate
2. Alice's Certificate (signed by CA)
3. Alice's private key

Upload following certificates to Gateway 2:

1. CA Certificate
2. Bob's Certificate (signed by CA)
3. Bob's private key

```
# Upload following certificates:
# Static IP address for inbound iface
WESTERMO-SW6-MIB::cfgNetIpEnabled.0 = 1
WESTERMO-SW6-MIB::cfgNetIpProto.0 = 0
WESTERMO-SW6-MIB::cfgNetIpAddr.0 = 192.168.10.1/24
WESTERMO-SW6-MIB::cfgNetIpInterface.0 = eth0
# IPsec alias for outbound iface
WESTERMO-SW6-MIB::cfgNetIpEnabled.3 = 1
WESTERMO-SW6-MIB::cfgNetIpProto.3 = 0
WESTERMO-SW6-MIB::cfgNetIpAddr.3 = 10.1.2.3
WESTERMO-SW6-MIB::cfgNetIpInterface.3 = eth1
# IPsec tunnel configuration
WESTERMO-SW6-MIB::cfgNetIpsecEnabled.0 = 1
WESTERMO-SW6-MIB::cfgVpnIpsecType.0 = 0
WESTERMO-SW6-MIB::cfgVpnIpsecAuto.0 = 1
WESTERMO-SW6-MIB::cfgVpnIpsecLeftAuth.0 = pubkey
WESTERMO-SW6-MIB::cfgVpnIpsecRightAuth.0 = pubkey
WESTERMO-SW6-MIB::cfgVpnIpsecLeft.0 = 10.1.2.3
WESTERMO-SW6-MIB::cfgVpnIpsecLeftId.0 = C=CH, O=Westermo, CN=Alice
```

```
WESTERMO-SW6-MIB::cfgVpnIpsecLeftSubnet.0 = 192.168.10.0/24
WESTERMO-SW6-MIB::cfgVpnIpsecRight.0 = 10.4.5.6
WESTERMO-SW6-MIB::cfgVpnIpsecRightId.0 = C=CH, O=Westermo, CN=Bob
WESTERMO-SW6-MIB::cfgVpnIpsecRightSubnet.0 = 192.168.11.0/24
WESTERMO-SW6-MIB::cfgVpnIpsecKeyExchange.0 = 2
WESTERMO-SW6-MIB::cfgVpnIpsecIke.0 = aes128-sha1-modp1024!
WESTERMO-SW6-MIB::cfgVpnIpsecEsp.0 = aes128-sha1-modp1024!
```

Listing 4.16: *VPN GW1: IPsec Public Key Authentication with CA*

```
# Static IP address for inbound iface
WESTERMO-SW6-MIB::cfgNetIpEnabled.0 = 1
WESTERMO-SW6-MIB::cfgNetIpProto.0 = 0
WESTERMO-SW6-MIB::cfgNetIpAddr.0 = 192.168.11.1/24
WESTERMO-SW6-MIB::cfgNetIpInterface.0 = eth0
# IPsec alias for outbound iface
WESTERMO-SW6-MIB::cfgNetIpEnabled.3 = 1
WESTERMO-SW6-MIB::cfgNetIpInterface.3 = eth1
WESTERMO-SW6-MIB::cfgNetIpProto.3 = 0
WESTERMO-SW6-MIB::cfgNetIpAddr.3 = 10.4.5.6
# IPsec tunnel configuration
WESTERMO-SW6-MIB::cfgNetIpsecEnabled.0 = 1
WESTERMO-SW6-MIB::cfgVpnIpsecType.0 = 0
WESTERMO-SW6-MIB::cfgVpnIpsecAuto.0 = 1
WESTERMO-SW6-MIB::cfgVpnIpsecLeftAuth.0 = pubkey
WESTERMO-SW6-MIB::cfgVpnIpsecRightAuth.0 = pubkey
WESTERMO-SW6-MIB::cfgVpnIpsecLeft.0 = 10.4.5.6
WESTERMO-SW6-MIB::cfgVpnIpsecLeftId.0 = C=CH, O=Westermo, CN=Bob
WESTERMO-SW6-MIB::cfgVpnIpsecLeftSubnet.0 = 192.168.11.0/24
WESTERMO-SW6-MIB::cfgVpnIpsecRight.0 = 10.1.2.3
WESTERMO-SW6-MIB::cfgVpnIpsecRightId.0 = C=CH, O=Westermo, CN=Alice
WESTERMO-SW6-MIB::cfgVpnIpsecRightSubnet.0 = 192.168.10.0/24
WESTERMO-SW6-MIB::cfgVpnIpsecKeyExchange.0 = 2
WESTERMO-SW6-MIB::cfgVpnIpsecIke.0 = aes128-sha1-modp1024!
WESTERMO-SW6-MIB::cfgVpnIpsecEsp.0 = aes128-sha1-modp1024!
```

Listing 4.17: *VPN GW2: IPsec Public Key Authentication with CA*

4.20.6.0.4 Example of using IPsec VPN with PSK as route-based VPN

This section illustrates configuration parameters for configuring route-based IPsec VPNs using PSK authentication.

Configuration of the two gateways then looks like this:

```
# Static IP address for inbound iface
WESTERMO-SW6-MIB::cfgNetIpEnabled.0 = 1
WESTERMO-SW6-MIB::cfgNetIpProto.0 = 0
```

```
WESTERMO-SW6-MIB::cfgNetIpAddr.0 = 192.168.10.1/24
WESTERMO-SW6-MIB::cfgNetIpInterface.0 = eth0
# IPsec alias for outbound iface
WESTERMO-SW6-MIB::cfgNetIpEnabled.3 = 1
WESTERMO-SW6-MIB::cfgNetIpProto.3 = 0
WESTERMO-SW6-MIB::cfgNetIpAddr.3 = 10.1.2.3/24
WESTERMO-SW6-MIB::cfgNetIpInterface.3 = eth1
# IPsec VTI
WESTERMO-SW6-MIB::cfgVpnIpsecGlblVirtualTunnelInterface.0 = 1
WESTERMO-SW6-MIB::cfgNetIpsecEnabled.0 = 1
WESTERMO-SW6-MIB::cfgVpnIpsecType.0 = 0
WESTERMO-SW6-MIB::cfgVpnIpsecAuto.0 = 1
WESTERMO-SW6-MIB::cfgVpnIpsecLeftAuth.0 = psk
WESTERMO-SW6-MIB::cfgVpnIpsecRightAuth.0 = psk
WESTERMO-SW6-MIB::cfgVpnIpsecPassword.0 = TopSecretPw1!
WESTERMO-SW6-MIB::cfgVpnIpsecLeft.0 = 10.1.2.3
WESTERMO-SW6-MIB::cfgVpnIpsecLeftId.0 = GW1
WESTERMO-SW6-MIB::cfgVpnIpsecLeftSubnet.0 = 192.168.10.0/24
WESTERMO-SW6-MIB::cfgVpnIpsecRight.0 = 10.4.5.6
WESTERMO-SW6-MIB::cfgVpnIpsecRightId.0 = GW2
WESTERMO-SW6-MIB::cfgVpnIpsecRightSubnet.0 = 192.168.11.0/24
WESTERMO-SW6-MIB::cfgVpnIpsecKeyExchange.0 = 2
WESTERMO-SW6-MIB::cfgVpnIpsecIke.0 = aes128-sha256-modp3072!
WESTERMO-SW6-MIB::cfgVpnIpsecEsp.0 = aes128-sha256!
# Add route to 192.168.11.0/24 over ipsec0 with src 10.1.2.3
WESTERMO-SW6-MIB::cfgRouteTableEnabled.0 = 1
WESTERMO-SW6-MIB::cfgRouteTableDestinationNetwork.0 = 192.168.11.0/24
WESTERMO-SW6-MIB::cfgRouteTableSource.0 = 10.1.2.3
WESTERMO-SW6-MIB::cfgRouteTableInterface.0 = ipsec0
```

Listing 4.18: *VPN GW1: Route-based IPsec VPN with PSK Authentication*

```
# Static IP address for inbound iface
WESTERMO-SW6-MIB::cfgNetIpEnabled.0 = 1
WESTERMO-SW6-MIB::cfgNetIpProto.0 = 0
WESTERMO-SW6-MIB::cfgNetIpAddr.0 = 192.168.11.1/24
WESTERMO-SW6-MIB::cfgNetIpInterface.0 = eth0
# IPsec alias for outbound iface
WESTERMO-SW6-MIB::cfgNetIpEnabled.3 = 1
WESTERMO-SW6-MIB::cfgNetIpProto.3 = 0
WESTERMO-SW6-MIB::cfgNetIpAddr.3 = 10.4.5.6/24
WESTERMO-SW6-MIB::cfgNetIpInterface.3 = eth1
# IPsec VTI
WESTERMO-SW6-MIB::cfgVpnIpsecGlblVirtualTunnelInterface.0 = 1
WESTERMO-SW6-MIB::cfgNetIpsecEnabled.0 = 1
WESTERMO-SW6-MIB::cfgVpnIpsecType.0 = 0
WESTERMO-SW6-MIB::cfgVpnIpsecAuto.0 = 1
WESTERMO-SW6-MIB::cfgVpnIpsecLeftAuth.0 = psk
WESTERMO-SW6-MIB::cfgVpnIpsecRightAuth.0 = psk
WESTERMO-SW6-MIB::cfgVpnIpsecPassword.0 = TopSecretPw1!
WESTERMO-SW6-MIB::cfgVpnIpsecLeft.0 = 10.4.5.6
```

```
WESTERMO-SW6-MIB::cfgVpnIpsecLeftId.0 = GW2
WESTERMO-SW6-MIB::cfgVpnIpsecLeftSubnet.0 = 192.168.11.0/24
WESTERMO-SW6-MIB::cfgVpnIpsecRight.0 = 10.1.2.3
WESTERMO-SW6-MIB::cfgVpnIpsecRightId.0 = GW1
WESTERMO-SW6-MIB::cfgVpnIpsecRightSubnet.0 = 192.168.10.0/24
WESTERMO-SW6-MIB::cfgVpnIpsecKeyExchange.0 = 2
WESTERMO-SW6-MIB::cfgVpnIpsecIke.0 = aes128-sha256-modp3072!
WESTERMO-SW6-MIB::cfgVpnIpsecEsp.0 = aes128-sha256!
# add route to 192.168.10.0/24 over ipsec0 with src 10.4.5.6
WESTERMO-SW6-MIB::cfgRouteTableEnabled.0 = 1
WESTERMO-SW6-MIB::cfgRouteTableDestinationNetwork.0 = 192.168.10.0/24
WESTERMO-SW6-MIB::cfgRouteTableSource.0 = 10.4.5.6
WESTERMO-SW6-MIB::cfgRouteTableInterface.0 = ipsec0
```

Listing 4.19: *VPN GW2: Route-based IPsec VPN with PSK Authentication*

4.21 OpenVPN Configuration

OpenVPN may operate as a client or server. *IbexOS* 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.

4.21.1 Interface Configuration

Each OpenVPN interface in the table cfgNetOpenvpnTable is managed by a separate instance of OpenVPN. These instances are created and enabled via the table cfgVpnOpenvpnTable. The indices of both tables refer to the same instance.

cfgNetOpenvpnTable is used to configure generic network related parameters like the MTU of the interface, bridge assignment, or vlan.

cfgVpnOpenvpnTable is used to configure OpenVPN specific parameters, like the used encryption or address of the server.

4.21.2 OpenVPN Configuration Example

OpenVPN, WireGuard, IPsec and GRE have been introduced to supplement cellular networks in *IbexOS*.

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 4.11. An encrypted tunnel is created to travers public networks such as the Internet or private transfer networks of cellular providers.

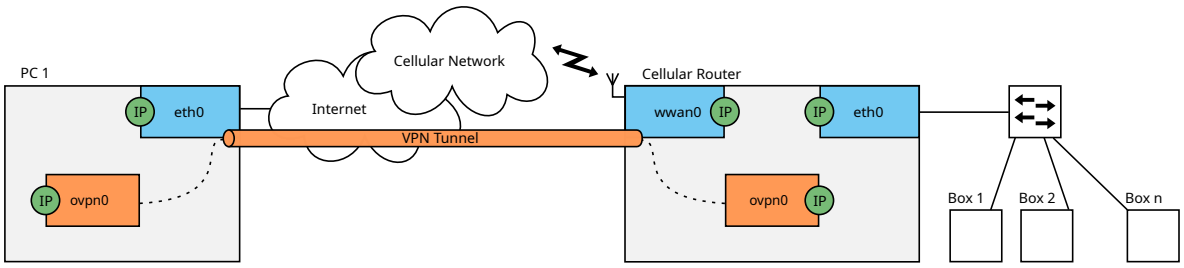


Figure 4.11: *OpenVPN in a Cellular Network*

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 4.18 explains how to configure cellular networks.

A working OpenVPN instance requires:

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

4.21.2.1 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.

4.21.2.2 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 4.11 are in the Listing 4.20:

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

Listing 4.20: *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 4.11 is of type tun. The interface may not be bridged, thus cfgNetOpenvpnBridge is set to `-1`. The remainig entries may be left with their default values.

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

4.21.2.3 Configuring the OpenVPN Instance

The OpenVPN instance, similar to its corresponding interface, is defined in the SNMP table cfgVpnOpenvpnTable.

The required entries for the example in Figure 4.11 are in the Listing 4.21:

```
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::cfgVpnOpenvpnAuth.0 = SHA256
WESTERMO-SW6-MIB::cfgVpnOpenvpnCipher.0 = AES-256-CBC
WESTERMO-SW6-MIB::cfgVpnOpenvpnCompress.0 = 2
```

Listing 4.21: *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 cfgSysNameserverTable. 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)`.

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.

4.22 Wireguard Configuration

WireGuard does not operate in a Client/Server fashion in the traditional sense. Each participant is a peer with a private key and a public key. One or multiple remote peers may be defined for each local peer. A peer accepts traffic from a remote peer when the inbound traffic matches the configured private key.

WireGuard is a pure L3 tunnel, thus it may not be bridged. If a L2 transparent tunnel is required, GRE in TAP mode may be used on top of WireGuard. WireGuard uses UDP as transport protocol.

WireGuard provides excellent roaming capabilities, since it does not need to re-establish its connection-state with its peers when moving from one network to another.

4.22.1 Interface Configuration

Each WireGuard interface in the table cfgNetWireguardTable is managed by a separate instance of WireGuard. These instances are created and enabled via the table cfgVpnWireguardTable. The indices of both tables refer to the same instance.

cfgNetWireguardTable is used to configure generic network related parameters like the MTU of the interface.

cfgVpnWireguardTable is used to configure WireGuard specific parameters, like the private key or address of remote peers.

4.22.2 Wireguard Configuration Example

OpenVPN, WireGuard, IPsec and GRE have been introduced to supplement cellular networks in *IbexOS*.

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

The goal is to provide private access from the Cellular Router to PC 1, see Figure 4.12. An encrypted tunnel is created to travers public networks such as the Internet or private transfer networks of cellular providers.

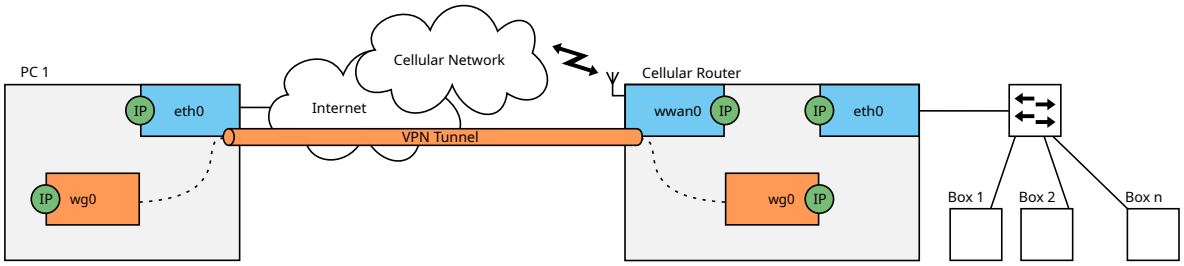


Figure 4.12: *WireGuard in a Cellular Network*

In this example, PC 1 acts as WireGuard peer that only receives incoming connections (responder) and the Cellular Router that actively establishes the connection (initiator). Multiple initiators may connect to the responder.

The focus of this example is WireGuard. A working cellular network is assumed. Section 4.18 explains how to configure cellular networks.

A working WireGuard instance requires:

- A configured and enabled WireGuard interface
- A configured WireGuard instance

4.22.2.1 Configuring the WireGuard Interface

The configuration of an WireGuard interface is defined by the entries of the table cfgNetWireguardTable.

The required entries for the example in Figure 4.12 are in the config example below.

```
WESTERMO-SW6-MIB::cfgNetWgEnabled.0 = 1
WESTERMO-SW6-MIB::cfgNetIpEnabled.3 = 1
WESTERMO-SW6-MIB::cfgNetIpAddr.3 = 10.0.99.2/24
WESTERMO-SW6-MIB::cfgNetIpInterface.3 = wg0
```

Listing 4.22: *WireGuard Interface Configuration*

The WireGuard interface and instance are created by setting cfgNetWgEnabled to `enabled(1)`. The remainig entries may be left with their default values.

Additionally the WireGuard interface requires an IP address.

4.22.2.2 Configuring the WireGuard Instance

The WireGuard instance, similar to its corresponding interface, is defined in the table cfgVpnWireguardTable.

The required entries for the example in Figure 4.12 are in the config example below.

```
WESTERMO-SW6-MIB::cfgVpnWgPrivateKey.0 ACwLw8p8UGdv6baTN1dMxhxVqxm779vl2IjDpSFecW8=

WESTERMO-SW6-MIB::cfgVpnWgPEnabled.0 1
WESTERMO-SW6-MIB::cfgVpnWgPPeer.0 dVbkEFbV4mvXkA/lbRxiJH6/cUPg8h+vFRluZN5PGRI=
WESTERMO-SW6-MIB::cfgVpnWgPEndpoint.0 1.2.3.4:51820
WESTERMO-SW6-MIB::cfgVpnWgPAllowedIps.0 10.0.99.0/24
WESTERMO-SW6-MIB::cfgVpnWgPPersistentKeepalive.0 25
```

Listing 4.23: *WireGuard Instance Configuration*

cfgVpnWgPrivateKey is the key with which incoming traffic is decrypted. This key should never be given to anyone. Based on the private key a public key is automatically generated and can be read from cfgVpnWgPublicKey. The public key has to be installed on the remote peer.

The remote peers public key has to be installed in cfgVpnWireguardPeersTable at cfgVpnWgPPeer. The Cellular Router acts as initiator, which is configured by setting cfgVpnWgPEndpoint to an address and port. Port number 51820 is the preferred port for a WireGuard responder.

This example is intended to travers the internet and a cellular network, it is probable that NAT is involved somewhere on the path from initiator to responder. When traversing NAT it is recommended to set cfgVpnWgPPersistentKeepalive to some low value to ensure traffic can flow in both directions. A sane value is 25s.

Finally cfgVpnWgPAllowedIps is used to specify which subnets are allowed to be received through the tunnel, and which destinations to send into the tunnel.

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

4.22.2.3 WireGuard Peer FQDN Re-Resolving

The endpoint cfgVpnWgPEndpoint can be specified as an FQDN in the form "host.example.com:51820". The FQDN is resolved to an IP at config time. Wireguard itself operates with an IP and Port only. The

FQDN may not be resolvable at config time, e.g. because the DNS server is not reachable. Also if the FQDN points to a dynamic DNS entry (RFC 2136) the IP to which the FQDN initially has been resolved to could not be up to date.

To re-resolve the FQDN at runtime and update the peer, the Network Link Monitor (NLM) has to be configured to monitor the rechability of the private IP of a peer.

The config example below describes the setup shown in Figure 4.13.

```
# Wireguard peer at index 0 with endpoint host.example.com:51820
WESTERMO-SW6-MIB::cfgVpnWgPEnabled.0 1
WESTERMO-SW6-MIB::cfgVpnWgPPeer.0 dVbkEFbV4mvXkA/lbRxiJH6/cUPg8h+vFRluZN5PGRI=
WESTERMO-SW6-MIB::cfgVpnWgPEndpoint.0 host.example.com:51820
WESTERMO-SW6-MIB::cfgVpnWgPAllowedIps.0 10.0.99.0/24
WESTERMO-SW6-MIB::cfgVpnWgPPersistentKeepalive.0 25
# Configure the IP 10.0.99.2/2 on the local wireguard interface wg0
WESTERMO-SW6-MIB::cfgNetIpEnabled.3 1
WESTERMO-SW6-MIB::cfgNetIpAddr.3 10.0.99.2/24
WESTERMO-SW6-MIB::cfgNetIpInterface.3 wg0
# Monitor the remote Wireguard IP 10.0.99.1 and re-resolve peer index 0 when down
WESTERMO-SW6-MIB::cfgNlmGlblEnabled.0 = 1
WESTERMO-SW6-MIB::cfgNlmMonEnabled.0 = 1
WESTERMO-SW6-MIB::cfgNlmMonType.0 = 1
WESTERMO-SW6-MIB::cfgNlmMonInterval.0 = 500
WESTERMO-SW6-MIB::cfgNlmMonDestination.0 = 10.0.99.1
WESTERMO-SW6-MIB::cfgNlmMonDownAction.0 = 4000
```

Listing 4.24: *WireGuard Peer FQDN Re-Resolving*

The cfgNlmMonType value of icmp(1) configures the NLM monitor to send ICMP (ping) to the IP in cfgNlmMonDestination. The ping is sent every interval specified in milliseconds in cfgNlmMonInterval. The destination is set to the IP that the remote peer has on its Wireguard interface. If this IP can no longer be reached, the peer is considered down.

An action value of 4xxx in cfgNlmMonDownAction means: Offset 4000 for FQDN peer re-resolution, with 'xxx' being the value of the reference to the index cfgVpnWgPIndex of the Wireguard peer to re-resolve. In this example the Wireguard peer at index 0 with the endpoint set to 'host.example.com:51820'. The FQDN is re-resolved every cfgNlmMonInterval until the monitor is back up.

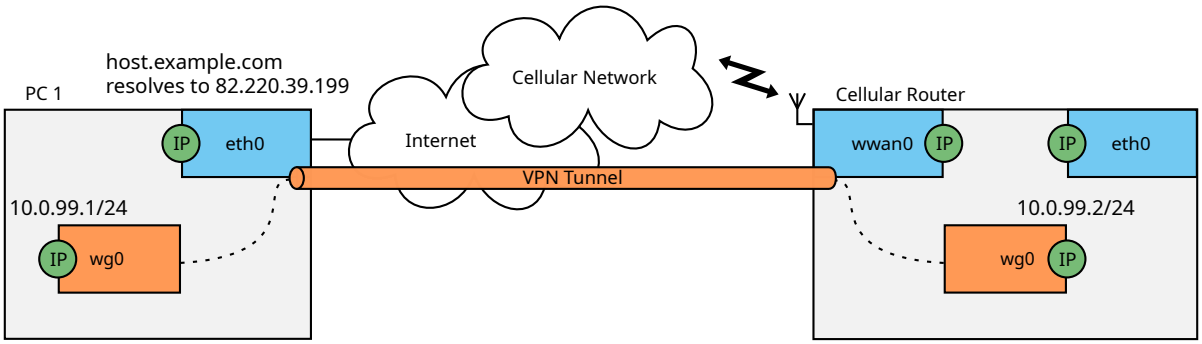


Figure 4.13: *WireGuard FQDN Re-Resolving*

4.22.2.4 WireGuard Policy Routing

When multiple tunnels are configured and different WANs are available, it may be desirable to force a specific tunnel to a specific WAN. The parameter cfgVpnWgMark assigns a mark to frames sent by a specific Wireguard instance. This mark can be matched in routing rules on the cfgRouteRuleTable (see Policy Routing).

4.23 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 (cfgWlanIfaceSsid) - example: "wpwnap"
- Wireless password (cfgWlanIfacePassword)
- 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+"

4.23.1 Configuration File Example: Access point 2.4GHz

```
WESTERMO-SW6-MIB::cfgSysHostname.0 = AP2G
WESTERMO-SW6-MIB::cfgWlanIfaceSsid.0 = wpwnap
```

Listing 4.25: *Access point 2.4GHz*

4.23.2 Configuration File Example: Access point 5GHz

```
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::cfgWlanIfaceScanList.0 = 1
```

Listing 4.26: *Access point 5GHz HT40+*

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

4.24 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. *IbexOS* supports fine grade optimization for bitrates.

4.24.1 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.

4.24.2 QMRR

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

QMMR logging into Syslog can be enabled by cfgWlanGlblLinkmonitorQmrrlogging and cfgWlanGlblLinkmonitorInterval

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)

4.25 802.11s 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 startes 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.

4.26 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).

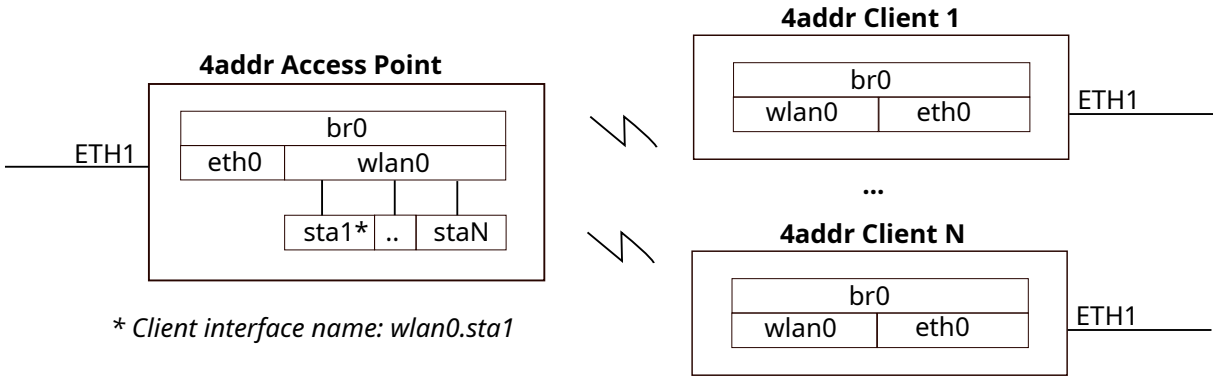


Figure 4.14: *4addr bridge mode setup*

```
WESTERMO-SW6-MIB::cfgWlanIface4addr.0 = 1
```

Listing 4.27: *Access point*

```
WESTERMO-SW6-MIB::cfgNetIpAddr.0 = 192.168.1.30/24
WESTERMO-SW6-MIB::cfgWlanIfaceMode.0 = 1
WESTERMO-SW6-MIB::cfgWlanIface4addr.0 = 1
```

Listing 4.28: *STA/client*

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

4.27 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.

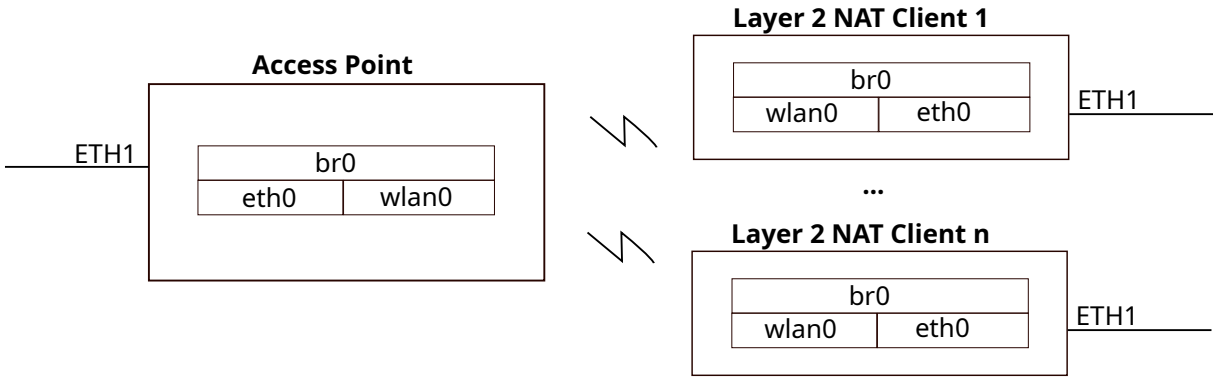


Figure 4.15: *Layer 2 NAT bridge mode setup*

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

```
WESTERMO-SW6-MIB::cfgNetIpAddr.0 = 192.168.1.30/24
WESTERMO-SW6-MIB::cfgWlanIfaceMode.0 = 1
```

```
WESTERMO-SW6-MIB::cfgWlanIfaceL2nat.0 = 1
```

Listing 4.29: *STA/client*

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.

4.28 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.

```
WESTERMO-SW6-MIB::cfgWlanIfaceBssid.0 = 00:11:22:33:44:55
```

Listing 4.30: *Configure new MAC for the wireless radio*

4.29 Wireless Network Optimization

4.29.1 Multicast to Unicast Conversion

At first glance, multicast appears to be a good way of sending the same data to multiple recipients efficiently and reliably. This may be true for a wired network, however, for wireless networks this is often not the case.

On wireless networks, the standard specifies that multicast packets must be sent at low, basic data rates, regardless of the data rate supported by each client. In contrast, unicast packets are sent at the best link rate supported by the client and the environment. The limitation on the data rate of multicast packets often results in the counterintuitive fact that multiple unicast packets are using less airtime than one slowed-down multicast packet.

An additional advantage of sending unicast packets to each client individually is that each packet is acknowledged by the receiver and retransmitted in the event of an error. Multicast packets are not acknowledged and are therefore less reliable.

In order to make use of the multicast to unicast conversion set cfgWlanIfaceMulticastToUnicast to `enabled(1)`.

4.30 Wireless Security

4.30.1 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 (Simultanious 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:

- cfgWlanIfaceEncryption 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

- cfgWlanIfacePassword to configure the WPA2 or WPA3 Personal passphrase (pre-shared key)

To further increase security, the Mangement 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.

4.30.1.1 Overview of the encryption modes

Encryption mode	cfgWlanIfaceEncryption	cfgWlan802dot1xCiphers
OPEN	open(0)	
WPA2 Personal (PSK)	psk(3)	
WPA3 Personal (SAE)	sae(7)	
WPA2 Enterprise	eap(6)	DEFAULT:!EXP:!LOW
WPA3 Enterprise	eap(10)	ECDHE-RSA-AES256-GCM-SHA384 ECDHE-ECDSA-AES256-GCM-SHA384 ECDHE-RSA-AES256-SHA384
Wi-Fi OWE	owe(8)	
WPA3 Transition Mode	saepsk(9)	

Table 4.5: *Overview of the encryption modes*

4.30.2 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)

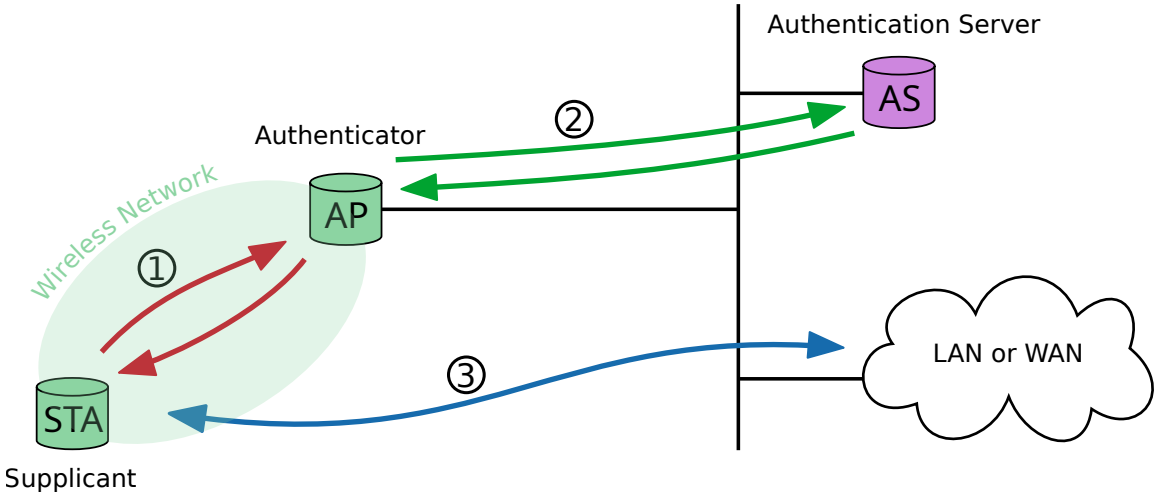


Figure 4.16: *Supplicant authentication using IEEE 802.1X*

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 server[1]
- 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:

```
# 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
```

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

```
WESTERMO-SW6-MIB::cfgWlanDevModulation.0 = 12
WESTERMO-SW6-MIB::cfgWlanDevFrequency.0 = 5180
WESTERMO-SW6-MIB::cfgWlanIfaceSsid.0 = radiustesting
WESTERMO-SW6-MIB::cfgWlanIfaceEncryption.0 = 6
WESTERMO-SW6-MIB::cfgWlanGlblCountry.0 = EU
WESTERMO-SW6-MIB::cfgNtpClientEnabled.0 = 1
WESTERMO-SW6-MIB::cfgNtpClientHost.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
```

Listing 4.31: *Authenticator*

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.

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

Listing 4.32: *Authenticator with second authentication server address*

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

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.

```
# 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::cfgWlanIfaceMode.0 = 1
WESTERMO-SW6-MIB::cfgWlanIfaceSsid.0 = radiustesting
WESTERMO-SW6-MIB::cfgWlanIfaceEncryption.0 = 6
WESTERMO-SW6-MIB::cfgWlanIfaceScanList.0 = 2
WESTERMO-SW6-MIB::cfgWlanGlblCountry.0 = EU
WESTERMO-SW6-MIB::cfgNtpClientEnabled.0 = 1
WESTERMO-SW6-MIB::cfgNtpClientHost.0 = 192.168.2.2
```

Listing 4.33: *Supplicant*

If no certificates are uploaded before applying WPA2-EAP as encryption mode (cfgWlanIfaceEncryption), 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*).

4.30.3 RadSec proxy

The 802.1X communication protocol to the RADIUS server is not encrypted. This may be acceptable in local networks, but it is critical when authenticating via public infrastructures. Instead of securing the connection with a VPN tunnel, the RadSec protocol ensures a certificate based secure connection to the RADIUS server.

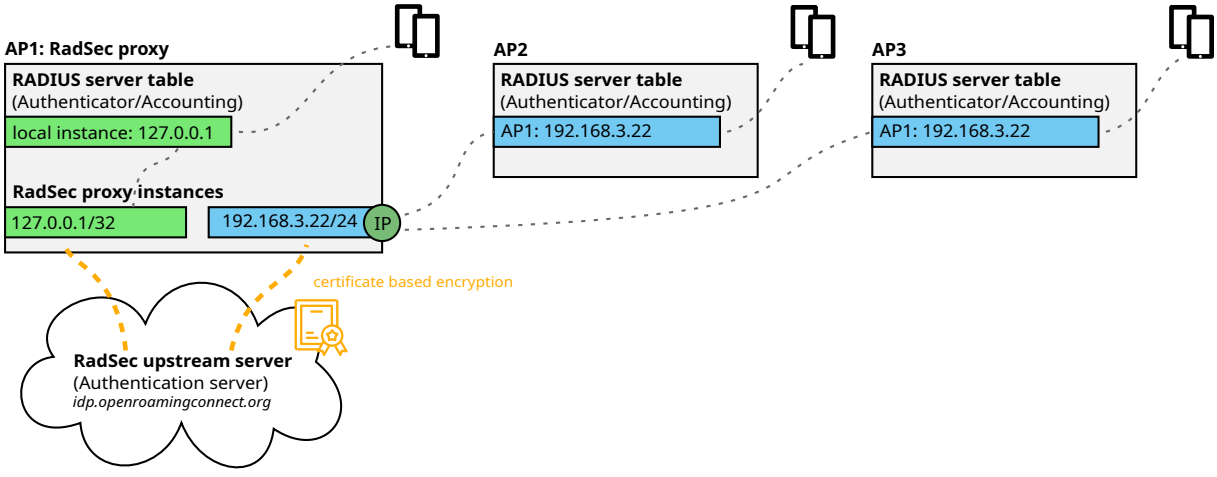


Figure 4.17: *RadSec proxy for locally and remotely connected clients*

In this example, AP1 is an access point with locally connected clients that runs a RadSec proxy. For the own connected clients, the AP must provide a local authenticator instance (the same for accounting). Supplicant authentication requests are forwarded to the RadSec proxy instance bound on localhost (127.0.0.1/32), encrypted and sent to the remote RadSec upstream server. The authentication server may accept the client request and authorizes the internet connection. For supplicants connected to other access points in the same local network, a second RadSec proxy instance is configured (192.168.3.22/24) to also forward authentication requests to the upstream RADIUS server.

```
# local network configuration
WESTERMO-SW6-MIB::cfgNetIpAddr.0 = 192.168.3.22/24
# RADIUS server table for authentication and accounting
WESTERMO-SW6-MIB::cfgWlan802dot1xAuthSrvEnabled.0 = 1
WESTERMO-SW6-MIB::cfgWlan802dot1xAuthSrvId.0 = 0
WESTERMO-SW6-MIB::cfgWlan802dot1xAuthSrvIpAddr.0 = 127.0.0.1
WESTERMO-SW6-MIB::cfgWlan802dot1xAuthSrvPort.0 = 1812
WESTERMO-SW6-MIB::cfgWlan802dot1xAuthSrvSharedSecret.0 = wireless4more
WESTERMO-SW6-MIB::cfgWlan802dot1xAcctSrvEnabled.0 = 1
WESTERMO-SW6-MIB::cfgWlan802dot1xAcctSrvId.0 = 0
WESTERMO-SW6-MIB::cfgWlan802dot1xAcctSrvIpAddr.0 = 127.0.0.1
WESTERMO-SW6-MIB::cfgWlan802dot1xAcctSrvPort.0 = 1813
WESTERMO-SW6-MIB::cfgWlan802dot1xAcctSrvSharedSecret.0 = wireless4more
# RadSec proxy instance for local connected clients (same AP)
WESTERMO-SW6-MIB::cfgRadsecProxyInstEnabled.0 = 1
WESTERMO-SW6-MIB::cfgRadsecProxyInstInstanceId.0 = 0
WESTERMO-SW6-MIB::cfgRadsecProxyInstLocalAddress.0 = 127.0.0.1/32
WESTERMO-SW6-MIB::cfgRadsecProxyInstLocalAuthSrvPort.0 = 1812
WESTERMO-SW6-MIB::cfgRadsecProxyInstLocalAcctSrvPort.0 = 1813
WESTERMO-SW6-MIB::cfgRadsecProxyInstLocalSecret.0 = wireless4more
# RadSec proxy instance for remote connected clients (other AP's in same network)
WESTERMO-SW6-MIB::cfgRadsecProxyInstEnabled.1 = 1
WESTERMO-SW6-MIB::cfgRadsecProxyInstInstanceId.1 = 0
WESTERMO-SW6-MIB::cfgRadsecProxyInstLocalAddress.1 = 192.168.3.22/24
WESTERMO-SW6-MIB::cfgRadsecProxyInstLocalAuthSrvPort.1 = 1812
WESTERMO-SW6-MIB::cfgRadsecProxyInstLocalAcctSrvPort.1 = 1813
WESTERMO-SW6-MIB::cfgRadsecProxyInstLocalSecret.1 = wireless4more
# RadSec upstream server
WESTERMO-SW6-MIB::cfgRadsecProxyUpSrvEnabled.0 = 1
WESTERMO-SW6-MIB::cfgRadsecProxyUpSrvInstanceId.0 = 0
WESTERMO-SW6-MIB::cfgRadsecProxyUpSrvAuthAddress.0 = idp.openroamingconnect.org
WESTERMO-SW6-MIB::cfgRadsecProxyUpSrvAcctAddress.0 = idp.openroamingconnect.org
WESTERMO-SW6-MIB::cfgRadsecProxyUpSrvPort.0 = 2083
WESTERMO-SW6-MIB::cfgRadsecProxyUpSrvSecret.0 = radsec
WESTERMO-SW6-MIB::cfgRadsecProxyUpSrvCaCertId.0 = 1
WESTERMO-SW6-MIB::cfgRadsecProxyUpSrvClientCertId.0 = 2
WESTERMO-SW6-MIB::cfgRadsecProxyUpSrvClientKeyId.0 = 1
WESTERMO-SW6-MIB::cfgRadsecProxyUpSrvRealms.0 = *
```

Listing 4.34: *RadSec proxy*

If multiple upstream RADIUS servers needs to be defined, different authenticator/accounting server

tables and RadSec proxy instances can be assigned by using different instance ID's.

Make sure that all required IDP certificates (CA, client cert, client key) are installed and referred in the configuration.

All other AP's in the same local network point to the RADIUS authenticator and accounting table of the RadSec proxy AP. Ports and shared secrets of the different instances must match.

```
WESTERMO-SW6-MIB::cfgWlan802dot1xAuthSrvEnabled.0 = 1
WESTERMO-SW6-MIB::cfgWlan802dot1xAuthSrvIpAddr.0 = 192.168.3.22
WESTERMO-SW6-MIB::cfgWlan802dot1xAuthSrvPort.0 = 1812
WESTERMO-SW6-MIB::cfgWlan802dot1xAuthSrvSharedSecret.0 = wireless4more
WESTERMO-SW6-MIB::cfgWlan802dot1xAcctSrvEnabled.0 = 1
WESTERMO-SW6-MIB::cfgWlan802dot1xAcctSrvIpAddr.0 = 192.168.3.22
WESTERMO-SW6-MIB::cfgWlan802dot1xAcctSrvPort.0 = 1813
WESTERMO-SW6-MIB::cfgWlan802dot1xAcctSrvSharedSecret.0 = wireless4more
```

Listing 4.35: *RADIUS server table on other AP's in same network*

4.30.4 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:

- cfgWlanIfaceIeee80211w to enable the Management Frame Protection (MFP) mechanism.
- cfgWlanIfaceIeee80211wMaxTimeout to configure the SA query max timeout
- cfgWlanIfaceIeee80211wRetryTimeout to configure the SA query retry timeout

4.30.5 Time Synchronization over WLAN

A device (STA or AP) can have a trusted or untrusted system time:

- trusted system time if set through ntp (cfgNtpEnabled) or SNMP (setSysTime)
- else untrusted time (e.g. after reboot)

If STA only has untrusted system time, but system time is needed for 802.1X certificate validation, then:

- if cfgWlanIfaceTimeAdvertisement is enabled on AP, then AP announces it's system time (AP_Time) when STA is connecting.

- STA updates it's local time (STA Time) with the given AP Time. An AP Time which is older than the time stored in the non-volatile memory of the STA (SAVED_TIME) is not accepted. For performance reasons the STA will update its STA Time only if the absolute difference between AP time and STA Time is more then 20 seconds.

If the device has a trusted system time, then this system time is hourly stored in non-volatile memory (SAVED_TIME).

During bootup, device (AP or STA) updates it's local time with above stored time (SAVED_TIME) from non-volatile memory (if available) or otherwise starts up with firmware default system time.

Note that the AP announces its system time as a combination of a reference epoch and TSF timer (Timing Synchronization Function). The current epoch is calculated by adding the TSF timer to the reference epoch. When the STA receives a time advertisement from the AP it dumps the reference epoch + TSF to the Syslog.

4.31 Dynamic VLAN

A common usecase to separate different types of users is by providing multiple SSIDs and mapping the different SSIDs to different VLANs. See Multi SSID and VLAN. However for larger numbers of different types of users one reaches the limit of 8 respective 16 SSIDs per radio. Also adding and removing user groups requires a reconfiguration of all Access Points. In addition having many SSIDs beaconing on the air reduces the available air-time.

A solution to this is Dynamic VLAN. Dynamic VLAN is available when the AP is configured to authenticate via 802.1X. When authenticating via 802.1X a RADIUS server decides if a client may connect to the network. Dynamic VLAN allows the RADIUS server to provide information which VLAN is used for a specific station. This information is parsed from the following RADIUS attributes based on RFC 3580 and RFC 2868:

- Tunnel-Type (value 13 = VLAN)

- Tunnel-Medium-Type (value 6 = IEEE 802)

- Tunnel-Private-Group-ID (value VLANID as a string)

Dynamic VLAN requires the wireless interface to be bridged. This feature does not work on an unbridged wireless interface.

Also this feature does not work in combination with 802.11r, since the required RADIUS attributes are not transferred from one AP to another when a client roams.

4.32 Wired 802.1X Authentication

Wired 802.1X authentication allows to restrict access to physical interfaces to authorized users only.

A common use case is that a laptop is connected to the device for maintenance on an unused Ethernet port. For this to work, the device needs to be connected to a local network through another Ethernet port which provides access to a RADIUS server. The laptop needs to run a Supplicant in order to gain access to the port.

Also see Port-based Network Access Control (802.1X) for a more detailed description of how 802.1X works in the context of wireless connectivity.

Note RT-280 devices do not offer 802.1X authentication on the two fiber ports X1 (eth1) and X2 (eth2). Wired 802.1X authentication is only possible on the copper port X3 (eth0). However unused ports may also be disabled.

4.32.1 Operation

When enabled, the Ethernet port on which wired 802.1X authentication is performed is blocked until the connecting Supplicant successfully authenticates itself against the configured RADIUS server(s).

The device acts as Authenticator and the connecting Supplicant needs to be connected directly to the port (without network switch). Upon successful authentication by the Supplicant, the port is opened. It is closed after a configurable time (cfgNetEth802dot1xEapReauthPeriod), unless it is reauthenticated by the Supplicant. The auto-closing time may also be specified by configuring the RADIUS attribute 'Session-Timeout'. The smaller value of the two is used. In addition the RADIUS attribute 'Termination-Action' may be configured on the RADIUS server to control the reauthentication behaviour (0: terminate immediately, 1: initiate reauthentication).

The EAP reauthentication process timeout is as follows:

- 1st try (t=0)
- 2nd retry: 3s (t=3)
- 3rd retry: 6s (t=9)
- 4th retry: 12s (t=21)
- 5th retry: 20s (t=41)
- 6th retry: 20s (t=61)

- 7th retry: 20s (t=81)

After approx. 81s the session is finally terminated if no reauthentication was possible.

The port is automatically locked when the port loses its link, e.g. when a laptop is unplugged.

4.32.2 Multi Radius Auth-Server Operation

When the wired 802.1X authentication is configured to operate with multiple RADIUS authentication servers, the fallback mechanism is set to a response timeout of 1 second and a try-count of 2. When the operator initiates the authentication process to gain access to the port, the operation is as follows:

1. Request is sent to primary server (first entry in cfgWlan802dot1xAuthServerTable)
2. Upon missing response after 1 second, the request is repeated
3. After another 1 second without response, the next RADIUS server is selected (next referenced entry in cfgWlan802dot1xAuthServerTable)
4. Repeat the above for additional referenced entries in the table.

As a result, the fallback to the additional RADIUS server(s) happens with a delay of 2s.

4.32.3 Configuration

- cfgWlan802dot1xAuthSrvEnabled Enable wired 802.1X authentication
- cfgNetEth802dot1xOwnIpAddr Own IP address used as NAS-Identifier
- cfgNetEth802dot1xAuthServerParameter Auth server ID selector. One or more authentication servers may be configured in Table cfgWlan802dot1xAuthServerTable
- cfgNetEth802dot1xEapReauthPeriod The EAP reauthentication time, after which the port is closed unless the Supplicant reauthenicates

```
WESTERMO-SW6-MIB::cfgNetEth802dot1xEnabled.1 = 1
WESTERMO-SW6-MIB::cfgNetEth802dot1xOwnIpAddr.1 = 192.168.1.20
WESTERMO-SW6-MIB::cfgNetEth802dot1xAuthServerParameter.1 = 3
WESTERMO-SW6-MIB::cfgNetEth802dot1xEapReauthPeriod.1 = 3600

WESTERMO-SW6-MIB::cfgWlan802dot1xAuthSrvEnabled.0 = 1
WESTERMO-SW6-MIB::cfgWlan802dot1xAuthSrvId.0 = 3
```

```
WESTERMO-SW6-MIB::cfgWlan802dot1xAuthSrvIpAddr.0 = 192.168.1.2
WESTERMO-SW6-MIB::cfgWlan802dot1xAuthSrvPort.0 = 1812
WESTERMO-SW6-MIB::cfgWlan802dot1xAuthSrvSharedSecret.0 = wireless4more
```

Listing 4.36: *802.1X Authentication on eth1 with server on 192.168.1.2*

4.33 IP Routing

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

IbexOS has support for Static Routing, Static Multicast Routing, Equal Cost Multi Path Routing (ECMP), Policy Routing and monitoring of links/interfaces to update routes dynamically. Multiple routing tables are supported to be used with Policy Routing.

4.33.1 Simple Default Gateway

To set a simple static default gateway the parameter cfgRouteDefGateway may be used.
This parameter creates a static route for 0.0.0.0/0 via the specified gateway on the main routing table (table 254) with a metric of 0 (highest metric).

4.33.2 Static Routing Table

The table cfgRouteTable may be used to create static routing table entries. Multiple entries may point to the same destination, to create an ECMP pool (see Equal Cost Multipath Routing (ECMP)).

Each static route is composed of:

- cfgRouteTableDestinationNetwork
- cfgRouteTableGateway

cfgRouteTableDestinationNetwork defines the destination network in CIDR notation.

cfgRouteTableGateway specifies the gateway over which the destination is reachable. Alternatively an interface may be specified with cfgRouteTableInterface for e.g IPsec with Virtual Tunnel Interfaces.

The example below shows a static route to the 192.168.11.0/24 subnet using 192.168.1.2 as gateway.

```
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
```

Listing 4.37: *Static route*

Additional optional parameters are:

- cfgRouteTableMetric
- cfgRouteTableRoutingTables
- cfgRouteTableSource
- cfgRouteTableCarpId
- cfgRouteTableMonitor
- cfgRouteTableWeight

cfgRouteTableMetric defines the priority of the route. A lower value means a higher priority.

cfgRouteTableRoutingTables may be used to create the route on one or multiple routing tables (see Policy Routing). By default the main routing table 254 is used.

cfgRouteTableSource may be used to specify a specific locally existing source address for a route.

cfgRouteTableCarpId and cfgRouteTableMonitor may be used to bring routes up or down when the respective CARP or NLM instance goes up or down (see Common Address Redundancy Protocol (CARP) and Network Link Monitor (NLM)). CARP and NLM monitoring are exclusive.

When the route is part of an ECMP pool, cfgRouteTableWeight defines the weight of the gateway (see Equal Cost Multipath Routing (ECMP)).

4.33.3 DHCP Routing Table

The table cfgRouteDhcpTable may be used to specify the metric, the routing table(s), the monitor, and the weight for routes received via DHCP. When no entries in this table exist, the default values used by the DHCP clients are:

- cellular interfaces (wwan): table 254, metric 600, weight 1
- ethernet/wireless/other interfaces (eth, wlan, tep): table 254, metric 400, weight 1

- openvpn interfaces (ovpn): table 254, metric 200, weight 1

Use cfgRouteDhcpInterface to select the interface on which a dhcp-client is running. Subsequently define the metric (cfgRouteDhcpMetric), routing table(s) (cfgRouteDhcpTable), monitor (cfgRouteDhcpMonitor) and weight (cfgRouteDhcpWeight) for this dhcp-client.

cfgRouteDhcpMonitor is used to reference an NLM instance of type `route(3)`, see Monitor Type route.

When the default gateway received via this DHCP client is set to the same metric on a table where a default gateway already exists, an ECMP pool will be created. In this case cfgRouteDhcpWeight defines the weight of this gateway in the pool (see Equal Cost Multipath Routing (ECMP)).

4.33.4 Equal Cost Multipath Routing (ECMP)

An ECMP route is a route that has more than one gateway and/or interface to which connections are routed. Frames of a flow are routed toward a gateway based on a hash over the source and destination addresses.

ECMP is set up automatically whenever multiple gateways and/or interfaces for a destination are defined. Different sources may provide a gateway, e.g a static configured default gateway via cfgRouteTable and additionally a dynamically received default gateway via cfgRouteDhcpTable from together an ECMP pool.

The following example defines a Default Gateway via two different routers. The two different routers have different bandwiths, thus different weights are defined.

```
WESTERMO-SW6-MIB::cfgRouteTableEnabled.0 = 1
WESTERMO-SW6-MIB::cfgRouteTableDestinationNetwork.0 = 0.0.0.0/0
WESTERMO-SW6-MIB::cfgRouteTableGateway.0 = 192.168.2.2
WESTERMO-SW6-MIB::cfgRouteTableWeight.0 = 1

WESTERMO-SW6-MIB::cfgRouteTableEnabled.1 = 1
WESTERMO-SW6-MIB::cfgRouteTableDestinationNetwork.1 = 0.0.0.0/0
WESTERMO-SW6-MIB::cfgRouteTableGateway.1 = 192.168.2.3
WESTERMO-SW6-MIB::cfgRouteTableWeight.1 = 3
```

Listing 4.38: *ECMP Default Gateway*

The resulting routing table is shown below:

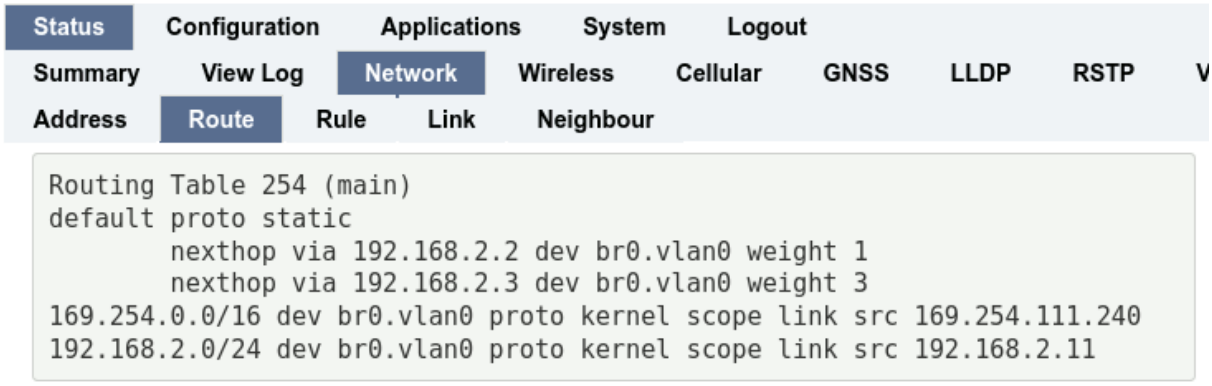


Figure 4.18: *The Default Gateway is an ECMP pool*

In this example if 4 connections are created, 1 connection is routed via the router at 192.168.2.2, and 3 connections are routed via 192.168.2.3.

4.33.5 Policy Routing

Policy Routing allows to use different routing tables based on:

- Source Network cfgRouteRuleFrom
- Destination Network cfgRouteRuleTo
- Input Interface cfgRouteRuleInputInterface
- QoS (TOS) cfgRouteRuleTos.
- SKB Mark cfgRouteRuleMark.

To allow overlapping rules, each routing rule has a preference defined by cfgRouteRulePreference. A Lower number means a higher priority. Overlapping rules with the same preference have undefined behaviour.

Preferences from 10000 to 1999999999 are available for the user.

The list of internally used preferences is:

- 0 (internal): local, required to receive frames
- 5000 to 5255 (internal): NLM, required for monitoring
- 2100000220 (internal): ipsec, directs frames into IPsec tunnel

- 2100000221 (internal): ipsec.unreachable, drops frames when tunnel is down
- 2100032766 (internal): main routing table
- 2100032767 (internal): default routing table

Frames that match the specified criteria are directed to the routing table defined by cfgRouteRuleLookupTable.

Rules on these routing tables may be defined via Static Routes (Static Routing Table) or DHCP Routes (DHCP Routing Table).

If no entry on a called routing table matches, the next routing table is called.

Example:

The main routing table (254) contains a default gateway via 10.0.0.1. On routing table 10000 a route for 192.168.3.0/24 via 10.0.0.2 exists.

There is a rule with preference 10000, defining that all traffic is processed by table 10000 (cfgRouteRuleFrom, cfgRouteRuleTo, cfgRouteRuleInputInterface and cfgRouteRuleTos set to `0.0.0.0/0` respectively `any`).

When a frame enters, it will be processed by table 10000. If the destination lies within 192.168.3.0/24, table 10000 will match and a routing decision is made (route via 10.0.0.2).

When the destination lies outside 192.168.3.0/24, table 10000 will not match and the routing process continues. It enters preference 2100000220 (table ipsec, no match), then preference 2100000221 (table ipsec.unreachable, no match) until it finally arrives at preference 2100032766 (table main) where it matches the defined default gateway and is routed via 10.0.0.1.

4.33.6 Static Multicast Routing

To forward mulicast traffic over a routing boundary, a multicast router is required. Use the table cfgMRouteTable to specify which multicast traffic shall be forwarded.

Define the interface on which mulicast traffic is received with the entry cfgMRouteTableInput. Optionally cfgMRouteTableSource may be used to limit allowed sources. Setting it to `0.0.0.0` will forward traffic from any source. cfgMRouteTableGroup defines which multicast destination is forwarded. Finally cfgMRouteTableOutput defines on which interface the multicast traffic is forwarded to.

4.33.7 Multi WAN and Failover Examples

To support failover or load balancing for multiple WAN interfaces, the Network Link Monitor (NLM) is a helper service to monitor different inputs such as interfaces state, reachability of IPs or signal levels. In combination with the routing tables, it is a powerful tool for fulfilling most desired use cases. This section contains some examples, which in particular address the use of the NLM.

4.33.7.1 Example 1 - WLAN Failover with DHCP Routing

When multiple gateways are available it is desirable to balance traffic between them (ECMP) or to do failover between them. This is an example how to do failover between Wireless and Cellular on an RT-630, RT-630-5G device. Traffic shall prioritise Wireless, and fall back to Cellular when Wireless is not available.

```
# Enable wwan interface and SIM slot
WESTERMO-SW6-MIB::cfgNetWwanEnabled.0 = 1
WESTERMO-SW6-MIB::cfgCellSimSlotEnabled.0 = 1

# Enable wireless interface and configure it to the desired AP
WESTERMO-SW6-MIB::cfgNetWlanEnabled.0 = 1
WESTERMO-SW6-MIB::cfgNetWlanBridge.0 = -1
WESTERMO-SW6-MIB::cfgWlanIfaceMode.0 = 1
WESTERMO-SW6-MIB::cfgWlanIfaceSsid.0 = multiwan-test
WESTERMO-SW6-MIB::cfgWlanIfacePassword.0 = multiwan-test-password
WESTERMO-SW6-MIB::cfgWlanIfaceScanList.0 = -2

# Enable DHCP client on wlan0
WESTERMO-SW6-MIB::cfgNetIpEnabled.1 = 1
WESTERMO-SW6-MIB::cfgNetIpAddr.1 = 0.0.0.0/0
WESTERMO-SW6-MIB::cfgNetIpProto.1 = 3
WESTERMO-SW6-MIB::cfgNetIpInterface.1 = wlan0

# Set dhcp-route for wlan0 to NLM monitor 0
WESTERMO-SW6-MIB::cfgRouteDhcpEnabled.0 = 1
WESTERMO-SW6-MIB::cfgRouteDhcpInterface.0 = wlan0
WESTERMO-SW6-MIB::cfgRouteDhcpMetric.0 = 400
WESTERMO-SW6-MIB::cfgRouteDhcpMonitor.0 = 0

# Set dhcp-route for wwan0 to NLM monitor 1
WESTERMO-SW6-MIB::cfgRouteDhcpEnabled.1 = 1
WESTERMO-SW6-MIB::cfgRouteDhcpInterface.1 = wwan0
WESTERMO-SW6-MIB::cfgRouteDhcpMetric.1 = 600
WESTERMO-SW6-MIB::cfgRouteDhcpMonitor.1 = 1

# Enable NLM
WESTERMO-SW6-MIB::cfgNlmGlblEnabled.0 = 1
```

```
# NLM monitor 0: Use gateway received via DHCP from wlan0 as monitor address.
# Send ICMP request every 200ms and mark as down after 3 failed requests.
WESTERMO-SW6-MIB::cfgNlmMonEnabled.0 = 1
WESTERMO-SW6-MIB::cfgNlmMonInterval.0 = 200
WESTERMO-SW6-MIB::cfgNlmMonCount.0 = 3
WESTERMO-SW6-MIB::cfgNlmMonType.0 = 3
WESTERMO-SW6-MIB::cfgNlmMonDestination.0 = 0.0.0.0
WESTERMO-SW6-MIB::cfgNlmMonUpAction.0 = 8000
WESTERMO-SW6-MIB::cfgNlmMonDownAction.0 = 8000

# NLM monitor 1: Use 8.8.8.8 as monitor address because the gateway received via
# DHCP from wwan0 does not respond to ICMP requests.
# Send ICMP request every 1000ms and mark as down after 3 failed requests.
WESTERMO-SW6-MIB::cfgNlmMonEnabled.1 = 1
WESTERMO-SW6-MIB::cfgNlmMonInterval.1 = 1000
WESTERMO-SW6-MIB::cfgNlmMonCount.1 = 3
WESTERMO-SW6-MIB::cfgNlmMonType.1 = 3
WESTERMO-SW6-MIB::cfgNlmMonDestination.1 = 8.8.8.8
WESTERMO-SW6-MIB::cfgNlmMonUpAction.1 = 8000
WESTERMO-SW6-MIB::cfgNlmMonDownAction.1 = 8000
```

Listing 4.39: *Wireless to Cellular Failover*

The DHCP routing table configuration in Figure 4.19 is referenced to a NLM monitor in Figure 4.20. If the monitor state transitions to **UP**, the default route received by the DHCP client on the specified interface is created on the defined routing table 254.

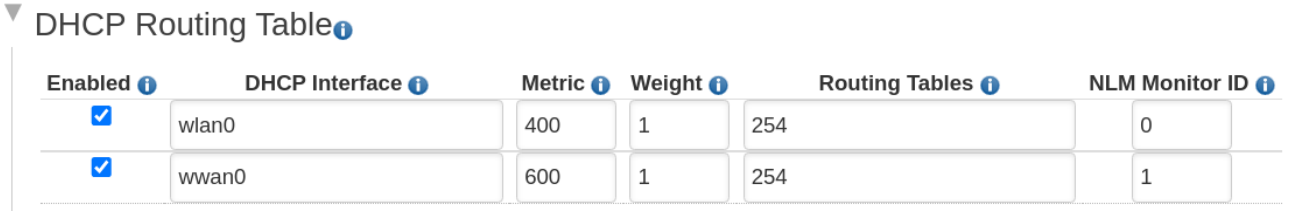


Figure 4.19: *DHCP Routing Table Configuration with references to NLM*

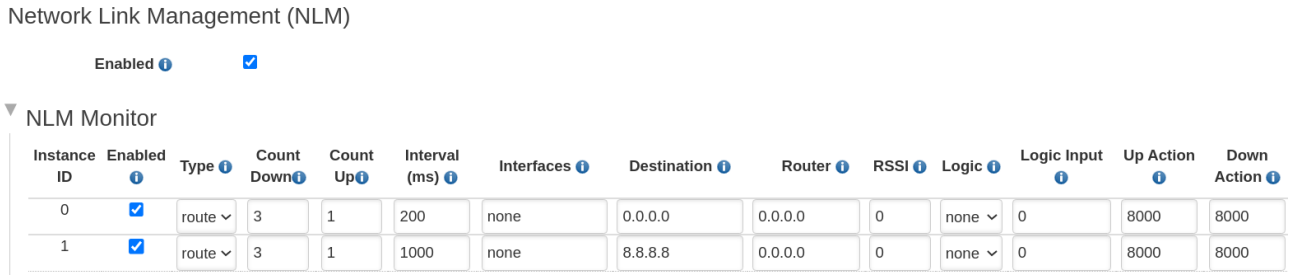


Figure 4.20: *NLM Configuration with routing rules for wlan0 and wwan0*

Figure 4.21 shows the state of the routing tables, Figure 4.22 shows the state of NLM, when both wlan0 and wwan0 are up.

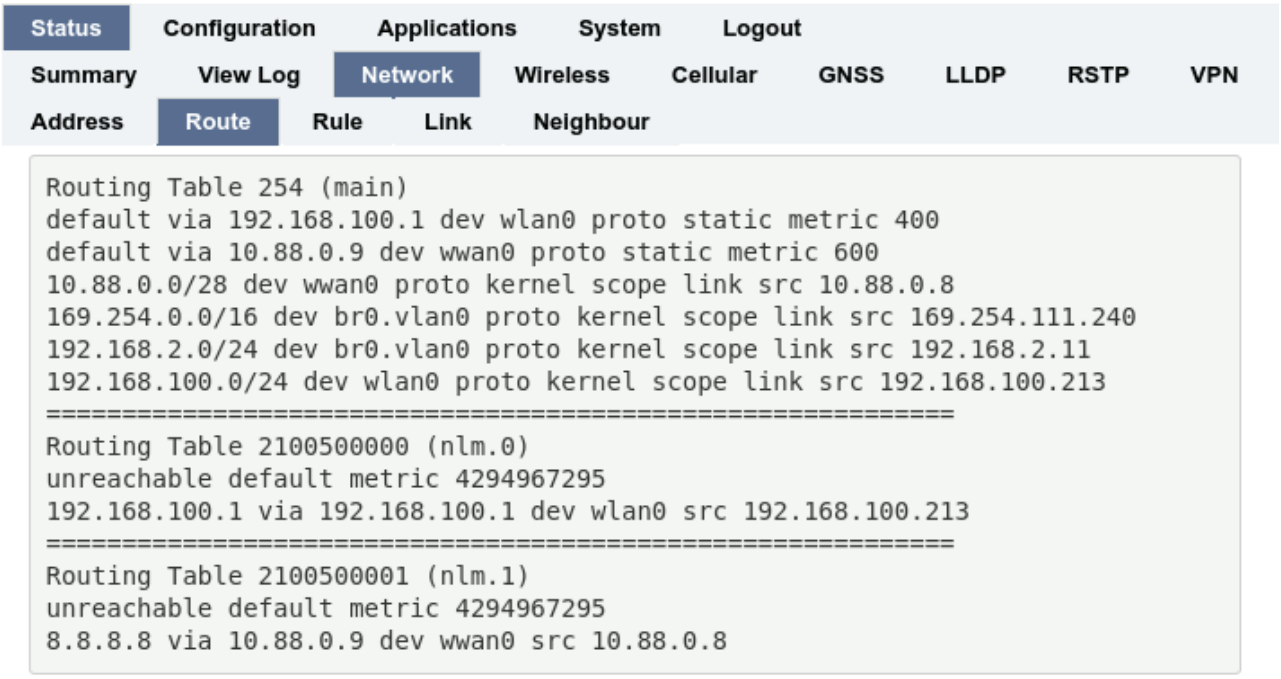


Figure 4.21: *Routing Tables when wlan0 and wwan0 up*

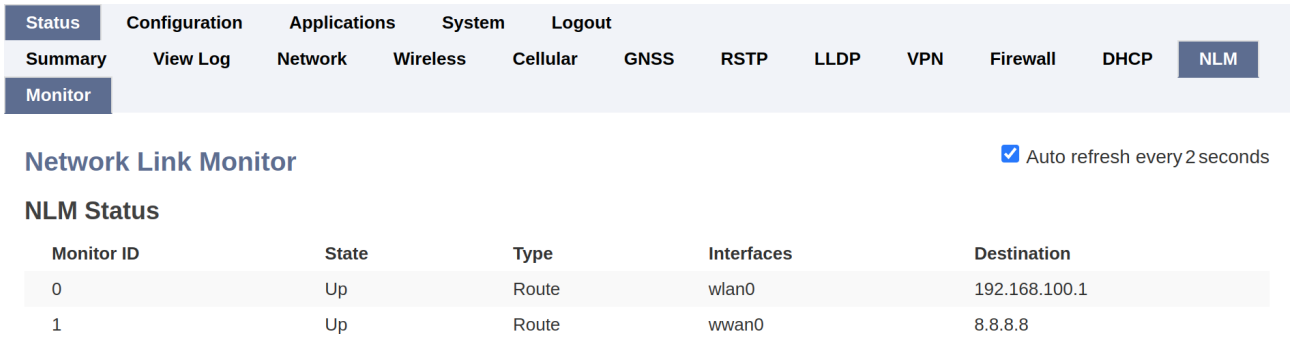


Figure 4.22: *NLM status view when wlan0 and wwan0 are up*

The interface wlan0 received via DHCP the IP address 192.168.100.213 with the default gateway 192.168.100.1. The interface wwan0 received via DHCP the IP address 10.88.0.8 with the default gateway 10.88.0.9.

The gateway from wlan0 is set up with a metric of 400 and the gateway from wwan0 with a metric of 600. A lower number implies a higher priority.

NLM automatically set additional routing tables up to monitor the state of the two links according

to the configuration. For wlan0 the reachability of 192.168.100.1, and for wwan0 the reachability of 8.8.8.8 are monitored to determine the state of the link.

With interface wlan0 out of reach of an AP, the link is down. Figure 4.23 shows the state of the routing tables with wlan0 down, the default gateway via wlan0 does not exist.

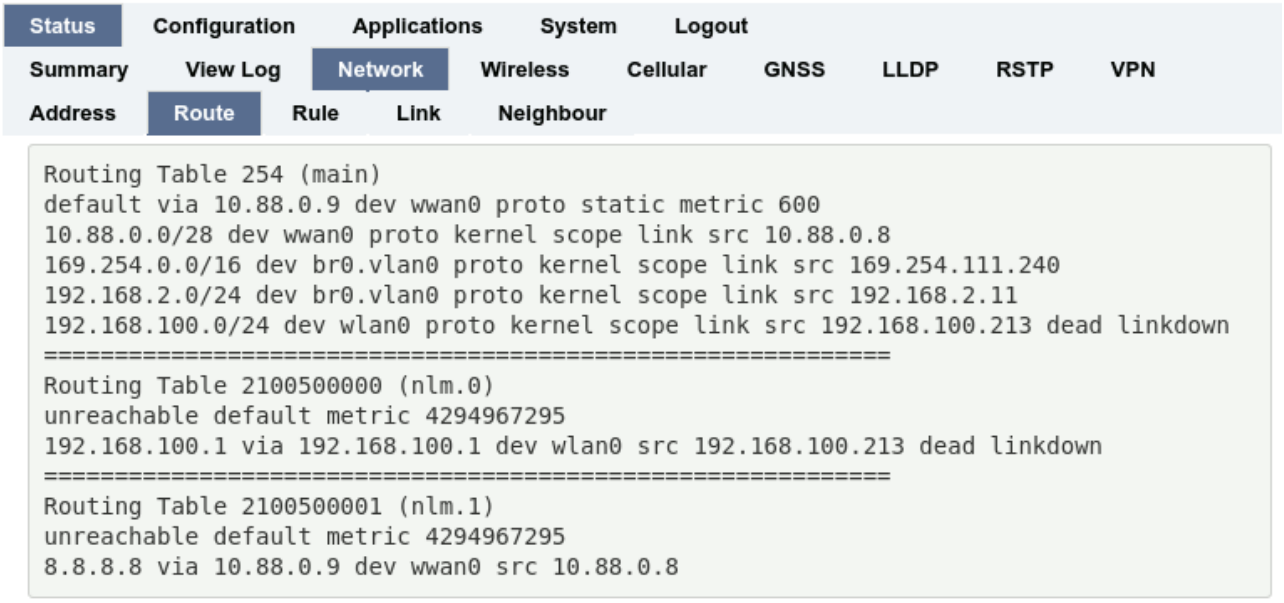


Figure 4.23: *Routing Tables with wlan0 down*

4.33.7.2 Example 2 - Static Routing with Logic

In this example, the client has a permanent connection via the cellular interface. In addition, if the vehicle is in a tunnel, station or a depot with good wireless coverage, it should use the wireless link instead. The default route is specified by the cellular connection. Once the signal strength of the radio link exceeds a certain value, the default gateway with a better metric should be added to the routing table.

The single static routing table entry only needs to define the desired gateway and the reference to the NLM ID cfgNlmMonIndex (in this case the index is 2).

```
WESTERMO-SW6-MIB::cfgRouteTableEnabled.0 = 1
WESTERMO-SW6-MIB::cfgRouteTableDestinationNetwork.0 = 0.0.0.0/0
WESTERMO-SW6-MIB::cfgRouteTableGateway.0 = 192.168.100.1
WESTERMO-SW6-MIB::cfgRouteTableMetric.0 = 0
WESTERMO-SW6-MIB::cfgRouteTableRoutingTables.0 = 254
WESTERMO-SW6-MIB::cfgRouteTableMonitor.0 = 2
```

Listing 4.40: *Static Routing Table with reference to NLM Logic Rule*

Figure 4.24: *Static Routing Table Configuration with reference to NLM Logic Rule*

The Listing 4.41 shows a reduced parameter set to configure three monitor rules to achieve the desired behavior of the use case. The same configuration is also illustraded in Figure 4.25.

```
WESTERMO-SW6-MIB::cfgNlmGlblEnabled.0 = 1
WESTERMO-SW6-MIB::cfgNlmMonEnabled.0 = 1
WESTERMO-SW6-MIB::cfgNlmMonEnabled.1 = 1
WESTERMO-SW6-MIB::cfgNlmMonEnabled.2 = 1
WESTERMO-SW6-MIB::cfgNlmMonInterval.0 = 200
WESTERMO-SW6-MIB::cfgNlmMonInterval.1 = 200
WESTERMO-SW6-MIB::cfgNlmMonType.0 = 3
WESTERMO-SW6-MIB::cfgNlmMonType.1 = 4
WESTERMO-SW6-MIB::cfgNlmMonType.2 = 5
WESTERMO-SW6-MIB::cfgNlmMonInterfaces.0 = wlan0
WESTERMO-SW6-MIB::cfgNlmMonInterfaces.1 = wlan0
WESTERMO-SW6-MIB::cfgNlmMonDestination.0 = 8.8.8.8
WESTERMO-SW6-MIB::cfgNlmMonUpAction.2 = 8000
WESTERMO-SW6-MIB::cfgNlmMonDownAction.2 = 8000
WESTERMO-SW6-MIB::cfgNlmMonCountUp.1 = 5
WESTERMO-SW6-MIB::cfgNlmMonRssi.1 = 70
WESTERMO-SW6-MIB::cfgNlmMonLogic.2 = 4
WESTERMO-SW6-MIB::cfgNlmMonLogicInput.2 = 0,1
WESTERMO-SW6-MIB::cfgNlmMonRouter.0 = 192.168.100.1
```

Listing 4.41: *NLM Logic with Route and RSSI Monitors*

- **ID 0** The monitor type `route(5)` observes the connection if the destination cfgNlmMonDestination is reachable over the gateway defined in cfgNlmMonRouter via the interface cfgNlmMonInterfaces. Otherwise the destination would be reached via the default active gateway on cellular. The monitor state becomes **UP** when the destination is reachable, but no UP Action is triggered.

- **ID 1** The monitor type `rssi` observes the signal strength of the current client connection. If the RSSI value is higher than the value defined in cfgNlmMonRssi, the monitor state becomes **UP**, but no UP Action is triggered.

- **ID 2** The monitor type `logic` combines the two referenced rules in cfgNlmMonLogicInput. The logic operation `and` defined in cfgNlmMonLogic sets the monitor state to **UP** when the referenced monitors are also **UP**. The UP Action `8000` is triggered which brings the referenced route of cfgRouteTableMonitor up. Vice-versa, when one of the referenced monitors transitions to **DOWN**, the logic monitor executes the down action `8000` which removes the assigned route.

NLM Monitor

Instance ID	Enabled	Type	Count Down	Count Up	Interval (ms)	Interfaces	Destination	Router	RSSI	Logic	Logic Input	Up Action	Down Action
0	☑	route	3	1	200	wlan0	8.8.8.8	192.168.100.1	0	none	0	0	0
1	☑	rssi	3	5	200	wlan0	0.0.0.0	0.0.0.0	70	none	0	0	0
2	☑	logic	3	1	5000	none	0.0.0.0	0.0.0.0	0	and	0,1	8000	8000

Figure 4.25: *NLM Logic Configuration with Route and RSSI Monitors*

The Figure 4.26 shows the routing table of the network when the monitor for the static route to the wireless gateway is present, but the connection is not established.

Status | Configuration | Applications | System | Logout

Summary | View Log | Network | Wireless | Cellular | GNSS | RSTP | LLDP | VPN | Firewall | DHCP | NLM

Address | Route | Rule | Link | Neighbour

Network Routes

Auto refresh every 2 seconds

```
Routing Table 254 (main)
default via 10.0.36.49 dev wwan0 proto static metric 600
10.0.36.32/27 dev wwan0 proto kernel scope link src 10.0.36.48
169.254.0.0/16 dev br0.vlan0 proto kernel scope link src 169.254.254.75
192.168.1.0/24 dev br0.vlan0 proto kernel scope link src 192.168.1.20
192.168.100.0/24 dev wlan0 proto kernel scope link src 192.168.100.20 dead linkdown
============================================================
Routing Table 2100500000 (nlm.0)
unreachable default metric 4294967295
8.8.8.8 via 192.168.100.1 dev wlan0 dead linkdown
```

Figure 4.26: *NLM Logic Status of Routes*

The Figure 4.27 shows the logical status of the NLM when the wireless connection is established but the RSSI signal requirements are not fulfilled. Thus, the route controlled by the logic monitor is not yet added.

Status | Configuration | Applications | System | Logout

Summary | View Log | Network | Wireless | Cellular | GNSS | RSTP | LLDP | VPN | Firewall | DHCP | NLM

Monitor

Network Link Monitor

Auto refresh every 2 seconds

NLM Status

Monitor ID	State	Type	Interfaces	Destination	Router	Logic Operator	Logic Input IDs	Info
0	Up	Route	wlan0	8.8.8.8	192.168.100.1			
1	Down	RSSI	wlan0					Threshold 70 RSSI
2	Down	Logic				AND	0,1	

Figure 4.27: *NLM Status of Logically Combined Monitors*

4.34 VLAN

The *IbexOS* 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.

4.34.1 Multi SSID and VLAN

The devices support multiple SSIDs one a single radio. It can broadcast up to 8 wireless networks per radio 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 4.28 below.

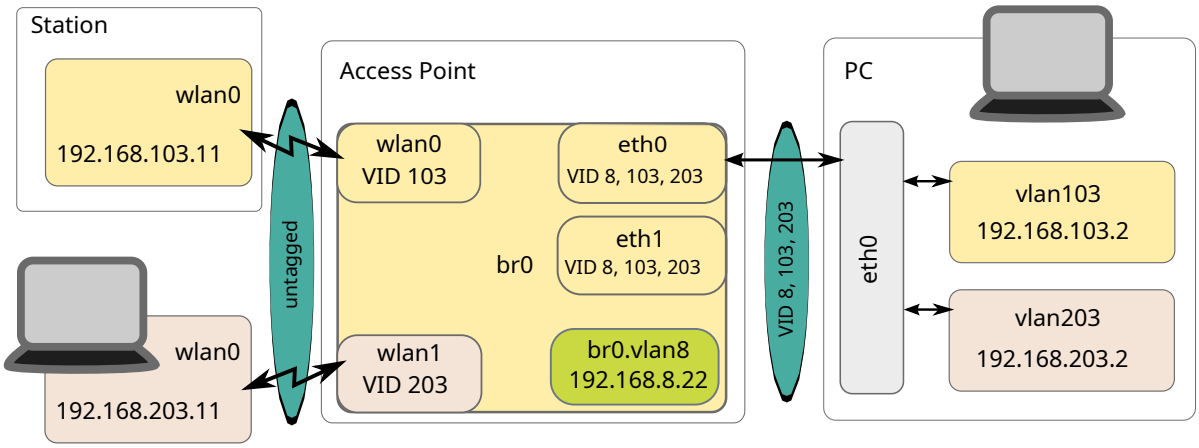


Figure 4.28: *Multi SSID and VLAN example setup*

```
# 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::cfgWlanIfaceSsid.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::cfgWlanIfaceDevice.1 = 0
WESTERMO-SW6-MIB::cfgWlanIfaceSsid.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
```

Listing 4.42: *Multiple SSID and VLAN*

4.35 Mobility

4.35.1 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:

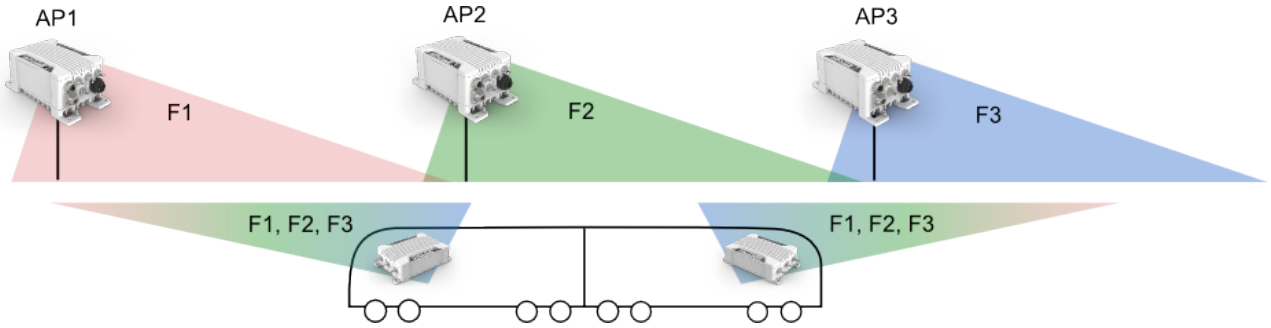
1. Loop through all frequencies defined in the cfgWlanIfaceScanList and
   a) tune the radio to a frequency on the cfgWlanIfaceScanList
   b) observe the channel for a few milliseconds. This is necessary to update the NAV (network allocation vector)
   c) send a Probe Request frame with SSID specified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

Once the client (STA) is successfully connected, it observes the Beacon frames and Pilot frames (when enabled) that are received from the associated Access Point. As long as the filtered RSSI level of the Beacon/Pilot frames are above the configured threshold (cfgWlanHoScanningLevel), the client stays connected. If the RSSI level is below the threshold or if too many Beacon frames are lost (cfgWlanIfaceBeaconMiss), the client will disconnect and scan for a better Access Point.

Note For installations where the Access Points locations are so far apart that no continuous coverage can be achieved, the cfgWlanHoScanningLevel) should be set to 0 so that the client is connected to an Access Point as long as possible.

4.35.2 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 Points, 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 and Pilot 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.

4.35.3 Handoff Filters

The client (STA) decides when to roam based on Beacon/Pilot 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 cfgWlanHoFilterLongX and cfgWlanHoFilterLongY. In combination with the cfgWlanIfaceBeaconInterval and the cfgWlanIfacePilotInterval, which are configured on the Access Point, the RSSI filter characteristic can be optimally tuned for the application. The RSSI filter takes both Beacon and Pilot frames into account if they are activated.

The filtered RSSI is calculated through a IIR low pass filter:

$$RSSI_{AVG} = \frac{1}{X+Y}(X * RSSI + Y * RSSI_{AVG}) \tag{4.1}$$

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.

The filtered Distance is calculated through a IIR low pass filter, the measurements are pre-filtered by a plausibility check:

$$Distance_{AVG} = \frac{1}{X+Y}(X * Distance_{AVG} + Y * Distance) \tag{4.2}$$

4.35.4 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).

4.35.4.1 Important Messages for Handoff

Key	Format	Syslog	SNMP Trap
cfgWlanDbgHandoff	Message Message Code: 434	yes	yes
cfgWlanDbgBeaconrssi	Message Message Code: 430	yes	no
cfgWlanDbgBeaconfiltered	Message Message Code: 431	yes	no
cfgWlanDbgRange	Distance	yes	no
cfgWlanDbgReports	Reports	yes	no

4.35.4.1.1 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>
```

This message is sent when the debug message for Range is enabled (see cfgWlanDbgRange). The reported value is the Distance, in units, to the associated counterpart.

- <uptime>: Time in milliseconds since boot-up
- <mac>: BSSID of the counterpart
- <raw>: Raw distance value
- <filtered>: Filtered distance value (see Handoff Filters, used for handoff decision)

4.35.4.1.2 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.

4.35.5 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.

4.35.5.1 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:

```
# 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::cfgWlan802dot11rR0KHHID.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::cfgWlan802dot11rR1KHHID.0 = 00:00:00:00:00:00
WESTERMO-SW6-MIB::cfgWlan802dot11rR1KHKey.0 = 000102030405060708090a0b0c0d0e0f
```

Listing 4.43: *Fast BBS Transition AP*

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 4.30.2.

4.35.5.2 Client (STA) 802.11r Configuration

The following configuration example configures a STA for using FT:

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

Listing 4.44: *Fast BBS Transition STA*

4.35.5.3 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

4.35.5.4 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 respecive 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.

4.36 Quality of Service (QoS)

The *IbexOS* 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.

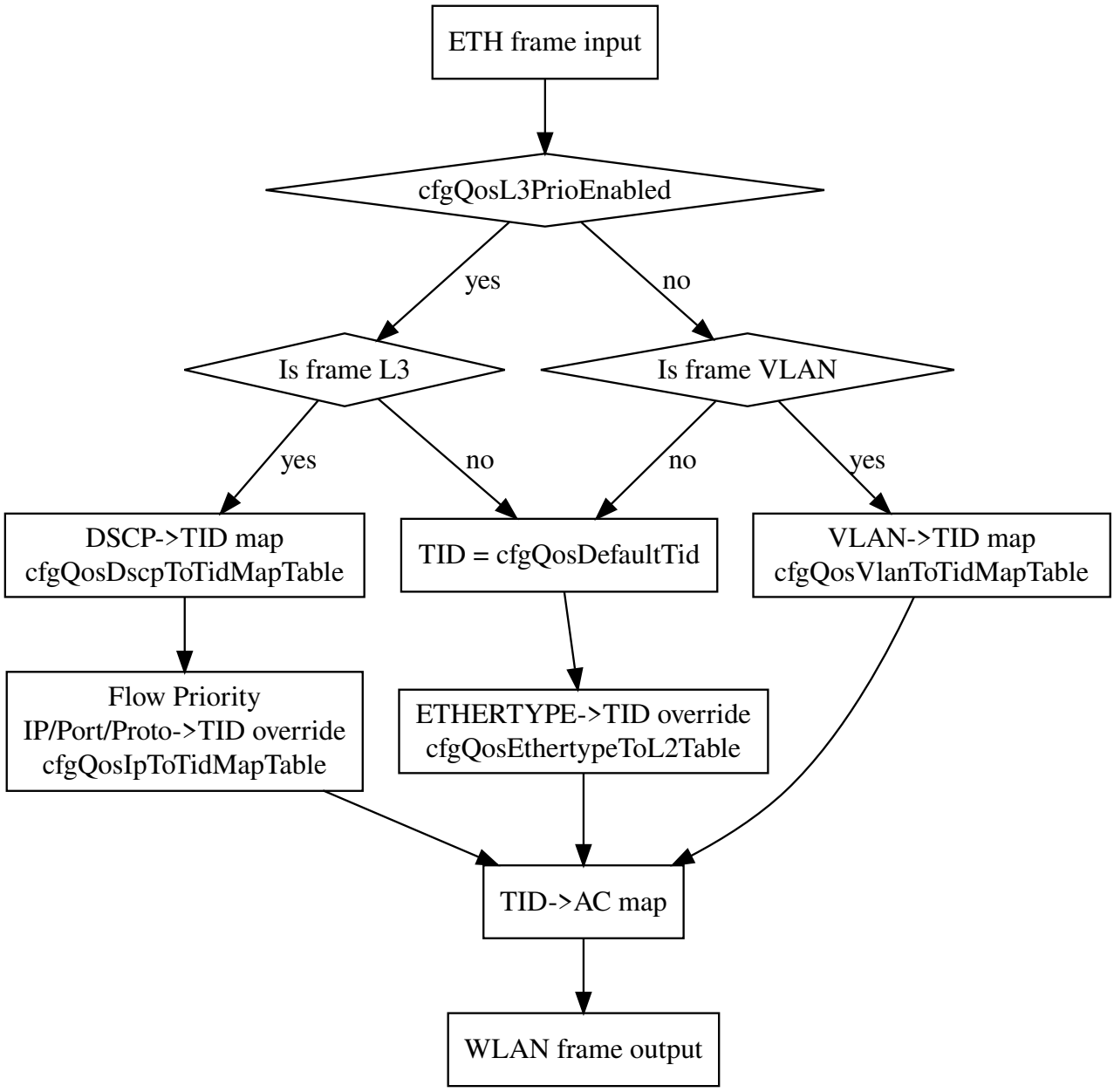
2a. cfgQosDscpToTidMapTable - Map from DSCP class selector (IP TOS) to wireless priority (TID).

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).

4. cfgQosEthertypeToL2Table - Map from ethertype 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 cfgQosL3PrioEnabled 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 cfgQosL3PrioEnabled is disabled, the 802.1p part of the 802.1q tag is mapped to the TID according to cfgQosVlanToTidMapTable. By default it is in a 1:1 manner.

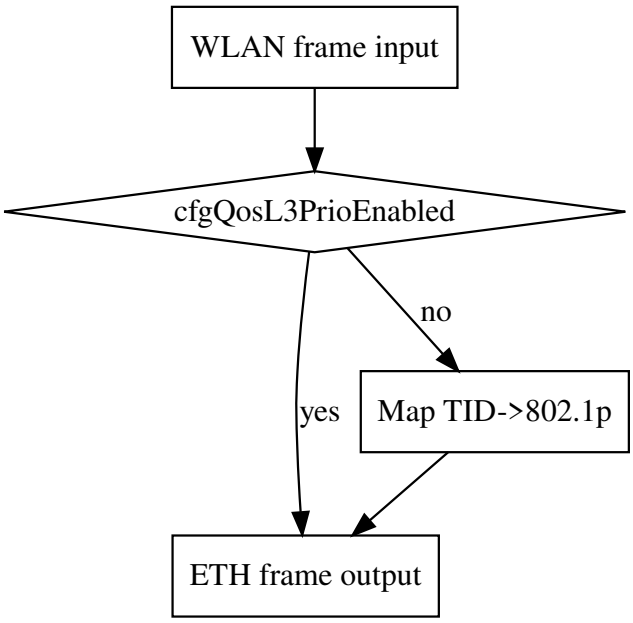
The TID is mapped to the AC classes according to the following table:

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

All frames that don't match the specified mode (e.g. non-IP frames when cfgQosL3PrioEnabled is enabled) are matched against cfgQosEthertypeToL2Table. If the ethertype of the frame matches an entry in the table, the TID is set to the value specified cfgQosEthertypeToL2Tid. If no match is found, then the TID is set to the value specified in cfgQosDefaultTid.

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.

4.37 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 *IbexOS* 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.

4.38 Network Link Monitor (NLM)

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

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.

4.38.1 NLM Monitor Types

4.38.1.1 Monitor Type phy

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.

4.38.1.2 Monitor Type icmp

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.

4.38.1.3 Monitor Type wlan

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.

4.38.1.4 Monitor Type route

The route monitor is active when cfgNlmMonType is set to 3. This is a polling based monitor.

It is the same as the icmp monitor with the difference that it binds the source interface. It has two operation modes: When used in conjunction with a DHCP-route entry in the table cfgRouteDhcpTable (see DHCP Routing Table), it expects the information necessary for operation to be supplied by a DHCP client. When used standalone, the required information has to be configured manually.

The DHCP client sends upon a successful bind or renew the interface it is active on, and the default gateway address it received to this monitor. This monitor binds the source interface and starts sending ICMP requests to the provided address. It is possible, that the received default gateway does not respond to ICMP requests. This is often the case with cellular networks. Set cfgNlmMonDestination to an alternative address that is reachable via the interface, or leave it on `0.0.0.0` to monitor the received default gateway.

Static routes can also be brought up and down by the route monitor. This can be useful if, for example, the connectivity of a wireless connection in combination with a cellular link is to be observed. In this use case, the default gateway points to the cellular network. To monitor the wireless connection, a gateway cfgNlmMonRouter and its interface cfgNlmMonInterfaces must be defined in addition to the monitored destination cfgNlmMonDestination. When the monitor of the wireless link is up the specified static routes are brought up.

4.38.1.5 Monitor Type rssi

The rssi monitor is active when cfgNlmMonType is set to 4. This is a polling based monitor.

It provides the means to specify an RSSI value in cfgNlmMonRssi and compare it to the current RSSI value of a wireless interface defined in cfgNlmMonInterfaces.

This allows to react to situations where the signal level gets lower, before it is too low. That may be a simple notification via cfgNlmMonTrap or a more active action like failover to another link.

4.38.1.6 Monitor Type logic

The logic monitor is active when cfgNlmMonType is set to 5. This is an event based monitor.

It allows to use other existing monitors as input (see cfgNlmMonLogicInput) and performs logical operation between those inputs.

The following logical operations are available in cfgNlmMonLogic:

- **none** - No logic operation. This essentially disables the monitor.
- **equal** - The monitor has the same state as a single referenced monitor. This allows to run multiple actions based on the state of a single monitor.
- **not** - Similar to **equal**, the **not** operation allows to reference a single monitor. It inverts the state of the referenced monitor and allows to run multiple actions based on the state of a single monitor. This type of logic is also useful to create a chain of more complex logic.
- All logical operators **or**, **and**, **nor** and **nand** take two or more referenced monitors as input and their state are the result of their respective logical operation between all input states.

4.38.2 NLM Monitor Actions

Monitor actions are executed when the state of the assigned monitor changes either to **UP** cfgNlmMonUpAction or **DOWN** cfgNlmMonDownAction. The action of a monitor can be disabled by the value `0`. This may be required when using a logic monitor that combines multiple monitors into a single action.

Actions are integer values that combine the type and, if applicable, the interface identifier.

4.38.2.1 Monitor Action CARP - 1xxx

To un-demote a CARP group in state **UP** or demote it in state **DOWN**, the reference to the CARP interface group cfgNetCarpLocalInterfaceGroup ID needs to be added to the action offset.

- **Offset**: 1000
- **x**: CARP group from 0 to 255

4.38.2.2 Monitor Action WLAN Interface - 20xx

Enable or disable Access Point operation on the referenced wifi interface.

Limitation: On 802.11n products all WLAN interfaces on radio0 are enabled or disabled, regardless of the specified WLAN interface.

- **Offset**: 2000
- **x**: WLAN interface from 0 to 63

4.38.2.3 Monitor Action Wireguard - 4xxx

Re-resolving the specified FQDN of the referenced peer of the specified interface in the monitor state **UP** or **DOWN**.

- **Offset**: 4000
- **x**: Wireguard Peer Index from 0 to 255

4.38.2.4 Monitor Action Routes - 8000/8001

Depending on the monitor action and state, all referenced routes of cfgRouteTableMonitor and/or cfgRouteDhcpMonitor will be brought either up or down. This action is applicable to monitor type cfgNlmMonType **route** or in combination with type **logic**.

Monitor Action	Value	Monitor State	Route State
cfgNlmMonUpAction	8000	UP	UP
cfgNlmMonDownAction	8000	DOWN	DOWN
cfgNlmMonUpAction	8001	UP	DOWN
cfgNlmMonDownAction	8001	DOWN	UP

4.38.3 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)).

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

Listing 4.45: *NLM Configuration for CARP Demote*

4.38.4 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.

```
WESTERMO-SW6-MIB::cfgNlmGlblEnabled.0 = 1
WESTERMO-SW6-MIB::cfgNlmMonEnabled.0 = 1
WESTERMO-SW6-MIB::cfgNlmMonInterval.0 = 1000
WESTERMO-SW6-MIB::cfgNlmMonCount.0 = 3
WESTERMO-SW6-MIB::cfgNlmMonType.0 = 1
WESTERMO-SW6-MIB::cfgNlmMonInterfaces.0 = br0.vlan0
WESTERMO-SW6-MIB::cfgNlmMonDestination.0 = 192.168.1.1
WESTERMO-SW6-MIB::cfgNlmMonUpAction.0 = 2000
WESTERMO-SW6-MIB::cfgNlmMonDownAction.0 = 2000
```

Listing 4.46: *NLM Configuration for Backbone Monitor*

4.39 DNS/DHCP Server

Each server instance is able to provide DNS functionality, DHCP functionality or both.

The DNS/DHCP configuration is described in cfgDhcp.

Use cfgDhcpDnsmasqCustomOptions to specify custom additonal dnsmasq options that are not available via provided configuration elements.

4.39.1 DNS Server

Enable or disable the DNS functionality of the server by setting cfgDhcpDnsmasqDnsEnabled.

The Listen Address (cfgDhcpDnsmasqDnsListenAddress) is set by default to 'auto', which binds the DNS server to all IP addresses on interfaces used to serve DHCP clients.

To use the DNS server standalone without DHCP functionality (cfgDhcpDnsmasqDhcpEnabled disabled) there are 3 options. Set cfgDhcpDnsmasqDnsListenAddress to:

- A list of addresses to bind to.
- 'wildcard' which binds to 0.0.0.0.
- 'auto' without referencing any DHCP scopes (e.g. cfgDhcpDnsmasqScopeParameter set to -1). This will bind any address which exists on the system.

Rebind Protection (cfgDhcpDnsmasqDnsStopDnsRebind) blocks upstream responses which reply with an internal address. Such responses may be used by an attacker to probe the local network and access local ports on the device itself.

Rebind Protection blocks responses resolving to:

Network	RFC	Usage
127.0.0.0/8	RFC 1122	loopback
0.0.0.0/8	RFC 5735	"this"
10.0.0.0/8	RFC 1918	private
172.16.0.0/12	RFC 1918	private
192.168.0.0/16	RFC 1918	private
169.254.0.0/16	RFC 3927	linklocal
192.0.2.0/24	RFC 1166	test
198.51.100.0/24	RFC 5737	test
203.0.113.0/24	RFC 5737	test
255.255.255.255/32	RFC 0919, RFC 0922	broadcast

In split-DNS situation it may be desirable to allow to respond with a blocked range. The option cfgDhcpDnsmasqDnsRebindDomainOk may be used to exclude domains and its subdomains from the rebind protection.

The DNS server can be configured to ask a specific DNS server for the requested domain. This configuration can be done in the cfgDhcpDomainOverrideTable.

The DNS server can responde with hard-coded IP address for a requested host. This configuration can be done in the cfgDhcpHostOverrideTable.

4.39.2 DHCP Server

Enable or disable the DHCP functionality of the server by setting cfgDhcpDnsmasqDhcpEnabled.

Use cfgDhcpDnsmasqScopeParameter to specify a reference to one or multiple scopes identified by cfgDhcpScopeId. Multiple scopes may have the same Scope ID to be handled by a single instance. Specify the interface to provide DHCP functionality by setting cfgDhcpScopeInterface.

Use cfgDhcpScopeStart and cfgDhcpScopeLimit to define the start address and the number of addresses provided by a scope. The start address is an offset from the network-address of the first IP address configured on the interface.

Use cfgDhcpScopeAutoGateway and cfgDhcpScopeAutoDns to simplify the configuration of the default gateway and DNS server. When they are enabled, the address of the interface used to serve a DHCP client will be sent in the DHCPOFFER.

Use cfgDhcpScopeDhcpOptions to configure arbitrary DHCP options. These custom DHCP options override configurations set by other fields. For example when cfgDhcpScopeGateway is set to an address, but cfgDhcpScopeDhcpOptions contains a configuration for DHCP option 3 as well, then the value set in cfgDhcpScopeDhcpOptions will take precedence.

```
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
```

Listing 4.47: *DHCP Server on br0.vlan0*

4.39.3 DHCP Relay

A DHCP Relay allows to forward DHCP requests to a remote DHCP server. This is useful in large installations with a central DHCP server that handles multiple endpoints that create a VPN connection to the DHCP server and do not run their own DHCP server.

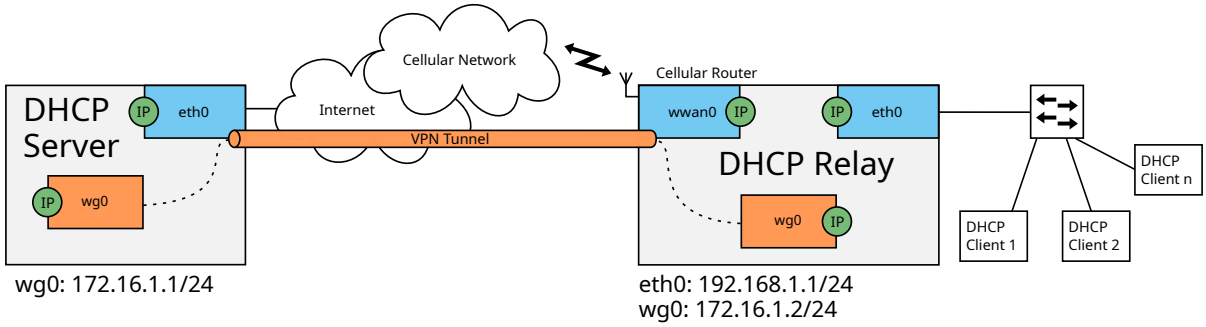


Figure 4.29: *DHCP Relay From eth0 to a Central DHCP Server*

To use the DHCP Relay, the DHCP service itself needs to be enabled with cfgDhcpGlobalEnabled. By default a DHCP Server instance exists that may need to be disabled with cfgDhcpDnsmasqDhcpEnabled.

The relay can be configured in the cfgDhcpRelayTable. Enable a relay instance with cfgDhcpRelayEnabled. Specify the interface on which the relay is listening on with cfgDhcpRelayInterface (eth0). cfgDhcpRelayLocalAddress (192.168.1.1) may be used to select the address configured on the interface that is used as Gateway IP Address (giaddr). Specify the address of the server to which requests are forwarded with cfgDhcpRelayServerAddress (172.16.1.1).

```
# Enable DHCP Service
WESTERMO-SW6-MIB::cfgDhcpGlobalEnabled.0 = 1

# Disable Default existing DHCP Server
WESTERMO-SW6-MIB::cfgDhcpDnsmasqDhcpEnabled.0 = 0

# Configure DHCP Relay
WESTERMO-SW6-MIB::cfgDhcpRelayEnabled.0 = 1
WESTERMO-SW6-MIB::cfgDhcpRelayInterface.0 = eth0
WESTERMO-SW6-MIB::cfgDhcpRelayLocalAddress.0 = 192.168.1.1
WESTERMO-SW6-MIB::cfgDhcpRelayServerAddress.0 = 172.16.1.1
```

Listing 4.48: *DHCP Relay From eth0 to a Central DHCP Server*

4.40 NTP Client/Server

The NTP client and server parameters descriptions can be found under cfgNtp.

Use cfgNtpEnabled to globally enable or disable NTP operation.

While NTP is enabled, all DHCP clients request the NTP option (DHCP option 42), thus the device can use servers that are not manually configured.

To manually define the NTP servers to be used by the NTP client, configure the cfgNtpClientTable table. This table allows to specify in cfgNtpClientHost either IPv4 servers, or an FQDN that may resolve to a pool of servers. If the FQDN resolves to multiple servers, the first 4 unique responses will be used.

On cellular devices, the NMEA protocol of the GNSS service can be selected as an additional time source by enabling cfgNtpClientNmeaEnabled. It requires that GNSS service cfgGnssGpsdEnabled is running and the corresponding NMEA sentences (GxGGA and/or GxRMC) are enabled.

To act as a NTP server for other devices, the parameter cfgNtpServerEnabled may be used. To force the server to provide time, even if the local time is not synchronized, the parameter cfgNtpServer-LocalReference allows to do that. While unsynchronized in Local Reference mode, a stratum of 10 is announced.

4.41 Simple Service Discovery Protocol

SSDP allows the device to be discovered by Microsoft (Windows 8 and up) devices.

It accomplishes this without assistance of server-based configuration mechanisms, such as Dynamic Host Configuration Protocol (DHCP) or Domain Name System (DNS) and without special static configuration of a network host.

It is enabled by default.

4.42 Service Indicators and Counters

4.42.1 SNMP Trap

The software provides notifications by means of SNMP Notifications as either Traps or Informs (see cfgSnmpTrapType).

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)

4.42.2 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

4.43 Logging Features

The *IbexOS* supports extensive logging features into Syslog. The Syslog is available in the Web Interface / via Web API, via CLI or it can be sent to a remote server (RFC 5424 and RFC5425). When using the TLS encrypted secure syslog (RFC5425) the key and certificates from the Certificate Store are used.

Multiple 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 can be configured with cfgLogRemoteProtocol.

When setting the cfgLogRemoteProtocol to tcptls(2), a key/certificate pair and a CA (or multiple CAs) are required. These can be selected from the Certificate Store with the parameters cfgLogRemoteCaIds and cfgLogRemoteCertId. CRL handling and other TLS specific configuration, including used ciphers can be set with cfgLogRemoteCrlExpiryExtension, cfgLogRemoteTlsControlParams and cfgLogRemoteCiphers.

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). Please refer to Security Log Messages for more information.

- **commissioning**: This type contains all message which are marked as commissioning messages (LOCAL1). Please refer to Commissioning Log Messages for more information.

For all 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).

4.44 Wireless Link Monitor

The *IbexOS* 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 cfgWlanGlblLinkmonitorInterval globally.

One message per connection is reported. The format of the message is described in Message Code: 130.

4.45 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.

4.45.1 Configuration of the Inter-Carriage Link Application

The ICL application supports multiple configuration modes:

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

4.45.1.1 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 4.30.

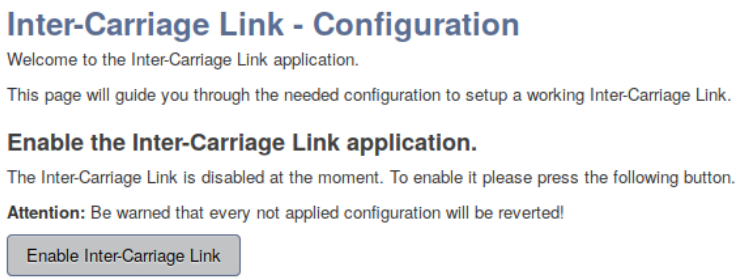


Figure 4.30: *Enable page*

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 4.31 where you will be able to customise a list of settings.

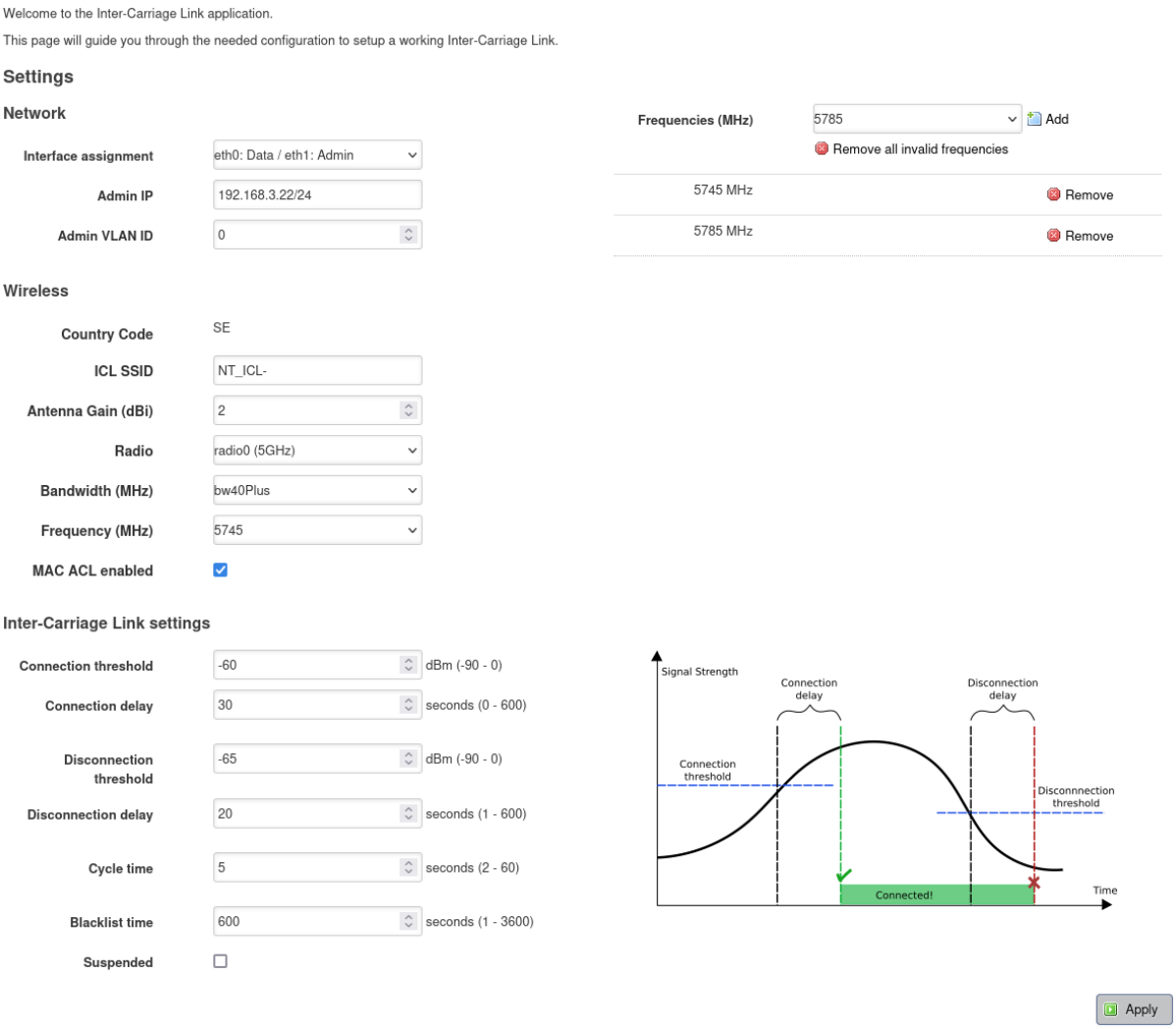


Figure 4.31: *Configuration page*

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.

Country Code that has been selected during initial setup of the device. This defines which frequencies are available. If the Country Code has to be changed use the Advanced Configuration.

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.

Radio selects on which radio to operate when multiple radios are available.

Bandwidth defines the preferred bandwidth. Depending on the product different rates are available. Always available are bw20, bw40Plus and bw40Minus. Additional bandwidths that may be available based on the used radio are bw80 and bw160. For more details, see cfgWlanDevBandwidth.

Frequency defines the primary operating frequency and depends on the selected bandwidth. See cfgWlanDevFrequency for more details.

MAC ACL enabled When the ACL is enabled only MAC addresses and ranges specified in cfgWlanAclWhiteTable are allowed to connect. By default the 3 ranges 00:14:5a:00:00:00/24, 00:07:7c:00:00:00/24 and 00:11:b4:00:00:00/24 are allowed. Use the **Configuration - Advanced** page to change these entries.

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 4.32, 4.33, 4.34 and 4.35 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.

Figure 4.32: *Status page: Scanning for an ICL partner*

Figure 4.33: *Status page: Evaluating one or more ICL partners*

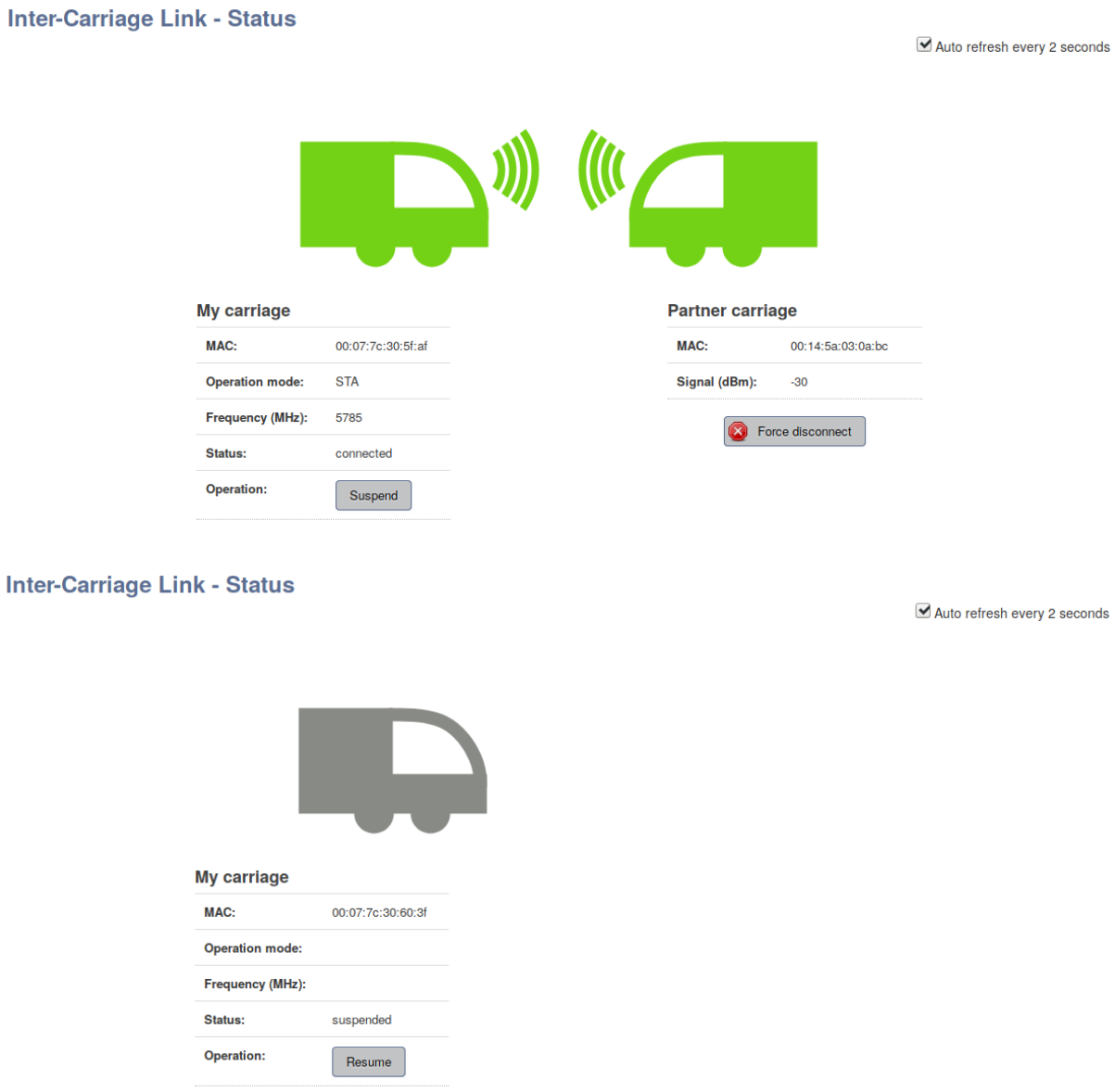


Figure 4.35: *Status page: ICL operation suspended*

Suspend

If the ICL functionallity 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 established a Inter-Carriage Link. The blacklist entries can be flushed by pressing the *Clear Blacklist* button or by setting rpcIclClearBlacklist 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 rpcIclForceDisconnect to 1. When operating as AP, 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*.

4.46 Global Navigation Satellite System

All products that support cellular communication come with a Global Navigation Satellite System (GNSS), see section 1.4. 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:

```
WESTERMO-SW6-GNSS-MIB::cfgGnssGpsdEnabled.0 = 1
WESTERMO-SW6-GNSS-MIB::cfgGnssGpsdAddress.0 = 0.0.0.0:2947
```

Listing 4.49: *Enable gpsd*

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.

4.46.1 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.

$$messageSamplingRate[Hz] = \frac{cfgGnssDevMsgsNmeaRate}{cfgGnssDevMeasurementPeriod[ms]} \times 1000 \quad (4.3)$$

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

4.46.2 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, use the parameter cfgGnssDevMsgsNmeaEnabled or set the related cfgGnssDevMsgsNmeaRate to 0.

4.46.3 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 type set cfgGnssDevMsgsUbxEnabled and cfgGnssDevMsgsUbxType accordingly and set the cfgGnssDevMsgsUbxRate greater than 0. In order to disable specific messages, use the parameter cfgGnssDevMsgsUbxEnabled or set the cfgGnssDevMsgsUbxRate to 0. The accessible UBX messages are listed in the description of cfgGnssDevMsgsUbxType.

4.46.4 GPSD Protocol

The GPSD project provides several libraries in different programming languages for accessing a remote `gpsd` client for querying. However, this protocol example is intended to help retrieve GNSS data directly from a TCP/IP session with helper tools like `netcat` or `PuTTY`.

For the full protocol description and library references, visit https://gpsd.gitlab.io/gpsd/client-howto.html.

Opening a TCP/IP session to the `gpsd` binding port will ship a banner that looks something like this:

```
{
    "class": "VERSION",
    "release": "3.19",
    "rev": "3.19",
    "proto_major": 3,
    "proto_minor": 14
```

```
}
```

The gpsd output is disabled by default, to enable recording, enter the following command. The data collection will remain after closing the session, it is best practice to set the watch command to `false` before leaving it.

```
?WATCH={"enable":true}

output:
{
    "class": "DEVICES",
    "devices": [
        {
            "class": "DEVICE",
            "path": "/dev/ttyACM0",
            "driver": "NMEA0183",
            "activated": "2022-06-14T13:49:08.203Z",
            ...
        }
    ]
}
{
    "class": "WATCH",
    "enable": true,
    "json": false,
    "nmea": false,
    ...
}
```

This will not generate any output, but the data can be retrieved by a poll command.

```
?POLL;

output:
{
    "class": "POLL",
    "time": "2022-06-14T13:51:17.493Z",
    "tpv": [
        {
            ...
        }
    ],
    "sky": [
        ...
        {
            "satellites": [
                {
```

```
                ...
            },
            ...
        ]
    }
]
}
```

Instead of manually polling the position and satellite information, the watch command can be extended with an output format. Defining the format type will enable continuous reporting, based on the measurement and reporting rate of the defined NMEA sentences.

```
?WATCH={"enable":true, "nmea":true}
```

Reporting is available in either `nmea` and/or `json` format.

4.47 Public Wireless Network (PWN)

The Ibex-1310, Ibex-1510, Ibex-3510, Ibex-1520, Ibex-3520 products are designed and optimized for Wi-Fi hotspots. They support amongst other things dual/triple concurrent operation in the 2.4GHz, 5GHz and 6GHz wireless bands, Multi-SSID mode and MU-MIMO.

4.47.1 Client Steering

Client Steering is a feature that makes use of the Radio Resource (802.11k) and the BSS Transition Management (802.11v) standards in order to steer clients (STAs) among various Basic Service Sets (BSS) to achive an overall improved wireless performance for all participating STAs in the same Extended Service Set (ESS). It combines Band Steering on a single AP and steering of STAs among multiple APs within the same Extended Service Set (ESS).

The core functionalities of Client Steering are:

- Synchronization of neighbor reports between the involved APs
- Policy-based decisions for probe-, association- and authentication requests received from STAs
- Requesting STAs to roam to a different AP based on signal strength
- Steering to a different Basic Service Set (BSS) based on channel load

In order to use the Client Steering feature it needs to be enabled by the cfgWlanCsteerEnabled parameter.

4.47.1.1 Steering the Clients

There are basically three options for an AP to steer an STA to another AP:

- Provide information about alternative APs to the STAs (802.11k),
- Actively ask the STA to connect to another AP (802.11v), or
- Disassociate the STA.

Provided the STAs support the 802.11k and/or the 802.11v standard, the steering of the STAs are expected to work just fine. However, neither standard regulates exactly how a STA must act. Thus, if a STA does not respond as expected or support any of the standards, the AP's only option is to kick the STA out by issuing a disassociation request.

4.47.1.2 Band Steering

Band Steering encourages STAs that are capable of both 5GHz and 2.4GHz bands to connect to the faster 5GHz band and leave the 2.4GHz band less-crowded for those STAs that support the 2.4GHz band only. Inactive STAs of a traffic overloaded 2.4GHz 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 low and a stronger 2.4GHz band is available, STAs will be steered to the 2.4GHz band until the 5GHz band is no longer overloaded. Such an approach is expected to improve the wireless performance for all STAs participating in the same ESS.

Listing 4.50 shows how Band Steering can be enabled on a single AP.

```
WESTERMO-SW6-MIB::cfgNetWlanEnabled.0 = 1
WESTERMO-SW6-MIB::cfgNetWlanEnabled.1 = 1
WESTERMO-SW6-MIB::cfgWlanIfaceSsid.0 = simple-bsteer
WESTERMO-SW6-MIB::cfgWlanIfaceSsid.1 = simple-bsteer
WESTERMO-SW6-MIB::cfgWlanCsteerEnabled.0 = 1
WESTERMO-SW6-MIB::cfgWlanCsteerMatchingWlan.0 = 0,1
```

Listing 4.50: *Band Steering on a single AP*

Note: Although theoretically each BSS can use different SSIDs and security parameters, for simplicity the examples for Client Steering always use the same SSID and security parameters for all BSSs in the relevant ESS. This is done by using the default security settings of the wireless interfaces and grouping the index of the involved wireless interfaces by means of the cfgWlanCsteerMatchingWlan parameter.

4.47.1.3 Steering Based on Signal Strength

There are various parameters to adapt the steering process to a given environment. *IbexOS* comes with default settings that should be suitable for a simple setup.

Figure 4.36 shows the different signal levels on which the Client Steering process is based. The signal strengths in the brackets are example values and can be adapted to a particular environment. As a general rule, it can be said that the weaker the signal strength of a STA, the more vigorous the AP attempts to steer the STA to another AP.

No Action If the signal strength of a STA is above the *Roaming Scan Level* (-70dBm) no action is taken on behalf of the AP. Nevertheless, a STA may request Radio Resource Management (RRM) data (e.g. neighbor reports, 802.11k) or send BSS Transition Management Queries (802.11v) in order to limit the need of active or passive scanning. Although no action based on signal strength is expected on part of the AP, it still may steer STAs away because of load-based policies, see Section 4.47.1.4.

Roaming Scan Level As soon as the signal strenght of a STA drops below the *Roaming Scan Level* (-70dBm) the AP requests a Beacon Report from the STA. The STA in turn starts scanning the environment in order to respond with a Beacon Report. At the same time the STA might find a better AP and makes a transition of its own initiative. The *Roaming Scan Level* can be configured by cfgWlanCsteerRoamScanLevel.

Roaming Trigger Level If the signal strength is lower than *Roaming Trigger Level* (-75dBm) and the signal level of a neighboring AP is stronger than the current signal plus the cfgWlanCsteerSignalLevelDiff (5dBm), the AP sends an unsoliciated BSS Transition Management Request to the STA. The *Roaming Trigger Level* can be configured by cfgWlanCsteerRoamTriggerLevel.

Remain Level The *Remain Level* (-82dBm) prevents STAs from being disassocated from the AP. A STA with a signal strength higher than this level remains connected. This is the area of resistant or legacy STAs which do not support either of the standards 802.11k or 11v. However, if they drop below this level for longer than cfgWlanCsteerRoamKickTimeout (10s) they will be disassocated, see cfgWlanCsteerRemainLevel.

Connect Level In case a STA's signal strenght is below *Connect Level* (-90dBm) it is not allowed to connect to the AP. This steering policy can be disabled by setting its value to 0, see cfgWlanCsteerConnectLevel and cfgWlanCsteerAssocSteeringEnabled for more details.

4.47.1.4 Steering Based on Channel Load

The idea of steering a STA to another AP based on the channel load is to distribute the total load evenly across all involved APs. At present, the Client Steering of *IbexOS* supports only disassociation of STAs. The disassociation request is sent with code 5. This indicates to the STA that the AP is

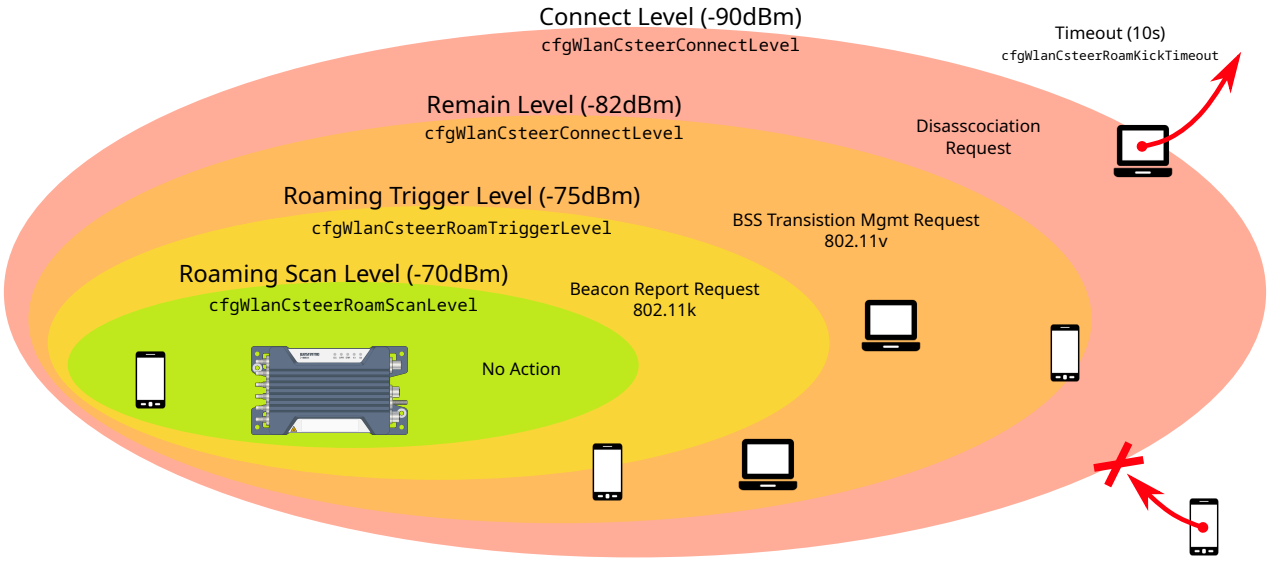


Figure 4.36: *Client Steering Roaming Levels*

currently not able to handle all STAs. It is then expected that the STA changes to another less busy AP.

But before a STA is encouraged in connecting to another AP, the following conditions must be fulfilled:

- Disassociation based on channel load must be enabled, see cfgWlanCsteerLoadBalEnabled;
- The AP must be connected with more than cfgWlanCsteerLoadBalMinNrStas; and
- The channel load of the AP must be higher than cfgWlanCsteerLoadBalMinLoad.

4.47.1.5 Steering Among Muliple Access Points

As soon as multiple APs are part of an ESS, the network topology requires that all APs are interconnected within the same data link layer (L2), e.g. by a common LAN switch. For each AP, the relevant network interface must be specified by the cfgWlanCsteerInterfaces parameter. The Client Steering process of *IbexOS* is then able to find other BSSs with enabled Client Steering and list them as remote instances.

Listing 4.51 shows how an AP with a single BSS can be integrated to a Client Steering ESS.

```
WESTERMO-SW6-MIB::cfgNetWlanEnabled.0 = 1
WESTERMO-SW6-MIB::cfgWlanIfaceSsid.0 = csteer
WESTERMO-SW6-MIB::cfgWlanCsteerEnabled.0 = 1
WESTERMO-SW6-MIB::cfgWlanCsteerMatchingWlan.0 = 0
```

```
WESTERMO-SW6-MIB::cfgWlanCsteerInterfaces.0 = br0.vlan0
```

Listing 4.51: *Client Steering Configuration*

4.47.1.6 Split APs Into Steering Zones

For largescale installations where Client Steering is desired, it makes sense to subdivide a large ESS into smaller steering zones. Since each AP collects data from STAs connected to other APs of the same data link layer (L2). It seems obvious that a large number of APs lead to higher memory usage, larger loads for the individual APs and higher traffic in the backbone network. Furthermore, it is pointless to collect data from a remote AP where the probability is close to zero that a STA could be steered there.

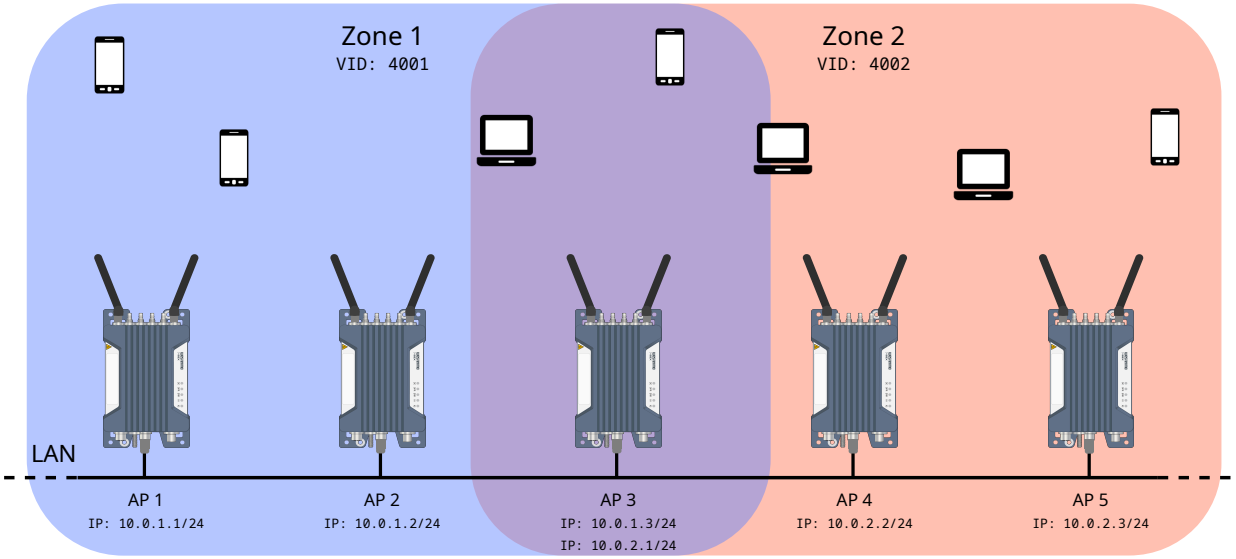


Figure 4.37: *Spliting into Steering Zones*

Different steering zones can be realised by using VLANs, as shown in Figure 4.37. How the APs are distributed amongst the zones depends very much on the installation and its environment. However, to ensure that a STA can be steered from one zone to another, it necessary to assign at least one AP to both zones, so that the zones overlap.

```
WESTERMO-SW6-MIB::cfgNetWlanEnabled.0 = 1
WESTERMO-SW6-MIB::cfgWlanIfaceSsid.0 = csteer
WESTERMO-SW6-MIB::cfgNetVlanEnabled.1 = 1
WESTERMO-SW6-MIB::cfgNetVlanVid.1 = 4001
WESTERMO-SW6-MIB::cfgNetIpEnabled.3 = 1
WESTERMO-SW6-MIB::cfgNetIpAddr.3 = 10.0.1.3/24
WESTERMO-SW6-MIB::cfgNetIpInterface.3 = br0.vlan4001
WESTERMO-SW6-MIB::cfgNetVlanEnabled.2 = 1
WESTERMO-SW6-MIB::cfgNetVlanVid.2= 4002
```

```
WESTERMO-SW6-MIB::cfgNetIpEnabled.4 = 1
WESTERMO-SW6-MIB::cfgNetIpAddr.4 = 10.0.2.1/24
WESTERMO-SW6-MIB::cfgNetIpInterface.4 = br0.vlan4002
WESTERMO-SW6-MIB::cfgWlanCsteerEnabled.0 = 1
WESTERMO-SW6-MIB::cfgWlanCsteerMatchingWlan.0 = 0
WESTERMO-SW6-MIB::cfgWlanCsteerInterfaces.0 = br0.vlan4001, br0.vlan4002
```

Listing 4.52: *Configuration of AP3 in Zone 1 and 2*

In the example in Figure 4.37, AP3 takes on the role of the intermediate AP between the two zones. Note that cfgWlanCsteerInterfaces defines both VLAN interfaces `br0.vlan4001` as well as `br0.vlan4002` to search for other Client Steering instances, see Listing 4.52. This allows AP3 to take over STAs from zone 1 and steer them to zone 2 and vice versa.

4.47.2 Airtime Fairness

Airtime Fairness (ATF) provides intelligent allocation of available airtime among connected stations. The following criteria are taken into account for the scheduling of airtime:

Airtime allocation based on SSID The total airtime is distributed to one or more SSIDs according to defined percentages.

Equal airtime allocation All stations connected to the same SSID get equal airtime allocation.

Unused airtime redistribution Depending on the scheduling algorithm, unused airtime is made available to stations that have already used up their quota.

4.47.2.1 Airtime Scheduling Algorithms

The ATF feature supports two mutually exclusive scheduling algorithms namely a strict and a fair one. The strict scheduling algorithm follows the configured scheduling rules rigorously and does not utilise any unused airtime. The fair algorithm on the other hand guarantees the configured airtime in congested environments and tries to redistribute unused airtime, see also cfgWlanDevAtfSchedulingAlgorithm.

4.47.2.2 Configuration Based on Airtime per SSID

Listing 4.53 shows how airtime can be fairly distributed among two SSIDs. The intention is to have a `private` and a `public` SSID, whereby 80% and 20% of airtime is allocated, respectively. All SSIDs,

which share airtime, must be part of the same physical device. The cfgWlanDevAtfSchedulingAlgorithm is set to `fair(0)` to achieve a fair distribution of airtime. This setting is only available per physical device.

```
WESTERMO-SW6-MIB::cfgWlanDevAtfSchedulingAlgorithm.0 = 1
WESTERMO-SW6-MIB::cfgWlanIfaceSsid.0 = private
WESTERMO-SW6-MIB::cfgWlanIfaceAtfSsidEnabled.0 = 1
WESTERMO-SW6-MIB::cfgWlanIfaceAtfSsidAirtime.0 = 80
WESTERMO-SW6-MIB::cfgWlanIfaceSsid.0 = public
WESTERMO-SW6-MIB::cfgWlanIfaceAtfSsidEnabled.1 = 1
WESTERMO-SW6-MIB::cfgWlanIfaceAtfSsidAirtime.1 = 20
```

Listing 4.53: *Airtime based on SSID*

The percentage of airtime for the respective SSIDs can be set by cfgWlanIfaceAtfSsidAirtime. This percentage is only taken into account if the corresponding cfgWlanIfaceAtfSsidEnabled parameter is set to `enabled(1)`. Otherwise, no airtime is reserved. The total of the allocated airtime shares must not exceed 100%, as otherwise the ATF configuration will not be accepted.

4.47.3 Passpoint and OpenRoaming

Starting with Wi-Fi 6 products, Passpoint is supported for public Wi-Fi network applications. For a specific Passpoint configuration, please contact support.

OpenRoaming is a Passpoint solution provided by the Wireless Broadband Alliance (WBA, please visit `https://wballiance.com/openroaming/`). It enables secure and scalable access to public Wi-Fi networks. The following configuration example gives access to the WBA OpenRoaming Identity Provider (IDP).

```
# WLAN configuration with WPA3 enterprise on 2.4/5 GHz
WESTERMO-SW6-MIB::cfgNetWlanEnabled.0 = 1
WESTERMO-SW6-MIB::cfgWlanIfaceDevice.0 = 0
WESTERMO-SW6-MIB::cfgWlanIfaceMode.0 = 0
WESTERMO-SW6-MIB::cfgWlanIfaceSsid.0 = Rmodem-OR
WESTERMO-SW6-MIB::cfgWlanIfaceEncryption.0 = 10
WESTERMO-SW6-MIB::cfgNetWlanEnabled.1 = 1
WESTERMO-SW6-MIB::cfgWlanIfaceDevice.0 = 1
WESTERMO-SW6-MIB::cfgWlanIfaceMode.1 = 0
WESTERMO-SW6-MIB::cfgWlanIfaceSsid.1 = Rmodem-OR
WESTERMO-SW6-MIB::cfgWlanIfaceEncryption.1 = 10
WESTERMO-SW6-MIB::cfgRouteDefGateway.0 = 192.168.1.2
# Passpoint configuration for SSID 'Rmodem-OR'
WESTERMO-SW6-MIB::cfgWlanPasspointEnabled.0 = 1
WESTERMO-SW6-MIB::cfgWlanPasspointSsid.0 = Rmodem-OR
WESTERMO-SW6-MIB::cfgWlanPasspointOperator.0 = eng:WestermoHotspot
WESTERMO-SW6-MIB::cfgWlanPasspointVenueName.0 = eng:WestermoVenue
WESTERMO-SW6-MIB::cfgWlanPasspointVenueGroup.0 = 0
WESTERMO-SW6-MIB::cfgWlanPasspointVenueType.0 = 0
```

```
WESTERMO-SW6-MIB::cfgWlanPasspointHessid.0 = 00:00:00:00:00:00
WESTERMO-SW6-MIB::cfgWlanPasspointNetworkType.0 = 3
WESTERMO-SW6-MIB::cfgWlanPasspointNetworkAuthType.0 = 00
WESTERMO-SW6-MIB::cfgWlanPasspointDomain.0 = westermo.com
WESTERMO-SW6-MIB::cfgWlanPasspointRoamingConsortium.0 = 004096, 5A03BA0000
WESTERMO-SW6-MIB::cfgWlanPasspointCellNet.0 = -1
WESTERMO-SW6-MIB::cfgWlanPasspointNaiRealm.0 = -1
WESTERMO-SW6-MIB::cfgWlanPasspointOperatorNamespace.0 = -1
# 802.1X configuration
WESTERMO-SW6-MIB::cfgWlan802dot1xAuthSrvEnabled.0 = 1
WESTERMO-SW6-MIB::cfgWlan802dot1xAuthSrvIpAddr.0 = 1.2.3.4
WESTERMO-SW6-MIB::cfgWlan802dot1xAuthSrvPort.0 = 1812
WESTERMO-SW6-MIB::cfgWlan802dot1xAuthSrvSharedSecret.0 = wireless4more
WESTERMO-SW6-MIB::cfgWlan802dot1xAcctSrvEnabled.0 = 1
WESTERMO-SW6-MIB::cfgWlan802dot1xAcctSrvIpAddr.0 = 1.2.3.4
WESTERMO-SW6-MIB::cfgWlan802dot1xAcctSrvPort.0 = 1813
WESTERMO-SW6-MIB::cfgWlan802dot1xAcctSrvSharedSecret.0 = wireless4more
```

Listing 4.54: *OpenRoaming with WBA IDP*

OpenRoaming network access is provided to all WLAN interfaces sharing the same SSID `Rmodem-OR`, referring in cfgWlanPasspointSsid. The encryption must either be set to `eap(6)` (WPA2 Enterprise) or `eap2(10)` (WPA3 Enterprise). Most Passpoint configuration parameters are used to describe or to limit the access to the network access node and can be left to defaults. The Roaming Consortium ID's (RCOIs) must be set according to the IDP requirements.

IDP's must be operated with a RadSec upstream server. The RADIUS authentication server cfgWlan802dot1xAuthSrvIpAddr, same for accounting, must either point to a local or remote RadSec proxy (RadSec proxy). Also a DHCP server in the local network is needed to provide the right network access parameters to the authenticated clients.

4.48 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.

4.49 Dynamic Frequency Selection (DFS)

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

- Wireless Standlone

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.

4.49.1 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 frequence the AP jumped to is not a DFS frequency, the AP will stay there until it is manually reconfigured.

4.49.2 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, RT-280 products because NWM requires two radios.

Radio0 is the so called communication interface providing the access point functionality. Radio1 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 4.50

Radio1 is used for CAC and Off-Channel-CAC. Therefore you need to assert that you have an antenna connected at antenna port of radio1.

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 radio1 of the RT-370, RT-280 products. 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.

4.49.2.1 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 4.38.

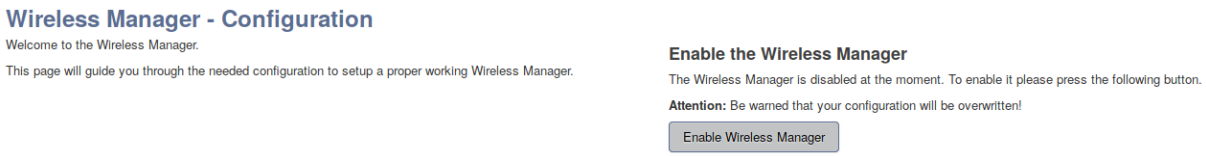


Figure 4.38: *Enable page*

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

Setting	Value
cfgNwmEnabled.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 4.6: *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 4.39 should be shown.

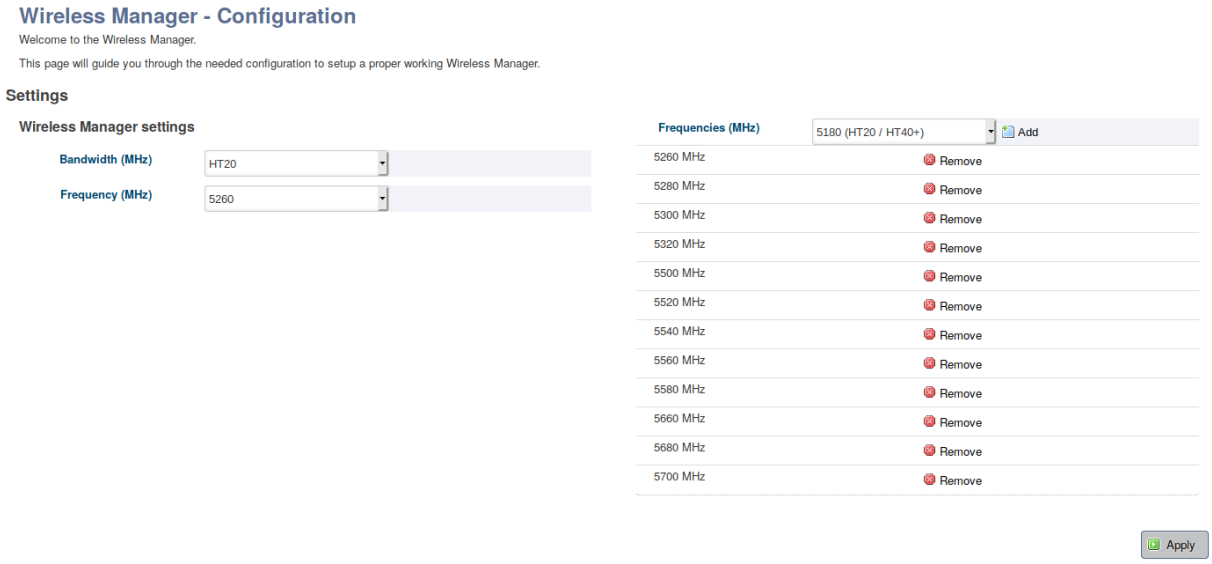


Figure 4.39: *Configuration page*

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

Frequency This define the prefered 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 prefered 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 4.40 shows all important status informations about the NWM. The page will be automatically refreshed so you will always see the actual status.

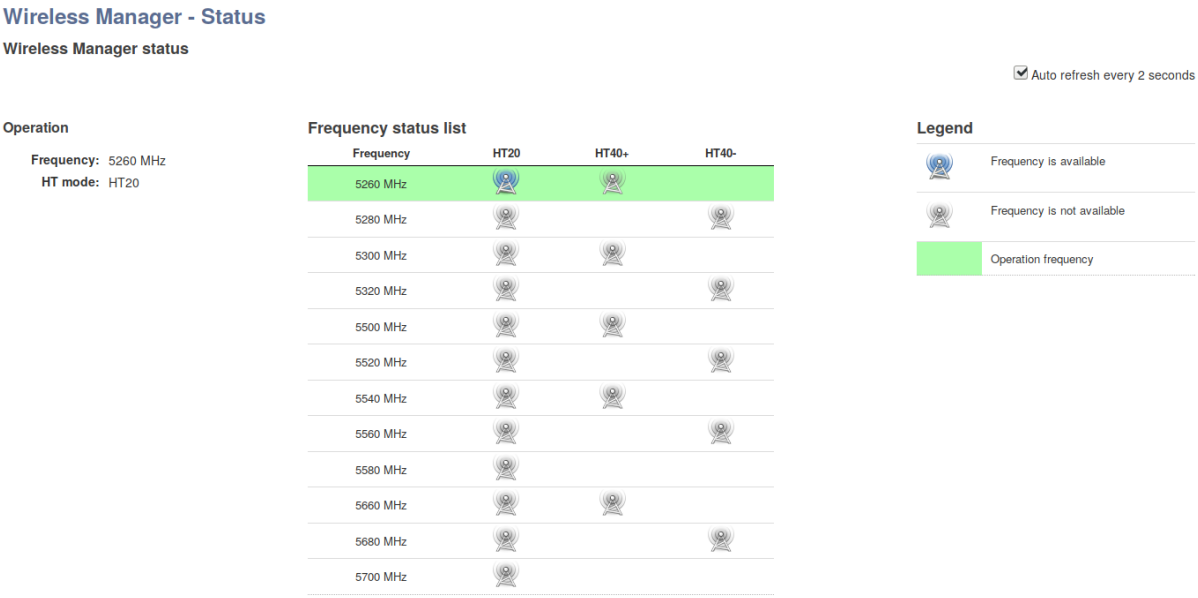


Figure 4.40: *Status page*

4.49.3 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

4.49.3.1 Functional Principle

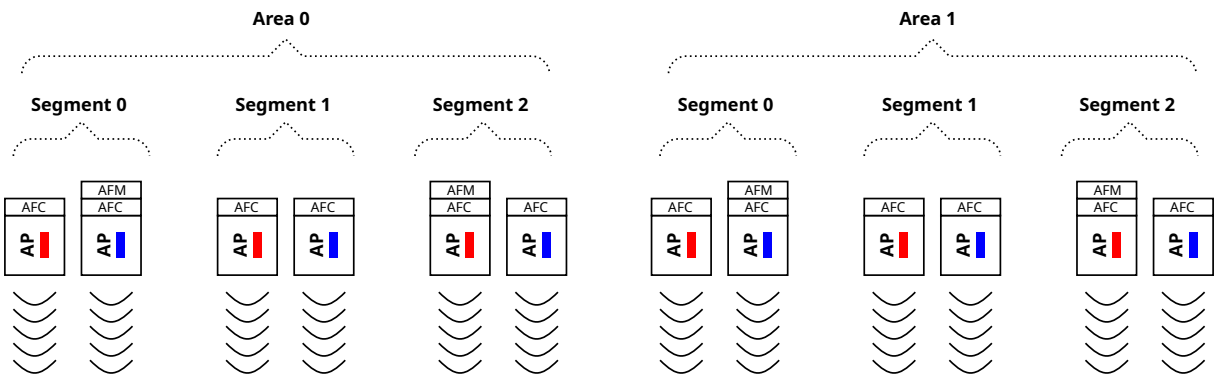


Figure 4.41: *Area Frequency Management Principle*

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. The ports used are UDP 4711 and UDP 4713. 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 4.42. In this example the DNL for AP5 contains information (i.e. frequency, BSSID, etc.) from AP3 to AP7.

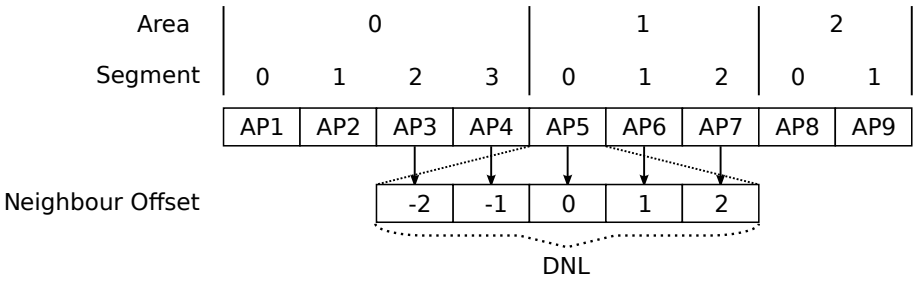


Figure 4.42: *Area, Segments in relation to Neighbour Offset and DNL*

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 required, 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.

4.49.3.2 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 cfgWlanIfaceMode.
2. Configure a common set of wireless parameters: cfgWlanIfaceSsid, cfgWlanIfaceEncryption, etc.
3. Enable the mandatory services: AFC cfgAfcEnabled.
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.

4.49.3.3 Example

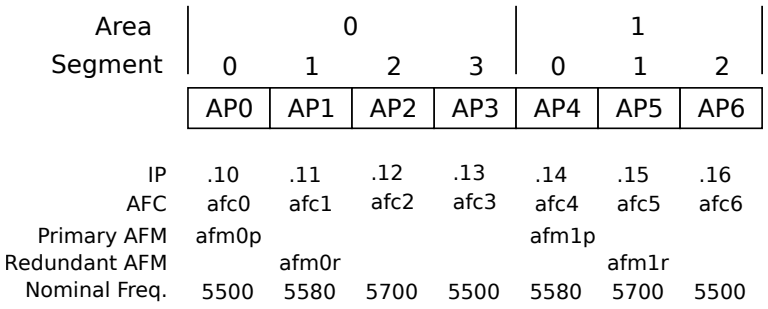


Figure 4.43: *Example AFM configuration*

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						
cfgChanCleanUsableFrequencyList	0						

4.50 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.

For RT-370, RT-280 products, the IDF functionality supports JSON reports which can be sent to a configured URL (cfgHttpRprtServerUrl) at a configured interval (cfgIdfInterval).

IDF is only available on RT-370, RT-280 products because it uses radio1 (i.e. wlan1) interface.

Please note that the radio1 is also used for DFS as described in section 4.49.2

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.

4.51 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

4.51.1 NWM and ChannelManager Report

For more information about the current status of the NWM and the Channel Manager (see section 4.49.2) 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
    }
  }
}
```

4.51.2 IDF Report

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

Please refer to section 4.50 for more information on how to configure the IDF

Elements of the IDF Report are:

- data.wlan_stats will only be present if cfgIdfScanWorkAction = 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 cfgIdfScanWorkAction = radar(3)
- data.spectral_reports will only be present if cfgIdfScanWorkAction = 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]
      ]
    }
  ]
 }
}
```

4.52 Firewall

4.52.1 L3 Network Address Translation (NAT)

The *IbexOS* 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.

4.52.1.1 Port forward (DNAT)

Port forwarding can be used to forward network traffic to another destination. This is also know 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.

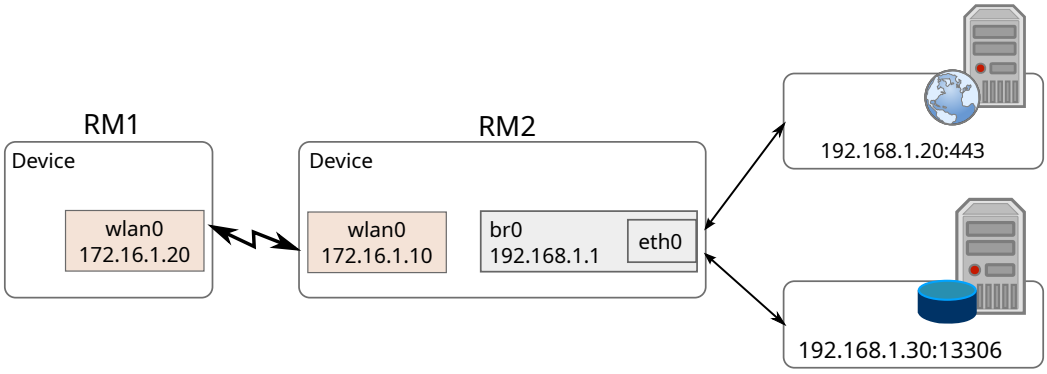


Figure 4.44: *Port forward to servers behind an device*

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.

```
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
```

Listing 4.55: *Port forward*

4.52.1.2 Outbound NAT (SNAT)

With outbound NAT the device can control how traffic leaving the device will be translated. This is also known as emphSource Network Address Translation (SNAT) or MASQUERADE and is used in most home routers to rewrite the source address to the address of the WAN interface of the router.

SNAT can be done by simple masquerade, meaning to rewrite to the address of the network interface or by manually defining the source address/port if there are multiple addresses available.

As for the port forward a simple example to illustrate the functionality is shown in Figure 4.45. The goal is to connect from a Laptop, through RM1 to the web interface of RM2, where RM2 does not have a default gateway set. For this example the Laptop uses RM1 as default gateway and the wlan0 interface of RM1 has a dynamic address.

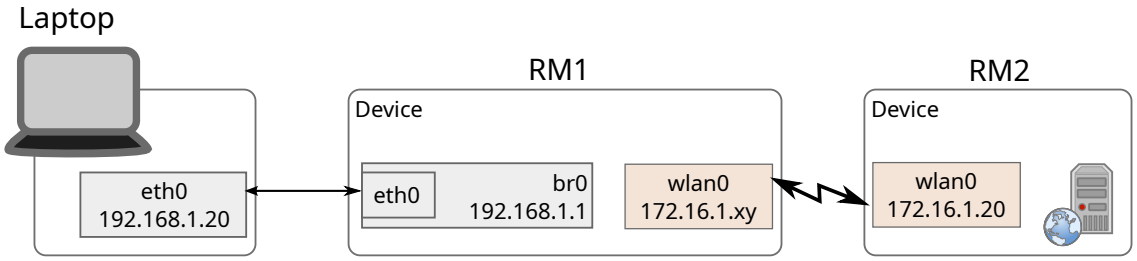


Figure 4.45: *Example for outbound NAT on a device*

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.

```
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
```

Listing 4.56: *Outbound NAT*

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.

4.52.1.3 Netmap NAT (1:1 NAT)

Contrary to SNAT and DNAT where the source or destination is rewritten to a single address, 1:1 NAT allows to rewrite a whole network. This is useful when remapping networks to overcome duplicate networks that have to communicate with each other.

Illustrated in Figure 4.46 is the situation where 3 devices act as router for their network. All networks have the same subnet, thus they can not directly communicate with each other. RM1, RM2 and RM3 each have a netmap NAT rule in place that rewrites the 10.x.0.0/24 subnet to the respective local 192.168.1.0/24 subnet. 172.16.1.0/24 acts as transfer network between the 3 routers.

Devices in the RM2 network can access devices behind RM1 and RM3 by accessing their virtual address in the 10.1.0.0/24 respective 10.3.0.0/24 range. Traffic arriving at a device behind RM1/RM3 appears as if it originates from the 10.2.0.0/24 range.

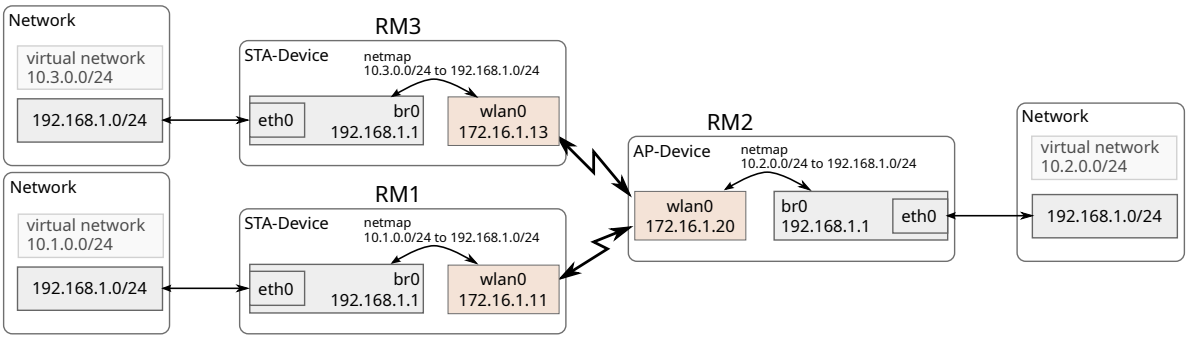


Figure 4.46: *Example for netmap NAT on a device*

```
WESTERMO-SW6-FIREWALL-MIB::cfgFwEnabled.0 = 1
WESTERMO-SW6-FIREWALL-MIB::cfgFwNatOneToOneEnabled.0 = 1
WESTERMO-SW6-FIREWALL-MIB::cfgFwNatOneToOneType.0 = 0
WESTERMO-SW6-FIREWALL-MIB::cfgFwNatOneToOneSourceNet.0 = 0.0.0.0/0
WESTERMO-SW6-FIREWALL-MIB::cfgFwNatOneToOneDestinationNet.0 = 10.3.0.0/24
WESTERMO-SW6-FIREWALL-MIB::cfgFwNatOneToOneRewriteNet.0 = 192.168.1.0/24
```

Listing 4.57: *Netmap NAT on RM3*

4.52.2 L3 Filter

The iptables filter allows to create stateful firewall rules based on:

- Interface (cfgFwFltRInputInterface and cfgFwFltROutputInterface)
- IP-Address (cfgFwFltRSourceAddress and cfgFwFltRDestinationAddress)
- IP-Protocol (cfgFwFltRProtocol)
- TCP/UDP Ports (cfgFwFltRSourcePortStart, cfgFwFltRSourcePortEnd, cfgFwFltRDestination-PortStart and cfgFwFltRDestinationPortEnd)
- ICMP Code/Type (cfgFwFltRIcmpType and cfgFwFltRIcmpCode)

Further it allows to limit the rate of connections (cfgFwFltRLimit and cfgFwFltRLimitBurst) and provides logging when rules are executed (cfgFwFltRLog).

There are 3 chains on which rules can be created selected with cfgFwFltRChain:

- 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 the default policy of the respective chains. In other words: what happens with frames when no rule matches.

Note: If the default policy of any chain is set to *DROP*, be aware that certain system services may also be blocked. For example, if cfgFwFltDefaultPolicyOutput is set to *DROP*, then communication to the cellular module for DHCP requests/responses will no longer work and must be explicitly allowed for the *wwanX* interface in the filter rules.

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

cfgFwFltInputSynLimit and cfgFwFltInputSynLimitBurst may be used to parameterise a SYN-flood filter. It is used to limit the number of TCP-SYN frames on the Input chain to prevent resource exhaustion of the device.

The options cfgFwFltLogDefaultAccept and cfgFwFltLogDefaultDrop define if frames processed by the default rule are logged. Each connection which is accepted and each frame that is dropped generate a Syslog message.

```
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
```

Listing 4.58: *Input filtering allow SNMP from 192.168.1.0/24 and block everywhere else*

```
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
```

Listing 4.59: *Forward filtering Allow HTTP from everywhere and block everything else*

```
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
```

Listing 4.60: *Output filtering Block access to the tftp server at 192.168.1.3*

4.52.3 L3 Mangle

Iptables mangle rules allow to modify frames that are received, transmitted or forwarded. A rule may match on:

- Interface (cfgFwMnglInputInterface and cfgFwMnglOutputInterface)
- IP-Address (cfgFwMnglSourceAddress and cfgFwMnglDestinationAddress)
- IP-Protocol (cfgFwMnglProtocol)
- TCP/UDP Port (cfgFwMnglSourcePortStart, cfgFwMnglSourcePortEnd, cfgFwMnglDestination-PortStart and cfgFwMnglDestinationPortEnd)
- DSCP (cfgFwMnglDscp)

cfgFwMnglAction specifies the action to be performed:

- setmss: Set the MSS field of TCP SYN frames to the value specified in cfgFwMnglValue.
- ttlset: Set the TTL of frames to the value specified in cfgFwMnglValue. The TTL has a range of 0 to 255. When a value of 0 is set, this frame can not be routed.
- setdscp: Set the DSCP of frames to the value specified in cfgFwMnglValue. The DSCP in hexadecimal form 00-FF without 0x. Must be a multiple of 4 (the lowest 2 bits 0).
- setmark: Set the MARK of frames to the value specified in cfgFwMnglValue. The MARK has a range of 0 to 4294967295.

cfgFwMnglChain specifies on which chain the frame manipulation is performed.

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

WESTERMO-SW6-FIREWALL-MIB::cfgFwMnglEnabled.0 = 1
WESTERMO-SW6-FIREWALL-MIB::cfgFwMnglChain.0 = 2
WESTERMO-SW6-FIREWALL-MIB::cfgFwMnglAction.0 = 1
WESTERMO-SW6-FIREWALL-MIB::cfgFwMnglValue.0 = 1400
WESTERMO-SW6-FIREWALL-MIB::cfgFwMnglInputInterface.0 = eth0
WESTERMO-SW6-FIREWALL-MIB::cfgFwMnglOutputInterface.0 = eth1
WESTERMO-SW6-FIREWALL-MIB::cfgFwMnglSourceAddress.0 = 0.0.0.0/0
WESTERMO-SW6-FIREWALL-MIB::cfgFwMnglDestinationAddress.0 = 0.0.0.0/0
```

Listing 4.61: *Set the MSS of frames forwarded from eth0 to eth1 to 1400*

4.52.4 L2 Filter

The L2 Filter can be used to filter frames on bridges.

To make use of the L2 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 256 rules may be defined in the cfgFwL2IpFilterTable.

The filter offers two modes of operation, selectable by cfgFwL2IpFilterMode:

- IP mode (0)
- Full mode (1)

When the filter is set to *IP mode* then the filter will only apply on IP frames and will allow any non-IP frames.

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.

The priority of a rule is used when multiple rules would match a frame. The rule with the highest priority will be executed for the matched frame. Multiple matching rules may not have the same priority since such a configuration results in undefined behaviour.

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.

```
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFilterEnabled.0 = 1
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFilterMode.0 = 0
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
```

Listing 4.62: *L2 IP Filter*

When the mode is set to *Full mode* then the filter will process any frame.

In addition to the options of the IP mode, the Full mode offers the ability to filter on the 'IP protocol', 'Source Port' (UDP/TCP), 'Destination Port' (UDP/TCP), Source MAC, Destination MAC, Ethertype, VLAN, and Input Interface. Please refer to cfgFwL2IpFilterTable for more information.

In a setup with multiple APs it may be desirable to prevent traffic between different APs (Wifi Hotspot).

cfgWlanIfaceApIsolate may be used to prevent traffic between clients on the same wireless interface.

cfgNetWlanProtected may be used to prevent traffic to flow between wlan0 and wlan1.

The following example of a ruleset may be used to prevent traffic between clients on different APs.

- Allow any frame by default
- br0: drop any unmatched frame on br0
- br0: allow frames from and to the MAC address 00:14:5a:02:11:95 of a gateway with the IP 192.168.3.1
- br0: allow ARP lookups for the gateway at 192.168.3.1
- br0: allow DHCP requests from wlan0 and wlan1
- br0: allow frames from the management interface on br0.vlan0
- br0: allow ARP lookups for the management interface at 192.168.3.22
- br0: allow IP frames to the management interface at 192.168.3.22

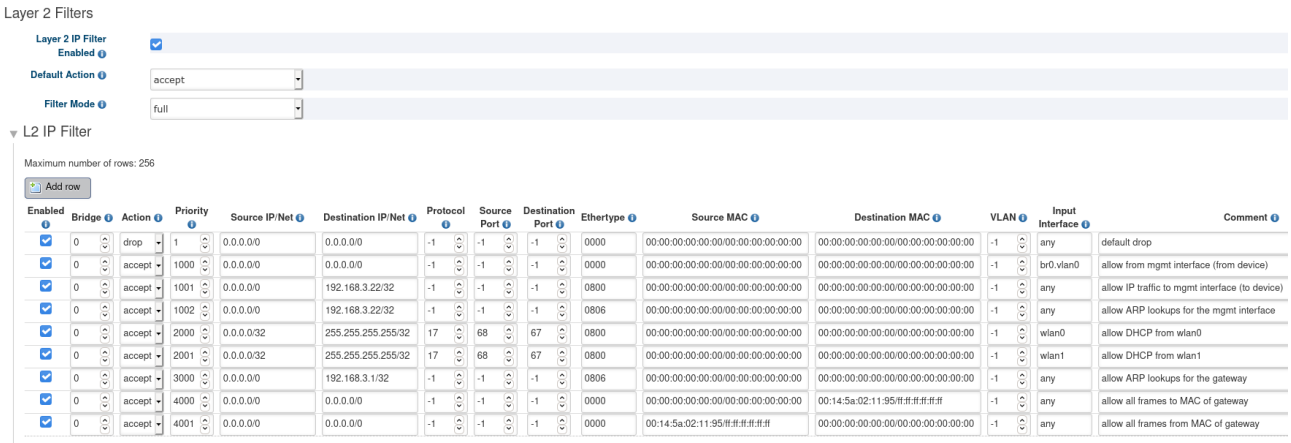


Figure 4.47: *Full L2 filter to separate clients on different APs*

```
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFilterEnabled.0 = 1
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFilterDefaultAction.0 = 0
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFilterMode.0 = 1

WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrEnabled.0 = 1
```

```
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrBridge.0 = 0
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrAction.0 = 1
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrPriority.0 = 1
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrSource.0 = 0.0.0.0/0
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrDestination.0 = 0.0.0.0/0
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrProtocol.0 = -1
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrSourcePort.0 = -1
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrDestinationPort.0 = -1
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrSourceMac.0 = 00:00:00:00:00:00/00:00:00:00:00:00
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrDestinationMac.0 = 00:00:00:00:00:00/00:00:00:00:00:00
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrEthertype.0 = 0000
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrVlan.0 = -1
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrInputInterface.0 = any
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrComment.0 = default drop

WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrEnabled.1 = 1
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrBridge.1 = 0
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrAction.1 = 0
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrPriority.1 = 1000
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrSource.1 = 0.0.0.0/0
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrDestination.1 = 0.0.0.0/0
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrProtocol.1 = -1
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrSourcePort.1 = -1
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrDestinationPort.1 = -1
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrSourceMac.1 = 00:00:00:00:00:00/00:00:00:00:00:00
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrDestinationMac.1 = 00:00:00:00:00:00/00:00:00:00:00:00
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrEthertype.1 = 0000
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrVlan.1 = -1
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrInputInterface.1 = br0.vlan0
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrComment.1 = allow anything from the mgmt interface (
    from device)

WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrEnabled.2 = 1
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrBridge.2 = 0
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrAction.2 = 0
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrPriority.2 = 1001
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrSource.2 = 0.0.0.0/0
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrDestination.2 = 192.168.3.22/32
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrProtocol.2 = -1
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrSourcePort.2 = -1
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrDestinationPort.2 = -1
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrSourceMac.2 = 00:00:00:00:00:00/00:00:00:00:00:00
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrDestinationMac.2 = 00:00:00:00:00:00/00:00:00:00:00:00
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrEthertype.2 = 0800
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrVlan.2 = -1
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrInputInterface.2 = any
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrComment.2 = allow IP traffic to the mgmt interface (to
    device)

WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrEnabled.3 = 1
```

```
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrBridge.3 = 0
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrAction.3 = 0
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrPriority.3 = 1002
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrSource.3 = 0.0.0.0/0
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrDestination.3 = 192.168.3.22/32
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrProtocol.3 = -1
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrSourcePort.3 = -1
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrDestinationPort.3 = -1
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrSourceMac.3 = 00:00:00:00:00:00/00:00:00:00:00:00
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrDestinationMac.3 = 00:00:00:00:00:00/00:00:00:00:00:00
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrEthertype.3 = 0806
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrVlan.3 = -1
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrInputInterface.3 = any
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrComment.3 = allow ARP lookups for the mgmt interface

WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrEnabled.4 = 1
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrBridge.4 = 0
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrAction.4 = 0
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrPriority.4 = 2000
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrSource.4 = 0.0.0.0/32
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrDestination.4 = 255.255.255.255/32
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrProtocol.4 = 17
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrSourcePort.4 = 68
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrDestinationPort.4 = 67
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrSourceMac.4 = 00:00:00:00:00:00/00:00:00:00:00:00
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrDestinationMac.4 = 00:00:00:00:00:00/00:00:00:00:00:00
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrEthertype.4 = 0800
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrVlan.4 = -1
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrInputInterface.4 = wlan0
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrComment.4 = allow DHCP from wlan0

WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrEnabled.5 = 1
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrBridge.5 = 0
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrAction.5 = 0
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrPriority.5 = 2001
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrSource.5 = 0.0.0.0/32
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrDestination.5 = 255.255.255.255/32
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrProtocol.5 = 17
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrSourcePort.5 = 68
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrDestinationPort.5 = 67
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrSourceMac.5 = 00:00:00:00:00:00/00:00:00:00:00:00
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrDestinationMac.5 = 00:00:00:00:00:00/00:00:00:00:00:00
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrEthertype.5 = 0800
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrVlan.5 = -1
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrInputInterface.5 = wlan1
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrComment.5 = allow DHCP from wlan1

WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrEnabled.6 = 1
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrBridge.6 = 0
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrAction.6 = 0
```

```
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrPriority.6 = 3000
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrSource.6 = 0.0.0.0/0
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrDestination.6 = 10.0.11.1/32
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrProtocol.6 = -1
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrSourcePort.6 = -1
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrDestinationPort.6 = -1
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrSourceMac.6 = 00:00:00:00:00:00/00:00:00:00:00:00
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrDestinationMac.6 = 00:00:00:00:00:00/00:00:00:00:00:00
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrEthertype.6 = 0806
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrVlan.6 = -1
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrInputInterface.6 = any
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrComment.6 = allow ARP lookups for the gateway

WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrEnabled.7 = 1
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrBridge.7 = 0
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrAction.7 = 0
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrPriority.7 = 4000
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrSource.7 = 0.0.0.0/0
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrDestination.7 = 0.0.0.0/0
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrProtocol.7 = -1
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrSourcePort.7 = -1
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrDestinationPort.7 = -1
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrSourceMac.7 = 00:00:00:00:00:00/00:00:00:00:00:00
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrDestinationMac.7 = 00:14:5a:02:11:95/ff:ff:ff:ff:ff:ff
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrEthertype.7 = 0000
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrVlan.7 = -1
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrInputInterface.7 = any
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrComment.7 = allow all frames to MAC of gateway

WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrEnabled.8 = 1
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrBridge.8 = 0
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrAction.8 = 0
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrPriority.8 = 4001
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrSource.8 = 0.0.0.0/0
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrDestination.8 = 0.0.0.0/0
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrProtocol.8 = -1
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrSourcePort.8 = -1
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrDestinationPort.8 = -1
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrSourceMac.8 = 00:14:5a:02:11:95/ff:ff:ff:ff:ff:ff
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrDestinationMac.8 = 00:00:00:00:00:00/00:00:00:00:00:00
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrEthertype.8 = 0000
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrVlan.8 = -1
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrInputInterface.8 = any
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2IpFltrComment.8 = allow all frames from MAC of gateway
```

Listing 4.63: *L2 Full Filter*

4.52.5 L2 Mangle

The L2 Mangle function can be used to moddify frames on bridges.

To make use of L2 mangling, it must be enabled with cfgFwL2MangleEnabled.

Up to 256 rules may be defined in the cfgFwL2MangleTable.

For each mangle rule match there are all the fields also available in the full l2 filter. These are the ‘Bridge‘, ‘IP Source Net‘, ‘IP Destination Net‘, ‘IP Protocol‘, ‘Source Port‘ (UDP/TCP), ‘Destination Port‘ (UDP/TCP), ‘Source MAC‘, ‘Destination MAC‘, ‘Ethertype‘, ‘VLAN‘, and ‘Input Interface‘. Please refer to cfgFwL2MangleTable for more information.

In addition for each rule there are the fields ‘Priority‘, ‘Action‘ and ‘Value‘.

The priority of a rule is used when multiple rules would match a frame. The rule with the highest priority will be executed for the matched frame. Multiple matching rules may not have the same priority since such a configuration results in undefined behaviour.

The action defines which field of the frame is moddified with the content defined in the value parameter.

The following example of an L2 Mangle rule matches IP frames with any source IP and 192.168.3.130 as destination IP on bridge 0. As action the matched frames have their DSCP bits set to 0xA0. All other frames continue normally.

```
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2MangleEnabled.0 = 1

WESTERMO-SW6-FIREWALL-MIB::cfgFwL2MnglEnabled.0 = 1
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2MnglBridge.0 = 0
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2MnglPriority.0 = 1
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2MnglSource.0 = 0.0.0.0/0
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2MnglDestination.0 = 192.168.3.130/32
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2MnglProtocol.0 = -1
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2MnglSourcePort.0 = -1
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2MnglDestinationPort.0 = -1
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2MnglSourceMac.0 = 00:00:00:00:00:00/00:00:00:00:00:00
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2MnglDestinationMac.0 = 00:00:00:00:00:00/00:00:00:00:00:00
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2MnglEthertype.0 = 0800
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2MnglVlan.0 = -1
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2MnglInputInterface.0 = any
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2MnglAction.0 = 1
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2MnglValue.0 = A0
```

Listing 4.64: *L2 Mangle*

5 Troubleshooting

5.1 LED Status

The LED status pattern differs from the 802.11n and cellular to the 802.11ax and 802.11be product series. The corresponding technology assignment is listed in Supported Products.

5.1.1 Ibex 802.11n and cellular products

The DC LED is dark:

The device is not powered. If it is supplied with power via PoE (Power over Ethernet), please ensure that the Ethernet connector that supports PoE is plugged in properly.

The OPR LED is solid orange:

The device is configured as an STA but it is not connected to an Access Point. If you expect a 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.

5.1.2 Ibex 802.11ax and 802.11be products

The DC LED is dark:

The device is not powered via DC input. If it is supplied with power via PoE (Power over Ethernet), the ON LED is indicating if the device is powered or not.

The DC LED is red:

The devices is only operated with a single power supply pair. The DC LED lights up green when both pairs are supplying power.

The ON 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 ON LED is solid red:

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

The ON LED is dark:

Critical hardware failure. The device must be repaired. If the device is supplied with power via PoE (Power over Ethernet), ensure that the PoE-capable connection is plugged in correctly.

5.1.3 Unspecified status

- 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.

5.2 Cellular Connection

Ensure that there is mobile network coverage and that the SIM card is working and active.

5.2.1 Peripherals

Antennas

Check that all antenna ports are connected and engaged. Ensure that the connected antennas support the frequency bands of the mobile operator's network.

SIM Card

Check that a SIM card is inserted and the SIM card drawer is engaged. If only one SIM card is used, it is recommended to disable the unused SIM card slot in the configuration (cfgCellSimSlotEnabled).

- SIM card not detected: Reboot device and try again
- SIM card state **Not Ready**: Increase the SIM card unlock timeout (cfgCellSimSlotUnlockTimeout).
- SIM card state **Sim Pin**: Ensure that the right SIM Pin is defined and the SIM Pin check cfgCellSimSlotPinEnabled is enabled.
- SIM card state **Sim Puk**: The SIM card could not be unlocked. The SIM PUK is required to unlock the SIM card. Refer to the CLI Commands `sim` to unlock or lock the SIM card.

5.2.2 Status

Roaming

The Roaming status indication is static according to the configuration parameter cfgCellDefaultBearerRoaming, it only tells if the SIM card is allowed to roam. If the device is actually roaming into a foreign network can be indicated by the **Service Name** and **Full Network Name**.

- Both names are equal or similar (case sensitivity might be different): Not roaming
- The names are different: Roaming from **Full Network Name** to **Service Name** (SIM card issuer).

Cellular Status

The various states are described in swCellServingCellState.

- **Searching**: Check the signal levels, if the antenna is properly connected and is the cell coverage good enough.

- **Not Registered**: Check the APN settings of the current SIM card. Check if the SIM card is not blocked by the mobile provider.

Modem Status

The various states are described in swCellConnectionStatus. Usually the states are not explicitly reported unless the procedure hangs in a particular state, which could be the reason for this:

- **Disabled**: Enable the Cellular interface cfgNetWwanEnabled in the Network settings.
- **Connecting**: The connection cannot be established, check the signal values and **Modem Status**.
- **Disconnected**: The link got lost, check the signal values and **Modem Status**.
- **Resetting**: An unexpected condition has been detected, the modem status is reset, a full reconnect is required.

Connection Message

- **Cellular interface has no SIM cards configured**: Check the SIM card settings, at least one SIM card must be active cfgCellSimSlotEnabled.
- **QMI interface is not initialized (HW problem)**: If this happens occasionally, try updating the mobile phone firmware. If it happens all the time, the device needs to be sent in for repair.

5.2.3 Signals

The data given are guide values and should not be regarded as absolute thresholds.

LTE/5G-NSA

The actual signal quality depends on various factors and is reported by different measurement methods. The **RSSI** is the received signal strength of the cellular base station, it does not include any signal quality information. It could be, that the **RSSI** signal is excellent but the link itself is poor. In this case, refer to the other signal levels **RSRQ**, **RSRP** and **SINR** which give more detailed information. The following columns are not related, each signal type is considered independently.

	RSSI [dBm]	RSRQ [dB]	RSRP [dBm]	SINR [dB]
Excellent	**> -65**	> -3	> -65	> 20
Good	**-75 to -66**	-5 to -4	-80 to -66	14 to 20
Fair	**-85 to -76**	-11 to -6	-110 to -81	0 to 13
Poor	**< -85**	< -12	< -110	< 0

5G-SA

The signal strength **RSSI** is not reported in 5G standalone mode, the signal quality can be analysed according the the following table.

	RSRQ [dB]	RSRP [dBm]	SINR [dB]
Excellent	> -3	> -65	> 20
Good	-5 to -4	-80 to -66	10 to 20
Fair	-11 to -6	-110 to -81	0 to 9
Poor	< -12	< -110	< 0

5.2.4 Network Information

When the **Modem Status** changes to `Connected`, a DHCP lease is received. Check if IP address, gateway and DNS per configured default bearer (APN) are listed.

- No DHCP lease received: Check the **Cellular Status**. If it is in the correct state `Registered`, check for error messages in the log.
- DHCP lease received but no connection
  - Cannot ping the internet (IP): Check the network settings, if the routes towards the `wwan` interfaces are setup correctly.
  - Cannot ping the internet (Domain): Check if the DNS servers are reachable. Configure a nameserver manually in the DNS/DHCP settings.
  - Cannot ping the gateway: The reported gateway is not a valid IP to ping (the received IP address is in a /32 subnet). No gateway is needed, the routing is interface based. When using a private APN, ask your mobile provider how to reach the core network.

5.2.5 Advanced Analysis Tools

Technical Support File

In most cases and for static setups it is sufficient to capture a Technical Support File to analyse the connection status or signal quality. This is a single snapshot with no history, debug logging should be considered for analysing the connection over time.

Debug Logging

Instead of retrieving multiple Technical Support Files, debug logging can be activated. Messages are reported at fixed intervals.

- cfgCellDbgSignal: Reporting cell information and signal levels, see Commissioning Log Messages 820.
- cfgCellDbgConnection: Reporting connection information, see Commissioning Log Messages 821.

Verbose Logging

If connection issues cannot be resolved, verbose logging of the link may help. Refer to the CLI Commands `qlog`. This tool records the cellular protocol and communication data and sends it to a remote host via TCP or FTP. Please contact technical support and provide the recorded data.

Note: It is recommended to avoid sending large amounts of data during recording. Also be aware that sensitive data may be recorded.

5.2.6 Performance & Connection Loss

If the overall performance is below expectations, check if the signal levels match with the quality indications. Please note that performance (troughput and latency) is dependent on the mobile operator's infrastructure and available speed may be limited.

Use debug and verbose logging to investigate random or periodic connection losses.

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	19 dBm
- EIRP (two antennas)	9 dBm	19 dBm
- EIRP (three antennas)	11 dBm	19 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 29 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	19 dBm
- EIRP (two antennas)	9 dBm	19 dBm
- EIRP (three antennas)	11 dBm	19 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	22 dBm, 29 dBm (Note 1)
- EIRP (two antennas)	9 dBm	22 dBm, 29 dBm (Note 1)
- EIRP (three antennas)	11 dBm	22 dBm, 29 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	14 dBm
- EIRP (two antennas)	9 dBm	14 dBm
- EIRP (three antennas)	11 dBm	14 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	19 dBm
- EIRP (two antennas)	9 dBm	19 dBm
- EIRP (three antennas)	11 dBm	19 dBm
Frequencies 5 GHz	5180, 5200, 5220, 5240, 5260, 5280, 5300, 5320	
HIER - Antenna Gain	0 dBi	14 dBi
- EIRP (one antenna)	6 dBm	22 dBm
- EIRP (two antennas)	9 dBm	22 dBm
- EIRP (three antennas)	11 dBm	22 dBm
- DFS (slave/master)	5260 to 5320	
Frequencies 5.8 GHz	5745, 5765, 5785, 5805, 5825	
- Antenna Gain	0 dBi	14 dBi
- EIRP (one antenna)	6 dBm	32 dBm
- EIRP (two antennas)	9 dBm	32 dBm
- EIRP (three antennas)	11 dBm	32 dBm

Note This country code is NOT available on Ibex-RT-630-5G product.

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 cfgWlanGlblCountry 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	26 dBm
- EIRP (two antennas)	9 dBm	26 dBm
- EIRP (three antennas)	11 dBm	26 dBm
Frequencies 5 GHz	5180, 5200, 5220, 5240, 5260, 5280, 5300, 5320	
- Antenna Gain	0 dBi	14 dBi
- EIRP (one antenna)	6 dBm	22 dBm
- EIRP (two antennas)	9 dBm	22 dBm
- EIRP (three antennas)	11 dBm	22 dBm
- DFS (slave/master)	5260 to 5320	
Frequencies 5.8 GHz	5745, 5765, 5785, 5805, 5825	
- Antenna Gain	0 dBi	14 dBi
- EIRP (one antenna)	6 dBm	32 dBm
- EIRP (two antennas)	9 dBm	32 dBm
- EIRP (three antennas)	11 dBm	32 dBm

Note This country code is NOT available on Ibex-RT-630-5G product.

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	35 dBm
- EIRP (two antennas)	9 dBm	35 dBm
- EIRP (three antennas)	11 dBm	35 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	29 dBm
- EIRP (two antennas)	9 dBm	29 dBm
- EIRP (three antennas)	11 dBm	29 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	35 dBm
- EIRP (two antennas)	9 dBm	35 dBm
- EIRP (three antennas)	11 dBm	35 dBm

6.2.2.5 Country code SINGAPORE

Country code 'SINGAPORE' applies for Singapore.

The country code can be set by:

- Set cfgWlanGlblCountry to SINGAPORE

	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	22 dBm
- EIRP (two antennas)	9 dBm	22 dBm
- EIRP (three antennas)	11 dBm	22 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	29 dBm
- EIRP (two antennas)	9 dBm	29 dBm
- EIRP (three antennas)	11 dBm	29 dBm
- Indoor frequencies	5180 to 5320	
- DFS	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	29 dBm
- EIRP (two antennas)	9 dBm	29 dBm
- EIRP (three antennas)	11 dBm	29 dBm

6.2.2.6 Country code MALAYSIA

Country code 'MALAYSIA' applies for Malaysia.

The country code can be set by:

- Set cfgWlanGlblCountry to MALAYSIA

	Min	Max
Frequencies 2.4 GHz	2412, 2417, 2422, 2427, 2432, 2437, 2442, 2447, 2452, 2457, 2462, 2467, 2472	
- 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
Frequencies 5 GHz	5180, 5200, 5220, 5240, 5260, 5280, 5300, 5320	
- Antenna Gain	0 dBi	15 dBi
- EIRP (one antenna)	6 dBm	22 dBm
- EIRP (two antennas)	9 dBm	22 dBm
- EIRP (three antennas)	11 dBm	22 dBm
- Indoor frequencies	5260 to 5320 (29 dBm)	
- DFS	5260 to 5320	
Frequencies 5.5 GHz	5500, 5520, 5540, 5560, 5580, 5600, 5620, 5640	
- Antenna Gain	0 dBi	19 dBi
- EIRP (one antenna)	6 dBm	29 dBm
- EIRP (two antennas)	9 dBm	29 dBm
- EIRP (three antennas)	11 dBm	29 dBm
- DFS	5500 to 5640	
Frequencies 5.8 GHz	5745, 5765, 5785, 5805, 5825, 5845, 5865	
- Antenna Gain	0 dBi	19 dBi
- EIRP (one antenna)	6 dBm	29 dBm
- EIRP (two antennas)	9 dBm	29 dBm
- EIRP (three antennas)	11 dBm	29 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 (2417 to 2457) = 20..22 dBm Max. EIRP (5180 to 5240) = 20 dBm (USA) Max. EIRP (5180 to 5240) = 22 dBm (CANADA) Max. EIRP (5260 to 5320) = 29 dBm (USA) Max. EIRP (5260 to 5320) = 22 dBm (CANADA) Max. EIRP (5500 to 5700) = 29 dBm Max. EIRP (5745 to 5825) = 18 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) = 20 dBm (USA) Max. EIRP (5180 to 5240) = 22 dBm (CANADA) Max. EIRP (5260 to 5320) = 29 dBm (USA) Max. EIRP (5260 to 5320) = 22 dBm (CANADA) Max. EIRP (5500 to 5700) = 29 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) = 20 dBm (USA) Max. EIRP (5180 to 5240) = 22 dBm (CANADA) Max. EIRP (5260 to 5320) = 29 dBm (USA) Max. EIRP (5260 to 5320) = 22 dBm (CANADA) Max. EIRP (5500 to 5700) = 29 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) = 20 dBm (USA) Max. EIRP (5180 to 5240) = 22 dBm (CANADA) Max. EIRP (5260 to 5320) = 29 dBm (USA) Max. EIRP (5260 to 5320) = 22 dBm (CANADA) Max. EIRP (5500 to 5700) = 29 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 802.11ax 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 two-letter country codes are supported for Europe: 'AT', 'BE', 'BG', 'CH', 'CY', 'CZ', 'DE', 'DK', 'EE', 'ES', 'FI', 'FR', 'GB', 'GR', 'HR', 'HU', 'IE', 'IS', 'IT', 'LI', 'LT', 'LU', 'LV', 'MT', 'NL', 'NO', 'PL', 'PT', 'RO', 'SE', 'SI', 'SK', '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
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
- Indoor frequencies	5180 to 5320	
- DFS (slave/master)	5260 to 5700	
Frequencies 5.8 GHz	5745, 5765, 5785, 5805, 5825	
- Antenna Gain	0 dBi	10 dBi
- EIRP (one antenna)	6 dBm	14 dBm
- EIRP (two antennas)	9 dBm	14 dBm
Frequencies 6 GHz	5955, 5975, 5995, 6015, 6035, 6055, 6075, 6095, 6115, 6135, 6155, 6175, 6195, 6215, 6235, 6255, 6275, 6295, 6315, 6335, 6355, 6375, 6395, 6415	
- Antenna Gain	0 dBi	10 dBi
- EIRP (one antenna)	0 dBm	12 dBm
- EIRP (two antennas)	3 dBm	12 dBm
- EIRP (three antennas)	5 dBm	12 dBm
- EIRP (four antennas)	6 dBm	12 dBm

6.4 Regions for 802.11be products

6.4.1 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)

6.4.1.1 Europe

Following two-letter country codes are supported for Europe: 'AT', 'BE', 'BG', 'CH', 'CY', 'CZ', 'DE', 'DK', 'EE', 'ES', 'FI', 'FR', 'GB', 'GR', 'HR', 'HU', 'IE', 'IS', 'IT', 'LI', 'LT', 'LU', 'LV', 'MT', 'NL', 'NO', 'PL', 'PT', 'RO', 'SE', 'SI', 'SK', '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
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
- Indoor frequencies	5180 to 5320	
- DFS (slave/master)	5260 to 5700	
Frequencies 5.8 GHz	5745, 5765, 5785, 5805, 5825	
- Antenna Gain	0 dBi	10 dBi
- EIRP (one antenna)	6 dBm	14 dBm
- EIRP (two antennas)	9 dBm	14 dBm
Frequencies 6 GHz	5955, 5975, 5995, 6015, 6035, 6055, 6075, 6095, 6115, 6135, 6155, 6175, 6195, 6215, 6235, 6255, 6275, 6295, 6315, 6335, 6355, 6375, 6395, 6415	
- Antenna Gain	0 dBi	10 dBi
- EIRP (one antenna)	0 dBm	12 dBm
- EIRP (two antennas)	3 dBm	12 dBm

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

7.1.0.1 Disable Unused Ports

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

```
WESTERMO-SW6-MIB::cfgNetEthEnabled.1 = 0
```

Listing 7.1: *Disable the second port (eth1)*

Reference Connector Labels which describe what ports are available.

7.1.0.2 Enable 802.1X Authentication on Unused Ports

When the local port is required for temporary access to the network, e.g. for maintenance work, it is recommended to only allow authenticated users. For this 802.1X RADIUS authentication may be enabled on the port(s). See Wired 802.1X Authentication for more information and an example.

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 without 802.1X RADIUS authentication. 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 with it's own IP. Additionally, firewall rules prevent traffic from being routed from this interface. This will allow local access to the administrative interface of the device, but no further access to the backbone.

```
# 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
# IP on eth1
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 = eth1
# Enable Firewall
WESTERMO-SW6-FIREWALL-MIB::cfgFwEnabled.0 = 1
# Do not forward traffic from eth1
WESTERMO-SW6-FIREWALL-MIB::cfgFwFltREnabled.0 = 1
WESTERMO-SW6-FIREWALL-MIB::cfgFwFltRChain.0 = 2
WESTERMO-SW6-FIREWALL-MIB::cfgFwFltRAction.0 = 0
WESTERMO-SW6-FIREWALL-MIB::cfgFwFltRInputInterface.0 = eth1
```

Listing 7.2: *Enable local administrative access on second port (eth1)*

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

```
# 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
```

Listing 7.3: *Allow remote administrative access on separate VLAN*

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
- SSDP - WESTERMO-SW6-MIB::cfgSsdpEnabled.0
- DHCP server - WESTERMO-SW6-MIB::cfgDhcpGlobalEnabled.0
- DHCP client - WESTERMO-SW6-MIB::cfgNetIpEnabled.3

The DHCP client is configured as an IP address alias by default and should either be removed from the configuration file or deactivated via the configuration parameter above if it is not used.

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 configuration parameters:

- CLI - username: WESTERMO-SW6-MIB::cfgCliUsername.0
- CLI - password: WESTERMO-SW6-MIB::cfgCliPassword.0
- HTTP - admin password: WESTERMO-SW6-MIB::cfgHttpAdminPasswordHash.0
- HTTP - monitor password: WESTERMO-SW6-MIB::cfgHttpMonitorPasswordHash.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

Note: HTTP password of user *admin* and *monitor* can be change on the Web Interface while logged-in as the respective user.

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.

7.5 Firewall

In addition to the previously described measures to limit access to the management interfaces of the device, the Firewall may be configured to prevent undesired traffic as well. Especially when an interface is connected to the Internet, the Firewall should be configured to prevent inbound traffic on the Internet facing interface.

In addition it may be desirable to add rules to prevent leaking of frames with a local address (192.168.0.0/16, 172.16.0.0/12 and 10.0.0.0/8).

```
# Enable Firewall
WESTERMO-SW6-FIREWALL-MIB::cfgFwEnabled.0 = 1

# Perform SNAT for any traffic egressing on wwan0
WESTERMO-SW6-FIREWALL-MIB::cfgFwNatOutEnabled.0 = 1
WESTERMO-SW6-FIREWALL-MIB::cfgFwNatOutInterface.0 = wwan0
```

```
WESTERMO-SW6-FIREWALL-MIB::cfgFwNatOutSourceAddress.0 = 0.0.0.0/0
WESTERMO-SW6-FIREWALL-MIB::cfgFwNatOutDestinationAddress.0 = 0.0.0.0/0
WESTERMO-SW6-FIREWALL-MIB::cfgFwNatOutSourceRewriteAddress.0 = 0.0.0.0

# Drop any traffic forwarded on wwan0 with a destination of 10.0.0.0/8, 172.16.0.0/12 or 192.168.0.0/16
WESTERMO-SW6-FIREWALL-MIB::cfgFwFltREnabled.0 = 1
WESTERMO-SW6-FIREWALL-MIB::cfgFwFltREnabled.1 = 1
WESTERMO-SW6-FIREWALL-MIB::cfgFwFltREnabled.2 = 1
WESTERMO-SW6-FIREWALL-MIB::cfgFwFltRChain.0 = 2
WESTERMO-SW6-FIREWALL-MIB::cfgFwFltRChain.1 = 2
WESTERMO-SW6-FIREWALL-MIB::cfgFwFltRChain.2 = 2
WESTERMO-SW6-FIREWALL-MIB::cfgFwFltRAction.0 = 0
WESTERMO-SW6-FIREWALL-MIB::cfgFwFltRAction.1 = 0
WESTERMO-SW6-FIREWALL-MIB::cfgFwFltRAction.2 = 0
WESTERMO-SW6-FIREWALL-MIB::cfgFwFltROutputInterface.0 = wwan0
WESTERMO-SW6-FIREWALL-MIB::cfgFwFltROutputInterface.1 = wwan0
WESTERMO-SW6-FIREWALL-MIB::cfgFwFltROutputInterface.2 = wwan0
WESTERMO-SW6-FIREWALL-MIB::cfgFwFltRSourceAddress.0 = 0.0.0.0/0
WESTERMO-SW6-FIREWALL-MIB::cfgFwFltRSourceAddress.1 = 0.0.0.0/0
WESTERMO-SW6-FIREWALL-MIB::cfgFwFltRSourceAddress.2 = 0.0.0.0/0
WESTERMO-SW6-FIREWALL-MIB::cfgFwFltRDestinationAddress.0 = 10.0.0.0/8
WESTERMO-SW6-FIREWALL-MIB::cfgFwFltRDestinationAddress.1 = 172.16.0.0/12
WESTERMO-SW6-FIREWALL-MIB::cfgFwFltRDestinationAddress.2 = 192.168.0.0/16
```

Listing 7.4: *Prevent Leaking of Private Traffic on wwan0*

7.6 Logging

There are two possibilities to get information about what's going on the device from the device:

- Syslog
- SNMP Traps

7.6.1 Syslog

Every user should consider to setup a dedicated syslog server to get the syslog information from the device. This is use-full for real-time monitoring of the device and also for further analysis.

The recommended syslog protocol is *TCPTLS* (Secure Syslog, encrypted) or *TCP* (unencrypted) because it's more reliable than UDP (see cfgLogRemoteProtocol).

See Logging Features for more information about how to use.

7.6.2 SNMP Traps

The SNMP traps can be used to get notified on explicit events. In contrast to syslog where every message is written to, every trap has a message code which can be used for filtering (e.g. get notified if a login to Web Interface was denied). See Message Codes for a list of all available traps.

These traps can be received from the most common NMS so it can react to this events and trigger further actions.

We recommend to use type *Informs* because it's more reliable than *Traps* (see cfgSnmpTrapType).

See SNMP Trap for more information about how to use.

8 WESTERMO-SW6-MIB

8.1 configuration

8.1.1 cfgSystem

8.1.1.1 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 or end with a hyphen.

Type	DisplayString
Range	1 - 63
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1.1

8.1.1.2 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.

Range	0 - 5
OID	1.3.6.1.4.1.16177.1.400.1.1.1.10

8.1.1.2.1 cfgSysSearchdomainTableEntry

Searchdomain configuration.

OID	1.3.6.1.4.1.16177.1.400.1.1.1.10.1

8.1.1.2.1.1 cfgSysSdIndex

Table Entry Index

Range	0 - 5
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.1.10.1.1

8.1.1.2.1.2 cfgSysSdSearch

Search Domain List Entry

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

Example:

- `example.com`
- `subdomain.otherdomain.org`

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1.10.1.2

8.1.1.3 cfgSysNameserverTable

Nameserver configuration

Configure a prioritised list of nameservers with which the device may resolve hostnames. The lower the index, the higher the priority.

Entries are tried either in parallel or sequentially. When querying sequentially, the timeout is 2 seconds. When querying in parallel the timeout is 5 seconds. Either way, 2 attempts are performed.

There are two types of entries: static and dynamic.

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.

Entries received by DHCP, may also be ignored and not used at all by setting `cfgSysNsType` to **ignoreinterface(3)**.

A maximum of 6 entries are queried. When more entries are defined, the additional ones are ignored.

Range	0 - 15
OID	1.3.6.1.4.1.16177.1.400.1.1.1.11

8.1.1.3.1 cfgSysNameserverTableEntry

Nameserver Configuration

OID	1.3.6.1.4.1.16177.1.400.1.1.1.11.1

8.1.1.3.1.1 cfgSysNsIndex

Table Entry Index

Range	0 - 15
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.1.11.1.1

8.1.1.3.1.2 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`
- **ignoreinterface(3):** Ignore nameservers provided by a DHCP client referenced by `cfgSysNsDhcpInterfac`

Enumeration	none (0), server (1), dhcpinterface (2), ignoreinterface (3)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1.11.1.2

8.1.1.3.1.3 cfgSysNsServer

Nameserver Address

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

Type	IpAddress
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1.11.1.3

8.1.1.3.1.4 cfgSysNsDhcpInterface

DHCP Client Interface

This parameter is only used when `cfgSysNsType` is set to **dhcpinterface(2)** or **ignoreinterface(3)**.

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

Examples:

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

Type	DisplayString
Range	1 - 15
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1.11.1.4

8.1.1.3.1.5 cfgSysNsDhcpDnsRouteEnabled

DHCP DNS Route Disabled or Enabled

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

When this parameter is set to **enabled(1)**, routing rules and routes are created to force DNS traffic to this server to the interface on which the DHCP client is running.

This means that DNS requests for DNS server entries received on wlan0 will be sent via wlan0, even if the routing table would send the requests over a different interface, e.g eth0.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1.11.1.5

8.1.1.4 cfgSysNameserverOrder

Nameserver Order

Defines the order in which the available nameservers are querried.

- **parallel(0)**: All servers are queried in parallel. The first response is used, including NXDomain (not found).
- **sequential(1)**: The servers are queried in the order in which they are defined. Use `cfgSysNameserverTabl` to specify the order of nameservers via DHCP.

Enumeration	parallel (0), sequential (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1.12

8.1.1.5 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

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1.2

8.1.1.6 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.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1.3

8.1.1.7 cfgSysComment

System Comment

This parameter has no operational function. A simple comment field to add customised information, such as configuration version, release notes or other remarks.

Type	DisplayString
Range	0 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1.4

8.1.2 cfgCli

8.1.2.1 cfgCliEnabled

Enable or Disable CLI

The CLI is available via Telnet or SSH.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.100.1

8.1.2.2 cfgCliSshSessionTimeout

SSH Session Timeout

Disconnect the SSH session if no traffic is transmitted or received for ‘session timeout’ seconds.

Setting this parameter to `0` disables the session timeout.

Range	0 - 86400
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.100.10

8.1.2.3 cfgCliPasswordHash

CLI Password Hash

The password hash for CLI access.

To set a new password, a hash of a password or a clear-text password may be written to this parameter. Reading the password always returns the hashed version.

To generate a hash offline, use the command `openssl passwd -6`.

Type	DisplayString
Range	10 - 126
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.100.11

8.1.2.4 cfgCliUsername

CLI Username

The username with which the CLI may be accessed.

The username can not be set to one of the following system users:

- root
- daemon
- ftp
- network
- nobody
- user
- openvpn
- ntp
- chrony
- dnymasq
- lldp
- logd
- ssdp

- ubus

Type	DisplayString
Range	1 - 31
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.100.2

8.1.2.5 cfgCliPassword

Deprecated. Use `cfgCliPasswordHash`

Type	DisplayString
Range	0 - 126
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.100.3

8.1.2.6 cfgCliTelnetEnabled

Enable or Disable Telnet for CLI

This parameter enables the telnet server on the specified address and port (see `cfgCliTelnetAddress`).

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.100.4

8.1.2.7 cfgCliSshEnabled

Enable or Disable SSH for CLI

This parameter enables the SSH server on the specified address and port (see `cfgCliSshAddress`).

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.100.6

8.1.2.8 cfgCliTelnetAddress

Telnet Server Address and Port

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

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.100.8

8.1.2.9 cfgCliSshAddress

SSH Server Address and Port

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

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.100.9

8.1.3 cfgCertificate

8.1.3.1 cfgCrtCrlTable

Certificate Revocation List configuration table.

Range	0 - 15
OID	1.3.6.1.4.1.16177.1.400.1.1.1001.1

8.1.3.1.1 cfgCrtCrlTableEntry

Certificate Revocation List configuration table entry.

OID	1.3.6.1.4.1.16177.1.400.1.1.1001.1.1

8.1.3.1.1.1 cfgCrtCrlIndex

Table Entry Index

Range	0 - 15
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.1001.1.1.1

8.1.3.1.1.2 cfgCrtCrlCaId

Reference to the CA ID in the certificate store.

The received CRL will be stored to the CRL ID of the CRL associated to the configured CA ID.

Applies to AP and STA.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1001.1.1.2

8.1.3.1.1.3 cfgCrtCrlEnabled

Enable automatic download of the CRL

The CRL must be in the DER format.

Applies to AP and STA.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1001.1.1.3

8.1.3.1.1.4 cfgCrtCrlUrl

CRL URL

This URL is used to download the new CRL. The CRL must be in the DER format.

Example:

- http://192.168.1.2/certs/example.crl

Applies to AP and STA.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1001.1.1.4

8.1.3.1.1.5 cfgCrtCrlTimeBeforeExpire

Time before the current CRL expire

The new CRL will be downloaded this number of days before the CRL expire.

Applies to AP and STA.

Range	0 - 365
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1001.1.1.5

8.1.3.1.1.6 cfgCrtCrlRetryPeriod

Retry period in minutes

If the download of a CRL failed the device will retry to download after this period.

Applies to AP and STA.

Range	1 - 1440
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1001.1.1.6

8.1.3.2 cfgCrtMonitoringTable

Certificate Monitoring configuration table.

Range	0 - 15
OID	1.3.6.1.4.1.16177.1.400.1.1.1001.2

8.1.3.2.1 cfgCrtMonitoringTableEntry

Certificate Monitoring configuration table entry.

OID	1.3.6.1.4.1.16177.1.400.1.1.1001.2.1

8.1.3.2.1.1 cfgCrtMonIndex

Table Entry Index

Range	0 - 15
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.1001.2.1.1

8.1.3.2.1.2 cfgCrtMonEnabled

Enable certificate/CRL expiry monitor

The monitor observes if a certificate/CRL is about to expire and informs via Trap/Syslog if this is the case.

Applies to AP and STA.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1001.2.1.2

8.1.3.2.1.3 cfgCrtMonType

Monitor type

Define the type to be monitored.

- **1**: CRL
- **2**: Certificate

Applies to AP and STA.

Enumeration	crl (1), cert (2)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1001.2.1.3

8.1.3.2.1.4 cfgCrtMonId

ID of the certificate to be monitored

Reference to the certificate ID in the certificate store.

In order to monitor CA and Client/Server certificates the corresponding ID needs to be specified and the `cfgCrtMonType` needs to set to **cert(2)**.

In order to monitor CRL the ID of the associated CA ID needs to be specified and the `cfgCrtMonType` needs to set to **crl(1)**.

Applies to AP and STA.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1001.2.1.4

8.1.3.2.1.5 cfgCrtMonTimeBeforeExpire

Start alert this time before the current CA certificate expire

This value is in days before the CA certificate expire.

Applies to STA.

Range	0 - 365
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1001.2.1.5

8.1.3.2.1.6 cfgCrtMonRepeatPeriod

Repeat period for CA certificate expiration alert

Repeat the alert every X hours.

Applies to STA.

Range	1 - 24
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1001.2.1.6

8.1.3.3 cfgCrtGlobal

8.1.3.3.1 cfgCrtGlblDailyRefreshEnabled

Enable Daily Certificate Refresh

Applies to AP and STA.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1001.3.1

8.1.3.3.2 cfgCrtGlblDailyRefreshTime

Daily Automatic Refresh time.

Define time (hour:minute) at which a certificate refresh is called (if `cfgCrtGlblDailyRefreshEnabled` is **enabled(1)**).

The time is referenced to the local time as define in `cfgSysTimezone`

Examples:

- 00:00 - force refresh each day at midnight
- 01:00 - force refresh each day at 01:00
- 23:05 - force refresh each day at 23:05

Applies to AP and STA.

Type	DisplayString
Range	5 - 5
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1001.3.2

8.1.4 cfgScep

8.1.4.1 cfgScepTable

SCEP Table

Applies to STA.

Range	0 - 255
OID	1.3.6.1.4.1.16177.1.400.1.1.1002.2

8.1.4.1.1 cfgScepTableEntry

SCEP Table Entry

Applies to STA.

OID	1.3.6.1.4.1.16177.1.400.1.1.1002.2.1

8.1.4.1.1.1 cfgScepIndex

Index of the Table Entry

Applies to STA.

Range	0 - 255
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.1002.2.1.1

8.1.4.1.1.2 cfgScepAutoRenewRetryPeriod

SCEP Certificate Renew Period

SCEP re-enrollment retry interval in minutes.

Applies to STA.

Range	5 - 1440
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1002.2.1.10

8.1.4.1.1.3 cfgScepCsrCN

CN (Common Name) field for CSR

Example:

- example.com

Applies to STA.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1002.2.1.11

8.1.4.1.1.4 cfgScepServerUrl

URL of SCEP server

Example:

- http://192.168.1.2:8080/scep

Applies to STA.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1002.2.1.12

8.1.4.1.1.5 cfgScepCsrC

C (Country) field for CSR

If set to 'none', C is not used for CSR.

Example:

- CH

Applies to STA.

Type	DisplayString
Range	1 - 4
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1002.2.1.13

8.1.4.1.1.6 cfgScepCsrST

ST (State) field for CSR

If set to 'none', ST is not used for CSR.

Example:

- Zurich

Applies to STA.

Type	DisplayString
Range	1 - 63
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1002.2.1.14

8.1.4.1.1.7 cfgScepCsrL

L (Locality) field for CSR

If set to 'none', L is not used for CSR.

Example:

- Bubikon

Applies to STA.

Type	DisplayString
Range	1 - 63
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1002.2.1.15

8.1.4.1.1.8 cfgScepCsrO

O (Organization) field for CSR

If set to 'none', O is not used for CSR.

Applies to STA.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1002.2.1.16

8.1.4.1.1.9 cfgScepCsrOU

OU (Organizational Unit) field for CSR

If set to 'none', OU is not used for CSR.

Applies to STA.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1002.2.1.17

8.1.4.1.1.10 cfgScepCaId

Reference ID for CA Certificate

This certificate is used for the signing and optionally also for the encryption. If the SCEP server requires two separate certificates, the RA encryption certificate must be selected with the `cfgScepRaId` parameter.

Set to -1 if not using a CA certificate.

Applies to STA.

Range	-1 - 65535
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1002.2.1.18

8.1.4.1.1.11 cfgScepCertId

Reference ID for Client Certificate

Set to -1 if not using a client certificate.

Applies to STA.

Range	-1 - 65535
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1002.2.1.19

8.1.4.1.1.12 cfgScepRaId

Reference ID for the RA Certificate

This is an optional RA encryption certificate. It is needed when the encryption is done with a separate certificate.

Set to -1 if not using a RA certificate.

Applies to STA.

Range	-1 - 65535
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1002.2.1.20

8.1.4.1.1.13 cfgScepCaIdentifier

CA Identifier

Certification authority (CA) issuer identifier (if your SCEP server requires it). A CA Identifier is any string that is understood by the SCEP server (e.g. a domain name).

If set to ‘none’, CA Identifier is not used.

Applies to STA.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1002.2.1.3

8.1.4.1.1.14 cfgScepChallengePassword

SCEP Challenge Password

If set to ‘none’, Challenge Password is not used.

Applies to STA.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1002.2.1.5

8.1.4.1.1.15 cfgScepPollingInterval

SCEP Polling Interval in seconds

Applies to STA.

Range	1 - 86400
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1002.2.1.6

8.1.4.1.1.16 cfgScepPollingMaxTries

Max number of SCEP GetCertInitial requests

Applies to STA.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1002.2.1.7

8.1.4.1.1.17 cfgScepAutoRenewEnabled

Enable/disable SCEP automatic re-enrollment

Applies to STA.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1002.2.1.8

8.1.4.1.1.18 cfgScepAutoRenewTimeBeforeExpire

SCEP Certificate Renew Days

Number of days before certificate expiration (i.e. automatic re-enrollment shall start these number of days before certificate expiration)

Applies to STA.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1002.2.1.9

8.1.5 cfgVpn

8.1.5.1 cfgVpnOpenvpn

8.1.5.1.1 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`.

Range	0 - 15
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.1.1

8.1.5.1.1.1 cfgVpnOpenvpnTableEntry

OpenVPN Instance Table Entry

OID	1.3.6.1.4.1.16177.1.400.1.1.1003.1.1.1

cfgVpnOpenvpnIndex

Table Entry Index

Range	0 - 15
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.1.1.1.1

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)**.

Enumeration	tun (0), tap (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.1.1.1.10

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
- auth-user-pass
- tls-verify
- ca
- cert
- key
- secret
- askpass
- tls-auth

Essentially everything which calls a script, changes files, or otherwise allows to adjust sensitive setting on the system.

When a custom option is set which is configurable via an existing parameter, the custom option will take precedence. A message is shown in syslog/TRAP notifing about the duplicate entry.

Type	DisplayString
Range	4 - 4095
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.1.1.1.1000

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 existance of its peer

Note: If OpenVPN is in client mode, this parameter may be overridden by the server.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.1.1.1.11

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.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.1.1.1.12

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.

Range	1 - 65535
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.1.1.1.13

cfgVpnOpenvpnConnectRetryLimit

Connect Retry Interval Limit

The maximum value of wait time in seconds (see `cfgVpnOpenvpnConnectRetry`) at which it gets capped.

Range	1 - 65535
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.1.1.1.14

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.

Enumeration	disabled (0), allowPush (1), lzo (2), lz4 (3), lz4v2 (4)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.1.1.1.15

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

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.1.1.1.16

cfgVpnOpenvpnKeyPassword

****OBSOLETE:** Password to Unlock Private Key**

This parameter is obsolete and has been replaced with the Certificate Store.

Key material on the device is always encrypted. The password to import the file has to be specified once during import via `setCrtFilePassphrase`

Type	DisplayString
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.1.1.1.17

cfgVpnOpenvpnKeyType

Key Type

- **asymmetric(0)**: Use an asymmetric encryption with certificate, private and public keys (TLS).
- **symmetric(1)**: Use encryption with a static key.
- **combinedAuth(2)**: Use asymmetric encryption and sign the control channel with a static key (tls-auth).
- **combinedCrypt(3)**: Use asymmetric encryption and encrypt the control channel with a static key (tls-crypt).

Note: The key material for the asymmectric encryption is managed by the Certificate Store.

Enumeration	asymmectric (0), symmetric (1), combinedAuth (2), combinedCrypt (3)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.1.1.1.18

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.

Enumeration	omitted (-1), zero (0), one (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.1.1.1.19

cfgVpnOpenvpnMode

Mode

- **client(0):** OpenVPN instance is a client and connects to a server
- **server(1): NOT IMPLEMENTED YET**

Enumeration	client (0), server (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.1.1.1.2

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.

Enumeration	disabled (0), client (1), server (2)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.1.1.1.20

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.

Enumeration	disabled (0), name (1), prefix (2), subject (3)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.1.1.1.21

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.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.1.1.1.22

cfgVpnOpenvpnUsername

Username

Authenticate with the server using the given username.

It is disabled when set to **none**.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.1.1.1.23

cfgVpnOpenvpnPassword

Password

Authenticate with the server using the given password.

It is disabled when set to **none**.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.1.1.1.24

cfgVpnOpenvpnCaIds

OpenVPN CA ID(s)

This value contain the id(s) to reference the ca certificate in the certificate store.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.1.1.1.25

cfgVpnOpenvpnCertId

OpenVPN Certificate ID

This value contain the id to reference a certificate in the certificate store.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.1.1.1.26

cfgVpnOpenvpnStaticKeyId

OpenVPN Static Key ID

This value contain the id to reference the static key in the certificate store.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.1.1.1.27

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.

Type	IpAddress
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.1.1.1.3

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.

Range	0 - 65535
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.1.1.1.4

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`).

Type	DisplayString
Range	4 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.1.1.1.5

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.

Range	1 - 65535
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.1.1.1.6

cfgVpnOpenvpnProto

Transport Protocol

The following transport protocols are available:

- **UDP(0)**: User Datagram Protocol
- **TCP(1)**: Transmission Control Protocol

Enumeration	udp (0), tcp (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.1.1.1.7

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**
- **SHA1**: according to blank `auth` entry in ovpn config file
- **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.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.1.1.1.8

cfgVpnOpenvpnCipher

Data-Channel Encryption Cipher List

Colon separated list of ciphers to allow for the OpenVPN data-channel encryption.

Set to 'none' to disable packet encryption.

Examples:

- **AES-256-GCM:AES-128-GCM**
- **AES-256-CBC**
- **AES-256-GCM**
- **none**: to disable packet encryption

For a full list of supported algorithms please consult the user manual.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.1.1.1.9

8.1.5.2 cfgVpnIpsec

8.1.5.2.1 cfgVpnIpsecTable

IPsec Table

The used IPsec implementation is strongSwan. For more detailed configuration explanations and examples see https://wiki.strongswan.org/projects/strongswan/wiki/IpsecConf

Range	0 - 15
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.2.1

8.1.5.2.1.1 cfgVpnIpsecTableEntry

IPsec Table Entry

OID	1.3.6.1.4.1.16177.1.400.1.1.1003.2.1.1

cfgVpnIpsecIndex

Table Entry Index

Range	0 - 15
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.2.1.1.1

cfgVpnIpsecType

IPsec Type

The type of the connection; currently the accepted values are:

- **tunnel(0)**: signifying a host-to-host, host-to-subnet, or subnet-to-subnet tunnel
- **transport(1)**: signifying host-to-host transport mode
- **transportProxy(2)**: signifying the special Mobile IPv6 transport proxy mode
- **passthrough(3)**: signifying that no IPsec processing should be done at all
- **drop(4)**: signifying that packets should be discarded.

Enumeration	tunnel (0), transport (1), transportProxy (2), passthrough (3), drop (4)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.2.1.1.100

cfgVpnIpsecCustomOptions

Custom IPsec Options

These options are appended to the IPsec 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://wiki.strongswan.org/projects/strongswan/wiki/ConnSection

Prohibited options are:

- left
- leftid
- leftsubnet
- leftauth
- right
- rightid
- rightauth
- rightsubnet
- keyexchange

- mobike
- ikelifetime
- lifetime
- keyingtries
- type
- ike
- esp
- dpddelay
- dpdtimeout

Essentially everything which is possible to set by other SNMP commands.

Type	DisplayString
Range	4 - 4095
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.2.1.1.1000

cfgVpnIpsecAuto

IPsec Auto Startup Operation

What operation, if any, should be done automatically at IPsec startup.

- **ignore(0)**: Ignores the connection. This is equal to deleting a connection from the config file.
- **add(1)**: Loads a connection without starting it.
- **route(2)**: Loads a connection and installs kernel traps. If traffic is detected between leftsubnet (`cfgVpnIpsecLeftSubnet`) and rightsubnet (`cfgVpnIpsecRightSubnet`), a connection is established.
- **start(3)**: loads a connection and brings it up immediately.

Relevant only locally, other end need not agree on it.

Enumeration	ignore (0), add (1), route (2), start (3)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.2.1.1.101

cfgVpnIpsecKeyExchange

Key Exchange Method

Which protocol should be used to initialize the connection.

Enumeration	ikev1 (1), ikev2 (2)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.2.1.1.102

cfgVpnIpsecMobIke

Disable or Enable IKEv2 MOBIKE Protocol

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.

Enumeration	no (0), yes (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.2.1.1.103

cfgVpnIpsecIke

IKE/ISAKMP SA Encryption/Authentication Algorithms

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!

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.2.1.1.104

cfgVpnIpsecEsp

ESP Encryption/Authentication Algorithms

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 Diffe-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.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.2.1.1.105

cfgVpnIpsecIkeLifetime

IKE Lifetime

How long the keying channel of a connection (ISAKMP or IKE SA) should last before being renegotiated.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.2.1.1.106

cfgVpnIpsecLifetime

Connection Instance 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.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.2.1.1.107

cfgVpnIpsecKeyingTries

Keying Tries

When set to -1 means %forever, otherwise what is set.

Examples:

- **-1**: try forever
- **3**

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.2.1.1.108

cfgVpnIpsecDpdAction

Dead Peer Detection Protocol Usage

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 liveliness of the IPsec peer. The values clear, hold, and restart all activate DPD and determine the action to perform on a timeout. With **clear(1)** the connection is closed with no further actions taken. **hold(2)** installs a trap policy, which will catch matching traffic and tries to re-negotiate the connection on demand. **restart(3)** will immediately trigger an attempt to re-negotiate the connection.

The default is **none(0)** which disables the active sending of DPD messages.

Enumeration	none (0), clear (1), hold (2), restart (3)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.2.1.1.109

cfgVpnIpsecDpdDelay

Dead Peer Detection Delay

Defines the period time interval (in seconds) 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.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.2.1.1.110

cfgVpnIpsecDpdTimeout

DPD Timeout

Defines the timeout interval in seconds, 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.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.2.1.1.111

cfgVpnIpsecKeyPassword

****OBSOLETE:** Password to Unlock Private Key**

This parameter is obsolete and has been replaced with the Certificate Store.

Key material on the device is always encrypted. The password to import the file has to be specified once during import via `setCrtFilePassphrase`

Type	DisplayString
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.2.1.1.112

cfgVpnIpsecPassword

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 repre-

sented 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.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.2.1.1.113

cfgVpnIpsecCloseAction

Action When Link is Closed by Remote Peer

Defines the action to take if the remote peer unexpectedly closes a CHILD_SA. This may happen when the remote site is reconfigured, or goes down for maintenance. A closeaction should not be used if the peer uses reauthentication (see `cfgVpnIpsecReauth`) or uniqueids checking, as these events might trigger the defined action when not desired.

Available actions are:

- **none(0)** No action is taken. This disables the Close Action.
- **clear(1)** The connection is closed with no further actions taken.
- **hold(2)** Installs a trap policy, which will catch matching traffic and tries to re-negotiate the connection on demand.
- **restart(3)** Will immediately trigger an attempt to re-negotiate the connection.

Enumeration	none (0), clear (1), hold (2), restart (3)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.2.1.1.114

cfgVpnIpsecReauth

Reauthentication of Peer During Rekeying

Whether rekeying of an IKE_SA should also reauthenticate the peer. In IKEv1, reauthentication is always done.

In IKEv2, a value of:

- **disabled(0)** Rekeys without uninstalling the IPsec SAs
- **enabled(1)** Creates a new IKE_SA from scratch and tries to recreate all IPsec SAs

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.2.1.1.115

cfgVpnIpsecCaIds

IPsec CA Certificate ID

This value contains the id(s) to reference the ca certificate in the certificate store.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.2.1.1.116

cfgVpnIpsecLeftCertId

IPsec Left Certificate ID

This value contains the id of the certificate in the certificate store used for the left side.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.2.1.1.117

cfgVpnIpsecRightCertId

IPsec Right Certificate ID

This value contains the id of the certificate in the certificate store used for the right side.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.2.1.1.118

cfgVpnIpsecLeftSigkeyId

IPsec Left Sig Key ID

This value contains the id of the sig key in the certificate store used for the left side.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.2.1.1.119

cfgVpnIpsecRightSigkeyId

IPsec Right Sig Key ID

This value contains the id of the sig key in the certificate store used for the right side.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.2.1.1.120

cfgVpnIpsecLeftKeyId

IPsec Left Key ID

This value contains the id of the private key in the certificate store used for the left side.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.2.1.1.121

cfgVpnIpsecLeft

Left or Local

left = ip address | fqdn | %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 reistricts 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.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.2.1.1.2

cfgVpnIpsecRight

Right or Remote

right = ip address | fqdn | %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 reistricts 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.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.2.1.1.3

cfgVpnIpsecLeftSubnet

IPsec Left Subnet

leftsubnet = ip subnet[[proto/port]][,...]

Private subnet behind the left/local participant, expressed as network/netmask. The configured subnets of the peers may differ, the protocol narrows it to the greatest common subnet.

This is also done for IKEv1, but as this may lead to problems with other implementations, make sure to configure identical subnets in such configurations. IKEv2 supports multiple subnets separated by commas, IKEv1 only interprets the first subnet of such a definition. This is due to a limitation of the IKEv1 protocol, which only allows a single pair of subnets per CHILD_SA. So to tunnel several subnets, a conn entry has to be defined and brought up for each pair of subnets.

The optional part after each subnet enclosed in square brackets specifies a protocol/port to restrict the selector for that subnet.

Examples:

- 10.0.0.1[tcp/http],10.0.0.2[6/80]
- fec1::1[udp],10.0.0.0/16[/53]

Instead of omitting either value %any can be used to the same effect, e.g. `fec1::1[udp/%any],10.0.0.0/16[%a`

If the protocol is icmp or ipv6-icmp the port is interpreted as ICMP message type if it is less than 256, or as type and code if it greater or equal to 256, with the type in the most significant 8 bits and the code in the least significant 8 bits.

The port value can alternatively take the value %opaque for RFC 4301 OPAQUE selectors, or a numerical range in the form 1024-65535. None of the kernel backends currently supports opaque or port ranges and uses %any for policy installation instead.

Instead of specifying a subnet, %dynamic can be used to replace it with the IKE address, having the same effect as omitting left|rightsubnet completely. Using %dynamic can be used to define multiple dynamic selectors, each having a potentially different protocol/port definition.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.2.1.1.4

cfgVpnIpsecRightSubnet

IPsec Right Subnet

rightsubnet = ip subnet[[proto/port]][,...]

Private subnet behind the right/remote participant, expressed as network/netmask. The configured subnets of the peers may differ, the protocol narrows it to the greatest common subnet.

This is also done for IKEv1, but as this may lead to problems with other implementations, make sure to configure identical subnets in such configurations. IKEv2 supports multiple subnets separated by commas, IKEv1 only interprets the first subnet of such a definition. This is due to a limitation of the IKEv1 protocol, which only allows a single pair of subnets per CHILD_SA. So to tunnel several subnets a conn entry has to be defined and brought up for each pair of subnets.

The optional part after each subnet enclosed in square brackets specifies a protocol/port to restrict the selector for that subnet.

Examples:

- 10.0.0.1[tcp/http],10.0.0.2[6/80]
- fec1::1[udp],10.0.0.0/16[/53]

Instead of omitting either value %any can be used to the same effect, e.g. `fec1::1[udp/%any],10.0.0.0/16[%a`

If the protocol is icmp or ipv6-icmp the port is interpreted as ICMP message type if it is less than 256, or as type and code if it greater or equal to 256, with the type in the most significant 8 bits and the code in the least significant 8 bits.

The port value can alternatively take the value %opaque for RFC 4301 OPAQUE selectors, or a numerical range in the form 1024-65535. None of the kernel backends currently supports opaque or port ranges and uses %any for policy installation instead.

Instead of specifying a subnet, %dynamic can be used to replace it with the IKE address, having the same effect as omitting left|rightsubnet completely. Using %dynamic can be used to define multiple dynamic selectors, each having a potentially different protocol/port definition.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.2.1.1.5

cfgVpnIpsecLeftId

IPsec Left ID

leftid = id

How the left/local participant should be identified for authentication; defaults to left or the subject of the certificate configured with leftcert. If leftcert is configured, the identity has to be confirmed by the certificate, that is, it has to match the full subject DN or one of the subjectAltName extensions contained in the certificate.

Can be an IP address, a fully-qualified domain name, an email address or a Distinguished Name for which the ID type is determined automatically and the string is converted to the appropriate encoding. The rules for this conversion are described on IdentityParsing (see https://wiki.strongswan.org/projects/strongswan/

In certain special situations the identity parsing above might be inadequate or produce the wrong result. Examples are the need to encode a FQDN as KEY_ID or the string parser being unable to produce the correct binary ASN.1 encoding of a certificate's DN. For these situations it is possible to enforce a specific identity type and to provide the binary encoding of the identity. To do this a prefix may be used, followed by a colon (:). If the number sign (#) follows the colon, the remaining data is interpreted as hex encoding, otherwise the string is used as is as the identification data. Note: The latter implies that no conversion is performed for non-string identities. For example, ipv4:10.0.0.1 does not create a valid ID_IPV4_ADDR IKE identity, as it does not get converted to binary 0x0a000001. Instead, one could use ipv4:#0a000001 to get a valid identity, but just using the implicit type with automatic conversion is usually simpler. The same applies to the ASN.1 encoded types.

The following prefixes are known: ipv4, ipv6, rfc822, email, userfqdn, fqdn, dns, asn1dn, asn1gn and keyid.

Custom type prefixes may be specified by surrounding the numerical type value with curly brackets.

rightid for IKEv2 connections optionally takes a % as prefix in front of the identity. If given it prevents the daemon from sending IDr in its IKE_AUTH request and will allow it to verify the configured identity against the subject and subjectAltNames contained in the responder's certificate (otherwise, it is only compared with the IDr returned by the responder). The IDr sent by the initiator might otherwise prevent the responder from finding a config if it has configured a different value for leftid.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.2.1.1.6

cfgVpnIpsecRightId

IPsec Right ID

rightid = id

How the right/remote participant should be identified for authentication; defaults to right or the subject of the certificate configured with rightcert. If rightcert is configured, the identity has to be confirmed by the certificate, that is, it has to match the full subject DN or one of the subjectAltName extensions contained in the certificate.

Can be an IP address, a fully-qualified domain name, an email address or a Distinguished Name for which the ID type is determined automatically and the string is converted to the appropriate encoding. The rules for this conversion are described on IdentityParsing (see https://wiki.strongswan.org/projects/strongswan/

In certain special situations the identity parsing above might be inadequate or produce the wrong result. Examples are the need to encode a FQDN as KEY_ID or the string parser being unable to produce the correct binary ASN.1 encoding of a certificate's DN. For these situations it is possible to enforce a specific identity type and to provide the binary encoding of the identity. To do this a prefix may be used, followed by a colon (:). If the number sign (#) follows the colon, the remaining data is interpreted as hex encoding, otherwise the string is used as is as the identification data. Note: The latter implies that no conversion is performed for non-string identities. For example, ipv4:10.0.0.1 does not create a valid ID_IPV4_ADDR IKE identity, as it does not get converted to binary 0x0a000001. Instead, one could use ipv4:#0a000001 to get a valid identity, but just using the implicit type with automatic conversion is usually simpler. The same applies to the ASN.1 encoded types.

The following prefixes are known: ipv4, ipv6, rfc822, email, userfqdn, fqdn, dns, asn1dn, asn1gn and keyid.

Custom type prefixes may be specified by surrounding the numerical type value with curly brackets.

rightid for IKEv2 connections optionally takes a % as prefix in front of the identity. If given it prevents the daemon from sending IDr in its IKE_AUTH request and will allow it to verify the configured identity against the subject and subjectAltNames contained in the responder's certificate (otherwise, it is only compared with the IDr returned by the responder). The IDr sent by the initiator might otherwise prevent the responder from finding a config if it has configured a different value for leftid.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.2.1.1.7

cfgVpnIpsecLeftAuth

IPsec Left Authentication

leftauth = auth method

Examples:

- psk
- pubkey
- pubkey-sha256-sha512
- ecdsa-384
- bliss-sha512

Authentication method to use locally (left). Acceptable values are pubkey for public key encryption (RSA/ECDSA/BLISS), psk for pre-shared key authentication, eap to use the Extensible Authentication Protocol, and xauth for IKEv1 eXtended Authentication.

To require a trustchain public key strength for the remote side, specify the key type followed by the

minimum strength in bits (for example ecdsa-384 or rsa-2048-ecdsa-256). To limit the acceptable set of hashing algorithms for trustchain validation, append hash algorithms to pubkey or a key strength definition (for example pubkey-sha256-sha512, rsa-2048-sha256-sha384-sha512, or rsa-2048-sha256-ecdsa-256-sha256-sha384).

Unless explicit IKEv2 signature constraints are configured (see below), such key types and hash algorithms are also applied as constraints against IKEv2 signature authentication schemes used by the remote side.

If both peers support RFC 7427 (Signature Authentication in IKEv2) specific hash algorithms to be used during IKEv2 authentication may be configured. The syntax is the same as above, but with ike: prefix. For example, with ike:pubkey-sha384-sha256 a public key signature scheme with either SHA-384 or SHA-256 would get used for authentication, in that order and depending on the hash algorithms supported by the peer. If no specific hash algorithms are configured, the default is to prefer an algorithm that matches or exceeds the strength of the signature key. If no constraints with ike: prefix are configured any signature scheme constraint (without ike: prefix) will also apply to IKEv2 authentication.

RSASSA-PSS signatures are supported. To use or require them, configure rsa/pss instead of rsa as in e.g. ike:rsa/pss-sha256. If pubkey or rsa constraints are configured, RSASSA-PSS signatures will only be used/accepted if enabled in strongswan.conf.

In the case of eap, an optional EAP method can be appended. Currently defined methods are eap-aka, eap-gtc, eap-md5, eap-mschapv2, eap-peap, eap-sim, eap-tls, eap-ttls, eap-dynamic, and eap-radius. Alternatively, IANA assigned EAP method numbers are accepted. Vendor specific EAP methods are defined in the form eap-type-vendor (e.g. eap-7-12345).

Signature and trust chain constraints for EAP-(T)TLS may be defined. To do so, append a colon to the EAP method, followed by the key type/size and hash algorithm as discussed above. For xauth, an XAuth authentication backend can be specified, such as xauth-generic or xauth-eap. If XAuth is used in leftauth, Hybrid authentication is used. For traditional XAuth authentication, define XAuth in leftauth2.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.2.1.1.8

cfgVpnIpsecRightAuth

IPsec Right Authentication

rightauth = auth method

Examples:

- psk
- pubkey
- pubkey-sha256-sha512
- ecdsa-384
- bliss-sha512

Authentication method to require from the remote (right) side. Acceptable values are pubkey for public key encryption (RSA/ECDSA/BLISS), psk for pre-shared key authentication, eap to require the use of the Extensible Authentication Protocol, and xauth for IKEv1 eXtended Authentication.

To require a trustchain public key strength for the remote side, specify the key type followed by the minimum strength in bits (for example ecdsa-384 or rsa-2048-ecdsa-256). To limit the acceptable set of hashing algorithms for trustchain validation, append hash algorithms to pubkey or a key strength definition (for example pubkey-sha256-sha512, rsa-2048-sha256-sha384-sha512, or rsa-2048-sha256-ecdsa-256-sha256-sha384).

Unless explicit IKEv2 signature constraints are configured (see below), such key types and hash algorithms are also applied as constraints against IKEv2 signature authentication schemes used by the remote side.

If both peers support RFC 7427 (Signature Authentication in IKEv2) specific hash algorithms to be used during IKEv2 authentication may be configured. The syntax is the same as above, but with ike: prefix. For example, with ike:pubkey-sha384-sha256 a public key signature scheme with either SHA-384 or SHA-256 would get used for authentication, in that order and depending on the hash algorithms supported by the peer. If no specific hash algorithms are configured, the default is to prefer an algorithm that matches or exceeds the strength of the signature key. If no constraints with ike: prefix are configured any signature scheme constraint (without ike: prefix) will also apply to IKEv2 authentication.

RSASSA-PSS signatures are supported. To use or require them, configure rsa/pss instead of rsa as in e.g. ike:rsa/pss-sha256. If pubkey or rsa constraints are configured, RSASSA-PSS signatures will only be used/accepted if enabled in strongswan.conf.

In the case of eap, an optional EAP method can be appended. Currently defined methods are eap-aka, eap-gtc, eap-md5, eap-mschapv2, eap-peap, eap-sim, eap-tls, eap-ttls, eap-dynamic, and eap-radius. Alternatively, IANA assigned EAP method numbers are accepted. Vendor specific EAP methods are defined in the form eap-type-vendor (e.g. eap-7-12345).

Signature and trust chain constraints for EAP-(T)TLS may be defined. To do so, append a colon to the EAP method, followed by the key type/size and hash algorithm as discussed above. For xauth, an XAuth authentication backend can be specified, such as xauth-generic or xauth-eap. If XAuth is used in leftauth, Hybrid authentication is used. For traditional XAuth authentication, define XAuth in leftauth2.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.2.1.1.9

8.1.5.2.2 cfgVpnIpsecGlobal

8.1.5.2.2.1 cfgVpnIpsecGlblVirtualTunnelInterface

Virtual Tunnel Interfaces Disabled or Enabled

Whether IPsec should use virtual tunnel interfaces (vti) or not.

Generally IPsec processing is based on policies. After regular route lookups are done, the OS kernel consults its SPD for a matching policy and if one is found that is associated with an IPsec SA, the packet is processed (e.g. encrypted and sent as ESP packet).

It is also possible to configure route-based VPNs. Here IPsec processing does not (only) depend on negotiated policies but may e.g. be controlled by routing packets to a specific interface.

Most of these approaches also allow easy capture of plaintext traffic, which, depending on the operating system, might not be that straight-forward with policy-based VPNs. Another advantage this approach is that the MTU can be specified for the tunneling devices allowing to fragment packets before tunneling them in case PMTUD does not work properly.

VTI devices act like a wrapper around existing IPsec policies. This means you can't just route arbitrary packets to a VTI device to get them tunneled, the established IPsec policies have to match too. However, you can negotiate 0.0.0.0/0 traffic selectors on both ends to allow tunneling anything that's routed via VTI device.

It's important to note that VTI tunnel devices are a local feature, no additional encapsulation (like with GRE) is added, so the other end does not have to be aware that VTI devices are used in addition to regular IPsec policies.

Note: When VTI is in use, the remote (`cfgVpnIpsecRight`) has to be specified as IP address. It is not possible to use an FQDN.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.2.2.1

8.1.5.2.2.2 cfgVpnIpsecGlblCustomOptions

Custom Global IPsec Options

Set to `none` when no additional options shall be added.

When setting multiple options, separate them with a semicolon `;`.

Example to enable aggressive mode for PSK with IKEv1:

- `charon.i_dont_care_about_security_and_use_aggressive_mode_psk=yes`

For a full list of all available options, please see: https://wiki.strongswan.org/projects/strongswan/wiki/Strongswanco

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.2.2.2

8.1.5.2.3 cfgVpnIpsecDebug

8.1.5.2.3.1 cfgVpnIpsecDbgAsn

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

Enumeration	silent (-1), basic (0), generic (1), detailed (2), raw (3), sensitive (4)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.2.3.1

8.1.5.2.3.2 cfgVpnIpsecDbgJob

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

Enumeration	silent (-1), basic (0), generic (1), detailed (2), raw (3), sensitive (4)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.2.3.10

8.1.5.2.3.3 cfgVpnIpsecDbgKnl

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

Enumeration	silent (-1), basic (0), generic (1), detailed (2), raw (3), sensitive (4)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.2.3.11

8.1.5.2.3.4 cfgVpnIpsecDbgLib

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

Enumeration	silent (-1), basic (0), generic (1), detailed (2), raw (3), sensitive (4)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.2.3.12

8.1.5.2.3.5 cfgVpnIpsecDbgMgr

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

Enumeration	silent (-1), basic (0), generic (1), detailed (2), raw (3), sensitive (4)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.2.3.13

8.1.5.2.3.6 cfgVpnIpsecDbgNet

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

Enumeration	silent (-1), basic (0), generic (1), detailed (2), raw (3), sensitive (4)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.2.3.14

8.1.5.2.3.7 cfgVpnIpsecDbgPts

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

Enumeration	silent (-1), basic (0), generic (1), detailed (2), raw (3), sensitive (4)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.2.3.15

8.1.5.2.3.8 cfgVpnIpsecDbgTls

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

Enumeration	silent (-1), basic (0), generic (1), detailed (2), raw (3), sensitive (4)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.2.3.16

8.1.5.2.3.9 cfgVpnIpsecDbgTnc

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

Enumeration	silent (-1), basic (0), generic (1), detailed (2), raw (3), sensitive (4)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.2.3.17

8.1.5.2.3.10 cfgVpnIpsecDbgCfg

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

Enumeration	silent (-1), basic (0), generic (1), detailed (2), raw (3), sensitive (4)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.2.3.2

8.1.5.2.3.11 cfgVpnIpsecDbgChd

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

Enumeration	silent (-1), basic (0), generic (1), detailed (2), raw (3), sensitive (4)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.2.3.3

8.1.5.2.3.12 cfgVpnIpsecDbgDmn

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

Enumeration	silent (-1), basic (0), generic (1), detailed (2), raw (3), sensitive (4)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.2.3.4

8.1.5.2.3.13 cfgVpnIpsecDbgEnc

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

Enumeration	silent (-1), basic (0), generic (1), detailed (2), raw (3), sensitive (4)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.2.3.5

8.1.5.2.3.14 cfgVpnIpsecDbgEsp

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

Enumeration	silent (-1), basic (0), generic (1), detailed (2), raw (3), sensitive (4)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.2.3.6

8.1.5.2.3.15 cfgVpnIpsecDbgIke

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

Enumeration	silent (-1), basic (0), generic (1), detailed (2), raw (3), sensitive (4)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.2.3.7

8.1.5.2.3.16 cfgVpnIpsecDbgImc

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

Enumeration	silent (-1), basic (0), generic (1), detailed (2), raw (3), sensitive (4)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.2.3.8

8.1.5.2.3.17 cfgVpnIpsecDbgImv

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

Enumeration	silent (-1), basic (0), generic (1), detailed (2), raw (3), sensitive (4)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.2.3.9

8.1.5.3 cfgVpnWireguard

8.1.5.3.1 cfgVpnWireguardTable

Wireguard Table

Range	0 - 15
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.5.1

8.1.5.3.1.1 cfgVpnWireguardTableEntry

Wireguard Table Entry

OID	1.3.6.1.4.1.16177.1.400.1.1.1003.5.1.1

cfgVpnWgIndex

Table Entry Index

Range	0 - 15
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.5.1.1.1

cfgVpnWgName

Name of the Wireguard Interface

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.5.1.1.2

cfgVpnWgListenPort

Wireguard Listen Port

Specify the port which is used by this wireguard instance.

Set this to a fixed value when expecting inbound connections. The official wireguard port is 51820.

A random port is used when set to 0.

Range	0 - 65535
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.5.1.1.3

cfgVpnWgPrivateKey

Wireguard Private Key

Base 64 encoded private key used by this wireguard instance. The public key in `cfgVpnWgPublicKey` is derived from this private key.

Will automatically generate a new private/public key-pair when set to `generate` or `not_yet_generated`.

May be generated on the CLI with the command `wg genkey`.

Examples:

- generate
- not_yet_generated
- 4CwLw8p8UGdv6baTN1dMxhxVq+m779vl2IjDpSFecW8=

Type	DisplayString
Range	8 - 44
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.5.1.1.4

cfgVpnWgPublicKey

Wireguard Public Key

Base 64 encoded public key provided by this wireguard instance. This key is derived from what is set in `cfgVpnWgPrivateKey`.

When the private key changes or is regenerated, the content of this field is updated during the apply.

Configure this public key on the remote peer(s).

Type	DisplayString
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.5.1.1.5

cfgVpnWgMark

Mark

Marks all frames sent by this wireguard instance with the value specified here. This mark may be matched in the `cfgRouteRuleTable` for policy routing.

Set to `-1` to disable setting a mark.

The maximum mark value is `4294967295`.

Type	DisplayString
Range	1 - 10
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.5.1.1.6

8.1.5.3.2 cfgVpnWireguardPeersTable

Wireguard Peers Table

Range	0 - 255
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.5.2

8.1.5.3.2.1 cfgVpnWireguardPeersTableEntry

Wireguard Peers Table Entry

OID	1.3.6.1.4.1.16177.1.400.1.1.1003.5.2.1

cfgVpnWgPIndex

Table Entry Index

Range	0 - 255
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.5.2.1.1

cfgVpnWgPEnabled

Wireguard Peer Disabled or Enabled

Disable or enabled this peer instance.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.5.2.1.2

cfgVpnWgPInstance

Wireguard Peer Instance

Specify a wireguard instance defined in `cfgVpnWireguardTable`. All peers with a matching instance are set up for the referenced instance.

Range	0 - 15
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.5.2.1.3

cfgVpnWgPPeer

Wireguard Peer Public Key

Specifys the remote peer by its public key. This Peer is considered disabled when set to `none`.

Example:

*Su05SN6WIje1PFlHMO8C2GmCzPp1R85ciwYAo6yvIhA=

Type	DisplayString
Range	4 - 44
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.5.2.1.4

cfgVpnWgPEndpoint

Wireguard Peer Endpoint

Specifies the remote end by IP and port.

Set to `0.0.0.0:0`, when the local peer accepts connections, but does not initiate by itself.

Examples:

- 192.168.1.20:51820
- 0.0.0.0:0

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.5.2.1.5

cfgVpnWgPAllowedIps

Wireguard Peer Allowed IPs

The Allowed IPs refers to the addresses inside the tunnel. It has two meanings:

In TX direction it acts as routing table. Frames with a destination matching the Allowed IPs are encrypted. All other frames that are routed to the interface but don't match the Allowed IPs are dropped.

In RX direction it acts as ACL. Only frames where the source matches the Allowed IPs are accepted. Everything else is dropped.

Multiple space and/or comma separated networks in CIDR notation may be specified.

When set to `none`, no frames will be sent nor received.

Examples:

- none
- 0.0.0.0/0
- 192.168.0.0/16, 172.16.0.0/12, 10.0.0.0/8

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.5.2.1.6

cfgVpnWgPPsk

Wireguard Peer Private Shared Key

Base 64 encoded privately shared key. Provides an additional layer of cryptography with a symmetric key for post-quantum resistance. Has negligible impact on performance. A separate key should be used for every peer.

Set to `none` when not used.

May be generated on the CLI with the command `wg genpsk`.

Examples:

- none
- F6wPakowIilChk4FRcHrAP+/jO5jdsQ7xphXi8UzG6Y=

Type	DisplayString
Range	4 - 44
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.5.2.1.7

cfgVpnWgPPersistentKeepalive

Wireguard Peer Persistent Keepalive

Interval in seconds to send a keepalive message to the peer.

Only set this, when connecting through NAT or a firewall blocking inbound connections. When not set to 0, a sane value is 25.

Range	0 - 65535
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.5.2.1.8

8.1.5.4 cfgVpnTunnelEndPoint

8.1.5.4.1 cfgVpnTunnelEndPointTable

Tunnel Endpoint Table

Range	0 - 15
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.6.1

8.1.5.4.1.1 cfgVpnTunnelEndPointTableEntry

Tunnel Endpoint Table Entry

OID	1.3.6.1.4.1.16177.1.400.1.1.1003.6.1.1

cfgVpnTepIndex

Table Entry Index

Range	0 - 15
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.6.1.1.1

cfgVpnTepVnid

Virtual Network ID

This parameter is active when `cfgVpnTepTunnelType` is set to **vxlan(2)**.

Specify the vxlan network id.

Range	0 - 16777215
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.6.1.1.11

cfgVpnTepDestinationPort

Destination Port

This parameter is active when `cfgVpnTepTunnelType` is set to **vxlan(2)**.

Specify the destination UDP port. At the same time this parameter defines on which port the local tunnel endpoint is listening for inbound vxlan frames.

Range	1 - 65535
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.6.1.1.12

cfgVpnTepName

Name of the Tunnel Endpoint Interface

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.6.1.1.2

cfgVpnTepTunnelType

Tunnel Endpoint Type

- **gre(0):** Allows to tunnel L3 frames. Can not be bridged.
- **gretap(1):** Allows to tunnel L2 frames. Can be bridged.
- **vxlan(2):** Allows to tunnel L2 frames. Can be bridged.

Enumeration	gre (0), gretap (1), vxlan (2)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.6.1.1.3

cfgVpnTepSource

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.

Type	IpAddress
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.6.1.1.4

cfgVpnTepDestination

Destination Address of the Tunnel

Encapsulated frames are sent to this address.

Type	IpAddress
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.6.1.1.5

cfgVpnTepTos

Outer TOS Field of the Tunnel

The outer IP header will have it's TOS field set to the value specified here.

To force the outer TOS header to an explicit value, set the field to a value between '00' and 'ff'.

The outer TOS header may inherit its TOS field from the inner IP header. To achieve this set to 'inherit'. Non-IP frames will have the value 00.

Usually, only the upper most 3 bits have a significant impact on prioritisation, thus it is recommended to preferably use the values: 00, 20, 40, 60, 80, a0, c0, e0

Examples:

- inherit
- 00
- a0
- ff

Type	DisplayString
Range	2 - 7
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.6.1.1.6

cfgVpnTepRxKeyId

RX Tunnel Key ID

This parameter is active when `cfgVpnTepTunnelType` is set to **gre(0)** or **gretap(1)**.

The key allows to run multiple tunnels between peers in parallel. The RX Key specifies which value is expected in the header of frames during reception.

This is a 32 bit number which may be specified directly (a number between 0 and 4294967295) or as an IP address-like dotted quad: '123.123.0.255'.

The value configured in this field should match the `cfgVpnTepTxKeyId` on the remote side.

Set to -1 to not expect a key.

Examples:

- -1
- 0
- 4000
- 100.0.0.1

Type	DisplayString
Range	1 - 15
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.6.1.1.7

cfgVpnTepTxKeyId

TX Tunnel Key ID

This parameter is active when `cfgVpnTepTunnelType` is set to **gre(0)** or **gretap(1)**.

The key allows to run multiple tunnels between peers in parallel. The TX Key specifies which value is set in the header of frames during transmission.

This is a 32 bit number which may be specified directly (a number between 0 and 4294967295) or as an IP address-like dotted quad: '123.123.0.255'.

The value configured in this field should match the `cfgVpnTepRxKeyId` on the remote side.

Set to -1 to not set a key.

Examples:

- -1

- 0
- 4000
- 100.0.0.1

Type	DisplayString
Range	1 - 15
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1003.6.1.1.8

8.1.6 cfgLdap

8.1.6.1 cfgLdapEnabled

Disable or Enable LDAP Authentication

Applies to AP and STA.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1005.1

8.1.6.2 cfgLdapAdminRoleDn

DN for Role Admin

Distinguished name for role admin.

Example:

- 'CN=net_admin,OU=Groups,OU=Company,DC=excompany,DC=ex'

Applies to AP and STA.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1005.10

8.1.6.3 cfgLdapMonitorRoleDn

DN for Role Monitor

Distinguished name for role monitor.

Example:

- 'CN=net_operator,OU=Groups,OU=Company,DC=excompany,DC=ex'

Applies to AP and STA.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1005.12

8.1.6.4 cfgLdapRequestTimeout

LDAP Request Timeout

Maximum time in seconds allowed for an LDAP request to take.

Applies to AP and STA.

Range	0 - 120
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1005.13

8.1.6.5 cfgLdapCrlExpiryExtension

CRL Validity Period Extension in Days

If set, the validity period of a CRL can be extended by the given amount of days.

- **0** no extension
- **1-1095** extension days
- **-1** extend to infinity => ignore CRL expiry

Applies to AP and STA.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1005.15

8.1.6.6 cfgLdapTlsControlParams

Bitfield to Control TLS Behavior

- **0x0** all validity checks will be performed
- **0x1** ignore certificate validity time
- **0x2** ignore ca certificate
- **0x4** ignore CRLs
- **0x8** ignore missing CRLs

Applies to AP and STA.

Range	0 - 15
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1005.16

8.1.6.7 cfgLdapTlsCiphers

OpenSSL Cipher String for LDAP

Specify which OpenSSL ciphers to use for the LDAP connection.

Please read the user manual and the OpenSSL documentation for a list of available ciphers and used syntax.

Set to 'none' to disable restriction.

Examples:

- ECDHE-RSA-AES256-GCM-SHA384:DHE-RSA-AES256-GCM-SHA384
- DHE-RSA-AES256-GCM-SHA384
- ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-SHA384
- none

Applies to AP and STA.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1005.17

8.1.6.8 cfgLdapUrl1

LDAP Server 1

Primary LDAP server name, ignored if set to '0.0.0.0'.

Applies to AP and STA.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1005.2

8.1.6.9 cfgLdapUrl2

LDAP Server 2

Secondary LDAP server name, ignored if set to '0.0.0.0'.

Applies to AP and STA.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1005.3

8.1.6.10 cfgLdapCaIds

LDAP CA Certificate IDs

Select the CAs to be used for LDAP server certificate validation. Multiple CAs can be referenced by writing the ids of the CAs as space and/or comma separated list.

Examples:

- `1, 3, 4`
- `1 3 4`

Applies to AP and STA.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1005.4

8.1.6.11 cfgLdapUserBaseDn

The Searchbase for LDAP User Base DN Retrieval

This is the starting point for the search in the LDAP database.

Using ldapsearch from openldap-utils, this corresponds to option '-b' (searchbase).

Example:

- 'dc=excompany, dc=ex'

Applies to AP and STA.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1005.5

8.1.6.12 cfgLdapAccessDn

The Bind-DN for LDAP User Search

This is the Distinguished Name (DN) to bind to the LDAP directory when searching for the user's DN.

Using ldapsearch from openldap-utils, this corresponds to option '-D'.

Example:

- 'cn=admin,ou=product accounts,dc=excompany,dc=ex'

Applies to AP and STA.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1005.6

8.1.6.13 cfgLdapAccessPassword

Simple Authentication Password for LDAP User Search

This is the password for simple authentication when searching for the user's DN.

Using ldapsearch from openldap-utils, this corresponds to option '-w'.

Applies to AP and STA.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1005.7

8.1.6.14 cfgLdapAccessFilter

Search Filter for LDAP User Search

Using ldapsearch from openldap-utils, this corresponds to argument 'filter'.

The string might have the following placeholder that is replaced with the according parameter: * '%USER%': username to retrieve role for

Example:

- `(sAMAccountName=%USER%)`

Applies to AP and STA.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1005.8

8.1.6.15 cfgLdapUserRoleAttribute

LDAP Attribute Name for User's Role Retrieval

Attribute name to be used to retrieve the user's role.

Example:

- `memberOf`

Applies to AP and STA.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1005.9

8.1.7 cfgRadsec

8.1.7.1 cfgRadsecProxyInstance

8.1.7.1.1 cfgRadsecProxyInstanceTable

RadsecProxy Instance Configuration

Range	0 - 255
OID	1.3.6.1.4.1.16177.1.400.1.1.1008.1.1

8.1.7.1.1.1 cfgRadsecProxyInstanceTableEntry

RadsecProxy Instance Entry

OID	1.3.6.1.4.1.16177.1.400.1.1.1008.1.1.1

cfgRadsecProxyInstIndex

RadsecProxy Instance Entry Index

Range	0 - 255
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.1008.1.1.1.1

cfgRadsecProxyInstEnabled

Disable or Enable Radsec Proxy Instance

Enable or disable a RadSec Proxy instance. The instance is only enabled if it has a valid reference to an enabled RadSec Proxy upstream server.

Applies to AP

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1008.1.1.1.2

cfgRadsecProxyInstInstanceId

RadsecProxy Instance ID

Reference to the upstream servers ID's in the `cfgRadsecProxyUpstreamServerTable` with the same value in `cfgRadsecProxyUpSrvInstanceId`.

Applies to AP

Range	0 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1008.1.1.1.3

cfgRadsecProxyInstLocalAddress

RadsecProxy Local Address

The local listen IP address of the RadSec Proxy. This could either be localhost `127.0.0.1/32` when a using enterprise encryption on this AP or a bind address where the RadSec Proxy receives authentication or accounting requests from other AP's in the same network.

For example `192.168.3.22/24` in CIDR notation to allow incoming requests from any AP in the same `/24` subnet.

Applies to AP

Type	DisplayString
Range	7 - 18
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1008.1.1.1.4

cfgRadsecProxyInstLocalAuthSrvPort

RadsecProxy Local Authentication Server Port

The local listening port for authentication requests. Set this entry to `-1` to disable it.

Applies to AP

Range	-1 - 65535
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1008.1.1.1.5

cfgRadsecProxyInstLocalAcctSrvPort

RadsecProxy Local Accounting Server Port

The local listening port for accounting requests. Set this entry to `-1` to disable it.

Applies to AP

Range	-1 - 65535
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1008.1.1.1.6

cfgRadsecProxyInstLocalSecret

RadsecProxy Local Server Secret

The password for incoming authentication or accounting requests. The same value is used for both protocols.

Applies to AP

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1008.1.1.1.7

8.1.7.2 cfgRadsecProxyUpstreamServer

8.1.7.2.1 cfgRadsecProxyUpstreamServerTable

RadSec Proxy Upstream Server Configuration

Range	0 - 255
OID	1.3.6.1.4.1.16177.1.400.1.1.1008.2.1

8.1.7.2.1.1 cfgRadsecProxyUpstreamServerTableEntry

RadSec Proxy Upstream Server Entry

OID	1.3.6.1.4.1.16177.1.400.1.1.1008.2.1.1

cfgRadsecProxyUpSrvIndex

RadSec Proxy Upstream Server Entry Index

Range	0 - 255
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.1008.2.1.1.1

cfgRadsecProxyUpSrvClientKeyId

RadSec Upstream Server Client Key ID

Reference to the private Client Key which should be used.

Applies to AP

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1008.2.1.1.10

cfgRadsecProxyUpSrvRealms

RadSec Upstream Server Realm Identification

A received Access-Request is checked against the defined realms. This is done by looking at the Username attribute in the request.

Example: username@westermo.com

The proxy will match against the blocks in the order they are specified, using the first match if any. If no realm matches, the proxy will simply ignore the request.

To accept all request, define ∗ as realm. Multiple realms can be defined in a comma and/or space separated list. If the realm name starts with a / the name is treated as a regular expression. It may optionally end with a '/'.

Examples:

- **any:** *
- **single domain:** westermo.com
- **regexp:** /@westermo.com$/

Applies to AP

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1008.2.1.1.11

cfgRadsecProxyUpSrvEnabled

Disable or Enable RadSec Proxy Upstream Server

Enable or disable a RadSec Proxy upstream server. The upstream server is only enabled if it has a valid reference to an enabled RadSec Proxy instance.

Applies to AP

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1008.2.1.1.2

cfgRadsecProxyUpSrvInstanceId

RadSec Proxy Upstream Server Instance ID

Reference to the instance ID's in the `cfgRadsecProxyInstanceTable` with the same value in `cfgRadsecProxyInstInstanceId`.

Applies to AP

Range	0 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1008.2.1.1.3

cfgRadsecProxyUpSrvAuthAddress

RadSec Upstream Server Authentication Address

IP address or FQDN of the RadSec server for forwarding authentication requests.

Set the value to `0.0.0.0` to disable the authentication server and to use it only for accounting.

Applies to AP

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1008.2.1.1.4

cfgRadsecProxyUpSrvAcctAddress

RadSec Upstream Server Accounting Address

IP address or FQDN of the RadSec server for forwarding accounting requests.

Set the value to `0.0.0.0` to disable the accounting server and to use it only for authentication.

Applies to AP

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1008.2.1.1.5

cfgRadsecProxyUpSrvPort

RadSec Upstream Server Port

The default RadSec server authentication port is 2083.

Applies to AP

Range	1 - 65535
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1008.2.1.1.6

cfgRadsecProxyUpSrvSecret

RadSec Upstream Server Secret

Password to connect to the specified RadSec server. The shared secret is used to allow the authenticator to communicate with the server, it is not used to authenticate clients.

Applies to AP

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1008.2.1.1.7

cfgRadsecProxyUpSrvCaCertId

RadSec Upstream Server Certificate Authority (CA) IDs

Reference to the CA certificate. If more CA needs to be referenced, they must be combined in the same file.

Applies to AP

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1008.2.1.1.8

cfgRadsecProxyUpSrvClientCertId

RadSec Upstream Server Client Certificate ID

Reference to the client certificate

Applies to AP

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1008.2.1.1.9

8.1.8 cfgCellular

8.1.8.1 cfgCellSimTable

****OBSOLETE:** SIM Parameter Table**

This parameter is obsolete and has been replaced with `cfgCellSimSlotTable` and `cfgCellDefaultBearerTab`

Range	0 - 0
OID	1.3.6.1.4.1.16177.1.400.1.1.101.1

8.1.8.1.1 cfgCellSimTableEntry

****OBSOLETE:** SIM Parameter Table Entry**

This parameter is obsolete and has been replaced with `cfgCellSimSlotTable` and `cfgCellDefaultBearerTab`

OID	1.3.6.1.4.1.16177.1.400.1.1.101.1.1

8.1.8.1.1.1 cfgCellSimIndex

****OBSOLETE:** Table Entry Index**

This parameter is obsolete and has been replaced with `cfgCellSimSlotTable` and `cfgCellDefaultBearerTab`

Range	0 - 0
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.101.1.1.1

8.1.8.1.1.2 cfgCellSimSlot1

****OBSOLETE:** SIM Slot 1**

This parameter is obsolete and has been replaced with `cfgCellSimSlotTable` and `cfgCellDefaultBearerTab`

Range	-1 - -1
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.101.1.1.2

8.1.8.1.1.3 cfgCellSimSlot2

****OBSOLETE:** SIM Slot 2**

This parameter is obsolete and has been replaced with `cfgCellSimSlotTable` and `cfgCellDefaultBearerTab`

Range	-1 - -1
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.101.1.1.3

8.1.8.1.1.4 cfgCellSimPrimarySlot

****OBSOLETE:** Primary SIM Slot**

This parameter is obsolete and has been replaced with `cfgCellSimSlotTable` and `cfgCellDefaultBearerTab`

Enumeration	obsolete (-1)
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.101.1.1.4

8.1.8.1.1.5 cfgCellSimUnlockTimeout

****OBSOLETE:** SIM Unlock Timeout**

This parameter is obsolete and has been replaced with `cfgCellSimSlotTable` and `cfgCellDefaultBearerTab`

Range	-1 - -1
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.101.1.1.5

8.1.8.2 cfgCellSimProfileTable

****OBSOLETE:** SIM Profiles**

This parameter is obsolete and has been replaced with `cfgCellSimSlotTable` and `cfgCellDefaultBearerTab`

Range	0 - 9
OID	1.3.6.1.4.1.16177.1.400.1.1.101.2

8.1.8.2.1 cfgCellSimProfileTableEntry

****OBSOLETE:** SIM Profile**

This parameter is obsolete and has been replaced with `cfgCellSimSlotTable` and `cfgCellDefaultBearerTab`

OID	1.3.6.1.4.1.16177.1.400.1.1.101.2.1

8.1.8.2.1.1 cfgCellSimProfileIndex

****OBSOLETE:** Table Entry Index**

This parameter is obsolete and has been replaced with `cfgCellSimSlotTable` and `cfgCellDefaultBearerTab`

Range	0 - 9
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.101.2.1.1

8.1.8.2.1.2 cfgCellSimProfileApn

****OBSOLETE:** Access Point Name**

This parameter is obsolete and has been replaced with `cfgCellSimSlotTable` and `cfgCellDefaultBearerTab`

Type	DisplayString
Range	1 - 255
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.101.2.1.2

8.1.8.2.1.3 cfgCellSimProfileUsername

****OBSOLETE:** Username**

This parameter is obsolete and has been replaced with `cfgCellSimSlotTable` and `cfgCellDefaultBearerTab`

Type	DisplayString
Range	1 - 255
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.101.2.1.3

8.1.8.2.1.4 cfgCellSimProfilePassword

****OBSOLETE:** Password**

This parameter is obsolete and has been replaced with `cfgCellSimSlotTable` and `cfgCellDefaultBearerTab`

Type	DisplayString
Range	1 - 255
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.101.2.1.4

8.1.8.2.1.5 cfgCellSimProfilePinEnabled

****OBSOLETE:** PIN Authentication Disabled or Enabled**

This parameter is obsolete and has been replaced with `cfgCellSimSlotTable` and `cfgCellDefaultBearerTab`

Enumeration	obsolete (-1)
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.101.2.1.5

8.1.8.2.1.6 cfgCellSimProfilePin

****OBSOLETE:** PIN of the SIM Card**

This parameter is obsolete and has been replaced with `cfgCellSimSlotTable` and `cfgCellDefaultBearerTab`

Type	DisplayString
Range	1 - 255
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.101.2.1.6

8.1.8.2.1.7 cfgCellSimProfileAuthType

****OBSOLETE:** Authentication Type**

This parameter is obsolete and has been replaced with `cfgCellSimSlotTable` and `cfgCellDefaultBearerTab`

Enumeration	obsolete (-1)
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.101.2.1.7

8.1.8.2.1.8 cfgCellSimProfileRoaming

****OBSOLETE:** Roaming Disabled or Enabled**

This parameter is obsolete and has been replaced with `cfgCellSimSlotTable` and `cfgCellDefaultBearerTab`

Enumeration	obsolete (-1)
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.101.2.1.8

8.1.8.3 cfgCellConnectionManagement

8.1.8.3.1 cfgCellConnMgmtSimRotationEnabled

SIM Rotation Disabled or Enabled

Set to **enabled(1)** to switch between primary and secondary SIM slot when a loss of connection is detected in use of the monitoring functionality `cfgCellConnMgmtMonMode`.

Only enabled SIM slots `cfgCellSimSlotEnabled` are taken into account.

Applies to cellular products only.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.101.3.1

8.1.8.3.2 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

Note: If cfgCellConnMgmtSimRotationEnabled and two SIM slots are enabled, the monitor event will trigger a SIM rotation between the two slots. If only one SIM slot is enabled or cfgCellConnMgmtSimRotation is disabled, the monitor event will trigger a re-connection with the same SIM slot.

Applies to cellular products only.

Enumeration	signal (0), remoteHosts (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.101.3.2

8.1.8.3.3 cfgCellConnMgmtMonPeriod

Monitor Period

The monitor period defines the time in seconds between two consecutive evaluations of the cellular connection with regard to the cfgCellConnMgmtMonMode.

Applies to cellular products only.

Range	1 - 86400
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.101.3.3

8.1.8.3.4 cfgCellConnMgmtMonCount

Monitor Count

The monitor count defines the needed amount of consecutive failed connection tests, with regard to the cfgCellConnMgmtMonMode, before a connection is considered down.

Applies to cellular products only.

Range	1 - 10
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.101.3.4

8.1.8.3.5 cfgCellConnMgmtMonRemoteTable

Monitored Remotes

All hosts listed in this table are used by the **network(1)** mode, defined in `cfgCellConnMgmtMonMode`. If the connection managmement 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.

Range	0 - 3
OID	1.3.6.1.4.1.16177.1.400.1.1.101.3.5

8.1.8.3.5.1 cfgCellConnMgmtMonRemoteTableEntry

Monitored Remote

Applies to cellular products only.

OID	1.3.6.1.4.1.16177.1.400.1.1.101.3.5.1

cfgCellConnMgmtMonRemoteIndex

Table Entry Index

Applies to cellular products only.

Range	0 - 3
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.101.3.5.1.1

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.

Enumeration	disabled (0), icmp (1), tcp (2)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.101.3.5.1.2

cfgCellConnMgmtMonRemoteAddress

Remote Address

The address of a remote host can either be defined by an IP address or a host name. If `cfgCellConnMgmtMonRemo` is set to **tcp(2)**, an additional port number must be specifed. 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.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.101.3.5.1.3

8.1.8.4 cfgCellDbgTable

Cellular Debug Parameters

Range	0 - 0
OID	1.3.6.1.4.1.16177.1.400.1.1.101.4

8.1.8.4.1 cfgCellDbgTableEntry

Cellular Debug Parameters Entry

OID	1.3.6.1.4.1.16177.1.400.1.1.101.4.1

8.1.8.4.1.1 cfgCellDbgIndex

Table Entry Index

Range	0 - 0
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.101.4.1.1

8.1.8.4.1.2 cfgCellDbgSignal

Disable or Enable Signal Level Trap

Persistent default value to disable or enable the signal level trap 820.

Applies to cellular products only.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.101.4.1.2

8.1.8.4.1.3 cfgCellDbgConnection

Disable or Enable Connection State Trap

Persistent default value to disable or enable the connection state trap 821.

Applies to cellular products only.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.101.4.1.3

8.1.8.5 cfgCellSimSlotTable

SIM Slot Table

Applies to cellular products only.

Range	0 - 1
OID	1.3.6.1.4.1.16177.1.400.1.1.101.5

8.1.8.5.1 cfgCellSimSlotTableEntry

SIM Slot Entry

Applies to cellular products only.

OID	1.3.6.1.4.1.16177.1.400.1.1.101.5.1

8.1.8.5.1.1 cfgCellSimSlotIndex

Table Entry Index

Range	0 - 1
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.101.5.1.1

8.1.8.5.1.2 cfgCellSimSlotName

Name of the SIM Slot

The name of the pysical SIM slot. At least one SIM slot must be assigned to the default bearer configuration `cfgCellDefaultBearerSimSlots`.

Applies to cellular products only.

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.1.101.5.1.2

8.1.8.5.1.3 cfgCellSimSlotEnabled

Enable SIM Slot

Enable the physical SIM slot if a SIM card is inserted. SIM slots without SIM card should be disabled when SIM rotation `cfgCellConnMgmtSimRotationEnabled` is enabled.

Applies to cellular products only.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.101.5.1.3

8.1.8.5.1.4 cfgCellSimSlotPinEnabled

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.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.101.5.1.4

8.1.8.5.1.5 cfgCellSimSlotPin

PIN of the SIM Card

The PIN is ignored, when PIN authentication is disabled, see `cfgCellSimSlotPinEnabled`.

Applies to cellular products only.

Type	DisplayString
Range	4 - 4
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.101.5.1.5

8.1.8.5.1.6 cfgCellSimSlotPriority

SIM Slot Priority

A lower value means a higher priority.

If two SIM slots have the same priority, the SIM slot with the lower index is preferred.

Applies to cellular products only.

Range	0 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.101.5.1.6

8.1.8.5.1.7 cfgCellSimSlotUnlockTimeout

SIM Unlock Timeout

The time in milliseconds how long the unlocking process waits until the SIM card is ready to be unlocked. Increase the timeout if older SIM cards cannot be unlocked. A high timeout affects the performance of SIM rotation.

Note: Changes become effective after restarting the device.

Applies to cellular products only.

Range	300 - 30000
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.101.5.1.7

8.1.8.6 cfgCellDefaultBearerTable

Default Bearer Table

Each cellular interface defined in `cfgNetWwanTable` enables a default bearer configuration.

Applies to cellular products only.

Range	0 - 7
OID	1.3.6.1.4.1.16177.1.400.1.1.101.6

8.1.8.6.1 cfgCellDefaultBearerTableEntry

Default Bearer Entry

Applies to cellular products only.

OID	1.3.6.1.4.1.16177.1.400.1.1.101.6.1

8.1.8.6.1.1 cfgCellDefaultBearerIndex

Table Entry Index

Range	0 - 7
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.101.6.1.1

8.1.8.6.1.2 cfgCellDefaultBearerSimSlots

Default Bearer SIM Slot Selection

A default bearer must refer to at least one physical SIM slot. The name of the SIM slot `cfgCellSimSlotName` is entered as a comma and/or space separated list.

Examples:

- slot1
- slot2
- slot1, slot2

Applies to cellular products only.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.101.6.1.2

8.1.8.6.1.3 cfgCellDefaultBearerApn

Default Bearer Access Point Name

auto No dedicated APN defined

If the APN is unknown, `auto` will establish a connection with the preferred default bearer of the cellular provider. This functionality must be supported by the cellular provider, who can also enforce a dedicated APN. Only one default bearer can be set to `auto` per SIM slot. If more than one WWAN interfaces are enabled, explicit APN's must be defined. FOTA selects the cellular provider default APN to access the update server.

Dedicated APN defined

If the APN name is known or if more than one default bearer shall be defined per SIM slot, a dedicated APN must be entered. Multi default bearer functionality must be supported by the cellular provider. FOTA selects the APN of the first WWAN interface `wwan0` to access the update server.

Applies to cellular products only.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.101.6.1.3

8.1.8.6.1.4 cfgCellDefaultBearerUsername

Default Bearer Username

If the service provider requries authentication for the selected default bearer, the username shall be specified by this entry. If no authentication type is selected, see `cfgCellDefaultBearerAuthType`, this entry is ignored.

Applies to cellular products only.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.101.6.1.4

8.1.8.6.1.5 cfgCellDefaultBearerPassword

Default Bearer Password

Set the password for the user defined in `cfgCellDefaultBearerUsername`.

Applies to cellular products only.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.101.6.1.5

8.1.8.6.1.6 cfgCellDefaultBearerAuthType

Default Bearer Authentication Type

Select the authentication type for the selected default bearer. 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.

Enumeration	none (0), pap (1), chap (2), both (3)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.101.6.1.6

8.1.8.6.1.7 cfgCellDefaultBearerRoaming

Default Bearer Roaming Disabled or Enabled

Set to **enabled(1)** to roam to other available cellular networks outside the range of the home network. The SIM card and network provider must support roaming.

Note: Activated roaming may incur additional costs!

Applies to cellular products only.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.101.6.1.7

8.1.8.7 cfgCellDeviceTable

Cellular Device Table

Applies to cellular products only.

Range	0 - 0
OID	1.3.6.1.4.1.16177.1.400.1.1.101.7

8.1.8.7.1 cfgCellDeviceTableEntry

Cellular Device Table Entry

Applies to cellular products only.

OID	1.3.6.1.4.1.16177.1.400.1.1.101.7.1

8.1.8.7.1.1 cfgCellDeviceIndex

Table Entry Index

Range	0 - 0
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.101.7.1.1

8.1.8.7.1.2 cfgCellDevice5gMode

5G Mode Selection

Limitation to one of the supported 5G modes.

- **auto(0)**: 5G-SA and 5G-NSA
- **nsa5g(1)**: 5G-NSA only
- **sa5g(2)**: 5G-SA only

Applies to cellular products only.

Enumeration	auto (0), nsa5g (1), sa5g (2)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.101.7.1.10

8.1.8.7.1.3 cfgCellDeviceName

Name of the Cellular Device

Applies to cellular products only.

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.1.101.7.1.2

8.1.8.7.1.4 cfgCellDeviceSearchMode

Network Search Mode

Bitmask of enabled scan modes. The search order is prioritised according to the latest technology, 5G (if available), LTE and then WCDMA.

- 0x00 (0) = AUTO (all enabled)
- 0x01 (1) = WCDMA
- 0x02 (2) = LTE
- 0x04 (4) = 5G

Example:

- Limit to LTE and 5G: 2 + 4 = 6

Applies to cellular products only.

Range	0 - 7
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.101.7.1.3

8.1.8.7.1.5 cfgCellDeviceOperatorMode

Operator Selection Mode

Configuration of operator selection with reference to the Service Provider Name (SPN) `cfgCellDeviceOperatorS`

- **auto(0)**: Automatic mode, `cfgCellDeviceOperatorSpn` is ignored.
- **fix(1)**: Manual operator selection according to `cfgCellDeviceOperatorSpn`.
- **fixWithFallbackAuto(2)**: Manual operator selection according to `cfgCellDeviceOperatorSpn` with fallback to **auto** if operator is not found.

Note: In **fix(1)** mode, only one service provider is supported. If multiple SIM cards are used, the same service provider must be available for all active SIM cards. If the service provider name is not defined or is incorrect, the modem cannot connect to the mobile network and remote access is lost.

Applies to cellular products only.

Enumeration	auto (0), fix (1), fixWithFallbackAuto (2)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.101.7.1.4

8.1.8.7.1.6 cfgCellDeviceOperatorSpn

Operator Selection Service Provider Name (SPN)

Name of the operator if the selection mode `cfgCellDeviceOperatorMode` is set to **fix(1)** or **fixWithFallbackAuto(2)**.

Long format alphanumeric up to 16 characters. The string of the currently connected provider can be read from `swCellServiceName`.

Note: If the service provider name is not defined or is incorrect, the modem cannot connect to the mobile network and remote access is lost.

Applies to cellular products only.

Type	DisplayString
Range	1 - 16
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.101.7.1.5

8.1.8.7.1.7 cfgCellDeviceBandsWcdma

WCDMA Band Configuration

A conclusive space and/or comma separated list of all WCDMA bands to be used. The value `-1` disables the band restriction, all available bands are enabled.

The available bands depend on the cellular module, please ask support for more information.

- **B1 WCDMA 2100**: 1
- **B2 WCDMA 1900**: 2
- **B3 WCDMA 1800**: 3
- **B4 WCDMA 1700**: 4
- **. . .**
- **B9 WCDMA Japan 1700**: 9
- **B19 WCDMA Japan 850**: 19

Example:

- 1,2,5,6

Applies to cellular products only.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.101.7.1.6

8.1.8.7.1.8 cfgCellDeviceBandsLte

LTE Band Configuration

A conclusive space and/or comma separated list of all LTE bands to be used. The value `-1` disables the band restriction, all available bands are enabled.

The available bands depend on the cellular module, please ask support for more information.

- **B1 LTE**: 1
- **B2 LTE**: 2
- **B3 LTE**: 3
- **B4 LTE**: 4
- **B5 LTE**: 5
- **. . .**
- **B71 LTE**: 71

Example:

- 3,7,20,26

Applies to cellular products only.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.101.7.1.7

8.1.8.7.1.9 cfgCellDeviceBands5gnsa

5G-NSA Band Configuration

A conclusive space and/or comma separated list of all 5G-NSA bands to be used. The value `-1` disables the band restriction, all available bands are enabled.

The available bands depend on the cellular module, please ask support for more information.

- **n1 5G-NSA**: 1
- **n2 5G-NSA**: 2
- **n3 5G-NSA**: 3
- **n5 5G-NSA**: 5
- **n8 5G-NSA**: 8
- **...**
- **n261 5G-NSA**: 261

Example:

- 1,12,48,66

Applies to cellular products only.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.101.7.1.8

8.1.8.7.1.10 cfgCellDeviceBands5gsa

5G-SA Band Configuration

A conclusive space and/or comma separated list of all 5G-SA bands to be used. The value `-1` disables the band restriction, all available bands are enabled.

The available bands depend on the cellular module, please ask support for more information.

- **n1 5G-SA**: 1
- **n2 5G-SA**: 2
- **n3 5G-SA**: 3
- **...**
- **n77 5G-SA**: 77
- **n78 5G-SA**: 78
- **n79 5G-SA**: 79

Example:

- 78

Applies to cellular products only.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.101.7.1.9

8.1.9 cfgStunnel

8.1.9.1 cfgStunnelGlobal

8.1.9.1.1 cfgStunnelGlobalEnabled

TECHPREVIEW: Enable stunnel daemon

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1010.1.1

8.1.9.1.2 cfgStunnelGlobalLogLevel

TECHPREVIEW: Log level for stunnel daemon.

Default is `notice(5)`.

Enumeration	emerg (0), alert (1), crit (2), err (3), warning (4), notice (5), info (6), debug (7)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1010.1.4

8.1.9.2 cfgStunnelInstanceTable

Stunnel Instance Configuration

Range	0 - 7
OID	1.3.6.1.4.1.16177.1.400.1.1.1010.2

8.1.9.2.1 cfgStunnelInstanceTableEntry

Stunnel Instance Entry

OID	1.3.6.1.4.1.16177.1.400.1.1.1010.2.1

8.1.9.2.1.1 cfgStunnelInstanceIndex

Table Entry Index

Range	0 - 7
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.1010.2.1.1

8.1.9.2.1.2 cfgStunnelInstListenAddress

TECHPREVIEW: Stunnel Instance Listen Address

If only port is supplied, the instance is accessible from all local IP addresses.

Corresponds to stunnel's `accept` config parameter.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1010.2.1.10

8.1.9.2.1.3 cfgStunnelInstDestinationAddress

TECHPREVIEW: Stunnel Instance Destination Address

Hostname/IP-address and/or port to which this stunnel instance connects.

If only port is supplied, the service to which stunnel connects needs to be running on localhost.

Corresponds to stunnel's `connect` config parameter.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1010.2.1.15

8.1.9.2.1.4 cfgStunnelInstEnabled

TECHPREVIEW: Enable this server

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1010.2.1.2

8.1.9.2.1.5 cfgStunnelInstTlsCertId

TECHPREVIEW: Stunnel Instance Certificate ID

Reference ID of certificate in Cert-Store to be used as TLS end-node/leaf certificate. Corresponds to certificate passed in stunnel's `cert` config parameter.

If `cfgStunnelInstMode` is `server(0)`, this param is required, otherwise, it is only needed if the server requires client authentication.

Depending on the client's configuration, the entire certificate chain must be available. Additionally required certificates can be passed via `cfgStunnelInstTlsCaIds`.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1010.2.1.20

8.1.9.2.1.6 cfgStunnelInstTlsCaIds

TECHPREVIEW: Stunnel Instance CA Certificate IDs

Set to `-1` to use stunnel without CA certificate. Multiple certificates may be referenced by writing the ids of the certificates as space and/or comma separated list.

The configured CA certificates will be concatenated to `cfgStunnelInstTlsCertId` and passed to stunnel's `cert` config parameter.

Examples:

- `-1`
- `12`
- `1, 3, 4`
- `1 3 4`

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1010.2.1.21

8.1.9.2.1.7 cfgStunnelInstVerificationMode

TECHPREVIEW: Stunnel Instance Verification Mode

Defines whether peers must authenticate themselves.

Default is `none(0)`. For all other values, the corresponding certificate(s) need to be specified in `cfgStunnelInstTlsCaIds`.

This parameter maps to the boolean `verifyPeer` and `verifyChain` in the stunnel config file.

- **none(0)** (default): Instance accepts all connections.
- **peer(1)**: The peer certificate must be in a locally installed certificate file.
- **chain(2)**: Verify the peer's certificate chain. I.e. peer's certificate needs to be signed by trusted root CA.
- **both(3)**: **peer(1)** and **chain(2)** apply at the same time.

Enumeration	none (0), peer (1), chain (2), both (3)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1010.2.1.22

8.1.9.2.1.8 cfgStunnelInstTlsTrustAnchorCerts

TECHPREVIEW: Stunnel Instance TLS Trust Anchor Certificates

Set to `-1` to use without trust anchor certificates. Multiple certificates may be referenced by writing the ids of the certificates as space and/or comma separated list.

Different certificates need to be supplied depending on the value of `cfgStunnelInstVerificationMode`:

- **none(0)**: No certificates are needed / loaded.
- **peer(1)**: Peer leaf certificate(s) must be supplied.
- **chain(2)**: Entire certificate chain of peer's signatory entity must be supplied (e.g. intermediate + root CA).
- **both(3)**: Both must be supplied.

The concatenated certificates will be passed in stunnel's `CAfile` configuration parameter.

Examples:

- `-1`
- `12`
- `1, 3, 4`
- `1 3 4`

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1010.2.1.23

8.1.9.2.1.9 cfgStunnelInstTlsPeerIdentities

TECHPREVIEW: Stunnel Instance TLS Peer Identities

List of IP addresses and/or hostnames, which the peer certificate's CN or SAN must contain. Check is only performed if `cfgStunnelInstVerificationMode` is nonzero.

Set to `-1` to not verify peer identities. Multiple hostnames and/or IP addresses may be supplied by writing them as a space and/or comma separated list.

When configuring a client in a regular PKI setup, it is strongly recommended to supply the server host name and/or IP address corresponding to certificate CN or SAN.

Maps to stunnel's `checkHost` and `checkIP` configuration parameters.

Examples:

- `-1`
- `example.com`
- `192.168.1.55`
- `some.example.com 192.168.1.32`
- `192.168.1.32, example.com, westermo.com`

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1010.2.1.24

8.1.9.2.1.10 cfgStunnelInstMode

TECHPREVIEW: Stunnel Instance Mode

- **server(0)** Instance acts as TLS server, may accept connections from multiple clients.
- **client(1)** Instance acts as TLS client, may connect to one server at a time.

Enumeration	server (0), client (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.1010.2.1.3

8.1.10 cfgLogging

8.1.10.1 cfgLogRemote

8.1.10.1.1 cfgLogRemoteTable

List of Syslog Destinations

Range	0 - 3
OID	1.3.6.1.4.1.16177.1.400.1.1.11.2.1

8.1.10.1.1.1 cfgLogRemoteTableEntry

List of Syslog Destinations

OID	1.3.6.1.4.1.16177.1.400.1.1.11.2.1.1

cfgLogRemoteIndex

Table Entry Index

Range	0 - 3
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.11.2.1.1.1

cfgLogRemoteCrlExpiryExtension

CRL Validity Period Extension in Days

If set, the validity period of a CRL can be extended by the given amount of days.

- **0** no extension
- **1-1095** extension days
- **-1** extend to infinity => ignore CRL expiry

Range	-1 - 1095
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.11.2.1.1.10

cfgLogRemoteTlsControlParams

Bitfield to Control TLS Behavior

When the upload server is to be accessed via certificates, this setting is used to control the TLS behaviour.

Supported bits:

- 0x0 all validity checks will be performed
- 0x1 ignore certificate validity time
- 0x2 ignore ca certificate
- 0x4 ignore CRLs
- 0x8 ignore missing CRLs

Range	0 - 15
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.11.2.1.1.11

cfgLogRemoteCiphers

OpenSSL Cipher String for Remote Syslog

This is an OpenSSL specific configuration option for configuring the default cipher.

Please read the user manual and the OpenSSL documentation for a list of available ciphers and used syntax.

Used only if `cfgSyslogRemoteProtocol` is **tcptls(2)**.

Example:

- ECDHE-RSA-AES128-GCM-SHA256
- DHE-RSA-AES128-GCM-SHA256
- ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-SHA384
- DEFAULT:!EXP:!LOW

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.11.2.1.1.12

cfgLogRemoteEnabled

Syslog Instance Enabled or Disabled

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.11.2.1.1.2

cfgLogRemoteLevel

Log only messages with equal or higher priority than prio N (0-7).

This parameter is intentionally fixed to 7.

Enumeration	emergency (0), alert (1), critical (2), error (3), warning (4), notice (5), info (6), debug (7)
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.1.11.2.1.1.3

cfgLogRemoteProtocol

Protocol to Send Log Messages

The udp(0) protocol complies with the syslog protocol according to RFC5424.

The tcptls(2) protocol complies with the secure syslog protocol according to RFC5425.

Enumeration	udp (0), tcp (1), tcptls (2)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.11.2.1.1.4

cfgLogRemoteIp

Remote IP Address

The IP Address to which syslog messages are sent to.

Type	IpAddress
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.11.2.1.1.5

cfgLogRemotePort

Remote Port

The remote port to which syslog messages are sent.

When set to `-1` uses the assigned port according to the protocol that is selected in `cfgLogRemoteProtocol`:

udp(0) Port 514 **tcp(1)** Port 514 **tcptls(2)** Port 6514

Range	-1 - 65535
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.11.2.1.1.6

cfgLogRemoteType

Bitfield to Control Remote Syslog Type

- **0x00** - no remote syslog
- **0x01** - standard syslog
- **0x02** - security syslog
- **0x04** - commissioning syslog

Range	0 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.11.2.1.1.7

cfgLogRemoteCaIds

CA IDs

Reference to the CA IDs to be used.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.11.2.1.1.8

cfgLogRemoteCertId

Certificate ID

Reference to the Certificate ID to be used.

Range	-1 - 1000
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.11.2.1.1.9

8.1.11 cfgSnmp

8.1.11.1 cfgSnmpd

8.1.11.1.1 cfgSnmpdLocation

SNMP System Location

A string to describe the location of the device.

This value may be read via `RFC1213-MIB::sysLocation.0`

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.12.1.1

8.1.11.1.2 cfgSnmpdCommunity

8.1.11.1.2.1 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.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.12.1.100.1

8.1.11.1.2.2 cfgSnmpdComMaintainer

****OBSOLETE:** Password for the maintainer.**

Applies to AP and STA.

Type	DisplayString
Range	1 - 255
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.12.1.100.2

8.1.11.1.2.3 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.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.12.1.100.3

8.1.11.1.3 cfgSnmpdContact

SNMP Contact

A string to describe the responsible person of the device.

This value may be read via `RFC1213-MIB::sysContact.0`

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.12.1.2

8.1.11.1.4 cfgSnmpdVersion

SNMP Version

Set to **v2c(0)** to use SNMP v2 with the simple community-based security scheme.

Set to **v3usm(1)** to use the User-based Security Model (USM). USM is the default security module for SNMPv3, with the authentication type specified in `cfgSnmpdAuthType` and the privacy protocol defined in `cfgSnmpdPrivType`.

Setting **v3usm(1)** disables access to the device via SNMPv2.

Please refer to the user guide for more information about the access rights of the predefined users.

When using **v3usm(1)**, the unique Engine ID used by the SNMP Agent for SNMPv3 is built from the MAC address of `eth0` and has the following format:

`0x80001F8803<MAC(6 octets)>`

Example:

- Engine ID: `0x80001F880300145A035042`

Enumeration	v2c (0), v3usm (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.12.1.3

8.1.11.1.5 cfgSnmpdName

SNMP Name

An administratively-assigned name for this managed node. By convention, this is the node's FQDN (Fully Qualified Domain Name).

This value may be read via `RFC1213-MIB::sysName.0`

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.12.1.4

8.1.11.1.6 cfgSnmpdEnabled

Disable or Enable the SNMP Agent

Disabling the SNMP Agent does not disable SNMP traps. See `cfgSnmpTrapEnabled` to disable SNMP traps.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.12.1.5

8.1.11.1.7 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 ommited. Multiple space and/or comma separated tuples are allowed.

Supported tranport 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

For more information see http://www.net-snmp.org/docs/man/snmpd.examples.html#lbAE

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.12.1.6

8.1.11.1.8 cfgSnmpdAuthType

Authentication Type for SNMPv3

This parameter is used when `cfgSnmpdVersion` is set to **v3usm(1)**.

Enumeration	sha1 (2), sha384 (5), sha512 (6)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.12.1.7

8.1.11.1.9 cfgSnmpdPrivType

Encryption Type for SNMPv3

This parameter is used when `cfgSnmpdVersion` is set to **v3usm(1)**.

Enumeration	aes128 (2), aes256 (4)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.12.1.8

8.1.11.2 cfgSnmpTrap

8.1.11.2.1 cfgSnmpTrapEnabled

Disable or Enable SNMP Notifications

This parameter does not configure operation of the SNMP Agent which may be enabled or disabled via `cfgSnmpdEnabled`.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.12.10.1

8.1.11.2.2 cfgSnmpTrapAuthProtocol

SNMP Authentication Protocol

This parameter is active when `cfgSnmpTrapVersion` is set to **v3usm(2)**.

Set the authentication protocol used for authenticated SNMPv3 notifications.

Enumeration	none (0), sha1 (2), sha384 (5), sha512 (6)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.12.10.10

8.1.11.2.3 cfgSnmpTrapAuthPassword

Authentication Password

This parameter is active when `cfgSnmpTrapVersion` is set to **v3usm(2)**.

Set the authentication password used for authenticated SNMPv3 notifications.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.12.10.11

8.1.11.2.4 cfgSnmpTrapPrivProtocol

SNMP Privacy Protocol

This parameter is active when `cfgSnmpTrapVersion` is set to **v3usm(2)**.

Set the privacy protocol used for authenticated SNMPv3 notifications.

Enumeration	none (0), aes128 (2), aes256 (4)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.12.10.12

8.1.11.2.5 cfgSnmpTrapPrivPassword

Privacy Password

This parameter is active when `cfgSnmpTrapVersion` is set to **v3usm(2)**.

Set the privacy password used for encrypted SNMPv3 notifications.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.12.10.13

8.1.11.2.6 cfgSnmpTrapTimeout

SNMP INFORM Timeout

This parameter is active when `cfgSnmpTrapType` is set to **inform(2)**.

Specifies the timeout in seconds between retries.

Range	1 - 2147483647
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.12.10.14

8.1.11.2.7 cfgSnmpTrapRetries

SNMP INFORM Retries

This parameter is active when `cfgSnmpTrapType` is set to **inform(2)**.

Specifies the number of retries.

Range	0 - 2147483647
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.12.10.15

8.1.11.2.8 cfgSnmpTrapVersion

SNMP Version

SNMP notifications can be sent in the following versions:

- **v1(0):** This version is obsolete and defaults to **v2c(1)**
- **v2c(1):** Version 2c
- **v3usm(2):** Version 3 USM

Enumeration	v1 (0), v2c (1), v3usm (2)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.12.10.2

8.1.11.2.9 cfgSnmpTrapCommunity

SNMP Community

This parameter is active when `cfgSnmpTrapVersion` is set to **v2c(1)**.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.12.10.3

8.1.11.2.10 cfgSnmpTrapDest

IP Address of the Notification Receiver

This is an IPv4 address.

Type	IpAddress
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.12.10.4

8.1.11.2.11 cfgSnmpTrapType

SNMP Notification Type

- **trap(1):** Send all SNMP notifications as TRAP
- **inform(2):** Send all SNMP notifications as INFORM

Enumeration	trap (1), inform (2)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.12.10.5

8.1.11.2.12 cfgSnmpTrapEngineId

SNMP Authoritative Engine ID

This parameter is active when `cfgSnmpTrapVersion` is set to **v3usm(2)**.

This parameter may be set to 'auto' (recommended). In 'auto' mode the behaviour is as follows:

- If `cfgSnmpTrapType` is **trap(1)** then the engine ID of the SNMP Agent is used in the notification (see `cfgSnmpdVersion`).
- If `cfgSnmpTrapType` is **inform(1)** then the SNMPv3 discovery process is used to retrieve authoritative engine ID from the receiver.

Otherwise, this parameter may be used to define the engine ID used as the authoritative engine ID for SNMPv3 notifications. The engine ID must be given as a hexadecimal string (optionally prefixed by '0x'). The value must be between 5 and 32 octets long. The string has to follow the engine ID definition in RFC 3411.

In this mode the behaviour is as follows:

- If `cfgSnmpTrapType` is **trap(1)** then this engine ID is used as authoritative engine ID in the notification.
- If `cfgSnmpTrapType` is **inform(1)** then this engine ID is enforce to be used as authoritative engine ID in the SNMPv3 discovery process. If the receiver has a different engine ID, sending notifications will fail.

Examples:

- auto
- 0x80001F8801C0A80014
- 0x80001F880300145A035042
- 0x80001F880441424344454647484950

Type	DisplayString
Range	4 - 66
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.12.10.6

8.1.11.2.13 cfgSnmpTrapUser

SNMP User / Security Name

This parameter is active when `cfgSnmpTrapVersion` is set to **v3usm(2)**.

Set the user name used for authenticated SNMPv3 notifications.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.12.10.9

8.1.12 cfgDhcp

8.1.12.1 cfgDhcpGlobal

8.1.12.1.1 cfgDhcpGlobalEnabled

Enable DNS/DHCP Server Functionality

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.13.1.1

8.1.12.2 cfgDhcpDnsmasqTable

DNS/DHCP Server Instances

Dnsmasq is used 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.

Range	0 - 9
OID	1.3.6.1.4.1.16177.1.400.1.1.13.2

8.1.12.2.1 cfgDhcpDnsmasqTableEntry

DNS/DHCP Server Instances

Dnsmasq is used 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.

OID	1.3.6.1.4.1.16177.1.400.1.1.13.2.1

8.1.12.2.1.1 cfgDhcpDnsmasqIndex

Table Entry Index

Range	0 - 9
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.13.2.1.1

8.1.12.2.1.2 cfgDhcpDnsmasqDnsDomainOverrideParameter

Reference to Domain Override ID

All entries in the `cfgDhcpDomainOverrideTable` with a `cfgDhcpDmnOvrId` that matches the value set here will be used for this DNS server instance.

Range	0 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.13.2.1.10

8.1.12.2.1.3 cfgDhcpDnsmasqDnsHostOverrideParameter

Reference to Host Override ID

All entries in the `cfgDhcpHostOverrideTable` with a `cfgDhcpHstOvrId` that matches the value set here will be used for this DNS server instance.

Range	0 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.13.2.1.11

8.1.12.2.1.4 cfgDhcpDnsmasqCustomOptions

Custom Dnsmasq Options

These options are appended to the dnsmasq 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 space.

The full list of all available options is at: https://thekelleys.org.uk/dnsmasq/docs/dnsmasq-man.html

Prohibited options are:

- addn-hosts
- hostsdir
- keep-in-foreground
- no-daemon
- pid-file
- user
- group
- version
- resolv-file
- dumpfile
- dumpmask
- dnssec-timestamp
- dhcp-hostsfile
- dhcp-optsfile
- dhcp-hostsdir
- dhcp-optsdir
- dhcp-leasefile
- dhcp-script
- dhcp-luascript
- dhcp-scriptuser
- script-arp
- script-on-renewal
- enable-tftp
- tftp-root
- conf-file
- conf-dir
- servers-file
- conf-script

Essentially everything which calls a script, changes files, or otherwise allows to adjust sensitive setting on the system.

When a custom option is set which is configurable via an existing parameter, the custom option will take precedence.

Examples:

- none
- log-queries
- log-dhcp; domain=example.org

Type	DisplayString
Range	4 - 4095
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.13.2.1.12

8.1.12.2.1.5 cfgDhcpDnsmasqScopeParameter

Parameter to Set Which Scope ID to use for the DHCP Server

This is used in conjunction with the scope ID parameter `cfgDhcpScopeId`.

Range	0 - 511
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.13.2.1.2

8.1.12.2.1.6 cfgDhcpDnsmasqDnsPort

DNS Server Port

The UDP and TCP port on which the DNS service is running.

Range	1 - 65535
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.13.2.1.3

8.1.12.2.1.7 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 except localhost (127.0.0.1).

When set to `wildcard`, this instance binds to `0.0.0.0:<port>`.

A DHCP server may not run on the same interface as a DHCP Relay specified in `cfgDhcpRelayInterface`.

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.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.13.2.1.4

8.1.12.2.1.8 cfgDhcpDnsmasqDnsResolveOrder

Upstream DNS Server Resolve Order

Defines in what order the upstream DNS servers shall be querried.

- **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.

Enumeration	any (0), strictorder (1), all (2)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.13.2.1.5

8.1.12.2.1.9 cfgDhcpDnsmasqDnsEnabled

DNS Server Disabled or Enabled

Enable the DNS Server functionality of this instance.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.13.2.1.6

8.1.12.2.1.10 cfgDhcpDnsmasqDhcpEnabled

DHCP Server Disabled or Enabled

Enable the DHCP Server functionality of this instance.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.13.2.1.7

8.1.12.2.1.11 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.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.13.2.1.8

8.1.12.2.1.12 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.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.13.2.1.9

8.1.12.3 cfgDhcpScopeTable

DHCP Instance Configs

Range	0 - 511
OID	1.3.6.1.4.1.16177.1.400.1.1.13.3

8.1.12.3.1 cfgDhcpScopeTableEntry

DHCP Instance Configs

OID	1.3.6.1.4.1.16177.1.400.1.1.13.3.1

8.1.12.3.1.1 cfgDhcpScopeIndex

Table Entry Index

Range	0 - 511
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.13.3.1.1

8.1.12.3.1.2 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
- 26,1420

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.

Type	DisplayString
Range	1 - 1024
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.13.3.1.14

8.1.12.3.1.3 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.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.13.3.1.15

8.1.12.3.1.4 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.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.13.3.1.16

8.1.12.3.1.5 cfgDhcpScopeId

Scope ID

This is used in conjunction with the DHCP parameter cfgDhcpDnsmasqScopeParameter.

Range	0 - 511
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.13.3.1.2

8.1.12.3.1.6 cfgDhcpScopeTag

DHCP Scope Tag

The tag for which this scope is active. Use the `cfgDhcpTagTable` to specify a tag that is matched here. Allowed are alpha-numeric characters.

Set to `none` to disable this field.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.13.3.1.20

8.1.12.3.1.7 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

Type	DisplayString
Range	1 - 15
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.13.3.1.3

8.1.12.3.1.8 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.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.13.3.1.4

8.1.12.3.1.9 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:

network address + `cfgDhcpScopeStart` + `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.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.13.3.1.5

8.1.12.3.1.10 cfgDhcpScopeLeasetime

DHCP Scope Lease Time

Specifies the lease time of addresses handed out to clients.

Examples:

- 12h
- 30m
- 180s

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.13.3.1.6

8.1.12.3.1.11 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.

Type	IpAddress
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.13.3.1.7

8.1.12.3.1.12 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.

Type	IpAddress
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.13.3.1.8

8.1.12.3.1.13 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.

Type	IpAddress
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.13.3.1.9

8.1.12.4 cfgDhcpDomainOverrideTable

DNS Server Host Override Table

Range	0 - 255
OID	1.3.6.1.4.1.16177.1.400.1.1.13.4

8.1.12.4.1 cfgDhcpDomainOverrideTableEntry

DNS Server Domain Override Table

OID	1.3.6.1.4.1.16177.1.400.1.1.13.4.1

8.1.12.4.1.1 cfgDhcpDmnOvrIndex

Table Entry Index

Range	0 - 255
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.13.4.1.1

8.1.12.4.1.2 cfgDhcpDmnOvrId

Domain Override Entry ID

This is used in conjunction with `cfgDhcpDnsmasqDnsDomainOverrideParameter`.

All entries set to the value referenced are used by the DNS server.

Range	0 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.13.4.1.2

8.1.12.4.1.3 cfgDhcpDmnOvrEnabled

Domain Override Entry Disabled or Enabled

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.13.4.1.3

8.1.12.4.1.4 cfgDhcpDmnOvrDomain

Domain Override Entry Domain

Domain to match. Queries for the matched domain are sent to the server specified in `cfgDhcpHstOvrServer`. This entry is considered disabled when set to `none`.

All subdomains of a specified domain are included. Multiple servers may be specified for the same domain. These servers are querried in the order defined in `cfgDhcpDnsmasqDnsResolveOrder`.

When set to `*`, this server is queried for all domains, and the servers specified in `cfgSysNameserverTable` are ignored. Multiple servers may be specified for the same domain. These servers are querried in the order defined in `cfgDhcpDnsmasqDnsResolveOrder`.

Examples:

- none
- example.com
- mydomain.net
- *

To exclude a subdomain from another override, set the corresponding `cfgDhcpDmnOvrServer` to `#`.

Example:

- specify example.com via 192.168.1.20
- specify blocked.example.com via #
- example.com is querried via 192.168.1.20
- subdomain.example.com is querried via 192.168.1.20
- blocked.example.com is querried via `cfgSysNameserverTable`

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.13.4.1.4

8.1.12.4.1.5 cfgDhcpDmnOvrServer

Domain Override Entry Server

The server which is querried for the domain and its subdomains specified in `cfgDhcpDmnOvrDomain`.

The format of the server is: `IPv4-Address[#Destination-Port][@Source-Address|@Source-Interface][`

- The IPv4-Address is the destination to which requests are sent.
- The Destination-Port specifies on which port the server is running. When omitted, the default port 53 is used.

- The Source-Address specifies from which address the requests are sent. Instead of a Source-Address, a Source-Interface may be specified. This is used in combination with dynamic addresses, for example CARP. When omitted, the routing table provides the source to be used.
- The Source-Port may be used to specify from which port requests are send. When ommited, a random port is used.

This entry is considered disabled when set to `none`.

Set to # to exempt a subdomain, (see `cfgDhcpDmnOvrDomain`).

Examples:

- none
- #
- 192.168.1.20
- 192.168.1.20#10053
- 192.168.1.20@192.168.1.2
- 192.168.1.20#10053@192.168.1.2
- 192.168.1.20@192.168.1.2#12345
- 192.168.1.20#10053@192.168.1.2#12345
- 192.168.1.20@br0.vlan0
- 192.168.1.20@br0.vlan0#12345
- 192.168.1.20#10053@br0.vlan0#12345

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.13.4.1.5

8.1.12.5 cfgDhcpHostOverrideTable

DNS Server Host Override Table

Range	0 - 255
OID	1.3.6.1.4.1.16177.1.400.1.1.13.5

8.1.12.5.1 cfgDhcpHostOverrideTableEntry

DNS Server Host Override Table

OID	1.3.6.1.4.1.16177.1.400.1.1.13.5.1

8.1.12.5.1.1 cfgDhcpHstOvrIndex

Table Entry Index

Range	0 - 255
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.13.5.1.1

8.1.12.5.1.2 cfgDhcpHstOvrId

Host Override Entry ID

This is used in conjunction with `cfgDhcpDnsmasqDnsHostOverrideParameter`.

All entries set to the value referenced are used by the DNS server.

Range	0 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.13.5.1.2

8.1.12.5.1.3 cfgDhcpHstOvrEnabled

Host Override Entry Disabled or Enabled

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.13.5.1.3

8.1.12.5.1.4 cfgDhcpHstOvrHost

Host Override Entry Host

FQDN or name to match. Is considered disabled when set to `none`.

Examples:

- none
- host.example.com
- somenamewithouttld

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.13.5.1.4

8.1.12.5.1.5 cfgDhcpHstOvrAddress

Host Override Entry Address

IP Address to return.

This is the address which is returned when a query is received for the FQDN or name configured in `cfgDhcpHstOvrDomain`.

Type	IpAddress
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.13.5.1.5

8.1.12.6 cfgDhcpRelayTable

DHCP Relay Table

A DHCP Relay allows to forward DHCP requests to a remote DHCP server.

Range	0 - 255
OID	1.3.6.1.4.1.16177.1.400.1.1.13.6

8.1.12.6.1 cfgDhcpRelayTableEntry

DHCP Relay Table

OID	1.3.6.1.4.1.16177.1.400.1.1.13.6.1

8.1.12.6.1.1 cfgDhcpRelayIndex

Table Entry Index

Range	0 - 255
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.13.6.1.1

8.1.12.6.1.2 cfgDhcpRelayEnabled

Relay Disabled or Enabled

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.13.6.1.2

8.1.12.6.1.3 cfgDhcpRelayInterface

DHCP Relay Listening Interface

The interface on which the DHCP Relay listens for requests to forward.

A relay may not run on the same interface as a DHCP Server specified in `cfgDhcpScopeInterface`.

Examples:

- br1.vlan0
- wlan0

Type	DisplayString
Range	1 - 15
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.13.6.1.3

8.1.12.6.1.4 cfgDhcpRelayLocalAddress

Local Address to Use as Source to Server

The local address is an IP address residing on the interface specified in `cfgDhcpRelayInterface`. This address is used as Gateway IP Address (giaddr) for the relayed requests.

When set to `auto` the first address on the interface is used.

Examples:

- auto
- 192.168.100.1

Type	DisplayString
Range	4 - 15
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.13.6.1.4

8.1.12.6.1.5 cfgDhcpRelayServerAddress

DHCP Server Address

This is the address to which all DHCP requests arriving on the interface specified in `cfgDhcpRelayInterface` are relayed to.

Multiple servers may be specified as a space and/or comma separated list.

When multiple servers are specified, the requests are concurrently forwarded to all of them.

Examples:

- 10.0.0.1
- 10.0.0.1, 10.0.0.2
- 10.0.0.1 10.0.0.2 10.0.0.3
- 10.0.0.1 10.0.0.2, 10.0.0.3 10.0.0.4

Type	DisplayString
Range	7 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.13.6.1.5

8.1.12.6.1.6 cfgDhcpRelayInterfaceToServer

Interface towards DHCP Server

This parameter defines on which interface DHCP replies from the server will be accepted. This is intended for configurations which have three or more interfaces: one being relayed from, a second connecting the DHCP server, and a third untrusted network, typically the internet. It avoids the possibility of spoof replies arriving via this third interface.

Set to `any` to accept replies from any interface.

Examples:

- any
- br0.vlan0
- eth0

Type	DisplayString
Range	3 - 15
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.13.6.1.6

8.1.12.7 cfgDhcpTagTable

DHCP Tag Table

A Tag allows to match a DHCP request and assign it to a specific DHCP scope with `cfgDhcpScopeTag`.

Range	0 - 255
OID	1.3.6.1.4.1.16177.1.400.1.1.13.7

8.1.12.7.1 cfgDhcpTagTableEntry

DHCP Tag Table

OID	1.3.6.1.4.1.16177.1.400.1.1.13.7.1

8.1.12.7.1.1 cfgDhcpTagIndex

Table Entry Index

Range	0 - 255
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.13.7.1.1

8.1.12.7.1.2 cfgDhcpTagType

DHCP Tag Type

Selects what `cfgDhcpTagMatch` is matching on:

disabled(0): Ignore this entry **mac(1)**: The MAC-address of the requesting client **clientid(2)**: The client id of the requesting client **hostname(3)**: The hostname of the requesting client **circuitid(4)**: The circuit-id (opt 82.1) of the request **remoteid(5)**: The remote-id (opt 82.2) of the request

Enumeration	disabled (0), mac (1), clientid (2), hostname (3), circuitid (4), remoteid (5)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.13.7.1.2

8.1.12.7.1.3 cfgDhcpTagMatch

DHCP Tag Match

Depending on the type of `cfgDhcpTagType` this is another format.

- **mac(1)**: The mac address in hex-colon, e.g `00:14:5a:01:23:45`.
- **clientid(2)**: The clientid in hex-colon or as string. e.g `98:76:54` or `test`.
- **hostname(3)**: The hostname as a string. Validity of a hostname applies.
- **circuitid(4)** and **remoteid(5)**: The circuit-id or remote-id is normally given as colon-separated hex, but is also allowed to be a simple string. e.g: `30:30:30:37:37:63:33:30:35:66:62:35:3a:61:74:`
  which is the equivalent of `00077c305fb5:ats17-mma`.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.13.7.1.3

8.1.12.7.1.4 cfgDhcpTagTag

DHCP Tag Match

The tag that is assigned to a request when `cfgDhcpTagMatch` matches.

Allowed are alpha-numeric characters.

This tag is used in a scope with `cfgDhcpScopeTag`.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.13.7.1.4

8.1.13 cfgNtp

8.1.13.1 cfgNtpEnabled

Disable or Enable Synchronisation via NTP

The received time is in UTC. See `cfgSysTimezone` to specify the timezone.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.14.1

8.1.13.2 cfgNtpClient

8.1.13.2.1 cfgNtpClientTable

NTP Server Table

Range	0 - 31
OID	1.3.6.1.4.1.16177.1.400.1.1.14.10.1

8.1.13.2.1.1 cfgNtpClientTableEntry

NTP Server Table

OID	1.3.6.1.4.1.16177.1.400.1.1.14.10.1.1

cfgNtpClientIndex

Table Entry Index

Range	0 - 31
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.14.10.1.1.1

cfgNtpClientEnabled

NTP Server Disabled or Enabled

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.14.10.1.1.2

cfgNtpClientHost

NTP Server Host

This entry is an IPv4 address or an FQDN (Fully Qualified Domain Name). An FQDN can only be resolved when a nameserver is configured in `cfgSysNameserverTable` or a nameserver is received via DHCP.

An FQDN may resolve to a pool of IPs. When an FQDN is configured, the client will attempt to re-resolve it multiple times until 4 different responses are received.

When set to `0.0.0.0` this entry will be ignored.

Examples:

- pool.ntp.org
- 192.168.1.2
- 0.0.0.0

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.14.10.1.1.3

8.1.13.2.2 cfgNtpClientNmeaEnabled

Disable or Enable Synchronisation via NMEA from a GNSS Receiver

This feature requires `cfgGnssGpsdEnabled` to be set to enabled(1).

Applies to cellular products only.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.14.10.2

8.1.13.3 cfgNtpServer1

****DEPRECATED:** NTP Server 1**

Please use `cfgNtpClientTable`.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.14.2

8.1.13.4 cfgNtpServer

8.1.13.4.1 cfgNtpServerEnabled

NTP Server Disabled or Enabled

When enabled an NTP server will start on UDP port 123.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.14.20.1

8.1.13.4.2 cfgNtpServerLocalReference

Local Reference

The local directive enables a local reference mode, which allows the NTP server to appear synchronised to real time (from the viewpoint of clients polling it), even when it was never synchronised or the last update of the clock happened a long time ago.

When enabled, will announce itself as stratum 10 while not synchronised to a better stratum server.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.14.20.2

8.1.13.5 cfgNtpServer2

****DEPRECATED:** NTP Server 2**

Please use `cfgNtpClientTable`.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.14.3

8.1.14 cfgHttp

8.1.14.1 cfgHttpUser

****OBSOLETE:** Web Administrator Username**

This parameter is obsolete and has no replacement.

There are two local users `admin` and `monitor`, that have the respective 'admin' and 'monitor' role.

Additional users with the 'admin' and 'monitor' role may be defined via LDAP. See `cfgLdapEnabled`.

Type	DisplayString
Range	1 - 255
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.15.1

8.1.14.2 cfgHttpMonitorPasswordHash

Monitor Password Hash

Used for configuration import/export only.

Use WebAPI or Web Interface to change the Password.

To generate a hash offline, use the command `openssl passwd -6 -salt monitor <password>`.

Type	DisplayString
Range	10 - 126
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.15.11

8.1.14.3 cfgHttpSessionLimit

HTTP Access Session Limit

To restrict the maximum number of connected users. Set value <= 0 to disable session limitation.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.15.12

8.1.14.4 cfgHttpSessionTimeout

HTTP Session Timeout

Timeout in seconds after a session is closed when there is no activity.

Range	10 - 18000
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.15.13

8.1.14.5 cfgHttpTlsCiphers

OpenSSL Cipher and Ciphersuites String for HTTPS server

Specify which OpenSSL ciphers and ciphersuites to use for HTTPS connections.

Please read the user manual and the OpenSSL documentation for a list of available ciphers, ciphersuites and used syntax.

Up to TLSv1.2, OpenSSL uses ciphers, while with TLSv1.3 ciphersuites are used. The format for this parameter is as follows: [|], i.e. the pipe ('|') sign is used to separate the ciphers from the ciphersuites.

If ciphers or ciphersuites is left empty, no restrictions are applied and all of the related built-ins are available. If ciphers is set with the special string 'disable', the support for TLSv1.2 and below is disabled, while 'disable' given as cyphersuites disables TLSv1.3.

Set to ‘none’ to disable restriction.

Examples:

- **ECDHE-RSA-AES256-GCM-SHA384:DHE-RSA-AES256-GCM-SHA384**

  limit TLSv1.2 to the selected two ciphers, leave TLSv1.3 ciphersuites all enabled

- **DHE-RSA-AES256-GCM-SHA384|TLS_AES_256_GCM_SHA384**

  limit TLSv1.2 to one cipher and TLSv1.3 to one ciphersuite

- **DHE-RSA-AES256-GCM-SHA384|disable**

  limit TLSv1.2 to one cipher and disable TLSv1.3 support

- **disable|TLS_AES_256_GCM_SHA384**

  disable TLSv1.2 support and limit TLSv1.3 to one ciphersuite

- **|TLS_AES_256_GCM_SHA384**

  leave all TLSv1.2 enabled, limit TLSv1.3 to one ciphersuite

- **|disable**

  leave all TLSv1.2 ciphers enabled, disable TLSv1.3 support

- **disable|**

  disable TLSv1.2 support, leave all TLSv1.3 ciphersuites enabled

- **none**

  no restrictions for TLSv1.2 and TLSv1.3

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.15.14

8.1.14.6 cfgHttpPassword

****OBSOLETE:** Web Administrator Password**

This parameter is obsolete and has been replaced with `cfgHttpAdminPasswordHash` and `cfgHttpMonitorPass`

Additional users with their own password and role may be defined via LDAP. See `cfgLdapEnabled`.

Type	DisplayString
Range	1 - 255
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.15.2

8.1.14.7 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.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.15.3

8.1.14.8 cfgHttpRedirectEnabled

If enabled, all access to `cfgHttpHttpAddress` shall be redirected to `cfgHttpHttpsAddress`. This does not disable http.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.15.4

8.1.14.9 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

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.15.5

8.1.14.10 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

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.15.6

8.1.14.11 cfgHttpTlsServerCertId

TLS Server Certificate ID

Reference ID of certificate in Cert-Store to be used as TLS Server Certificate.

Set to `-1` to use the default self signed server certificate.

Range	-1 - 1000
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.15.7

8.1.14.12 cfgHttpAdminPasswordHash

Admin Password Hash

Used for configuration import/export only.

Use WebAPI or Web Interface to change the Password.

To generate a hash offline, use the command `openssl passwd -6 -salt admin <password>`.

Type	DisplayString
Range	10 - 126
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.15.9

8.1.15 cfgLldp

8.1.15.1 cfgLldpEnabled

Enable LLDP.

Applies to AP and STA.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.16.1

8.1.15.2 cfgLldpDescription

LLDP Description

Applies to AP and STA.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.16.2

8.1.16 cfgMdns

8.1.16.1 cfgMdnsEnabled

Disable or Enable mDNS

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.17.1

8.1.16.2 cfgMdnsNetwork

mDNS Aware Network Interfaces

When set to the keywork `all` it will bind to all valid interfaces.

Multiple interfaces may be specified as a space and/or comma separated list.

Examples:

- br0.vlan0
- br0.vlan0, br0.vlan7, br0.vlan99
- br0.vlan0 br0.vlan66

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.17.2

8.1.17 cfgSsdp

8.1.17.1 cfgSsdpEnabled

Disable or Enable SSDP

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.19.1

8.1.18 cfgNetwork

8.1.18.1 cfgNetEthernetTable

Ethernet Network Interfaces

Range	0 - 9
OID	1.3.6.1.4.1.16177.1.400.1.1.2.1

8.1.18.1.1 cfgNetEthernetTableEntry

Ethernet Network Interfaces

OID	1.3.6.1.4.1.16177.1.400.1.1.2.1.1

8.1.18.1.1.1 cfgNetEthIndex

Table Entry Index

Range	0 - 9
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.2.1.1.1

8.1.18.1.1.2 cfgNetEthTrunk

Ethernet Interface 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 and/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'

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.1.1.10

8.1.18.1.1.3 cfgNetEthTag

Ethernet Interface 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.

Range	-1 - 4094
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.1.1.11

8.1.18.1.1.4 cfgNetEthVlanMode

Ethernet Interface 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).

Enumeration	trunk (0), access (1), nativeuntagged (3)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.1.1.12

8.1.18.1.1.5 cfgNetEthProtected

Ethernet Interface Port Protection

This feature only applies to bridged interfaces.

The protected port feature allows bridged ports to be designated as protected. Traffic between protected ports is blocked. Protected ports can send traffic to unprotected ports. Unprotected ports can send traffic to any port.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.1.1.13

8.1.18.1.1.6 cfgNetEthLldpEnabled

Ethernet Interface LLDP Disabled or Enabled

When `cfgLldpEnabled` is enabled, this parameter controls if the interface takes part in LLDP operation.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.1.1.15

8.1.18.1.1.7 cfgNetEthMtu

Ethernet Interface MTU

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

Range	-1 - 9000
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.1.1.16

8.1.18.1.1.8 cfgNetEthMac

Ethernet Interface MAC Address

Set 00:00:00:00:00:00 to use the MAC address stored in the flash of the device.

Format: `00:14:5a:02:10:42`

Type	DisplayString
Range	17 - 17
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.1.1.19

8.1.18.1.1.9 cfgNetEthName

Name of the Ethernet Interface

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.1.2.1.1.2

8.1.18.1.1.10 cfgNetEthEnabled

Ethernet Interface Disabled or Enabled

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.1.1.3

8.1.18.1.1.11 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

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.1.1.7

8.1.18.1.1.12 cfgNetEthAutoneg

Disable or Enable 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 and 2500Mbit require the mode to be auto.
- **auto(1):** Advertises the supported auto negotiation defined by `cfgNetEthSpeed`.

Enumeration	forced (0), auto (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.1.1.8

8.1.18.1.1.13 cfgNetEthSpeed

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
- 0x80 (128) = 2500Mbit/Full

When `cfgNetEthAutoneg` is **forced(0)** only a single bit may be active. Only 10Mbit and 100Mbit rates are allowed in forced mode. 1000Mbit and 2500Mbit require the mode to be auto.

Examples:

- **1:** Force 10Mbit half duplex
- **8:** Force 100Mbit full duplex

When `cfgNetEthAutoneg` 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 up to 1000Mbit (1 + 2 + 4 + 8 + 32)
- **175:** Advertise all speeds up to 2500Mbit (1 + 2 + 4 + 8 + 32 + 128)

Range	1 - 175
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.1.1.9

8.1.18.2 cfgNetEth802dot1xTable

Wired 802.1X

This table allows to configure authentication of clients connecting to the Ethernet ports.

The Ethernet port stays locked until the client has authenticated itself via 802.1X.

Depending on the value in `cfgNetEth802dot1xEapReauthPeriod`, client have to reauthenticate regularly.

A port is automatically locked when the link of the interface goes down (the cable is unplugged), and has to be authenticated again until it can be further used.

Range	0 - 9
OID	1.3.6.1.4.1.16177.1.400.1.1.2.10

8.1.18.2.1 cfgNetEth802dot1xTableEntry

Wired 802.1X Entry

OID	1.3.6.1.4.1.16177.1.400.1.1.2.10.1

8.1.18.2.1.1 cfgNetEth802dot1xIndex

Table Entry Index

Range	0 - 9
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.2.10.1.1

8.1.18.2.1.2 cfgNetEth802dot1xName

Name of the Ethernet Interface

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.1.2.10.1.2

8.1.18.2.1.3 cfgNetEth802dot1xEnabled

Wired Port Security Interface Disabled or Enabled

When this is enabled, connecting clients will not be admitted until they have authenticated against the configured RADIUS server.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.10.1.3

8.1.18.2.1.4 cfgNetEth802dot1xOwnIpAddr

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.

Type	IpAddress
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.10.1.4

8.1.18.2.1.5 cfgNetEth802dot1xAuthServerParameter

Reference ID to the RADIUS Auth Server Table

Uses all auth servers in the `cfgWlan802dot1xAuthServerTable` which have as `cfgWlan802dot1xAuthSrvId` the value set here.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.10.1.5

8.1.18.2.1.6 cfgNetEth802dot1xEapReauthPeriod

EAP Reauthentication Period in Seconds

To disable reauthentication, set this value to `0`.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.10.1.6

8.1.18.3 cfgNetWwanTable

Cellular Network Interfaces

Applies to cellular products only.

Range	0 - 7
OID	1.3.6.1.4.1.16177.1.400.1.1.2.11

8.1.18.3.1 cfgNetWwanTableEntry

Cellular Network Interface

Applies to cellular products only.

OID	1.3.6.1.4.1.16177.1.400.1.1.2.11.1

8.1.18.3.1.1 cfgNetWwanIndex

Table Entry Index

Applies to cellular products only.

Range	0 - 7
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.2.11.1.1

8.1.18.3.1.2 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.

Range	-1 - 65535
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.11.1.16

8.1.18.3.1.3 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`.

The default carrier configuration is set in `cfgCellDefaultBearerTable`.

Applies to cellular products only.

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.1.2.11.1.2

8.1.18.3.1.4 cfgNetWwanEnabled

Cellular Network Interface Disabled or Enabled

Applies to cellular products only.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.11.1.3

8.1.18.4 cfgNetOpenvpnTable

OpenVPN Interface Table

This table is in a one-to-one relation with the `cfgVpnOpenvpnTable`, where both indices match.

Range	0 - 15
OID	1.3.6.1.4.1.16177.1.400.1.1.2.12

8.1.18.4.1 cfgNetOpenvpnTableEntry

OpenVPN Interface Entry

OID	1.3.6.1.4.1.16177.1.400.1.1.2.12.1

8.1.18.4.1.1 cfgNetOpenvpnIndex

Table Entry Index

Range	0 - 15
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.2.12.1.1

8.1.18.4.1.2 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 and/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'

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.12.1.10

8.1.18.4.1.3 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.

Range	-1 - 4094
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.12.1.11

8.1.18.4.1.4 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).

Enumeration	trunk (0), access (1), nativeuntagged (3)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.12.1.12

8.1.18.4.1.5 cfgNetOpenvpnProtected

OpenVPN Port Protection

This feature only applies to bridged interfaces.

The protected port feature allows bridged ports to be designated as protected. Traffic between protected ports is blocked. Protected ports can send traffic to unprotected ports. Unprotected ports can send traffic to any port.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.12.1.13

8.1.18.4.1.6 cfgNetOpenvpnLldpEnabled

OpenVPN Interface LLDP Disabled or Enabled

When `cfgLldpEnabled` is enabled, this parameter controls if the interface takes part in LLDP operation.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.12.1.15

8.1.18.4.1.7 cfgNetOpenvpnMtu

OpenVPN Interface MTU

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).

Range	-1 - 65535
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.12.1.16

8.1.18.4.1.8 cfgNetOpenvpnName

Name of the OpenVPN Interface

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.1.2.12.1.2

8.1.18.4.1.9 cfgNetOpenvpnEnabled

OpenVPN Interface Disabled or Enabled

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.12.1.3

8.1.18.4.1.10 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 managment interface

Note: Interfaces may only be configured in a bridge when `cfgVpnOpenvpnDevType` is set to 1 (tap).

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.12.1.7

8.1.18.5 cfgNetIpsecTable

IPsec Network Interfaces

Range	0 - 15
OID	1.3.6.1.4.1.16177.1.400.1.1.2.13

8.1.18.5.1 cfgNetIpsecTableEntry

IPsec Network Interfaces

OID	1.3.6.1.4.1.16177.1.400.1.1.2.13.1

8.1.18.5.1.1 cfgNetIpsecIndex

Table Entry Index

Range	0 - 15
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.2.13.1.1

8.1.18.5.1.2 cfgNetIpsecMtu

IPsec Interface MTU

When VTI is enabled (see `cfgVpnIpsecGlblVirtualTunnelInterface`), this allows to manually set the MTU of the IPsec tunnel.

Range	-1 - 65515
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.13.1.16

8.1.18.5.1.3 cfgNetIpsecName

Name of the IPsec Interface

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.1.2.13.1.2

8.1.18.5.1.4 cfgNetIpsecEnabled

IPsec Interface Disabled or Enabled

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.13.1.3

8.1.18.6 cfgNetFlowControllerTable

FlowController Interfaces

Range	0 - 15
OID	1.3.6.1.4.1.16177.1.400.1.1.2.14

8.1.18.6.1 cfgNetFlowControllerTableEntry

FlowController Network Interfaces

OID	1.3.6.1.4.1.16177.1.400.1.1.2.14.1

8.1.18.6.1.1 cfgNetFcIndex

Table Entry Index

Range	0 - 15
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.2.14.1.1

8.1.18.6.1.2 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 and/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’

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.14.1.10

8.1.18.6.1.3 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.

Range	-1 - 4094
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.14.1.11

8.1.18.6.1.4 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).

Enumeration	trunk (0), access (1), nativeuntagged (3)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.14.1.12

8.1.18.6.1.5 cfgNetFcProtected

TECHPREVIEW: Flow Controller Port Protection

This feature only applies to bridged interfaces.

The protected port feature allows bridged ports to be designated as protected. Traffic between protected ports is blocked. Protected ports can send traffic to unprotected ports. Unprotected ports can send traffic to any port.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.14.1.13

8.1.18.6.1.6 cfgNetFcLldpEnabled

TECHPREVIEW: Flow Controller LLDP Disabled or Enabled

When `cfgLldpEnabled` is enabled, this parameter controls if the interface takes part in LLDP operation.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.14.1.15

8.1.18.6.1.7 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 bride. The order in which interfaces are added is ascending in interface number: eth, wlan, ovpn, fc.
- **lacpactive(2):** LACP active mode with fast time

Enumeration	normal (0), fullduplex (1), lacpactive (2)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.14.1.17

8.1.18.6.1.8 cfgNetFcName

TECHPREVIEW: Name of the Flow Controller Bridge

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.1.2.14.1.2

8.1.18.6.1.9 cfgNetFcEnabled

TECHPREVIEW: Flow Controller Bridge Disabled or Enabled

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.14.1.3

8.1.18.6.1.10 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 managment interface

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.14.1.7

8.1.18.7 cfgNetTunnelEndPointTable

Tunnel Endpoint Interfaces

Range	0 - 15
OID	1.3.6.1.4.1.16177.1.400.1.1.2.15

8.1.18.7.1 cfgNetTunnelEndPointTableEntry

Tunnel Endpoint Interfaces

OID	1.3.6.1.4.1.16177.1.400.1.1.2.15.1

8.1.18.7.1.1 cfgNetTepIndex

Table Entry Index

Range	0 - 15
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.2.15.1.1

8.1.18.7.1.2 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 and/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'

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.15.1.10

8.1.18.7.1.3 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.

Range	-1 - 4094
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.15.1.11

8.1.18.7.1.4 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).

Enumeration	trunk (0), access (1), nativeuntagged (3)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.15.1.12

8.1.18.7.1.5 cfgNetTepProtected

Tunnel Endpoint Port Protection

This feature only applies to bridged interfaces.

The protected port feature allows bridged ports to be designated as protected. Traffic between protected ports is blocked. Protected ports can send traffic to unprotected ports. Unprotected ports can send traffic to any port.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.15.1.13

8.1.18.7.1.6 cfgNetTepLldpEnabled

Tunnel Endpoint LLDP Disabled or Enabled

When `cfgLldpEnabled` is enabled, this parameter controls if the interface takes part in LLDP operation.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.15.1.15

8.1.18.7.1.7 cfgNetTepMtu

Tunnel Endpoint MTU

The minimum allowed valid value is 68. The maximum allowed valid value depends on the used tunnel type specified in `cfgVpnTepTunnelType`. The default value is `-1`, and sets the MTU automatically according to the list below. The default value depends on the tunnel type specified in `cfgVpnTepTunnelType`.

Type	Default	Maximum
gre	1476 (1500-20-4)	65504
gretap	1462 (1500-20-4-14)	65490
vxlan	1450 (1500-20-8-8-14)	65484

The outer IP header is 20 bytes. The GRE header is 4 bytes. For gretap there is an additional ethernet header of 14 bytes. Vxlan has an UDP header of 8 bytes and the vxlan header of 8 bytes.

When the TEP interface is bridged and the payload is VLAN tagged the resulting frames have another 4 byte overhead. The MTU in the above list has to be reduced accordingly.

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

Range	-1 - 65504
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.15.1.16

8.1.18.7.1.8 cfgNetTepName

Name of the Tunnel Endpoint

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.1.2.15.1.2

8.1.18.7.1.9 cfgNetTepEnabled

Tunnel Endpoint Disabled or Enabled

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.15.1.3

8.1.18.7.1.10 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 `cfgVpnTepTunnelType` are able to be bridged. Currently supported types to be bridged are:

- **gretap(1)**
- **vxlan(2)**

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.15.1.7

8.1.18.8 cfgNetWireguardTable

Wireguard Interfaces

Range	0 - 15
OID	1.3.6.1.4.1.16177.1.400.1.1.2.16

8.1.18.8.1 cfgNetWireguardTableEntry

Wireguard Interfaces

OID	1.3.6.1.4.1.16177.1.400.1.1.2.16.1

8.1.18.8.1.1 cfgNetWgIndex

Table Entry Index

Range	0 - 15
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.2.16.1.1

8.1.18.8.1.2 cfgNetWgMtu

Wireguard Interface MTU

The default value is -1, which does not change what is set by the system (usually 1420).

Range	-1 - 65504
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.16.1.16

8.1.18.8.1.3 cfgNetWgName

Name of the Wireguard Interface

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.1.2.16.1.2

8.1.18.8.1.4 cfgNetWgEnabled

Wireguard Interface Disabled or Enabled

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.16.1.3

8.1.18.9 cfgNetWlanTable

WLAN Network Interfaces

Range	0 - 63
OID	1.3.6.1.4.1.16177.1.400.1.1.2.2

8.1.18.9.1 cfgNetWlanTableEntry

WLAN Network Interfaces

OID	1.3.6.1.4.1.16177.1.400.1.1.2.2.1

8.1.18.9.1.1 cfgNetWlanIndex

Table Entry Index

Range	0 - 63
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.2.2.1.1

8.1.18.9.1.2 cfgNetWlanTrunk

Wireless Interface 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 and/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'

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.2.1.10

8.1.18.9.1.3 cfgNetWlanTag

Wireless Interface 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.

Range	-1 - 4094
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.2.1.11

8.1.18.9.1.4 cfgNetWlanVlanMode

Wireless Interface 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.

Enumeration	trunk (0), access (1), nativeuntagged (3)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.2.1.12

8.1.18.9.1.5 cfgNetWlanProtected

Wireless Interface Port Protection

This feature only applies to bridged interfaces.

The protected port feature allows bridged ports to be designated as protected. Traffic between protected ports is blocked. Protected ports can send traffic to unprotected ports. Unprotected ports can send traffic to any port.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.2.1.13

8.1.18.9.1.6 cfgNetWlanLldpEnabled

Wireless Interface LLDP Disabled or Enabled

When `cfgLldpEnabled` is enabled, this parameter controls if the interface takes part in LLDP operation.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.2.1.15

8.1.18.9.1.7 cfgNetWlanMtu

Wireless Interface MTU

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

Range	-1 - 2304
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.2.1.16

8.1.18.9.1.8 cfgNetWlanDhcpOption82

Wireless Interface DHCP Option 82

When the wireless interface is bridged, enabling this option causes all DHCP request frames that pass through the interface to be extended with the DHCP option 82.

The circuit-id (suboption 1) defines the identity of the AP and the circuit where the request was received. This is in the form: `<hostname>:<iface>`

Example:

- rmodem:wlan0

The remote-id (suboption 2) defines the remote host. It is the MAC of the AP (static 6 bytes) followed by the SSID. This is in the form: `<mac>:<ssid>`

Example:

- 00145a012345:ExampleSsid

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.2.1.17

8.1.18.9.1.9 cfgNetWlanName

Name of the Wireless Interface

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.1.2.2.1.2

8.1.18.9.1.10 cfgNetWlanEnabled

Wireless Interface Disabled or Enabled

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.2.1.3

8.1.18.9.1.11 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 managment interface

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.2.1.7

8.1.18.10 cfgNetVlanTable

VLAN Network Interfaces

Range	0 - 127
OID	1.3.6.1.4.1.16177.1.400.1.1.2.3

8.1.18.10.1 cfgNetVlanTableEntry

VLAN Network Interfaces

OID	1.3.6.1.4.1.16177.1.400.1.1.2.3.1

8.1.18.10.1.1 cfgNetVlanIndex

Table Entry Index

Range	0 - 127
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.2.3.1.1

8.1.18.10.1.2 cfgNetVlanProtected

VLAN Interface Port Protection

This feature only applies to bridged interfaces.

The protected port feature allows bridged ports to be designated as protected. Traffic between protected ports is blocked. Protected ports can send traffic to unprotected ports. Unprotected ports can send traffic to any port.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.3.1.13

8.1.18.10.1.3 cfgNetVlanMtu

VLAN Interface MTU

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 minumum 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.

Range	-1 - 9000
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.3.1.16

8.1.18.10.1.4 cfgNetVlanPriority

VLAN Interface Priority (PCP)

Sets the PCP of the 802.1Q header to the value specified.

When set to -1, will not set anything. However the PCP may be inherited from the priority of a frame received via another interface, e.g. wlan (TID), when L2 prioritisation is enabled (see `cfgQosL3PrioEnabled`).

Range	-1 - 7
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.3.1.18

8.1.18.10.1.5 cfgNetVlanComment

VLAN Interface User Comment

This parameter has no operational function. It allows to store a comment about the use of this VLAN.

Type	DisplayString
Range	0 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.3.1.19

8.1.18.10.1.6 cfgNetVlanName

Name of the VLAN Interface

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.1.2.3.1.2

8.1.18.10.1.7 cfgNetVlanEnabled

VLAN Interface Disabled or Enabled

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.3.1.3

8.1.18.10.1.8 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`.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.3.1.7

8.1.18.10.1.9 cfgNetVlanParent

VLAN Interface 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).

Type	DisplayString
Range	1 - 15
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.3.1.8

8.1.18.10.1.10 cfgNetVlanVid

VLAN Interface VID

Range	0 - 4094
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.3.1.9

8.1.18.11 cfgNetIpTable

IP Address Configuration

Range	0 - 127
OID	1.3.6.1.4.1.16177.1.400.1.1.2.6

8.1.18.11.1 cfgNetIpTableEntry

IP Address Configuration

OID	1.3.6.1.4.1.16177.1.400.1.1.2.6.1

8.1.18.11.1.1 cfgNetIpIndex

Table Entry Index

Range	0 - 127
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.2.6.1.1

8.1.18.11.1.2 cfgNetIpCarpId

Reference to CARP-Instance ID

The referenced ID is reflected by `cfgNetCarpIndex`. This parameter is only active when `cfgNetIpProto` is set to **carp(5)**. Indicates that the referenced CARP instance processes this IP.

Range	0 - 15
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.6.1.10

8.1.18.11.1.3 cfgNetIpComment

IP User Comment

This parameter has no operational function. It allows to store a comment about the use of this IP address.

Type	DisplayString
Range	0 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.6.1.11

8.1.18.11.1.4 cfgNetIpEnabled

IP Address Disabled or Enabled

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.6.1.3

8.1.18.11.1.5 cfgNetIpAddr

IP Address

The IPv4 address (using CIDR notation) of the interface specified in `cfgNetIpInterface`.

Type	DisplayString
Range	5 - 50
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.6.1.4

8.1.18.11.1.6 cfgNetIpProto

Protocol

This parameter defines which protocol is used to configure the IPv4 address 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 may 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.

Enumeration	static (0), dhcp (1), dhcpForceRenew (2), dhcpForceRelease (3), linkLocal (4), carp (5)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.6.1.6

8.1.18.11.1.7 cfgNetIpInterface

Interface Name

The name of the network interface on which the specified IP address is created.

Type	DisplayString
Range	1 - 15
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.6.1.8

8.1.18.12 cfgNetCarpTable

Redundant IP Addresses with the Common Address Redundancy Protocol (CARP)

Range	0 - 15
OID	1.3.6.1.4.1.16177.1.400.1.1.2.7

8.1.18.12.1 cfgNetCarpTableEntry

Redundant IP Addresses with the Common Address Redundancy Protocol (CARP)

OID	1.3.6.1.4.1.16177.1.400.1.1.2.7.1

8.1.18.12.1.1 cfgNetCarpIndex

Table Entry Index

Range	0 - 15
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.2.7.1.1

8.1.18.12.1.2 cfgNetCarpVhid

CARP 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.

Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.7.1.10

8.1.18.12.1.3 cfgNetCarpPassword

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.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.7.1.11

8.1.18.12.1.4 cfgNetCarpAdvbase

CARP Advertisement Base

This parameter has a unit of 1 Second.

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.

Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.7.1.12

8.1.18.12.1.5 cfgNetCarpAdvskew

CARP Advertisement Skew

This parameter has a unit of 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.

Range	0 - 254
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.7.1.13

8.1.18.12.1.6 cfgNetCarpAdvdivider

CARP Divider

The 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 advertisie time of 100ms instead of 1s.

This must be the same on all members of the group.

Range	1 - 100
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.7.1.14

8.1.18.12.1.7 cfgNetCarpRatio

CARP Dead Ratio

The 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.

Range	1 - 100
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.7.1.15

8.1.18.12.1.8 cfgNetCarpPreempt

CARP Preemption

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.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.7.1.16

8.1.18.12.1.9 cfgNetCarpPreemptdemote

CARP Preempt Demotion

Preemt other masters when they demote themself.

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.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.7.1.17

8.1.18.12.1.10 cfgNetCarpLocalInterfaceGroup

CARP Local Interface Group

The local carp interface group to which the carp interface belongs.

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.

Range	0 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.7.1.18

8.1.18.12.1.11 cfgNetCarpSyncInterface

CARP Control Interface

This interface is used to transmit and receive CARP advertisements.

This interface is required to have its own unique IP address.

Type	DisplayString
Range	1 - 15
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.7.1.19

8.1.18.12.1.12 cfgNetCarpMcastIp

CARP Multicast Address

The default multicast address for CARP is 224.0.0.18.

Use this parameter to specify a custom multicast destination.

Type	IpAddress
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.7.1.21

8.1.18.12.1.13 cfgNetCarpEnabled

CARP Interface Disabled or Enabled

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.7.1.3

8.1.18.13 cfgNetMacVlanTable

MACVLAN Network Interfaces

Range	0 - 63
OID	1.3.6.1.4.1.16177.1.400.1.1.2.9

8.1.18.13.1 cfgNetMacVlanTableEntry

MACVLAN Network Interfaces

OID	1.3.6.1.4.1.16177.1.400.1.1.2.9.1

8.1.18.13.1.1 cfgNetMacVlanIndex

Table Entry Index

Range	0 - 63
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.2.9.1.1

8.1.18.13.1.2 cfgNetMacVlanMac

MACVLAN MAC Address

Set `00:00:00:00:00:00` to use a random MAC address.

Format: `00:14:5a:02:10:42`

Type	DisplayString
Range	17 - 17
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.9.1.15

8.1.18.13.1.3 cfgNetMacVlanMtu

MACVLAN Interface MTU

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.

Range	-1 - 9000
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.9.1.16

8.1.18.13.1.4 cfgNetMacVlanComment

MACVLAN Interface User Comment

This parameter has no operational function. It allows to store a comment about the use of this MacVlan.

Type	DisplayString
Range	0 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.9.1.19

8.1.18.13.1.5 cfgNetMacVlanName

Name of the MACVLAN Interface

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.1.2.9.1.2

8.1.18.13.1.6 cfgNetMacVlanEnabled

MACVLAN Interface Disabled or Enabled

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.9.1.3

8.1.18.13.1.7 cfgNetMacVlanParent

MACVLAN Interface Parent

Name of the parent interface on which the MACVLAN resides.

Type	DisplayString
Range	1 - 15
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.9.1.8

8.1.19 cfgWireless

8.1.19.1 cfgWlanDeviceTable

Wireless Hardware Modules

Range	0 - 2
OID	1.3.6.1.4.1.16177.1.400.1.1.3.1

8.1.19.1.1 cfgWlanDeviceTableEntry

Wireless Hardware Modules

OID	1.3.6.1.4.1.16177.1.400.1.1.3.1.1

8.1.19.1.1.1 cfgWlanDevIndex

Table Entry Index

Range	0 - 2
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.3.1.1.1

8.1.19.1.1.2 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.

Range	0 - 114750
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.1.1.10

8.1.19.1.1.3 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.

Range	-1 - 65535
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.1.1.11

8.1.19.1.1.4 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.

Range	-1 - 2346
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.1.1.12

8.1.19.1.1.5 cfgWlanDevShortRetry

RTS Transmission Retries

This parameter only applies to legacy bitrates (802.11b/g/a). To affect the retry behaviour of 802.11n rates see `cfgWlanDevQmrrString`.

Number of times the transmission of the RTS frame will be retried if there is no CTS received from the AP. This is in addition to the initial RTS attempt.

Applies to AP and STA. 802.11n products only.

Range	1 - 15
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.1.1.13

8.1.19.1.1.6 cfgWlanDevLongRetry

Data Frame Tries

This parameter only applies to legacy bitrates (802.11b/g/a). To affect the retry behaviour of 802.11n rates see `cfgWlanDevQmrrString`.

Number of times the unicast data frames will be retried if there is no ACK from the receiver. This value includes the initial attempt.

Applies to AP and STA. 802.11n products only.

Range	1 - 15
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.1.1.14

8.1.19.1.1.7 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.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.1.1.15

8.1.19.1.1.8 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.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.1.1.16

8.1.19.1.1.9 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.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.1.1.17

8.1.19.1.1.10 cfgWlanDevPhy

The Map Between Physical Device and Radio

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.1.3.1.1.18

8.1.19.1.1.11 cfgWlanDevName

Name of the Wireless Device.

Applies to AP and STA.

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.1.3.1.1.2

8.1.19.1.1.12 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.

Range	0 - 65535
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.1.1.24

8.1.19.1.1.13 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.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.1.1.26

8.1.19.1.1.14 cfgWlanDevAtfSchedulingAlgorithm

ATF Scheduling Algorithm

The following airtime scheduling algorithms are available:

- **fair(0)**: try to redistribute unused airtime.
- **strict(1)**: strictly follow the given airtime.

Applies to 802.11ac products in AP mode.

Enumeration	fair (0), strict (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.1.1.27

8.1.19.1.1.15 cfgWlanDevModulation

Physical Wireless Device Modulation Mode

For an AP 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.
- **ng(10):** For 2.4GHz
- **na(12):** For 5GHz.
- **ac(28):** 11ac mode for 5GHz.
- **axg(58):** 11ax mode for 2.4GHz.
- **axa(60):** 11ax mode for 5GHz or 6GHz.
- **beg(122):** 11be mode for 2.4GHz.
- **bea(124):** 11be mode for 5GHz or 6GHz.

Note: Some products do not support all modulations. Please check the data-sheet of your product.

Note:: The STA scans both frequency bands (2.4 GHz and 5GHz) regardless of the modulation. The STA will adapt to the modulation of the Access Point during connection.

Note:: For a STA the modulation can be used to explicitly disable n-rates even tough n-rates would be supported by the Access Point. The n-rates data rates support up to 300 MBit/s in the 2.4GHz and 5GHz frequency band and only OFDM modulation.

Setting modulation to **10(ng)** or **12(na)** will enable n-rates if n-rates are supported by the Access Point, while setting modulation to **g(2)**, **bg(3)** or **a(4)** will disable n-rates. This feature applies to 802.11n products only.

Applies to AP and STA.

Enumeration	g (2), bg (3), a (4), ng (10), na (12), ac (28), axg (58), axa (60), beg (122), bea (124)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.1.1.4

8.1.19.1.1.16 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/ax mode)
- **bw160(6):** 160MHz wide channel (only in 11ac/ax mode)
- **bw8080(7):** 80+80MHz wide channel (only in 11ac/ax mode)
- **bwAuto(8):** automatic channel width (only in 11ac/ax mode)
- **bw320(9):** 320MHz wide channel (only in 11be 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
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.

Enumeration	bw20 (0), bw40Plus (1), bw40Minus (2), bwQuarter (3), bwHalf (4), bw80 (5), bw160 (6), bw8080 (7), bwAuto (8), bw320 (9)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.1.1.5

8.1.19.1.1.17 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, this parameter defindes the single scan frequency when `cfgWlanIfaceScanList` is set to `-1`.

Applies to AP and STA.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.1.1.6

8.1.19.1.1.18 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.

Range	0 - 50
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.1.1.8

8.1.19.2 cfgWlan802dot1xTable

Wireless 802dot1x

Range	0 - 63
OID	1.3.6.1.4.1.16177.1.400.1.1.3.10

8.1.19.2.1 cfgWlan802dot1xTableEntry

Wireless 802dot1x entry

OID	1.3.6.1.4.1.16177.1.400.1.1.3.10.1

8.1.19.2.1.1 cfgWlan802dot1xIndex

Table Entry Index

Range	0 - 63
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.3.10.1.1

8.1.19.2.1.2 cfgWlan802dot1xIdentity

Identity for the RADIUS Client

This is the identity that is sent to the RADIUS server (User-Name).

Applies to STA.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.10.1.10

8.1.19.2.1.3 cfgWlan802dot1xClientKeyPassword

****OBSOLETE:** Password to Unlock the Private Key**

This parameter is obsolete and has been replaced with the Certificate Store.

Key material on the device is always encrypted. The password to import the file has to be specified once during import via `setCrtFilePassphrase`.

Type	DisplayString
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.3.10.1.17

8.1.19.2.1.4 cfgWlan802dot1xTlsControlParams

Bitfield to Control TLS Behaviour

- **0x0** all validity checks will be performed

- **0x1** ignore certificate validity time
- **0x2** ignore ca certificate
- **0x4** ignore CRLs
- **0x8** ignore missing CRLs

Applies to STA.

Range	0 - 15
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.10.1.18

8.1.19.2.1.5 cfgWlan802dot1xRetryMax

Number of Tries Before a RADIUS Server is Considered Down

Make sure that the product of `cfgWlan802dot1xRetryMax` and `cfgWlan802dot1xRetryTimeout` does not exceed the STA connection timeout of 2 seconds.

Applies to AP.

Range	1 - 10
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.10.1.19

8.1.19.2.1.6 cfgWlan802dot1xName

Name of the Virtual Wireless Interface

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.1.3.10.1.2

8.1.19.2.1.7 cfgWlan802dot1xRetryTimeout

RADIUS Connection Timeout in Milliseconds

Make sure that the product of `cfgWlan802dot1xRetryMax` and `cfgWlan802dot1xRetryTimeout` does not exceed the STA connection timeout of 2 seconds.

Applies to AP.

Range	100 - 10000
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.10.1.20

8.1.19.2.1.8 cfgWlan802dot1xCiphers

OpenSSL Cipher String for the RADIUS 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` is **eap(6)**, **eap2(10)** or **eap192(11)**.

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
- DEFAULT:!EXP:!LOW

Applies to STA.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.10.1.21

8.1.19.2.1.9 cfgWlan802dot1xPrimaryTestMode

Primary RADIUS Authentication Server Test Mode

- **status(0)**: Send Status-Server messages (RADIUS message code 12)
- **access(1)**: Send Access-Request messages (RADIUS message code 1)

Note: Status-Server messages are experimental and might not be supported by all RADIUS authentication servers.

Applies to AP.

Enumeration	status (0), access (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.10.1.22

8.1.19.2.1.10 cfgWlan802dot1xCrlExpiryExtension

CRL Validity Period Extension in Days

If set, the validity period of a CRL can be extended by the given amount of days.

- **0**: no extension
- **1-1095**: number of extension days
- **-1**: extend to infinity => ignore CRL expiry

Applies to STA.

Range	-1 - 1095
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.10.1.24

8.1.19.2.1.11 cfgWlan802dot1xCalds

Certificate Authority (CA) IDs

Reference to the CA which should be used.

Multiple CAs may be referenced by writing the ids of the CAs as space and/or comma separated list. The order of the list is the order how the CAs will be concatenated.

Setting the CA ID to `-1` disables the CA verification.

Examples:

- `-1`
- `12`
- `1, 3, 4`
- `1 3 4`

Applies to STA.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.10.1.25

8.1.19.2.1.12 cfgWlan802dot1xClientCertId

Client Certificate ID

Reference to the Client Certificate which should be used.

Applies to STA.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.10.1.27

8.1.19.2.1.13 cfgWlan802dot1xDynamicVlan

Dynamic VLAN

This parameter is only active when the wlan interface is bridged using `cfgNetWlanBridge`. Dynamic VLAN can not be used together with 802.11r (`cfgWlan802dot11rEnabled`).

Allow the RADIUS authentication server to decide which VLAN is used for the stations. This information is parsed from following RADIUS attributes based on RFC 3580 and RFC 2868: Tunnel-Type (value 13 = VLAN), Tunnel-Medium-Type (value 6 = IEEE 802) Tunnel-Private-Group-ID (value VLANID as a string).

- **disabled(0)**: No dynamic VLANs are used
- **optional(1)**: Use default interface if the RADIUS server does not include a VLAN ID
- **required(2)**: Reject authentication if the RADIUS server does not include a VLAN ID

Applies to AP.

Enumeration	disabled (0), optional (1), required (2)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.10.1.28

8.1.19.2.1.14 cfgWlan802dot1xOwnIpAddr

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.

See also `cfgWlan802dot1xNasId`.

Applies to AP.

Type	IpAddress
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.10.1.3

8.1.19.2.1.15 cfgWlan802dot1xAuthServerParameter

Reference ID to the RADIUS Auth Server Table

Uses all auth servers in the cfgWlan802dot1xAuthServerTable which have as cfgWlan802dot1xAuthSrvId the value set here.

Applies to AP.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.10.1.4

8.1.19.2.1.16 cfgWlan802dot1xAcctServerParameter

Reference ID to the RADIUS Acct Server Table

Uses all acct servers in the cfgWlan802dot1xAcctServerTable which have as cfgWlan802dot1xAcctSrvId the value set here.

Applies to AP.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.10.1.5

8.1.19.2.1.17 cfgWlan802dot1xRetryPrimaryInterval

Retry Interval to Return to the Primary RADIUS Server in Seconds

The RADIUS client automatically tries to use the next server when the current server is not replying to requests. If this interval is set, the primary server is retried after the configured amount of time even if the currently used secondary server is still working.

Set to 0 to disable.

Applies to AP.

Range	0 - 86400
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.10.1.6

8.1.19.2.1.18 cfgWlan802dot1xInterimAccountingInterval

Interim Accounting Update Interval in Seconds

If this is set to a value larger than 0 and at least one accounting server is configured in cfgWlan802dot1xAcctServ hostapd will send interim accounting updates every N seconds.

Set to 0 to disable.

This value should not be less than 600 (10 minutes) and may not be less than 60 (1 minute).

Note: When this value is set, this overrides possible Acct-Interim-Interval attribute in Access-Accept message. Thus, this value should not be configured when the RADIUS server is used to control the interim interval.

Applies to AP.

Range	0 - 86400
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.10.1.7

8.1.19.2.1.19 cfgWlan802dot1xNasId

NAS Identifier for RADIUS messages or 802.11r

It is mandatory to configure either `cfgWlan802dot1xOwnIpAddr` or `cfgWlan802dot1xNasId` to be compliant with the RADIUS protocol. When using RADIUS accounting, it is strongly recommended that `cfgWlan802dot1xNasId` is set to a unique value for each AP.

When using IEEE 802.11r (`cfgWlan802dot11rEnabled`), this value is mandatory and it must be unique per AP.

When set to `autoMAC` the unique MAC address of the respective wlan interface will be used.

Applies to AP.

Type	DisplayString
Range	1 - 48
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.10.1.8

8.1.19.2.1.20 cfgWlan802dot1xEapType

EAP Type for the RADIUS Client

- **tls(0)**: Transport Layer Security Uses the configured CA(s), Certificate and Key referenced by `cfgWlan802dot1xCaIds` and `cfgWlan802dot1xClientCertId`
- **peap(1)**: Protected Extensible Authentication Protocol Uses the Identity and Password specified in `cfgWlan802dot1xIdentity` and `cfgWlanIfacePassword`. Additionally uses the CA(s) referenced in `cfgWlan802dot1xCaIds` to verify the identity of the RADIUS server. When no CA is specified, a rogue 3rd party AP will be able to steal your credentials.

- **ttls(2)**: Tunneled Transport Layer Security Uses the Identity and Password specified in `cfgWlan802dot1xIdentity` and `cfgWlanIfacePassword`. Additionally uses the CA(s) referenced in `cfgWlan802dot1xCaIds` to verify the identity of the RADIUS server. When no CA is specified, a rogue 3rd party AP will be able to steal your credentials.

Applies to STA.

Enumeration	tls (0), peap (1), ttls (2)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.10.1.9

8.1.19.3 cfgWlan802dot1xAuthServerTable

Wired and Wireless 802dot1x AuthServer

Range	0 - 255
OID	1.3.6.1.4.1.16177.1.400.1.1.3.11

8.1.19.3.1 cfgWlan802dot1xAuthServerTableEntry

Wired and Wireless 802dot1x AuthServer Entry

OID	1.3.6.1.4.1.16177.1.400.1.1.3.11.1

8.1.19.3.1.1 cfgWlan802dot1xAuthSrvIndex

Table Entry Index

Range	0 - 255
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.3.11.1.1

8.1.19.3.1.2 cfgWlan802dot1xAuthSrvEnabled

Disable or Enable this Entry in the Authentication Server List

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.11.1.2

8.1.19.3.1.3 cfgWlan802dot1xAuthSrvId

ID of the Authentication Server Table

The configuration item `cfgWlan802dot1xAuthServerParameter` and `cfgNetEth802dot1xAuthServerParame` references to this ID.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.11.1.3

8.1.19.3.1.4 cfgWlan802dot1xAuthSrvIpAddr

IP of the RADIUS Server Against Which Will be Authenticated

Type	IpAddress
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.11.1.4

8.1.19.3.1.5 cfgWlan802dot1xAuthSrvPort

Port of the RADIUS Server Against Which Will be Authenticated

The default RADIUS autentication port is 1812.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.11.1.5

8.1.19.3.1.6 cfgWlan802dot1xAuthSrvSharedSecret

Password to Connect to the Specified RADIUS Server

The shared secret is used to allow the authenticator to communicate with the server.

This password is not used to authenticate clients.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.11.1.6

8.1.19.4 cfgWlan802dot1xAcctServerTable

Wireless 802dot1x AcctServer

Range	0 - 255
OID	1.3.6.1.4.1.16177.1.400.1.1.3.12

8.1.19.4.1 cfgWlan802dot1xAcctServerTableEntry

Wireless 802dot1x AcctServer entry

OID	1.3.6.1.4.1.16177.1.400.1.1.3.12.1

8.1.19.4.1.1 cfgWlan802dot1xAcctSrvIndex

Table Entry Index

Range	0 - 255
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.3.12.1.1

8.1.19.4.1.2 cfgWlan802dot1xAcctSrvEnabled

Enable this entry in the acct server list.

Applies to AP.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.12.1.2

8.1.19.4.1.3 cfgWlan802dot1xAcctSrvId

ID of the accounting server table

The configuration item `cfgWlan802dot1xAcctServerParameter` refers to this ID.

Applies to AP.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.12.1.3

8.1.19.4.1.4 cfgWlan802dot1xAcctSrvIpAddr

IP of the RADIUS accounting server Set to 0.0.0.0 to disable.

Applies to AP.

Type	IpAddress
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.12.1.4

8.1.19.4.1.5 cfgWlan802dot1xAcctSrvPort

Port of the RADIUS accounting server

Applies to AP.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.12.1.5

8.1.19.4.1.6 cfgWlan802dot1xAcctSrvSharedSecret

Password to connect to the specified RADIUS accounting server

Applies to AP.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.12.1.6

8.1.19.5 cfgWlan802dot11rTable

Wireless 802dot11r

Range	0 - 63
OID	1.3.6.1.4.1.16177.1.400.1.1.3.13

8.1.19.5.1 cfgWlan802dot11rTableEntry

Wireless 802dot11r Entry

OID	1.3.6.1.4.1.16177.1.400.1.1.3.13.1

8.1.19.5.1.1 cfgWlan802dot11rIndex

Table Entry Index

Range	0 - 63
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.3.13.1.1

8.1.19.5.1.2 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.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.13.1.10

8.1.19.5.1.3 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.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.13.1.11

8.1.19.5.1.4 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 `cfgWlan802dot11rExpiratio`

Applies to AP. 802.11n products only.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.13.1.12

8.1.19.5.1.5 cfgWlan802dot11rExpirationTime

Daily PMK-R0/-R1 expiration time

Define time (hour:minute) at which PMK-R0s and PMK-R1s expiration is daily forced (if `cfgWlan802dot11rExpira` 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.

Type	DisplayString
Range	5 - 5
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.13.1.13

8.1.19.5.1.6 cfgWlan802dot11rVlan

The 802.1q VLAN tag of 802.11r backbone traffic

Defines the VLAN tag with which all 802.11r managment 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.

Range	0 - 4094
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.13.1.14

8.1.19.5.1.7 cfgWlan802dot11rName

Name of the virtual wireless interface

Applies to AP and STA. 802.11n products only.

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.1.3.13.1.2

8.1.19.5.1.8 cfgWlan802dot11rEnabled

Enable usage of 802.11r on this device

When **enabled(1)** on an AP, `cfgWlan802dot1xNasId` must also be defined.

Applies to AP and STA. 802.11n products only.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.13.1.3

8.1.19.5.1.9 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.

Type	DisplayString
Range	4 - 4
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.13.1.4

8.1.19.5.1.10 cfgWlan802dot11rPmkR0KeyHolderIdentifier

PMK-R0 Key Holder identifier (dot11FTR0KeyHolderID)

Configure this in the field `cfgWlan802dot1xNasId`.

Applies to AP. 802.11n products only.

Type	DisplayString
Range	1 - 48
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.1.3.13.1.5

8.1.19.5.1.11 cfgWlan802dot11rPmkR0Lifetime

PMK-RO lifetime in seconds (dot11FTR0KeyLifetime)

When Session-Timeout attribute is provided by RADIUS server, then min(Session-Timeout, `cfgWlan802dot11rPm` 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.

Range	0 - 2147483647
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.13.1.6

8.1.19.5.1.12 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 000000000000 means use own mac address (bssid).

Format: `020102030405`

Applies to AP. 802.11n products only.

Type	DisplayString
Range	12 - 12
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.13.1.7

8.1.19.5.1.13 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.

Enumeration	donotpush (0), push (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.13.1.9

8.1.19.6 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.

Range	0 - 511
OID	1.3.6.1.4.1.16177.1.400.1.1.3.14

8.1.19.6.1 cfgWlan802dot11rR0KHTableEntry

Wireless 802dot11r R0KH Entry

OID	1.3.6.1.4.1.16177.1.400.1.1.3.14.1

8.1.19.6.1.1 cfgWlan802dot11rR0KHIndex

Table Entry Index

Range	0 - 511
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.3.14.1.1

8.1.19.6.1.2 cfgWlan802dot11rR0KHId

ID of the R0KH table

The configuration item `cfgWlan802dot11rR0KHParameter` references to this ID.

Applies to AP. 802.11n products only.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.14.1.2

8.1.19.6.1.3 cfgWlan802dot11rR0KHEnabled

Enable this entry in the R0KH list

Applies to AP. 802.11n products only.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.14.1.3

8.1.19.6.1.4 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.

Type	DisplayString
Range	17 - 17
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.14.1.4

8.1.19.6.1.5 cfgWlan802dot11rR0KHHID

NAS Identifier of all R0KHs to map to the MAC address

See the field: `cfgWlan802dot11rPmkR0KeyHolderIdentifier`.

Applies to AP. 802.11n products only.

Type	DisplayString
Range	1 - 48
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.14.1.5

8.1.19.6.1.6 cfgWlan802dot11rR0KHKey

Static Key of the R0KH

Connecting R1KHs need to have this key configured.

Format: `000102030405060708090a0b0c0d0e0f`

Applies to AP. 802.11n products only.

Type	DisplayString
Range	32 - 32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.14.1.6

8.1.19.7 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.

Range	0 - 511
OID	1.3.6.1.4.1.16177.1.400.1.1.3.15

8.1.19.7.1 cfgWlan802dot11rR1KHTableEntry

Wireless 802dot11r R1KH Entry

OID	1.3.6.1.4.1.16177.1.400.1.1.3.15.1

8.1.19.7.1.1 cfgWlan802dot11rR1KHIndex

Table Entry Index

Range	0 - 511
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.3.15.1.1

8.1.19.7.1.2 cfgWlan802dot11rR1KHId

ID of the R1KH table

The configuration item `cfgWlan802dot11rR1KHParameter` references to this ID.

Applies to AP.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.15.1.2

8.1.19.7.1.3 cfgWlan802dot11rR1KHEnabled

Enable this entry in the R1KH list

Applies to AP. 802.11n products only.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.15.1.3

8.1.19.7.1.4 cfgWlan802dot11rR1KHDestinationMac

MAC addresses of R1KHs which can reqest PMK-R1 from the local R0KH

Format: `02:01:02:03:04:05`

Applies to AP.

Type	DisplayString
Range	17 - 17
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.15.1.4

8.1.19.7.1.5 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.

Type	DisplayString
Range	17 - 17
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.15.1.5

8.1.19.7.1.6 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.

Type	DisplayString
Range	32 - 32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.15.1.6

8.1.19.8 cfgWlanNeighbourTable

Hostapd Neighbour Table

Range	0 - 511
OID	1.3.6.1.4.1.16177.1.400.1.1.3.16

8.1.19.8.1 cfgWlanNeighbourTableEntry

Hostapd Neighbour Table Entry

OID	1.3.6.1.4.1.16177.1.400.1.1.3.16.1

8.1.19.8.1.1 cfgWlanNeighbourIndex

Table Entry Index

Range	0 - 511
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.3.16.1.1

8.1.19.8.1.2 cfgWlanNeighbourId

ID of the neighbour table

The configuration item `cfgWlanIfaceNeighbourParameter` references to this ID.

Applies to AP. 802.11n products only.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.16.1.2

8.1.19.8.1.3 cfgWlanNeighbourEnabled

Enable this entry in the list

Applies to AP. 802.11n products only.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.16.1.3

8.1.19.8.1.4 cfgWlanNeighbourBSSID

BSSID (MAC address) of the neighbour

Format: `00:14:5a:02:10:42`

Applies to AP.

Type	DisplayString
Range	17 - 17
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.16.1.4

8.1.19.8.1.5 cfgWlanNeighbourFrequency

Frequency in MHz of the neighbour

Applies to AP. 802.11n products only.

Range	1 - 6000
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.16.1.5

8.1.19.9 cfgWlanInterfaceTable

Wireless Virtual Interfaces

Range	0 - 63
OID	1.3.6.1.4.1.16177.1.400.1.1.3.2

8.1.19.9.1 cfgWlanInterfaceTableEntry

Wireless Virtual Interface Entry

OID	1.3.6.1.4.1.16177.1.400.1.1.3.2.1

8.1.19.9.1.1 cfgWlanIfaceIndex

Table Entry Index

Range	0 - 63
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.1

8.1.19.9.1.2 cfgWlanIfaceDtim

Number of beacons between transmission of DTIM element

This attribute specifies the number of beacon intervals that shall elapse between transmission of beacons 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.

Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.10

8.1.19.9.1.3 cfgWlanIfaceApIsolate

Enable AP clients isolation

If enabled, clients connected to this AP can't communicate to each other.

Applies to AP.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.100

8.1.19.9.1.4 cfgWlanIfaceBitrates

Fixed MCS Index For 802.11n Rates

Set to `-1` to not force an MCS index (auto-rate).

Allows multiple space and/or comma separated indices which are then used in auto rate.

This entry is only active when `cfgWlanDevModulation` is set to **ng(10)** or **na(12)**

Examples:

- -1
- 0 1 2 3 4 5 6 7
- 0, 4, 7, 8, 12, 15

Applies to AP and STA. 802.11n products only.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.11

8.1.19.9.1.5 cfgWlanIfaceBeaconInterval

Beacon Frame Generation Interval

Time in `kus` (TU = 1.024 ms) between the sending of beacon frames.

Example:

- 100 = 102.4ms

Applies to AP.

Range	15 - 65535
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.12

8.1.19.9.1.6 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 frames are sent in the order in which the VLANs are specified.

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.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.120

8.1.19.9.1.7 cfgWlanIfaceAtfSsidEnabled

ATF SSID Disabled or Enabled

When this parameter is set to **disabled(0)** no airtime is reserved for the SSID.

Applies to AP. 802.11ax and 802.11be products only.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.121

8.1.19.9.1.8 cfgWlanIfaceAtfSsidAirtime

ATF SSID Airtime Percentage

Specify the percentage of each second of airtime assigned to the SSID on this virual interface. The sum of all assigned airtime shares on the same physical device must not exceed 100%.

Applies to AP. 802.11ax and 802.11be products only.

Range	0 - 100
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.122

8.1.19.9.1.9 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.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.13

8.1.19.9.1.10 cfgWlanIfacePilotMode

Pilot Frame Generation Mode

If set to **enabled(1)**, Pilot frames are always generated. In the **auto(2)** mode, Pilot frames are only generated when at least one station is connected.

Applies to AP. 802.11n products only.

Enumeration	disabled (0), enabled (1), auto (2)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.130

8.1.19.9.1.11 cfgWlanIfacePilotInterval

Pilot Frame Generation Interval

Time in `kus` (TU = 1.024 ms) between the sending of Pilot frames. The generation of the Pilot frames is clocked with a resolution of 4ms and the interval should be divisible by 4. The scheduler is initiated by the `cfgWlanIfaceBeaconInterval`, which means that the Pilot interval must be smaller than the Beacon interval. In the following example, the AP sends 4 Pilots between each Beacon.

Example:

- Beacon Interval: 150 = 153.600ms
- Pilot Interval: 32 = 32.768ms

Applies to AP. 802.11n products only.

Range	0 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.131

8.1.19.9.1.12 cfgWlanIfaceMulticastToUnicast

Multicast to Unicast Conversion

If set to **enabled(1)**, the AP is requested to perform multicast to unicast conversion for ARP, IPv4 and IPv6 frames. If enabled, such frames are to be sent to each station separately, with the DA replaced by their own MAC address rather than the group address.

Note: This may break certain expectations of the receiver, such as the ability to drop unicast IP packets received within multicast L2 frames, or the ability to not send ICMP destination unreachable messages for packets received in L2 multicast.

Applies to AP. 802.11n products only.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.134

8.1.19.9.1.13 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.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.14

8.1.19.9.1.14 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.

Range	-2 - 23
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.15

8.1.19.9.1.15 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.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.16

8.1.19.9.1.16 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. The local Accept and Deny list are searched first and take priority.

Applies to AP.

Enumeration	acceptunlessdeny (0), denyunlessaccept (1), radius (2)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.17

8.1.19.9.1.17 cfgWlanIfaceMaxNumSta

Maximum Number of Clients

802.11ac products have an upper limit of 512 clients. Additionally on the wave 1 card (radio1) only 60 simultaneous clients are supported.

Applies to AP.

Range	1 - 2007
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.18

8.1.19.9.1.18 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.

Type	DisplayString
Range	17 - 17
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.19

8.1.19.9.1.19 cfgWlanIfaceName

Name of the virtual wireless interface

Applies to AP and STA.

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.2

8.1.19.9.1.20 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.

Range	0 - 2048
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.20

8.1.19.9.1.21 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.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.21

8.1.19.9.1.22 cfgWlanIfaceInactivityTimeout

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.

Range	15 - 65535
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.23

8.1.19.9.1.23 cfgWlanIfaceUseVendorSsid

Enable Vendor Element containing the SSID

When `cfgWlanIfaceIgnoreBroadcastSsid` 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.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.26

8.1.19.9.1.24 cfgWlanIfaceDevice

Maps the virtual wireless interface to the radio device

Applies to AP and STA.

Enumeration	radio0 (0), radio1 (1), radio2 (2)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.3

8.1.19.9.1.25 cfgWlanIfaceIeee80211w

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. 802.11n products only.

Enumeration	disabled (0), optional (1), required (2)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.30

8.1.19.9.1.26 cfgWlanIfaceIeee80211wMaxTimeout

802.11w Management Frame Protection (MFP) timeout

Association SA query maximum timeout in `kus` (TU = 1.024 ms; for MFP). This is the maximum time to wait for a SA query response.

Applies to AP.

Range	1 - 4000
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.31

8.1.19.9.1.27 cfgWlanIfaceIeee80211wRetryTimeout

802.11w Management Frame Protection (MFP) retry timeout

Association SA query retry timeout in `kus` (TU = 1.024 ms; for MFP) This is the time between two subsequent SA query requests.

Applies to AP.

Range	1 - 4000
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.32

8.1.19.9.1.28 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.

Enumeration	ap (0), sta (1), monitor (2), mesh (3)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.4

8.1.19.9.1.29 cfgWlanIfaceAcsList

Index to specify a frequency list from `cfgWlanFreqTable`

Used for Automated Channel Selection (ACS) and for channel switch on Radar event to specify a list of frequencies to be chosen from.

To disable set to **-1**.

Applies to AP.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.40

8.1.19.9.1.30 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.

Type	DisplayString
Range	1 - 32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.5

8.1.19.9.1.31 cfgWlanIfaceEncryption

Wireless Encryption Mode

Supported encryption modes:

- **open(0)** open network without encryption
- **psk(3)** WPA2 Personal with PSK
- **eap(6)** WPA2 Enterprise (TLS/TTLS/PEAP)
- **sae(7)** WPA3 Personal with SAE
- **owe(8)** Opportunistic Wireless Encryption (WPA3 Enhanced Open)
- **saepsk(9)** WPA2/WPA3 Personal transition mode SAE+PSK
- **eap2(10)** WPA3 Enterprise (TLS/TTLS/PEAP).
- **eap192(11)** WPA3 Enterprise 192-bit (TLS/TTLS/PEAP). 802.11n products only

eap(6), **eap2(10)** and **eap192(11)** enable 802.1X support.

Applies to AP and STA.

Enumeration	open (0), psk (3), eap (6), sae (7), owe (8), saepsk (9), eap2 (10), eap192 (11)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.6

8.1.19.9.1.32 cfgWlanIfaceNeighbourReport

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.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.60

8.1.19.9.1.33 cfgWlanIfaceNeighbourParameter

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.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.61

8.1.19.9.1.34 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.

Type	DisplayString
Range	8 - 63
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.7

8.1.19.9.1.35 cfgWlanIfacePmkLifetime

Maximum Lifetime for PMKSA in Seconds

If this parameter is set to '-1', it automatically uses a value depending on the encryption defined in `cfgWlanIfaceEncryption`:

For SAE(7) or SAEPSK(9) a disabled timeout is simulated by setting `cfgWlanIfacePmkLifetime` to 2147483647, which gives a lifetime of 68+ years.

For all other encryptions, cfgWlanIfacePmkLifetime is set to 86460.

If this parameter is set to '0', a lifetime of 68+ years is used.

Note: With 802.11r, the used lifetime is the smaller value of `cfgWlanIfacePmkLifetime` and `cfgWlan802dot11rPmkR0Lifetime`.

Applies to AP and STA.

Range	-1 - 2147483647
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.70

8.1.19.9.1.36 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 country code restricts active scanning it is automatically set to passive.

If the scanning mode is set to passive(1) the station will always perform passive scanning to detect available access points. Since the listening time is related to the beacon period used at the AP, the passive scan mode `cfgWlanHoPassiveChanTime` (STA) must be at least as long as the maximum beacon period `cfgWlanIfaceBeaconInterval` (AP).

Applies to STA. 802.11n products only.

Enumeration	active (0), passive (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.8

8.1.19.9.1.37 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.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.80

8.1.19.9.1.38 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.

Enumeration	both (0), arp (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.81

8.1.19.9.1.39 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.

Type	DisplayString
Range	17 - 17
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.82

8.1.19.9.1.40 cfgWlanIfaceBeaconMiss

Number of consecutive beacons misses before the station disconnects

Applies to STA. 802.11n products only.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.9

8.1.19.9.1.41 cfgWlanIfaceTimeAdvertisement

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.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.2.1.90

8.1.19.10 cfgWlanHandoffTable

Wireless handoff parameters

Range	0 - 63
OID	1.3.6.1.4.1.16177.1.400.1.1.3.3

8.1.19.10.1 cfgWlanHandoffTableEntry

Wireless Handoff Parameters Entry

OID	1.3.6.1.4.1.16177.1.400.1.1.3.3.1

8.1.19.10.1.1 cfgWlanHoIndex

Table Entry Index

Range	0 - 63
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.3.3.1.1

8.1.19.10.1.2 cfgWlanHoFilterLongX

IIR Low-Pass Filter Parameter X for RSSI measurements.

Parameter X is the weighting for previous filtered RSSI value.

Applies to STA. 802.11n products only.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.3.1.10

8.1.19.10.1.3 cfgWlanHoFilterLongY

IIR Low-Pass Filter Parameter Y for RSSI measurements.

The parameter Y is the weighting for RSSI current value.

Applies to STA. 802.11n products only.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.3.1.11

8.1.19.10.1.4 cfgWlanHoScanRateLimitTime

****OBSOLETE:** Scan Rate Limit Time**

This option used to be active when `cfgWlanHoProfile` was set to t2gv1(1). t2gv1(1) is no longer supported.

Range	4 - 65535
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.3.3.1.12

8.1.19.10.1.5 cfgWlanHoScanRateLimitTries

****OBSOLETE:** Scan Rate Limit Tries**

This option used to be active when `cfgWlanHoProfile` was set to t2gv1(1). t2gv1(1) is no longer supported.

Range	1 - 255
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.3.3.1.13

8.1.19.10.1.6 cfgWlanHoPassiveChanTime

Time in milliseconds (4ms steps) we stay on a channel during passive scanning and wait for beacons. Must be at least as long as the maximum beacon period `cfgWlanIfaceBeaconInterval` used at the AP.

Applies to STA. 802.11n products only.

Range	1 - 65535
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.3.1.16

8.1.19.10.1.7 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.

Note: On a STA, the RSSI handoff in T2Gv3 is only active if a handoff level low greater than 0 is defined.

Applies to AP and STA. 802.11n products only.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.3.1.18

8.1.19.10.1.8 cfgWlanHoLevelHigh

Scanning Level High in RSSI

This value is only active when `cfgWlanHoProfile` is set to T2Gv3.

When the RSSI level to the current connected AP is higher or equal this value, perform a handoff to the next AP.

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.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.3.1.19

8.1.19.10.1.9 cfgWlanHoIfaceName

Name of the virtual wireless interface

Applies to STA. 802.11n products only.

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.1.3.3.1.2

8.1.19.10.1.10 cfgWlanHoDistanceNear

Distance Near

This value is only active when `cfgWlanHoProfile` is set to T2Gv3.

When the measured distance (units) to the current connected AP is lower or equal this value, perform a handoff to the next AP.

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.

Range	0 - 114750
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.3.1.20

8.1.19.10.1.11 cfgWlanHoDistanceFar

Distance Far

This value is only active when `cfgWlanHoProfile` is set to T2Gv3.

When the measured distance (units) to the current connected AP is greater or equal this value, perform a handoff to the next AP.

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.

Range	0 - 114750
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.3.1.21

8.1.19.10.1.12 cfgWlanHoDistanceMeasurementPeriod

Distance Ranging Measurement Period

This value is only active when `cfgWlanHoProfile` is set to T2Gv3.

The period is defined in milliseconds.

Setting a value of zero disables distance handoff.

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.

Range	0 - 100000
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.3.1.22

8.1.19.10.1.13 cfgWlanHoDistanceFilterX

IIR Low-Pass Filter Parameter X for Distance Measurements

This value is only active when `cfgWlanHoProfile` is set to T2Gv3.

Parameter X is the weighting for the previous filtered distance value.

Applies to STA. 802.11n products only.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.3.1.23

8.1.19.10.1.14 cfgWlanHoDistanceFilterY

IIR Low-Pass Filter Parameter Y for Distance Measurements

This value is only active when `cfgWlanHoProfile` is set to T2Gv3.

Parameter Y is the weighting for the current distance value.

Applies to STA. 802.11n products only.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.3.1.24

8.1.19.10.1.15 cfgWlanHoProfile

Handoff Profile

A description of available handoff profiles in terms of use case, functionality and associated parameters.

t2gv2(2) Train to Ground v2 (background scan)

Performs a background scan that optimizes operation in ISO-frequency systems. If neighbor reporting is enabled on the AP `cfgWlanIfaceNeighbourReport`, the STA will scan directly. Intended for operation with third-party products.

- Encryption: psk/eap/sae (WPA2/WPA3)
- Scanning: Background
- Neighbour reporting: Frequency
- Associated Parameters: `cfgWlanHoScanningLevel`

t2gv2fg(3) Train to Ground v2 (foreground scan)

Performs a foreground scan that optimizes operation in multi-frequency systems. If neighbor reporting is enabled on the AP `cfgWlanIfaceNeighbourReport`, the STA will scan directly. Intended for operation with third-party products.

- Encryption: psk/eap/sae (WPA2/WPA3)
- Scanning: Foreground
- Neighbour reporting: Frequency
- Associated Parameters: `cfgWlanHoScanningLevel`

t2gv3(4) Train to Ground v3 (RSSI low/high, Distance near/far)

If neighbor reporting is enabled on the AP `cfgWlanIfaceNeighbourReport`, the STA will scan directly and set the handoff parameters according to the received AP values. Otherwise, or if a

neighbour report is lost, the handoff values for RSSI and Distance defined on the STA are used as backup. To enable distance handoff, the measurement period must be set.

Note: The RSSI handoff in t2gv3(4) is only active if a `cfgWlanHoLevelLow` greater than 0 is defined.

- Encryption: psk/eap/sae (WPA2/WPA3)
- Scanning: Foreground
- Neighbour reporting: Frequency, RSSI low/high, Distance near/far
- Associated Parameters: `cfgWlanHoLevelLow`, `cfgWlanHoLevelHigh`, `cfgWlanHoDistanceNear`, `cfgWlanHoDistanceFar`, `cfgWlanHoDistanceMeasurementPeriod`

Applies to STA and AP. 802.11n products only.

Enumeration	t2gv2 (2), t2gv2fg (3), t2gv3 (4)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.3.1.3

8.1.19.10.1.16 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 cfgWlanIfaceScanList and cfgWlanFreqTable.

Note: This parameter is inactive when using `cfgWlanHoProfile` t2gv3(4). Define the thresholds in `cfgWlanHoLevelLow` and `cfgWlanHoLevelHigh` instead.

Applies to STA. 802.11n products only.

Range	0 - 95
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.3.1.5

8.1.19.10.1.17 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 `cfgWlanHoBeacons * cfgWlanIfaceBeaconInterv` before doing another handoff.

Applies to STA. 802.11n products only.

Range	4 - 20
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.3.1.6

8.1.19.10.1.18 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.

Range	0 - 2000
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.3.1.7

8.1.19.11 cfgWlanRogueApDetection

8.1.19.11.1 cfgWlanRapdServiceEnabled

TECHPREVIEW: Enable Rogue AP Detection (RAPD)

This feature enables the rogue AP detection to monitor potential security issues. Rogue APs are defined as foreign ones which preted to provide one AP's own SSID.

This is a global enable flag that activates the detection service, while each interface to be monitored is defined in cfgWlanRapdTable and can be enabled individually.

The detection is available for 802.11ax and 802.11be APs.

Applies to AP.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.30.1

8.1.19.11.2 cfgWlanRapdTable

RAPD Table

Range	0 - 15
OID	1.3.6.1.4.1.16177.1.400.1.1.3.30.10

8.1.19.11.2.1 cfgWlanRapdTableEntry

RAPD Table Entry

OID	1.3.6.1.4.1.16177.1.400.1.1.3.30.10.1

cfgWlanRapdIndex

Index of the Table Entry

Range	0 - 15
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.3.30.10.1.1

cfgWlanRapdEnabled

TECHPREVIEW: Enable this Instance of RAPD Detection

Applies to AP.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.30.10.1.2

cfgWlanRapdInterface

TECHPREVIEW: WLAN Interface Index for RAPD

An index in `cfgWlanInterfaceTable` may be referenced.

Applies to AP.

Range	0 - 63
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.30.10.1.3

cfgWlanRapdCheckInterval

TECHPREVIEW: Check Interval for RAPD Instance in Seconds

This defines the interval of the scan processes to look for potential rogue APs in seconds.

Note that scanning for RAPs comes at a cost, since the AP temporarily leaves its operating channel and during that time can't provide services to stations. Therefore, the check interval shall be chosen as a trade-off between a sufficient detection probability and an acceptable service degradation.

Applies to AP.

Range	1 - 86400
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.30.10.1.4

cfgWlanRapdScanFreqList

TECHPREVIEW: Scan frequency list to use for this RAPD instance

This is the index of the `cfgWlanFreqTable` that shall be used to scan for rogue APs. The chosen table is to be configured with the frequencies that should be monitored.

Setting the value to `-2` scans all frequencies possibly in the configured country code and allowed by the radio.

Setting the value to `-1` scans only the configured frequency on which the currently radio operates.

Note: scanning many frequencies increases service degradation, since the AP interrupts its operation for the scan process and the off-channel time is proportional to the amount of frequencies scanned. Therefore, a full scan of all available frequencies is not advised.

Applies to AP.

Range	-2 - 23
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.30.10.1.5

cfgWlanRapdAllowedMacParameter

TECHPREVIEW: MAC Addresses Ignored as Rogue APs

This is a reference ID to the `cfgWlanRapdMacTable`. Uses all referenced parameters which have as `cfgWlanRapdMacId` the value set here.

This allows to have an installation with multiple own APs providing the same SSID and allowing those. When no MAC addresses are referenced, any AP other using the own SSID will be detected as rogue.

A value of `-1` means standlone mode.

Applies to AP.

Range	-1 - 63
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.30.10.1.6

cfgWlanRapdSsidParameter

TECHPREVIEW: SSIDs to Scan for Rogue APs

This is a reference ID to the `cfgWlanRapdSsidTable`. Uses all referenced parameters which have as `cfgWlanRapdSsidId` the value set here.

All referenced SSIDs are monitored for rogue APs. When this value is set to `-1` the SSID to scan for is the interface's AP SSID.

Setting arbitrary SSIDs allows for one AP to do the detection work for a different AP. Assume an installation where a company has a public and a private wlan network, of which the private is not allowed to have service interruption. The AP providing the public network can be configured to scan for both SSIDs.

Applies to AP.

Range	-1 - 63
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.30.10.1.7

8.1.19.11.3 cfgWlanRapdMacTable

Rogue Access Point Detection Allowed MAC Addresses Table

Range	0 - 255
OID	1.3.6.1.4.1.16177.1.400.1.1.3.30.11

8.1.19.11.3.1 cfgWlanRapdMacTableEntry

Rogue Access Point Detection Allowed MAC Addresses Table Entry

OID	1.3.6.1.4.1.16177.1.400.1.1.3.30.11.1

cfgWlanRapdMacIndex

Index of the Table Entry

Range	0 - 255
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.3.30.11.1.1

cfgWlanRapdMacId

TECHPREVIEW: ID of the RAPD MAC Address Table

The configuration item `cfgWlanRapdAllowedMacParameter` references to this ID.

Applies to AP.

Range	0 - 63
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.30.11.1.2

cfgWlanRapdMacEnabled

TECHPREVIEW: Disable or Enable This Entry in the MAC List

Applies to AP.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.30.11.1.3

cfgWlanRapdMacAddr

TECHPREVIEW: MAC Address to be Allowed in RAPD

Format: `00:14:5a:09:01:23`

Applies to AP.

Type	DisplayString
Range	17 - 17
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.30.11.1.4

8.1.19.11.4 cfgWlanRapdSsidTable

Rogue Access Point Detection SSID Table

Range	0 - 63
OID	1.3.6.1.4.1.16177.1.400.1.1.3.30.12

8.1.19.11.4.1 cfgWlanRapdSsidTableEntry

Rogue Access Point Detection SSID Table Entry

OID	1.3.6.1.4.1.16177.1.400.1.1.3.30.12.1

cfgWlanRapdSsidIndex

Index of the Table Entry

Range	0 - 63
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.3.30.12.1.1

cfgWlanRapdSsidId

TECHPREVIEW: ID of the RAPD SSID List

The configuration item `cfgWlanRapdSsidParameter` references to this ID.

Applies to AP.

Range	0 - 63
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.30.12.1.2

cfgWlanRapdSsidEnabled

TECHPREVIEW: Disable or Enable This Entry in the SSID List

Applies to AP.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.30.12.1.3

cfgWlanRapdSsidSsid

TECHPREVIEW: Rogue SSID to Scan for

Applies to AP.

Type	DisplayString
Range	1 - 32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.30.12.1.4

8.1.19.12 cfgWlanFreqTable

Frequency list entry

Range	0 - 23
OID	1.3.6.1.4.1.16177.1.400.1.1.3.4

8.1.19.12.1 cfgWlanFreqTableEntry

Frequency list entry

OID	1.3.6.1.4.1.16177.1.400.1.1.3.4.1

8.1.19.12.1.1 cfgWlanFIndex

Table Entry Index

Range	0 - 23
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.3.4.1.1

8.1.19.12.1.2 cfgWlanFFreq8

Frequency in MHz, 0 is interpreted as empty

Applies to AP and STA.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.4.1.10

8.1.19.12.1.3 cfgWlanFFreq9

Frequency in MHz, 0 is interpreted as empty

Applies to AP and STA.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.4.1.11

8.1.19.12.1.4 cfgWlanFFreq10

Frequency in MHz, 0 is interpreted as empty

Applies to AP and STA.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.4.1.12

8.1.19.12.1.5 cfgWlanFFreq11

Frequency in MHz, 0 is interpreted as empty

Applies to AP and STA.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.4.1.13

8.1.19.12.1.6 cfgWlanFFreq12

Frequency in MHz, 0 is interpreted as empty

Applies to AP and STA.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.4.1.14

8.1.19.12.1.7 cfgWlanFFreq13

Frequency in MHz, 0 is interpreted as empty

Applies to AP and STA.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.4.1.15

8.1.19.12.1.8 cfgWlanFFreq14

Frequency in MHz, 0 is interpreted as empty

Applies to AP and STA.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.4.1.16

8.1.19.12.1.9 cfgWlanFFreq15

Frequency in MHz, 0 is interpreted as empty

Applies to AP and STA.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.4.1.17

8.1.19.12.1.10 cfgWlanFFreq16

Frequency in MHz, 0 is interpreted as empty

Applies to AP and STA.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.4.1.18

8.1.19.12.1.11 cfgWlanFFreq17

Frequency in MHz, 0 is interpreted as empty

Applies to AP and STA.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.4.1.19

8.1.19.12.1.12 cfgWlanFFreq0

Frequency in MHz, 0 is interpreted as empty

Applies to AP and STA.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.4.1.2

8.1.19.12.1.13 cfgWlanFFreq18

Frequency in MHz, 0 is interpreted as empty

Applies to AP and STA.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.4.1.20

8.1.19.12.1.14 cfgWlanFFreq19

Frequency in MHz, 0 is interpreted as empty

Applies to AP and STA.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.4.1.21

8.1.19.12.1.15 cfgWlanFFreq20

Frequency in MHz, 0 is interpreted as empty

Applies to AP and STA.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.4.1.22

8.1.19.12.1.16 cfgWlanFFreq21

Frequency in MHz, 0 is interpreted as empty

Applies to AP and STA.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.4.1.23

8.1.19.12.1.17 cfgWlanFFreq22

Frequency in MHz, 0 is interpreted as empty

Applies to AP and STA.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.4.1.24

8.1.19.12.1.18 cfgWlanFFreq23

Frequency in MHz, 0 is interpreted as empty

Applies to AP and STA.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.4.1.25

8.1.19.12.1.19 cfgWlanFFreq1

Frequency in MHz, 0 is interpreted as empty

Applies to AP and STA.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.4.1.3

8.1.19.12.1.20 cfgWlanFFreq2

Frequency in MHz, 0 is interpreted as empty

Applies to AP and STA.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.4.1.4

8.1.19.12.1.21 cfgWlanFFreq3

Frequency in MHz, 0 is interpreted as empty

Applies to AP and STA.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.4.1.5

8.1.19.12.1.22 cfgWlanFFreq4

Frequency in MHz, 0 is interpreted as empty

Applies to AP and STA.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.4.1.6

8.1.19.12.1.23 cfgWlanFFreq5

Frequency in MHz, 0 is interpreted as empty

Applies to AP and STA.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.4.1.7

8.1.19.12.1.24 cfgWlanFFreq6

Frequency in MHz, 0 is interpreted as empty

Applies to AP and STA.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.4.1.8

8.1.19.12.1.25 cfgWlanFFreq7

Frequency in MHz, 0 is interpreted as empty

Applies to AP and STA.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.4.1.9

8.1.19.13 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.

Range	0 - 31
OID	1.3.6.1.4.1.16177.1.400.1.1.3.5

8.1.19.13.1 cfgWlanWmeTableEntry

WME table entry

OID	1.3.6.1.4.1.16177.1.400.1.1.3.5.1

8.1.19.13.1.1 cfgWlanWmeIndex

Table Entry Index

Range	0 - 31
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.3.5.1.1

8.1.19.13.1.2 cfgWlanWmeApAifs

Arbitration inter-frame space (AIFS)

Is used on the AP itself.

Applies to AP.

Range	0 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.5.1.10

8.1.19.13.1.3 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.

Range	0 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.5.1.11

8.1.19.13.1.4 cfgWlanWmeId

ID of the WME parameter table

The virtual wireless interface references to this ID, specified by `cfgWlanIfaceWmeParameter`.

Applies to AP. 802.11n products only.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.5.1.2

8.1.19.13.1.5 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.

Enumeration	none (0), background (1), besteffort (2), video (3), voice (4)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.5.1.3

8.1.19.13.1.6 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.

Range	1 - 10
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.5.1.4

8.1.19.13.1.7 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.

Range	1 - 10
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.5.1.5

8.1.19.13.1.8 cfgWlanWmeAifs

Arbitration inter-frame space (AIFS)

Is used on STAs connected to this AP.

Applies to AP. 802.11n products only.

Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.5.1.6

8.1.19.13.1.9 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.

Range	0 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.5.1.7

8.1.19.13.1.10 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.

Range	1 - 1023
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.5.1.8

8.1.19.13.1.11 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.

Range	1 - 1023
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.5.1.9

8.1.19.14 cfgWlanDbgTable

Wireless Handoff Debug Parameters

Range	0 - 63
OID	1.3.6.1.4.1.16177.1.400.1.1.3.6

8.1.19.14.1 cfgWlanDbgTableEntry

Wireless Handoff Debug Parameters Entry

OID	1.3.6.1.4.1.16177.1.400.1.1.3.6.1

8.1.19.14.1.1 cfgWlanDbgIndex

Table Entry Index

Range	0 - 63
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.3.6.1.1

8.1.19.14.1.2 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.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.6.1.10

8.1.19.14.1.3 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.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.6.1.11

8.1.19.14.1.4 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.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.6.1.12

8.1.19.14.1.5 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.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.6.1.13

8.1.19.14.1.6 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.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.6.1.14

8.1.19.14.1.7 cfgWlanDbgIfaceName

Name of the virtual wireless interface.

Applies to STA. 802.11n products only.

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.1.3.6.1.2

8.1.19.14.1.8 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.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.6.1.3

8.1.19.14.1.9 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.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.6.1.4

8.1.19.14.1.10 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.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.6.1.5

8.1.19.14.1.11 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.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.6.1.6

8.1.19.14.1.12 cfgWlanDbgBeaconrssi

Persistent default value to enable/disable the Beacon and Pilot RSSI messages in commissioning syslog.

These log messages are subject to change. DO NOT PARSE!

Applies to STA. 802.11n products only.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.6.1.7

8.1.19.14.1.13 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.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.6.1.8

8.1.19.14.1.14 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.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.6.1.9

8.1.19.15 cfgWlanAclWhiteTable

Wireless MAC Access Control Whitelist

Range	0 - 63
OID	1.3.6.1.4.1.16177.1.400.1.1.3.7

8.1.19.15.1 cfgWlanAclWhiteTableEntry

Wireless MAC Access Control Whitelist Entry

OID	1.3.6.1.4.1.16177.1.400.1.1.3.7.1

8.1.19.15.1.1 cfgWlanAclWhiteIndex

Table Entry Index

Range	0 - 63
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.3.7.1.1

8.1.19.15.1.2 cfgWlanAclWhiteEnabled

Disable or Enable this Entry in the ACL

Applies to AP.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.7.1.2

8.1.19.15.1.3 cfgWlanAclWhiteInterface

Name of the Wireless Interface

The ACL entry is on this specified wireless interface active.

Applies to AP.

Type	DisplayString
Range	1 - 15
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.7.1.3

8.1.19.15.1.4 cfgWlanAclWhiteAddr

MAC Address in the ACL

Examples:

- `00:14:5a:02:10:42`
- `00:07:7c:00:00:00`

Applies to AP.

Type	DisplayString
Range	17 - 17
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.7.1.4

8.1.19.15.1.5 cfgWlanAclWhiteMask

Mask of the MAC Address

Allows the use of ranges of MAC addresses. To be used like CIDR notation of IP addresses.

Examples:

- To match a single MAC address, specify a mask of 48.
- To match a vendor OUI, specify a mask of 24.

Applies to AP.

Range	1 - 48
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.7.1.5

8.1.19.16 cfgWlanAclBlackTable

Wireless MAC Access Control Blacklist

Range	0 - 63
OID	1.3.6.1.4.1.16177.1.400.1.1.3.8

8.1.19.16.1 cfgWlanAclBlackTableEntry

Wireless MAC Access Control Blacklist Entry

OID	1.3.6.1.4.1.16177.1.400.1.1.3.8.1

8.1.19.16.1.1 cfgWlanAclBlackIndex

Table Entry Index

Range	0 - 63
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.3.8.1.1

8.1.19.16.1.2 cfgWlanAclBlackEnabled

Disable or Enable This Entry in the ACL

Applies to AP.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.8.1.2

8.1.19.16.1.3 cfgWlanAclBlackInterface

Name of the Wireless Interface

The ACL entry is on this specified wireless interface active.

Applies to AP.

Type	DisplayString
Range	1 - 15
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.8.1.3

8.1.19.16.1.4 cfgWlanAclBlackAddr

MAC Address in the ACL

Examples:

- `00:14:5a:02:10:42`
- `00:07:7c:00:00:00`

Applies to AP.

Type	DisplayString
Range	17 - 17
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.8.1.4

8.1.19.16.1.5 cfgWlanAclBlackMask

Mask of the MAC Address

Allows the use of ranges of MAC addresses. To be used like CIDR notation of IP addresses.

Examples:

- To match a single MAC address, specify a mask of 48.
- To match a vendor OUI, specify a mask of 24.

Applies to AP.

Range	1 - 48
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.8.1.5

8.1.19.17 cfgWlanGlobal

8.1.19.17.1 cfgWlanGlblCountry

Wireless country code

Note: Refer to documentation which countries are supported for your device.

Applies to AP and STA.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.9.1

8.1.19.17.2 cfgWlanGlblLinkmonitorInterval

LinkMonitor interval in milliseconds

A new trap is sent at each interval.

Note: A short interval and/or numerous connections may affect system performance negatively.

Applies to AP and STA. 802.11n products only.

Range	200 - 60000
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.9.2

8.1.19.17.3 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.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.9.3

8.1.19.17.4 cfgWlanGlblConnectionStatusWlanInterface

Persistent Default of the Volatile Setting `swDrvConStatWlanIf`

The value set here is used to initialize `swDrvConStatWlanIf`. Initialisation happens on startup or configuration change.

Specify `all` to get the connection status of all wlan interfaces.

Applies to AP and STA.

Type	DisplayString
Range	3 - 17
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.9.4

8.1.19.17.5 cfgWlanGlblAclRejectLog

ACL reject logging

If enabled, hostapd will log stations rejected based on MAC ACL.

Applies to AP.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.3.9.5

8.1.20 cfgRouting

8.1.20.1 cfgRouteDefault

8.1.20.1.1 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.

The default gateway specified here is always configured on routing table 254 with a metric of 0.

Type	IpAddress
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.4.1.1

8.1.20.2 cfgRouteTable

Static Routes

Range	0 - 265
OID	1.3.6.1.4.1.16177.1.400.1.1.4.2

8.1.20.2.1 cfgRouteTableEntry

Static Route

OID	1.3.6.1.4.1.16177.1.400.1.1.4.2.1

8.1.20.2.1.1 cfgRouteTableIndex

Table Entry Index

Range	0 - 265
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.4.2.1.1

8.1.20.2.1.2 cfgRouteTableMetric

Metric of the Route

The metric allows to create multiple routes to the same destination. The higher a metric, the less priority the route has.

Range	0 - 2147483647
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.4.2.1.10

8.1.20.2.1.3 cfgRouteTableRoutingTables

Routing Tables on Which Route are Created

This is a space and/or comma separated list of tables on which the route will be created.

Specify 254 to create routes on the `main` table.

Valid values are > 0 and < 2000000000. However in this range there are reserved values that may not be used:

- 128: prelocal
- 253: default
- 255: local

Use `cfgRouteRuleTable` to create policies which use the tables specified here.

Examples:

- 5000
- 254, 7000
- 100 254, 8000

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.4.2.1.11

8.1.20.2.1.4 cfgRouteTableMonitor

NLM Monitor Handling This Route

This is a reference to an NLM instance (`cfgNlmMonIndex`).

The referenced monitor has to have as actions a value of `8000` or `8001` (see `cfgNlmMonUpAction` and `cfgNlmMonDownAction`).

Set to `-1` to not handle this route by a monitor.

CARP (`cfgNetCarpTable`) and NLM monitoring are exclusive.

Range	-1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.4.2.1.12

8.1.20.2.1.5 cfgRouteTableComment

User Comment

This parameter has no operational function. It allows to store a comment about the use of this route.

Type	DisplayString
Range	0 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.4.2.1.13

8.1.20.2.1.6 cfgRouteTableWeight

Weight for ECMP

When multiple gateways for the same destination exist, connections are distributed to the gateways according to their weight. Connections are distinguished on IP source and IP destination.

Use Case:

- The net 192.168.0.0/24 is reachable via 10.0.1.1 and 172.16.1.1
- 10.0.1.1 has a weight of 1 and 172.16.1.1 has a weight of 2
- When 3 connections are opened, 1 connection is sent via 10.0.1.1 and 2 connections via 172.16.1.1

Range	1 - 256
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.4.2.1.14

8.1.20.2.1.7 cfgRouteTableEnabled

Disable or Enable Route Entry

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.4.2.1.2

8.1.20.2.1.8 cfgRouteTableDestinationNetwork

Destination Network in CIDR Notation

Set to `0.0.0.0/0` to match any destination. This is the equivalent to the default gateway in `cfgRouteDefGateway`.

Type	DisplayString
Range	5 - 50
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.4.2.1.3

8.1.20.2.1.9 cfgRouteTableGateway

Gateway to Destination Network

Specify the IP address of the gateway over which the destination specified in `cfgRouteTableDestinationNetwo` is reachable.

Alternatively options are:

- `unreachable`: Create a route that blocks traffic and responds with ICMP code 1 (Host unreachable)
- `prohibit`: Create a route that blocks traffic and responds with ICMP code 13 (Communication administratively filtered)
- `blackhole`: Create a route that silently drops all traffic
- `throw`: Create a route that skips processing on the current routing table. May be used for policy routing

Type	DisplayString
Range	5 - 50
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.4.2.1.5

8.1.20.2.1.10 cfgRouteTableSource

Source for Traffic to Destination Network

Optional, use only if you have multiple possible sources.

Type	IpAddress
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.4.2.1.6

8.1.20.2.1.11 cfgRouteTableCarpId

The CARP Instance Which Brings the 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.

Range	-1 - 15
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.4.2.1.8

8.1.20.2.1.12 cfgRouteTableInterface

Interface of Static Route

The interface on which the static route is created. Usually this is not required, and the system determines automatically to which interface a route applies to.

Leave this on `none` unless you need an explicit interface for a route. For example, this is required when using IPsec with VTI (see `cfgVpnIpsecGlblVirtualTunnelInterface`).

Type	DisplayString
Range	1 - 15
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.4.2.1.9

8.1.20.3 cfgMRouteTable

Static Multicast Routes

Range	0 - 255
OID	1.3.6.1.4.1.16177.1.400.1.1.4.3

8.1.20.3.1 cfgMRouteTableEntry

Static Multicast Routes

OID	1.3.6.1.4.1.16177.1.400.1.1.4.3.1

8.1.20.3.1.1 cfgMRouteTableIndex

Table Entry Index

Range	0 - 255
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.4.3.1.1

8.1.20.3.1.2 cfgMRouteTableEnabled

Disable or Enable this Multicast Route Entry

Applies to AP and STA.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.4.3.1.2

8.1.20.3.1.3 cfgMRouteTableInput

Input Interface

The interface on which multicast traffic is received.

Type	DisplayString
Range	1 - 15
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.4.3.1.3

8.1.20.3.1.4 cfgMRouteTableSource

Unicast Source Address to Listen for

May be set to a specific address, or to a range in CIDR notation.

If it is set to `0.0.0.0` multicast traffic from all sources is forwarded.

Examples:

- 0.0.0.0
- 192.168.1.15
- 172.16.1.0/24

Type	DisplayString
Range	7 - 18
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.4.3.1.4

8.1.20.3.1.5 cfgMRouteTableGroup

Multicast Group to Forward

Multicast addresses are in the range of ‘224.0.0.0’ to ‘239.255.255.255’.

Type	IpAddress
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.4.3.1.5

8.1.20.3.1.6 cfgMRouteTableOutput

Output Interface(s)

This is a space and/or comma separated list of interfaces from which the forwarded multicast traffic is sent.

Examples:

- br0.vlan0
- wlan0
- eth0, eth1, wlan1

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.4.3.1.6

8.1.20.4 cfgRouteRuleTable

Routing Rules

Also called `policy routing`.

These rules may be used to match frames based on:

- Source (`cfgRouteRuleFrom`)
- Destination (`cfgRouteRuleTo`)
- Input Interface (`cfgRouteRuleInputInterface`)
- TOS (Type of Service, QoS) value (`cfgRouteRuleTos`)
- IP Protocol (`cfgRouteRuleIpProto`)
- Source Port (UDP and TCP, `cfgRouteRuleSourcePort`)
- Destination Port (UDP and TCP, `cfgRouteRuleDestinationPort`)

These frames are then processed with the specified routing table in `cfgRouteRuleLookupTable`.

Each rule should have its own unique preference set via `cfgRouteRulePreference`. Thus multiple overlapping rules may be created.

Range	0 - 255
OID	1.3.6.1.4.1.16177.1.400.1.1.4.4

8.1.20.4.1 cfgRouteRuleTableEntry

Routing Rule

OID	1.3.6.1.4.1.16177.1.400.1.1.4.4.1

8.1.20.4.1.1 cfgRouteRuleIndex

Table Entry Index

Range	0 - 255
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.4.4.1.1

8.1.20.4.1.2 cfgRouteRuleSourcePort

Source Port

When the frames are TCP or UDP, match on their source port.

May be a single number or a range of ports.

Examples:

- 53
- 443
- 1000-2000
- 8000-8001

Set to `any` to ignore this field.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.4.4.1.10

8.1.20.4.1.3 cfgRouteRuleDestinationPort

Destination Port

When the frames are TCP or UDP, match on their destination port.

May be a single number or a range of ports.

Examples:

- 53

- 443
- 1000-2000
- 8000-8001

Set to `any` to ignore this field.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.4.4.1.11

8.1.20.4.1.4 cfgRouteRuleLookupTable

Routing Table to Lookup

Frames which match the configured criteria are processed by the routing table specified here.

The default routing table is table 254 - the main table. All traffic not explicitly directed to a different table is processed by the main table.

Reserved routing table numbers may not be used and are:

- 128: prelocal
- 253: default
- 255: local

Range	1 - 1999999999
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.4.4.1.12

8.1.20.4.1.5 cfgRouteRuleMark

Mark

Frames may be marked with a mark in the `cfgFwMangleTable`. Use this parameter to match frames with the specified mark.

Set to `-1` to disable matching on mark.

The maximum mark value is `4294967295`.

Type	DisplayString
Range	1 - 10
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.4.4.1.13

8.1.20.4.1.6 cfgRouteRuleEnabled

Disable or Enable this Rule

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.4.4.1.2

8.1.20.4.1.7 cfgRouteRulePreference

Preference of the Rule

The preference defines in what order rules are evaluated. This enables multiple overlapping rules. The lower the value, the higher the priority.

Each match should have it's own unique preference.

Range	10000 - 1999999999
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.4.4.1.3

8.1.20.4.1.8 cfgRouteRuleFrom

Source Network

This is an address or network in CIDR notation.

Set to `0.0.0.0/0` to match any source.

Type	DisplayString
Range	5 - 50
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.4.4.1.4

8.1.20.4.1.9 cfgRouteRuleTo

Destination Network

This is an address or network in CIDR notation.

Set to `0.0.0.0/0` to match any destination.

Type	DisplayString
Range	5 - 50
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.4.4.1.5

8.1.20.4.1.10 cfgRouteRuleInputInterface

Input Interface

Match an interface on which traffic is received. If the interface is the loopback `lo`, only traffic originating on the device itself is matched. Thus different rules for local traffic and forwarding traffic can be created.

Set to `any` to match any interface.

Type	DisplayString
Range	1 - 15
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.4.4.1.6

8.1.20.4.1.11 cfgRouteRuleTos

Type Of Service

Match frames based on the TOS field in their IP header according to RFC 791.

```
|  7  |  6  |  5  |  4  |  3  |  2  |  1  |  0  |
+-----+-----+-----+-----+-----+-----+-----+-----+
|   PRECEDENCE    |       TOS       | Unused(0) |
+-----+-----+-----+-----+-----+-----+-----+-----+
```

Thus valid values are:

- 0
- 4
- 8
- c
- 10
- 14
- 18
- 1c

Set to `any` to match any TOS value.

Type	DisplayString
Range	1 - 3
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.4.4.1.8

8.1.20.4.1.12 cfgRouteRuleIpProto

IP Protocol

Match frames based on the IP protocol in their IP header.

Common protocols are:

- **1**: ICMP
- **2**: IGMP
- **6**: TCP
- **17**: UDP
- **50**: ESP (IPsec)
- **51**: AH (IPsec)
- **112**: VRRP / CARP

Set to -1 to ignore this field.

Range	-1 - 254
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.4.4.1.9

8.1.20.5 cfgRouteDhcpTable

DHCP Received Routes

This table may be used to specify the metric and table of routes received via DHCP. When using ECMP also the weight may be changed. To dynamically bring the routes up and down, a monitor may be specified.

Range	0 - 255
OID	1.3.6.1.4.1.16177.1.400.1.1.4.5

8.1.20.5.1 cfgRouteDhcpTableEntry

DHCP Received Routes

OID	1.3.6.1.4.1.16177.1.400.1.1.4.5.1

8.1.20.5.1.1 cfgRouteDhcpIndex

Table Entry Index

Range	0 - 255
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.4.5.1.1

8.1.20.5.1.2 cfgRouteDhcpEnabled

Disable or Enable This Entry

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.4.5.1.2

8.1.20.5.1.3 cfgRouteDhcpInterface

Interface of DHCP Client

The name of an interface on which a DHCP Client is running.

Examples:

- wlan0
- br0.vlan7

Type	DisplayString
Range	1 - 15
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.4.5.1.3

8.1.20.5.1.4 cfgRouteDhcpMetric

Metric of Routes Received

The metric allows to create multiple routes to the same destination. The higher a metric, the less priority the route has.

Set to -1 to not explicitly set a metric. Depending on the DHCP client a different metric is implied:

- **wwan**: 600
- **eth**: 400
- **wlan**: 400
- **ovpn**: 200

Range	-1 - 2147483647
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.4.5.1.4

8.1.20.5.1.5 cfgRouteDhcpRoutingTables

Routing Tables on Which Received Routes are Created

This is a space and/or comma separated list of tables on which the route will be created.

Specify 254 to create routes on the `main` table.

Valid values are > 0 and < 2000000000. However in this range there are reserved values that may not be used:

- 128: prelocal
- 253: default
- 255: local

Use `cfgRouteRuleTable` to create policies which use the table specified here.

Examples:

- 5000
- 254, 7000
- 100 254, 8000

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.4.5.1.5

8.1.20.5.1.6 cfgRouteDhcpMonitor

NLM Monitor Handling of Routes Received

This is a reference to an NLM instance (`cfgNlmMonIndex`).

The referenced monitor has to be of type **route(3)** or **logic(5)** when the logic monitor has a route monitor in its dependency tree (see `cfgNlmMonType`).

Set to `-1` to not handle this route by a monitor.

Range	-1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.4.5.1.6

8.1.20.5.1.7 cfgRouteDhcpWeight

Weight for ECMP of Routes Received

When multiple gateways for the same destination exist, connections are distributed to the gateways according to their weight. Connections are distinguished on IP source and IP destination.

Use Case:

- The net 192.168.0.0/24 is reachable via 10.0.1.1 and 172.16.1.1
- 10.0.1.1 has a weight of 1 and 172.16.1.1 has a weight of 2
- When 3 connections are opened, 1 connection is sent via 10.0.1.1 and 2 connections via 172.16.1.1

Range	1 - 256
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.4.5.1.7

8.1.20.6 cfgRouteGlobal

8.1.20.6.1 cfgRouteGlblReversePathFilter

Reverse Path Filter

The reverse path filter has 3 modes of operation:

- **none(0)**: No source address validation is performed and any packet is forwarded to the destination network.
- **strict(1)**: Strict Mode as defined in RFC 3704. Each incoming packet to a router is tested against the routing table and if the interface that the packet is received on is not the best return path for the packet then the packet is dropped.
- **loose(2)**: Loose mode as defines in RFC 3704 Loose Reverse Path. Each incoming packet is tested against the route table and the packet is dropped if the source address is not routable through any interface. This allows for asymmetric routing where the return path may not be the same as the source path.

Enumeration	none (0), strict (1), loose (2)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.4.6.2

8.1.21 cfgNlm

8.1.21.1 cfgNlmGlobal

8.1.21.1.1 cfgNlmGlblEnabled

Enable the Network Link Monitor

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.40.1.1

8.1.21.2 cfgNlmMonitorTable

NLM Monitor Table

Range	0 - 255
OID	1.3.6.1.4.1.16177.1.400.1.1.40.2

8.1.21.2.1 cfgNlmMonitorTableEntry

NLM Monitor Table Entry

OID	1.3.6.1.4.1.16177.1.400.1.1.40.2.1

8.1.21.2.1.1 cfgNlmMonIndex

Table Entry Index

Range	0 - 255
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.40.2.1.1

8.1.21.2.1.2 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 are:

- **1xxx**: Offset: 1000, x: CARP group from 0 to 255. Un-demote CARP group defined by `cfgNetCarpLocalInterfaceGroup`.
- **20xx**: Offset: 2000, x: WLAN interface from 0 to 63. Enable Access Point operation on the wlan interface. Note: On 802.11n products all wlan interfaces on radio0 are enabled, regardless of the specified WLAN interface.
- **4xxx**: Offset: 4000, x: Wireguard Peer Index from 0 to 255. Re-resolve the specified FQDN of the referenced peer.

- **8000**: Bring up all routes which reference to this NLM instance. References are defined via `cfgRouteTableMonitor` and `cfgRouteDhcpMonitor`.
- **8001**: Bring down all routes which reference to this NLM instance. References are defined via `cfgRouteTableMonitor` and `cfgRouteDhcpMonitor`.

Range	0 - 9999
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.40.2.1.10

8.1.21.2.1.3 cfgNlmMonDownAction

NLM 'DOWN' State Monitor Action

The action is executed on a monitor-state transition to 'down'.

Set to `0` to disable (i.e. no action).

Supported actions are:

- **1xxx**: Offset: 1000, x: CARP group from 0 to 255. Demote CARP group defined by `cfgNetCarpLocalInterfaceGroup`.
- **20xx**: Offset: 2000, x: wlan interface from 0 to 63. Disable Access Point operation on the wlan interface. Note: On 802.11n products all wireless interfaces on radio0 are disabled, regardless of the specified WLAN interface.
- **4xxx**: Offset: 4000, x: Wireguard Peer Index from 0 to 255. Re-resolve the specified FQDN of the referenced peer.
- **8000**: Bring down all routes which reference to this NLM instance. References are defined via `cfgRouteTableMonitor` and `cfgRouteDhcpMonitor`.
- **8001**: Bring up all routes which reference to this NLM instance. References are defined via `cfgRouteTableMonitor` and `cfgRouteDhcpMonitor`.

Range	0 - 9999
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.40.2.1.11

8.1.21.2.1.4 cfgNlmMonScanLoopInterval

Scan Loop Debounce Interval In Milliseconds

This parameter is active when `cfgNlmMonType` is set to **wlan(2)**.

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.

Range	100 - 2147483647
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.40.2.1.12

8.1.21.2.1.5 cfgNlmMonCountUp

NLM Monitor Count for UP Transition

The number of times the measured criteria has to be up, until the monitor is reported as up.

This parameter is used for polling based monitor types only (see `cfgNlmMonType`).

Range	1 - 65535
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.40.2.1.13

8.1.21.2.1.6 cfgNlmMonRssi

NLM Monitor RSSI Threshold

This parameter is active when `cfgNlmMonType` is set to **rssi(4)**.

This defines the threshold where an RSSI monitor indicates a down condition if the current rssi value is below this number and up otherwise.

Range	0 - 127
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.40.2.1.14

8.1.21.2.1.7 cfgNlmMonLogic

NLM Logic Monitor Subtype

This parameter is active when `cfgNlmMonType` is set to **logic(5)**.

This parameter defines the logical operation on the input(s) defined in `cfgNlmMonLogicInput`.

When the executed action fails, the action is retried at the interval specified in `cfgNlmMonInterval`.

Supported operations are:

- **none(0)**: Monitor is ignored
- **equal(1)**: The input defined is selected as is, this is essentially an alias to an existing monitor, allowing to define additional up/down actions

- **not(2)**: This is the inversion of equal(1), i.e. it references the configured monitor input, but with inverted state
- **or(3)**: The state is the OR combined state of referenced inputs, i.e. if one of them is UP, this monitor is UP
- **and(4)**: This is the AND equivalent for or(3), i.e. if one of the referenced inputs is DOWN, this monitor is DOWN
- **nor(5)**: The state is the NOR combined state of referenced inputs, i.e. if one of them is UP, this monitor is DOWN
- **nand(6)**: This is the NAND equivalent for nor(5), i.e. if one of the referenced inputs is DOWN, this monitor is UP

Enumeration	none (0), equal (1), not (2), or (3), and (4), nor (5), nand (6)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.40.2.1.15

8.1.21.2.1.8 cfgNlmMonLogicInput

NLM Input(s) for Logic Monitor

This parameter is active when `cfgNlmMonType` is set to **logic(5)**.

Specifies the monitor input(s) as space and/or comma-separated indices.

Examples:

- 1
- 2 3,4

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.40.2.1.16

8.1.21.2.1.9 cfgNlmMonTrap

NLM Trap Sending

This allows monitors to issue traps on state changes.

Supported operations are:

- **none(0)**: No traps are sent out
- **up(1)**: Trap 340 is sent out when monitor state gets UP
- **down(2)**: Trap 341 is sent out when monitor state gets DOWN
- **both(3)**: Both traps 340 and 341 are sent out

Enumeration	none (0), down (1), up (2), both (3)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.40.2.1.17

8.1.21.2.1.10 cfgNlmMonRouter

NLM Monitor Router

This parameter is active when `cfgNlmMonType` is set to **route(3)**.

It specifies an IPv4 gateway on the interface defined in `cfgNlmMonInterfaces` over which the destination in `cfgNlmMonDestination` is reachable.

The content of this field may be dynamically supplied by a DHCP client in `cfgRouteDhcpTable` that references this monitor. When referenced as such, then the specified interface will be overwritten by the DHCP client supplied interface.

Type	IpAddress
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.40.2.1.18

8.1.21.2.1.11 cfgNlmMonEnabled

Disable or Enable this Entry in the Monitor List

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.40.2.1.2

8.1.21.2.1.12 cfgNlmMonInterval

Execution Interval Of This Monitor In Milliseconds

The minimum interval is 100ms.

Polling based monitors check their condition at each interval.

Event based monitors have a different meaning depending on the value of `cfgNlmMonType`:

- **wlan(2)**: defines the long handoff timeout.
- **logic(5)** defines the retry-interval when the action fails.

Range	0 - 2147483647
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.40.2.1.3

8.1.21.2.1.13 cfgNlmMonCount

NLM Monitor Count for DOWN Transition

The number of times the measured criteria has to be down, until the monitor is reported as down.

This parameter is used for polling based monitor types only (see `cfgNlmMonType`).

Range	1 - 65535
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.40.2.1.4

8.1.21.2.1.14 cfgNlmMonType

Objects That 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. This is a polling based monitor.

- **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. This is a polling based monitor.

- **wlan(2)** monitor listens to link status events of the wireless interface defined by `cfgNlmMonInterfaces`.
  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 `cfgNlmMonScanLoopInterv`
  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.

- **route(3)** monitor is the same as **icmp(1)**, but it binds the source interface statically, which is provided by a DHCP client, or configured manually with `cfgNlmMonInterfaces`, `cfgNlmMonDestination` and `cfgNlmMonRouter`. When supplied by a DHCP client this is set by referencing a monitor with `cfgRouteDhcpMonitor`. When used to control routes it must be used with actions `8000` and `8001` to bring up and tear down routes dynamically. The action may be set to `0` if it is referenced by a logic monitor in `cfgNlmMonLogicInput` that sets up the routes. When `cfgNlmMonDestination` is configured to `0.0.0.0`, this monitor is dynamically set to the default gateway received by the DHCP client. When a specific IP is set, this IP will be checked instead. This is only necessary when the gateway does not respond to ICMP requests. This is a polling based monitor.

- **rssi(4)** monitors the RSSI value of the wlan interface defined in `cfgNlmMonInterfaces`. A value below the threshold defined in `cfgNlmMonRssi` indicates a DOWN status. This is a polling based monitor.

- **logic(5)** monitors reference existing monitors and allow logical combinations of those. This type of monitor requires `cfgNlmMonLogic` to select the logical operation, along with `cfgNlmMonLogicInput` to define the monitors as input. A logic monitor may reference only a single monitor of type **route(3)** if that monitor is referenced by a DHCP client. This is an event based monitor.

Note: wlan(2) monitor is supported for 802.11n products only.

Enumeration	phy (0), icmp (1), wlan (2), route (3), rssi (4), logic (5)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.40.2.1.5

8.1.21.2.1.15 cfgNlmMonInterfaces

NLM Monitor Interface(s)

This parameter is active when `cfgNlmMonType` is set to **phy(0)**, **wlan(2)**, **route(3)** or **rssi(4)**.

This parameter specifies the name of the interface(s) that are monitored.

Depending on the content of `cfgNlmMonType`:

- **phy(0):** List of space and/or comma separated interfaces
- **wlan(2):** A single wlan interface (802.11n products only)
- **route(3):** A single interface. When this monitor is referenced by an entry in `cfgRouteDhcpTable`, this field will be overwritten by the DHCP client supplied interface. When the monitor is referenced by an entry in `cfgRouteTable`, the interface must have an IP address assigned.
- **rssi(4):** A single wlan interface (802.11n products only)

Examples:

- eth0, eth1
- br0.vlan0, br1.vlan200
- wlan0

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.40.2.1.6

8.1.21.2.1.16 cfgNlmMonDestination

NLM Monitor Destination

This parameter is active when `cfgNlmMonType` is set to **icmp(1)** or **route(3)**.

This parameter specifies the IPv4 address that is monitored.

Type	IpAddress
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.40.2.1.7

8.1.22 cfgQos

8.1.22.1 cfgQosL3PrioEnabled

Layer 3 Prioritization

Controls IP Precedence based priority assignments.

Actual prioritization on the wireless link only occurs if `cfgQosWmeEnabled` is enabled as well.

The L3 prioritization is only applicable to bridge interfaces such as `br0.vlan0` on a bridged AP or `br1.vlan0` on a routed STA. The IP address of the handled network must not be assigned to a physical wireless interface like `wlan0`.

Applies to AP and STA. 802.11n products only.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.6.1

8.1.22.2 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.

Range	0 - 7
OID	1.3.6.1.4.1.16177.1.400.1.1.6.2

8.1.22.2.1 cfgQosDscpToTidMapTableEntry

QoS DSCP To TID Mapping Table Entry

OID	1.3.6.1.4.1.16177.1.400.1.1.6.2.1

8.1.22.2.1.1 cfgQosDscpToTidMapTableIndex

QoS DSCP To TID Mapping Table Index

Range	0 - 7
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.6.2.1.1

8.1.22.2.1.2 cfgQosDscpToTidMapValue

Layer 2 Priorities For IP Precedence Values 0-7

Applies to AP and STA. 802.11n products only.

Range	0 - 7
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.6.2.1.2

8.1.22.3 cfgQosVlanToTidMapTable

802.1p To TID Map

Mapping table from layer 2 priorities (802.1p) to wireless priority (TID).

Range	0 - 7
OID	1.3.6.1.4.1.16177.1.400.1.1.6.3

8.1.22.3.1 cfgQosVlanToTidMapTableEntry

QoS VLAN To TID Priority Table Entry

OID	1.3.6.1.4.1.16177.1.400.1.1.6.3.1

8.1.22.3.1.1 cfgQosVlanToTidMapTableIndex

Table Entry Index

Range	0 - 7
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.6.3.1.1

8.1.22.3.1.2 cfgQosVlanToTidMapValue

Layer 2 Priorities For VLAN Priorities 0-7.

Applies to AP and STA. 802.11n products only.

Range	0 - 7
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.6.3.1.2

8.1.22.4 cfgQosIpToTidMapTable

IP Header To TID Map

Mapping table from IP header (Source, Destination, Protocol, Port), to wireless priority (TID).

Range	0 - 127
OID	1.3.6.1.4.1.16177.1.400.1.1.6.4

8.1.22.4.1 cfgQosIpToTidMapTableEntry

IP To TID Priority Table Entry

OID	1.3.6.1.4.1.16177.1.400.1.1.6.4.1

8.1.22.4.1.1 cfgQosIpToTidMapTableIndex

Priority of IP To TID Rule

A higher value represents a higher priority. This is important when multiple rules have overlapping matches.

Example:

- Rule 0: Match 0.0.0.0/0 as source and 0.0.0.0/0 as destination
- Rule 1: Match 172.16.0.0/12 as source and 0.0.0.0/0 as destination

Since Rule 1 has a higher index, the match on 172.16.0.0/12 takes precedence over Rule 0 with a lower index.

Range	0 - 127
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.6.4.1.1

8.1.22.4.1.2 cfgQosIpToTidMapSrcNet

Source Network For IP Prioritization Rule

In CIDR format.

Applies to AP and STA. 802.11n products only.

Type	DisplayString
Range	5 - 50
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.6.4.1.2

8.1.22.4.1.3 cfgQosIpToTidMapDestNet

Destination Network For IP Prioritization Rule

In CIDR format.

Applies to AP and STA. 802.11n products only.

Type	DisplayString
Range	5 - 50
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.6.4.1.3

8.1.22.4.1.4 cfgQosIpToTidMapProto

Deprecated. Use `cfgQosIpToTidMapProtoFull`

Enumeration	any (0), udp (1), tcp (2)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.6.4.1.4

8.1.22.4.1.5 cfgQosIpToTidMapSrcPort

Source Port For IP Prioritization Rule

Use port `-1` to match any port. This setting can only be used if the IP protocol `cfgQosIpToTidMapProtoFull` is set to **6(tcp))** or **17(udp)**.

Applies to AP and STA. 802.11n products only.

Range	-1 - 65535
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.6.4.1.5

8.1.22.4.1.6 cfgQosIpToTidMapDestPort

Destination Port For IP Prioritization Rule

Use port `-1` to match any port. This setting can only be used if the IP protocol `cfgQosIpToTidMapProtoFull` is set to **6(tcp))** or **17(udp)**.

Applies to AP and STA. 802.11n products only.

Range	-1 - 65535
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.6.4.1.6

8.1.22.4.1.7 cfgQosIpToTidMapPrecedence

Precedence To Set For IP Prioritization Rule

Applies to AP and STA. 802.11n products only.

Range	0 - 7
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.6.4.1.7

8.1.22.4.1.8 cfgQosIpToTidMapEnabled

Disable Or Enable IP Prioritisation Rule

Applies to AP and STA. 802.11n products only.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.6.4.1.8

8.1.22.4.1.9 cfgQosIpToTidMapProtoFull

Protocol for IP Prioritization Rule

When set to `-2` uses the deprecated `cfgQosIpToTidMapProto`.

Set to `-1` to match any protocol. Otherwise matches the specified IP protocol.

Examples:

- **-1**: any protocol
- **1**: ICMP
- **2**: IGMP
- **6**: TCP
- **17**: UDP
- **50**: ESP (IPsec)
- **51**: AH (IPsec)
- **112**: VRRP / CARP

For a full list of available protocols see https://en.wikipedia.org/wiki/List_of_IP_protocol_numbers

Applies to AP and STA. 802.11n products only.

Range	-2 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.6.4.1.9

8.1.22.5 cfgQosDefaultTid

Default TID For Frames That Do Not Match Any Other Rule

The default TID is set to to frames to be transmitted that do not match the mode to which `cfgQosL3PrioEnabled` is set and there is no matching entry in `cfgQosEthertypeToL2Table`. This means when `cfgQosL3PrioEnabled` is set to **disabled(0)**, that the frame is not a VLAN frame. And when `cfgQosL3PrioEnabled` is set to **enabled(1)**, that the frame is not an IP frame.

Applies to AP and STA. 802.11n products only.

Range	0 - 7
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.6.5

8.1.22.6 cfgQosEthertypeToL2Table

Set TID For Specific Ethertype

The ethertype rules are applied, when the frame to be transmitted does not match the mode to which `cfgQosL3PrioEnabled` is set. This means when `cfgQosL3PrioEnabled` is set to **disabled(0)**, that the frame is not a VLAN frame. And when `cfgQosL3PrioEnabled` is set to **enabled(1)**, that the frame is not an IP frame.

Applies to AP and STA. 802.11n products only.

Range	0 - 127
OID	1.3.6.1.4.1.16177.1.400.1.1.6.6

8.1.22.6.1 cfgQosEthertypeToL2TableEntry

Ethertype To TID Table Entry

OID	1.3.6.1.4.1.16177.1.400.1.1.6.6.1

8.1.22.6.1.1 cfgQosEthertypeToL2TableIndex

IP To TID Priority Table Index

Range	0 - 127
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.1.6.6.1.1

8.1.22.6.1.2 cfgQosEthertypeToL2Enabled

Disable Or Enable Ethertype Rule

Applies to AP and STA. 802.11n products only.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.6.6.1.2

8.1.22.6.1.3 cfgQosEthertypeToL2Ethertype

Ethertype To Match

Some popular Ethertypes:

- 0800: IPv4
- 0806: ARP
- 0835: RARP
- 8100: VLAN
- 86DD: IPv6
- 8847: MPLS unicast
- 8848: MPLS multicast
- 8892: Profinet
- 9100: stacked VLAN

Applies to AP and STA. 802.11n products only.

Type	DisplayString
Range	4 - 4
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.6.6.1.3

8.1.22.6.1.4 cfgQosEthertypeToL2Tid

TID To Set For Ethertype

Applies to AP and STA. 802.11n products only.

Range	0 - 7
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.6.6.1.4

8.2 rpc

8.2.1 rpcConfiguration

8.2.1.1 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.

Enumeration	errorAll (-1), nop (0), all (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.3.1.1

8.2.1.2 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 then 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.

Enumeration	errorAll (-1), nop (0), all (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.3.1.2

8.2.1.3 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.

Enumeration	errorImport (-2), errorExport (-1), nop (0), export (1), import (2)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.3.1.4

8.2.2 rpcScep

8.2.2.1 rpcScepTable

SCEP RPCs

Range	0 - 255
OID	1.3.6.1.4.1.16177.1.400.1.3.100.1

8.2.2.1.1 rpcScepTableEntry

SCEP RPCs

OID	1.3.6.1.4.1.16177.1.400.1.3.100.1.1

8.2.2.1.1.1 rpcScepIndex

Table Entry Index

Range	0 - 255
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.3.100.1.1.1

8.2.2.1.1.2 rpcScepGetCaCrt

Start SCEP getca for SCEP entry in `cfgScepTable`

Enumeration	error (-1), done (0), start (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.3.100.1.1.2

8.2.2.1.1.3 rpcScepEnroll

SCEP enroll/re-enroll

Writing this value will:

- **start(1)** start SCEP enroll
- **reenroll(3)** start SCEP re-enroll
- **stop(2)** stop SCEP enroll or re-enroll

for corresponding SCEP entry in `cfgScepTable`.

Writing **reenroll(3)** to this value will start SCEP re-enroll process for corresponding SCEP entry in `cfgScepTable`.

Reading this value will return:

- **start(1)** as long as the enrollment is in process
- **reenroll(3)** as long as the re-enrollment is in process
- **done(0)** when enrollment or re-enrollment has finished
- **eerror(-1)** when enrollment failed
- **serror(-2)** when stopping enrollment or re-enrollment failed
- **reerror(-3)** when re-enrollment failed

Enumeration	reerror (-3), serror (-2), eerror (-1), done (0), start (1), stop (2), reenroll (3)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.3.100.1.1.3

8.2.3 rpcCellular

8.2.3.1 rpcCellModuleInit

Initialize Cellular Module

This initialization must be performed once to enable QMI (Qualcomm MSM Interface). The configuration will be stored in non-volatile memory.

Read the status from `hwCellModuleInitialized` to determine if initialization is required.

- **init(1):** Initialize module. After the module has been initialized, a restart is required for the module to become active.

Enumeration	nop (0), init (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.3.101.2

8.2.3.2 rpcCellFwUpgrade

Upgrade Cellular Module

Before starting a firmware upgrade of the cellular module, define a valid URL, which is accessible by the device. This URL can be specified by `setCellFwUrl`.

Writing **upgrade(1)** to this parameter downloads the firmware from the specified URL and upgrades the cellular module.

This RPC parameter follows the standard behaviour, thus a readout indicates the status of the upgrade process. Whilst the upgrade is running, **upgrade(1)** is returned. Otherwise, it returns **nop(0)** or **error(-1)** to indicate that the upgrade is complete or failed.

The device has to be rebooted after successfully flashing for the upgrade to take effect.

Enumeration	error (-1), nop (0), upgrade (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.3.101.3

8.2.3.3 rpcCellFotaUpgrade

FOTA Upgrade Cellular Module

The Firmware Over The Air (FOTA) upgrade for the cellular module checks for the availability of a new firmware package and installs it.

Check the `hwCellFwVersionPreferred` status if a firmware upgrade is required.

By default the firmware package is downloaded from a Westermo web service. If the device operates in a private network without internet access, the URL `setCellFotaUrl` must be set to override the public domain with a local web server.

Please contact support for instructions on how to set up the web server and obtain the required fw packages.

NOTE: The FOTA upgrade requires a cellular connection. The firmware download may incur additional data subscription charges.

Writing **upgrade(1)** to this parameter will download the firmware and put the module into download mode. The upgrade will start automatically if the download is successful. During the download and upgrade, remote access is not available. The unit will reboot automatically and connectivity will be restored.

The parameter returns **error(-1)** if the download fails. After the reboot the parameter will return **nop(0)**. To check that the firmware upgrade is complete, `hwCellFwVersionPreferred` should return **true(1)**.

**IMPORTANT: Do not turn off the unit during the cellular firmware is being upgraded. The upgrade may take several minutes and remote access will be restored when it is complete. If there is a power failure during the upgrade, the module will attempt to restore the previous firmware.

Enumeration	error (-1), nop (0), upgrade (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.3.101.4

8.2.4 rpcPartitions

8.2.4.1 rpcPartGenerateHashes

Generate Hashes of all Partitions

These hashes may be used to assert the integrity of the partitions.

Writing this value will:

- **generate(1)** Generate the partition hash information
- **delete(2)** Erase the partition hash information

Enumeration	nop (0), generate (1), delete (2)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.3.103.1

8.2.5 rpcFirmware

8.2.5.1 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 **errorFlash(-2)** indicates that the flash process failed during writing. A return value of **errorDownload(-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.

Enumeration	errorFlash (-2), errorDownload (-1), nop (0), flash (2), flashWithConfig (3)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.3.2.1

8.2.6 rpcSystem

8.2.6.1 rpcSysReboot

Reboot system after n seconds.

Applies to AP and STA.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.3.3.1

8.2.6.2 rpcSysFactoryReset

Factory Reset

Perform a factory reset and reboot the device. This will reset device configuration, including administrator password and certificates, to its default state.

Writing **reset(1)** the device will reboot after resetting.

Note: You will not be able to communicate with the device until the factory reset has finished and the device has been rebooted.

Writing **resetAndHalt(2)** will halt the device after resetting and shutting down.

Applies to AP and STA.

Enumeration	nop (0), reset (1), resetAndHalt (2)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.3.3.2

8.2.6.3 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.

Enumeration	nop (0), reset (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.3.3.3

8.2.6.4 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 then 0 the process is still running.
- A negative return value (-1) indicates that something went wrong while resetting the kernel logs.

Applies to AP and STA.

Enumeration	errorReset (-1), nop (0), reset (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.3.3.4

8.2.7 rpcCertificate

8.2.7.1 rpcCrtFile

Import, export or delete certificate to or from a file respectively

Please refer to `setCrtFileUrl` for more information on how to set the certificate 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.

Enumeration	errorValidate (-4), errorDelete (-3), errorExport (-2), errorImport (-1), nop (0), import (1), export (2), delete (3)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.3.4.1

8.2.7.2 rpcCrtRefresh

****OBSOLETE:** Refresh Certificates**

This parameter is obsolete and has been replaced with `rpcCfgApply`.

Range	-1 - -1
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.3.4.2

8.2.7.3 rpcCrtCrlTable

Certificate CRL RPCs

Range	0 - 15
OID	1.3.6.1.4.1.16177.1.400.1.3.4.3

8.2.7.3.1 rpcCrtCrlTableEntry

Certificate CRL RPCs entries

OID	1.3.6.1.4.1.16177.1.400.1.3.4.3.1

8.2.7.3.1.1 rpcCrtCrlIndex

Table Entry Index

Range	0 - 15
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.3.4.3.1.1

8.2.7.3.1.2 rpcCrtCrlGet

Import a CA CRL

- **importcrl(1)**: Import the CRL from `cfgCrtCrlUrl` and store it in certificate store to CRL ID corresponding to the `cfgCrtCrlCaId`.

Note: The CA CRL must be in the DER format.

Applies to AP and STA.

Enumeration	errorImportcrl (-1), nop (0), importcrl (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.3.4.3.1.2

8.2.7.4 rpcCrtAttribute

Read/Write a certificate attribute

To read an attribute the `setCrtFileId`, `setCrtFileType` and `setCrtAttributeKey` need to be set. After call this rpc the value is available in `setCrtAttributeValue`.

To write an attribute the `setCrtFileId`, `setCrtFileType`, `setCrtAttributeKey` and `setCrtAttributeValue` need to be set.

Applies to AP and STA.

Enumeration	errorWrite (-2), errorRead (-1), nop (0), read (1), write (2)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.3.4.4

8.2.8 rpcDriver

8.2.8.1 rpcDrvTable

RPC driver module

Range	0 - 1
OID	1.3.6.1.4.1.16177.1.400.1.3.6.1

8.2.8.1.1 rpcDrvTableEntry

RPC driver module

OID	1.3.6.1.4.1.16177.1.400.1.3.6.1.1

8.2.8.1.1.1 rpcDrvIndex

Table Entry Index

Range	0 - 1
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.3.6.1.1.1

8.2.8.1.1.2 rpcDrvName

Name of the Radio Device

Applies to AP.

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.3.6.1.1.2

8.2.8.1.1.3 rpcDrvDfsSimulateRadar

Simulate Radar Detection on the Current Channel

Applies to AP.

Enumeration	nop (0), fire (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.3.6.1.1.5

8.3 settings

8.3.1 setConfiguration

8.3.1.1 setCfgFileUrl

Configuration File URL

The URL defines the location of the configuration file where it will be downloaded from or uploaded to when using `rpcCfgFile`.

For import files to the device the TFTP and HTTP/HTTPS protocols are supported. For export from the device to a server only the TFTP protocol is supported.

Allowed characters are [a-zA-Z0-9] and ._-~. Additionally the URL can contain spaces which will be encoded by the device. All other character need to be encoded by the user.

Examples:

- tftp://192.168.1.1/device.cfg
- http://192.168.1.1/device.cfg
- https://192.168.1.1/device.cfg

Note: If you use the HTTPS protocol it is highly recommended to install the TLS Client CA Certificate on the device. Otherwise the device will connect to any web server which use TLS, but the connection must be considered insecure. See `setTlsClient` for more information.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.4.1.1

8.3.1.2 setCfgFileFormat

Configuration File Format

This parameter specifies which format shall be used to export the current configuration.

- **snmp(0)** A flat list of the configuration parameters in an SNMP-based format.
- **cli(1)** A hierarchical structure of the configuration parameters in the way they are used in the CLI.

Note: Because the configuration format is automatically recognised whilst importing configurations, this parameter has no influence in this case.

Enumeration	snmp (0), cli (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.4.1.4

8.3.1.3 setCfgFileType

Configuration File Type

This parameter specifies the type of the configuration file and how to handle the confguration file by the `rpcCfgFile`.

- **standard(0)** Plain text configuration file.

- **cyber(1)** Encrypted configuration file using `setCfgFilePassword`.

Enumeration	standard (0), cyber (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.4.1.5

8.3.1.4 setCfgFilePassword

Configuration File Password

This parameter is only needed for encrypted configuration files (when `setCfgFileType` is set to **cyber(1)**).

It specifies the password used to encrypt/decrypt the configuration file during import/export.

Type	DisplayString
Range	0 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.4.1.7

8.3.2 setWireless

8.3.2.1 setWlanDeviceTable

Wireless Hardware Modules

Range	0 - 2
OID	1.3.6.1.4.1.16177.1.400.1.4.3.1

8.3.2.1.1 setWlanDeviceTableEntry

Wireless Hardware Module

OID	1.3.6.1.4.1.16177.1.400.1.4.3.1.1

8.3.2.1.1.1 setWlanDevIndex

Table Entry Index

Range	0 - 2
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.4.3.1.1.1

8.3.2.1.1.2 setWlanDevName

Name of the Wireless Device

Applies to AP and STA.

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.4.3.1.1.2

8.3.2.1.1.3 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.

Enumeration	interfaceDisabled (-1), rfOutputOff (0), rfOutputOn (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.4.3.1.1.3

8.3.2.1.1.4 setWlanDevFrequency

Wireless Frequency in MHz

Set and get the operating frequency of the device (radio).

Set frequency is supported on 802.11n products only.

Applies to AP and STA.

Range	2300 - 6300
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.4.3.1.1.6

8.3.2.1.1.5 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.

Range	0 - 100
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.4.3.1.1.8

8.3.2.2 setWlanDbgTable

Wireless Handoff Debug Parameters

Range	0 - 63
OID	1.3.6.1.4.1.16177.1.400.1.4.3.6

8.3.2.2.1 setWlanDbgTableEntry

Wireless Handoff Debug Parameters Entry

OID	1.3.6.1.4.1.16177.1.400.1.4.3.6.1

8.3.2.2.1.1 setWlanDbgIndex

Table Entry Index

Range	0 - 63
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.4.3.6.1.1

8.3.2.2.1.2 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.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.4.3.6.1.10

8.3.2.2.1.3 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.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.4.3.6.1.12

8.3.2.2.1.4 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.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.4.3.6.1.13

8.3.2.2.1.5 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.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.4.3.6.1.14

8.3.2.2.1.6 setWlanDbgIfaceName

Name of the virtual wireless interface.

Applies to AP and STA. 802.11n products only.

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.4.3.6.1.2

8.3.2.2.1.7 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.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.4.3.6.1.3

8.3.2.2.1.8 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.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.4.3.6.1.4

8.3.2.2.1.9 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.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.4.3.6.1.5

8.3.2.2.1.10 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.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.4.3.6.1.6

8.3.2.2.1.11 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.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.4.3.6.1.7

8.3.2.2.1.12 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.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.4.3.6.1.8

8.3.2.2.1.13 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.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.4.3.6.1.9

8.3.3 setFirmware

8.3.3.1 setFwFileUrl

Firmware File URL

The URL defines from which location the new firmware will be downloaded when using the `rpcFwFlash`.

Supported protocols are TFTP, HTTP/HTTPS.

Allowed characters are [a-zA-Z0-9] and ._-~. Additionally the URL can contain spaces which will be encoded by the device. All other character need to be encoded by the user itself.

Examples:

- tftp://192.168.1.1/firmware.img
- http://192.168.1.1/firmware.img
- https://192.168.1.1/firmware.img

Note: If you use the HTTPS protocol it is highly recommended to install the TLS Client CA Certificate on the device. Otherwise the device will connect to any web server which uses TLS, but the connection must be considered insecure. See `setTlsClient` for more information.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.4.5.1

8.3.3.2 setFwKeepConfig

Try to Import Configuration From the Previous Firmware Version

Enumeration	reset (0), keep (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.4.5.2

8.3.4 setCellular

8.3.4.1 setCellFwUrl

Cellular Firmware URL

This URL specifies where the upgrading process is downloading the cellular firmware.

Supported protocols are: tftp and http.

Example:

- http://192.168.1.1/firmware.rec

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.4.50.1

8.3.4.2 setCellSimSlot

Cellular SIM Slot Selection

This parameter allows to switch the SIM slot at runtime.

If the selected slot is already active or no SIM card is inserted, no action is performed.

Range	1 - 2
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.4.50.2

8.3.4.3 setCellFotaUrl

Cellular FOTA URL

By default the firmware package is downloaded from a Westermo web service. If the device operates in a private network without internet access, the URL must be set to override the public domain with a local web server.

Please contact support for instructions on how to set up the web server and obtain the required fw packages. The package name must be in the specified format, the URL points to the package folder.

Example:

- http://192.168.1.1/path/to/upgrade/packages/

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.4.50.3

8.3.5 setCertificate

8.3.5.1 setCrtFileUrl

Certificate File URL

The URL defines the location of the certificate file where it will be downloaded from or uploaded to.

Supported protocols are TFTP, HTTP/HTTPS. For export only the TFTP protocol is supported.

Examples:

- tftp://192.168.1.1/uttpd.crt
- http://192.168.1.1/uttpd.crt
- https://192.168.1.1/uttpd.crt

Note: If you use the HTTPS protocol is highly recommended to install the TLS Client CA Certificate on the device. Otherwise the device will connect to any web server which uses TLS, but the connection must be considered insecure. See `setTlsClient` for more information.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.4.6.1

8.3.5.2 setCrtFileSelector

****OBSOLETE:** Field to Select Which File Should be Imported/Exported** via `rpcCrtFile`

This parameter is obsolete and has been replaced with the Certificate Store.

See `setCrtFileId`, `setCrtFileType`, `setCrtAttributeKey`, `setCrtAttributeValue` and `setCrtFilePass`

Range	-1 - -1
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.4.6.2

8.3.5.3 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.

Enumeration	pem (0), der (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.4.6.3

8.3.5.4 setCrtFilePkcs12Passphrase

****OBSOLETE:** Set the PKCS#12 passphrase used to import with** `rpcCertFile`

This parameter is obsolete and has been replaced with `setCrtFilePassphrase`

Type	DisplayString
Range	0 - 255
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.4.6.4

8.3.5.5 setCrtFileId

Id of the Certificate

Define the id of the certificate processed by the `rpcCrtFile`.

For **export(2)** and **delete(3)** this setting need to be set so the right certificate will be processed.

For **import(1)** this setting can be set to `-1` to automatically use the next free id. Otherwise any existing certificate with this id will be overwritten.

This id is then used by the services to reference the desired certificate.

Range	-1 - 1000
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.4.6.5

8.3.5.6 setCrtFileType

File type for the certificate action

Define the type of the certificate processed by the `rpcCrtFile`.

- **1**: CRL
- **2**: Certificate
- **3**: Private Key
- **4**: Static Key
- **5**: PKCS12
- **6**: Public Key

Enumeration	crl (1), cert (2), key (3), statickey (4), pkcs12 (5), pubkey (6)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.4.6.6

8.3.5.7 setCrtAttributeKey

Key of the attribute to read/write

Select the attribute which would be read/write by the `rpcCrtAttribute`.

- **0**: Label of a certificate

Enumeration	label (0)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.4.6.7

8.3.5.8 setCrtAttributeValue

Value of the attribute

This value will be written to the attribute defined by `setCrtFileId`, `setCrtFileType` and `setCrtAttributeKey` by using the `rpcCrtAttribute`.

The value read by using the `rpcCrtAttribute` will be available here.

Type	DisplayString
Range	0 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.4.6.8

8.3.5.9 setCrtFilePassphrase

Set the passphrase used to import with `rpcCertFile`

This passphrase will be used during the import to decrypt the PKCS#12 container data or the private key.

Type	DisplayString
Range	0 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.4.6.9

8.3.6 setSystem

8.3.6.1 setSysTime

System time as epoch

Range	100000 - 2114384400
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.4.7.1

8.3.6.2 setSysSupportFile

8.3.6.2.1 setSysSfEncryptionEnabled

Disable or Enable Support File Encryption

If set to **enabled(1)**, the Technical Support File is encrypted using `setSysSfEncryptionPassword`.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.4.7.11.1

8.3.6.2.2 setSysSfEncryptionPassword

Support File Encryption Password

Used to encrypt the Technical Support File if `setSysSfEncryptionEnabled` is **enabled(1)**.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.4.7.11.2

8.3.7 setTechPreview

8.3.7.1 setTechPreviewEnabled

Disable or Enable Technical Preview

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.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.4.8.1

8.3.8 setTlsClient

8.3.8.1 setTlsCltCalds

TLS Client CA Certificate IDs

Set to -1 to use TLS client without CA certificate. Multiple CAs may be referenced by writing the ids of the CAs as space and/or comma separated list.

Examples:

- `-1`
- `12`
- `1, 3, 4`
- `1 3 4`

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.4.9.1

8.3.8.2 setTlsCltCrlExpiryExtension

CRL Validity Period Extension in Days

If set, the validity period of a CRL can be extended by the given amount of days.

- **0** no extension
- **1-1095** extension days
- **-1** extend to infinity => ignore CRL expiry

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.4.9.3

8.3.8.3 setTlsCltTlsControlParams

Bitfield to Control the TLS Client Behavior

- **0x0** all validity checks will be performed
- **0x1** ignore certificate validity time
- **0x2** ignore ca certificate
- **0x4** ignore CRLs
- **0x8** ignore missing CRLs

Range	0 - 15
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.4.9.4

8.3.8.4 setTlsCltTlsCiphers

OpenSSL Cipher String for the TLS Client

This is an OpenSSL specific configuration option for configuring the cipher.

Please read the user manual and the OpenSSL documentation for a list of available ciphers and used syntax.

Set to ‘none’ to disable restriction.

Examples:

- ECDHE-RSA-AES256-GCM-SHA384:DHE-RSA-AES256-GCM-SHA384

- DHE-RSA-AES256-GCM-SHA384
- ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-SHA384
- none

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.4.9.5

8.4 hardware

8.4.1 hwSystem

8.4.1.1 hwSysProduct

Product Type

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.5.1.1

8.4.1.2 hwSysSerial

Serial Number of the Product

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.5.1.2

8.4.1.3 hwSysRevision

ERP Revision of the Product

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.5.1.3

8.4.1.4 hwSysVersion

Version of the Product

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.5.1.4

8.4.2 hwBaseBoard

8.4.2.1 hwBbType

Product Type of the Base Board

Type	DisplayString
Range	0 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.5.10.1

8.4.2.2 hwBbSerial

Serial Number of the Base Board

Type	DisplayString
Range	0 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.5.10.2

8.4.2.3 hwBbRevision

ERP Revision of the Base Board

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.5.10.3

8.4.2.4 hwBbVersion

Version of the Base Board

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.5.10.4

8.4.2.5 hwBbPcbId

Hardware Assembly ID

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.5.10.5

8.4.2.6 hwBbAssemblyId

Hardware Assembly ID

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.5.10.6

8.4.3 hwIfaceBoard

8.4.3.1 hwIfBrdAssembled

Interface Board Present or Not

Enumeration	inexistent (0), present (1)
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.5.11.1

8.4.3.2 hwIfBrdType

Product Type of the Interface Board

Type	DisplayString
Range	0 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.5.11.2

8.4.3.3 hwIfBrdSerial

Serial Number of the Interface Board

Type	DisplayString
Range	0 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.5.11.3

8.4.3.4 hwIfBrdRevision

ERP Revision of the Interface Board

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.5.11.4

8.4.3.5 hwIfBrdVersion

Version of the Interface Board

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.5.11.5

8.4.3.6 hwIfBrdPcbId

Hardware Assembly ID

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.5.11.6

8.4.3.7 hwIfBrdAssemblyId

Hardware Assembly ID

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.5.11.7

8.4.4 hwNetwork

8.4.4.1 hwNetEthernetTable

Ethernet Network Interfaces

Range	0 - 2
OID	1.3.6.1.4.1.16177.1.400.1.5.2.1

8.4.4.1.1 hwNetEthernetTableEntry

Ethernet Network Interface

OID	1.3.6.1.4.1.16177.1.400.1.5.2.1.1

8.4.4.1.1.1 hwNetEthIndex

Table Entry Index

Range	0 - 2
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.5.2.1.1.1

8.4.4.1.1.2 hwNetEthName

Name of the Ethernet Interface

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.5.2.1.1.2

8.4.4.1.1.3 hwNetEthAssembled

Ethernet Interface Present or Not

Enumeration	inexistent (0), present (1)
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.5.2.1.1.3

8.4.4.1.1.4 hwNetEthMacAddress

Ethernet MAC Address

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.5.2.1.1.4

8.4.4.1.1.5 hwNetEthOperation

Ethernet Interface Plugged or Unplugged

Enumeration	down (0), up (1)
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.5.2.1.1.5

8.4.4.1.1.6 hwNetEthSpeed

Ethernet Speed in Mbps

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.5.2.1.1.6

8.4.4.1.1.7 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.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.5.2.1.1.7

8.4.5 hwSensor

8.4.5.1 hwSensorTable

Hardware sensor information table.

Range	0 - 15
OID	1.3.6.1.4.1.16177.1.400.1.5.21.1

8.4.5.1.1 hwSensorTableEntry

Hardware Sensor Entry

OID	1.3.6.1.4.1.16177.1.400.1.5.21.1.1

8.4.5.1.1.1 hwSensorIndex

Table Entry Index

Range	0 - 15
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.5.21.1.1.1

8.4.5.1.1.2 hwSensorName

Name of the Hardware Sensor

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.5.21.1.1.2

8.4.5.1.1.3 hwSensorUnit

Unit of the Hardware Sensor

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.5.21.1.1.3

8.4.5.1.1.4 hwSensorValue

Value of the Hardware Sensor

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.5.21.1.1.4

8.4.6 hwWireless

8.4.6.1 hwWlanDeviceTable

Hardware information of the wireless LAN Devices

Range	0 - 2
OID	1.3.6.1.4.1.16177.1.400.1.5.3.1

8.4.6.1.1 hwWlanDeviceTableEntry

Wireless LAN Devices

OID	1.3.6.1.4.1.16177.1.400.1.5.3.1.1

8.4.6.1.1.1 hwWlanDevIndex

Table Entry Index

Range	0 - 2
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.5.3.1.1.1

8.4.6.1.1.2 hwWlanDevAntennaProfileId

Antenna Profile ID

Please check the user manual for antenna details.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.5.3.1.1.10

8.4.6.1.1.3 hwWlanDevAntennaGain

Antenna Gain in dBi

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.5.3.1.1.11

8.4.6.1.1.4 hwWlanDevCableLoss

Cable Loss in dB

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.5.3.1.1.12

8.4.6.1.1.5 hwWlanDevAssembled

Wireless Device Present or Not

Enumeration	inexistent (0), present (1)
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.5.3.1.1.2

8.4.6.1.1.6 hwWlanDevType

Type of the Wireless Device

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.5.3.1.1.3

8.4.6.1.1.7 hwWlanDevSerial

Serial Number / Customer Field

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.5.3.1.1.4

8.4.6.1.1.8 hwWlanDevRevision

ERP Revision of the RF Board

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.5.3.1.1.5

8.4.6.1.1.9 hwWlanDevVersion

Version of the RF board

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.5.3.1.1.6

8.4.6.1.1.10 hwWlanDevPcbId

Hardware Assembly ID

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.5.3.1.1.7

8.4.6.1.1.11 hwWlanDevAssemblyId

Hardware Assembly ID

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.5.3.1.1.8

8.4.6.1.1.12 hwWlanDevMacAddress

MAC Address

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.5.3.1.1.9

8.4.6.2 hwWlanGlobal

8.4.6.2.1 hwWlanGlblRegulatoryRegionId

Regulatory Region ID

Applies to AP and STA.

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.5.3.2.2

8.4.7 hwCellular

8.4.7.1 hwCellAssembled

Cellular Module Assembled

Enumeration	inexistent (0), present (1)
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.5.50.1

8.4.7.2 hwCellType

Cellular Module Type

Type	DisplayString
Range	0 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.5.50.2

8.4.7.3 hwCellSerial

Cellular Module Serial Number

Type	DisplayString
Range	0 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.5.50.3

8.4.7.4 hwCellImei

Cellular Module IMEI Number

Type	DisplayString
Range	0 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.5.50.4

8.4.7.5 hwCellFwVersion

Cellular Module Firmware Version

Type	DisplayString
Range	0 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.5.50.5

8.4.7.6 hwCellModuleInitialized

Cellular Module Initialization State

Enumeration	false (0), true (1)
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.5.50.6

8.4.7.7 hwCellFwVersionPreferred

Cellular Module Firmware Version Preferred State

The parameter returns **true(1)** if the latest cellular firmware is installed. Otherwise it returns **false(0)** if a firmware upgrade is available.

Enumeration	false (0), true (1)
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.5.50.7

8.4.7.8 hwCellFwVersionPreferredVersion

Cellular Module Firmware Version Preferred Version

Type	DisplayString
Range	0 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.5.50.8

8.4.8 hwGnss

8.4.8.1 hwGnssAssembled

GNSS Module Assembled

Enumeration	inexistent (0), present (1)
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.5.51.1

8.4.8.2 hwGnssType

GNSS Module Type

Type	DisplayString
Range	0 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.5.51.2

8.4.8.3 hwGnssSerial

GNSS Module Serial Number

Type	DisplayString
Range	0 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.5.51.3

8.4.9 hwPowerSupply

8.4.9.1 hwPsAssembled

Power Supply Assembled

Enumeration	inexistent (0), present (1)
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.5.52.1

8.4.9.2 hwPsType

Power Supply Type

Type	DisplayString
Range	0 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.5.52.2

8.4.9.3 hwPsSerial

Power Supply Serial Number

Type	DisplayString
Range	0 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.5.52.3

8.4.9.4 hwPsInputRange

Power Supply Input Range

Reports the input range of the power supply:

- **POWER_INPUT_RANGE_LOW**: 12V-48V DC
- **POWER_INPUT_RANGE_WIDE**: 24V-110V DC

Type	DisplayString
Range	0 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.5.52.4

8.5 software

8.5.1 swFirmware

8.5.1.1 swFwName

Firmware Name

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.2.1

8.5.1.2 swFwVersion

Firmware Version

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.2.2

8.5.1.3 swFwRevision

Firmware Revision

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.2.3

8.5.1.4 swFwPackageName

Firmware Package Name

The name of the firmware package inside the combined firmware image. The name consists of the platform name and the firmware config.

Example:

- dt50-sw6
- dt50-ac
- dt50-lte

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.2.4

8.5.2 swBootloader

8.5.2.1 swBootName

Name of the Bootloader

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.20.1

8.5.2.2 swBootVersion

Version of the Bootloader

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.20.2

8.5.2.3 swBootBuildDate

Date when the Bootloader was Built

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.20.3

8.5.3 swSystem

8.5.3.1 swSysRebootReason

System Reboot Reason

Enumeration	coldstart (0), warmstart (1), watchdog (2), oops (3), unknown (9)
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.3.1

8.5.3.2 swSysMessageTable

System messages (e.g. Errors, Warnings)

Range	0 - 63
OID	1.3.6.1.4.1.16177.1.400.1.6.3.100

8.5.3.2.1 swSysMessageTableEntry

System message entry

OID	1.3.6.1.4.1.16177.1.400.1.6.3.100.1

8.5.3.2.1.1 swSysMsgIndex

Table Entry Index

Range	0 - 63
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.6.3.100.1.1

8.5.3.2.1.2 swSysMsgPriority

Message Priority/Level

Enumeration	emergency (0), alert (1), critical (2), error (3), warning (4), notice (5), info (6), debug (7)
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.3.100.1.2

8.5.3.2.1.3 swSysMsgCode

Message Code

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.3.100.1.3

8.5.3.2.1.4 swSysMsgText

Message

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.3.100.1.4

8.5.3.2.1.5 swSysMsgEpoch

Message timestamp as Epoch.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.3.100.1.5

8.5.3.2.1.6 swSysMsgDate

Message timestamp as date string.

Example:

- `Sun Feb 20 18:07:53 2022`

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.3.100.1.6

8.5.3.3 swSysBootStatus

The boot status indicates whether the booting sequence of the device has been completed.

Enumeration	done (0), booting (1)
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.3.2

8.5.3.4 swSysInitialized

Initial Configuration Status

This entry reflects the status of the initial configuration. Once a first configuration has been applied, its value changes to true(1). It can only be reset to false(0) by performing a factory reset.

Enumeration	false (0), true (1)
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.3.3

8.5.4 swConfiguration

8.5.4.1 swCfgChangesCount

Number of not yet Applied Device Configuration Changes

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.30.1

8.5.4.2 swCfgHash

Configuration Hash

Hash in hexadicimal of the current applied configuration which can be used to verify the configuration integrity.

During boot-up or during apply of a new configuration this element will return an empty string.

The used hash algorithm is SHA384.

Type	DisplayString
Range	0 - 96
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.30.5

8.5.5 swOperatingSystem

8.5.5.1 swOsName

Operating System Name

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.4.1

8.5.5.2 swOsPartitions

8.5.5.2.1 swOsPartTable

Hashes of all Partiton Tables

These hashes can be used to assert the integrity of all partitions.

The hashes are not generated automatically. To generate them use `rpcPartGenerateHashes`

Range	0 - 127
OID	1.3.6.1.4.1.16177.1.400.1.6.4.101.1

8.5.5.2.1.1 swOsPartTableEntry

Hashes of all Partiton Tables

These hashes can be used to assert the integrity of all partitions.

The hashes are not generated automatically. To generate them use `rpcPartGenerateHashes`

OID	1.3.6.1.4.1.16177.1.400.1.6.4.101.1.1

swOsPartIndex

Table Entry Index

Range	0 - 127
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.6.4.101.1.1.1

swOsPartDevice

The Device of the Partition

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.4.101.1.1.2

swOsPartLabel

The Label of the Partition

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.4.101.1.1.3

swOsPartHash

The MD5 Hash of the Partition

Type	DisplayString
Range	32 - 32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.4.101.1.1.4

8.5.5.3 swOsVersion

Operating System Version

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.4.2

8.5.5.4 swOsRevision

Operating System Revision

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.4.3

8.5.5.5 swOsUptime

Uptime of the Operating System

This is the time in TimeTicks (hundredth of a second) the device has been up and running since the last reboot.

Example:

456732 means up since 4567.32 seconds

Applies to AP and STA.

Type	TimeTicks
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.4.4

8.5.6 swDriver

8.5.6.1 swDrvDfsTable

DFS Driver Statistics

Range	0 - 1
OID	1.3.6.1.4.1.16177.1.400.1.6.5.1

8.5.6.1.1 swDrvDfsTableEntry

DFS Driver Statistics

OID	1.3.6.1.4.1.16177.1.400.1.6.5.1.1

8.5.6.1.1.1 swDrvDfsIndex

Table Entry Index

Range	0 - 1
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.6.5.1.1.1

8.5.6.1.1.2 swDrvDfsName

Name of The Wireless Device

Applies to AP and STA. 802.11n products only.

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.1.1.2

8.5.6.1.1.3 swDrvDfsPulsesDetected

Pulses Detected by The Wireless Device

Applies to AP. 802.11n products only.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.1.1.3

8.5.6.1.1.4 swDrvDfsPulsesProcessed

Pulses Processed by The Wireless Device

Applies to AP. 802.11n products only.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.1.1.4

8.5.6.1.1.5 swDrvDfsRadarDetected

Radar Sequences Detected by The Wireless Device

Applies to AP. 802.11n products only.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.1.1.5

8.5.6.2 swDrvCntWlanMacTable

Wireless MAC-Layer Statistics

Range	0 - 63
OID	1.3.6.1.4.1.16177.1.400.1.6.5.4

8.5.6.2.1 swDrvCntWlanMacTableEntry

Wireless MAC-Layer Statistics

OID	1.3.6.1.4.1.16177.1.400.1.6.5.4.1

8.5.6.2.1.1 swDrvCntWlanMacIndex

Table Entry Index

Range	0 - 63
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.6.5.4.1.1

8.5.6.2.1.2 swDrvCntWlanMacRxHandlersDrop

MAC Debug Entry

Applies to AP and STA. 802.11n products only.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.4.1.10

8.5.6.2.1.3 swDrvCntWlanMacRxHandlersQueued

MAC Debug Entry

Applies to AP and STA. 802.11n products only.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.4.1.11

8.5.6.2.1.4 swDrvCntWlanMacRxHandlersDropNullfunc

MAC Debug Entry

Applies to AP and STA. 802.11n products only.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.4.1.12

8.5.6.2.1.5 swDrvCntWlanMacRxHandlersDropDefrag

MAC Debug Entry

Applies to AP and STA. 802.11n products only.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.4.1.13

8.5.6.2.1.6 swDrvCntWlanMacRxHandlersDropShort

MAC Debug Entry

Applies to AP and STA. 802.11n products only.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.4.1.14

8.5.6.2.1.7 swDrvCntWlanMacTxExpandSkbHead

MAC Debug Entry

Applies to AP and STA. 802.11n products only.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.4.1.15

8.5.6.2.1.8 swDrvCntWlanMacTxExpandSkbHeadCloned

MAC Debug Entry

Applies to AP and STA. 802.11n products only.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.4.1.16

8.5.6.2.1.9 swDrvCntWlanMacRxExpandSkbHead

MAC Debug Entry

Applies to AP and STA. 802.11n products only.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.4.1.17

8.5.6.2.1.10 swDrvCntWlanMacRxExpandSkbHead2

MAC Debug Entry

Applies to AP and STA. 802.11n products only.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.4.1.18

8.5.6.2.1.11 swDrvCntWlanMacRxHandlersFragments

MAC Debug Entry

Applies to AP and STA. 802.11n products only.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.4.1.19

8.5.6.2.1.12 swDrvCntWlanMacName

Name of The Wireless Device

Applies to AP and STA. 802.11n products only.

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.4.1.2

8.5.6.2.1.13 swDrvCntWlanMacTxstatusDrop

MAC Debug Entry

Applies to AP and STA. 802.11n products only.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.4.1.20

8.5.6.2.1.14 swDrvCntWlanMacTxHandlersDrop

MAC Debug Entry

Applies to AP and STA. 802.11n products only.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.4.1.3

8.5.6.2.1.15 swDrvCntWlanMacTxHandlersQueued

MAC Debug Entry

Applies to AP and STA. 802.11n products only.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.4.1.4

8.5.6.2.1.16 swDrvCntWlanMacTxHandlersDropUnencrypted

MAC Debug Entry

Applies to AP and STA. 802.11n products only.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.4.1.5

8.5.6.2.1.17 swDrvCntWlanMacTxHandlersDropFragment

MAC Debug Entry

Applies to AP and STA. 802.11n products only.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.4.1.6

8.5.6.2.1.18 swDrvCntWlanMacTxHandlersDropWep

MAC Debug Entry

Applies to AP and STA. 802.11n products only.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.4.1.7

8.5.6.2.1.19 swDrvCntWlanMacTxHandlersDropNotAssoc

MAC Debug Entry

Applies to AP and STA. 802.11n products only.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.4.1.8

8.5.6.2.1.20 swDrvCntWlanMacTxHandlersDropUnauthPort

MAC Debug Entry

Applies to AP and STA. 802.11n products only.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.4.1.9

8.5.6.3 swDrvCntWlanWmmTable

WMM statistics

Range	0 - 4
OID	1.3.6.1.4.1.16177.1.400.1.6.5.6

8.5.6.3.1 swDrvCntWlanWmmTableEntry

WMM statistics

OID	1.3.6.1.4.1.16177.1.400.1.6.5.6.1

8.5.6.3.1.1 swDrvCntWlanWmmTableIndex

Table Entry Index

Range	0 - 4
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.6.5.6.1.1

8.5.6.3.1.2 swDrvCntWlanWmmName

Name of The Queue

Applies to AP and STA. 802.11n products only.

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.6.1.2

8.5.6.3.1.3 swDrvCntWlanWmmTx

Number of Frames Sent in his Queue

Applies to AP and STA. 802.11n products only.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.6.1.3

8.5.6.3.1.4 swDrvCntWlanWmmRx

Number of Frames Received in This Queue

Applies to AP and STA. 802.11n products only.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.6.1.4

8.5.6.3.1.5 swDrvCntWlanWmmShortRetries

Number of Retries for Frames Shorter Than RTS

Applies to AP and STA. 802.11n products only.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.6.1.5

8.5.6.3.1.6 swDrvCntWlanWmmLongRetries

Number of Retries for Frames Longer Than RTS

Applies to AP and STA. 802.11n products only.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.6.1.6

8.5.6.3.1.7 swDrvCntWlanWmmExceededRetries

Number of Failed Transmissions Due to Exceeding of The Retry Limit

Applies to AP and STA. 802.11n products only.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.6.1.7

8.5.6.4 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.

Specify `all` to get the connection status of all wlan interfaces.

Applies to AP and STA.

Type	DisplayString
Range	3 - 17
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.6.5.7

8.5.6.5 swDrvConStatTable

Connection Status Information.

Range	0 - 9
OID	1.3.6.1.4.1.16177.1.400.1.6.5.8

8.5.6.5.1 swDrvConStatTableEntry

Connection Status Information per Station.

OID	1.3.6.1.4.1.16177.1.400.1.6.5.8.1

8.5.6.5.1.1 swDrvConStatIndex

Table Entry Index

Range	0 - 9
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.6.5.8.1.1

8.5.6.5.1.2 swDrvConStatTxBrType

****OBSOLETE:** Station TX Bitrate Type**

This information is now available in `swDrvConStatTxBrExtra`.

Type	DisplayString
Range	1 - 255
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.6.5.8.1.10

8.5.6.5.1.3 swDrvConStatTxBrValue

Station TX Bitrate Value in Mbps

Rounded down to integer, for a more detailed representation check `swDrvConStatTxBrExtra`.

Applies to AP and STA.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.8.1.11

8.5.6.5.1.4 swDrvConStatTxBytes

Station TX Bytes

Applies to AP and STA. 802.11n products only.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.8.1.12

8.5.6.5.1.5 swDrvConStatTxPackets

Station TX Packets

Applies to AP and STA. 802.11n products only.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.8.1.13

8.5.6.5.1.6 swDrvConStatSigChain0

Station Signal Chain 0 in dBm

Applies to AP and STA. 802.11n products only.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.8.1.14

8.5.6.5.1.7 swDrvConStatSigChain1

Station Signal Chain 1 in dBm

Applies to AP and STA. 802.11n products only.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.8.1.15

8.5.6.5.1.8 swDrvConStatSigChain2

Station Signal Chain 2 in dBm

Applies to AP and STA. 802.11n products only.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.8.1.16

8.5.6.5.1.9 swDrvConStatSigAvgChain0

Station Signal Average Chain 0 in dBm

Applies to AP and STA. 802.11n products only.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.8.1.17

8.5.6.5.1.10 swDrvConStatSigAvgChain1

Station Signal Average Chain 1 in dBm

Applies to AP and STA. 802.11n products only.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.8.1.18

8.5.6.5.1.11 swDrvConStatSigAvgChain2

Station Signal Average Chain 2 in dBm

Applies to AP and STA. 802.11n products only.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.8.1.19

8.5.6.5.1.12 swDrvConStatWlanName

WLAN Interface Name

Applies to AP and STA.

Type	DisplayString
Range	4 - 5
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.8.1.2

8.5.6.5.1.13 swDrvConStatTxRetries

Station TX Retries

Applies to AP and STA.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.8.1.20

8.5.6.5.1.14 swDrvConStatTxFailed

Station TX Failed

Applies to AP and STA. 802.11n products only.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.8.1.21

8.5.6.5.1.15 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.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.8.1.22

8.5.6.5.1.16 swDrvConStatSigCombined

Station Signal Combined of All Active Chains in dBm

Applies to AP and STA.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.8.1.23

8.5.6.5.1.17 swDrvConStatSigAvgCombined

Station Signal Average Combined of All Active Chains in dBm

Applies to AP and STA.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.8.1.24

8.5.6.5.1.18 swDrvConStatSigChain3

Station Signal Chain 3 in dBm

Applies to AP and STA.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.8.1.25

8.5.6.5.1.19 swDrvConStatSigAvgChain3

Station Signal Average Chain 3 in dBm

Applies to AP and STA.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.8.1.26

8.5.6.5.1.20 swDrvConStatConnectedTime

Station Connected Time in Seconds

Applies to AP and STA.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.8.1.27

8.5.6.5.1.21 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.

Type	DisplayString
Range	1 - 17
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.8.1.3

8.5.6.5.1.22 swDrvConStatRxBrExtra

Station RX Bitrate Details

This privides the effective RX bitrate in Mbps, and the unterlying settings (bandwidth, guard interval, MCS, NSS), depending on selected operation mode.

Applies to AP and STA.

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.8.1.4

8.5.6.5.1.23 swDrvConStatRxBrType

****OBSOLETE:** Station RX Bitrate Type**

This information is now available in `swDrvConStatRxBrExtra`.

Type	DisplayString
Range	1 - 255
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.6.5.8.1.5

8.5.6.5.1.24 swDrvConStatRxBrValue

Station RX Bitrate Value in Mbps

Rounded down to integer, for a more detailed representation check `swDrvConStatRxBrExtra`.

Applies to AP and STA.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.8.1.6

8.5.6.5.1.25 swDrvConStatRxBytes

Station RX Bytes

Applies to AP and STA. 802.11n products only.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.8.1.7

8.5.6.5.1.26 swDrvConStatRxPackets

Station RX Packets

Applies to AP and STA. 802.11n products only.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.8.1.8

8.5.6.5.1.27 swDrvConStatTxBrExtra

Station TX Bitrate Details

This privides the effective TX bitrate in Mbps, and the unterlying settings (bandwidth, guard interval, MCS, NSS), depending on selected operation mode.

Applies to AP and STA.

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.8.1.9

8.5.6.6 swDrvCntWlanTable

Wireless Dev Counters

Range	0 - 63
OID	1.3.6.1.4.1.16177.1.400.1.6.5.9

8.5.6.6.1 swDrvCntWlanTableEntry

Wireless Dev Counters

OID	1.3.6.1.4.1.16177.1.400.1.6.5.9.1

8.5.6.6.1.1 swDrvCntWlanIndex

Table Entry Index

Range	0 - 63
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.6.5.9.1.1

8.5.6.6.1.2 swDrvCntWlanChannelActive

Time in ms Which the Device Has Been on the Current Channel

On STA when it's no associated the current channel is the lowest channel of the selected country code.

Applies to AP and STA. 802.11n products only.

Type	Counter32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.9.1.10

8.5.6.6.1.3 swDrvCntWlanChannelBusy

Time in ms Which the Medium Has Been Busy on the Current Channel

Applies to AP and STA. 802.11n products only.

Type	Counter32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.9.1.11

8.5.6.6.1.4 swDrvCntWlanChannelTransmit

Time in ms Which the Device Has Been Transmitting on the Current Channel

Applies to AP and STA. 802.11n products only.

Type	Counter32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.9.1.12

8.5.6.6.1.5 swDrvCntWlanChannelReceive

Time in ms Which the Device Has Been Receiving on the Current Channel

Applies to AP and STA. 802.11n products only.

Type	Counter32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.9.1.13

8.5.6.6.1.6 swDrvCntWlanChannelNoise

Measured Channel Noise in 1 dB Resolution

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.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.9.1.14

8.5.6.6.1.7 swDrvCntWlanName

Name of the Wireless Interface

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.9.1.2

8.5.6.6.1.8 swDrvCntWlanEapAuthStartedFT

EAP Sessions Started Through FT

Applies to AP. 802.11n products only.

Type	Counter32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.9.1.20

8.5.6.6.1.9 swDrvCntWlanEapAuthStartedFILS

EAP Sessions Started Through FILS

Applies to AP. 802.11n products only.

Type	Counter32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.9.1.21

8.5.6.6.1.10 swDrvCntWlanEapAuthStartedPKMSA

EAP Sessions Started Through PKMSA

Applies to AP. 802.11n products only.

Type	Counter32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.9.1.22

8.5.6.6.1.11 swDrvCntWlanAssocSuccess

Number of Successful Associations

802.11n products only.

Type	Counter32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.9.1.4

8.5.6.6.1.12 swDrvCntWlanBeaconMiss

Number of Beacon Misses

Applies to STA.

Type	Counter32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.9.1.40

8.5.6.6.1.13 swDrvCntWlanBeaconRx

Number of Beacons Received

Applies to STA.

Type	Counter32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.9.1.41

8.5.6.6.1.14 swDrvCntWlanApBeaconMiss

Number of AP Missed Beacons

Applies to AP.

Type	Counter32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.9.1.42

8.5.6.6.1.15 swDrvCntWlanPilotMiss

Number of Pilot Frame Misses

Applies to STA.

Type	Counter32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.9.1.43

8.5.6.6.1.16 swDrvCntWlanPilotRx

Number of Pilot Frames Received

Applies to STA.

Type	Counter32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.9.1.44

8.5.6.6.1.17 swDrvCntWlanAssocFailure

Number of Unsuccessful Associations

802.11n products only.

Type	Counter32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.9.1.5

8.5.6.6.1.18 swDrvCntWlanAssocFailureMaxSta

Number of Times the Maximum Number of Stations has been Exceeded

Applies to AP. 802.11n products only.

Type	Counter32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.9.1.6

8.5.6.6.1.19 swDrvCntWlanNumAssocSta

Number of Associated STA

Applies to AP. 802.11n products only.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.9.1.7

8.5.6.6.1.20 swDrvCntWlanEapAuthStarted

Number of EAP Authentication Sessions Started Since AP Start

Applies to AP. 802.11n products only.

Type	Counter32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.9.1.8

8.5.6.6.1.21 swDrvCntWlanEapAuthFailed

Number of Failed EAP Authentication Sessions Since AP Start

Applies to AP. 802.11n products only.

Type	Counter32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.5.9.1.9

8.5.7 swCellular

8.5.7.1 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.

Range	0 - 0
OID	1.3.6.1.4.1.16177.1.400.1.6.50.1

8.5.7.1.1 swCellTableEntry

Cellular Status Information Entry

Applies to cellular products only.

OID	1.3.6.1.4.1.16177.1.400.1.6.50.1.1

8.5.7.1.1.1 swCellIndex

Table Entry Index

Range	0 - 0
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.6.50.1.1.1

8.5.7.1.1.2 swCellConnectionStatus

Connection Status

- **unknown(0)**: Unknown state.
- **disabled(1)**: Interface is disabled.
- **switchingSim(2)**: Switching the SIM slot.
- **deregistering(3)**: Deregistering from the cellular network.
- **resetting(4)**: Resetting the cellular modem.
- **registering(5)**: Registering to the cellular network.
- **connecting(6)**: Connecting to the network.
- **connected(7)**: Connected to the network.
- **disconnected(8)**: Disconnected from the network.
- **unlockingSim(9)**: Unlocking the SIM card.
- **reloading(10)**: Reloading the configuration.
- **invalid(11)**: Invalid state.

Applies to cellular products only.

Enumeration	unknown (0), disabled (1), switchingSim (2), deregistering (3), resetting (4), registering (5), connecting (6), connected (7), disconnected (8), unlockingSim (9), reloading (10), invalid (11)
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.1.1.10

8.5.7.1.1.3 swCellConnectionMessage

Connection Status Message

Applies to cellular products only.

Type	DisplayString
Range	0 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.1.1.11

8.5.7.1.1.4 swCellSignalType

Signal Type

The detailed signal type of the connection.

unknown(0): Unknown signal type **wcdma(3)**: UMTS (3G) **lte(4)**: LTE (4G) **nsa5g(5)**: 5G-NSA (5G Not Standalone) **sa5g(6)**: 5G-SA (5G Standalone)

Applies to cellular products only.

Enumeration	unknown (0), wcdma (3), lte (4), nsa5g (5), sa5g (6)
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.1.1.12

8.5.7.1.1.5 swCellSignalRssi

Received Signal Strength Indication (RSSI)

Deprecated. Use `swCellLteRssi`. This parameter is only active when `swCellSignalType` is of type **lte(4)**.

The unit of the RSSI is in dB.

Applies to cellular products only.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.1.1.13

8.5.7.1.1.6 swCellSignalRsrq

Reference Signal Received Quality (RSRQ)

Deprecated. Use `swCellLteRsrq`. This parameter is only active when `swCellSignalType` is of type **lte(4)**.

The unit of the RSRQ is in dB.

Applies to cellular products only.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.1.1.14

8.5.7.1.1.7 swCellSignalRsrp

Reference Signal Received Power (RSRP)

Deprecated. Use `swCellLteRsrp`. This parameter is only active when `swCellSignalType` is of type **lte(4)**.

The unit of the RSRP is in dBm.

Applies to cellular products only.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.1.1.15

8.5.7.1.1.8 swCellSignalSinr

Signal to Interference Plus Noise Ratio (SINR)

Deprecated. Use `swCellLteSinr`. This parameter is only active when `swCellSignalType` is of type **lte(4)**.

The SINR in dB is calculated from the value *v* of this entry using the following equation:

SINR = (v * 2) - 20

Examples:

- **0**: SINR = -20dB
- **10**: SINR = 0dB
- **25**: SINR = 30dB

Applies to cellular products only.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.1.1.16

8.5.7.1.1.9 swCellSignalRscp

Received Signal Code Power (RSCP)

Deprecated. Use `swCellWcdmaRscp`. This parameter is only active when `swCellSignalType` is of type **wcdma(0)**.

The unit of the RSCP is in dBm.

Applies to cellular products only.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.1.1.17

8.5.7.1.1.10 swCellSignalEcio

Carrier to Noise Ratio (EC/IO)

Deprecated. Use `swCellWcdmaEcio`. This parameter is only active when `swCellSignalType` is of type **wcdma(0)**.

The unit of the ECIO is in dBm.

Applies to cellular products only.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.1.1.18

8.5.7.1.1.11 swCellLteMode

LTE Duplex Mode

Deprecated. Use `swCellLteDuplexMode`.

LTE modulation mode FDD or TDD.

Applies to cellular products only.

Enumeration	unknown (0), tdd (1), fdd (2)
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.1.1.19

8.5.7.1.1.12 swCellWwanName

Name of the Cellular Radio

Applies to cellular products only.

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.1.1.2

8.5.7.1.1.13 swCellEarfcn

E-UTRA-ARFCN

Deprecated. Use `swCellLteEarfcn`.

The parameter determines the E-UTRA-ARFCN of the cell.

Applies to cellular products only.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.1.1.20

8.5.7.1.1.14 swCellCellId

Cell ID

Deprecated. Use `swCellLteCellId` or `swCell5gsaCellId`.

The parameter determines the 28-bit (WCDMA, LTE) or 36-bit (5G-NR) cell ID as a hexadecimal value.

Applies to cellular products only.

Type	DisplayString
Range	0 - 9
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.1.1.21

8.5.7.1.1.15 swCellBandwidthUl

Upload Bandwidth

Deprecated. Use `swCellLteBandwidthUl`.

- **bw1dot4(0)**: 1.4 MHz
- **bw3(1)**: 3 MHz
- **bw5(2)**: 5 MHz
- **bw10(3)**: 10 MHz
- **bw15(4)**: 15 MHz
- **bw20(5)**: 20 MHz

Applies to cellular products only.

Enumeration	bw1dot4 (0), bw3 (1), bw5 (2), bw10 (3), bw15 (4), bw20 (5)
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.1.1.22

8.5.7.1.1.16 swCellBandwidthDl

Download Bandwidth

Deprecated. Use `swCellLteBandwidthDl`.

- **bw1dot4(0)**: 1.4 MHz
- **bw3(1)**: 3 MHz
- **bw5(2)**: 5 MHz
- **bw10(3)**: 10 MHz
- **bw15(4)**: 15 MHz
- **bw20(5)**: 20 MHz

Applies to cellular products only.

Enumeration	bw1dot4 (0), bw3 (1), bw5 (2), bw10 (3), bw15 (4), bw20 (5)
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.1.1.23

8.5.7.1.1.17 swCellSimIccid

ICCID

ICCID of the active SIM card.

Applies to cellular products only.

Type	DisplayString
Range	0 - 22
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.1.1.24

8.5.7.1.1.18 swCellServingCellState

Service Cell State

The cellular device can be in the following states:

- **searching** The cellular device is searching but could not (yet) find a suitable 3G/4G/5G cell.
- **notRegistered** The cellular device is camping on a cell but has not registered on the network.

- **registered** The cellular device is camping on a cell and has registered on the network, and it is in idle mode.
- **registeredCall** The cellular device is camping on a cell and has registered on the network, and a call is in progress.

Applies to cellular products only.

Enumeration	unknown (0), searching (1), notRegistered (2), registered (3), registeredCall (4)
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.1.1.25

8.5.7.1.1.19 swCellFullNetworkName

Full Network Name (FNN)

The full name of the network.

Applies to cellular products only.

Type	DisplayString
Range	0 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.1.1.26

8.5.7.1.1.20 swCellConnectedTime

Connected Time

Time in seconds since the connection was established.

Applies to cellular products only.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.1.1.27

8.5.7.1.1.21 swCellSupports5g

Connected cellular network supports 5G

Indication if the currently connected cellular network supports 5G technology.

- **no(0)**
- **yes(1)**

Check `swCellSignalType` to get the exact signal type of the current connection. It can also be set to LTE if only the LTE cell is visible but the cellular network basically supports 5G. Some internet service provider enable carrier aggregation with a 5G-NSA cell only when the network load is heavy and fall back to LTE on idle.

Applies to 5G cellular products only.

Enumeration	no (0), yes (1)
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.1.1.28

8.5.7.1.1.22 swCellSimSlot

Active SIM Slot

- **none(0)**: No SIM slot is active
- **slot1(1)**: SIM slot 1 is active
- **slot2(2)**: SIM slot 2 is active

Applies to cellular products only.

Enumeration	none (0), slot1 (1), slot2 (2)
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.1.1.3

8.5.7.1.1.23 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.

Enumeration	noSim (0), ready (1), pinReq (2), pukReq (3)
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.1.1.4

8.5.7.1.1.24 swCellSimPrimary

Is SIM Primary

Is the SIM card of the primary slot used?

- **no(0)**
- **yes(1)**

Applies to cellular products only.

Enumeration	no (0), yes (1)
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.1.1.5

8.5.7.1.1.25 swCellSimRoaming

Is SIM Roaming

Is the active SIM card currently roaming?

- **no(0)**
- **yes(1)**

Applies to cellular products only.

Enumeration	no (0), yes (1)
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.1.1.6

8.5.7.1.1.26 swCellServiceName

Service Provider Name (SPN)

Applies to cellular products only.

Type	DisplayString
Range	0 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.1.1.7

8.5.7.1.1.27 swCellServiceMcc

Mobile Country Code (MCC)

The Mobile Country Code is defined by the ITU-T Recommendation E.212.

Applies to cellular products only.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.1.1.8

8.5.7.1.1.28 swCellServiceMnc

Mobile Network Code (MNC)

The Mobile Network Code is defined by the ITU-T Recommendation E.212.

Applies to cellular products only.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.1.1.9

8.5.7.2 swCellWcdmaTable

Cellular WCDMA Signal Status Information Table

These values are only valid when `swCellSignalType` is of type **wcdma(0)**.

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.

Range	0 - 0
OID	1.3.6.1.4.1.16177.1.400.1.6.50.10

8.5.7.2.1 swCellWcdmaTableEntry

Cellular WCDMA Signal Status Information Entry

These values are only valid when `swCellSignalType` is of type **wcdma(0)**.

Applies to cellular products only.

OID	1.3.6.1.4.1.16177.1.400.1.6.50.10.1

8.5.7.2.1.1 swCellWcdmaIndex

Table Entry Index

Range	0 - 0
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.6.50.10.1.1

8.5.7.2.1.2 swCellWcdmaRscp

Received Signal Code Power (RSCP)

This parameter is only valid when `swCellSignalType` is of type **wcdma(0)**.

The unit of the RSCP is in dBm.

Applies to cellular products only.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.10.1.17

8.5.7.2.1.3 swCellWcdmaEcio

Carrier to Noise Ratio (EC/IO)

This parameter is only valid when `swCellSignalType` is of type **wcdma(0)**.

The unit of the ECIO is in dBm.

Applies to cellular products only.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.10.1.18

8.5.7.2.1.4 swCellWcdmaWwanName

Name of the Cellular Radio

Applies to cellular products only.

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.10.1.2

8.5.7.2.1.5 swCellWcdmaUarfcn

UTRA Absolute Radio Frequency Channel Number (UARFCN)

This parameter is only valid when `swCellSignalType` is of type **wcdma(0)**.

The parameter reports the UARFCN of the cell.

Applies to cellular products only.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.10.1.20

8.5.7.2.1.6 swCellWcdmaCellId

Cell ID

This parameter is only valid when `swCellSignalType` is of type **wcdma(0)**.

The parameter reports the WCDMA 28-bit cell ID in hexadecimal form.

Applies to cellular products only.

Type	DisplayString
Range	0 - 7
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.10.1.21

8.5.7.2.1.7 swCellWcdmaRac

Routing Area Code (RAC)

This parameter is only valid when `swCellSignalType` is of type **wcdma(0)**.

The parameter reports the RAC as an integer between 0 and 255.

Applies to cellular products only.

Range	0 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.10.1.35

8.5.7.2.1.8 swCellWcdmaLac

Location Area Code (LAC)

This parameter is only valid when `swCellSignalType` is of type **wcdma(0)**.

The parameter reports the 2-byte LAC in hexadecimal form.

Applies to cellular products only.

Type	DisplayString
Range	0 - 4
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.10.1.36

8.5.7.2.1.9 swCellWcdmaPhyCh

Physical Channel (PhyCH)

This parameter is only valid when `swCellSignalType` is of type **wcdma(0)**.

The parameter reports the type of the dedicated physical channel:

- **dpch(0): Downlink Dedicated Physical Channel
- **fdpch(1): Fractional DPCH is a special type of DPCH channel which can transmit only power control signal

Applies to cellular products only.

Enumeration	dpch (0), fdpch (1)
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.10.1.37

8.5.7.2.1.10 swCellWcdmaCpm

Compress Mode (CPM)

This parameter is only valid when `swCellSignalType` is of type **wcdma(0)**.

The parameter reports support for the Compress Mode:

- **notsupport(0)**: Not support Compress Mode
- **support(1)**: Support Compress Mode

Applies to cellular products only.

Enumeration	notsupport (0), support (1)
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.10.1.38

8.5.7.2.1.11 swCellWcdmaSpeechCode

Speech Code

This parameter is only valid when `swCellSignalType` is of type **wcdma(0)**.

Destination number on which call is to be deflected

Applies to cellular products only.

Type	DisplayString
Range	0 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.10.1.39

8.5.7.2.1.12 swCellWcdmaSlot

Slot

This parameter is only valid when `swCellSignalType` is of type **wcdma(0)**.

Depending on the Physical Channel reported in `swCellWcdmaPhych` this parameter reports the slot format:

- **dpch(0): Slot format range 0-16
- **fdpch(1): Slot format range 0-9

Applies to cellular products only.

Range	0 - 16
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.10.1.40

8.5.7.2.1.13 swCellWcdmaSf

Spreading Factor (SF)

This parameter is only valid when `swCellSignalType` is of type **wcdma(0)**.

The parameter reports the Spreading Factor.

Applies to cellular products only.

Enumeration	sf4 (0), sf8 (1), sf16 (2), sf32 (3), sf64 (4), sf128 (5), sf256 (6), sf512 (7), unknown (8)
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.10.1.41

8.5.7.2.1.14 swCellWcdmaPsc

Primary Scrambling Code (PSC)

This parameter is only valid when `swCellSignalType` is of type **wcdma(0)**.

The parameter reports the primary scrambling code of the cell that was scanned.

Applies to cellular products only.

Type	DisplayString
Range	0 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.10.1.42

8.5.7.2.1.15 swCellWcdmaActiveBands

Active Band

Currently active band as colon separated values.

Applies to cellular products only.

Type	DisplayString
Range	0 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.10.1.43

8.5.7.3 swCellLteTable

Cellular LTE Signal Status Information Table

These values are only valid when `swCellSignalType` is of type **lte(4)**.

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.

Range	0 - 0
OID	1.3.6.1.4.1.16177.1.400.1.6.50.11

8.5.7.3.1 swCellLteTableEntry

Cellular LTE Signal Status Information Entry

These values are only valid when `swCellSignalType` is of type **lte(4)**.

Applies to cellular products only.

OID	1.3.6.1.4.1.16177.1.400.1.6.50.11.1

8.5.7.3.1.1 swCellLteIndex

Table Entry Index

Range	0 - 0
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.6.50.11.1.1

8.5.7.3.1.2 swCellLteRssi

Received Signal Strength Indication (RSSI)

This parameter is only valid when `swCellSignalType` is of type **lte(4)** or **nsa5g(5)**.

The unit of the RSSI is in dB.

Applies to cellular products only.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.11.1.13

8.5.7.3.1.3 swCellLteRsrq

Reference Signal Received Quality (RSRQ)

This parameter is only valid when `swCellSignalType` is of type **lte(4)** or **nsa5g(5)**.

The unit of the RSRQ is in dB.

Applies to cellular products only.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.11.1.14

8.5.7.3.1.4 swCellLteRsrp

Reference Signal Received Power (RSRP)

This parameter is only valid when `swCellSignalType` is of type **lte(4)** or **nsa5g(5)**.

The unit of the RSRP is in dBm.

Applies to cellular products only.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.11.1.15

8.5.7.3.1.5 swCellLteSinr

Signal to Interference Plus Noise Ratio (SINR)

This parameter is only valid when `swCellSignalType` is of type **lte(4)** or **nsa5g(5)**.

The SINR in dB is calculated from the value *v* of this entry using the following equation:

SINR = (v * 2) - 20

Examples:

- **0**: SINR = -20dB
- **10**: SINR = 0dB
- **25**: SINR = 30dB

Applies to cellular products only.

Range	0 - 25
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.11.1.16

8.5.7.3.1.6 swCellLteDuplexMode

LTE Modulation Mode

This parameter is only valid when `swCellSignalType` is of type **lte(4)** or **nsa5g(5)**.

This parameter reports if the LTE modulation mode is FDD or TDD.

Applies to cellular products only.

Enumeration	unknown (0), tdd (1), fdd (2)
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.11.1.19

8.5.7.3.1.7 swCellLteWwanName

Name of the Cellular Radio

Applies to cellular products only.

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.11.1.2

8.5.7.3.1.8 swCellLteEarfcn

E-UTRA Absolute Radio Frequency Channel Number (EARFCN)

This parameter is only valid when `swCellSignalType` is of type **lte(4)** or **nsa5g(5)**.

The parameter reports the EARFCN of the cell.

Applies to cellular products only.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.11.1.20

8.5.7.3.1.9 swCellLteCellId

Cell ID

This parameter is only valid when `swCellSignalType` is of type **lte(4)** or **nsa5g(5)**.

The parameter reports the LTE 28-bit cell ID in hexadecimal form.

Applies to cellular products only.

Type	DisplayString
Range	0 - 7
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.11.1.21

8.5.7.3.1.10 swCellLteBandwidthUl

Upload Bandwidth

This parameter is only valid when `swCellSignalType` is of type **lte(4)** or **nsa5g(5)**.

- **bw1dot4(0)**: 1.4 MHz
- **bw3(1)**: 3 MHz
- **bw5(2)**: 5 MHz
- **bw10(3)**: 10 MHz
- **bw15(4)**: 15 MHz
- **bw20(5)**: 20 MHz

Applies to cellular products only.

Enumeration	bw1dot4 (0), bw3 (1), bw5 (2), bw10 (3), bw15 (4), bw20 (5)
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.11.1.22

8.5.7.3.1.11 swCellLteBandwidthDl

Download Bandwidth

This parameter is only valid when `swCellSignalType` is of type **lte(4)** or **nsa5g(5)**.

- **bw1dot4(0)**: 1.4 MHz
- **bw3(1)**: 3 MHz
- **bw5(2)**: 5 MHz
- **bw10(3)**: 10 MHz
- **bw15(4)**: 15 MHz
- **bw20(5)**: 20 MHz

Applies to cellular products only.

Enumeration	bw1dot4 (0), bw3 (1), bw5 (2), bw10 (3), bw15 (4), bw20 (5)
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.11.1.23

8.5.7.3.1.12 swCellLteTac

Tracking Area Code (TAC)

This parameter is only valid when `swCellSignalType` is of type **lte(4)** or **nsa5g(5)**.

The parameter reports the 2-byte TAC in hexadecimal form.

Applies to cellular products only.

Type	DisplayString
Range	0 - 4
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.11.1.27

8.5.7.3.1.13 swCellLtePcid

Physical Cell ID (PCID)

This parameter is only valid when `swCellSignalType` is of type **lte(4)** or **nsa5g(5)**.

Applies to cellular products only.

Range	0 - 503
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.11.1.29

8.5.7.3.1.14 swCellLteSrxlev

Cell Selection RX Level Value (SRXLEV)

This parameter is only valid when `swCellSignalType` is of type **lte(4)** or **nsa5g(5)**.

The unit of the srxlev is in dB (see 3GPP 25.304).

Applies to cellular products only.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.11.1.30

8.5.7.3.1.15 swCellLteTxPower

Tx Power

This parameter is only valid when `swCellSignalType` is of type **lte(4)** or **nsa5g(5)**.

The TX power value in 1/10 dBm. It is the maximum of all upload channel TX power. This value is only meaningful while the device is transmitting, i.e data transfer is active.

Applies to cellular products only.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.11.1.31

8.5.7.3.1.16 swCellLteCqi

Channel Quality Indicator (CQI)

This parameter is only valid when `swCellSignalType` is of type **lte(4)** or **nsa5g(5)**.

Applies to cellular products only.

Range	1 - 30
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.11.1.32

8.5.7.3.1.17 swCellLteBand

E-UTRA Frequency Band Indicator

This parameter is only valid when `swCellSignalType` is of type **lte(4)** or **nsa5g(5)**.

See 3GPP 36.101 or https://en.wikipedia.org/wiki/LTE_frequency_bands

Applies to cellular products only.

Range	0 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.11.1.33

8.5.7.3.1.18 swCellLteActiveBands

Active Band

Currently active band as colon separated values.

Applies to cellular products only.

Type	DisplayString
Range	0 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.11.1.38

8.5.7.4 swCell5gnsaTable

Cellular 5G-NSA Signal Status Information Table

These values are only valid when `swCellSignalType` is of type **nsa5g(5)**.

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.

Range	0 - 0
OID	1.3.6.1.4.1.16177.1.400.1.6.50.12

8.5.7.4.1 swCell5gnsaTableEntry

Cellular 5G-NSA Signal Status Information Entry

These values are only valid when `swCellSignalType` is of type **nsa5g(5)**.

Applies to cellular products only.

OID	1.3.6.1.4.1.16177.1.400.1.6.50.12.1

8.5.7.4.1.1 swCell5gnsaIndex

Table Entry Index

Range	0 - 0
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.6.50.12.1.1

8.5.7.4.1.2 swCell5gnsaRsrq

Reference Signal Received Quality (RSRQ)

This parameter is only valid when `swCellSignalType` is of type **nsa5g(5)**.

The unit of the RSRQ is in dB.

Applies to cellular products only.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.12.1.14

8.5.7.4.1.3 swCell5gnsaRsrp

Reference Signal Received Power (RSRP)

This parameter is only valid when `swCellSignalType` is of type **nsa5g(5)**.

The unit of the RSRP is in dBm.

Applies to cellular products only.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.12.1.15

8.5.7.4.1.4 swCell5gnsaSinr

Signal to Interference Plus Noise Ratio (SINR)

This parameter is only valid when `swCellSignalType` is of type **nsa5g(5)**.

This parameter reports the signal of 5G NR Signal-to-Interface plus Noise Ratio in dB.

Applies to cellular products only.

Range	-20 - 30
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.12.1.16

8.5.7.4.1.5 swCell5gnsaWwanName

Name of the Cellular Radio

Applies to cellular products only.

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.12.1.2

8.5.7.4.1.6 swCell5gnsaArfcn

Absolute Radio Frequency Channel Number (ARFCN)

This parameter is only valid when `swCellSignalType` is of type **nsa5g(5)**.

The parameter reports the ARFCN of the cell.

Applies to cellular products only.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.12.1.20

8.5.7.4.1.7 swCell5gnsaBandwidthDl

Download Bandwidth

This parameter is only valid when `swCellSignalType` is of type **nsa5g(5)**.

This parameter indicates the download bandwith in MHz.

- **bw5(0)**: 5 MHz
- **bw10(1)**: 10 MHz
- **bw15(2)**: 15 MHz
- **bw20(3)**: 20 MHz
- **bw25(4)**: 25 MHz
- **bw30(5)**: 30 MHz
- **bw40(6)**: 40 MHz
- **bw50(7)**: 50 MHz
- **bw60(8)**: 60 MHz
- **bw70(9)**: 70 MHz
- **bw80(10)**: 80 MHz
- **bw90(11)**: 90 MHz
- **bw100(12)**: 100 MHz
- **bw200(13)**: 200 MHz
- **bw400(14)**: 400 MHz

Applies to cellular products only.

Enumeration	bw5 (0), br10 (1), bw15 (2), bw20 (3), bw25 (4), bw30 (5), bw40 (6), bw50 (7), bw60 (8), bw70 (9), bw80 (10), bw90 (11), bw100 (12), bw200 (13), bw400 (14)
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.12.1.23

8.5.7.4.1.8 swCell5gnsaScs

NR Sub-Carrier Space

This parameter is only valid when `swCellSignalType` is of type **nsa5g(5)**.

This parameter indicates the subcarrier spacing in kHz.

- **scs15(0)**: 15 kHz
- **scs30(1)**: 10 kHz
- **scs60(2)**: 15 kHz
- **scs120(3)**: 120 kHz
- **scs240(4)**: 240 kHz

Applies to cellular products only.

Enumeration	scs15 (0), scs30 (1), scs60 (2), scs120 (3), scs240 (4)
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.12.1.28

8.5.7.4.1.9 swCell5gnsaPcid

Physical Cell ID (PCID)

This parameter is only valid when `swCellSignalType` is of type **nsa5g(5)**.

Applies to cellular products only.

Range	0 - 1007
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.12.1.29

8.5.7.4.1.10 swCell5gnsaBand

5G NR Frequency Band

This parameter is only valid when `swCellSignalType` is of type **nsa5g(5)**.

The 5G NR frequency band are defined in 3GPP 38.101.

Also see https://en.wikipedia.org/wiki/5G_NR_frequency_bands

Applies to cellular products only.

Range	0 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.12.1.33

8.5.7.4.1.11 swCell5gnsaActiveBands

Active Band

Currently active band as colon separated values.

Applies to cellular products only.

Type	DisplayString
Range	0 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.12.1.38

8.5.7.5 swCell5gsaTable

Cellular 5G-SA Signal Status Information Table

These values are only valid when `swCellSignalType` is of type **sa5g(6)**.

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.

Range	0 - 0
OID	1.3.6.1.4.1.16177.1.400.1.6.50.13

8.5.7.5.1 swCell5gsaTableEntry

Cellular 5G-SA Signal Status Information Entry

These values are only valid when `swCellSignalType` is of type **sa5g(6)**.

Applies to cellular products only.

OID	1.3.6.1.4.1.16177.1.400.1.6.50.13.1

8.5.7.5.1.1 swCell5gsaIndex

Table Entry Index

Range	0 - 0
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.6.50.13.1.1

8.5.7.5.1.2 swCell5gsaRsrq

Reference Signal Received Quality (RSRQ)

This parameter is only valid when `swCellSignalType` is of type **sa5g(6)**.

The unit of the RSRQ is in dB.

Applies to cellular products only.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.13.1.14

8.5.7.5.1.3 swCell5gsaRsrp

Reference Signal Received Power (RSRP)

This parameter is only valid when `swCellSignalType` is of type **sa5g(6)**.

The unit of the RSRP is in dBm.

Applies to cellular products only.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.13.1.15

8.5.7.5.1.4 swCell5gsaSinr

Signal to Interference Plus Noise Ratio (SINR)

This parameter is only valid when `swCellSignalType` is of type **sa5g(6)**.

This parameter reports the signal of 5G NR Signal-to-Interface plus Noise Ratio in dB.

Applies to cellular products only.

Range	-20 - 30
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.13.1.16

8.5.7.5.1.5 swCell5gsaDuplexMode

5G NR SA Duplex Mode

This parameter is only valid when `swCellSignalType` is of type **sa5g(6)**.

This parameter reports if the 5G NR SA Duplex Mode is `FDD` or `TDD`.

Applies to cellular products only.

Enumeration	unknown (0), tdd (1), fdd (2)
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.13.1.19

8.5.7.5.1.6 swCell5gsaWwanName

Name of the Cellular Radio

Applies to cellular products only.

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.13.1.2

8.5.7.5.1.7 swCell5gsaArfcn

Absolute Radio Frequency Channel Number (ARFCN)

This parameter is only valid when `swCellSignalType` is of type **sa5g(6)**.

The parameter reports the ARFCN of the cell.

Applies to cellular products only.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.13.1.20

8.5.7.5.1.8 swCell5gsaCellId

Cell ID

This parameter is only valid when `swCellSignalType` is of type **sa5g(6)**.

The parameter reports the 5G NR SA 36-bit cell ID in hexadecimal form.

Applies to cellular products only.

Type	DisplayString
Range	0 - 9
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.13.1.21

8.5.7.5.1.9 swCell5gsaBandwidthDl

Download Bandwidth

This parameter is only valid when `swCellSignalType` is of type **sa5g(6)**.

This parameter indicates the download bandwith in MHz.

- **bw5(0)**: 5 MHz
- **bw10(1)**: 10 MHz
- **bw15(2)**: 15 MHz
- **bw20(3)**: 20 MHz
- **bw25(4)**: 25 MHz
- **bw30(5)**: 30 MHz
- **bw40(6)**: 40 MHz
- **bw50(7)**: 50 MHz
- **bw60(8)**: 60 MHz
- **bw70(9)**: 70 MHz
- **bw80(10)**: 80 MHz
- **bw90(11)**: 90 MHz
- **bw100(12)**: 100 MHz
- **bw200(13)**: 200 MHz
- **bw400(14)**: 400 MHz

Applies to cellular products only.

Enumeration	bw5 (0), br10 (1), bw15 (2), bw20 (3), bw25 (4), bw30 (5), bw40 (6), bw50 (7), bw60 (8), bw70 (9), bw80 (10), bw90 (11), bw100 (12), bw200 (13), bw400 (14)
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.13.1.23

8.5.7.5.1.10 swCell5gsaTac

Tracking Area Code (TAC)

This parameter is only valid when `swCellSignalType` is of type **sa5g(6)**.

The parameter reports the 2-byte TAC in hexadecimal form.

Applies to cellular products only.

Type	DisplayString
Range	0 - 4
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.13.1.27

8.5.7.5.1.11 swCell5gsaScs

NR Sub-Carrier Space (SCS)

This parameter is only valid when `swCellSignalType` is of type **sa5g(6)**.

This parameter indicates the subcarrier spacing in kHz.

- **scs15(0)**: 15 kHz
- **scs30(1)**: 10 kHz
- **scs60(2)**: 15 kHz
- **scs120(3)**: 120 kHz
- **scs240(4)**: 240 kHz

Applies to cellular products only.

Enumeration	scs15 (0), scs30 (1), scs60 (2), scs120 (3), scs240 (4)
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.13.1.28

8.5.7.5.1.12 swCell5gsaPcid

Physical Cell ID (PCID)

This parameter is only valid when `swCellSignalType` is of type **sa5g(6)**.

Applies to cellular products only.

Range	0 - 1007
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.13.1.29

8.5.7.5.1.13 swCell5gsaSrxlev

Select RX Level Value (SRXLEV)

This parameter is only valid when `swCellSignalType` is of type **sa5g(6)**.

The unit of the srxlev is in dB (see 3GPP 25.304).

Applies to cellular products only.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.13.1.30

8.5.7.5.1.14 swCell5gsaBand

5G NR Frequency Band

This parameter is only valid when `swCellSignalType` is of type **sa5g(6)**.

The 5G NR frequency band are defined in 3GPP 38.101.

Also see https://en.wikipedia.org/wiki/5G_NR_frequency_bands

Applies to cellular products only.

Range	0 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.13.1.33

8.5.7.5.1.15 swCell5gsaActiveBands

Active Band

Currently active band as colon separated values.

Applies to cellular products only.

Type	DisplayString
Range	0 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.50.13.1.38

8.5.8 swNlm

8.5.8.1 swNlmMonitorTable

NLM Monitor Status Table

Range	0 - 255
OID	1.3.6.1.4.1.16177.1.400.1.6.52.1

8.5.8.1.1 swNlmMonitorTableEntry

NLM Monitor Status Table Entry

OID	1.3.6.1.4.1.16177.1.400.1.6.52.1.1

8.5.8.1.1.1 swNlmMonIndex

Table Entry Index

Range	0 - 255
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.6.52.1.1.1

8.5.8.1.1.2 swNlmMonState

Status Of Monitor

This value is cached and updates at most once a second.

Enumeration	down (0), up (1), disabled (2)
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.52.1.1.2

8.5.9 swRdm

8.5.9.1 swRdmMaxEirp

Maximal equivalent isotropically radiated power (EIRP) in dBm.

This value shows the maximal aggregated transmit power over all configured chains.

Applies to AP.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.6.1

8.5.9.2 swRdmMaxApp

Maximal antenna port power in dBm.

This value shows the maximal transmit power of a single chain.

Applies to AP.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.6.2

8.5.10 swCertificate

8.5.10.1 swCrtExpirationTime

Certificate expiration date/time (UTC).

Get the certificate expiration date/time for certificate type `setCrtFileType` at ID `setCrtFileId`

Example:

Oct 22 09:42:27 2018 GMT

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.9.1

8.5.10.2 swCrtFingerprint

SHA384 fingerprint of certificate

Get the certificate fingerprint for certificate type `setCrtFileType` at ID `setCrtFileId`

Example:

9C:A6:6D:7C:AD:93:A2:29:68:82:7F:50:AA:B0:5F:40:BB:82:D3:97:D5:97:28:A1:20:AE:A7:83:0C:7B:1A:CB:18:3A

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.9.2

8.5.10.3 swCrtCertTable

Certificate Table for Type CERT

Range	0 - 63
OID	1.3.6.1.4.1.16177.1.400.1.6.9.3

8.5.10.3.1 swCrtCertTableEntry

Certificate Table for Type CERT

OID	1.3.6.1.4.1.16177.1.400.1.6.9.3.1

8.5.10.3.1.1 swCrtCertTableIndex

Table Entry Index

Range	0 - 63
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.6.9.3.1.1

8.5.10.3.1.2 swCrtCertId

Certificate Id in the Certificate Store

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.9.3.1.2

8.5.10.3.1.3 swCrtCertLabel

Certificate Id Label in the Certificate Store

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.9.3.1.3

8.5.10.3.1.4 swCrtCertExpirationTime

Certificate Expiration Date/Time (UTC)

Example:

Oct 22 09:42:27 2018 GMT

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.9.3.1.4

8.5.10.3.1.5 swCrtCertFingerprint

SHA384 Fingerprint of Certificate

Example:

9C:A6:6D:7C:AD:93:A2:29:68:82:7F:50:AA:B0:5F:40:BB:82:D3:97:D5:97:28:A1:20:AE:A7:83:0C:7B:1A:CB:18:3A

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.9.3.1.5

8.5.10.4 swCrtCrlTable

Certificate Table for Type CERT

Range	0 - 63
OID	1.3.6.1.4.1.16177.1.400.1.6.9.4

8.5.10.4.1 swCrtCrlTableEntry

Certificate Table for Type CERT

OID	1.3.6.1.4.1.16177.1.400.1.6.9.4.1

8.5.10.4.1.1 swCrtCrlTableIndex

Table Entry Index

Range	0 - 63
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.1.6.9.4.1.1

8.5.10.4.1.2 swCrtCrlId

CRL Id in the Certificate Store

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.9.4.1.2

8.5.10.4.1.3 swCrtCrlLabel

CRL Id Label in the Certificate Store

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.9.4.1.3

8.5.10.4.1.4 swCrtCrlExpirationTime

CRL Expiration Date/Time (UTC)

Example:

Oct 22 09:42:27 2018 GMT

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.9.4.1.4

8.5.10.4.1.5 swCrtCrlFingerprint

SHA384 Fingerprint of CRL

Example:

9C:A6:6D:7C:AD:93:A2:29:68:82:7F:50:AA:B0:5F:40:BB:82:D3:97:D5:97:28:A1:20:AE:A7:83:0C:7B:1A:CB:18:3A

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.9.4.1.5

9 NERATEC-SW6-BLUETOOTH-MIB

9.1 bluetooth

9.1.1 configuration

9.1.1.1 cfgBluetooth

9.1.1.1.1 cfgBtEnabled

Bluetooth Service Disabled or Enabled

If the Bluetooth service is **enabled(1)**

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.41524.1.2.11.1.1.1

9.1.1.1.2 cfgBtScanner

9.1.1.1.2.1 cfgBtScnWindow

TECHPREVIEW: Bluetooth Scanner Scan Window

Defines the scan window duration in seconds for the BLE scanner. Values -1 and 0 use the default scanner behavior.

Range	-1 - 7200
Access	readwrite
OID	1.3.6.1.4.1.41524.1.2.11.1.1.100.1

9.1.1.1.2.2 cfgBtScnMacOutput

TECHPREVIEW: Bluetooth Scanner MAC Output

Controls whether individual MAC addresses are printed on each scan window summary.

- **disabled(0)**: Only the MAC count is printed (default)
- **enabled(1)**: MAC addresses are printed along with the count

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.41524.1.2.11.1.1.100.2

9.1.1.1.3 cfgBtApplication

Bluetooth Application Selector

- **none(0)**: No application specified (default)
- **scanner(1)**: Scanner application

Enumeration	none (0), scanner (1)
Access	readwrite
OID	1.3.6.1.4.1.41524.1.2.11.1.1.2

9.1.2 software

9.1.2.1 swBluetooth

9.1.2.1.1 swBtScanner

9.1.2.1.1.1 swBtScnMacCount

TECHPREVIEW: Scanner MAC Count

Number of unique MAC addresses scanned during the last scan window.

Type	Integer32
Access	readonly
OID	1.3.6.1.4.1.41524.1.2.11.2.1.100.1

9.1.2.1.1.2 swBtScnResultTable

Scanner MAC Result Table

Range	0 - 1023
OID	1.3.6.1.4.1.41524.1.2.11.2.1.100.100

swBtScnResultEntry

Entry of Scanner MAC Result Table

OID	1.3.6.1.4.1.41524.1.2.11.2.1.100.100.1

swBtScnResultIndex

Entry Index of Scanner MAC Result Table

Range	0 - 1023
Access	noaccess
OID	1.3.6.1.4.1.41524.1.2.11.2.1.100.100.1.1

swBtScnMacAddress

TECHPREVIEW: Scanned MAC address

Type	DisplayString
Range	0 - 17
Access	readonly
OID	1.3.6.1.4.1.41524.1.2.11.2.1.100.100.1.2

swBtScnMacType

TECHPREVIEW: Scanned MAC address type

Type	DisplayString
Range	0 - 16
Access	readonly
OID	1.3.6.1.4.1.41524.1.2.11.2.1.100.100.1.3

swBtScnName

TECHPREVIEW: Scanned Bluetooth name

Type	DisplayString
Range	0 - 255
Access	readonly
OID	1.3.6.1.4.1.41524.1.2.11.2.1.100.100.1.4

10 WESTERMO-SW6-BRIDGE-MIB

10.1 rstp

10.1.1 configuration

10.1.1.1 cfgRstpBridge

10.1.1.1.1 cfgRstpBridgeEnabled

Disable or Enable RSTP on Bridge

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.8.1.1.1

10.1.1.1.2 cfgRstpBridgePriority

RSTP Bridge Priority

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.

Range	0 - 61440
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.8.1.1.2

10.1.1.1.3 cfgRstpBridgeHelloTime

RSTP Hello Message Send Interval

The interval in seconds between transmissions of hello messages by designated ports. With RSTP this value is 2 seconds.

Range	2 - 2
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.2.8.1.1.3

10.1.1.1.4 cfgRstpBridgeForwardDelay

RSTP Bridge Forward Delay

The delay in seconds used by STP bridges to transition root and designated ports to forwarding.

Range	4 - 30
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.8.1.1.4

10.1.1.1.5 cfgRstpBridgeMaxAge

RSTP Bridge Maximum Age

The maximum age of the information in seconds transmitted by the bridge when it is the root bridge.

Range	6 - 40
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.8.1.1.5

10.1.1.1.6 cfgRstpBridgeTransmitHoldCount

RSTP Bridge Transmit Hold Count

The transmit hold count used by the port transmit state machine to limit transmission rate.

Range	1 - 10
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.8.1.1.6

10.1.1.2 cfgRstpPort

10.1.1.2.1 cfgRstpPortTable

RSTP Port Configuration Table

Range	0 - 18
OID	1.3.6.1.4.1.16177.1.400.2.8.1.2.1

10.1.1.2.1.1 cfgRstpPortTableEntry

RSTP Port Configuration Table Entry

OID	1.3.6.1.4.1.16177.1.400.2.8.1.2.1.1

cfgRstpPortIndex

Entry Index of Port Forward Table

Range	0 - 18
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.2.8.1.2.1.1.1

cfgRstpPortEnabled

Disable or Enable RSTP on Port

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.8.1.2.1.1.2

cfgRstpPortName

Port Name to Apply Settings

Examples:

- eth0
- wlan0

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.2.8.1.2.1.1.3

cfgRstpPortPriority

RSTP Port Priority

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.

Range	0 - 240
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.8.1.2.1.1.4

cfgRstpPortPathCost

RSTP Port Path Cost

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

Range	-1 - 5000000
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.8.1.2.1.1.5

cfgRstpPortAutoEdge

RSTP Port Auto Edge

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.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.8.1.2.1.1.6

11 WESTERMO-SW6-FIREWALL-MIB

11.1 firewall

11.1.1 configuration

11.1.1.1 cfgFwEnabled

L3 Firewall Disabled or Enabled

Globally disable or enable the functionality of NAT `cfgFwNat`, Layer 3 filters `cfgFwFilter` and L3 mangle `cfgFwMangle`.

Note: Layer 2 filters `cfgFwL2IpFilter` Layer 2 mangle are not related. They must be enabled globally with the parameter `cfgFwL2IpFilterEnabled`. respectively `cfgFwL2MangleEnabled`.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.1

11.1.1.2 cfgFwNat

11.1.1.2.1 cfgFwNatPortForwardTable

Firewall Port Forward Rules Table (DNAT)

Range	0 - 255
OID	1.3.6.1.4.1.16177.1.400.2.1.1.2.1

11.1.1.2.1.1 cfgFwNatPortForwardTableEntry

Firewall Port Forward Rules Table (DNAT)

OID	1.3.6.1.4.1.16177.1.400.2.1.1.2.1.1

cfgFwNatPrtFwdIndex

Entry Index of Port Forward Table

Range	0 - 255
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.2.1.1.2.1.1.1

cfgFwNatPrtFwdDestinationPortEnd

Destination End Port to Redirect

This parameter is active when `cfgFwNatPrtFwdProtocol` is set to **udp(1)**, **tcp(2)** or **udptcp(3)**.

Specifies the end of a port range to which a connection is destined.

Set to `-1` to disable this parameter and only match the port specified in `cfgFwNatPrtFwdDestinationPortStart`

Note: Usually this should be `-1`.

Range	-1 - 65535
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.2.1.1.10

cfgFwNatPrtFwdRedirectDestinationAddress

Redirect Traffic to this Redirection Destination Address

Frames matched by this rule have their destination address rewritten to the value specified here.

Type	IpAddress
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.2.1.1.11

cfgFwNatPrtFwdRedirectDestinationPort

Redirect Traffic to this Destination Port

This parameter is active when `cfgFwNatPrtFwdProtocol` is set to **udp(1)**, **tcp(2)** or **udptcp(3)**.

Frames matched by this rule have their destination port rewritten to the value specified here.

Note: Even when a range of ports is matched, the destination port is rewritten to the single port specified here.

Range	-1 - 65535
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.2.1.1.12

cfgFwNatPrtFwdComment

User Comment

This parameter has no operational function. It allows to store a comment about the use of this port forward.

Type	DisplayString
Range	0 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.2.1.1.13

cfgFwNatPrtFwdEnabled

Disable or Enable this Rule

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.2.1.1.2

cfgFwNatPrtFwdInterface

Name of the Network Interface on Which the Rule Applies

Defines on which interface traffic is comming in. Groups of interfaces may be matched by adding the character + at the end.

Examples:

- br0.vlan0
- eth1
- eth+
- br0.vlan+
- -1

If you don't know the inbound interface and want to match all, set this parameter to `-1`.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.2.1.1.3

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

Enumeration	any (0), udp (1), tcp (2), udptcp (3)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.2.1.1.4

cfgFwNatPrtFwdSourceAddress

Source Address to Match

This is a specific IP address or a range in CIDR notation.

An exclamation mark `!` before the address/network may be used to invert the sense of the rule, e.g `!192.168.0.0/24`.

Use this parameter to restrict the source of traffic on which the rule is applied.

Examples:

- 172.17.29.7/32: Match the specific source IP 172.17.29.7
- 0.0.0.0/0: Match all sources addresses
- 192.168.0.0/24: Match the specified source network
- !192.168.0.0/24: Match any source, except the specified network

Note: Usually this should be `0.0.0.0/0`.

Type	DisplayString
Range	7 - 19
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.2.1.1.5

cfgFwNatPrtFwdSourcePortStart

Source Start Port to Match

This parameter is active when `cfgFwNatPrtFwdProtocol` is set to **udp(1)**, **tcp(2)** or **udptcp(3)**.

Specify a port or the start of a port range from which a connection originates.

Set to `-1` to disable this parameter.

An exclamation mark `!` before the port may be used to invert the sense of the rule, e.g `!80`.

When used in a range, the inversion applies to the range.

Use `cfgFwNatPrtFwdSourcePortEnd` to specify the end of the range.

Note: Usually this should be `-1`.

Type	DisplayString
Range	1 - 6
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.2.1.1.6

cfgFwNatPrtFwdSourcePortEnd

Source End Port to Match

This parameter is active when `cfgFwNatPrtFwdProtocol` is set to **udp(1)**, **tcp(2)** or **udptcp(3)**.

Specifies the end of a port range from which a connection originates.

Set to `-1` to disable this parameter and only match the port specified in `cfgFwNatPrtFwdSourcePortStart`.

Note: Usually this should be `-1`.

Range	-1 - 65535
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.2.1.1.7

cfgFwNatPrtFwdDestinationAddress

Destination Address to Redirect

This is a specific IP address or a range in CIDR notation.

Use this parameter to restrict the destination of traffic on which the rule is applied.

Set to `0.0.0.0/0` to match all destinations of inbound traffic on the interface specified in `cfgFwNatPrtFwdInterf`

An exclamation mark `!` before the destination may be used 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.

Examples:

- 172.17.29.7/32: Match the specific destination IP 172.17.29.7
- 0.0.0.0/0: Match all destination addresses
- 192.168.0.0/24: Match the specified destination network
- !192.168.0.0/24: Match any destination, except this network

Note: This should be `0.0.0.0/0` when using DHCP.

Type	DisplayString
Range	7 - 19
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.2.1.1.8

cfgFwNatPrtFwdDestinationPortStart

Destination Start Port to Redirect

This parameter is active when `cfgFwNatPrtFwdProtocol` is set to **udp(1)**, **tcp(2)** or **udptcp(3)**.

Specify the destination port or start of a port range.

Set to `-1` to disable this parameter and match any port.

An exclamation mark `!` before the port may be used to invert the sense of the rule, e.g `!80`.

When used in a range, the inversion applies to the range.

Use `cfgFwNatPrtFwdDestinationPortEnd` to specify the end of the range.

Type	DisplayString
Range	1 - 6
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.2.1.1.9

11.1.1.2.2 cfgFwNatOutboundTable

Firewall Outbound NAT Rules Table (SNAT)

Range	0 - 255
OID	1.3.6.1.4.1.16177.1.400.2.1.1.2.2

11.1.1.2.2.1 cfgFwNatOutboundTableEntry

Firewall Outbound NAT Rules Table (SNAT)

OID	1.3.6.1.4.1.16177.1.400.2.1.1.2.2.1

cfgFwNatOutIndex

Table Entry Index

Range	0 - 255
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.2.1.1.2.2.1.1

cfgFwNatOutDestinationPortEnd

Destination End Port to Redirect

This parameter is active when `cfgFwNatOutProtocol` is set to **udp(1)**, **tcp(2)** or **udptcp(3)**.

Specifies the end of a port range to which a connection is destined.

Set to `-1` to disable this parameter and only match the port specified in `cfgFwNatOutDestinationPortStart`.

Note: Usually this should be `-1`.

Range	-1 - 65535
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.2.2.1.10

cfgFwNatOutSourceRewriteAddress

Source Rewrite Address

Frames matched by this rule have their source address rewritten to the value specified here.

When this parameter is set to `0.0.0.0`, the address is rewritten to the configured primary (first) address of the interface.

In case you are using DHCP where the address is unknown in advance, also set this parameter on `0.0.0.0`.

Type	IpAddress
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.2.2.1.11

cfgFwNatOutSourceRewritePort

Source Rewrite Port

This parameter is active when `cfgFwNatOutProtocol` is set to **udp(1)**, **tcp(2)** or **udptcp(3)**.

Frames matched by this rule have their source port rewritten to the value specified here.

Set this parameter to `-1` to disable source port rewriting.

Note: Usually this should be `-1`.

Range	-1 - 65535
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.2.2.1.12

cfgFwNatOutComment

User Comment

This parameter has no operational function. It allows to store a comment about the use of this outbound rule.

Type	DisplayString
Range	0 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.2.2.1.13

cfgFwNatOutEnabled

Disable or Enable this Rule

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.2.2.1.2

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, and match the traffic with `cfgFwNatOutDestinationAddress` instead.

Groups of interfaces may be matched by adding the character + at the end. E.g. `eth+` to match the interfaces eth0, eth1.

An exclamation mark `!` before the interface may be used to invert the sense of the rule, e.g `!wlan0`.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.2.2.1.3

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

Enumeration	any (0), udp (1), tcp (2), udptcp (3)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.2.2.1.4

cfgFwNatOutSourceAddress

Source Address to Match

This is a specific IP address or a range in CIDR notation.

An exclamation mark `!` before the address/network may be used to invert the sense of the rule, e.g `!192.168.0.0/24`.

Use this parameter to restrict the source of traffic on which the rule is applied.

Examples:

- 172.17.29.7/32: Match the specific source IP 172.17.29.7
- 0.0.0.0/0: Match all sources addresses
- 192.168.0.0/24: Match the specified source network

- !192.168.0.0/24: Match any source, except the specified network

Note: Usually this should be `0.0.0.0/0`.

Type	DisplayString
Range	7 - 19
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.2.2.1.5

cfgFwNatOutSourcePortStart

Source Start Port to Match

This parameter is active when `cfgFwNatOutProtocol` is set to **udp(1)**, **tcp(2)** or **udptcp(3)**.

Specify a port or the start of a port range from which a connection originates.

Set to `-1` to disable this parameter.

An exclamation mark `!` before the port may be used to invert the sense of the rule, e.g `!80`.

When used in a range, the inversion applies to the range.

Use `cfgFwNatOutSourcePortEnd` to specify the end of the range.

Note: Usually this should be `-1`.

Type	DisplayString
Range	1 - 6
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.2.2.1.6

cfgFwNatOutSourcePortEnd

Source End Port to Match

This parameter is active when `cfgFwNatOutProtocol` is set to **udp(1)**, **tcp(2)** or **udptcp(3)**.

Specifies the end of a port range from which a connection originates.

Set to `-1` to disable this parameter and only match the port specified in `cfgFwNatOutSourcePortStart`.

Note: Usually this should be `-1`.

Range	-1 - 65535
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.2.2.1.7

cfgFwNatOutDestinationAddress

Destination Address to Match

This is a specific IP address or a range in CIDR notation.

Use this parameter to restrict the destination of traffic on which the rule is applied.

Set to `0.0.0.0/0` to match all destinations of outbound traffic on the interface specified in `cfgFwNatOutInterfac`

An exclamation mark `!` before the destination may be used to invert the sense of the rule, e.g `!192.168.0.0/24`.

Examples:

- 172.17.29.7/32: Match the specific destination IP 172.17.29.7
- 0.0.0.0/0: Match all destination addresses
- 192.168.0.0/24: Match the specified destination network
- !192.168.0.0/24: Match any destination, except this network

Type	DisplayString
Range	7 - 19
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.2.2.1.8

cfgFwNatOutDestinationPortStart

Destination Start Port to Match

This parameter is active when `cfgFwNatOutProtocol` is set to **udp(1)**, **tcp(2)** or **udptcp(3)**.

Specify the destination port or start of a port range.

Set to `-1` to disable this parameter and match any port.

An exclamation mark `!` before the port may be used to invert the sense of the rule, e.g `!80`.

When used in a range, the inversion applies to the range.

Use `cfgFwNatOutDestinationPortEnd` to specify the end of the range.

Note: Usually this should be `-1`.

Type	DisplayString
Range	1 - 6
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.2.2.1.9

11.1.1.2.3 cfgFwNatOneToOneTable

Netmap Table

The netmap table allows to rewrite the source respective destination of a whole network. This is an advantage over the `cfgFwNatPortForwardTable` and `cfgFwNatOutboundTable` which only allow to rewrite a single source respective a single destination.

It provides 3 types of operation:

- Auto mode where inbound and outbound frames automatically have their source, respectively destination, rewritten.
- DNAT mode where inbound frames to the specified `cfgFwNatOneToOneDestinationNet` have their destination rewritten to `cfgFwNatOneToOneRewriteNet`
- SNAT mode where outbound frames from the the specified `cfgFwNatOneToOneSourceNet` have their source rewritten to `cfgFwNatOneToOneRewriteNet`.

Range	0 - 255
OID	1.3.6.1.4.1.16177.1.400.2.1.1.2.3

11.1.1.2.3.1 cfgFwNatOneToOneTableEntry

Netmap Table Entry

OID	1.3.6.1.4.1.16177.1.400.2.1.1.2.3.1

cfgFwNatOneToOneIndex

Table Entry Index

Range	0 - 255
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.2.1.1.2.3.1.1

cfgFwNatOneToOneEnabled

Disable or Enable this Rule

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.2.3.1.2

cfgFwNatOneToOneType

Type of Rule

This parameter defines the type of netmap:

- **auto(0)**: A rule in auto mode creates automatically a DNAT and an SNAT rule for the specified source, destination and rewrite networks, also known as 1:1 NAT.
- **dnat(1)**: A rule in DNAT mode rewrites the destination network to the specified rewrite network.
- **snat(2)**: A rule in SNAT mode rewrites the source network to the specified rewrite network.

Enumeration	auto (0), dnat (1), snat (2)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.2.3.1.3

cfgFwNatOneToOneInterface

Interface

Depending on the selected type in `cfgFwNatOneToOneType`:

For the types **auto(0)** and **dnat(1)** it refers to the interface on which traffic ingresses (input interface).

For the type **snat(2)** it refers to the interface on which traffic egresses (output interface).

Set to `-1` to perform netmap on traffic ingressing/egressing on any interface, only constrained by the configuration in `cfgFwNatOneToOneSourceNet` and `cfgFwNatOneToOneDestinationNet`.

Groups of interfaces can be matched by adding the character + at the end. E.g. `eth+` to match the interfaces eth0, eth1.

Examples:

- -1
- br0.vlan0
- br0.vlan+
- wlan0
- eth0

Type	DisplayString
Range	1 - 15
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.2.3.1.4

cfgFwNatOneToOneSourceNet

Source Network

This is a source network in CIDR notation.

When `cfgFwNatOneToOneType` is set to **auto(0)** or **dnat(1)**, frames matched by this field are processed by this rule.

Set to `0.0.0.0/0` to match all inbound traffic.

When `cfgFwNatOneToOneType` is set to **snat(2)** the network specified here will be rewritten to the network specified in `cfgFwNatOneToOneRewriteNet`.

Type	DisplayString
Range	9 - 18
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.2.3.1.5

cfgFwNatOneToOneDestinationNet

Destination Network

This is a destination network in CIDR notation.

When `cfgFwNatOneToOneType` is set to **auto(0)** or **dnat(1)**, the network specified here will be rewritten to the network specified in `cfgFwNatOneToOneRewriteNet`.

When `cfgFwNatOneToOneType` is set to **snat(2)**, only frames matched by this field are processed by this rule.

Set to `0.0.0.0/0` to match all outbound traffic.

Type	DisplayString
Range	9 - 18
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.2.3.1.6

cfgFwNatOneToOneRewriteNet

Rewrite Network

This is the network in CIDR notation to/from which frames are rewritten.

Depending on the setting of `cfgFwNatOneToOneType`:

- **auto(0)**: For inbound traffic `cfgFwNatOneToOneRewriteNet` is the network to which the destination is rewritten to `cfgFwNatOneToOneDestinationNet`. For outbound traffic `cfgFwNatOneToOneRewri` is the network which matches the source that is rewritten to `cfgFwNatOneToOneRewriteNet`.
- **dnat(1)**: `cfgFwNatOneToOneRewriteNet` is the network to which the dest. net (`cfgFwNatOneToOneDesti` is rewritten to.
- **snat(2)**: `cfgFwNatOneToOneRewriteNet` is the network to which the source net (`cfgFwNatOneToOneSour` is rewritten to.

Type	DisplayString
Range	9 - 18
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.2.3.1.7

cfgFwNatOneToOneComment

User Comment

This parameter has no operational function. It allows to store a comment about the use of this netmap rule.

Type	DisplayString
Range	0 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.2.3.1.8

11.1.1.3 cfgFwL2IpFilter

11.1.1.3.1 cfgFwL2IpFilterEnabled

Globally Disable or Enabled the L2 Filter

When enabled, filter rules will be installed on all bridges.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.3.1

11.1.1.3.2 cfgFwL2IpFilterDefaultAction

Global Default Action

The default action that is performed on a filtering bridge. Two bridges can not have a different default action.

Take care to not lock yourself out when the default action is set to **drop(1)**'.

Enumeration	accept (0), drop (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.3.2

11.1.1.3.3 cfgFwL2IpFilterTable

L2 Filter

The L2 filter allows to create filter rules on a bridge. Such a filter may be useful to prohibit traffic between clients of different APs.

Range	0 - 255
OID	1.3.6.1.4.1.16177.1.400.2.1.1.3.3

11.1.1.3.3.1 cfgFwL2IpFilterTableEntry

L2 Filter Entry

OID	1.3.6.1.4.1.16177.1.400.2.1.1.3.3.1

cfgFwL2IpFltrIndex

Table Entry Index

Range	0 - 255
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.2.1.1.3.3.1.1

cfgFwL2IpFltrSourcePort

The Source Port On Which The Rule Matches

Set to `-1`, to not use the source port to match.

When `cfgFwL2IpFltrEthertype` is set to `0800` and `cfgFwL2IpFltrProtocol` is `6` (TCP) or `17` (UDP), this field may be used to match a specific source port.

Range	-1 - 65535
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.3.3.1.10

cfgFwL2IpFltrDestinationPort

The Destination Port On Which The Rule Matches

Set to `-1`, to not use the destination port to match.

When `cfgFwL2IpFltrEthertype` is set to `0800` and `cfgFwL2IpFltrProtocol` is `6` (TCP) or `17` (UDP), this field may be used to match a specific source port.

Range	-1 - 65535
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.3.3.1.11

cfgFwL2IpFltrSourceMac

The Source MAC Addresses On Which The Rule Matches

This is a MAC address and a mask in the form of `xx:xx:xx:xx:xx:xx/yy:yy:yy:yy:yy:yy`.

Examples:

- `00:00:00:00:00:00/00:00:00:00:00:00` match any source
- `00:14:5a:02:04:4c/ff:ff:ff:ff:ff:ff` exact match of the source address 00:14:5a:02:04:4c
- `00:14:5a:00:00:00/ff:ff:ff:00:00:00` match any source addresses with the vendor OUI `00:14:5a`

Type	DisplayString
Range	35 - 35
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.3.3.1.12

cfgFwL2IpFltrDestinationMac

The Destination MAC Addresses On Which The Rule Matches

This is a MAC address and a mask in the form of `xx:xx:xx:xx:xx:xx/yy:yy:yy:yy:yy:yy`.

Examples:

- `00:00:00:00:00:00/00:00:00:00:00:00` match any destination

- `01:00:00:00:00:00/01:00:00:00:00:00` match all frames with a multicast destination (including broadcast)
- `00:14:5a:02:04:4c/ff:ff:ff:ff:ff:ff` exact match of the destination address 00:14:5a:02:04:4c
- `00:14:5a:00:00:00/ff:ff:ff:00:00:00` match any destination addresses with the vendor OUI `00:14:5a`

Type	DisplayString
Range	35 - 35
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.3.3.1.13

cfgFwL2IpFltrEthertype

The Ethertype On Which The Rule Matches

This is an ethertype in the form of `XXXX` where each X is a hexadecimal character [0-9A-Fa-f].

Default is `0800`, which represents IPv4.

Set to `0000`, to match any ethertype.

Examples:

- 0000: any
- 0800: IPv4
- 0806: ARP
- 8035: RARP
- 86DD: IPv6
- 8847: MPLS unicast
- 8848: MPLS multicast
- 8892: Profinet

For a full list of IANA assigned ethertypes see https://www.iana.org/assignments/ieee-802-numbers/ieee-802-numbers.xhtml

Type	DisplayString
Range	4 - 4
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.3.3.1.14

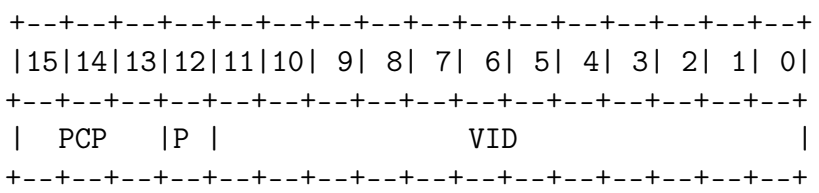
cfgFwL2IpFltrVlan

The VLAN On Which The Rule Matches

Set to `-1`, to not use the VLAN to match.

This field has two modes of operation:

- When set to a decimal number between 0 and 4094, this matches on the VLAN number in the 802.1Q header when present. Can only be used to match tagged frames.
- When set in the format `0x0000/0x0000` matches the raw vlan_tci.



  Bits 0-11 (VID) are the vlan number. Bit 12 (P) represents the existence of a vlan header. Bits 13-15 (PCP) are the vlan priority. The first part before the / is the value to match on. The second part after the / is the mask applied on the value. This can be used to match tagged or untagged frames, ranges of vlans or the priority.

Examples:

- `-1`: Disable matching on VLAN
- `200`: Match on VLAN 200
- `0x0000/0x1000`: Match frames with no 802.1Q header
- `0x1000/0x1000`: Match any frame with a 802.1Q header
- `0x5000/0xf000`: Match frames with priority 2 in any VLAN
- `0x0000/0x0fff`: Match frames with no 802.1Q header or tagged with VLAN 0 and any priority
- `0x0000/0xefff`: Match frames with no 802.1Q header or tagged with VLAN 0 and priority 0

NOTE: Frames that ingress on an access ports are not considered part of the assigned vlan yet. Local VLAN interfaces are access ports, e.g. br0.vlan7.

Type	DisplayString
Range	1 - 13
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.3.3.1.15

cfgFwL2IpFltrInputInterface

The Ingress Interface On Which The Rule Matches

The name of the interface on which a frame ingresses.

Set to `any`, to match any interface.

Examples:

- **any**
- **wlan0**
- **eth0**

Type	DisplayString
Range	1 - 15
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.3.3.1.16

cfgFwL2IpFltrEnabled

Disabled or Enabled this Rule

Only enabled rules are installed on filtering bridges.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.3.3.1.2

cfgFwL2IpFltrBridge

Bridge on Which the Rule is Created

This rule is only created on the bridge selected here.

Take care when setting `cfgFwL2IpFilterDefaultAction` to **drop(1)** that you have a rule on your management bridge allowing access to the management interface.

Range	-1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.3.3.1.3

cfgFwL2IpFltrAction

Action to Perform

- **accept(0)**: let the frame continue to be processed
- **drop(1)**: drop the frame immediatelly

Enumeration	accept (0), drop (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.3.3.1.4

cfgFwL2IpFltrPriority

Filter Priority

When multiple rules match, the rule with the highest priority is applied.

Multiple overlapping rules with the same priority have undefined behaviour, thus are not allowed.

Range	1 - 65535
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.3.3.1.5

cfgFwL2IpFltrSource

Source Network/IP On Which The Rule Matches (CIDR Notation)

When `cfgFwL2IpFilterMode` is set to **full(1)**, then this field only has an effect when `cfgFwL2IpFltrEthertype` is set to `0800`, `0806` or `8035`.

When `cfgFwL2IpFltrEthertype` is set to `0800` this is an IP or network.

When `cfgFwL2IpFltrEthertype` is set to `0806` or `8035` it is the source address or a range of source addresses of an ARP/RARP frame (ARP_SPA).

Type	DisplayString
Range	7 - 18
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.3.3.1.6

cfgFwL2IpFltrDestination

Destination Network/IP On Which The Rule Matches (CIDR Notation)

When `cfgFwL2IpFilterMode` is set to **full(1)**, then this field only has an effect when `cfgFwL2IpFltrEthertype` is set to `0800`, `0806` or `8035`.

When `cfgFwL2IpFltrEthertype` is set to `0800` this is an IP or network.

When `cfgFwL2IpFltrEthertype` is set to `0806` or `8035` it is the target address or a range of target addresses of an ARP/RARP frame (ARP_TPA).

Type	DisplayString
Range	7 - 18
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.3.3.1.7

cfgFwL2IpFltrComment

User Comment

This parameter has no operational function. It allows to store a comment about the use of this l2 filter rule.

Type	DisplayString
Range	0 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.3.3.1.8

cfgFwL2IpFltrProtocol

IP Protocol On Which The Rule Matches

Set to `-1`, to not use the protocol to match.

When `cfgFwL2IpFltrEthertype` is set to `0800`, this field may be used to match a specific IP protocol.

Examples:

- **-1**: any protocol
- **1**: ICMP
- **2**: IGMP
- **6**: TCP
- **17**: UDP
- **50**: ESP (IPsec)
- **51**: AH (IPsec)
- **112**: VRRP / CARP

For a full list of available protocols see https://en.wikipedia.org/wiki/List_of_IP_protocol_numbers

When `cfgFwL2IpFltrEthertype` is set to `0806` or `8035`, this field may be used to match the OP code of an ARP frame.

Examples:

- **-1**: any OP code
- **1**: request
- **2**: response
- **3**: request reverse
- **4**: response reverse

Range	-1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.3.3.1.9

11.1.1.3.4 cfgFwL2IpFilterMode

Filter Mode

The filtering bridge provides two modes of operation:

- **ip(0)**: A simplified mode to filter IP traffic. All non-IP traffic is allowed.
- **full(1)**: The full mode allows to filter on any kind of header information present.

Enumeration	ip (0), full (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.3.4

11.1.1.4 cfgFwFilter

11.1.1.4.1 cfgFwFltDefaultPolicyInput

The Default Filter Policy on the Input Path

Enumeration	drop (0), accept (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.4.1

11.1.1.4.2 cfgFwFilterRulesTable

Firewall Filter Rules Table

Range	0 - 255
OID	1.3.6.1.4.1.16177.1.400.2.1.1.4.10

11.1.1.4.2.1 cfgFwFilterRulesTableEntry

Firewall Filter Rules Table

OID	1.3.6.1.4.1.16177.1.400.2.1.1.4.10.1

cfgFwFltRIndex

Table Entry Index

Range	0 - 255
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.2.1.1.4.10.1.1

cfgFwFltRSourcePortEnd

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.

Range	-1 - 65535
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.4.10.1.10

cfgFwFltRDestinationAddress

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`.

Type	DisplayString
Range	1 - 20
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.4.10.1.11

cfgFwFltRDestinationPortStart

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.

Type	DisplayString
Range	1 - 20
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.4.10.1.12

cfgFwFltRDestinationPortEnd

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.

Range	-1 - 65535
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.4.10.1.13

cfgFwFltRComment

User Comment

This parameter has no operational function. It allows to store a comment about the use of this filter rule.

Type	DisplayString
Range	0 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.4.10.1.14

cfgFwFltRIcmpType

ICMP Type

When `cfgFwFltRProtocol` is set to **icmp(1)** this parameter may be used to filter specific ICMP types.

Set to `-1` to match any type.

For a list of types officially assigned by IANA see: https://www.iana.org/assignments/icmp-parameters/icmp-parameters.xhtml

Examples:

- **any(-1)**
- **echo reply(0)**
- **destination unreachable(3)**
- **redirect(5)**
- **echo(8)**
- **ttl exceeded(11)**

When blocking type 3, be aware that this breaks PMTUD (Path MTU Discovery).

Range	-1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.4.10.1.15

cfgFwFltRIcmpCode

ICMP Code

When `cfgFwFltRProtocol` is set to **icmp(1)** this parameter may be used to filter specific ICMP codes.

Depending on `cfgFwFltRIcmpType` different types have different codes. This parameter has no function when `cfgFwFltRIcmpType` is set to **any(-1)**.

Set to `-1` to match any code.

For a list of codes of each type officially assigned by IANA see: https://www.iana.org/assignments/icmp-parameters/icmp-parameters.xhtml

Examples:

- **any(-1)**
- **Protocol Unreachable(2)** of type destination unreachable(3)
- **Redirect Datagram for the Host(1)** of type redirect(5)

Range	-1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.4.10.1.16

cfgFwFltRLog

Enable Logging For This Rule

Configures whether frames handled by this rule are logged.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.4.10.1.17

cfgFwFltRLimit

Rule Frame Limit

This parameter is active when `cfgFwFltRAction` is set to **accept(1)** and `cfgFwFltRLimitBurst` has a value greater than 0.

This is the rate at which ACCEPT-tokens are added to the bucket for this rules.

In the form of rate[/second|/minute|/hour|/day]

The maximum rate is 10’000/second, respective 600’000/minute, 36’000’000/hour and 864’000’000/day.

Examples:

- 1/second
- 5/minute
- 100/hour
- 10000/day

Type	DisplayString
Range	1 - 13
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.4.10.1.18

cfgFwFltRLimitBurst

Rule Frame Limit Burst

This parameter is active when `cfgFwFltRAction` is set to **accept(1)**.

This is the size of the bucket containing ACCEPT-tokens, for this rule. Each new frame consumes 1 token. When the bucket has zero tokens, new frames are dropped.

New tokens are added to the bucket at the rate specified in `cfgFwFltRLimit`.

Set to `0` to disable the rate limiter.

This value is also the amount of tokens in the bucket at start.

Range	0 - 10000
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.4.10.1.19

cfgFwFltREnabled

Disable or Enable This Rule

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.4.10.1.2

cfgFwFltRChain

Chain on Which Action is Performed

Enumeration	none (0), input (1), forward (2), output (3)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.4.10.1.3

cfgFwFltRAction

Action to be Performed

- **drop(0):** New frames that match this rule are dropped
- **accept(1):** New frames that match this rule are accepted

Enumeration	drop (0), accept (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.4.10.1.4

cfgFwFltRInputInterface

Name of the Input Interface to Match

This parameter may be used when `cfgFwFltRChain` is set to **input(1)** or **forward(2)**.

Groups of interfaces can be matched by adding the character '+' at the end. E.g. eth+ to match the interfaces eth0, eth1 and eth2.

Set to `-1` to not use this parameter.

Type	DisplayString
Range	1 - 16
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.4.10.1.5

cfgFwFltROutputInterface

Name of the Output Interface to Match

This parameter may be used when `cfgFwFltRChain` is set to **forward(2)** or **output(3)**.

Groups of interfaces can be matched by adding the character '+' at the end. E.g. eth+ to match the interfaces eth0, eth1 and eth2.

Set to `-1` to not use this parameter.

Type	DisplayString
Range	1 - 16
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.4.10.1.6

cfgFwFltRProtocol

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

Examples:

- **any(-1)**: 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)**

Range	-1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.4.10.1.7

cfgFwFltRSourceAddress

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`.

Type	DisplayString
Range	1 - 20
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.4.10.1.8

cfgFwFltRSourcePortStart

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.

Type	DisplayString
Range	1 - 20
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.4.10.1.9

11.1.1.4.3 cfgFwFltDefaultPolicyForward

The Default Filter Policy on the Forward Path

Enumeration	drop (0), accept (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.4.2

11.1.1.4.4 cfgFwFltDefaultPolicyOutput

The Default Filter Policy on the Output Path

Enumeration	drop (0), accept (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.4.3

11.1.1.4.5 cfgFwFltLogDefaultDrop

Disable or Enable Logging of the Default Drop Policy

When this option is set to **enabled(1)**, all frames that are dropped by the default policy on a chain are logged.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.4.4

11.1.1.4.6 cfgFwFltLogDefaultAccept

Disable or Enable Logging of the Default Accept Policy

When this option is set to **enabled(1)**, all frames that are accepted by the default policy on a chain are logged.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.4.5

11.1.1.4.7 cfgFwFltInputSynLimit

Input Syn Frame Limit

This parameter is active when `cfgFwFltInputSynLimitBurst` has a value greater than 0.

This is the rate at which ACCEPT-tokens are added to the bucket for the TCP SYN flood protection on the INPUT chain.

In the form of rate[/second|/minute|/hour|/day]

The maximum rate is 10'000/second, respective 600'000/minute, 36'000'000/hour and 864'000'000/day.

Examples:

- 1/second
- 5/minute
- 100/hour
- 10000/day

Type	DisplayString
Range	1 - 13
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.4.6

11.1.1.4.8 cfgFwFltInputSynLimitBurst

Input Syn Frame Limit Burst

This is the size of the bucket containing ACCEPT-tokens, for the TCP syn flood protection on the INPUT chain. Each new TCP SYN frame consumes 1 token. When the bucket has zero tokens, new TCP SYN frames are dropped.

New tokens are added to the bucket at the rate specified in `cfgFwFltInputSynLimit`.

Set to `0` to disable the INPUT TCP SYN rate limiter.

This value is also the amount of tokens in the bucket at start.

Range	0 - 10000
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.4.7

11.1.1.5 cfgFwMangle

11.1.1.5.1 cfgFwMangleTable

Firewall Mangle Table

Iptables mangle rules allow to modify frames that are received, transmitted or forwarded.

Range	0 - 255
OID	1.3.6.1.4.1.16177.1.400.2.1.1.5.1

11.1.1.5.1.1 cfgFwMangleTableEntry

Firewall Mangle Table

OID	1.3.6.1.4.1.16177.1.400.2.1.1.5.1.1

cfgFwMnglIndex

Table Entry Index

Range	0 - 255
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.2.1.1.5.1.1.1

cfgFwMnglComment

User Comment

This parameter has no operational function. It allows to store a comment about the use of this mangle rule.

Type	DisplayString
Range	0 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.5.1.1.10

cfgFwMnglProtocol

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

Examples:

- **any(-1)**: 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)**

Range	-1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.5.1.1.11

cfgFwMnglSourcePortStart

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.

Type	DisplayString
Range	1 - 20
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.5.1.1.12

cfgFwMnglSourcePortEnd

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.

Range	-1 - 65535
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.5.1.1.13

cfgFwMnglDestinationPortStart

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.

Type	DisplayString
Range	1 - 20
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.5.1.1.14

cfgFwMngIDestinationPortEnd

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.

Range	-1 - 65535
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.5.1.1.15

cfgFwMngIDscp

DSCP to Match

The DSCP field in the IP header used to specify a priority of a frame.

The DSCP in hexadecimal form 00-FF without 0x. Must be a multiple of 4 (the lowest 2 bits 0).

Set to `-1` when no DSCP is to be matched.

Type	DisplayString
Range	2 - 2
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.5.1.1.16

cfgFwMngIEnabled

Disable or Enable This Rule

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.5.1.1.2

cfgFwMnglChain

Chain on Which Mangle Action is Performed

Depending on the selected chain, `cfgFwMnglInputInterface` and/or `cfgFwMnglOutputInterface` may become active.

Enumeration	input (1), forward (2), output (3), prerouting (4), postrouting (5)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.5.1.1.3

cfgFwMnglAction

Action to be Performed

- **none(0)**: No action is performed.
- **setmss(1)**: Sets the MSS to the value in `cfgFwMnglValue`.
- **ttlset(2)**: Sets the TTL to the value in `cfgFwMnglValue`.
- **setdscp(3)**: Sets the DSCP field in the IP header to the value in `cfgFwMnglValue`.
- **setmark(4)**: Sets the mark of the frame to the value in `cfgFwMnglValue`.
- **clampMssToPmtu(5)**: Clamps the MSS to the detected value of the PMTU.

Enumeration	none (0), setmss (1), ttlset (2), setdscp (3), setmark (4), clampMssToPmtu (5)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.5.1.1.4

cfgFwMnglValue

Value to be Set

Depending on the mode of `cfgFwMnglAction` this value has a different meaning.

- **setmss(1)**: Set the MSS of TCP SYN frames to the specified value. This value should not be greater then the value of the MTU - 40. The value has a range of 28 to 8960.
- **ttlset(2)**: Set the TTL of frames to the specified value. The value has a range of 0 to 255. When a value of 0 is set, this frame can not be routed.
- **setdscp(3)**: Set the DSCP of frames to the specified value. The DSCP in hexadecimal form 00-FF without 0x. Must be a multiple of 4 (the lowest 2 bits 0).
- **setmark(4)**: Set the mark of frames to the specified value. The value has a range of 0 to 4294967295.

Type	DisplayString
Range	1 - 10
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.5.1.1.5

cfgFwMnglInputInterface

Name of the Input Interface to Match

This parameter may be used when `cfgFwMnglChain` is set to **input(1)**, **forward(2)** and **prerouting(4)**.

Groups of interfaces can be matched by adding the character '+' at the end. E.g. `eth+` to match the interfaces eth0, eth1 and eth2.

Set to `-1` to not use this parameter.

Type	DisplayString
Range	1 - 16
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.5.1.1.6

cfgFwMnglOutputInterface

Name of the Output Interface to Match

This parameter may be used when `cfgFwMnglChain` is set to **forward(2)**, **output(3)** and **postrouting(5)**.

Groups of interfaces can be matched by adding the character '+' at the end. E.g. `eth+` to match the interfaces eth0, eth1 and eth2.

Set to `-1` to not use this parameter.

Type	DisplayString
Range	1 - 16
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.5.1.1.7

cfgFwMnglSourceAddress

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 may use ! to invert the sense of

the rule, e.g. `!192.168.0.0/24`.

Type	DisplayString
Range	1 - 20
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.5.1.1.8

cfgFwMnglDestinationAddress

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 may use ! to invert the sense of the rule, e.g. `!192.168.0.0/24`.

Type	DisplayString
Range	1 - 20
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.5.1.1.9

11.1.1.6 cfgFwL2Mangle

11.1.1.6.1 cfgFwL2MangleEnabled

Globally Disable or Enabled L2 Mangling

When enabled, mangle rules will be installed on all bridges.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.6.1

11.1.1.6.2 cfgFwL2MangleTable

L2 Mangle

The L2 mangling allows to create mangle rules on a bridge. Such rules may be used to adjust various header fields of the forwarded frames.

Range	0 - 255
OID	1.3.6.1.4.1.16177.1.400.2.1.1.6.3

11.1.1.6.2.1 cfgFwL2MangleTableEntry

L2 Mangle Entry

OID	1.3.6.1.4.1.16177.1.400.2.1.1.6.3.1

cfgFwL2MnglIndex

Table Entry Index

Range	0 - 255
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.2.1.1.6.3.1.1

cfgFwL2MnglSourcePort

The Source Port On Which The Rule Matches

Set to `-1`, to not use the source port to match.

When `cfgFwL2MnglEthertype` is set to `0800` and `cfgFwL2MnglProtocol` is `6` (TCP) or `17` (UDP), this field may be used to match a specific source port.

Range	-1 - 65535
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.6.3.1.10

cfgFwL2MnglDestinationPort

The Destination Port On Which The Rule Matches

Set to `-1`, to not use the destination port to match.

When `cfgFwL2MnglEthertype` is set to `0800` and `cfgFwL2MnglProtocol` is `6` (TCP) or `17` (UDP), this field may be used to match a specific source port.

Range	-1 - 65535
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.6.3.1.11

cfgFwL2MnglSourceMac

The Source MAC Addresses On Which The Rule Matches

This is a MAC address and a mask in the form of `xx:xx:xx:xx:xx:xx/yy:yy:yy:yy:yy:yy`.

Examples:

- `00:00:00:00:00:00/00:00:00:00:00:00` match any source
- `00:14:5a:02:04:4c/ff:ff:ff:ff:ff:ff` exact match of the source address 00:14:5a:02:04:4c
- `00:14:5a:00:00:00/ff:ff:ff:00:00:00` match any source addresses with the vendor OUI `00:14:5a`

Type	DisplayString
Range	35 - 35
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.6.3.1.12

cfgFwL2MngIDestinationMac

The Destination MAC Addresses On Which The Rule Matches

This is a MAC address and a mask in the form of `xx:xx:xx:xx:xx:xx/yy:yy:yy:yy:yy:yy`.

Examples:

- `00:00:00:00:00:00/00:00:00:00:00:00` match any destination
- `01:00:00:00:00:00/01:00:00:00:00:00` match all frames with a multicast destination (including broadcast)
- `00:14:5a:02:04:4c/ff:ff:ff:ff:ff:ff` exact match of the destination address 00:14:5a:02:04:4c
- `00:14:5a:00:00:00/ff:ff:ff:00:00:00` match any destination addresses with the vendor OUI `00:14:5a`

Type	DisplayString
Range	35 - 35
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.6.3.1.13

cfgFwL2MngIEthertype

The Ethertype On Which The Rule Matches

This is an ethertype in the form of `XXXX` where each X is a hexadecimal character [0-9A-Fa-f].

Default is `0800`, which represents IPv4.

Set to `0000`, to match any ethertype.

Examples:

- 0000: any

- 0800: IPv4
- 0806: ARP
- 8035: RARP
- 86DD: IPv6
- 8847: MPLS unicast
- 8848: MPLS multicast
- 8892: Profinet

For a full list of IANA assigned ethertypes see https://www.iana.org/assignments/ieee-802-numbers/ieee-802-numbers.xhtml

Type	DisplayString
Range	4 - 4
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.6.3.1.14

cfgFwL2MngIVlan

The VLAN On Which The Rule Matches

Set to `-1`, to not use the VLAN to match.

This field has two modes of operation:

- When set to a decimal number between 0 and 4094, this matches on the VLAN number in the 802.1Q header when present. Can only be used to match tagged frames.

- When set in the format `0x0000/0x0000` matches the raw vlan_tci.

  ```
  +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
  |15|14|13|12|11|10| 9| 8| 7| 6| 5| 4| 3| 2| 1| 0|
  +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
  |  PCP   |P |                VID                 |
  +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
  ```

  Bits 0-11 (VID) are the vlan number. Bit 12 (P) represents the existence of a vlan header. Bits 13-15 (PCP) are the vlan priority. The first part before the / is the value to match on. The second part after the / is the mask applied on the value. This can be used to match tagged or untagged frames, ranges of vlans or the priority.

Examples:

- `-1`: Disable matching on VLAN
- `200`: Match on VLAN 200
- `0x0000/0x1000`: Match frames with no 802.1Q header

- `0x1000/0x1000`: Match any frame with a 802.1Q header
- `0x5000/0xf000`: Match frames with priority 2 in any VLAN
- `0x0000/0x0fff`: Match frames with no 802.1Q header or tagged with VLAN 0 and any priority
- `0x0000/0xefff`: Match frames with no 802.1Q header or tagged with VLAN 0 and priority 0

NOTE: Frames that ingress on an access ports are not considered part of the assigned vlan yet. Local VLAN interfaces are access ports, e.g. br0.vlan7.

Type	DisplayString
Range	1 - 13
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.6.3.1.15

cfgFwL2MnglInputInterface

The Ingress Interface On Which The Rule Matches

The name of the interface on which a frame ingresses.

Set to `any`, to match any interface.

Examples:

- **any**
- **wlan0**
- **eth0**

Type	DisplayString
Range	1 - 15
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.6.3.1.16

cfgFwL2MnglValue

The Value to be set by the Action

Depending on the setting of `cfgFwL2MnglAction` the accepted format of this field is different.

- **setdscp(1)**: The DSCP in hexadecimal form 00-FF without 0x. Must be a multiple of 4 (the lowest 2 bits 0)

Type	DisplayString
Range	1 - 2
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.6.3.1.17

cfgFwL2MnglEnabled

Disabled or Enabled this Rule

Only enabled rules are installed on mangling bridges.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.6.3.1.2

cfgFwL2MnglBridge

Bridge on Which the Rule is Created

This rule is only created on the bridge specified here.

Range	0 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.6.3.1.3

cfgFwL2MnglAction

Action to Perform

- **none(0)**: No action, implicitly disabled the rule
- **setdscp(1)**: Set the 6-bit DSCP field of IP frames

Enumeration	none (0), setdscp (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.6.3.1.4

cfgFwL2MnglPriority

Mangle Priority

When multiple rules match, the rule with the highest priority is applied.

Multiple overlapping rules with the same priority have undefined behaviour, thus are not allowed.

Range	1 - 65535
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.6.3.1.5

cfgFwL2MnglSource

Source Network/IP On Which The Rule Matches (CIDR Notation)

This field only has an effect when `cfgFwL2MnglEthertype` is set to `0800`, `0806` or `8035`.

When `cfgFwL2MnglEthertype` is set to `0800` this is an IP or network.

When `cfgFwL2MnglEthertype` is set to `0806` or `8035` it is the source address or a range of source addresses of an ARP/RARP frame (ARP_SPA).

Type	DisplayString
Range	7 - 18
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.6.3.1.6

cfgFwL2MnglDestination

Destination Network/IP On Which The Rule Matches (CIDR Notation)

This field only has an effect when `cfgFwL2MnglEthertype` is set to `0800`, `0806` or `8035`.

When `cfgFwL2MnglEthertype` is set to `0800` this is an IP or network.

When `cfgFwL2MnglEthertype` is set to `0806` or `8035` it is the target address or a range of target addresses of an ARP/RARP frame (ARP_TPA).

Type	DisplayString
Range	7 - 18
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.6.3.1.7

cfgFwL2MnglComment

User Comment

This parameter has no operational function. It allows to store a comment about the use of this l2 mangle rule.

Type	DisplayString
Range	0 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.6.3.1.8

cfgFwL2MnglProtocol

IP Protocol On Which The Rule Matches

Set to `-1`, to not use the protocol to match.

When `cfgFwL2MnglEthertype` is set to `0800`, this field may be used to match a specific IP protocol.

Examples:

- **-1**: any protocol
- **1**: ICMP
- **2**: IGMP
- **6**: TCP
- **17**: UDP
- **50**: ESP (IPsec)
- **51**: AH (IPsec)
- **112**: VRRP / CARP

For a full list of available protocols see https://en.wikipedia.org/wiki/List_of_IP_protocol_numbers

When `cfgFwL2MnglEthertype` is set to `0806` or `8035`, this field may be used to match the OP code of an ARP frame.

Examples:

- **-1**: any OP code
- **1**: request
- **2**: response
- **3**: request reverse
- **4**: response reverse

Range	-1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.6.3.1.9

12 WESTERMO-SW6-GNSS-MIB

12.1 gnss

12.1.1 configuration

12.1.1.1 cfgGnss

12.1.1.1.1 cfgGnssGpsd

12.1.1.1.1.1 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 802.11be and cellular products only.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.10.1.1.1.1

12.1.1.1.1.2 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 `0.0.0.0:2947`.

Examples:

- **0.0.0.0:2947**

- **127.0.0.1:1234**

Applies to 802.11be and cellular products only.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.10.1.1.1.2

12.1.1.1.2 cfgGnssDevice

12.1.1.1.2.1 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
- **gpsGloBei(7)**: GPS, 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 Galileo
- **gloBeiGal(14)**: Glonass, BeiDou and Galileo
- **gpsGloBeiGal(15)**: GPS, Glonass, BeiDou and Galileo

This entry can also be interpreted as bitmask, with each bit reflecting a satellite system as follows:

- Bit 1 (0x1): GPS
- Bit 2 (0x2): Glonass
- Bit 3 (0x4): BeiDou
- Bit 4 (0x8): Galileo

Additional position information may be received from QZSS (Quasi Zenith Satellite System) in Japan and SBAS (WAAS - North America), EGNOS - Europe, MSAS - Japan, GAGAN - India). These

position augmentation is always active and can not be disabled.

Depending on the product in use, different combinations of the enumeration above are valid.

Cellular 4G products:

- Valid single constellations: 1, 2, 4, 8
- Valid dual constellations: 3, 5, 9, 10, 12
- Valid tripel constellations: 11, 13
- Invalid: 6, 7, 14, 15

Cellular 5G products:

- All combinations are valid

802.11be products:

- Valid single constellations: 1
- Valid dual constellations: 3, 5, 9
- Valid quadrupel constellations: 15
- Invalid: 2, 4, 6, 7, 8, 10, 11, 12, 13, 14

Applies to 802.11be and cellular products only

Enumeration	gps (1), glonass (2), gpsGlo (3), beidou (4), gpsBei (5), gloBei (6), gpsGloBei (7), galileo (8), gpsGal (9), gloGal (10), gpsGloGal (11), beiGal (12), gpsBeiGal (13), gloBeiGal (14), gpsGloBeiGal (15)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.10.1.1.2.1

12.1.1.1.2.2 cfgGnssDevMeasurementPeriod

Measurement Period

The measurement period *T* defines the time interval in milliseconds between two consecutive localisations. The minimal time is limited to 40 ms (25 Hz). Please refer to the User Guide for the maximum measuring rate supported by the device.

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 802.11be and cellular products only.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.10.1.1.2.2

12.1.1.1.2.3 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.

Range	0 - 12
OID	1.3.6.1.4.1.16177.1.400.2.10.1.1.2.3.1

cfgGnssDevMsgsNmeaTableEntry

NMEA Sentence

Applies to cellular products only.

OID	1.3.6.1.4.1.16177.1.400.2.10.1.1.2.3.1.1

cfgGnssDevMsgsNmeaTableIndex

Table Entry Index

Range	0 - 12
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.2.10.1.1.2.3.1.1.1

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.

Type	Integer32
Access	readwrite
OID	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* multiples 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.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.10.1.1.2.3.1.1.3

cfgGnssDevMsgsNmeaEnabled

Enable or Disable NMEA Sentences

Applies to cellular products only.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.10.1.1.2.3.1.1.4

cfgGnssDevMsgsUbxTable

UBX Messages

UBX is a proprietary protocol from the company u-blox for exchanging GNSS data.

Applies to cellular products only.

Range	0 - 127
OID	1.3.6.1.4.1.16177.1.400.2.10.1.1.2.3.2

cfgGnssDevMsgsUbxTableEntry

UBX Message

Applies to cellular products only.

OID	1.3.6.1.4.1.16177.1.400.2.10.1.1.2.3.2.1

cfgGnssDevMsgsUbxTableIndex

Table Entry Index

Range	0 - 127
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.2.10.1.1.2.3.2.1.1

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.

Range	0 - 65535
Access	readwrite
OID	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* multiples 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.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.10.1.1.2.3.2.1.3

cfgGnssDevMsgsUbxEnabled

Enable or Disable UBX Messages

Applies to cellular products only.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.10.1.1.2.3.2.1.4

12.1.1.1.2.4 cfgGnssDevDebugEnabled

Disable or Enable Debug Messages

Applies to cellular products only.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.10.1.1.2.4

12.1.2 software

12.1.2.1 swGnss

12.1.2.1.1 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.

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.2.10.2.1.1

12.1.2.1.2 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.

Enumeration	na (0), none (1), twoD (2), threeD (3)
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.2.10.2.1.2

12.1.2.1.3 swGnssLon

GNSS Longitude

Longitude of the current location in decimal degrees (DD).

Example:

- 8.825446333

Applies to cellular products only.

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.2.10.2.1.3

12.1.2.1.4 swGnssLat

GNSS Latitude

Latitude of the current location in decimal degrees (DD).

- 47.268953167

Applies to cellular products only.

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.2.10.2.1.4

12.1.2.1.5 swGnssAlt

GNSS Altitude

Altitude of the current location in meters.

Applies to cellular products only.

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.2.10.2.1.5

13 WESTERMO-SW6-ICL-MIB

13.1 icl

13.1.1 configuration

13.1.1.1 cfgIcl

13.1.1.1.1 cfgIclEnabled

Enable Inter-Carriage Link application.

Applies to AP. 802.11n and 802.11ax products only.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.5.1.1.1

13.1.1.1.2 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 and 802.11ax products only.

Range	0 - 600
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.5.1.1.2

13.1.1.1.3 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 and 802.11ax products only.

Range	-99 - 99
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.5.1.1.3

13.1.1.1.4 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 and 802.11ax products only.

Range	1 - 600
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.5.1.1.4

13.1.1.1.5 cfgIclDisconnectionThreshold

This value defines the minimum signal level necessary for a ICL pair to stay connected. If the singal 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 and 802.11ax products only.

Range	-99 - 99
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.5.1.1.5

13.1.1.1.6 cfgIclInterfaceName

This value describes the interface the ICL Application will use for its services.

Applies to AP. 802.11n and 802.11ax products only.

Type	DisplayString
Range	1 - 15
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.5.1.1.6

13.1.1.1.7 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 and 802.11ax products only.

Range	2 - 60
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.5.1.1.7

13.1.1.1.8 cfgIclBlacklistTime

Duration of blacklisting in seconds.

Applies to AP. 802.11n and 802.11ax products only.

Range	1 - 3600
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.5.1.1.8

13.1.1.1.9 cfgIclSuspended

Initial state of ICL when it starts up.

Applies to AP. 802.11n and 802.11ax products only.

Enumeration	resumed (0), suspended (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.5.1.1.9

13.1.2 rpc

13.1.2.1 rpcIcl

13.1.2.1.1 rpcIclForceDisconnect

Force the device to disconnect from the current ICL partner and resume background scanning for a new partner.

Enumeration	nop (0), disconnect (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.5.3.1.1

13.1.2.1.2 rpcIclClearBlacklist

Clear all currently blacklisted entries.

Enumeration	nop (0), clear (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.5.3.1.2

13.1.3 settings

13.1.3.1 setIcl

13.1.3.1.1 setIclSuspended

Suspend or resume ICL operation.

When suspended, ICL brings down the wireless interface and pauses operation. It remains silent until it is resumed.

Enumeration	resumed (0), suspended (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.5.4.1.1

13.1.4 software

13.1.4.1 swIcl

13.1.4.1.1 swIclStatus

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.

Enumeration	disabled (0), scanning (1), connected (2), suspended (3)
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.2.5.6.1.1

14 WESTERMO-SW6-NWM-MIB

14.1 nwm

14.1.1 configuration

14.1.1.1 cfgHttpReport

14.1.1.1.1 cfgHttpRprtServerUrl

URL Of The Remote Server Where AFM, AFC And IDF Reports Are Sent To

Format: `http://<ipv4>[:<port>][/<path>]`

`<port>` is optional and defaults to 80 when not specified. `<path>` is optional and defaults to / when not specified.

Applies to AP. 802.11n products only.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.3.1.1.1

14.1.1.2 cfgChannelManager

14.1.1.2.1 cfgChMgrUsableFrequencyList

Usable Frequency List

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.

Range	-1 - 23
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.3.1.3.2

14.1.1.2.2 cfgChMgrDfsUseNvram

Use Non-volatile Memory for Channel Manager

When 'enabled' the DFS states will be load / stored from / to the non-volatile ram device.

Applies to AP. 802.11n products only.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.3.1.3.3

14.1.1.3 cfgNwm

14.1.1.3.1 cfgNwmEnabled

Enable the Wireless Manager

Applies to AP. 802.11n products only.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.3.1.4.1

14.1.1.4 cfgIdf

14.1.1.4.1 cfgIdfEnabled

Enable Interference Detection Function

IDF is only supported for devices with two radios.

Applies to AP. 802.11n products only.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.3.1.5.1

14.1.1.4.2 cfgIdfScanWorkTable

Table of Scan Work Items

Applies to AP. 802.11n products only.

Range	0 - 31
OID	1.3.6.1.4.1.16177.1.400.2.3.1.5.10

14.1.1.4.2.1 cfgIdfScanWorkTableEntry

Scan Work List

Applies to AP. 802.11n products only.

OID	1.3.6.1.4.1.16177.1.400.2.3.1.5.10.1

cfgIdfScanWorkIndex

Entry Index of Table

Applies to AP. 802.11n products only.

Range	0 - 31
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.2.3.1.5.10.1.1

cfgIdfScanWorkFreq

Center Frequency in MHz of the Channel to Scan

Set the center frequency to `0` to disable this scan work item and all following items.

Applies to AP. 802.11n products only.

Range	0 - 6100
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.3.1.5.10.1.2

cfgIdfScanWorkAction

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 IS collected from Antenna port 3 and spectral statistics are 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 is collected from Antenna port 3 and wifi statistics are generated at the end of the scan work item.

Applies to AP. 802.11n products only.

Enumeration	none (0), spectral (2), radar (3), wifi (4)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.3.1.5.10.1.3

cfgIdfScanWorkSeconds

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.

Range	0 - 86400
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.3.1.5.10.1.4

14.1.1.4.3 cfgIdfInterval

Reporting Interval in Seconds

This value defines the interval when all available Reports are sent to the URL defined in cfgIdfHttpReportUrl.

Applies to AP. 802.11n products only.

Range	1 - 86400
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.3.1.5.3

14.1.1.4.4 cfgIdfName

IDF Name

Can be used to set an unique identifier for the reports.

Applies to AP. 802.11n products only.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.3.1.5.4

14.1.1.4.5 cfgIdfTrigger

14.1.1.4.5.1 cfgIdfTrigRadarCntTh

IDF Trigger radar_count

Defines the number of radar events to trigger a report.

Applies to AP. 802.11n products only.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.3.1.5.5.1

14.1.1.4.5.2 cfgIdfTrigChanLoadTh

IDF Trigger channel_load

Defines the channel load ratio in percent to trigger a report.

Applies to AP. 802.11n products only.

Range	0 - 100
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.3.1.5.5.2

14.1.1.4.5.3 cfgIdfTrigAlienLoadTh

IDF Trigger alien_load

Defines the ratio of alien channel load in percent to trigger a report.

Applies to AP. 802.11n products only.

Range	0 - 100
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.3.1.5.5.3

14.1.1.4.5.4 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.

Range	0 - 100
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.3.1.5.5.4

14.1.1.4.5.5 cfgIdfTrigAlienMaxRssiTh

IDF Trigger alien_avg_rssi

Defines the average RSSI of alien traffic to trigger a report.

Applies to AP. 802.11n products only.

Range	0 - 127
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.3.1.5.5.5

14.1.1.4.5.6 cfgIdfTrigDomMaxRssiTh

IDF Trigger domestic_avg_rssi

Defines the average RSSI of domestic traffic to trigger a report.

Applies to AP. 802.11n products only.

Range	0 - 127
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.3.1.5.5.6

14.1.1.5 cfgChannelCleaner

14.1.1.5.1 cfgChanCleanUsableFrequencyList

Usable Frequency List

Configure which frequency list cfgWlanFIndex is to be used as 'list of usable frequencies' by Channel-Cleaner.

Applies to AP. 802.11n products only.

Range	-1 - 23
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.3.1.6.10

14.1.1.5.2 cfgChanCleanDfsUseNvram

Use Non-volatile Memory for Channel Cleaner

When set to enabled(1) the DFS states will be load/stored from/to the non-volatile ram device.

Applies to AP. 802.11n products only.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.3.1.6.9

14.1.1.6 cfgAfm

14.1.1.6.1 cfgAfmEnabled

Enable the Area Frequency Manager on the Local Device

AFM and AFC in the same area require L3 communication.

The ports used are:

- UDP: 4711
- UDP: 4713

Applies to AP. 802.11n products only.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.3.1.7.1

14.1.1.6.2 cfgAfmRedundant

14.1.1.6.2.1 cfgAfmRedundantIp

IP Address of the Redundant Area Frequency Manager

Set to `0.0.0.0` to disable.

Applies to AP. 802.11n products only.

Type	IpAddress
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.3.1.7.10.1

14.1.1.6.2.2 cfgAfmRedundantName

Name of the Redundant Area Frequency Manager

Applies to AP. 802.11n products only.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.3.1.7.10.2

14.1.1.6.3 cfgAfmName

Name of the Area Frequency Manager on the Local Device

Applies to AP. 802.11n products only.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.3.1.7.2

14.1.1.6.4 cfgAfmNeighbourTable

A Table of the Two Neighbouring (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.

Range	0 - 4
OID	1.3.6.1.4.1.16177.1.400.2.3.1.7.20

14.1.1.6.4.1 cfgAfmNeighbourTableEntry

Neighbour AFM List

Applies to AP. 802.11n products only.

OID	1.3.6.1.4.1.16177.1.400.2.3.1.7.20.1

cfgAfmNeighbourTableIndex

Entry Index of Table

Applies to AP. 802.11n products only.

Range	0 - 4
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.2.3.1.7.20.1.1

cfgAfmNeighbourIp

IP Address of the Neighbour (Adjacent) Area Frequency Manager

Applies to AP. 802.11n products only.

Type	IpAddress
Access	readwrite
OID	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.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.3.1.7.20.1.3

14.1.1.6.5 cfgAfmIndex

Area Index

The index of the Area that the locally running Area Frequency Manager manages.

Primary and secondary AFM in the same area must share the same index.

Applies to AP. 802.11n products only.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.3.1.7.3

14.1.1.6.6 cfgAfmAfcTable

AFC Table

A table of all Area Frequency Clients controlled by the Area Frequency Manager running on this device.

Applies to AP. 802.11n products only.

Range	0 - 63
OID	1.3.6.1.4.1.16177.1.400.2.3.1.7.30

14.1.1.6.6.1 cfgAfmAfcTableEntry

AFC Table Entry

Applies to AP. 802.11n products only.

OID	1.3.6.1.4.1.16177.1.400.2.3.1.7.30.1

cfgAfmAfcTableIndex

Entry Index of Table

Applies to AP. 802.11n products only.

Range	0 - 63
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.2.3.1.7.30.1.1

cfgAfmAfcName

Name of the Area Frequency Client to Control

Applies to AP. 802.11n products only.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	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.

Type	IpAddress
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.3.1.7.30.1.3

14.1.1.6.7 cfgAfmAreaSize

Number of Segments Controlled by this AFM

Applies to AP. 802.11n products only.

Range	1 - 32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.3.1.7.5

14.1.1.6.8 cfgAfmPrimary

Enable to set this AFM as Primary in this Area

Set to disabled for stand-by mode.

Applies to AP. 802.11n products only.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.3.1.7.6

14.1.1.6.9 cfgAfmReportEnabled

Enable AFM Reporting

When enabled(1) the AFM 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.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.3.1.7.7

14.1.1.6.10 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.

Range	1 - 1024
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.3.1.7.8

14.1.1.7 cfgAfc

14.1.1.7.1 cfgAfcEnabled

Enable the Area Frequency Client on the Local Device

AFC operation requires neighbour reporting, ensure `cfgWlanIfaceNeighbourReport` is enabled.

AFM and AFC in the same area require L3 communication.

The ports used are:

- UDP: 4711
- UDP: 4713

Applies to AP. 802.11n products only.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.3.1.8.1

14.1.1.7.2 cfgAfcAfmTable

AFC AFM Table

A table of Area Frequency Managers which are allowed to control this Area Frequency Client (AFC).

Applies to AP. 802.11n products only.

Range	0 - 4
OID	1.3.6.1.4.1.16177.1.400.2.3.1.8.10

14.1.1.7.2.1 cfgAfcAfmTableEntry

AFC AFM Table

Applies to AP. 802.11n products only.

OID	1.3.6.1.4.1.16177.1.400.2.3.1.8.10.1

cfgAfcAfmTableIndex

Entry Index of Table

Applies to AP. 802.11n products only.

Range	0 - 4
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.2.3.1.8.10.1.1

cfgAfcAfmName

Name of the Connecting Area Frequency Manager

Applies to AP. 802.11n products only.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.3.1.8.10.1.2

cfgAfcAfmIp

IP Address of the Connecting Area Frequency Manager

Applies to AP. 802.11n products only.

Type	IpAddress
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.3.1.8.10.1.3

14.1.1.7.3 cfgAfcName

Name of the Area Frequency Client on the Local Device

Applies to AP. 802.11n products only.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.3.1.8.2

14.1.1.7.4 cfgAfcIndex

Index of the Area Frequency Client within the Local Area

Applies to AP. 802.11n products only.

Range	0 - 31
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.3.1.8.3

14.1.1.7.5 cfgAfcNeighbourOffsetTable

Table of up to 24 Segment Offsets Relevant for Neighbour List

Applies to AP. 802.11n products only.

Range	0 - 23
OID	1.3.6.1.4.1.16177.1.400.2.3.1.8.40

14.1.1.7.5.1 cfgAfcNeighbourOffsetTableEntry

Neighbour Segment Offset Table

Applies to AP. 802.11n products only.

OID	1.3.6.1.4.1.16177.1.400.2.3.1.8.40.1

cfgAfcNeighbourOffsetTableIndex

Entry Index of Table

Applies to AP. 802.11n products only.

Range	0 - 23
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.2.3.1.8.40.1.1

cfgAfcNeighbourOffset

Segment Offset to be Considered for Neighbour List

Applies to AP. 802.11n products only.

Range	-15 - 15
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.3.1.8.40.1.2

14.1.1.7.6 cfgAfcBackupFreq

Backup Frequency

The frequency that the Area Frequency Client will use if the Area Frequency Manager is not present at startup. Also this frequency will be used when no frequencies are available for use, e.g when all configured frequencies are blocked by DFS.

This frequency must be a non-DFS frequency.

Applies to AP. 802.11n products only.

Range	0 - 6100
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.3.1.8.5

14.1.1.7.7 cfgAfcReportEnabled

Disable or Enable AFC Reporting

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.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.3.1.8.6

14.1.2 rpc

14.1.2.1 rpcChannelManager

14.1.2.1.1 rpcChMgrHttpReport

Requests a HTTP Report from the Channel Manager

Applies to AP. 802.11n products only.

Enumeration	nop (0), freqstate (1), channels (2)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.3.3.1.1

14.1.2.2 rpcNwm

14.1.2.2.1 rpcNwmHttpReport

Requests a HTTP Report from the Wireless Manager

Applies to AP. 802.11n products only.

Enumeration	nop (0), status (1), freqstate (2)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.3.3.2.1

14.1.2.3 rpcNvram

14.1.2.3.1 rpcNvramFreqStatesReset

Nvram Frequency States Reset

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.

Enumeration	reset (0)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.3.3.3.1

15 WESTERMO-SW6-PWN-MIB

15.1 pwn

15.1.1 configuration

15.1.1.1 cfgWireless

15.1.1.1.1 cfgWlanBandsteering

15.1.1.1.1.1 cfgWlanBsteerEnabled

****OBSOLETE:** Enable Band Steering**

This parameter is obsolete and has been replaced by `cfgWlanCsteerEnabled`.

Enumeration	disabled (0), enabled (1)
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.2.9.1.1.1.1

15.1.1.1.1.2 cfgWlanBsteerMatchingSsid

****OBSOLETE:** List of Matching SSIDs**

This parameter is obsolete and has been replaced by `cfgWlanBsteerMatchingWlan`.

Type	DisplayString
Range	1 - 255
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.2.9.1.1.1.2

15.1.1.1.1.3 cfgWlanBsteerMatchingWlan

****OBSOLETE:** WLAN Interface Indices for Band Steering**

This parameter is obsolete and has been replaced by `cfgWlanCsteerMatchingWlan`.

Type	DisplayString
Range	1 - 255
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.2.9.1.1.1.3

15.1.1.1.2 cfgWlanClientSteering

15.1.1.1.2.1 cfgWlanClientSteeringTable

Client Steering Table

Client steering is a technique used to encourage STAs to use a less-congested AP.

Applies to AP. 802.11ax and 802.11be products only.

Range	0 - 0
OID	1.3.6.1.4.1.16177.1.400.2.9.1.1.2.1

cfgWlanClientSteeringTableEntry

Client Steering Table Entry

Client steering is a technique used to encourage STAs to use a less-congested AP.

Applies to AP. 802.11ax and 802.11be products only.

OID	1.3.6.1.4.1.16177.1.400.2.9.1.1.2.1.1

cfgWlanCsteerIndex

Table Entry Index

Applies to AP. 802.11ax and 802.11be products only.

Range	0 - 0
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.2.9.1.1.2.1.1.1

cfgWlanCsteerLoadBalNrStaDiff

Load Balancing STA Difference

Minimal difference of connected STAs between an other AP before load balancing policy is active.

Applies to AP. 802.11ax and 802.11be products only.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.9.1.1.2.1.1.13

cfgWlanCsteerBandSteeringNrStaDiff

Band Steering STA Difference

Minimal difference of connected STAs between bands before band steering policy is active.

Applies to AP. 802.11ax and 802.11be products only.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.9.1.1.2.1.1.14

cfgWlanCsteerAssocSteeringEnabled

Disable or Enable Association Steering

Allow rejecting association requests for steering purposes, see also `cfgWlanCsteerConnectLevel`.

Applies to AP. 802.11ax and 802.11be products only.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.9.1.1.2.1.1.16

cfgWlanCsteerConnectLevel

Connect Level

Minimum signal level (dBm) to allow connections. The AP will not response to probe requests of STAs that are below this level. If `cfgWlanCsteerAssocSteeringEnabled` is enabled, it will not answer association requests either.

If this parameter is set to `0`, all STAs are allowed to connect regardless of their signal level.

Applies to AP. 802.11ax and 802.11be products only.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.9.1.1.2.1.1.17

cfgWlanCsteerRemainLevel

Remain Level

Minimum signal level (dBm) to remain connected.

Applies to AP. 802.11ax and 802.11be products only.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.9.1.1.2.1.1.18

cfgWlanCsteerRoamScanLevel

Roaming Scan Level

Minimum signal level (dBm) before the AP requests a Beacon Report (802.11k) from the STA.

Applies to AP. 802.11ax and 802.11be products only.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.9.1.1.2.1.1.19

cfgWlanCsteerEnabled

Disable or Enable Client Steering

Client Steering makes use of the Radio Resource (802.11k) and the BSS Transition Management (802.11v) standards in order to steer clients (STAs) among various Basic Service Sets (BSS) to achieve an overall improved wireless performance for all participating STAs in the same Extended Service Set (ESS).

Applies to AP. 802.11ax and 802.11be products only.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.9.1.1.2.1.1.2

cfgWlanCsteerRoamTriggerLevel

Roaming Trigger Level

Minimum signal level (dBm) before the AP sends a BSS Transition Management Request (802.11v) to the STA.

Applies to AP. 802.11ax and 802.11be products only.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.9.1.1.2.1.1.22

cfgWlanCsteerRoamKickTimeout

Roaming Kick Timeout

Timeout (in millisecons) before the AP requests the STA to dissasociate STA after an unsuccessful BSS Transition Management Request.

Applies to AP. 802.11ax and 802.11be products only.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.9.1.1.2.1.1.24

cfgWlanCsteerSignalLevelDiff

Signal Level Difference

Minimum signal level difference (in dB) before any steering policy of the AP becomes active.

Applies to AP. 802.11ax and 802.11be products only.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.9.1.1.2.1.1.25

cfgWlanCsteerInitConnectTimeout

Initial Connection Timeout

Initial timeout (in millisecons) before responding to probe requests. This forces the STA to search for other APs and allows the other APs to see packets as well.

Applies to AP. 802.11ax and 802.11be products only.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.9.1.1.2.1.1.26

cfgWlanCsteerLoadBalEnabled

Disable or Enable Load-based Steering

Enable kicking of STAs on excessive channel load when the following conditions are met:

- The channel load of the current AP must be equal or higher than the defined value in `cfgWlanCsteerLoadBalMinLoad`.
- At least the number of STAs defined in `cfgWlanCsteerLoadBalMinNrStas` must be associated to the AP.
- The difference of client connections between the current and another AP must be at least the number defined in `cfgWlanCsteerLoadBalNrStaDiff`.

Applies to AP. 802.11ax and 802.11be products only.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.9.1.1.2.1.1.27

cfgWlanCsteerLoadBalMinLoad

Minimal Load Before Steering

Minimum load in percent before STAs are disassociated based on channel load.

Applies to AP. 802.11ax and 802.11be products only.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.9.1.1.2.1.1.28

cfgWlanCsteerLoadBalMinNrStas

Minimum STAs for Load-based Kick

Minimum number of connected STAs before disassociating them based on channel load.

Applies to AP. 802.11ax and 802.11be products only.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.9.1.1.2.1.1.30

cfgWlanCsteerScanBand2GhzOpClass

Operating Class 2GHz for Beacon Reports

Operating class used for Beacon Report requests (802.11k) for 2GHz band. It defines the channel range to scan which is also depending on the used country code.

Operating classes are defined in ANNEX E of IEEE 802.11 standard.

Applies to AP. 802.11ax and 802.11be products only.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.9.1.1.2.1.1.34

cfgWlanCsteerScanBand5GhzOpClass

Operating Class 5GHz for Beacon Reports

Operating class used for Beacon Report requests (802.11k) for 5GHz band. It defines the channel range to scan which is also depending on the used country code.

Operating classes are defined in ANNEX E of IEEE 802.11 standard.

Applies to AP. 802.11ax and 802.11be products only.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.9.1.1.2.1.1.35

cfgWlanCsteerMatchingWlan

WLAN Interface Indices for Client Steering

Multiple indices in `cfgWlanInterfaceTable` may be referenced as a space and/or comma separated list.

All referenced wireless interfaces that are involved in the Client Steering process.

Examples:

- 0
- 1, 3
- 1,2 3,4

Applies to AP. 802.11ax and 802.11be products only.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.9.1.1.2.1.1.4

cfgWlanCsteerInterfaces

Network Interfaces

The network interfaces to search for other Client Steering instances. The Client Steering process uses a Data Link (L2) protocol to look for other instances.

One or more network interfaces can specifed in a space and/or comma separated list.

Example:

- br0.vlan0
- br0.vlan100, br0.vlan101, br0.vlan102

Applies to AP. 802.11ax and 802.11be products only.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.9.1.1.2.1.1.5

cfgWlanCsteerDebugLevel

Minimum Level of Logged Messages

Applies to AP. 802.11ax and 802.11be products only.

Enumeration	fatal (0), info (1), verbose (2), debug (3), network (4), full (5)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.9.1.1.2.1.1.7

15.1.1.1.3 cfgWlanPasspoint

15.1.1.1.3.1 cfgWlanPasspointTable

Passpoint Table

TODO: description Applies to AP. 802.11ax and 802.11be products only.

Range	0 - 63
OID	1.3.6.1.4.1.16177.1.400.2.9.1.1.3.1

cfgWlanPasspointTableEntry

Passpoint Table Entry

TODO: description Applies to AP. 802.11ax and 802.11be products only.

OID	1.3.6.1.4.1.16177.1.400.2.9.1.1.3.1.1

cfgWlanPasspointIndex

Table Entry Index

Applies to AP. 802.11ax and 802.11be products only.

Range	0 - 63
Access	noaccess
OID	1.3.6.1.4.1.16177.1.400.2.9.1.1.3.1.1.1

cfgWlanPasspointNetworkAuthType

Network Authentication Type

This parameter indicates what type of authentication is used in the network.

- 00: Acceptance of terms and conditions
- 01: On-line enrollment supported
- 02: http/https redirection
- 03: DNS redirection

Example for https redirection

- 02https://www.example.com/redirect/me/here/

Applies to AP. 802.11ax and 802.11be products only.

Type	DisplayString
Range	2 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.9.1.1.3.1.1.10

cfgWlanPasspointDomain

Domain Name

A comma separated list of domain names of the hotspot operator.

Applies to AP. 802.11ax and 802.11be products only.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.9.1.1.3.1.1.11

cfgWlanPasspointRoamingConsortium

Roaming Consortium

A comma and/or space separated list of supported roaming consortiums.

OpenRoaming ROCIs

- 004096: Legacy
- 5A03BA0000: Open
- 5A03BA0001: Service Provider
- 5A03BA0010: Cloud Provider
- 5A03BA0011: Generic Enterprise
- 5A03BA0100: Governmnent (City)
- 5A03BA0101: Automotive
- 5A03BA0110: Hospitality
- 5A03BA0111: Aviation Industry
- 5A03BA1000: Education or Research
- 5A03BA1001: Cable Industry
- 5A03BA1010: Manufacturer
- 5A03BA1011: Retail

Applies to AP. 802.11ax and 802.11be products only.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.9.1.1.3.1.1.12

cfgWlanPasspointCellNet

3GPP Cellular Network Information

A comma separated list of allowed cellular network providers. The coding format is: `<MCC1,MNC1>[;<MCC2,MNC2>`

Example

- 240,01
- 240,01;240,02;240,03
- 240,01;242,02

The parameter value `-1` disables the field.

Applies to AP. 802.11ax and 802.11be products only.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.9.1.1.3.1.1.13

cfgWlanPasspointNaiRealm

NAI Realm Information

One or more realm can be advertised (currently limited to a single realm). These parameters provide information for stations using Interworking network selection to allow automatic connection to a network based on credentials.

The coding format is:

`<encoding>,<NAI Realm(s)>[,<EAP Method 1>][,<EAP Method 2>][,...]`

The parameter value `-1` disables the field.

Multiple NAI realms can be defined as semicolon `;` separeted list.

Applies to AP. 802.11ax and 802.11be products only.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.9.1.1.3.1.1.14

cfgWlanPasspointOperatorNamespace

Operator Namespace Identifier

The operator namespace identifier adds the RADIUS attribute `126` in the following coding format: `<identifier><operator>`

Identifier

- 0: TADIG
- 1: REALM
- 2: E212
- 3: ICC
- 4: WBAID

Example

- 0OperatorNameForTadig
- 4WESTERMO:SE

The parameter value `-1` disables the field.

Applies to AP. 802.11ax and 802.11be products only.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.9.1.1.3.1.1.15

cfgWlanPasspointEnabled

Disable or Enable Passpoint

Applies to AP. 802.11ax and 802.11be products only.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.9.1.1.3.1.1.2

cfgWlanPasspointSsid

SSID of the Wireless Interface

The SSID must match with the SSID in `cfgWlanIfaceSsid` of the wireless interface.

Applies to AP. 802.11ax and 802.11be products only.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.9.1.1.3.1.1.3

cfgWlanPasspointOperator

Operator's Friendly Name

The operator name is normally not visible for the user but may assist during manual hotspot selection.

The field has a language prefix, be default it is set to `eng:` for English.

Multiple names can be defined as comma and/or space separeted list.

Applies to AP. 802.11ax and 802.11be products only.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.9.1.1.3.1.1.4

cfgWlanPasspointVenueName

Venue Name

The venue name is the description of the location. It might be a building name or an address.

The field has a language prefix, be default it is set to `eng:` for English.

Multiple venues can be defined as comma and/or space separeted list.

Applies to AP. 802.11ax and 802.11be products only.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.9.1.1.3.1.1.5

cfgWlanPasspointVenueGroup

Venue Group

Optionally field to describe the passpoint group. Please refer to the IEEE 802.11-2016 standard for more information.

- 0: Unspecified
- 1: Assembly
- 2: Business
- 3: Educational
- 4: Factory and Industrial
- 5: Institutional
- 6: Mercantile
- 7: Residential
- 8: Storage
- 9: Utility and Misc
- 10: Vehicular
- 11: Outdoor

Applies to AP. 802.11ax and 802.11be products only.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.9.1.1.3.1.1.6

cfgWlanPasspointVenueType

Venue Type

Optionally field to describe the passpoint type. The type depends on the venue group value. Please refer to the IEEE 802.11-2016 standard for more information.

- 0: Unspecified
- x: Depends on Venue Group

Applies to AP. 802.11ax and 802.11be products only.

Type	Integer32
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.9.1.1.3.1.1.7

cfgWlanPasspointHessid

Homogeneous Extended Service Set Identifier (HESSID)

If two APs have different SSIDs, they are considered to be different wireless networks. If two APs have the same SSID, they are considered to be part of the same wireless network. But because SSIDs are not globally administered, it is possible that two APs with the same SSID are, in different

wireless networks. The HESSID allows mobile devices to detect this condition. If two access points (APs) have the same SSID but are on different wireless networks, the two networks will have different HESSIDs.

Example

- 02:14:5a:11:22:33

The parameter value `00:00:00:00:00:00` disables this parameter.

Applies to AP. 802.11ax and 802.11be products only.

Type	DisplayString
Range	17 - 17
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.9.1.1.3.1.1.8

cfgWlanPasspointNetworkType

Access Network Type

The access network type is indicative of the privacy or cost of the given hotspot.

- 0: Private Network
- 1: Private Network with Guest Access
- 2: Chargeable Public Network
- 3: Free Public Network
- 4: Personal Device Network
- 5: Emergency Services Only Network
- 6: Test or Experimental Network
- 7: Wildcard Network

Applies to AP. 802.11ax and 802.11be products only.

Enumeration	private (0), privateguest (1), publicchargeable (2), public (3), personal (4), emergency (5), test (14), wildcard (15)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.9.1.1.3.1.1.9

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": "admin",
    "password": "10SoaL_98"
  }
}
```

Example response

```
{
  "id": 1, "jsonrpc": "2.0",
  "result": { "token": "4855d491f8dbde3628edc4e5be4861fe" }
}
```

16.1.1.2 Session

- method: `session()`

Example POST request:

```
{
  "id": 1, "jsonrpc": "2.0",
  "method": "session"
}
```

Example response

```
{
  "id": 1, "jsonrpc": "2.0",
  "result": {
        "username": "admin",
        "role": "admin",
        "provider": "local",
    }
}
```

16.1.1.3 Logout

- method: `logout()`

Example POST request:

```
{
  "id": 1, "jsonrpc": "2.0",
  "method": "logout"
}
```

Example response

```
{
  "id": 1, "jsonrpc": "2.0",
  "result": { "message": "Logout successful"}
}
```

16.1.1.4 Change Password

- method: `change_password(username, old_password, new_password)`

Example POST request

```
{
  "id": 1, "jsonrpc": "2.0",
  "method": "change_password",
  "params": {
    "username": "admin",
    "old_password": "10SoaL_98",
    "new_password": "10SoNewaL_77"
  }
}
```

Example response

```
{
  "id": 1, "jsonrpc": "2.0",
  "result": {"message": "Password change successful"}
}
```

16.1.1.5 List active sessions

- method: `get_session_list`

Example GET request

```
{
  "id": 1, "jsonrpc": "2.0",
  "method": "get_session_list"
}
```

Example response

```
{
  "id":1, "jsonrpc":"2.0",
  "result": [
    {"username": "admin", "role": "admin", "id": "779
       cff87743d018001bfab1310047f13", "auth": "local", "interface": "cli"},
    {"username": "admin", "role": "admin", "id": "
       a3c0c8f9279250ef06fa95a148b6f56a", "auth": "local", "interface":"Web
       Server"}
  ]
}
```

16.1.1.6 Destroy session

- method: `destroy_session(sid)`

Example GET request

```
{
  "id": 1, "jsonrpc": "2.0",
  "method": "destroy_session",
  "params": {"sid": "5acaeff48a72647ed4a43c2c2fdeab53"}
}
```

Example response

```
{
       "id":1, "jsonrpc":"2.0",
       "result": {"message": "Session destroyed!"}
}
```

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

Only users with role admin are allowed to import / export configuration files.

- URL: `/cgi-bin/luci/api/files/config`
- HTTP GET request for configuration file export (deprecated)
- Multi part HTTP POST request for configuration file import and export
  - Name field for configuration file part is `config`.
  - Optional form value `format` (default: `snmp`) to specify configuration file format for export:
    - `snmp`: Use SNMP file format
    - `cli`: Use CLI file format
  - Optional form value `reset` (default: `false`). If reset is `true`, then the device configuration will be reset before importing.
  - Optional form value `action` with values `import` or `export`
    - not provided: import non-encrypted configuration file (deprecated)
    - `import`: Import configuration file.
    - `export`: Export configuration file.
  - Optional form value `enc_password`.
    - not provided: import or export non-encrypted configuration file.
    - provided: import or export encrypted configuration file.
- 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"`
- For file encryption, the following openssl command is used: `openssl enc -aes-256-cbc -pbkdf2 -iter 2048 -in <config_file> -pass pass:<password>`
- For file decryption, the following openssl command is used: `openssl enc -d -aes-256-cbc -iter 2048 -in <config_file> -pass pass:<password>`

16.2.2 Syslog export

Only users with role admin are allowed to export syslog.

- 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 Messages export

16.2.4 Security Log export

Only users with role admin are allowed to export security log.

- URL: `/cgi-bin/luci/api/files/securelog`
- HTTP GET request for Security Log export

16.2.5 Support File export

- URL: `/cgi-bin/luci/api/files/support`
- HTTP GET request for support file export

16.2.6 Cellular Support File export

- URL: `/cgi-bin/luci/api/files/cellular_support`
- HTTP GET request for cellular support file export

16.2.7 Firmware upload and upgrade

Only users with role admin are allowed to upload and upgrade the Firmware.

- 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.
  - The contained items in the list of all available configuration depend on the logged-in role.

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/unauthorized keys. This element is only present if invalid/unauthorized 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"}
    ]
  }
}
```

16.4 Status API

- Implemented as JSON RPC version 2.0 API with `/cgi-bin/luci/api/status` as it's top level URL.
- Providing status information.

16.4.1 RPC Methods

16.4.1.1 RPC Methods Errors

Number	Category	Message
-1	General Error	E.g. "general error" or "set failed"
-2	Not Available	E.g. "element not available"

16.4.1.2 Get status

- method: `get(view)`
  - `view`: Supported are all views which are also supported by CLI.
- Note:
  - If `"params"` is not given, then return list of supported views.

Example POST request: get list of views

```
{
  "id": 1, "jsonrpc": "2.0",
  "method": "get"
}
```

Example response: list of views

```
{
  "id": 1, "jsonrpc": "2.0",
  "result": {
    "views": [ "list", "ac", "cellular", "ip", "link",
               "lldp", "neighbour", "network", "nlm",
               "route", "rule", "steering", "summary" ]
  }
}
```

Example POST request: get "ip" view

```
{
  "id": 1, "jsonrpc": "2.0",
  "method": "get",
  "params": { "view": "ip" }
}
```

Example response: "ip" view

```
{
  "id": 1, "jsonrpc": "2.0",
  "result": {
    "name": "ip", "type": "view",
    "addresses":[
      "lo UP 127.0.0.1/8 ",
      "br0.vlan0 UP 192.168.1.20/24 169.254.214.168/16"
    ]
  }
```

```
}
```

16.5 Using WebAPI with curl

This chapter will show you how to work with the WebAPI using the curl command line tool which is well established and supported on the most common platforms.

The examples only cover the basic usage. Depending on the use case more arguments are needed for success (e.g. `-insecure` for self-signed certificates).

The placeholders (e.g. `<username>`) need to be replaced with proper values.

The example contain the curl call and the response if available.

16.5.1 Authentication

Before any request can be made login is needed to acquire a new session:

```
curl \
    --request POST \
    --data '{"id": 1,"jsonrpc": "2.0", "method": "login", "params": { "
       username": "<username>", "password": "<password>" }}' \
    --cookie-jar cookies \
    https://192.168.1.20/cgi-bin/luci/api/auth
{"id":1,"jsonrpc":"2.0","result":{"token":"<your_token>"}}
```

After successful login the session data can be loaded from the device:

```
curl \
    --request POST \
    --data '{"id": 1,"jsonrpc": "2.0", "method": "get_session"}' \
    --cookie cookies \
    https://192.168.1.20/cgi-bin/luci/api/auth
{"id":1,"jsonrpc":"2.0","result":{"username":"admin","role":"admin","provider
    ":"local"}}
```

Every session should be closed after the session is not needed anymore. This can be done by using the `logout` method:

```
curl \
    --request POST \
    --data '{"id": 1,"jsonrpc": "2.0", "method": "logout"}' \
```

```
    --cookie cookies \
    https://192.168.1.20/cgi-bin/luci/api/auth
{"id":1,"jsonrpc":"2.0","result":{"message":"Logout successful"}}
```

Changing password is done by using the `change_password` method:

```
curl \
       --request POST \
       --data '{"id": 1,"jsonrpc": "2.0", "method":"change_password", "params
          ": { "username": "<username>", "old_password": "<old_pw>", "
          new_password": "<new_pw>" }}' \
       https://192.168.1.20/cgi-bin/luci/api/auth
{"id":1,"jsonrpc":"2.0","result":{"message":"Password change successful"}}
```

16.5.2 Files

There are different files to download/uploaded from the device through the JSON-RPI interface. Please read Files API for an overview of all available files.

The following example will export and download the configuration file and store the data in `config.txt`:

```
curl \
    --output config.txt \
    --cookie cookies \
    https://192.168.1.20/cgi-bin/luci/api/files/config
```

The following example will upload and import the configuration files stored in `config.txt`. In this example current staged configuration changes will not be reset:

```
curl \
  --request POST \
  --form config=@config.txt \
  --form reset=false \
  --cookie cookies \
  https://192.168.1.20/cgi-bin/luci/api/files/config
{"result":{"nof_changes":"<number_of_config_changes>"}}
```

16.5.3 Device Configuration

The device configuration can be read and written through the JSON-RPC. The calls use the same configuration element names as they are used by the SNMP interface.

Get the hostname from the device:

```
curl \
    --request POST \
    --data '{"id": 1,"jsonrpc": "2.0", "method": "get", "params": { "items": [
        { "key": "WESTERMO-SW6-MIB::cfgSysHostname.0" } ] }}' \
    --cookie cookies \
    https://192.168.1.20/cgi-bin/luci/api/mib

{"id":1,"jsonrpc":"2.0","result":{"items":[{"value":"Rmodem","type":"string","
   key":"WESTERMO-SW6-MIB::cfgSysHostname.0"}]}}
```

Set a new hostname:

```
curl \
    --request POST \
    --data '{"id": 1,"jsonrpc": "2.0", "method": "set", "params": { "items": [
        { "key": "WESTERMO-SW6-MIB::cfgSysHostname.0", "value": "NewHostName"
       } ] }}' \
    --cookie cookies \
    https://192.168.1.20/cgi-bin/luci/api/mib
{"id":1,"jsonrpc":"2.0","result":{"items":[{"key":"WESTERMO-SW6-MIB::
   cfgSysHostname.0","value":"NewHostName"}]}}
```

Multiple configuration elements can be read/written in the same request.

16.5.4 Status

Get device status information through the JSON-RPC.

Get list of available status views:

```
curl \
       --request POST \
       --data '{"id": 1,"jsonrpc": "2.0", "method": "get"}' \
       --cookie cookies \
       https://192.168.1.20/cgi-bin/luci/api/status
{"id":1,"jsonrpc":"2.0","result":{"views":["list", "ip", "link", "network", "
   route", "summary"]}}
```

Get ip (IP Address) view:

```
curl \
       --request POST \
       --data '{"id": 1,"jsonrpc": "2.0", "method": "get", "params": { "view":
           "ip" }}' \
```

```
       --cookie cookies \
       https://192.168.1.20/cgi-bin/luci/api/status
{"id":1,"jsonrpc":"2.0","result":{"addresses":["lo UP 127.0.0.1/8 ",
 "br0.vlan0 UP 192.168.1.20/24 169.254.214.168/16"]}}
```

17 Message Codes

All Message Codes are recorded in Syslog and can be optionally send as SNMP Notifications (Trap or Inform) (see SNMP Trap).

There are three kinds of message groups:

- Security Log Messages
- Standard Log Messages
- Commissioning Log Messages

It is possible to configure multiple Syslog remotes which can be configured to send messages from all or only selected message groups (see Logging Features).

Example Syslog for Message Code: 502:

```
Mon Feb 21 01:09:48.889 2022 daemon.notice LogBackend: [ NOTICE 502 ] CONFIG:
   Apply process started
```

Example SNMP Notification for Message Code: 502:

```
sysUpTime=60454
rmTraps=notifyNotice(6)
rmTrapMsg="[ NOTICE 502 ] CONFIG: Apply process started"
```

17.1 Security Log Messages

Security Log Message are additionally stored in a non-volatile memory. This allows to retrieve Security Log Messages after reboot or power down via Web Interface or WebAPI.

Security Log Rate Limiter: New events triggering a Security Log Message are ignored for the next minute if the same Security Log Message has already been generated at least 5 times during the last minute.

Whenever the Rate Limiter for a Message gets activated, this is reported with Message Code: 3044.

Example Security Log Message for Message Code: 502:

```
Sun May 1 17:14:37 2022 [ NOTICE 502 ] CONFIG: Apply process started
```

17.1.0.1 Message Code: 202

```
[ NOTICE 202 ] Firmware update started by <user>
```

This message is sent when a firmware update is initiated.

17.1.0.2 Message Code: 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".

17.1.0.3 Message Code: 324

```
[ INFO 324 ] Kernel log(s) reset successful
```

This message is generated when kernel log(s) has been reset successfully.

17.1.0.4 Message Code: 325

```
[ ERROR 325 ] Unexpected Bootloader Version (<val> <<val>). You may fix this
   by reflashing the desired firmware.
```

This message is generated when the version of the currently installed bootloader is older than expected. A potential way to fix this is to reinstall the firmware.

17.1.0.5 Message Code: 435

```
[ ERROR 435 ] Radius <type> server <ip>:<port> not available (reason <reason>)
```

This message is sent when the connection to the RADIUS server fails. [AP only]

- <type>:
  - authentication: Authentication server
  - accounting: Accounting server
- <ip>: IP address of server
- <port>: Port of server
- <reason>: Possible reasons:
  - not available: Server is not available
  - shared secret: Shared secret does not match
  - rejected: Connection is rejected

17.1.0.6 Message Code: 436

```
[ INFO 436 ] EAP-TLS: Start session
```

This message is sent when a full EAP-(T)TLS authentication has been started. [STA only]

17.1.0.7 Message Code: 437

```
[ ERROR 437 ] Certificate <type> is <state>
```

This message is sent when the T(TLS) certificate validation fails. [STA only]

- <type>: Certificate types:
  - ca: CA certificate
  - client: Client certificate
- <state>: Certificate state:
  - wrong: Certificate contains invalid information
  - missing: Certificate is missing
  - expired: Certificate is expired

17.1.0.8 Message Code: 438

```
[ ERROR 438 ] TTLS: Invalid username and/or password
```

This message is sent when the TTLS Username and/or password is not recognized by the RADIUS server. [STA only]

17.1.0.9 Message Code: 440

```
[ WARNING 440 ] EAP: Auth. session failure: EAP ID: <ip_address>, Reason: <
   reason>
```

This message is sent when the EAP authentication session fails.

- <ip_address>: IP address of the RADIUS authentication server
- <reason>: Possible reasons (as defined in IEEE 802.1X standard '8.5.4 Authenticator PAE state machine'):
  - 1: initialize
  - 2: disconnected
  - 3: connecting
  - 4: authenticating
  - 5: authenticated
  - 6: aborting
  - 7: held
  - 8: forceAuth
  - 9: forceUnauth
  - 10: restart

17.1.0.10 Message Code: 441

```
[ NOTICE 441 ] RADIUS: Access-Reject: RADIUS Server: <ip_address>
```

This message is sent when the access to the RADIUS server is rejected.

- <ip_address>: IP address of the RADIUS server

17.1.0.11 Message Code: 442

```
[ WARNING 442 ] RADIUS: Response Auth. failed: RADIUS Server: <ip_address>
```

This message is sent when the authentication to the RADIUS server fails.

- <ip_address>: IP address of the RADIUS server

17.1.0.12 Message Code: 443

```
[ INFO 443 ] RADIUS: Switching to <priority> server
```

This message is sent when the connection to the RADIUS server is switched to an alternative server.

- <priority>: Possible priorities:
  - PRIMARY: Primary configured RADIUS server
  - SECONDARY: Secondary configured RADIUS server

17.1.0.13 Message Code: 444

```
[ INFO 444 ] Certmgmt: Certificate with ID <id> expiration. Serial: <serial>;
   Exp.date: <exp date>
```

This message is sent when a certificate is about to expire.

- <id>: ID of the certificate in the certificate store.
- <serial>: The serial of the certificate.
- <exp date>: The expire date (enddate) of the certificate.

17.1.0.14 Message Code: 445

```
[ INFO 445 ] Certmgmt: CRL with ID <id> expiration. Exp.date: <exp date>
```

This message is sent when a CRL is about to expire.

- <id>: ID of the CRL in the certificate store.
- <exp date>: The expire date (nextupdate) of the CRL.

17.1.0.15 Message Code: 446

```
[ INFO 446 ] SCEP: CA certificate retrieval (<index>): <result>
```

This message is sent when SCEP get a CA.

- <index>: Index of the entry in the cfgScepTable
- <result>:
  - SUCCESS: Import of the CA certificate was successful.
  - FAILURE <1>: Import error (e.g. download failed)
  - FAILURE <3>: Verification of the certificate failed (e.g. wrong format)

17.1.0.16 Message Code: 447

```
[ INFO 447 ] SCEP: Client certificate enrollment (<index>): <result>
```

This message is sent on SCEP enroll.

- <index>: Index of the entry in the cfgScepTable.
- <result>
  - SUCCESS: Enroll was successful.
  - FAILURE <1>: General error
  - FAILURE <2>: General failure

- FAILURE <3>: Download pending
- FAILURE <70>: Invalid algorithm identifier
- FAILURE <71>: Integrity check failure
- FAILURE <72>: Invalid request
- FAILURE <73>: Invalid signing time
- FAILURE <74>: Invalid certificate identifier
- FAILURE <89>: Network timeout
- FAILURE <91>: Self sign failure
- FAILURE <92>: File handling error
- FAILURE <95>: Network send failure
- FAILURE <97>: PKCS7 error
- FAILURE <99>: Missing PKI status

17.1.0.17 Message Code: 448

```
[ INFO 448 ] SCEP: Client certificate enrollment (<index>): <result>
```

This message is sent on SCEP re-enroll.

- <index>: Index of the entry in the cfgScepTable.
- <result>
  - SUCCESS: Re-enroll was successful.
  - FAILURE <1>: General error
  - FAILURE <2>: General failure
  - FAILURE <3>: Download pending
  - FAILURE <4>: System busy
  - FAILURE <70>: Invalid algorithm identifier

  - FAILURE <71>: Integrity check failure
  - FAILURE <72>: Invalid request
  - FAILURE <73>: Invalid signing time
  - FAILURE <74>: Invalid certificate identifier
  - FAILURE <89>: Network timeout
  - FAILURE <91>: Self sign failure
  - FAILURE <92>: File handling error
  - FAILURE <95>: Network send failure
  - FAILURE <97>: PKCS7 error
  - FAILURE <99>: Missing PKI status

17.1.0.18 Message Code: 449

```
[ INFO 449 ] Certmgmt: CRL download for CA ID <id>: <result>
```

This message is sent when a CRL was downloaded.

- <id>: ID of the related CA certificate.
- <result>
  - SUCCESS: Download of the CRL and import into the certificate store was successful.
  - FAILURE <1>: Download/import error.
  - FAILURE <3>: Verification of the certificate failed (e.g. wrong format).

17.1.0.19 Message Code: 463

```
[ INFO 463 ] SCEP: re-enroll (<index>): Transaction ID: <transaction id>; DN:
   <subject>; CA DN: <ca subject>
```

This message is sent when SCEP re-enroll was started.

- <index>: Index of the entry in the cfgScepTable.
- <transaction id>: Transaction Identifier.
- <subject>: Subject of the CSR.
- <ca subject>: Subject of the CA certificate.

17.1.0.20 Message Code: 465

```
[ INFO 465 ] SCEP: certificate polling (<index>): retry in <delay> minutes
```

This message is sent when SCEP re-enroll failed and will be retried later.

- <index>: Index of the entry in the cfgScepTable.
- <delay>: Retry to re-enroll after this delay.

17.1.0.21 Message Code: 466

```
[ INFO 466 ] Import <data type> from <url>
```

This message is sent when a certificate gets importer through SNMP.

- <data type>:
  - crl: Import a CRL.
  - cert: Import a certificate.
  - key: Import a private key.
  - pkcs12: Import files from a PKCS12 container.
- <url>: URL to download the certificate from.

17.1.0.22 Message Code: 467

```
[ INFO 467 ] SCEP: Retrieving CA certificate (<index>): CA ID: <id>
```

This message is sent when SCEP retrieve a CA certificate.

- <index>: Index of the entry in the cfgScepTable.
- <id>: Destination ID in the certificate store for the retrieved CA certificate.

17.1.0.23 Message Code: 468

```
[ INFO 468 ] Certificate import: <result>
```

This message is sent after import of a certificate into the certificate store. The content of <result> can be one of the following values:

- SUCCESS: Import of the certificate was successful.
- FAILURE <1>: Import error (e.g. download failed)
- FAILURE <3>: Verification of the certificate failed (e.g. wrong format)

17.1.0.24 Message Code: 469

```
[ INFO 469 ] Certmgmt: Import CRL for CA ID <id> from <url>
```

This message is sent when before a CRL gets imported into the certificate store.

- <id>: ID of the related CA in the certificate store.
- <url>: URL to download the CRL from.

17.1.0.25 Message Code: 470

```
[ INFO 470 ] SCEP: enroll (<index>): Transaction ID: <transaction id>; DN: <
   subject>; CA DN: <ca subject>
```

This message is sent when SCEP enroll was started.

- <index>: Index of the entry in the cfgScepTable.
- <transaction id>: Transaction Identifier.
- <subject>: Subject of the CSR.
- <ca subject>: Subject of the CA certificate.

17.1.0.26 Message Code: 471

```
[ INFO 471 ] Certmgmt: CRL with ID <id> stored. ISSUE_DATE: <issue date>;
   ISSUER_DN: <issuer dn>
```

This message is sent when after a CRL was imported into the certificate store.

- <id>: ID of the CRL in the certificate store.
- <issue date>: Issue date of the CRL (lastupdate).
- <issuer dn>: CRL issuer subject.

17.1.0.27 Message Code: 472

```
[ INFO 472 ] AP_TIME: invalid AP_TIME from <mac>: ap time (<ap time>) /
   systime (<sys time>)
```

This message is sent when the date and time provided by an AP (<ap time>) is considered invalid.

- <mac>: MAC address of the AP.
- <ap time>: Time provided by the AP in Epoch.
- <sys time>: System time of the unit in Epoch.

17.1.0.28 Message Code: 473

```
[ NOTICE 473 ] RADIUS: retransmit to auth server <ip_address>
```

This message is sent when a retransmission to the RADIUS authentication server occurs.

- <ip_address>: IP address of the RADIUS authentication server

17.1.0.29 Message Code: 474

```
[ INFO 474 ] AP_TIME: successfully set AP_TIME from <mac>: ap time (<ap time>)
    / systime (<sys time>)
```

This message is sent when the date and time provided by an AP (<ap time>) was correctly set.

- <mac>: MAC address of the AP.
- <ap time>: Time provided by the AP in Epoch.
- <sys time>: System time of the unit in Epoch.

17.1.0.30 Message Code: 475

```
[ WARNING 475 ] EAP-TLS: session failed, code=<reason>
```

This message is sent when the EAP-TLS session fails. The reason codes represent the peer state machine status (as defined in section 4.5 'Peer State Machine States' of rfc4137)

- <reason>: Possible reason codes
  - 0: INITIALIZE
  - 1: DISABLED
  - 2: IDLE
  - 3: RECEIVED
  - 4: GET_METHOD
  - 5: METHOD
  - 6: SEND_RESPONSE
  - 7: DISCARD
  - 8: NOTIFICATION
  - 9: RETRANSMIT
  - 10: SUCCESS
  - 11: FAILURE

17.1.0.31 Message Code: 476

```
[ INFO 476 ] EAP-TLS: successfully closed session
```

17.1.0.32 Message Code: 477

```
[ INFO 477 ] EAP: Auth. Session started. EAP ID: <id>
```

This message is sent when the EAP-TLS session is started.

- <id>: EAP session ID

17.1.0.33 Message Code: 478

```
[ INFO 478 ] EAP: Auth. Session success. EAP ID: <id>
```

This message is sent when the EAP-TLS session is successfully authenticated.

- <id>: EAP session ID

17.1.0.34 Message Code: 479

```
[ INFO 479 ] Certmgmt: Certificate with ID <id> stored. DN: <subject>; SN: <
   serial>; ISSUER: <issuer>
```

This message is sent when after a new certificate is stored in the certificate store.

- <id>: ID of the certificate in the certificate store.
- <subject>: Certificate subject.
- <serial>: Certificate serial number.
- <issuer>: Issuer of the certificate

17.1.0.35 Message Code: 480

```
[ INFO 480 ] Certmgmt: Certificate with ID <id> (DN: <curr subject>; SN: <curr
    serial>; ISSUER: <curr issuer>) replaced by new certificate (DN: <next
   subject>; SN: <next serial>; ISSUER: <next issuer>)
```

This message is sent when a certificate in the certificate store is replaced by the new, prepared certificate.

- <id>: ID of the certificate in the certificate store.
- <curr subject>: Subject of the current certificate.
- <curr serial>: Serial number of the new certificate.
- <curr issuer>: Issuer of the current certificate.
- <next subject>: Subject of the new certificate.
- <next serial>: Serial number of the new certificate.
- <next issuer>: Issuer of the new certificate.

17.1.0.36 Message Code: 481

```
[ INFO 481 ] Certificate (DN: <subject>; SN: <serial>) is set as certificate
   for TLS handshake for 802.1X NAC interface
```

This message is sent when a certificate is used for a TLS handshake by a 802.1X NAC interface.

- <subject>: Subject of the certificate.
- <serial>: Serial number of the certificate.

17.1.0.37 Message Code: 482

```
[ INFO 482 ] Certificate (DN: <subject>; SN: <serial>) is set as certificate
   for TLS handshake for HTTPS server interface
```

This message is sent when a certificate is used for a TLS handshake by a connection to an HTTPS server.

- <subject>: Subject of the certificate.
- <serial>: Serial number of the certificate.

17.1.0.38 Message Code: 483

```
[ INFO 483 ] Certificate (DN: <subject>; SN: <serial>) is set as trust anchor
   for 802.1X NAC interface
```

This message is sent when a CA certificate is used as a trusted anchor by a 802.1X NAC interface.

- <subject>: Subject of the certificate.
- <serial>: Serial number of the certificate.

17.1.0.39 Message Code: 484

```
[ INFO 484 ] Certificate (DN: <subject>; SN: <serial>) is set as trust anchor
   for LDAPS client
```

This message is sent when a CA certificate is used as a trusted anchor for a connection to an LDAPS server.

- <subject>: Subject of the certificate.
- <serial>: Serial number of the certificate.

17.1.0.40 Message Code: 485

```
[ INFO 485 ] Certmgmt: Key with ID <id> stored
```

This message is sent after a private key was imported to the certificate store.

- <id>: ID of the private key in the certificate store.

17.1.0.41 Message Code: 486

```
[ INFO 486 ] Certificate (DN: <subject>; SN <serial>) set as trust anchor for
   HTTPS client
```

This message is sent when a CA certificate is used as a trusted anchor for a connection to an HTTPS server.

- <subject>: Subject of the certificate.
- <serial>: Serial number of the certificate.

17.1.0.42 Message Code: 487

```
[ INFO 487 ] Certmgmt: CRL with ID <id> (ISSUER: <curr issuer>) replaced by
   new CRL (ISSUER: <next issuer>)
```

This message is sent when a CRL in the certificate store is replaced by a new, prepared CRL.

- <id>: ID of the CRL in the certificate store.
- <curr issuer>: Issuer of the current CRL.
- <next issuer>: Issuer of the new CRL.

17.1.0.43 Message Code: 488

```
[ INFO 488 ] Certificate (DN: <subject>; SN: <serial>) is set as certificate
   for TLS handshake for SYSLOG over TLS interface
```

This message is sent when a certificate is used for a TLS handshake by a connection to an SYSLOG over TLS link.

- <subject>: Subject of the certificate.
- <serial>: Serial number of the certificate.

17.1.0.44 Message Code: 489

```
[ INFO 489 ] Certificate (DN: <subject>; SN: <serial>) is set as trust anchor
   for SYSLOG over TLS interface
```

This message is sent when a CA certificate is used as a trusted anchor by a SYSLOG over TLS interface.

- <subject>: Subject of the certificate.
- <serial>: Serial number of the certificate.

17.1.0.45 Message Code: 490

```
[ NOTICE 490 ] RADIUS: retransmit to acct server <RadiusAcctServerIPAddr>
```

This message is sent when a retransmission to the RADIUS accounting server occurs.

- <ip_address>: IP address of the RADIUS accounting server

17.1.0.46 Message Code: 503

```
[ NOTICE 503 ] CONFIG: Apply by <user> successful. Changes: <changes>, Hash: <
   hash>
```

This message is sent when an new configuration was successfully applied.

- <changes>: 'Yes' if there were changes to apply. 'No' if there were no changes to apply, no services were changed/restarted.
- <hash>: Hash of the new applied configuration as it's also available in swCfgHash. Used hash algorithm is SHA384.

17.1.0.47 Message Code: 504

```
[ WARNING 504 ] CONFIG: Apply by <user> failed
```

This message is sent when the new configuration is invalid and apply process failed.

17.1.0.48 Message Code: 714

```
[ ERROR 714 ] NET: Network init failed - RECOVERY MODE!
```

This message is sent when the device is in recovery mode due to network init failure during boot-up. In recovery mode a minimal network configuration is applied where all Ethernet/Fiber interfaces are assigned to a single bridge with default IP (192.168.1.20). Please perform a Factory Reset to restore a valid network configuration.

17.1.0.49 Message Code: 826

```
[ INFO 826 ] FOTA: <msg>
```

This message is used to send informational fota messages.

- <msg>: arbitrary message string

17.1.0.50 Message Code: 827

```
[ ERROR 827 ] FOTA: <msg>
```

This message is used to send error fota messages.

- <msg>: arbitrary message string

17.1.0.51 Message Code: 2700

```
[ NOTICE 2700 ] IF-MIB: linkUp (<ifIndex>, <ifAdminStatus>, <ifOperStatus>)
```

This message is sent when an interface comes up.

- <ifIndex>: unique network interface index
- <ifAdminStatus>:
  - 0: interface is disabled
  - 1: interface is enabled
- <ifOperStatus>:
  - 0: interface is down
  - 1: interface is up

17.1.0.52 Message Code: 2701

```
[ NOTICE 2701 ] IF-MIB: linkDown (<ifIndex>, <ifAdminStatus>, <ifOperStatus>)
```

This message is sent when an interface goes down.

- <ifIndex>: unique network interface index
- <ifAdminStatus>:
  - 0: interface is disabled
  - 1: interface is enabled

- <ifOperStatus>:
  - 0: interface is down
  - 1: interface is up

17.1.0.53 Message Code: 3001

```
[ NOTICE 3001 ] Start up of the device <fw version>, <fw revision>, <
   bootloader version>, <bootloader build date>, <config hash>
```

This message is sent when the device started up.

- <fw version>: Firmware version (see swFwVersion).
- <fw revision>: Firmware revision (see swFwRevision).
- <bootloader version>: Bootloader version.
- <bootloader build date>: Bootloader build date.
- <config hash>: Configuration has (see swCfgHash).

17.1.0.54 Message Code: 3010

```
[ NOTICE 3010 ] <user> credentials have been modified
```

This message is sent when a local users password has been changed.

17.1.0.55 Message Code: 3011

```
[ NOTICE 3011 ] <user> successfully connected to the Web Interface / WebAPI
```

This message is sent when a local or LDAP user has successfully logged-in to the Web Interface / WebAPI.

17.1.0.56 Message Code: 3012

```
[ NOTICE 3012 ] Connection attempt to the web-based management interface
   rejected
```

Connection attempt to the web-based management interface rejected.

17.1.0.57 Message Code: 3014

```
[ NOTICE 3014 ] Too many rejected connection attempts in a short lapse of time
    to the device administration interface: anti brute-force mechanism
   triggered
```

Too many rejected connection attempts in a short lapse of time to the device administration interface: anti brute-force mechanism triggered.

17.1.0.58 Message Code: 3021

```
[ NOTICE 3021 ] TLS Client: Fatal alert error <TLS error> appears. Connection
   closed
```

This message is sent when a TLS client connection is closed due to a fatal error. A TLS client connection is used to access HTTPS and LDAPS services.

The content of the <TLS error> message is dependent on the SSL component providing the TLS, which with currently used openssl is '<error ID> (<error text>)' with

- <error ID>: integer value of the fatal alert error occured
- <error text>: human readable description of the error

17.1.0.59 Message Code: 3022

```
[ NOTICE 3022 ] TLS Server: Fatal alert error appears. Connection closed.
```

TLS Server: Fatal alert error appears. Connection closed.

17.1.0.60 Message Code: 3023

```
[ INFO 3023 ] Web Server: Session opened
```

Web Server: Session opened.

17.1.0.61 Message Code: 3024

```
[ NOTICE 3024 ] Web Server: Session closed due to timeout
```

Web Server: Session closed due to timeout.

17.1.0.62 Message Code: 3026

```
[ NOTICE 3026 ] SNMP packet received not valid.
```

SNMP packet received not valid.

17.1.0.63 Message Code: 3027

```
[ NOTICE 3027 ] SNMP packet authentication failed.
```

SNMP packet authentication failed.

17.1.0.64 Message Code: 3028

```
[ INFO 3028 ] Web Server: Session closed
```

Web Server: Session closed.

17.1.0.65 Message Code: 3035

```
[ WARNING 3035 ] Firmware update by <user> failed
```

This message is sent when the firmware updated started by a logged-in user failed.

17.1.0.66 Message Code: 3036

```
[ NOTICE 3036 ] Factory reset requested by <user> has been performed
```

This message is sent when factory reset is requested by a logged-in user or by the physical reset plug.

17.1.0.67 Message Code: 3037

```
[ NOTICE 3037 ] SNMP packet received not valid: non-implemented OID.
```

SNMP packet received not valid: non-implemented OID.

17.1.0.68 Message Code: 3044

```
[ NOTICE 3044 ] Security log: rate-limiter triggered for message <code>
```

This message is sent when the rate-limiter is triggered for a security log message.

17.1.0.69 Message Code: 3045

```
[ INFO 3045 ] CLI: Session opened for '<user>' (<role>) from <host>
```

This message is sent when a CLI session opened.

17.1.0.70 Message Code: 3046

```
[ NOTICE 3046 ] CLI: Session closed due to timeout for '<user>' (<role>)
```

This message is sent when a CLI session is closed due to timeout.

17.1.0.71 Message Code: 3047

```
[ INFO 3047 ] CLI: Session closed for '<user>' (<role>)
```

This message is sent when a CLI session is closed by the client.

17.1.0.72 Message Code: 3048

```
[ NOTICE 3048 ] CLI: Login attempt with invalid credentials from <host>
```

This message is sent when a CLI session is rejected for the given credentials.

17.1.0.73 Message Code: 3481

```
[ INFO 3481 ] Export <data type> with ID <id> to <location>
```

This message is sent on export a certificate from the certificate store.

- <data type>: One of cert, key or crl
- <id>: ID of the certificate file used to reference it from the services
- <location>: An URL if uploaded to a TFTP server or *Web Interface/WebAPI*

17.1.0.74 Message Code: 3482

```
[ INFO 3482 ] Delete <data type> with ID <id>
```

This message is sent if a certificate from the certificate store was removed.

- <data type>: One of cert, key or crl
- <id>: ID of the certificate file used to reference it from the services

17.1.0.75 Message Code: 3483

```
[ WARNING 3483 ] EAP-TLS: verification failure: <reason>
```

This message is sent when the EAP-TLS session verification fails.

- <reason>: Possible reasons
  - INVALID_DN

17.2 Standard Log Messages

17.2.0.1 Message Code: 0

```
[ INFO 0 ] <error reset cause>
```

This message informs that a pending error condition was reset, where <error reset cause> can be one of 'Reset errors by WebGUI' or 'Manual Error Reset'

17.2.0.2 Message Code: 100

```
[ ERROR 100 ] SYS_MON: Voltage sensor '<sensor name>' is out of range <value>
   / [<min value>, <max value>]
```

This message is sent when the supply voltage or one of the internal voltages are outside the specified limits. Those limits are hardware dependent and are provided as:

- <sensor name>: given name to identify the sensor
- <value>: measured value that triggered the alarm
- [<min value>, <max value>]: valid range for this sensor

17.2.0.3 Message Code: 101

```
[ ERROR 101 ] SYS_MON: Temperature sensor '<sensor name>' is out of range <
   value> / [<min value>, <max value>]
```

This is the equivalent for temperature sensors, with identical parameters:

- <sensor name>: given name to identify the sensor
- <value>: measured value that triggered the alarm
- [<min value>, <max value>]: valid range for this sensor

17.2.0.4 Message Code: 105

```
[ CRITICAL 105 ] SYS_MON: Failure by reading value <sensor name>. This value
   isn't monitored anymore
```

When a sensor can't be accessed, its monitoring is suspended to prevent continuous alarms. This message informs of such failures, including provision of the <sensor name>.

17.2.0.5 Message Code: 130

```
[ 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>
```

This message is sent periodically containing current 'iw <iface> station dump'. The format is:

- <interface>, <mac>, <inactive time>: the wireless interface, station MAC and its global inactivity time
- <rx bytes>, <rx packets>: number of bytes and packets received
- <tx bytes>, <tx packets>, <tx retries>, <tx failed>: number of bytes, packets, retries and failures transmitted
- <signal combined>, <avg signal combined>: current and averaged signal combined over all chains
- <signal ch0>, <avg signal ch0>: current and averaged signal for chain 0
- <signal ch1>, <avg signal ch1>: current and averaged signal for chain 1
- <signal ch2>, <avg signal ch2>: current and averaged signal for chain 2

17.2.0.6 Message Code: 131

```
[ INFO 131 ] <number of stations>|<tx packets>|<tx retries>|<6 mbps>|<9 mbps>|
   <12 mbps>|<18 mbps>|<24 mbps>|<36 mbps>|<48 mbps>|<54 mbps>
```

This message is sent periodically containing counters of the wireless interface. The format is:

- <number of stations>: number of connected stations
- <tx packets>, <tx retries>: number of total packets and retries transmitted
- <X mbps>: number of total packets transmitted at rate X

17.2.0.7 Message Code: 200

```
[ NOTICE 200 ] Device has restarted because of <reset cause>
```

Durig the boot-process this message is sent to provide the reason for the boot, where <reset cause> can be 'coldstart', 'warmstart', 'watchdog' or 'oops'

17.2.0.8 Message Code: 201

```
[ NOTICE 201 ] System startup
```

This message is sent when the system starts up.

17.2.0.9 Message Code: 203

```
[ NOTICE 203 ] System reboot
```

This message is sent when a system reboot is issued.

17.2.0.10 Message Code: 205

```
[ NOTICE 205 ] Factory Reset process started
```

A Factory Reset removes all changes done on the device, i.e. the configuration is set to default and all files (e.g. certificates) are removed. During the Factory Reset process the device will reboot.

17.2.0.11 Message Code: 207

```
[ WARNING 207 ] Invalid upgrade image for this platform
```

17.2.0.12 Message Code: 208

```
[ ERROR 208 ] Corrupt firmware package!
```

17.2.0.13 Message Code: 209

```
[ ERROR 209 ] Transfer firmware to device failed!
```

17.2.0.14 Message Code: 210

```
[ WARNING 210 ] Decryption of the upgrade image failed!
```

17.2.0.15 Message Code: 220

```
[ NOTICE 220 ] Start writing device identification memory!
```

17.2.0.16 Message Code: 221

```
[ NOTICE 221 ] Device identification memory successfully updated!
```

17.2.0.17 Message Code: 222

```
[ CRITICAL 222 ] Failed to update device identification memory!
```

17.2.0.18 Message Code: 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.

17.2.0.19 Message Code: 320

```
[ ERROR 320 ] BIST: Daemon '<daemon>' isn't running, recover it
```

This message is sent when the BIST detects a daemon which is not running.

- <daemon>: name of daemon which is not running and which BIST will restart

17.2.0.20 Message Code: 321

```
[ ERROR 321 ] BIST: Daemon watchdog is not running - force restart
```

This message is generated when watchdog process is not running. System reboots afterwards.

17.2.0.21 Message Code: 322

```
[ CRITICAL 322 ] Critical hardware failure: <failure_description>
```

This message is generated when a critical hardware failure has been detected during boot up. Possible failure descriptions:

- RF card <card_index> was not found in slot <pci_slot>!
- RF card <card_index> has a wrong PCI device name!
- RF card <card_index> was not recognized as wireless device!
- RF card <card_index> MAC address file is missing!
- RF card <card_index> has a wrong MAC address signature: <mac_address>
- LTE device not recognized, missing: <device>!
- GNSS device not recognized, missing: <device>!
- Development MAC address found
- Unknown product type!
- Failed to access MRAM!

17.2.0.22 Message Code: 330

```
[ INFO 330 ] BBMON: Backbone is down
```

This message is generated when the connection to the backbone is detected as lost.

17.2.0.23 Message Code: 331

```
[ INFO 331 ] BBMON: Backbone is up
```

This message is generated when the connection to the backbone is regained.

17.2.0.24 Message Code: 340

```
[ INFO 340 ] NLM: monitor <id> is UP - <info>
```

This message is generated when a monitor changes its state to UP.

- <id>: monitor ID
- <info>: monitor type specific info, e.g. 'PHY: eth0' or 'ICMP: 10.8.6.1

17.2.0.25 Message Code: 341

```
[ INFO 341 ] NLM: monitor <id> is DOWN - <info>
```

This message is generated when a monitor changes its state to DOWN.

- <id>: monitor ID
- <info>: monitor type specific info, e.g. 'PHY: eth0' or 'ICMP: 10.8.6.1

17.2.0.26 Message Code: 350

```
[ INFO 350 ] ICL: connection established (mode: <mode>)
```

This message is generated when ICL successfully established a connection to a partner.

- <mode>: 'AP' or 'STA'

17.2.0.27 Message Code: 351

```
[ INFO 351 ] ICL: connection lost (mode: <mode>)
```

This message is generated when ICL lost connection to the partner.

- <mode>: 'AP' or 'STA'

17.2.0.28 Message Code: 360

```
[ INFO 360 ] CELLULAR: Link is up (iface: '<interface>', slot: '<slot>')
```

This message is generated when cellular link is detected as up:

- <iface>: interface name
- <slot>: number of the active SIM slot

17.2.0.29 Message Code: 361

```
[ INFO 361 ] CELLULAR: Link is down (iface: '<iface>', slot <slot>, reason: '<
   reason>')
```

This message is generated when cellular link is detected as down:

- <iface>: interface name
- <slot>: number of the active SIM slot
- <reason>: Possible reasons:
  - IP link down
  - SIM <slot> was removed or is inaccessible
  - low RSSI
  - weak serving-cell
  - can't reach monitored hosts
  - QMI link down

17.2.0.30 Message Code: 450

```
[ NOTICE 450 ] Forced deauthentication of all stations
```

Forced deauthentication of all associated stations. [AP only]

17.2.0.31 Message Code: 451

```
[ WARNING 451 ] Noisefloor above limit
```

Triggered when station's measured noise floor is above configured limit. [STA only]

17.2.0.32 Message Code: 452

```
[ NOTICE 452 ] Noisefloor below limit
```

Triggered when station's measured noise floor is below configured limit. [STA only]

17.2.0.33 Message Code: 460

```
[ NOTICE 460 ] Authorization timed out
```

Triggered when authentication is timed out i.e. no reply from radius server within the given limit. [STA only]

17.2.0.34 Message Code: 462

```
[ NOTICE 462 ] WPA Key Group Fatal Error
```

DEBUG: The group key state machine has a fatal error

17.2.0.35 Message Code: 500

```
[ CRITICAL 500 ] CONFIG: Unable to connect to IPC system (ubus)!
```

17.2.0.36 Message Code: 501

```
[ CRITICAL 501 ] CONFIG: Unable to read from UCI!
```

17.2.0.37 Message Code: 502

```
[ NOTICE 502 ] CONFIG: Apply process started
```

17.2.0.38 Message Code: 510

```
[ ERROR 510 ] CONFIG: Invalid configuration, reverting to previous
   configuration!
```

17.2.0.39 Message Code: 511

```
[ ERROR 511 ] CONFIG: Unable to save new configuration!
```

17.2.0.40 Message Code: 512

```
[ ERROR 512 ] CONFIG: Unable to apply previous configuration!
```

17.2.0.41 Message Code: 513

```
[ WARNING 513 ] CONFIG: Unable to set config parameter: <parameter>
```

Warning during configuration import.

17.2.0.42 Message Code: 514

```
[ ERROR 514 ] CONFIG: <error>
```

Error detected during configuration validation.

17.2.0.43 Message Code: 515

```
[ ERROR 515 ] CONFIG: Invalid WLAN operation mode: <mode>
```

Some devices may not support all operation modes. Please check the data-sheet of the product.

17.2.0.44 Message Code: 516

```
[ WARNING 516 ] Country code <country code> not supported on this product/
   revision!
```

This message is sent when country code is not supported by this product/revision

17.2.0.45 Message Code: 517

```
[ WARNING 517 ] <function> is not supported on this product!
```

Some devices may not support all functions. Please check the manual of the product.

17.2.0.46 Message Code: 518

```
[ WARNING 518 ] CONFIG: Invalid Certificate Chain or Private Key configured.
   Using default self-signed Certificate/Key.
```

When an invalid Certificate Chain or Private Key is configured, the HTTPS server falls back to its own self-signed Certificate/Key.

17.2.0.47 Message Code: 530

```
[ ERROR 530 ] PENDING CHANGES: <text>
```

A firmware upgrade is not allowed if there are pending changes in the configuration.

17.2.0.48 Message Code: 531

```
[ ERROR 531 ] System upgrade to a major version other than <current_major> is
   not supported
```

17.2.0.49 Message Code: 532

```
[ ERROR 532 ] 'Keep config' on a system downgrade is not supported
```

17.2.0.50 Message Code: 533

```
[ CRITICAL 533 ] Config manipulation for an upgrade failed. This is not
   correctable situation. Do a factory reset.
```

17.2.0.51 Message Code: 534

```
[ ERROR 534 ] Firmware up/downgrade not possible because: <reason>
```

Message when a firmware up/down-grade is not possible.

17.2.0.52 Message Code: 540

```
[ ERROR 540 ] Failed to verify key or certificate file
```

17.2.0.53 Message Code: 580

```
[ ERROR 580 ] CONFIG FILE: Transfer failed!
```

17.2.0.54 Message Code: 581

```
[ WARNING 581 ] CONFIG FILE: <warning>
```

Configuration file import warning: Config version, Product, Firmware Name or Version mismatch.

17.2.0.55 Message Code: 582

```
[ ERROR 582 ] CONFIG FILE: Unable to read or parse!
```

17.2.0.56 Message Code: 583

```
[ ERROR 583 ] CONFIG FILE: Incorrect encryption key!
```

17.2.0.57 Message Code: 585

```
[ ERROR 585 ] Certificate import/export failed!
```

17.2.0.58 Message Code: 600

```
[ NOTICE 600 ] AFC <afc name>: AFM did not declare after <seconds> seconds
```

This message is sent when an AFM did not declare at the AFC within a defined timeout.

- <afc name>: name of the reporting AFC
- <seconds>: declare timeout in seconds, defaults to 5 minutes

17.2.0.59 Message Code: 601

```
[ NOTICE 601 ] AFM <afm name>: AFM did not declare after <seconds> seconds
```

This message is sent when an adjacent AFM did not declare at the AFM within a defined timeout.

- <afm name>: name of the reporting AFM
- <seconds>: declare timeout in seconds, defaults to 2 minutes

17.2.0.60 Message Code: 602

```
[ NOTICE 602 ] AFM <afm name>: redundant AFM did not declare after <seconds>
   seconds
```

Not used, since role of AFM is unknown at startup.

17.2.0.61 Message Code: 603

```
[ ERROR 603 ] AFC <afc name>: invalid AFM <afm name> tried to declare
```

This message is sent when an AFM tries to declare at an AFC that is not configured in the list of valid AFMs.

- <afc name>: name of the reporting AFC
- <afm name>: name of the AFM who tries to declare

17.2.0.62 Message Code: 604

```
[ NOTICE 604 ] AFM <afm name>: could not connect to AFC <afc name>
```

This message is sent when an AFM fails to declare to an AFC within 30s.

- <afm name>: name of the reporting AFM
- <afc name>: name of the missing AFC

17.2.0.63 Message Code: 605

```
[ NOTICE 605 ] <smu name>: could not send report to <server>
```

This message is sent when a SMU report can't be sent to the configured server.

- <smu name>: name of the SMU interface
- <server>: target to send the report to

17.2.0.64 Message Code: 630

```
[ INFO 630 ] AFM <afm name>: AFC <afc name> set to operational frequency <freq
   > with penalty <penalty>
```

This message is sent when an AFM has to select an operating frequency for an AFC which has a penalty.

- <afm name>: name of the reporting AFM
- <afc name>: name of the affected AFC
- <freq>: operating frequency selected for the AFC
- <penalty>: penalty for the selected frequency according to channel planning rules

17.2.0.65 Message Code: 631

```
[ INFO 631 ] AFM <afm name>: becoming active for area <area index>
```

This message is sent when an AFM takes over control of an area, which is either it is the first AFM brought up within an area, or it becomes active after the redundant AFM disappeared.

- <afm name>: name of the reporting AFM
- <area index>: area index the AFM controls

17.2.0.66 Message Code: 632

```
[ ERROR 632 ] AFM <afm name>: invalid AFM configured: <remote afm name>
```

This is a generic error message used to inform that an AFM to AFM configuration is invalid, e.g. one AFM being configured as adjacent to another but not vice-versa.

- <afm name>: name of the reporting AFM
- <remote afm name>: name of the remote AFM which caused the misconfiguration

17.2.0.67 Message Code: 650

```
[ INFO 650 ] RADIUS: <role> authentication server is up (server <ipaddr>:<port
   >)
```

This message is sent when the Authenticator declares a RADIUS server as up.

- <role>: role of the RADIUS server, either 'Primary' or 'Secondary'
- <ipaddr>: IP address of the RADIUS server
- <port>: port of the RADIUS server

17.2.0.68 Message Code: 651

```
[ INFO 651 ] RADIUS: <role> authentication server is down (server <ipaddr>:<
   port>)
```

This message is sent when the Authenticator declares a RADIUS server as down.

- <role>: role of the RADIUS server, either 'Primary' or 'Secondary'

- <ipaddr>: IP address of the RADIUS server
- <port>: port of the RADIUS server

17.2.0.69 Message Code: 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.

17.2.0.70 Message Code: 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.

17.2.0.71 Message Code: 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.

17.2.0.72 Message Code: 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.

17.2.0.73 Message Code: 712

```
[ ERROR 712 ] NET: Unable to set a static route!
```

This message is sent when an static route couldn't be set.

17.2.0.74 Message Code: 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.

17.2.0.75 Message Code: 720

```
[ ERROR 720 ] NET: Wireless configuration failed!
```

This message is sent when the configuration manager is not able to set up a wireless interface.

17.2.0.76 Message Code: 721

```
[ ERROR 721 ] NET: A wireless interface can not be bridged to an eth without 4
   addr 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.

17.2.0.77 Message Code: 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.

17.2.0.78 Message Code: 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.

17.2.0.79 Message Code: 731

```
[ WARNING 731 ] NET: The interface <iface> to create the IP address on doesn't
    exist
```

This message is sent if the parent interface for an ip doesn't exist.

- <iface>: parent interface name

17.2.0.80 Message Code: 732

```
[ WARNING 732 ] NET: The interface <iface> 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).

- <iface>: parent interface name

17.2.0.81 Message Code: 740

```
[ ERROR 740 ] NET: Creation of VLAN failed!
```

This message is sent if the creation of a vlan failed.

17.2.0.82 Message Code: 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.

17.2.0.83 Message Code: 742

```
[ ERROR 742 ] NET: Creation of MacVLAN <macvlan> failed!
```

This message is sent if the creation of a macvlan failed.

- <macvlan>: MacVLAN intended to be created

17.2.0.84 Message Code: 743

```
[ ERROR 743 ] NET: Setting MAC-address <mac> on Interface <iface> failed!
```

This message is sent if a MAC address for an interface could not be set.

- <mac>: MAC address intended to be set
- <iface>: interface name

17.2.0.85 Message Code: 750

```
[ WARNING 750 ] NET: Adding a QoS rule failed
```

17.2.0.86 Message Code: 760

```
[ WARNING 760 ] NET: Bridges which forward link local traffic can not have
   more than 2 interfaces (bridge index >=100)
```

17.2.0.87 Message Code: 770

```
[ ERROR 770 ] NET: Error during network init: <failure message>
```

This message is sent when an error occurs during network initialisation. It contains additional information regarding what went wrong.

- <failure message>: arbitrary message why the network initialization failed

17.2.0.88 Message Code: 771

```
[ WARNING 771 ] NET: Runtime Error: <failure message>
```

This message is sent when a network error occurs during operation.. It contains additional information regarding what went wrong.

- <failure message>: arbitrary message describing the network runtime failure

17.2.0.89 Message Code: 772

```
[ ERROR 772 ] WIFI: Error during wireless init: <failure message>
```

This message is sent when an error occurs during wireless initialisation. It contains additional information regarding what went wrong.

- <failure message>: arbitrary message why the wireless initialization failed

17.2.0.90 Message Code: 774

```
[ ERROR 774 ] SERVICE: Error during service init: <failure message>
```

This message is sent when an error occurs during initialisation of a service. It contains additional information regarding what went wrong.

- <failure message>: arbitrary message why the service initialization failed

17.2.0.91 Message Code: 775

```
[ WARNING 775 ] SERVICE: Warning from service: <failure message>
```

This message is sent when a service issues a warning. It contains additional information regarding what went wrong.

- <failure message>: arbitrary warning message from a service

17.2.0.92 Message Code: 800

```
[ INFO 800 ] DFS: Starting CAC on <freq> MHz
```

This message is sent when a CAC or Off-Channel CAC is started.

- <freq>: frequency

17.2.0.93 Message Code: 801

```
[ INFO 801 ] DFS: Radar found on <freq> MHz
```

This message is sent when a radar pattern during In-Service Monitoring, CAC or Off-Channel CAC is detected.

- <freq>: frequency

17.2.0.94 Message Code: 802

```
[ INFO 802 ] DFS: Channel on <freq> MHz becomes Available
```

This message is sent when a channel on the given frequency becomes Available after a CAC or Off-Channel CAC.

- <freq>: frequency

17.2.0.95 Message Code: 803

```
[ INFO 803 ] DFS: Channel on <freq> MHz becomes Usable again
```

This message is sent when a channel on the given frequency becomes Usable after the NOP time.

- <freq>: frequency

17.2.0.96 Message Code: 804

```
[ INFO 804 ] DFS: Starting In-Service Monitoring on <freq> MHz
```

This message is sent when the In-Service Monitoring for the Operating Channel on the given frequency.

- <freq>: frequency

17.2.0.97 Message Code: 805

```
[ INFO 805 ] DFS: All initial CACs done
```

This message is sent when all DFS frequencies have passed the inital CAC.

17.2.0.98 Message Code: 810

```
[ ERROR 810 ] Cellular interface disabled, reason: <reason>
```

This message is sent when the cellular interface was disabled due to HW reasons or wrong PIN configuration.

The provided <reason> can be one of

- *Cellular interface disabled (wwan0)*
- *SIM card could not be initialized (wrong PIN)*
- *QMI interface is not initialized (HW problem)*
- *missing or inaccessible SIM1/2*
- *PIN verification failed*
- *Cellular interface has no SIM cards configured*
- *Failed to switch to SIM slot 1/2*
- *No SIM found in slot 1/2*
- *Switching to SIM slot 1/2 timed out after <seconds> secs*
- *Invalid SIM state for slot 1/2: <state>*
- *failed to enable/disable roaming for slot 1/2*
- *PIN verification failed: invalid SIM pin state <state> in slot 1/2*
- *PIN verification failed: inactive PIN configured for locked SIM in slot 1/2*
- *PIN verification failed: failed to set PIN <pin> for slot 1/2*
- *PIN verification failed: failed to unlock SIM in slot 1/2*

17.2.0.99 Message Code: 820

```
[ INFO 820 ] TS|<uptime>|LTE|<csq_rssi>|<rssi>|<rsrq>|<rsrp>|<sinr>|<mcc>|<mnc
   >|<cellid>|<pcid>|<tac>|<freq_band_ind>|<earfcn>|<dl_bandwidth>|<
   ul_bandwidth>|<FDD or TDD>|<cell_state>

 or

[ INFO 820 ] TS|<uptime>|WCDMA|<csq_rssi>|<rscp>|<ecio>|<mcc>|<mnc>|<cellid>|<
   rac>|<psc>|<uarfcn>|<cell_state>

 or

[ INFO 820 ] TS|<uptime>|NR5G-SA|<csq_rssi>|<rsrq>|<rsrp>|<sinr>|<mcc>|<mnc>|<
   cellid>|<pcid>|<tac>|<scs>|<srxlev>|<duplex_mode>|<arfcn>|<band>|<
   nr_dl_bandwidth>|<cell_state>

 or

[ INFO 820 ] TS|<uptime>|NR5G-NSA|<csq_rssi>|<rsrq>|<rsrp>|<sinr>|<mcc>|<mnc>|
   <pcid>|<scs>|<arfcn>|<band>|<nr_dl_bandwidth>|<cell_state>

 or

[ INFO 820 ] TS|<uptime>|<cell_state>
```

This message is sent when the cellular debug message for signal is enabled (see cfgCellDbgSignal). The output format for LTE, WCDMA and NOTCONNECTED is different as shown above.

- <uptime>: Time in milliseconds since boot-up
- <csq_rssi>: Signal Quality Report RSSI
  - 0: -113dBm or less
  - 1: -111dBm
  - 2..30: -109dBm... -53dBm
  - 31: -51dBm or greater
  - 99: Not known or not detectable
- <rssi>: Received Signal Strength Indication in dBm
- <rsrq>: Reference Signal Received Quality in dB

- <rsrp>: Reference Signal Received Power in dBm
- <sinr>: Signal to Interference plus Noise Ratio. Value range 0-250, corresponding to a measured value from -20dB to +30dB
- <mcc>: Mobile Country Code (first part of the PLMN code)
- <mnc>: Mobile Network Code (second part of the PLMN code)
- <cellid>: Cell ID, 16-bit (GSM) or 28-bit (UMTS). Range: 0-0xFFFFFFF
- <pcid>: Physical Cell ID
- <tac>: Tracking Area Code
- <freq_band_ind>: E-UTRA frequency band
- <earfcn>: E-UTRA-ARFCN of the cell that was scanned
- <dl_bandwidth>: DL bandwidth
  - 0: 1.4MHz
  - 1: 3MHz
  - 2: 5MHz
  - 3: 10MHz
  - 4: 15MHz
  - 5: 20MHz
- <ul_bandwidth>: UL bandwidth
  - 0: 1.4MHz
  - 1: 3MHz
  - 2: 5MHz
  - 3: 10MHz
  - 4: 15MHz
  - 5: 20MHz

- <FDD or TDD>: FDD or TDD mode
- <cell_state>: SEARCH, LIMSRV, NOCONN or CONNECT
  - SEARCH: UE is searching but could not (yet) find a suitable 3G/4G/5G cell
  - LIMSRV: UE is camping on a cell but has not registered on the network
  - NOCONN: UE is camping on a cell and has registered on the network, and it is in idle mode
  - CONNECT: UE is camping on a cell and has registered on the network, and a call is in progress
- <rscp>: Received Signal Code Power level of the cell that was scanned in dBm
- <ecio>: Carrier to noise ratio EC/IO in dB
- <rac>: Routing Area Code. Value range 0-255
- <psc>: Primary Scrambling Code of the cell that was scanned
- <uarfcn>: UTRA-ARFCN of the cell that was scanned
- <scs>: Integer type. NR sub-carrier space
  - 0: 15 kHz
  - 1: 30 kHz
  - 2: 60 kHz
  - 3: 120 kHz
  - 4: 240 kHz
- <srxlev>: Select reception level value for base station in dB (see 3GPP 25.304)
- <duplex_mode>: The 5G NR SA network mode
- <arfcn>: Indicates the SA-ARFCN of the cell that was scanned
- <band>: 32-bit unsigned integer. Frequency band in 5G NR SA network mode
- <nr_dl_bandwidth>: DL bandwidth. (The value is only valid in RRC connected state)
  - 0: 5 MHz

- 1: 10 MHz
- 2: 15 MHz
- 3: 20 MHz
- 4: 25 MHz
- 5: 30 MHz
- 6: 40 MHz
- 7: 50 MHz
- 8: 60 MHz
- 9: 70 MHz
- 10: 80 MHz
- 11: 90 MHz
- 12: 100 MHz
- 13: 200 MHz
- 14: 400 MHz

17.2.0.100 Message Code: 821

```
[ INFO 821 ] TS|<uptime>|<sim_active_slot>|<primary>|<roaming>|<pin_status>|<
   mcc>|<mnc>|<imsi>|<failure_count>|<state_timeout>|<prev_state>|<new_state>|
   <cell_state>
```

This message is sent when the cellular debug message for connection is enabled (see cfgCellDbg-Connection).

- <uptime>: Time in milliseconds since boot-up
- <sim_active_slot>: see swCellSimSlot
- <primary>: see swCellSimPrimary
- <roaming>: see swCellSimRoaming

- <pin_status>: NOT READY, NOT INSERTED, SIM PIN, SIM PUK or READY
  - NOT INSERTED: SIM card not inserted
  - NOT READY: SIM card not detected or registered
  - SIM PIN: SIM PIN requested
  - SIM PUK: SIM PUK requested (must be unlocked externally)
  - READY: SIM card unlocked and registered
- <mcc>: Mobile Country Code (first part of the PLMN code)
- <mnc>: Mobile Network Code (second part of the PLMN code)
- <imsi>: International Mobile Subscriber Identity
- <failure_count>: Number of failed connection attempts
- <state_timeout>: Remaining time in milliseconds until the next forced state change
- <prev_state>: Previous connection state, see swCellConnectionStatus
- <new_state>: New connection state, see swCellConnectionStatus
- <cell_state>: SEARCH, LIMSRV, NOCONN or CONNECT
  - SEARCH: UE is searching but could not (yet) find a suitable 3G/4G/5G cell
  - LIMSRV: UE is camping on a cell but has not registered on the network
  - NOCONN: UE is camping on a cell and has registered on the network, and it is in idle mode
  - CONNECT: UE is camping on a cell and has registered on the network, and a call is in progress

17.2.0.101 Message Code: 825

```
[ INFO 825 ] FOTA: <msg>
```

This message is used to send informational non-secure fota messages.

- <msg>: arbitrary message string

17.2.0.102 Message Code: 830

```
[ INFO 830 ] TS|<uptime>|<time>|<mode>|<latitude>|<longitude>|<altitude>
```

This message is sent when the GNSS debug message is enabled (see cfgGnssDevDebugEnabled).

- <uptime>: Time in milliseconds since boot-up
- <time>: Time in (UTC) according to the ISO 8601 standard.
- <mode>: GNSS Navigation Mode.
  - Not available: Service is disabled.
  - None: No satellites found.
  - 2D: No full satellite lock available.
  - 3D: Full satellite lock.
- <latitude>: Latitude of the current location in decimal degrees (DD).
- <longitude>: Longitude of the current location in decimal degrees (DD).
- <altitude>: Altitude of the current location in meters.

If no GNSS signal is available, the message is: TS|<uptime>|No GNSS signal|-|-|-|-"

17.2.0.103 Message Code: 900

```
[ INFO 900 ] CARP: instance <carp instance> is now a BACKUP
```

This message is sent when a CARP instance became a backup.

- <carp instance>: CARP instance

17.2.0.104 Message Code: 901

```
[ INFO 901 ] CARP: instance <carp instance> is now a MASTER
```

This message is sent when a CARP instance became a master.

- <carp instance>: CARP instance

17.2.0.105 Message Code: 910

```
[ ERROR 910 ] CARP: The syncinterface <iface> does not have an unique IP
```

This message is sent when during carp-init a sync-interface is found to have no unique IP address.

- <iface>: interface name

17.2.0.106 Message Code: 911

```
[ ERROR 911 ] CARP: No valid ipaddr for the primary IP of the carp instance <
   carp instance> found
```

This message is sent when during carp-init a primary interface is found to have no valid IP address.

- <carp instance>: CARP instance

17.2.0.107 Message Code: 950

```
[ WARNING 950 ] RAPD/<interface>: Rogue AP '<mac>' detected for SSID '<ssid>'
```

This message is sent whenever RAPD found a rogue access-point during cyclic scans.

- <interface>: WLAN interface the RAPD is operating on
- <mac>: MAC address of rogue AP
- <ssid>: affected SSID

17.2.0.108 Message Code: 2710

```
[ INFO 2710 ] BRIDGE-MIB: newRoot (<bridge root>)
```

This message is sent when a new root-bridge is detected.

- <bridge root>: name of the new root bridge

17.2.0.109 Message Code: 2711

```
[ INFO 2711 ] BRIDGE-MIB: topologyChange (<bridge name>:<port number>:<old
   state>_<new state>)
```

This message is sent when a topology change was detected through RSTP.

- <bridge name>: name of the affected bridge
- <port number>: RSTP port number
- <old state>: state before topology change
- <new state>: state after topology change

Note that the provided states are implementation specific, e.g. in the current case of openvswitch providing the RSTP functionality they are used as enums:

- 0: INIT
- 1: INACTIVE_EXEC
- 2: INACTIVE
- 3: LEARNING_EXEC
- 4: LEARNING
- 5: DETECTED_EXEC
- 6: ACTIVE_EXEC
- 7: ACTIVE

17.2.0.110 Message Code: 3002

```
[ INFO 3002 ] Product information: <type>, <serial>
```

This message is sent when the device started up.

- <type>: Product type (see hwSysProduct).
- <serial>: Serial number of the product (see hwSysSerial).

17.3 Commissioning Log Messages

17.3.0.1 Message Code: 400

```
[ INFO 400 ] <iface>: Station is associated
```

This message is sent when association connection stage is reached. It is used to trigger led status. [STA only]

- <iface>: Wireless interface name

17.3.0.2 Message Code: 401

```
[ INFO 401 ] <iface>: Station is disassociated, Reason: <reason>
```

This message is sent when station deauthenticates and disassociates from an AP. It is used to trigger led status. [STA only]

- <iface>: Wireless interface name
- <reason>: see WLAN Reason Codes

17.3.0.3 Message Code: 402

```
[ INFO 402 ] <iface>: Station is authorized
```

This messages is sent when all three connection stages association, authentication and authorization have been completed successfully. It is used to trigger various daemons. [STA only]

- <iface>: Wireless interface name

17.3.0.4 Message Code: 403

```
[ NOTICE 403 ] WLAN: Authentication failure
```

This message is sent when there is an authentication timeout. An authentication management frame has been sent but an authentication response frame hasn't been received. [STA only]

17.3.0.5 Message Code: 404

```
[ NOTICE 404 ] <iface>: Association failure
```

This message is sent when there is an association timeout. An association management frame has been sent but an association response frame hasn't been received. [STA only]

- <iface>: Wireless interface name

17.3.0.6 Message Code: 406

```
[ NOTICE 406 ] WLAN: Max number of station exceeded
```

Maximum allowed associated station counter has been exceeded. [AP only]

17.3.0.7 Message Code: 408

```
[ INFO 408 ] WLAN: Failed to connect to <bssid>: <error_message>
```

Format: WLAN: Failed to connect to <bssid>: <error_message>. [STA only]

- <bssid>: BSSID (MAC) of the AP
- <error_message>: Error message as text

17.3.0.8 Message Code: 410

```
[ NOTICE 410 ] <iface>: Handoff:|<reason>|<count>|<prev_bssid>|<cur_bssid>|<
   ssid_mgmt>|<ho_time>|
```

This handoff notification message is sent for Handoff with profile t2gv1 (see cfgWlanHoProfile).

- <iface>: Wireless interface name (e.g 'wlan0')
- <reason>: Reason of last handoff, see WLAN Reason Codes
- <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>: Status of SSID management
  - 0: SSID Management is disabled
  - 1: Primary SSID
  - 2: Secondary SSID
- <ho_time>: Handoff time between disassociation and association in milliseconds

17.3.0.9 Message Code: 415

```
[ NOTICE 415 ] STA couldn't find better AP after <count> consecutive scans. [
   STA only]
```

This message is sent when no better AP can be discovered after several consecutive scans. [STA only]

- <count>: Number of consecutive scans

17.3.0.10 Message Code: 421

```
[ NOTICE 421 ] <iface>: CSA <old_freq> => <new_freq>
```

This message is sent as a channel switch announcement (CSA) to notify the change of frequency. [STA only]

- <iface>: Wireless interface name
- <old_freq>: Old selected frequency
- <new_freq>: New selected frequency

17.3.0.11 Message Code: 429

```
TS|<uptime>|<bssid>|RSSI_PLT|<rssi>
```

This message is sent when the debug message for Beacon RSSI is enabled (see cfgWlanDbgBeaconrssi). The Pilot (PLT) RSSI is reported for all wireless Pilot frames received by the radio. This can cause a high load of messages.

- <uptime>: Time in milliseconds since boot-up
- <bssid>: BSSID (MAC) of the counterpart
- <rssi>: Measured signal strength in RSSI

Note: This message is only available in Syslog and not sent as trap.

17.3.0.12 Message Code: 430

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

This message is sent when the debug message for Beacon RSSI is enabled (see cfgWlanDbgBeaconrssi). The Beacon (BCN) RSSI is reported for all wireless Beacon frames received by the radio. This can cause a high load of messages.

- <uptime>: Time in milliseconds since boot-up
- <bssid>: BSSID (MAC) of the counterpart
- <rssi>: Measured signal strength in RSSI

Note: This message is only available in Syslog and not sent as trap.

17.3.0.13 Message Code: 431

```
TS|<uptime>|<bssid>|RSSI_BCN|<raw_rssi>|<cur_filter_rssi>|<short_filter_rssi>|
   <long_filter_rssi>
```

This message is sent when the debug message for Filtered Beacon RSSI is enabled (see cfgWlanDbgBeaconfiltered). The Filtered Beacon (BCN) RSSI is reported only for the wireless Beacon frames from the AP the STA is connected to.

- <uptime>: Time in milliseconds since boot-up
- <bssid>: BSSID (MAC) of the counterpart
- <raw_rssi>: Unfiltered measured signal strength in RSSI
- <cur_filter_rssi>: Current filtered RSSI used. Depending on filter mode this field contains same value as either <short_filter_rssi> or <long_filter_rssi> (see Handoff Filters, used for handoff decision)

- <short_filter_rssi>: Short filtered RSSI
- <long_filter_rssi>: Long filtered RSSI

Note: This message is only available in Syslog and not sent as trap.

17.3.0.14 Message Code: 432

```
TS|<uptime>|<bssid>|RSSI_PRESP|<rssi>
```

This message is sent when the debug message for Beacon RSSI is enabled (see cfgWlanDbgBeaconrssi). The Probe Response (PRESP) RSSI is reported for all wireless Probe Response frames received by the radio.

- <uptime>: Time in milliseconds since boot-up
- <bssid>: BSSID (MAC) of the counterpart
- <rssi>: Measured signal strength in RSSI

Note: This message is only available in Syslog and not sent as trap.

17.3.0.15 Message Code: 433

```
[ INFO 433 ] WLAN: Low RSSI event: <rssi>
```

This message is sent when the Filtered RSSI from the current AP is below low RSSI event threshold. It is used to trigger various daemons. [STA only]

- <rssi>: Measured signal strength in RSSI

17.3.0.16 Message Code: 434

```
[ NOTICE 434 ] <iface>: Handoff:|<reason>|<count>|<prev_bssid>|<cur_bssid>|<
   ssid_mgmt>|<ho_time>|<offchan_scan_time>|
```

This handoff notification message is sent for Handoff with profile t2gv2 (see cfgWlanHoProfile).

- <iface>: Wireless interface name (e.g 'wlan0')
- <reason>: Reason of last handoff, see WLAN Reason Codes

- <count>: Number of handoffs since the last configuration
- <prev_bssid>: BSSID (MAC) of the previous associated AP
- <cur_bssid>: BSSID (MAC) of the current associated AP
- <ssid_mgmt>: Status of SSID management
  - 0: SSID Management is disabled
  - 1: Primary SSID
  - 2: Secondary SSID
- <ho_time>: Handoff time between disassociation and association in milliseconds
- <offchan_scan_time>: Handoff time used for offchannel scanning in milliseconds

18 WLAN Reason Codes

18.1 Wireless

This section lists the reason codes of wireless disconnections or handoff events in mobility applications. Most reported reason codes are proprietary, unhandled codes are covered by the IEEE wireless standard.

18.1.1 Proprietary Reason Codes

- 0 - No reason - No reason or unspecified (after restart)
- 2 - Low RSSI - STA local handoff decision: Received Beacon RSSI is below threshold
- 3 - Beacon miss - STA local handoff decision: Beacon miss (connection lost)
- 4 - Low Ack - STA local handoff decision: Consecutive ACK misses
- 5 - High RSSI - STA local handoff decision: Received Beacon RSSI is above threshold
- 6 - Near Distance - STA local handoff decision: Distance measurement result is below threshold
- 7 - Far Distance - STA local handoff decision: Distance measurement result is above threshold
- 8 - Redundancy loss - STA local handoff decision: Redundancy change on AP, own backbone lost
- 11 - De-auth - AP disconnect reason: De-auth to force a new 802.1X session
- 200 - De-auth startup - AP disconnect reason: startup complete, disconnect all prematurely connected clients
- 201 - De-auth down - AP disconnect reason: AP going down, e.g for reconfiguration

18.1.2 IEEE Reason Codes

Reason codes according to the wireless IEEE Std 802.11-2016, 9.4.1.7, Table 9-45.

The reported reason code is calculated as follows:

$$CODE = CODE_{IEEE} - 100 \tag{18.1}$$

- -99: Unspecific Reason
- -98: Previous authentication no longer valid - Client has associated but is not authorised.
- -97: Deauthenticated because sending STA is leaving (or has left) IBSS or ESS - The access point went offline, deauthenticating the client.
- -96: Disassociated due to inactivity - Client session timeout exceeded.
- -95: Disassociated because AP is unable to handle all currently associated STAs - The access point is busy, performing load balancing, for example.
- -94: Class 2 frame received from nonauthenticated STA - Client attempted to transfer data before it was authenticated.
- -93: Class 3 frame received from nonassociated STA - Client attempted to transfer data before it was associated.
- -92: Disassociated because sending STA is leaving (or has left) BSS - Operating System moved the client to another access point using non-aggressive load balancing.
- -91: STA requesting (re)association is not authenticated with responding STA - Client not authorized yet, still attempting to associate with an access point.
- -90: Disassociated because the information in the Power Capability element is unacceptable.
- -89: Disassociated because the information in the Supported Channels element is unacceptable.
- -88: Reserved - Not Used or Special Purpose.
- -87: Invalid information element.
- -86: Message integrity code (MIC) failure.
- -85: 4-Way Handshake timeout.
- -84: Group Key Handshake timeout.

- -83: Information element in 4-Way Handshake different from (Re)Association Request/Probe Response/Beacon frame.
- -82: Invalid group cipher or Association denied due to requesting STA not supporting all of the data rates in the BSSBasicRateSet parameter. The link speed requested by the client or AP is incompatible. (i,e. trying to operate N only speeds on a G AP)
- -81: Invalid pairwise cipher.
- -80: Invalid AKMP.
- -79: Unsupported RSN information element version.
- -78: Invalid RSN information element capabilities.
- -77: IEEE 802.1X authentication failed.
- -76: Cipher suite rejected because of the security policy.
- -68: Disassociated for unspecified, QoS-related reason Quality of Service has denied the action.
- -67: Disassociated because QoS AP lacks sufficient bandwidth for this QoS STA.
- -66: Disassociated because excessive number of frames need to be acknowledged, but are not acknowledged due to AP transmissions and/or poor channel conditions.
- -65: Disassociated because STA is transmitting outside the limits of its TXOPs.
- -64: Requested from peer STA as the STA is leaving the BSS (or resetting).
- -63: Requested from peer STA as it does not want to use the mechanism.
- -62: Requested from peer STA as the STA received frames using the mechanism for which a setup is required.
- -61: Requested from peer STA due to timeout.
- -60: Peer STA does not support the requested cipher suite.

19 CLI Commands

19.1 apply - Apply all pending configuration changes

```
USAGE: apply

The apply command applies all pending configuration changes. To show a list of
all pending changes use the changes command.

EXAMPLES:
  * Apply all pending changes:
            apply
```

19.2 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.

EXAMPLES:
  * Show a list of all pending changes:
            changes
```

19.3 configure - Manage the global configuration

```
USAGE: configure [COMMAND [ARGS...]]

The configuration tool provides access to the global configuration. It is
grouped into COMMANDs and their respective arguments ARGS. If no COMMAND is
specified, it defaults to the 'terminal' command.

COMMANDS:
    terminal
```

```
            Enter the configuration in the terminal.

EXAMPLES:
  * Enter the configuration:
            configure
```

19.4 date - Display the current system time

```
USAGE: date

The date program may be used to display the current system time.

EXAMPLES:
  * Display the current system time:
            date
```

19.5 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
```

19.6 factory - Reset the system to its factory settings

```
USAGE: factory

This will reset the whole configuration of the system, including administrator
password and certificates, to its factory settings.

EXAMPLES:
  * Reset the system to its factory settings:
            factory
```

19.7 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
```

19.8 gpsmon - GPS packet monitor and control utility

```
USAGE: gpsmon [-hVn] IP:PORT

gpsmon is a monitor that watches packets coming from a GPS and displays
them along with diagnostic information.

OPTIONS:
  -h        Show detailed options
  -V        Show version of gpsmon
  -n        Force gpsmon to request NMEA0183 packets instead of the raw
            data stream from gpsd.

EXAMPLES:
  * Bind to given ip:port
            gpsmon 0.0.0.0:2947
```

19.9 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
```

19.10 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
```

19.11 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.

EXAMLES:
  * 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
```

19.12 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
```

19.13 ipsec - Invoke IPsec utilities

```
USAGE: ipsec COMMAND [ARGUMENTS ...]
       ipsec --help

The ipsec tool invokes any of several utilities involved in controlling the
IPsec encryption/authentication system, running the specified command with the
```

```
specified arguments as if it had been invoked directly.

The most important COMMANDs are: start, stop, up, down, ...

OPTIONS:
  --help  Show a more detailed help text.

EXAMLES:
  * Show more detailed help of the starter utility:
            ipsec start --help
```

19.14 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 -
```

19.15 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
```

19.16 ovs-appctl - OpenVswitch Application Control

```
USAGE: ovs-appctl [OPTIONS]

Manage OpenVswitch Application.

OPTIONS:
  -h        Show a more detailed help text.
  lacp/show Show the status of LACP instances
  bond/show Show the status of bond interfaces

EXAMPLES:
  * Display the status of all LACP instances:
            ovs-appctl lacp/show
  * Display the status of all bond interfaces:
            ovs-appctl bond/show
```

19.17 ovs-dpctl - OpenVswitch DataPath Control

```
USAGE: ovs-dpctl [OPTIONS]

Manage OpenVswitch DataPath.

OPTIONS:
  -h         Show a more detailed help text.
  show       Display lookup stats and datapath ports.
  dump-flows Display dump of kernel datapath flows.

EXAMPLES:
  * Dump the currently active kernel datapath flows:
             ovs-dpctl dump-flows
```

19.18 ovs-ofctl - OpenVswitch OpenFlow Control

```
USAGE: ovs-ofctl [OPTIONS]

Manage OpenVswitch OpenFlow.

OPTIONS:
  -h         Show a more detailed help text.
  show       Display overview of OpenFlow ports of a bridge.
  dump-flows Display dump of OpenFlow flows on a bridge.

EXAMPLES:
  * Dump the currently active OpenFlow flows on bridge br0:
            ovs-ofctl dump-flows br0
  * Show the OpenFlow Ports of bridge br0:
            ovs-ofctl show br0
```

19.19 ovs-vsctl - OpenVswitch VirtualSwitch Control

```
USAGE: ovs-vsctl [OPTIONS]

Manage OpenVswitch VirtualSwitch.

OPTIONS:
  -h        Show a more detailed help text.
  show      Show an overview of all virtual switches and their ports.

EXAMPLES:
  * Display a list of all Virtual Switches and their ports:
            ovs-vsctl show
```

19.20 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
```

19.21 ps - Show current processes

```
USAGE: ps [OPTIONS]

Report a snapshot of the current processes.

OPTIONS:
  -h        Show a more detailed help text.

EXAMPLES:
  * Show a list of currently running processes:
            ps
```

19.22 qlog - Dump cellular log to TCP/FTP server

```
USAGE: qlog [start|stop|status] [-i] [ARGS]

This debug tool starts a connection to a TCP/FTP server,
depending on the configuration of ARGS.

On the host, either a TCP/FTP server must be running.

NOTE    It is recommended to avoid sending large amounts of data during
   recording.
        Also be aware that sensitive data may be recorded.

OPTIONS
  -i    Full module initialization (connection will be interrupted)
ARGS
  TCP:  <ip>:<port>
        Example: 192.168.1.2:9000
  FTP:  ftp:<ip>-user:<username>-pass:<password>
        Allowed characters for the user are [a-zA-Z0-9_]
        Allowed characters for the pass are [a-zA-Z0-9.:_%?+!-]
        The user and pass have an allowed length of 1-32 characters
        Example: ftp:192.168.1.2-user:myuser-pass:mypassword

Host
  TCP:  TCP server
        Example: netcat -l 9000 -k > qlog.qmdl
  FTP:  FTP server
        Example: FileZilla
```

19.23 reboot - Reboot the system

```
USAGE: reboot

The reboot program ensures that the system is shut down in a safe manner and
immediately restarts the entire system.

EXAMPLES:
  * Reboot the system:
            reboot
```

19.24 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
```

19.25 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
```

19.26 session-manager - List or destroy active CLI, WebInterface and WebAPI sessions

```
USAGE: session-manager <cmd>
Commands:
- list                  Lists active sessions
```

```
- destroy <session-id>  Destroys a session based on id
```

19.27 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
```

19.28 sim - SIM Tool for PIN and PUK

```
USAGE: sim [change|lock|unlock|puk|status] <slot> <PIN> <PUK> ...

This helper tool can be used to change the SIM card PIN or
to recover the SIM when it is in PUK status.

WARNING: Wrong SIM settings or in case of a failure, the remote
         access may be lost. Make sure you have alternative access
         to the device.
         It is recommended to disable all wwanX interfaces before proceeding.

NOTE:    Switching the SIM slot takes up to 15 seconds.

ARGS
  change    Change the SIM PIN of the given SIM slot 1 or 2.
            command: sim change [1|2] <CURRENT_PIN> <NEW_PIN>

  lock      Lock the SIM card with the current SIM PIN.
            The SIM PIN must be enabled and defined in the configuration.
            command: sim lock [1|2] <CURRENT_PIN>

  unlock    Unlock the SIM card with the current SIM PIN.
            command: sim unlock [1|2] <CURRENT_PIN>

  puk       Define a new SIM PIN when the SIM card is in PUK status.
            command: sim puk [1|2] <PUK1|PUK2> <NEW_PIN>
```

```
  status    Get the current SIM PIN status of the given SIM slot 1 or 2.
            command: sim status [1|2]
```

19.29 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
```

19.30 status - Show system status

```
USAGE: status [OPTIONS] VIEW

The status tool shows various system status views.

OPTIONS:
  -f        Change output format to (e.g. plain or json) the default
            format is plain text.
  -h        Show a more detailed help text.


EXAMPLES:
  * Show a list of available status views:
            status list
  * Show a summary of the system status:
            status summary
```

19.31 support - Display support information

```
USAGE: support [section]

The support program displays support information.
This information is the same as when downloading the support-file.

EXAMPLES:
```

```
  * Display support information:
            support
  * Optional:
            section
              Print only selected section:
                * general
                * cellular
                * config
                * el_label
                * kernel_log
                * logs
                * network
                * ntservices
                * phy_stats
                * vpn
                * wireless
```

19.32 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 icmp
  * Show all packets on eth0 except for port 22 (SSH):
            tcpdump -i eth0 -n not port 22
```

19.33 traceroute - Traceroute network hosts

```
USAGE: traceroute [OPTIONS] HOST
```

```
Trace the path through a network to a host.

OPTIONS:
  -h        Show a more detailed help text.

EXAMPLES:
  * Traces to path to 192.168.1.1
            traceroute 192.168.1.1
```

19.34 upgrade - Upgrade firmware with optional configuration file import

```
USAGE:
  upgrade URI://ADDRESS[:PORT]/FIRMWARE_FILE [URI://ADDRESS[:PORT]/CONFIG_FILE]

  To upgrade the firmware, a valid URI to a server must be defined.
  Supported protocols are TFTP, HTTP/HTTPS.
  The configuration is preserved.
  The upgrade command accepts as an optional argument the URI to a config file,
  which is imported after the firmware upgrade process.

RESULTS:
   0: No Result
      Firmware upgrade succeeded, the device is rebooting.
  -1: Download Error
      Indicates an error during the download of the firmware or config file.
  -2: Flash Error
      Indicates an error during the validation of the firmware or config file.

EXAMPLES:
  * Firmware upgrade
      upgrade tftp://192.168.1.1/firmware.img
      upgrade http://192.168.1.1/firmware.img
      upgrade https://192.168.1.1/subfolder/firmware.img
      upgrade https://192.168.1.1:8080/firmware.img

  * Firmware upgrade with configuration file import
      upgrade http://192.168.1.1/firmware.img http://192.168.1.1/config.cfg
      upgrade tftp://192.168.1.1/firmware.img http://192.168.100.1/config.cfg

NOTE:
  If upgrading with an configuration file, please make sure that the config
  file is tested and compatible with the firmware version to be installed.
  Validation of the configuration file takes place after the firmware
  upgrade process.
```

```
  If you use the HTTPS protocol it is highly recommended
  to install the TLS Client CA Certificate on the device.
  Otherwise the device will connect to any web server which uses TLS,
  but the connection must be considered insecure.
  See ‘setTlsClient‘ for more information.
```

19.35 upgrade-cellular - Upgrade cellular firmware via FOTA (Firmware Over The Air)

```
USAGE:
  upgrade-cellular start

RESULTS:
   0: No Result
      Firmware upgrade succeeded, the device is rebooting.
   1: Error
      Indicates any error during the download/upgrade of the firmware.

Use ‘status summary‘ to check if a new cellular firmware is available.
```

19.36 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
```

19.37 wg - Wireguard management tool

```
USAGE: wg

The wireguard command allows to display the status of wireguard and generate
keys.

EXAMPLES:
  * Show the current wireguard configuration
            wg
  * Generate a wireguard private key
            wg genkey
  * Generate a wireguard private shared key
            wg genpsk
```

20 Deterministic Interface Index

In some cases it is very useful to have deterministic interface indexes (ifindex), especially when working with SNMP IF-MIB. In the following the interfaces with deterministic interface indexes are listed.

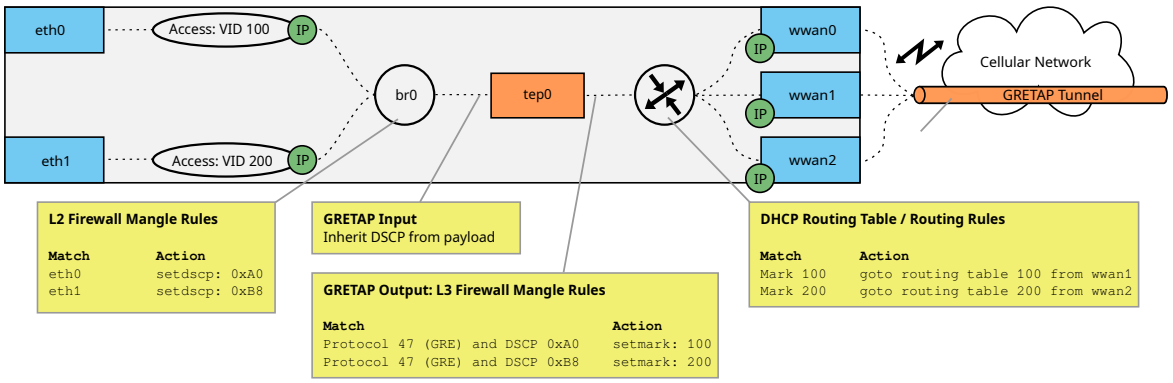
Interface	iface Prefix	ifindex Range	Examples (iface=ifindex)
Loopback	lo	1	lo=1
Ethernet	eth	100 - 199	eth0=100, eth1=101, eth2=102, etc.
Wireless LAN	wlan	200 - 1700	wlan0=200, wlan1=300, wlan2=400, etc.
Wireless LAN (4addr clients)	wlan.sta	200 - 1799	wlan0.sta1=201, wlan0.sta2=202, etc. wlan1.sta1=301, wlan1.sta2=302, etc.
Radio	wifi	1800 - 1899	wifi0=1800, wifi1=1801, etc.
Wireless WAN	wwan	1900 - 1999	wwan0=1900, wwan1=1901, etc.
OpenVPN	ovpn	2000 - 2099	ovpn0=2000, ovpn1=2001, etc.
IPSec	ipsec	2100 - 2199	ipsec0=2100, ipsec1=2101, etc.
Wireguard	wg	2200 - 2299	wg0=2200, wg1=2201, etc.
Tunnel Endpoint	tep	2300 - 2399	tep0=2300, tep1=2301, etc.
Cellular	cellular	2400 - 2499	cellular0=2400, cellular1=2401, etc.

Table 20.1: *Deterministic Interface Index (ifindex)*

21 Application Notes

21.1 L2 Tunnel Over Multiple EPS Bearer

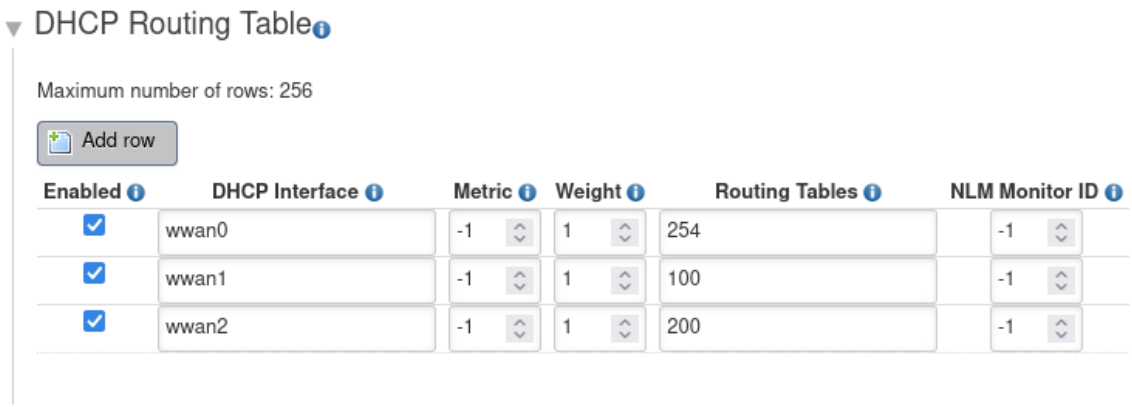
When using multiple default EPS bearers, it may be desirable to have a single L2 tunnel (e.g GRETAP or VXLAN) that is routed via a different wwan based on the type of traffic.



To implement such a configuration, multiple parts are involved:

- Routing Table per wwan (DHCP client) - cfgRouteDhcpTable
- L2 Mangling - cfgFwL2MangleTable
- L2 Tunnel - cfgNetTunnelEndPointTable and cfgVpnTunnelEndPoint
- L3 Mangling (set Mark) - cfgFwMangleTable
- Policy Routing on Mark - cfgRouteRuleTable

21.1.1 Routing Table per wwan



Each entry in cfgRouteDhcpTable may be used to create routes received from a DHCP client on another routing table than the default 254. The below configuration excample, creates the default gateway received on wwan0 on the routing table 254, and the default gateway received from wwan1 and wwan2 on the routing table 100 respective 200.

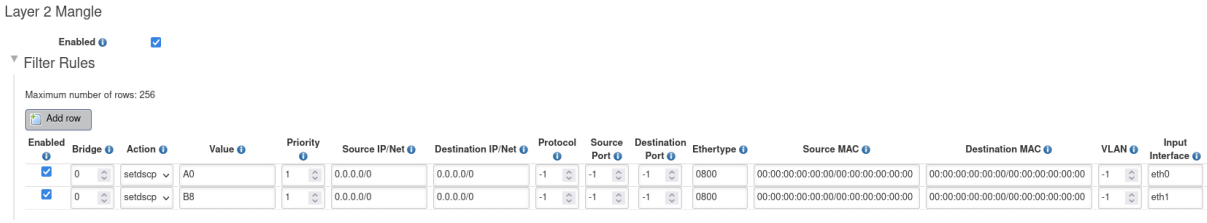
```
WESTERMO-SW6-MIB::cfgRouteDhcpEnabled.0 = 1
WESTERMO-SW6-MIB::cfgRouteDhcpInterface.0 = wwan0
WESTERMO-SW6-MIB::cfgRouteDhcpRoutingTables.0 = 254

WESTERMO-SW6-MIB::cfgRouteDhcpEnabled.1 = 1
WESTERMO-SW6-MIB::cfgRouteDhcpInterface.1 = wwan1
WESTERMO-SW6-MIB::cfgRouteDhcpRoutingTables.0 = 100

WESTERMO-SW6-MIB::cfgRouteDhcpEnabled.2 = 1
WESTERMO-SW6-MIB::cfgRouteDhcpInterface.2 = wwan2
WESTERMO-SW6-MIB::cfgRouteDhcpRoutingTables.2 = 200
```

Listing 21.1: *dhcp routing table*

21.1.2 L2 Mangling



The L2 Mangle Table may be used to match frames and modify the content of their headers. In this example we match on IP frames that are received on the interfaces eth0 and eth1 to set the DSCP field in their IP header to 0xA0 (CS5) respective 0xB8 (EF).

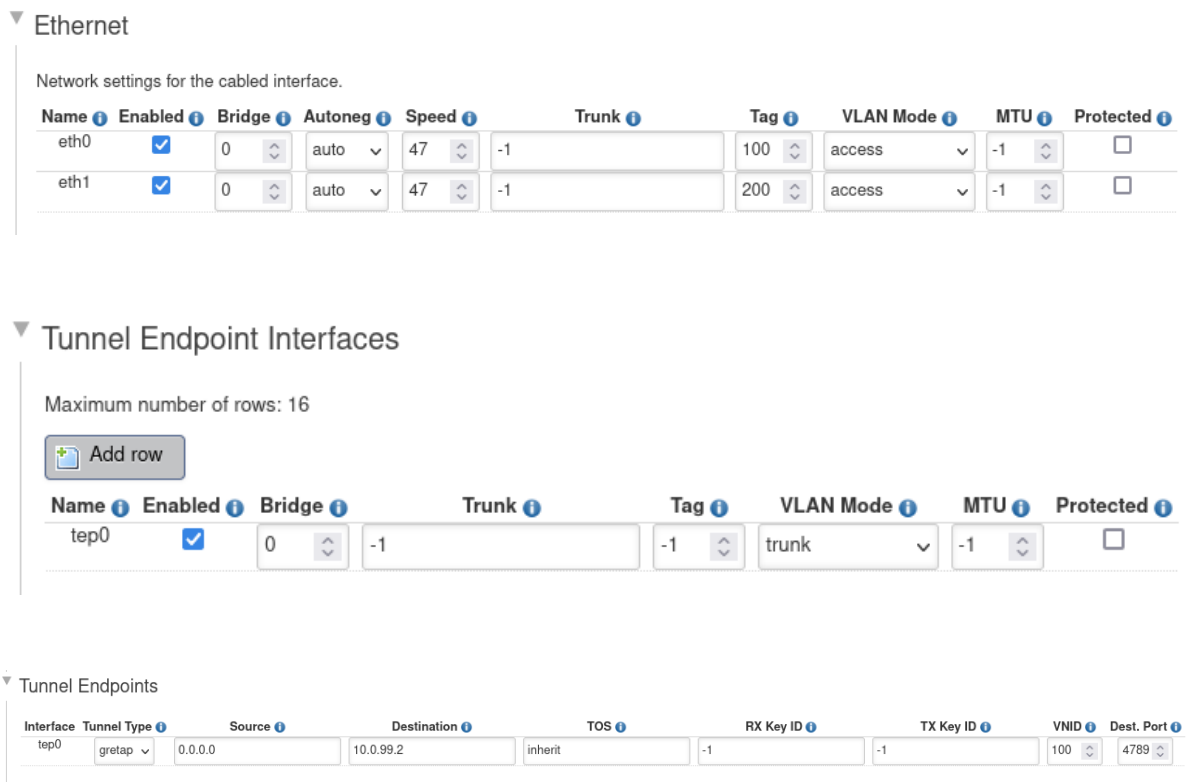
```
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2MangleEnabled.0 = 1

WESTERMO-SW6-FIREWALL-MIB::cfgFwL2MnglEnabled.0 = 1
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2MnglAction.0 = 1
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2MnglValue.0 = A0
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2MnglInputInterface.0 = eth0

WESTERMO-SW6-FIREWALL-MIB::cfgFwL2MnglEnabled.1 = 1
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2MnglAction.1 = 1
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2MnglValue.1 = B8
WESTERMO-SW6-FIREWALL-MIB::cfgFwL2MnglInputInterface.1 = eth1
```

Listing 21.2: *l2 mangle table*

21.1.3 L2 Tunnel



There are multiple L2 tunneling protocols available. This example uses GRETAP. The Tunnel-End-Point (tep0) is added to the bridge 0 of which eth0 and eth1 are also members. The ETH interfaces are access-ports to vlan 100 respective 200. The parameter cfgVpnTepTos is set to ‘inherit‘ resulting in the encapsulating GRE frames inheriting the TOS of their payload that has been set in the L2 Mangle Table to 0xA0 and 0xB8.

```
WESTERMO-SW6-MIB::cfgNetEthVlanMode.0 = 1
WESTERMO-SW6-MIB::cfgNetEthTag.0 = 100
WESTERMO-SW6-MIB::cfgNetEthVlanMode.1 = 1
WESTERMO-SW6-MIB::cfgNetEthTag.0 = 200

WESTERMO-SW6-MIB::cfgNetTepEnabled.0 = 1
WESTERMO-SW6-MIB::cfgNetTepBridge.0 = 0
WESTERMO-SW6-MIB::cfgVpnTepTunnelType.0 = 1
WESTERMO-SW6-MIB::cfgVpnTepDestination.0 = 10.0.99.2
```

Listing 21.3: *l2 tep*

21.1.4 L3 Mangling

The GRE frames that are sent by the Tunnel-End-Point have configured a destination of 10.0.99.2 and as DSCP the value as inherited from the payload. For the routing process to be able to match on these frames, they need to be marked. Two rules are created that match on the destination 10.0.99.2, protocol 47 (GRE) and the DSCP 0xA0 respective 0xB8. Matched frames are marked with the value 100 respective 200. Frames that do not match the configured criteria are not marked.

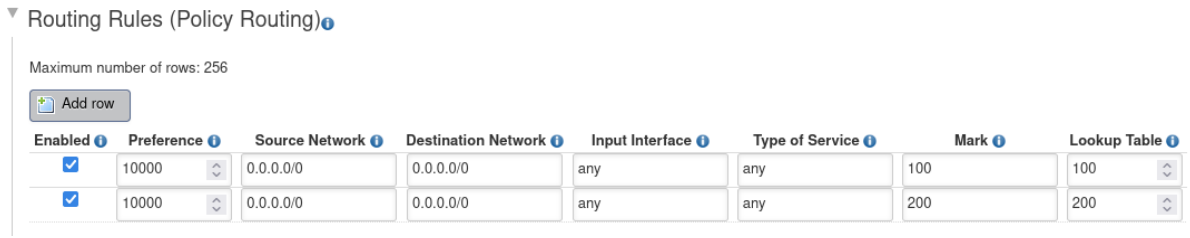
```
WESTERMO-SW6-FIREWALL-MIB::cfgFwEnabled.0 = 1

WESTERMO-SW6-FIREWALL-MIB::cfgFwMngIEnabled.0 = 1
WESTERMO-SW6-FIREWALL-MIB::cfgFwMngIChain.0 = 3
WESTERMO-SW6-FIREWALL-MIB::cfgFwMngIAction.0 = 4
WESTERMO-SW6-FIREWALL-MIB::cfgFwMngIDestinationAddress.0 = 10.0.99.2/32
WESTERMO-SW6-FIREWALL-MIB::cfgFwMngIProtocol.0 = 47
WESTERMO-SW6-FIREWALL-MIB::cfgFwMngIDscp.0 = A0
WESTERMO-SW6-FIREWALL-MIB::cfgFwMngIValue.0 = 100

WESTERMO-SW6-FIREWALL-MIB::cfgFwMngIEnabled.1 = 1
WESTERMO-SW6-FIREWALL-MIB::cfgFwMngIChain.1 = 3
WESTERMO-SW6-FIREWALL-MIB::cfgFwMngIAction.1 = 4
WESTERMO-SW6-FIREWALL-MIB::cfgFwMngIDestinationAddress.1 = 10.0.99.2/32
WESTERMO-SW6-FIREWALL-MIB::cfgFwMngIProtocol.1 = 47
WESTERMO-SW6-FIREWALL-MIB::cfgFwMngIDscp.0 = B8
WESTERMO-SW6-FIREWALL-MIB::cfgFwMngIValue.1 = 200
```

Listing 21.4: *l3 mangle table*

21.1.5 Policy Routing on Mark



Routing rules have to be installed that match on the mark set in the L3 Mangle Table. Since the first step in this example directs the default gateway received by wwan1 and wwan2 to the routing tables 100 and 200, traffic matching the mark 100 and 200 are sent to their respective routing table 100 and 200. Unmarked traffic will never be sent to these tables and always ends up on the default routing table 254.

```
WESTERMO-SW6-MIB::cfgRouteRuleEnabled.0 = 1
WESTERMO-SW6-MIB::cfgRouteRuleMark.0 = 100
WESTERMO-SW6-MIB::cfgRouteRuleLookupTable.1 = 100

WESTERMO-SW6-MIB::cfgRouteRuleEnabled.0 = 1
WESTERMO-SW6-MIB::cfgRouteRuleMark.0 = 200
WESTERMO-SW6-MIB::cfgRouteRuleLookupTable.0 = 200
```

Listing 21.5: *policy routing*

21.1.6 Remote configuration

This example outlines the configured of a cellular client. A similar configuration has to be done on the remote end, which is outside the context of this documentation.