

NetModule Router NB3800

User Manual for Software Version 4.1



Manual Version 1.10

NetModule AG, Switzerland

December 14, 2020



NetModule Router NB3800

This manual covers all variants of the *NB3800* product type.

The specifications and information regarding the products in this manual are subject to change without notice. We would like to point out that NetModule makes no representation or warranties with respect to the contents herein and shall not be responsible for any loss or damage caused to the user by the direct or indirect use of this information. This document may contain information about third party products or processes. Such third party information is generally out of influence of NetModule and therefore NetModule shall not be responsible for the correctness or legitimacy of this information. Users must take full responsibility for their application of any products.

Copyright ©2020 NetModule AG, Switzerland All rights reserved

This document contains proprietary information of NetModule. No parts of the work described herein may be reproduced. Reverse engineering of the hardware or software is prohibited and protected by patent law. This material or any portion of it may not be copied in any form or by any means, stored in a retrieval system, adopted or transmitted in any form or by any means (electronic, mechanical, photographic, graphic, optic or otherwise), or translated in any language or computer language without the prior written permission of NetModule.

A large amount of the source code to this product is available under licenses which are both free and open source. Most of it is covered by the GNU General Public License which can be obtained from www.gnu.org. The remainder of the open source software which is not under the GPL, is usually available under one of a variety of more permissive licenses. A detailed license information for a particular software package can be provided on request.

All other products or company names mentioned herein are used for identification purposes only and may be trademarks or registered trademarks of their respective owners. The following description of software, hardware or process of NetModule or other third party provider may be included with your product and will be subject to the software, hardware or other license agreements.

Contact

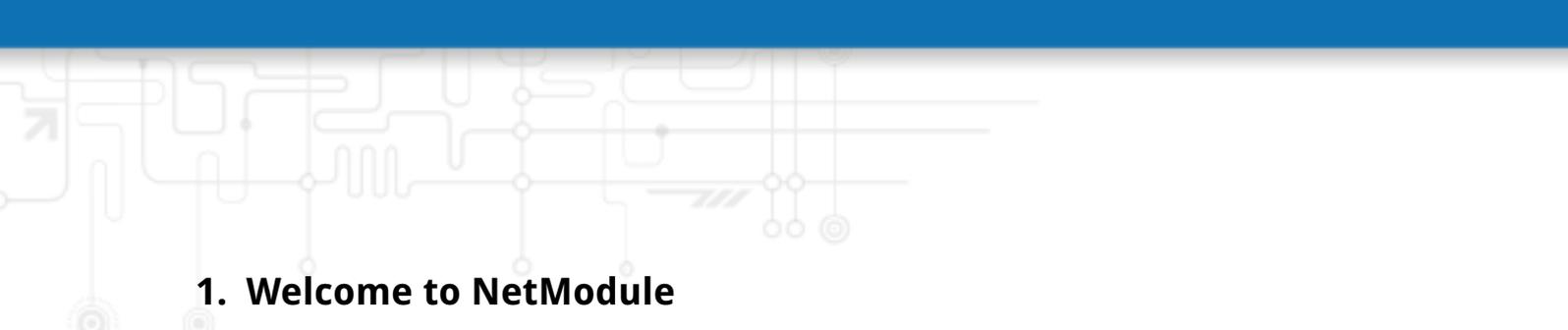
www.netmodule.com/support

NetModule AG	Tel +41 31 985 25 10
Maulbeerstrasse 10	Fax +41 31 985 25 11
CH-3011 Bern	info@netmodule.com
Switzerland	http://www.netmodule.com

5.9.	LOGOUT	172
6.	Command Line Interface	173
6.1.	General Usage	173
6.2.	Print Help	174
6.3.	Getting Config Parameters	175
6.4.	Setting Config Parameters	175
6.5.	Checking Config Completed	175
6.6.	Getting Status Information	175
6.7.	Scanning Networks	176
6.8.	Sending E-Mail or SMS	176
6.9.	Updating System Facilities	177
6.10.	Manage keys and certificates	177
6.11.	Restarting Services	178
6.12.	Debug System	178
6.13.	Resetting System	179
6.14.	Rebooting System	179
6.15.	Running Shell Commands	179
6.16.	Working with History	180
6.17.	CLI-PHP	180
A.	Appendix	185
A.1.	Abbreviations	185
A.2.	System Events	187
A.3.	Factory Configuration	189
A.4.	SNMP VENDOR MIB	190
A.5.	SDK Examples	209

5.45. SNMP Agent	133
5.46. Web Server	137
5.47. VRRP Configuration	139
5.48. Voice Gateway Administration	141
5.49. Voice Gateway Endpoint Configuration	142
5.50. Voice Gateway Routing Configuration	145
5.51. System	147
5.52. Regional settings	150
5.53. User Accounts	153
5.54. Remote Authentication	154
5.55. Manual File Configuration	159
5.56. Automatic File Configuration	160
5.57. Factory Configuration	161
5.58. Log Viewer	163
5.59. Tech Support File	164
5.60. Keys and certificates	165
5.61. Certificate Configuration	167
5.62. Licensing	170

A.3. SDK Examples 211



1. Welcome to NetModule

Thank you for purchasing a NetModule Router. This document should give you an introduction to the router and its features. The following chapters describe any aspects of commissioning the device, installation procedure and provide helpful information towards configuration and maintenance.

Please find further information such as sample SDK script or configuration samples in our wiki on <http://wiki.netmodule.com>.

2. Conformity

This chapter provides general information for putting the router into operation.

2.1. Safety Instructions

Please carefully observe all safety instructions in the manual that are marked with the symbol .



Compliance information: The NetModule routers must be used in compliance with any and all applicable national and international laws and with any special restrictions regulating the utilization of the communication module in prescribed applications and environments.



Information about the accessories / changes to the device:

- Please only use original accessories to prevent injuries and health risks.
- Changes made to the device or the use of non-authorized accessories will render the warranty null and void and potentially invalidate the operating license.
- NetModule routers must not be opened (SIM cards may be used according to the instructions).



Information about the device interfaces:

- All systems that are connected to the NetModule router interfaces must meet the requirements for SELV (Safety Extra Low Voltage) systems.
- Interconnections must not leave the building nor penetrate the body shell of a vehicle.
- Connections for antennas may only exit the building or the vehicle hull if transient overvoltages (according to IEC 62368-1) are limited by external protection circuits down to 1 500 V_{peak}. All other connections must remain within the building or the vehicle hull.
- Always keep a distance of more than 40 cm from the antenna in order to reduce exposure to electromagnetic fields below the legal limits.
- Devices with a WLAN interface may be operated only with applicable Regulatory Domain configured. Special attention must be paid to country, number of antennas and the antenna gain (see also chapter 5.3.4). The maximum allowed gain is 3dBi in the relevant frequency range. WLAN antennas with a higher amplification may be used with the NetModule router "Enhanced-RF-Configuration" software license and the antenna gain and cable attenuation that have been correctly configured by certified specialized personnel. A misconfiguration will lead to loss of the approval.
- Cellular antennas attached to the router must have an antenna gain of equal or less than 2.5 dBi. The user is responsible for the compliance with the legal regulations.
- Only CE-compliant power supplies with a current-limited SELV output voltage range may be used with the NetModule routers.¹

¹**Note:** Power supplies for routers with the Pb option (72-110 V_{DC}) cannot be a SELV circuit, since the voltage is greater than 60 V_{DC}.



General safety instructions:

- Observe the usage limitations of radio units at filling stations, in chemical plants, in systems with explosives or potentially explosive locations.
- The devices may not be used in airplanes.
- Exercise particular caution near personal medical aids, such as pacemakers and hearing aids.
- The NetModule routers may also cause interference in the nearer distance of TV sets, radio receivers and personal computers.
- Never perform work on the antenna system during a thunderstorm.
- The devices are generally designed for normal indoor use. Do not expose the devices to extraordinary environmental conditions worse than IP40.
- Protect them against aggressive chemical atmospheres and humidity or temperatures outside specifications.
- We highly recommended creating a copy of a working system configuration. It can be easily applied to a newer software release afterwards.

2.2. Declaration of Conformity



NetModule hereby declares that under our own responsibility that the routers comply with the relevant standards following the provisions of the *RED Directive 2014/53/EU*. The signed version of the *Declaration of Conformity* can be obtained from <http://www.netmodule.com/downloads>

2.3. Waste Disposal



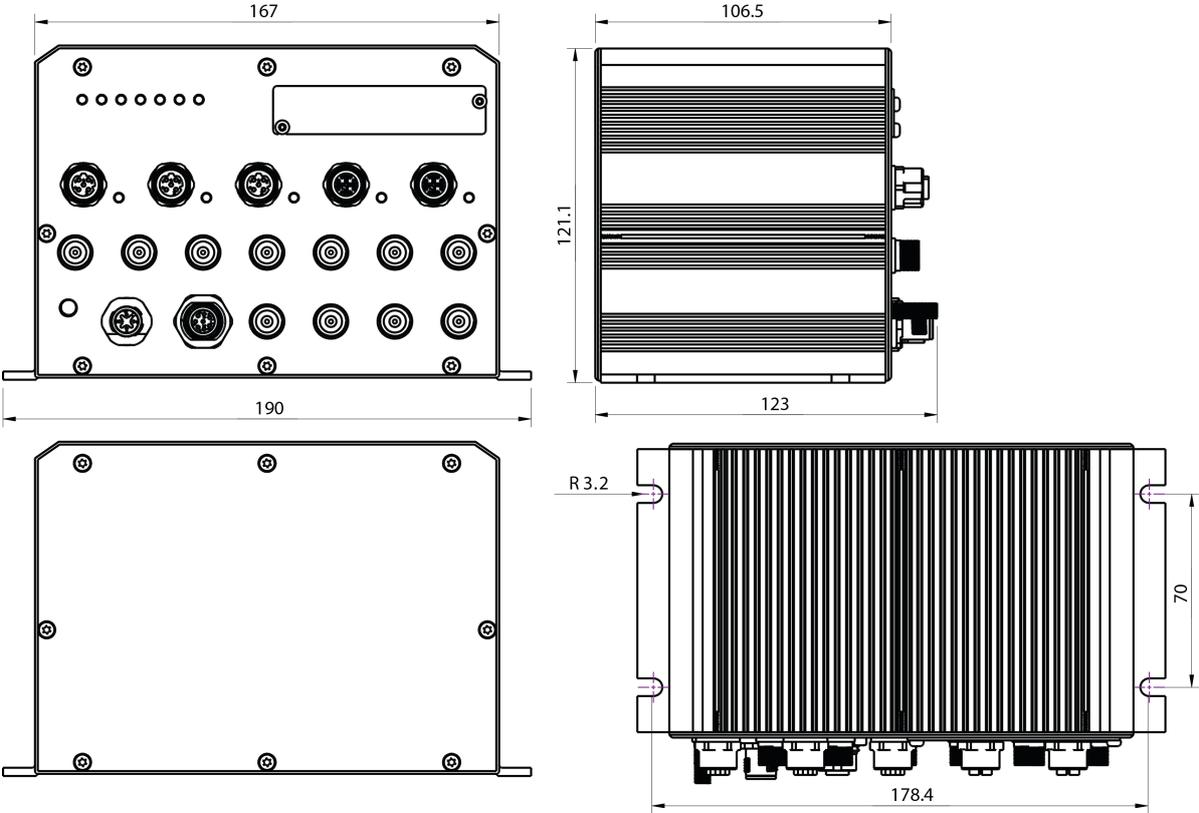
In accordance with the requirements of the *Council Directive 2012/19/EU* regarding Waste Electrical and Electronic Equipment (WEEE), you are urged to ensure that this product will be segregated from other waste at end-of-life and delivered to the WEEE collection system in your country for proper recycling.

2.4. National Restrictions

This product may be generally used in all EU countries (and other countries following the *RED Directive 2014/53/EU*) without any limitation. Please refer to our WLAN Regulatory Database for getting further national radio interface regulations and requirements for a particular country.

3. Specifications

3.1. Appearance



3.2. Features

All models of NB3800 have the following standard functionalities:

- Galvanically isolated power supply
- 2 Gbit Ethernet ports (M12, x-coded)
- 3 Fast Ethernet ports (M12, d-coded)
- 1 USB 2.0 host port
- 4 mini SIM card slots
- Extension port

The NB3800 can be equipped with the following options:

- LTE
- WLAN IEEE 802.11ac
- GPS/GNSS
- GSM-R
- RS-232
- RS-485
- IBIS
- CAN
- Audio
- 1 TB internal storage
- Software Key Virtualization
- Software Key Server
- Software Key Voice

Due to its modular approach, the NB3800 router and its hardware components can be arbitrarily assembled according to its indented usage or application. Please contact us in case of special project requirements.

3.3. Environmental Conditions

Parameter	Rating
Input Voltage (Variant Pa)	24 V _{DC} to 48 V _{DC} (−30% / +30%)
Input Voltage (Variant Pb)	72 V _{DC} to 110 V _{DC} (−30% / +30%)
Operating Temperature Range	24-48 V _{DC} : EN50155 TX (−40 °C to +70 °C) with max. 4 radio modules 24-48 V _{DC} : EN50155 T1 (−25 °C to +55 °C) with max. 6 radio modules 72-110 V _{DC} : EN50155 TX (−40 °C to +70 °C) with max. 4 radio modules 72-110 V _{DC} : EN50155 T2 (−40 °C to +55 °C) with max. 5 radio modules
Storage Temperature Range	−40 °C to +85 °C
Humidity	0 to 95% (non-condensing)
Altitude (Variant Pa)	up to 4000m
Altitude (Variant Pb)	up to 2000m
Over-Voltage Category	I
Pollution Degree	2
Ingress Protection Rating	IP40 (with SIM and USB covers mounted)

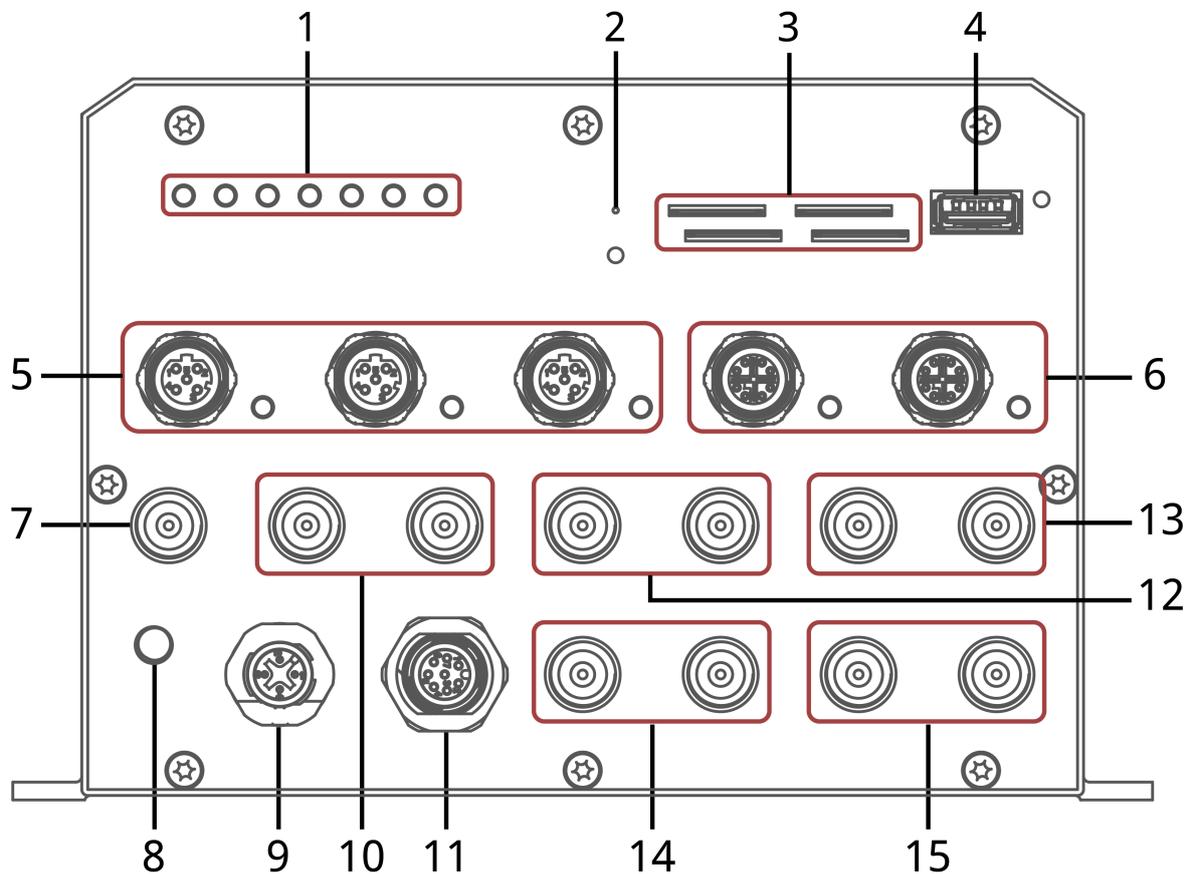
Table 3.1.: Environmental Conditions



Attention: When using the Pb variant with an input voltage higher than 60 V_{DC}, the router **MUST** be connected to an earth protection.

3.4. Interfaces

3.4.1. Overview



Nr.	Label	Function
1	LED Indicators	LED Indicators for the different interfaces
2	Reset	Reboot and factory reset button
3	SIM 1-4	SIM 1-4, they can be assigned dynamically to any modem by configuration.
4	USB	USB 2.0 host port, can be used for software/configuration updates.
5	ETH 1-3	FastEthernet switch ports, can be used as LAN or WAN interface.
6	ETH 4-5	Gigabit Ethernet switch ports, can be used as LAN or WAN interface.
7	GNSS	TNC female connector for GPS/GLONASS antenna

GNSS (Option Ge)

The GNSS module supports Dead Reckoning with onboard 3D accelerometer and 3D gyroscope.

Feature	Specification
Systems	GPS/GLONASS/BeiDu/Galileo ready
Data stream	NMEA or UBX
Tracking sensitivity	-160 dBm
Channels	72
Accuracy	2m
Supported antennas	Active and passive

Table 3.10.: GNSS Specifications option Ge

The GNSS antenna port have the following specification:

Feature	Specification
Max. allowed cable length	30 m
Max. allowed antenna gain	3.0 dBi
Min. distance between collocated radio transmitter antennas (Example: WLAN1 to MOB1)	20 cm
Connector type	TNC

Table 3.11.: GNSS / GPS Antenna Port Specification

3.4.7. USB 2.0 Host Port

The USB 2.0 host port has the following specification:

Feature	Specification
Speed	Low, Full & Hi-Speed
Current	max. 500 mA
Max. cable length	3 m
Cable shield	mandatory
Connector type	Type A

Table 3.12.: USB 2.0 Host Port Specification

Pin Assignment on M12, 8 poles, X-coded female

Pin	Signal	Pinning
1	M1+ / DA+	
2	M1- / DA-	
3	M0+ / DB+	
4	M0- / DB-	
5	M2+ / DD+	
6	M2- / DD-	
7	M3- / DC-	
8	M3+ / DC+	

Table 3.15.: Pin Assignments of 8 Poles Ethernet Connectors

3.4.9. Power Supply

Standard variant Pa (24 V_{DC} to 48 V_{DC})

The power input has the following specifications:

Feature	Specification
Power supply nominal voltages	24 V _{DC} , 36 V _{DC} and 48 V _{DC} (according to EN 50155)
Voltage range	24 V _{DC} to 48 V _{DC} (-30% / +30%)
Max. power consumption	25 W
Typ. Inrush-Current-Integral	0.23 A ² s at 24 V _{in} 0.57 A ² s at 36 V _{in} 1.05 A ² s at 48 V _{in}
Max. cable length	30m
Cable shield	not required
Galvanic isolation	yes, 1500 V _{DC} (according to EN 50155 & EN 62368-1)
Power interruption	Class S2: Sustains power interruptions up to 10 ms, there are no batteries included
Supply change over	Class C1: 0.6 U _n during 100 ms (w/o interruption)
Connector type	M12, 4 poles, A-coded male

Table 3.16.: Power Input Specifications Variant Pa

Audio Port Specification (Option A)

The Audio port has the following specification:

Feature	Specification
Protocol	Audio Line In/Out
Input impedance	44 k Ω , signal level 2 V _{pp}
Input bandwidth	100 Hz- 15 kHz
Input galvanic isolation to enclosure	functional (max. 250 V _{DC})
Output impedance	100 Ω , signal level 2 V _{pp}
Output bandwidth	300 Hz- 4 kHz
Output galvanic isolation to enclosure	functional (max. 250 V _{DC})
Max. cable length	30 m
Cable shield	mandatory

Table 3.20.: Audio Port Specification

EP Pins	Signal
1 / 5	Input Channel +
2 / 6	Input Channel –
3 / 7	Output Channel –
4 / 8	Output Channel +

Table 3.21.: Pin Assignments of Audio Port Signals (EP1 / EP2)

CAN Port Specification (Option C)

The CAN port has the following specification:

Feature	Specification
Protocol	CAN V2.0B
Speed	Up to 1 Mbit/s Default: 125 kbit/s
Galvanic isolation to enclosure	1500 V _{DC}
Internal bus termination	none
External bus termination ²	120 Ω
Max. cable length	100 m
Cable shield	mandatory
Cable type	twisted pair
Max. number of nodes	110
Reactionless	Option Cb: CAN-Passive (monitoring only) Option Ca: CAN-Active (rx and tx enabled)

Table 3.22.: CAN Port Specification

EP Pins	Signal
1 / 5	GND
2 / 6	L
3 / 7	H
4 / 8	—

Table 3.23.: Pin Assignments of CAN Port Signals (EP1 / EP2)

²**Note:** On each end of the CAN bus is a 120 Ω termination mandatory

Parameter	IP Pass-Through Settings
Interface	Specifies the interface on which the address shall be passed-through
WAN network	Specifies the WAN network
WAN netmask	Specifies the WAN netmask

Parameter	Supervision Settings
Retry interval	The interval in seconds at which pings are re-transmitted in case a first ping failed
Max. number of failed trials	The maximum number of failed ping trials until the link will be declared as down
Emergency action	The emergency action which should be taken after a maximum downtime has been reached. Using <code>reboot</code> would perform a reboot of the system, <code>restart link services</code> will restart all link-related applications including a reset of the modem.

Under some circumstances (e.g. in case the modem flaps between base stations) it might be necessary to set a specific service type or assign a fixed operator. The list of operators around can be obtained by initiating a network scan (may take up to 60 seconds). Further details can be retrieved by querying the modem directly, a set of suitable commands can be provided on request.

Configuration

A SIM card is generally assigned to a default modem but might be switched, for instance if you set up two WWAN interfaces with one modem but different SIM cards.

Close attention has to be paid when other services (such as SMS or Voice) are operating on that modem, as a SIM switch will naturally affect their operation.

The following settings can be applied:

Parameter	WWAN SIM Configuration
PIN code	The PIN code for unlocking the SIM card
PUK code	The PUK code for unlocking the SIM card (optional)
Default modem	The default modem assigned to this SIM card
Preferred service	The preferred service to be used with this SIM card. Remember that the link manager might change this in case of different settings. The default is to use <code>automatic</code> , in areas with interfering base stations you can force a specific type (e.g. 3G-only) in order to prevent any flapping between the stations around.
Registration mode	The desired registration mode
Network selection	Defines which network shall be selected. This can be bound to a specific LAI which can be retrieved by running a network scan.

Standard	Frequencies	Bandwidth	Data Rate
802.11g	2.4 GHz	20 MHz	54 Mbit/s
802.11n	2.4/5 GHz	20/40 MHz	300 Mbit/s
802.11ac ²	5 GHz	20/40/80 MHz	866.7 Mbit/s

Table 5.20.: IEEE 802.11 Network Standards

Note: NetModule Routers with 802.11n and 802.11ac support 2x2 MIMO

²Only available for NetModule Routers NB2800, NB3701, NB3711 and NB3800

Parameter	WLAN Client Background Scan Parameters
Long interval	The time in seconds when a background scan should be performed if the threshold is above the given threshold value
Short interval	The time in seconds when a background scan should be performed if the threshold is below the given threshold value

USB Devices

This page show the currently connected devices and it can be used to enable a specific device based on its Vendor and Product ID. Only enabled devices will be recognized by the system and raise additional ports and interfaces.

Figure 5.16.: USB Device Management

Parameter	USB Devices
Vendor ID	The USB Vendor ID of the device
Product ID	The USB Product ID of the device
Module	The USB module and type of driver to be applied for this device

Any ID must be specified in hexadecimal notation, wildcards are supported (e.g. AB[0-1][2-3] or AB*) A USB network device will be referenced as LAN10.

USB Autorun

This feature can be used to automatically launch a shell script or perform a software/config update as soon as an USB storage stick has been plugged in. For authentication, a file called `autorun.key` must exist in the root directory of a FAT16/32 formatted stick. It can be downloaded from that page and holds the SHA256 hash key of the admin password. The file can hold multiple hashes which will be processed line-by-line during authentication which can be used for setting up more systems with different admin passwords.

For new devices with an empty password the hash key

```
e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855
```

can be used.

The hash keys can be generated by running the command `echo -n "<admin-password>" | sha256sum` on a Linux system or an Internet hash key generator (search for "sha-256 hash calculator").

Once authentication has succeeded, the system scans for other files in the root directory which can perform the following actions:

1. For running a script: `autorun.sh`
2. For a configuration update: `cfg-<SERIALNO>.zip` (e.g. `cfg-00112B000815.zip`), or if not available `cfg.zip`
3. For a software update: `sw-update.img`

5.3.7. Serial Port

This page can be used to manage your serial ports. A serial port can be used by:

Parameter	Serial Port Usage
none	The serial port is not used
login console	The serial port is used to open a console which can be accessed with a serial terminal client from the other side. It will provide helpful bootup and kernel messages and spawns a login shell, so that users can login to the system.
device server	The serial port will be exposed over a TCP/IP port and can be used to implement a Serial/IP gateway.
modem bridge	Bridges the Serial Interface to the Modem TTY of an intergrated WWAN Modem.
modem emulator	Emulates a classical AT command driven modem on the serial interface. See http://wiki.netmodule.com/app-notes/hayes-modem-at-simulator for detailed information.
SDK	The serial port will be reserved for SDK scripts.

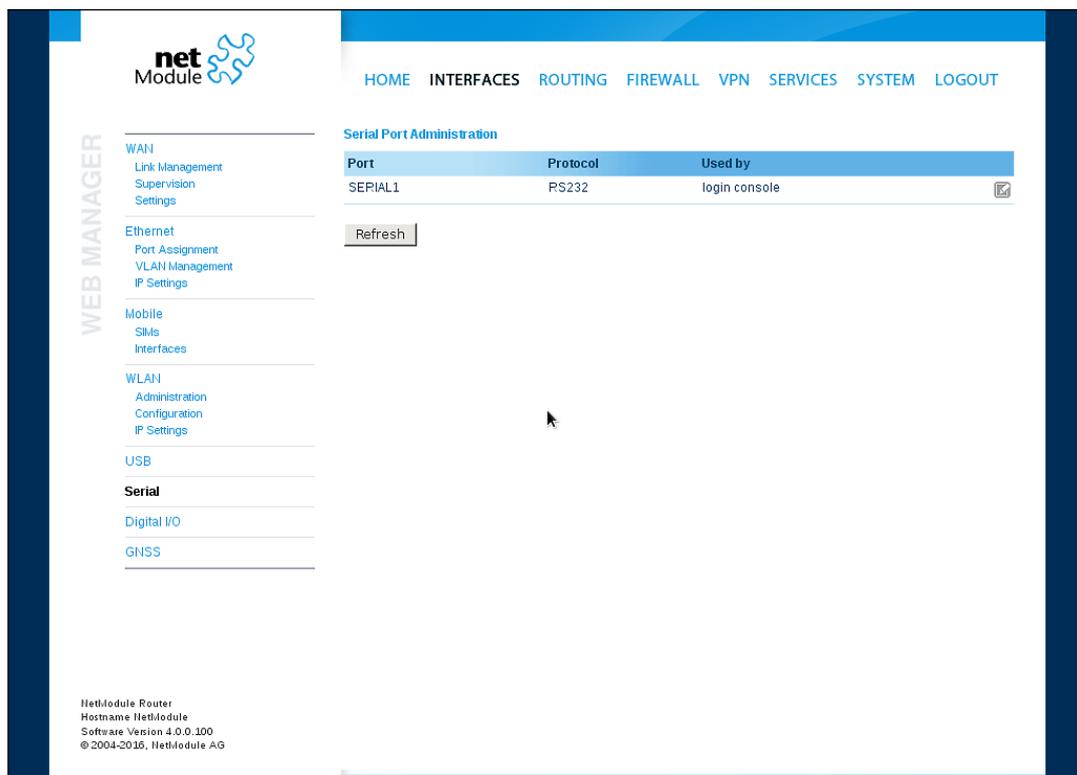


Figure 5.17.: Serial Port Administration

5.5.3. Rules

In general, the firewall is set up of a range of rules which control each packet's permission to pass the router. Please note that the rules are processed by order, that means traversing the list from top to bottom until a matching rule is found. Packets which are not matching any of the rules configured will be ALLOWED.

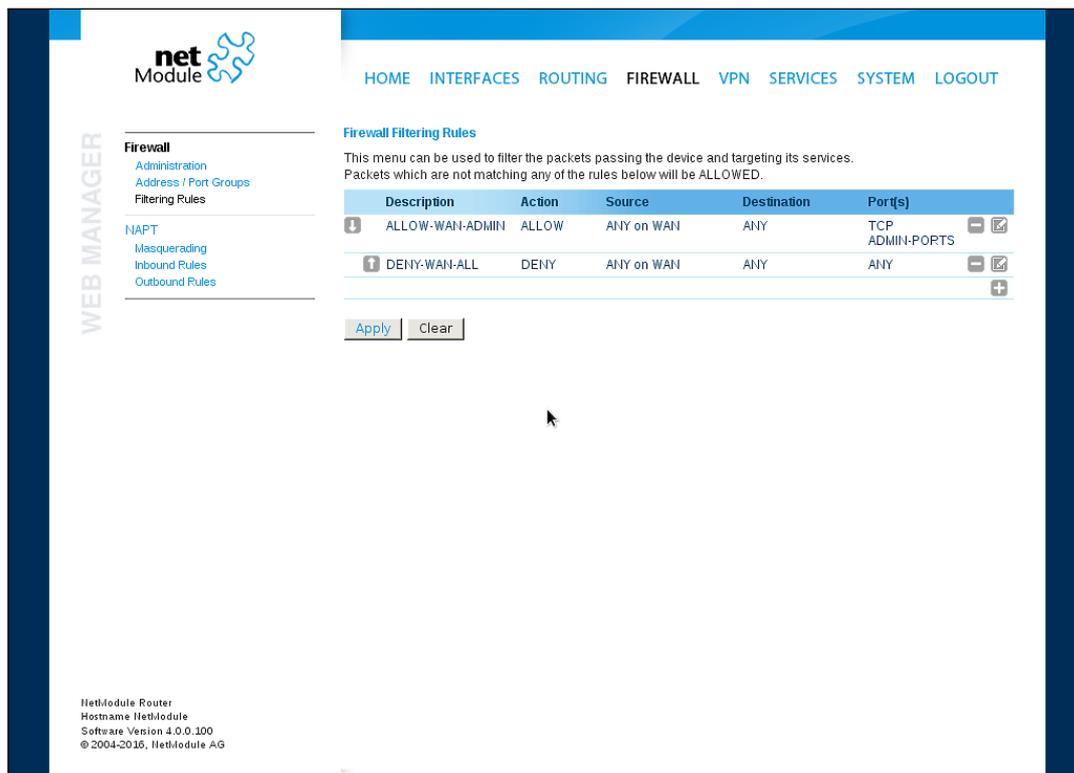


Figure 5.24.: Firewall Rules

Parameter	Firewall Rule Configuration
Description	A meaningful description about the purpose of this rule
Action	Specifies whether the packets of this rule should be allowed or denied
log matches	Throw a syslog message if rule matches
Source	The source address of matching packets, can be any or specified by address/network. Selecting on source MAC addresses is possible as well.
Destination	The destination address of matching packets, can be any, local (addressed to the system itself) or specified by address/network
Incoming interface	The interface on which matching packets are received

Parameter	Firewall Rule Configuration
Protocol	The used IP protocol of matching packets (UDP, TCP or ICMP)
Destination port(s)	The destination port of matching packets, which can be specified by a single port or a range of ports (only UDP/TCP).

The statistics page can be used to figure out if rules have matched any packets and provides a convenient way to debug your firewall setup.

5.5.4. NAT

This page can be used to configure Network Address and Port Translation (NAPT) for packets traversing the system. NAT hereby modifies IP addresses or/and TCP/UDP ports in matching IP packets. By tracking those connections, it will also automatically adjust the returning packets of an IP session.

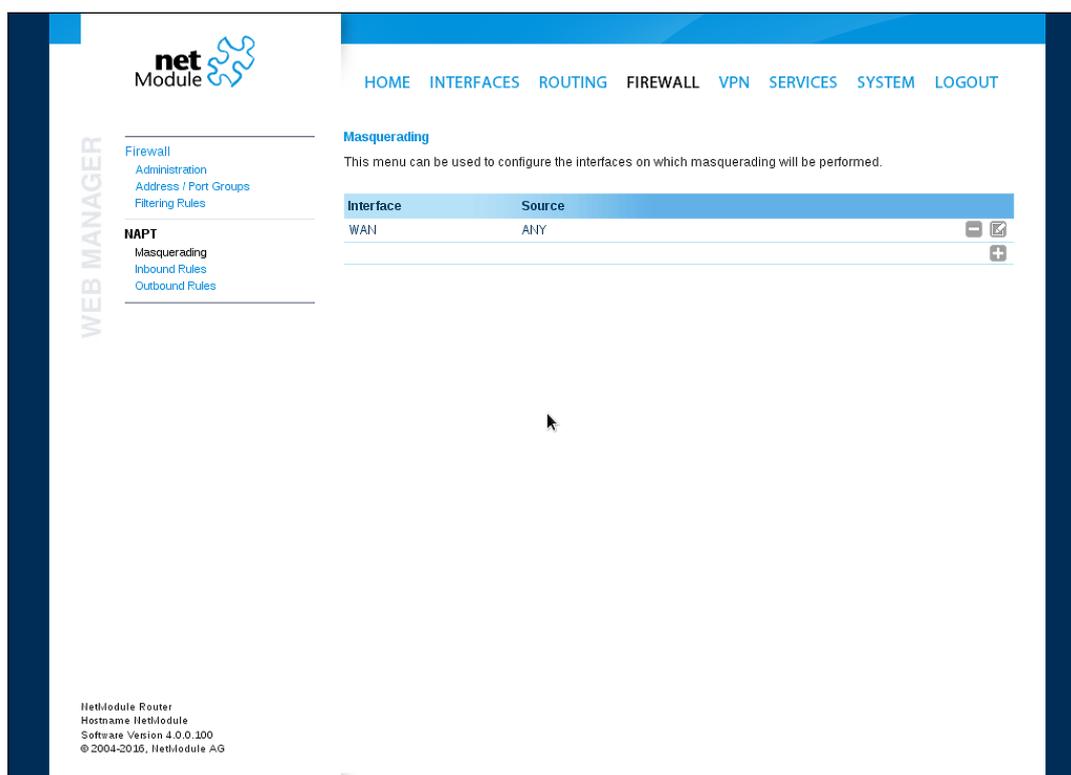


Figure 5.25.: Masquerading

The administration page lets you specify the interfaces on which masquerading will be performed. NAT will hereby use the address of the selected interface and choose a random source port for outgoing connections and thus enables communication between hosts from a private local area network towards hosts on the public network.

Parameter	Masquerading Rules
Interface	The outgoing interface on which connections will be masqueraded
Source address	The source address or network from which matching packets are masqueraded
Source netmask	The source netmask of the network from which matching packets are masqueraded

NAPT Inbound Rules

Inbound rules can be used to modify the target section of IP packets and, for instance, forward a service or port to an internal host. By doing so, you can expose that service and make it available from the Internet. You may also establish 1:1 NAT mapping for a single host using additional outbound rules.

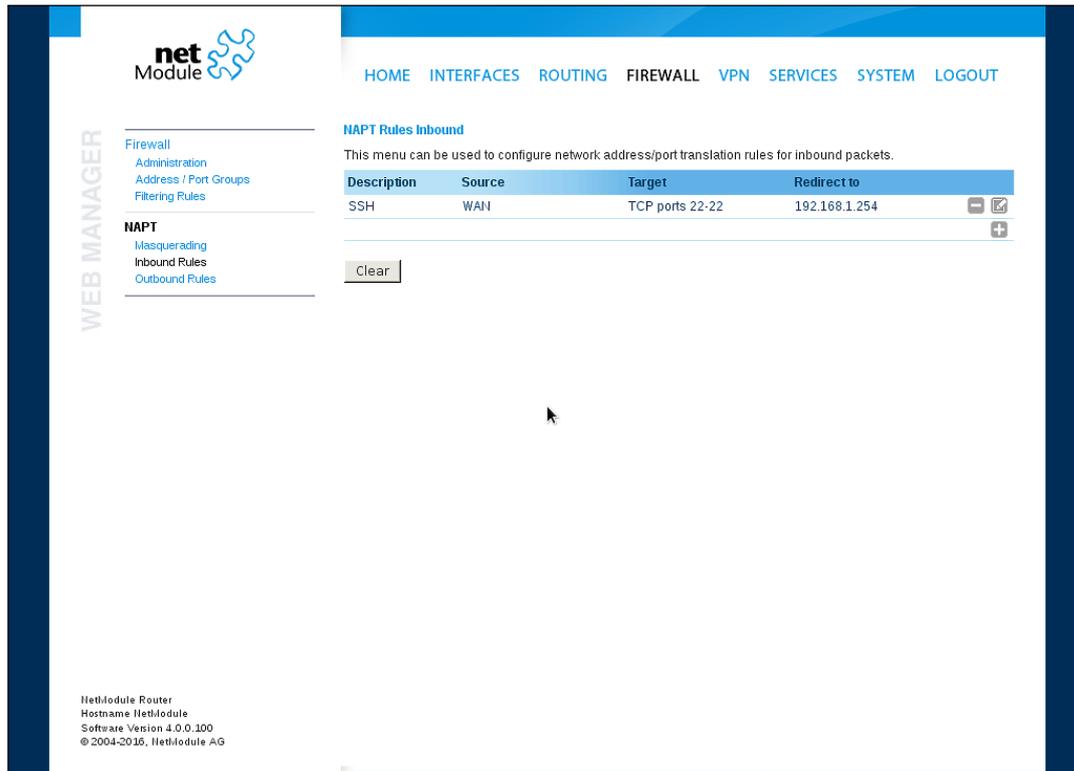


Figure 5.26.: Inbound NAPT

Please note that the specified rules are processed by order, that means, traversing the list from top to bottom until a matching rule is found. If there is no matching rule found, the packet will pass as is.

Parameter	Inbound NAPT Rules
Description	A meaningful description of this rule
Map	Context for this rule: Host, Network or Port-Range - see table below
Incoming interface	The interface from which matching packets are received
Source	The source address or network from which matching packets are received
Target address	The destination address of matching packets (optional)
Protocol	The used protocol of matching packets

Parameter	Inbound NAT Rules
Ports	The used UDP/TCP port of matching packets
Redirect to	The address to which matching packets shall be redirected
Redirect port	The port to which matching packets will be redirected

Select mapping context according to your needs:

Parameter	Mapping contexts
host	Rewrite destination address and port for one given host (i.e. 10.0.0.1:8080 → 192.168.1.100:80)
network	Rewrite destination address for a full network (i.e. 10.0.0.0/24 → 192.168.1.0/24)
port range	Rewrite destination address and port based on the incoming port (i.e. 10.0.0.1:22000-22255 → 192.168.1.0/24:22). There is no corresponding rule for port range translation in outbound rules. Use network based mapping there.

NAPT Outbound Rules

Outbound rules will modify the source section of IP packets and can be used to establish 1:1 NAT mappings but also to redirect packets to a specific service.

Parameter	Outbound NAT Rules
Description	A meaningful description of this rule
Outgoing interface	The outgoing interface on which matching packets are leaving the router
Target	The target address or network to which matching packets are destined
Source address	The source address of matching packets (optional)
Protocol	The used protocol of matching packets
Ports	The used UDP/TCP port of matching packets
Rewrite source address	The address to which the source address of matching packets shall be rewritten
Rewrite source port	The port to which the source port of matching packets shall be rewritten

5.6. VPN

5.6.1. OpenVPN

OpenVPN Administration

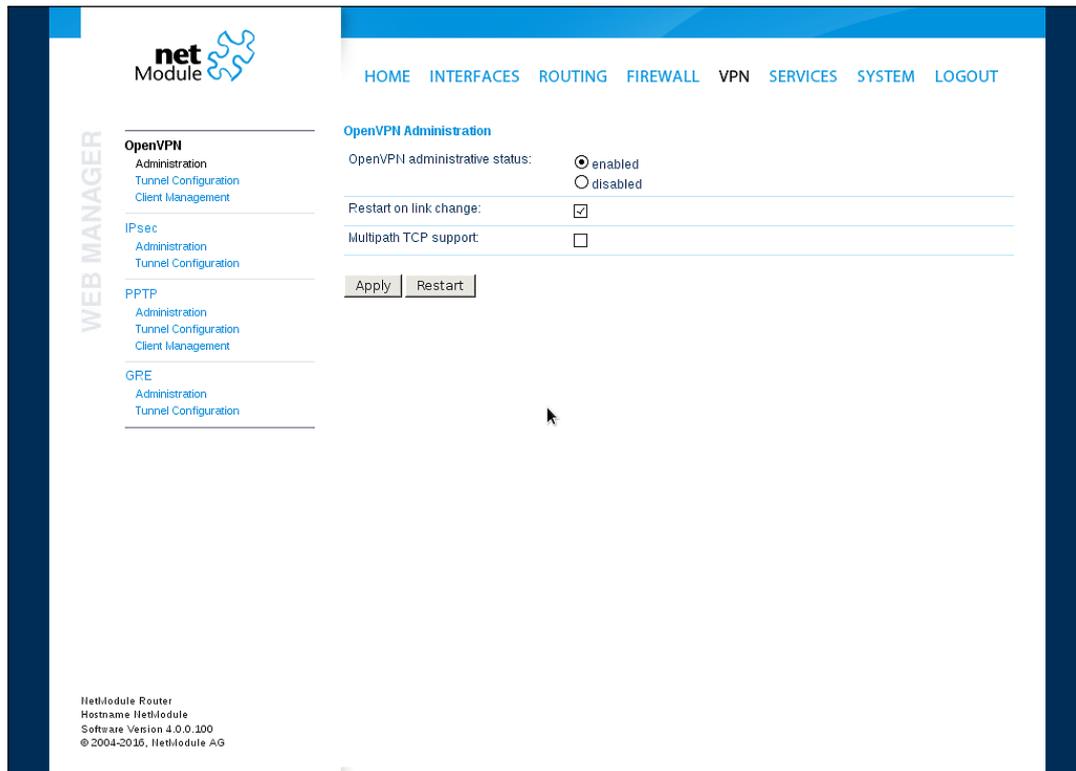


Figure 5.27.: OpenVPN Administration

Tunnel Configuration

NetModule routers support one single server tunnel and up to four client tunnels. You can specify tunnel parameters either in standard configuration or upload an expert mode file which has been created in advance. Refer to chapter 5.6.1 to learn more about how to manage clients and generate the files.

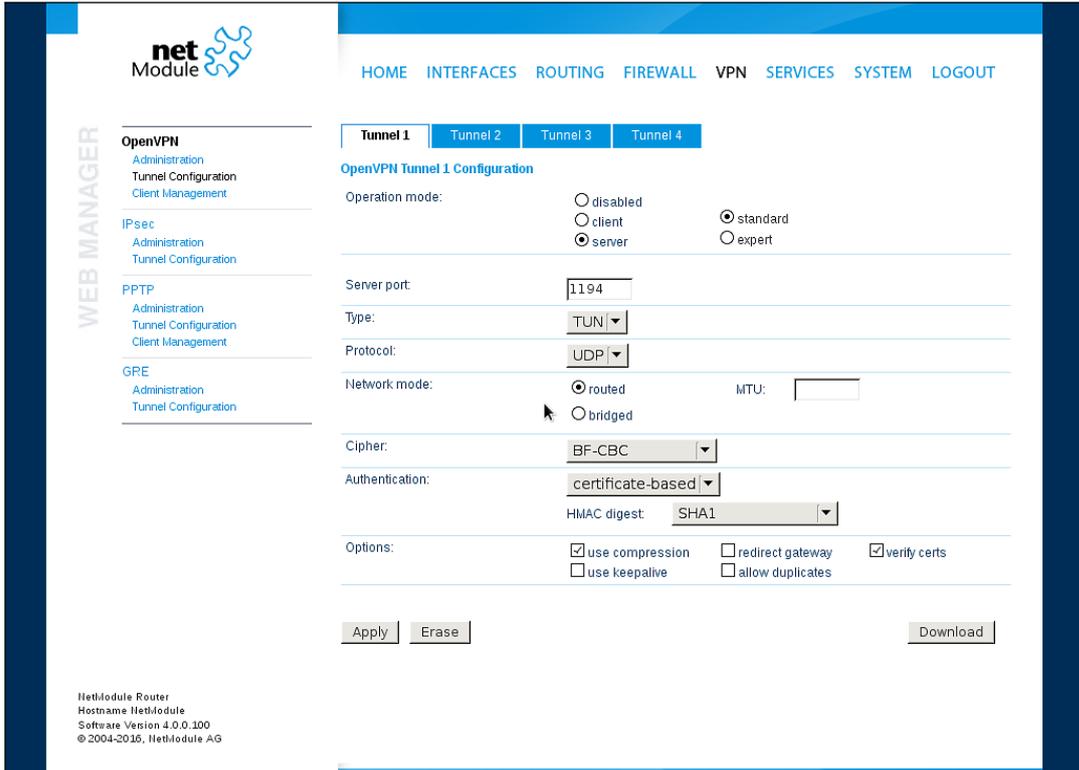


Figure 5.28.: OpenVPN Configuration

Parameter	OpenVPN Configuration
Operation mode	Specifies whether client or server mode should be used for this tunnel, it further specifies if tunnel shall be configured in a standard way or if an expert mode file shall be used.
Multipath TCP	Enables OpenVPN mulipath TCP support

If the tunnel is operated in client mode, the following settings can be applied:

Parameter	OpenVPN Client Configuration
Peer selection	Specifies how the remote peer shall be selected, besides a single server you may configure multiple servers which can, in case of failures, either be selected sequentially (i.e. failover) or randomly (i.e. load balancing)
Server	The address or hostname of the remote server
Port	The port of the remote server (1194 by default)

The following settings can be used to configure a tunnel:

Parameter	OpenVPN Configuration
Interface type	The device type for this tunnel which can be either TUN (typically used for routed connections) or TAP (required for bridged networks)
Protocol	The tunnel protocol to be used for the transport connection
Network mode	Defines how the packets should be forwarded, which can be either routed or bridged from/to a particular LAN interface. If required, you can also specify the maximum transfer unit for the tunnel interface.
MTU	The Maximum Transmission Unit of the tunnel interface
Encryption	The required cipher mechanism used for encryption
Digest	The digest algorithm used for authenticating

Authentication can be done in the following ways:

Parameter	OpenVPN Authentication
certificate-based	Certificates and keys for authenticating the tunnel. Please take care that the proper keys/certificates have been either uploaded or generated (see 5.8.8).
credential-based	Username and password are used for authentication.
both	Verifying the tunnel uses certificates and credentials.
none	Tunnel is not authenticated (discouraged)

The following further options can be applied:

Parameter	OpenVPN Options
use compression	Enable or disable LZO packet compression
use keepalive	Can be used to send a periodic keepalive packet in order to keep the tunnel up despite of inactivity
redirect gateway	By redirecting the gateway, all packets will be directed to the VPN tunnel. Please ensure that essential services (such as DNS or NTP servers) can be reached at the network behind the tunnel. In doubt, create an extra static route pointing to the correct interface.
allow duplicates	Allow multiple clients with the same common name to concurrently connect.
verify certs	Check peer certificate against local CRL.
negotiate DNS	If enabled, the system will use the nameservers which have been negotiated over the tunnel.

OpenVPN Expert Configuration (Client)

The expert configuration mode offers a straightforward way to configure a tunnel by simply uploading a zip package containing the required configuration and optionally key/certificate files. A client tunnel usually consists of the following files:

Parameter	Client Expert Files
client.conf	OpenVPN configuration file (see http://www.openvpn.net for available options)
ca.crt	Root certificate authority file
client.crt	Certificate file
client.key	Private key file
client.p12	PKCS#12 file
ta.key	TLS authentication key file

Please note that you may specify arbitrary file names, however, the configuration file suffix must be `.conf` and all files referred in the configuration file must correspond to relative path names.

OpenVPN Expert Configuration (Server)

A server tunnel typically requires the following files:

Parameter	Server Expert Files
server.conf	OpenVPN configuration file
ca.crt	Root certificate authority file
server.crt	Certificate file
server.key	Private key file
dh1024.pem	Diffie-Hellman parameters file
ccd	A directory containing client-specific configuration files

Keep in mind that a certificate becomes valid once its validity time has been reached, thus an accurate system has to be set prior to creating certificates and establishing a tunnel connection. Please ensure that all NTP servers are reachable. Using host names also requires a working DNS server.

Client Management

Once you have successfully set up an OpenVPN server tunnel, you can manage and enable clients connecting to your service. Currently connected clients can be seen on this page, including the connect time and IP address. You may kick connected clients by disabling them.

The screenshot shows the NetModule Web Manager interface. The top navigation bar includes links for HOME, INTERFACES, ROUTING, FIREWALL, VPN, SERVICES, SYSTEM, and LOGOUT. The left sidebar, labeled 'WEB MANAGER', lists various services: OpenVPN (Administration, Tunnel Configuration, Client Management), IPsec (Administration, Tunnel Configuration), PPTP (Administration, Tunnel Configuration, Client Management), and GRE (Administration, Tunnel Configuration). The main content area is titled 'Client Management' and features a table with columns for Client, Address, and Networks. A single client entry is visible: Client1 with a dynamic address. Below the table is an 'Apply' button. The bottom left corner of the interface displays system information: NetModule Router, Hostname NetModule, Software Version 4.0.0.100, and © 2004-2016, NetModule AG.

Figure 5.29.: OpenVPN Client Management

In the Networking section you can specify a fixed tunnel endpoint address for each client. Please note that, if you intend to use a fixed address for a particular client, you would have to apply fixed addresses to the other ones as well.

You may specify the network behind the clients as well as the routes to be pushed to each client. This can be useful for routing purposes, e.g. in case you want to redirect traffic for particular networks towards the server. Routing between the clients is generally not allowed but you can enable it if desired.

Finally, you can generate and download all expert mode files for enabled clients which can be used to easily populate each client.

Operating in server mode with certificates, it is possible to block a specific client by revoking a possibly stolen client certificate (see [5.8.8](#)).

5.6.2. IPsec

IPsec is a protocol suite for securing IP communications by authenticating and encrypting each packet of a communication session and thus establishing a secure virtual private network.

IPsec includes various cryptographic protocols and ciphers for key exchange and data encryption and can be seen as one of the strongest VPN technologies in terms of security. It uses the following mechanisms:

Mechan	Description
AH	Authentication Headers (AH) provide connectionless integrity and data origin authentication for IP datagrams and ensure protection against replay attacks.
ESP	Encapsulating Security Payloads (ESP) provide confidentiality, data-origin authentication, connectionless integrity, an anti-replay service and limited traffic-flow confidentiality.
SA	Security Associations (SA) provide a secure channel and a bundle of algorithms that provide the parameters necessary to operate the AH and/or ESP operations. The Internet Security Association Key Management Protocol (ISAKMP) provides a framework for authenticated key exchange.

Negotiating keys for encryption and authentication is generally done by the Internet Key Exchange protocol (IKE) which consists of two phases:

Phase	Description
IKE phase 1	IKE authenticates the peer during this phase for setting up an ISAKMP secure association. This can be carried out by either using <code>main</code> or <code>aggressive</code> mode. The <code>main</code> mode approach utilizes the Diffie-Hellman key exchange and authentication is always encrypted with the negotiated key. The <code>aggressive</code> mode just uses hashes of the pre-shared key and therefore represents a less-secure mechanism which should generally be avoided as it is prone to dictionary attacks.
IKE phase 2	IKE finally negotiates IPsec SA parameters and keys and sets up matching IPsec SAs in the peers which is required for AH/ESP later on.

Administration

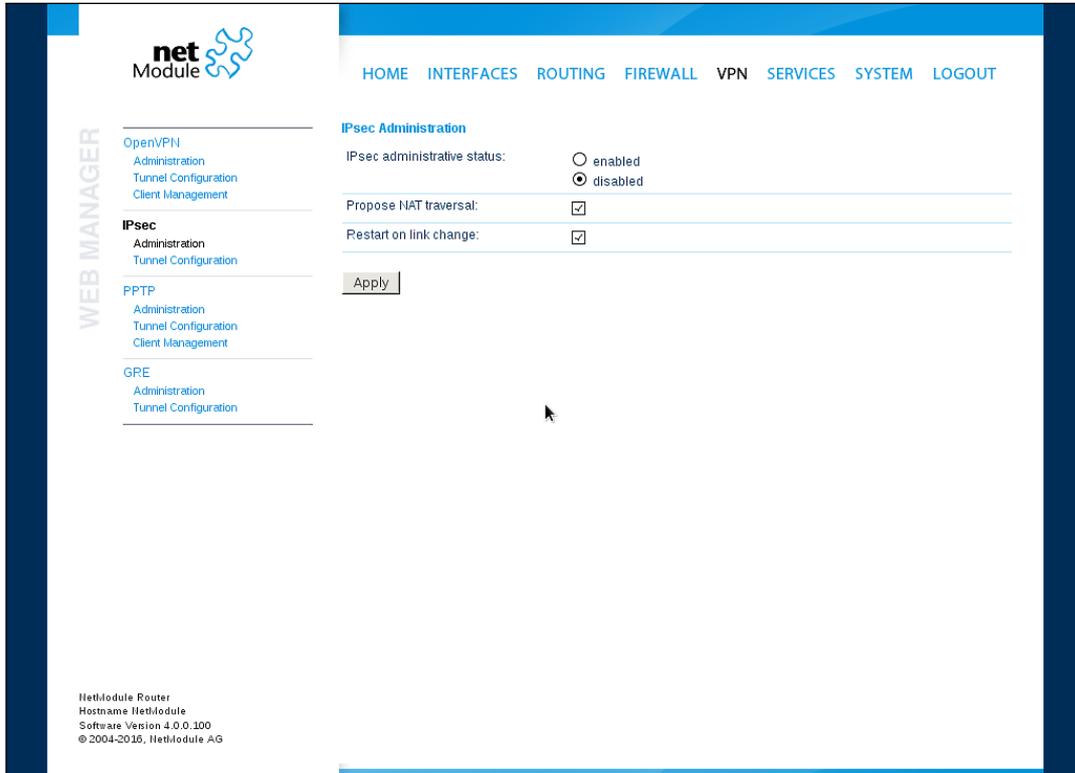


Figure 5.30.: IPsec Administration

This page can be used to enable/disable IPsec, you may also specify whether NAT-Traversal should be used.

NAT-Traversal is mainly used for connections which traverse a path where a router modifies the IP address/port of packets. It encapsulates packets in UDP and therefore requires a slight overhead which has to be taken into account when running over small-sized MTU interfaces.

Please note that running NAT-Traversal makes IKE using UDP port 4500 rather than 500 which has to be taken into account when setting up firewall rules.

Configuration

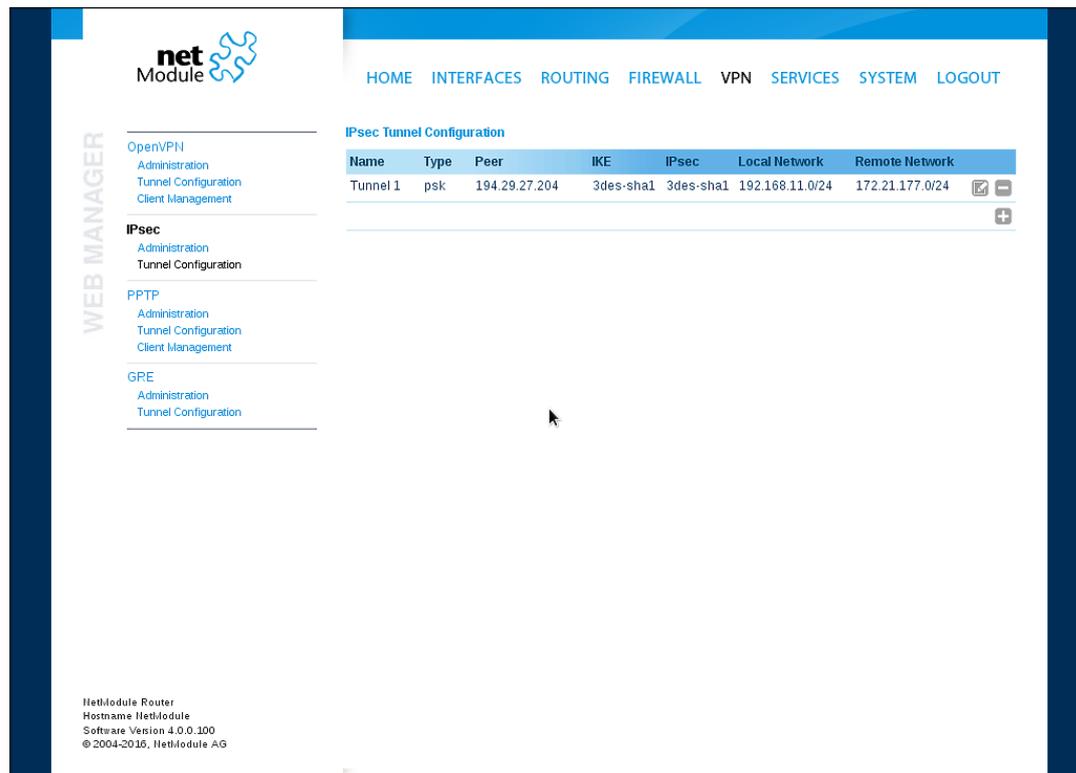


Figure 5.31.: IPsec Configuration

General

For setting up the tunnel you will have to configure the following parameters first:

Parameter	IPsec General Settings
Remote peer	IP address or host name of the remote IPsec peer. You may specify 0.0.0.0 to act as a responder for roadwarrior clients.
DPD Status	Specifies whether Dead Peer Detection (see RFC 3706) shall be used. DPD will detect any broken IPsec connections, in particular the ISAKMP tunnel, and refresh the corresponding SAs (Security Associations) and SPIs (Security Payload Identifier) for a faster re-establishment of the tunnel.
Detection cycle	The delay (in seconds) between DPD keepalives that are sent for this connection (default 30 seconds)
Failure threshold	The number of unanswered DPD requests until the IPsec peer is considered dead (the router will then try to re-establish a dead connection automatically)

Parameter	IPsec General Settings
Action	The action to perform if a peer disconnects. Available choices from the drop-down menu are to clear, hold or to Restart the peer.

IKE Authentication

NetModule routers support IKE authentication through pre-shared keys (PSK) or certificates within a public key infrastructure. Extended Authentication (XAUTH) leverages RADIUS-like authentication and can be used to apply user level access control over IPsec.

Using PSK requires the following settings:

Parameter	IPsec IKE Authentication Settings
PSK	The pre-shared key used to authenticate at the peer
Local ID Type	The type of identification for the local ID which can be a FQDN, username@FQDN or IP address
Local ID	The local ID value
Remote ID Type	The type of identification for the remote ID
Remote ID	The remote ID value

When using certificates you would need to specify the operation mode. When run as PKI client (initiator) you can create a Certificate Signing Request (CSR) in the certificates section which needs to be submitted at your Certificate Authority and imported to the router afterwards. In PKI server mode (concentrator), the router represents the Certificate Authority and issues the certificates for remote peers. They are revokable.

Using XAUTH the following settings can be made:

Parameter	IPsec XAUTH Settings
User name	The name of the XAUTH user
User password	The password of the XAUTH user
Group name	The group ID
Group password	The group secret

IKE Proposal

This section can be used to configure the phase 1 settings:

Parameter	IPsec IKE Proposal Settings
Negotiation mode	Choose the desired negotiation mode. Preferably, <code>main</code> mode should be used but <code>aggressive</code> mode might be applicable when dealing with dynamic endpoint addresses.
Encryption algorithm	The desired IKE encryption method (we recommend AES256)
Authentication algorithm	The desired IKE authentication method (we prefer SHA1 over MD5)
IKE Diffie-Hellman Group	The IKE Diffie-Hellman Group
SA life time	The lifetime of Security Associations
Perfect Forward Secrecy	Specifies whether Perfect Forward Secrecy (PFS) should be used. This feature increases security as PFS avoids penetration of the key-exchange protocol and prevents compromise of previous keys.
Pseudo-random function	PRF algorithms that can optionally be used.

IPsec Proposal

This section can be used to configure the phase 2 settings:

Parameter	IPsec Proposal Settings
Encapsulation mode	The desired encapsulation mode (Tunnel or Transport)
IPsec protocol	The desired IPsec protocol (AH or ESP)
Encryption algorithm	The desired IKE encryption method (we recommend AES256)
Authentication algorithm	The desired IKE authentication method (we prefer SHA1 over MD5)
SA life time	The lifetime of Security Associations
Perfect forward secrecy (PFS)	Specifies whether Perfect Forward Secrecy (PFS) should be used. This feature increases security as PFS avoids penetration of the key-exchange protocol and prevents compromise of previous keys.
Force encapsulation	Force UDP encapsulation for ESP packets even if no NAT situation is detected.

Networks

When creating Security Associations, IPsec will keep track of routed networks within the tunnel. Packets will be only transmitted when a valid SA with matching source and destination network is present. Therefore, you may need to specify the networks right and left of the endpoints by applying the following settings:

Parameter	IPsec Network Settings
Local network	The address of your local area network
Local netmask	The netmask of your local area network
Peer network	The address of the remote network behind the peer
Peer netmask	The netmask of the remote network behind the peer
NAT address	Optionally, you can apply NAT (masquerading) for packets coming from a different local network. The NAT address must reside in the network previously specified as local network.

Client Management

Once you have successfully set up an IPsec tunnel, you can manage and enable clients connecting to your service. It is possible to generate and download expert mode files for enabled clients which can be used to easily populate each client.

5.6.3. PPTP

The Point-to-Point Tunneling Protocol (PPTP) is a method for implementing virtual private networks between two hosts. PPTP is easy to configure and widely deployed amongst Microsoft Dial-up networking servers. However, due to its weak encryption algorithms, it is nowadays considered insecure but it still provides a straightforward way for establishing tunnels.

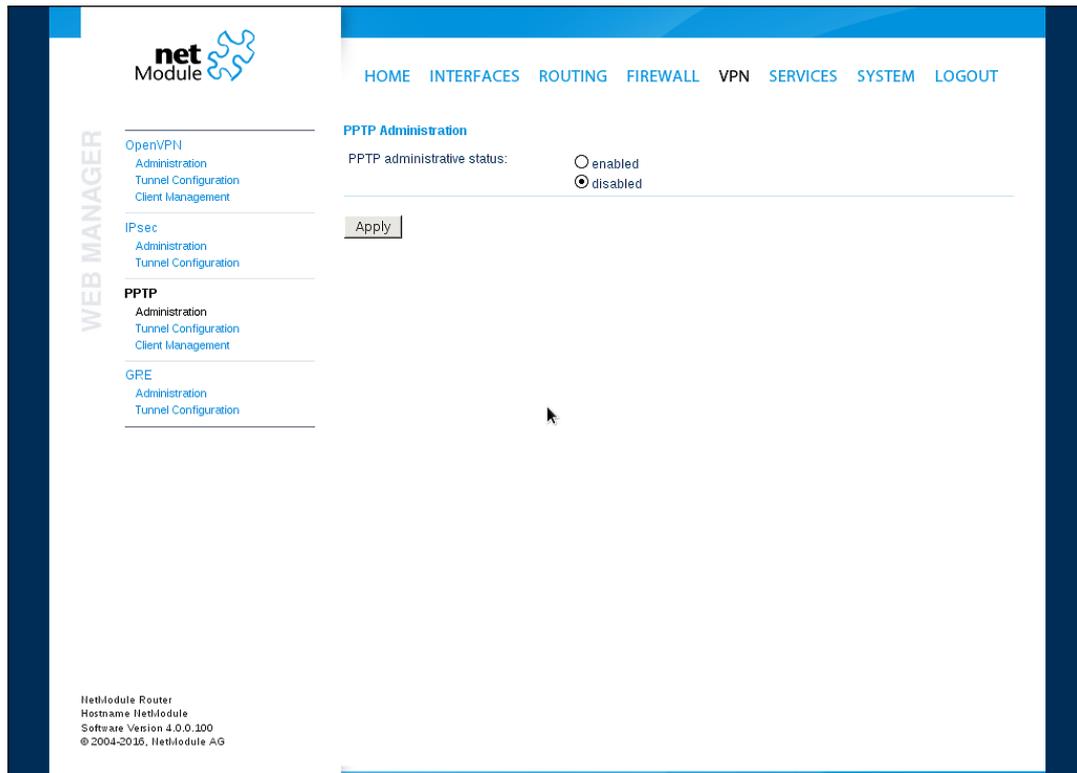


Figure 5.32.: PPTP Administration

When setting up a PPTP tunnel, you would need to choose between server or client. A client tunnel requires the following parameters to be set:

Parameter	PPTP Client Settings
Server address	The address of the remote server
Username	The user-name used for authentication
Password	The password used for authentication

Please note that username and password are not used when setting up clients with fixed addresses.

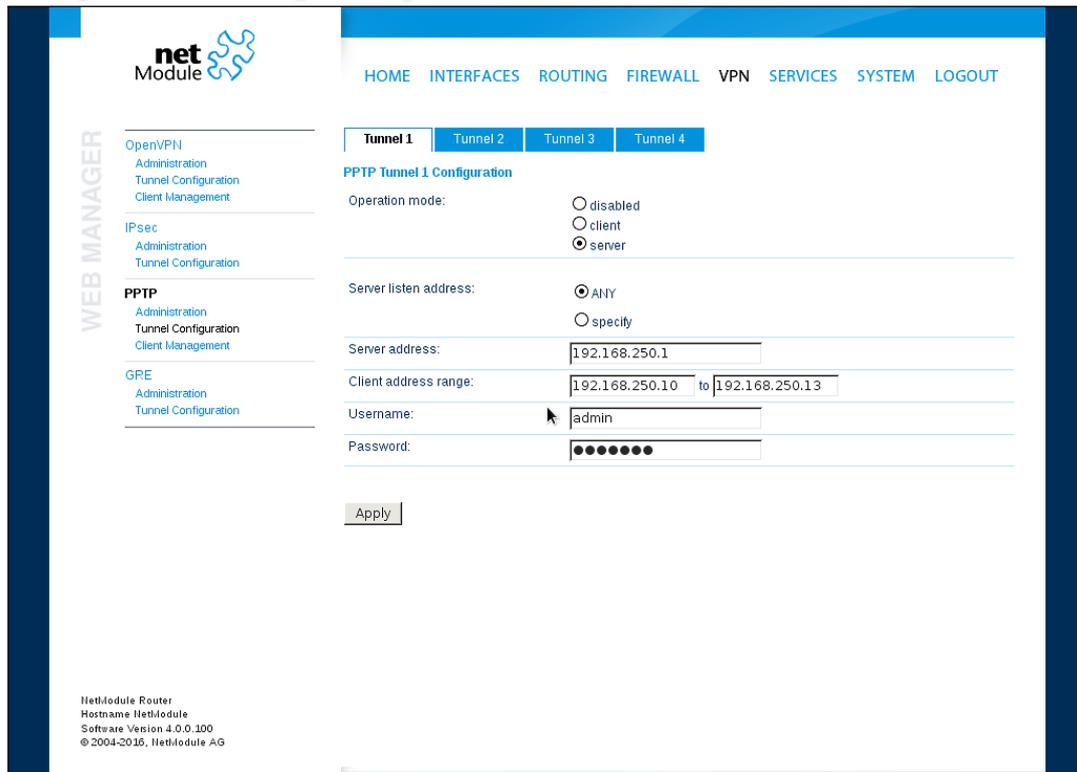


Figure 5.33.: PPTP Tunnel Configuration

Setting up a server requires the following settings:

Parameter	PPTP Server Settings
Listen address	Specifies on which IP address should be listened for incoming client connections
Server address	The server address within the tunnel
Client address range	Specifies a range of IP addresses assigned to each client

PPTP Client Management

PPTP clients for a server tunnel need to be configured here. They are made up of user-name and password. A fixed IP address can be assigned to them which can be used to point any routes to a dedicated tunnel.

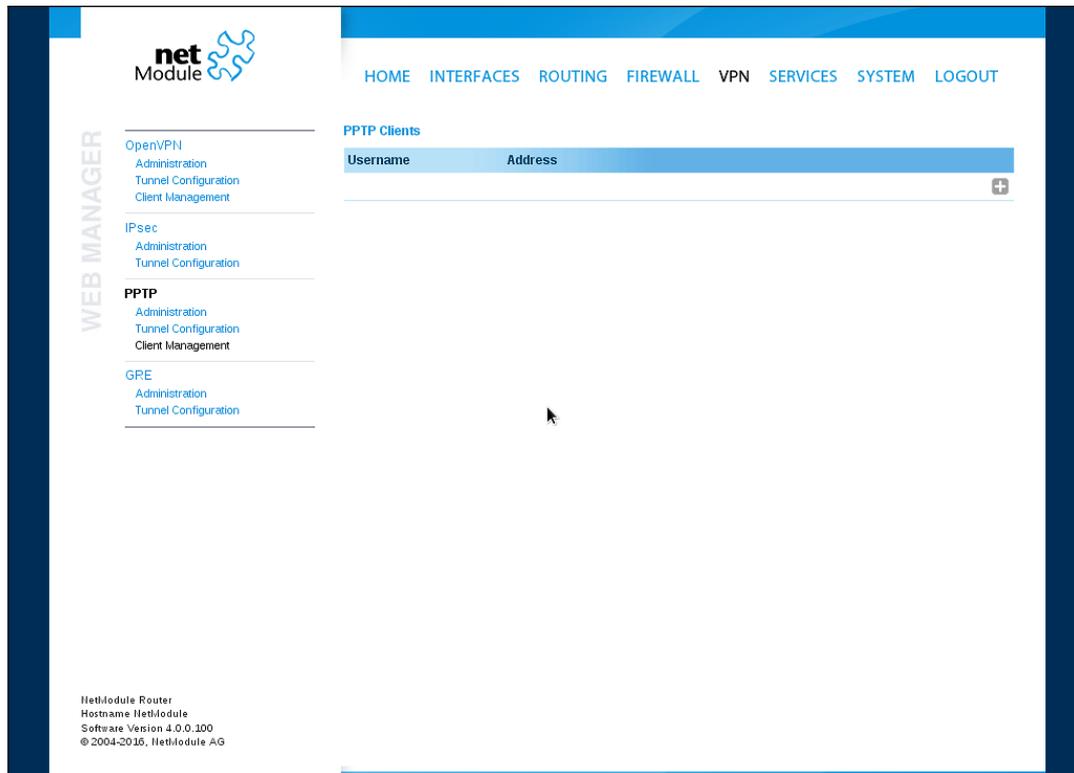


Figure 5.34.: PPTP Client Management

5.6.4. GRE

The Generic Routing Encapsulation (GRE) is a tunneling protocol that can encapsulate a wide variety of network layer protocols inside virtual point-to-point links over IP. GRE is defined in RFC 1701, 1702 and 2784. It does not provide encryption nor authorization but can be used on an address-basis on top of other VPN techniques (such as IPsec) for tunneling purposes. The following parameters are required for setting up a tunnel:

Parameter	GRE Configuration
Peer address	The IP address of the remote peer
Interface	The device type for this tunnel
Local tunnel address	The local IP address of the tunnel
Local tunnel netmask	The local subnet mask of the tunnel
Remote network	The remote network address of the tunnel
Remote netmask	The remote subnet mask of the tunnel

In general, the local tunnel address/netmask should not conflict with any other interface addresses. The remote network/netmask will result in an additional route entry in order to control which packets should be encapsulated and transferred over the tunnel.

5.6.5. Dial-In

On this page you can configure the Dial-In server in order to establish a data connection over GSM calls. Thus, one would generally apply a required service type of 2G-only, so that the modem registers to GSM only. Naturally, a concurrent use of outgoing WWAN interfaces and Dial-In connection is not possible.

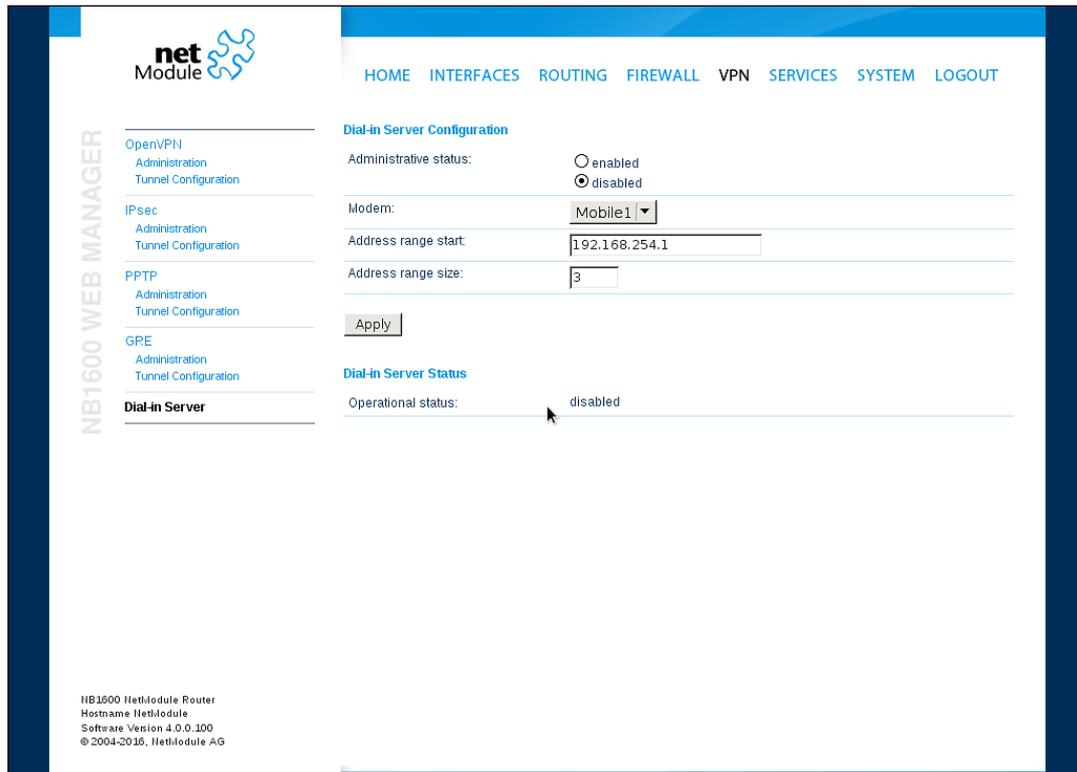


Figure 5.35.: Dial-in Server Settings

The following settings can be set:

Parameter	Dial-in Server Configuration
Administrative status	Specifies whether incoming calls shall be answered or not
Modem	Specifies the modem on which calls can come in
Address range start	Start of the IP address range assigned to incoming clients
Address range size	Number of addresses for client IP address range

The PPP dial-in connection is validated by username admin and the administrator password. Please note that Dial-In connections are generally discouraged. As they are implemented as GSM voice calls, they suffer from unreliability and poor bandwidth.

5.7. SERVICES

5.7.1. SDK

NetModule routers are shipping with a Software Development Kit (SDK) which offers a simple and fast way to implement customer-specific functions and applications. It consists of:

1. An SDK host which defines the runtime environment (a so-called sandbox), that is, controlling access to system resources (such as memory, storage and CPU) and, by doing so, catering for the right scalability
2. An interpreter language called `arena`, a light-weight scripting language optimized for embedded systems, which uses a syntax similar to ANSI-C but adds support for exceptions, automatic memory management and runtime polymorphism on top of that
3. A NetModule-specific Application Programming Interface (API), which ships with a comprehensive set of functions for accessing hardware interfaces (e.g. digital IO ports, GPS, external storage media, serial ports) but also for retrieving system status parameters, sending E-Mail or SMS messages or simply just to configure the router

Anyone, reasonably experienced in the C language, will find an environment that is easy to dig in. However, feel free to contact us via router@support.netmodule.com and we will happily support you in finding a programming solution to your specific problem.

The Language

The `arena` scripting language offers a broad range of POSIX functions (like `printf` or `open`) and provides, together with tailor-made API functions, a simple platform for implementing any sort of applications to interconnect your favourite device or service with the router.

Here comes a short example:

```
/* We are going to eavesdrop on the first serial port
 * and turn on lights via a digital I/O output port,
 * otherwise we'd have to send a short message.
 */

for (attempts = 0; attempts < 3; attempts++) {
  if (nb_serial_read("serial0") == "Knock Knock!") {
    nb_serial_write("serial0", "Who's there?");

    if (nb_serial_read("serial0") == "Santa") {
      printf("Hurray!\n");
      nb_dio_set("out1", 1);
    }
  }
}
nb_sms_send("+123456789", "No presents this year :(")
```

A set of example scripts can be downloaded directly from the router, you can find a list of them in the appendix. The manual which can be obtained from the [NetModule support web page](#) gives a detailed introduction of the language, including a description of all available functions.

SDK API Functions

The current range of API functions can be used to implement the following features:

1. Send/Retrieve SMS
2. Send E-mail
3. Read/Write from/to serial device
4. Control digital input/output ports
5. Run TCP/UDP servers
6. Run IP/TCP/UDP clients
7. Access files of mounted media (e.g. an USB stick)
8. Retrieve status information from the system
9. Get or set configuration parameters
10. Write to syslog
11. Transfer files over HTTP/FTP
12. Perform config/software updates
13. Control the LEDs
14. Get system events, restart services or reboot system
15. Scan for networks in range
16. Create your own web pages
17. Voice control functions
18. SNMP functions
19. CAN socket functions
20. Various network-related functions
21. Other system-related functions

The SDK API manual (which can be downloaded from the router) provides an overview but also explains all functions in detail.

Please note that some functions require the corresponding services (e.g. E-Mail, SMS) or configured interfaces (e.g. CAN) to be properly configured prior to utilizing them in the SDK.

Let's now pay some attention to the very powerful API function `nb_status`. It can be used to query the router's status values in the same manner as they can be shown with the CLI. It returns a structure of variables for a specific section (a list of available sections can be obtained by running `cli status -h`).

By using the `dump` function you can figure out the content of the returned structure:

```
/* dump current location */
dump(nb_status("location"));
```

The script will then generate lines like maybe these:

```
struct(8): {
  .LOCATION_STREET      = string[11]: "Bahnhofquai"
  .LOCATION_CITY        = string[10]: "Zurich"
  .LOCATION_COUNTRY_CODE = string[2]: "ch"
  .LOCATION_COUNTRY     = string[11]: "Switzerland"
  .LOCATION_POSTCODE    = string[4]: "8001"
  .LOCATION_STATE       = string[6]: "Zurich"
  .LOCATION_LATITUDE    = string[9]: "47.3778058"
  .LOCATION_LONGITUDE   = string[8]: "8.5412757"
}
```

In combination with the `nb_config_set` function, it is possible to start a re-configuration of any parts of the system upon status changes. You may query possible sections and parameters again with the CLI:

```
~ $ cli get -c wanlink.0
cli get -c wanlink.0
Showing configuration entities (matching 'wanlink.0'):
```

```
wanlink.0.mode          wanlink.0.multipath    wanlink.0.name
wanlink.0.options      wanlink.0.passthru     wanlink.0.prio
wanlink.0.suspend     wanlink.0.switchback   wanlink.0.weight
```

Running the CLI in interactive mode, you will be also able to step through possible configuration parameters by the help of the TAB key.

Here is an example how one might adopt those functions:

```
/* check current city and enable the second WAN link */

location = nb_status("location");
if (location) {
    city = struct_get(location, "LOCATION_CITY");

    if (city == "Wonderland") {
        for (led = 0; led < 5; led++) {
            nb_led_set(led, LED_BLINK_FAST|LED_COLOR_RED);
        }
    } else {
        printf("You'll never walk alone in %s ...\n", city);
        nb_config_set("wanlink.1.mode=1");
    }
}
```

Running SDK

In the SDK, we are speaking of scripts and triggers which form jobs.

Any arena script can be uploaded to the router or imported by using dedicated user configuration packages. You may also edit the script directly at the Web Manager or select one of our examples. You will further have a testing section on the router which can be used to check your syntax or doing test runs.

Once uploaded, you will have to specify a trigger, that is, telling the router when the script is to be executed. This can be either time-based (e.g. each Monday) or triggered by one of the pre-defined system events (e.g. wan-up) as described in Events chapter 5.7.7. With both, a script and a trigger, you can finally set up an SDK job now. The test event usually serves as a good facility to check whether your job is doing well. The admin section also offers facilities to troubleshoot any issues and control running jobs.

The SDK host (sdkhost) corresponds to the daemon managing the scripts and their operations and thus avoiding any harm to the system. In terms of resources, it will limit CPU and memory for running scripts and also provide a pre-defined portion of the available space of the storage device. You may, however, extend it by external USB storage or (depending on your model) extended flash storage.

Files written to `/tmp` will be hold in memory and will be cleared upon a restart of the script. As your scripts operate in the sandbox, you will have no access to tools on the system (such as `ifconfig`).

Administration

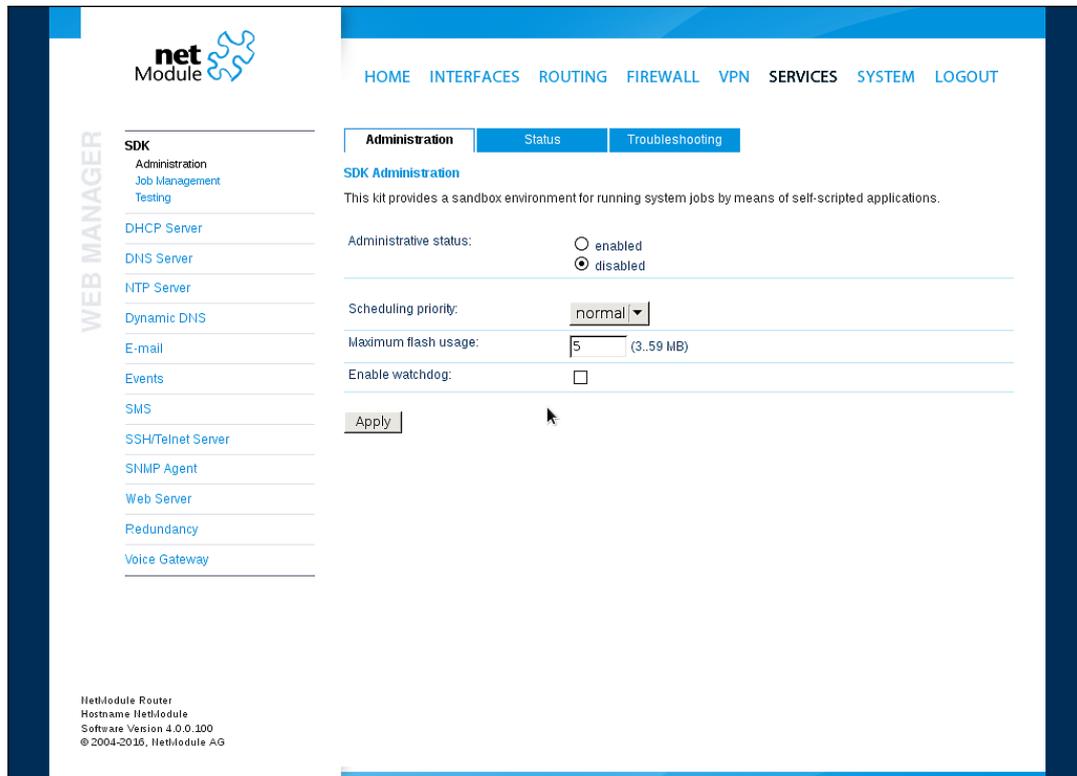


Figure 5.36.: SDK Administration

This page can be used to control the SDK host and apply the following settings:

Parameter	SDK Administration Settings
Administrative status	Specifies whether SDK scripts should run or not
Storage	The storage device on which the sandbox shall be stored (see chapter 5.8.1)
Max. size	The maximum amount of MBytes your scripts can consume on the storage device
Scheduling priority	Specifies the process priority of the sdkhost, higher priorities will speed up scheduling your scripts, lower ones will have less impact to the host system
Enable watchdog	This option will enable watchdog supervision for each script which leads to a reboot of the system if the script does not respond or stopped with an exit code not equal zero.

The status page informs you about the current status of the SDK. It provides an overview about any finished jobs, you can also stop a running job there and view the script output in the troubleshooting section where you will also find links for downloading the manuals and examples.

Job Management

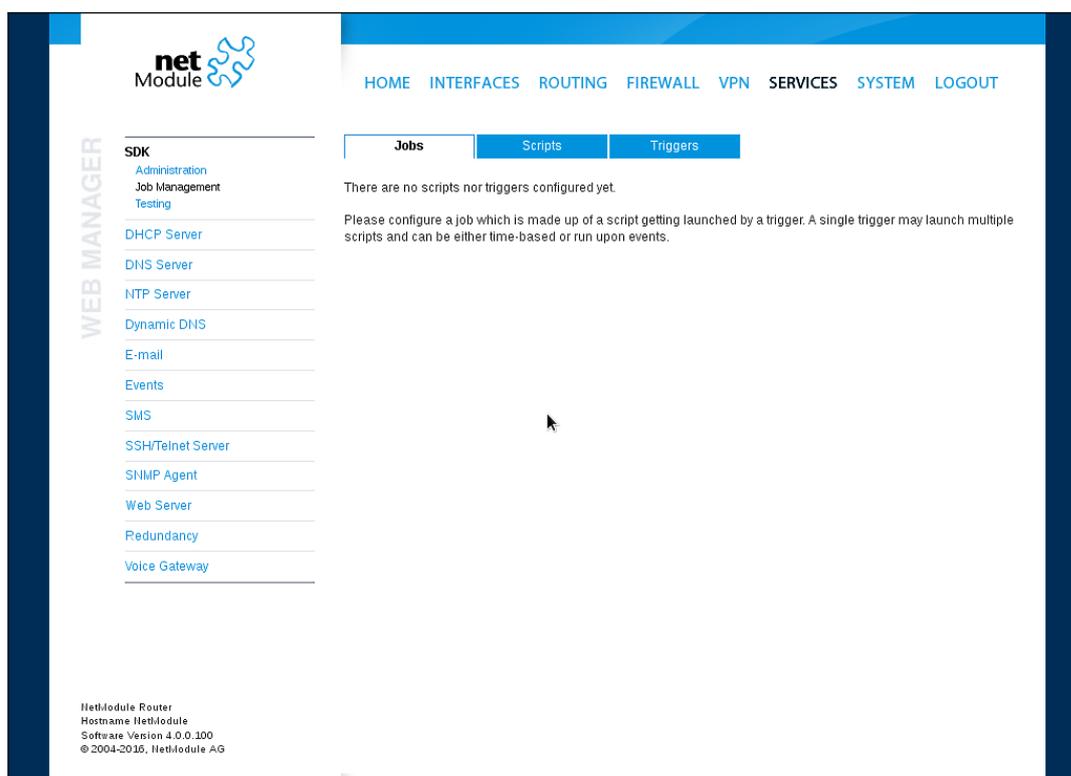


Figure 5.37.: SDK Jobs

This page can be used to set up scripts, triggers and jobs. It is usually a good idea to create a trigger first which is made up by the following parameters:

Parameter	SDK Trigger Parameters
Name	A meaningful name to identify the trigger
Type	The type of the trigger, either time-based or event-based
Condition	Specifies the time condition for time-based triggers (e.g. hourly)
Timespec	The time specification which, together with the condition, specifies the time(s) when the trigger should be pulled
Event	The system event upon which the trigger should be pulled

You can now add your personal script to the system by applying the following parameters:

Parameter	SDK Script Parameters
Name	A meaningful name to identify the script
Description	An optional description of the script

Parameter	SDK Script Parameters
Arguments	An optional set of arguments passed to the script (supports quoting)
Action	You may either edit a script, upload it to the system or select one of the example scripts or an already uploaded script

You are ready to set up a job afterwards, it can be created by using the following parameters:

Parameter	SDK Job Parameters
Name	A meaningful name to identify the job
Trigger	Specifies the trigger that should launch the job
Script	Specifies the script to be executed
Arguments	Defines arguments which can be passed to the script (supports quoting), they will precede the arguments you formerly may have assigned to the script itself

You can trigger each configured job directly which can be helpful for testing purposes.

Pages

Any programmed SDK pages will show up here.

Testing

The testing page offers an editor and an input field for optional arguments which can be used to perform test runs of your script or test dedicated portions of it or upload an entire file. Please note that you might need to quote arguments as they will otherwise be separated by white-spaces.

```
/* arguments: 'schnick schnack "s c h n u c k"'
for (i = 0; i < argc; i++) {
    printf("argv%d: %s\n", i, argv[i]);
}

/* generates:
 *   argv0: scriptname
 *   argv1: schnick
 *   argv2: schnack
 *   argv3: s c h n u c k
 */
```

In case of syntax errors, arena will usually print error messages as follows (indicating the line and position where the parsing error occurred):

```
/scripts/testrun:2:10:FATAL: parse error, unexpected $, expecting ';''
```

SDK Sample Application

As an introduction, you can step through a sample application, namely the SMS control script, which implements remote control over short messages and can be used to send a status of the system back to the sender. The source code is listed in the appendix.

Once enabled, you can send a message to the phone number associated with a SIM / modem. It generally requires a password to be given on the first line and a command on the second, such as:

```
admin01
status
```

We strongly recommend to use authentication in order to avoid any unintended access, however you may pass `noauth` as argument to disable it. You can then skip the first line containing the password. Having a closer look to the script, you will see that you will also be able to restrict the list of permitted senders. Please inspect the system log for troubleshooting any issues.

The following commands are supported:

Command	Action
status	Will reply a message to the sender including a short system overview
connect	Will enable the first WAN link configured on the system
disconnect	Will disable the first WAN link configured on the system
reboot	Initiates a reboot of the system
output 1 on	Turns on the first digital output port
output 1 off	Turns off the first digital output port
output 2 on	Turns on the second digital output port
output 2 off	Turns off the second digital output port

Table 5.89.: SMS Control Commands

A response to the status command typically looks like:

```
System: NB2700 hostname (00:11:22:AA:BB:CC)
WAN1: WWAN1 is up (10.0.0.1, Mobile1, UMTS, -83 dBm, LAI 12345)
GPS: lat 47.377894, lon 8.540055, alt 282.200
OVPN: client on tun0 is up (10.0.8.4)
DIO: IN1=off, IN2=off, OUT1=on, OUT2=off
```

5.7.2. DHCP Server

This section can be used to individually configure the Dynamic Host Configuration Protocol (DHCP) service for each LAN interface which will serve dynamic IP addresses to hosts in the local network. You may also have a look to the status page where you can find an overview about negotiated client addresses.

Please note that WLAN interfaces (for each SSID) will pop up here as well in case you have configured an access point respectively.

The screenshot shows the NetModule web interface for configuring the DHCP Server on LAN1. The interface includes a sidebar menu with options like Administration, Job Management, and Testing. The main content area shows the DHCP Server configuration for LAN1, including fields for First lease address (192.168.1.100), Last lease address (192.168.1.199), Lease duration (7200 seconds), and DHCP options (use default selected). The Static Hosts section is currently empty. The footer of the interface displays: NetModule Router, Hostname NetModule, Software Version 4.0.0.100, © 2004-2016, NetModule AG.

Figure 5.38.: DHCP Server

The following settings for each interface can be applied then:

Parameter	DHCP Server Settings
Operation mode	Specifies whether the DHCP server is enabled or not
First lease address	The first address out of the range of IP addresses given to hosts
Last lease address	The last address out of this range
Lease duration	Number of seconds how long a given lease shall be valid until it has to be requested again

Parameter	DHCP Server Settings
Persistent leases	By turning on this option the router will remember issued leases even after a reboot. This can be used to ensure that the same IP address will be assigned to a particular host.
DHCP options	By default the DHCP will hand out the interface address as default gateway and the current DNS server addresses if not configured elsewhere. You can specify fixed addresses here.
Only allow static hosts	Any requests coming from none-static hosts will be ignored.

It is also possible to configure specific lease addresses for particular clients.

Parameter	DHCP Static Hosts Settings
IP address	The IP address of the lease
Identified by	Specifies by which criteria the client shall be identified
MAC address	The MAC address of the client
hostname	The client identifier (DHCP option 61)
port	The Ethernet port on which the DHCP request is received

5.7.3. DNS Server

The DNS server can be used to proxy DNS requests towards servers on the net which have for instance been negotiated during WAN link negotiation. By pointing DNS requests to the router, one can reduce outbound DNS traffic as it is caching already resolved names but it can be also used for serving fixed addresses for particular host names.

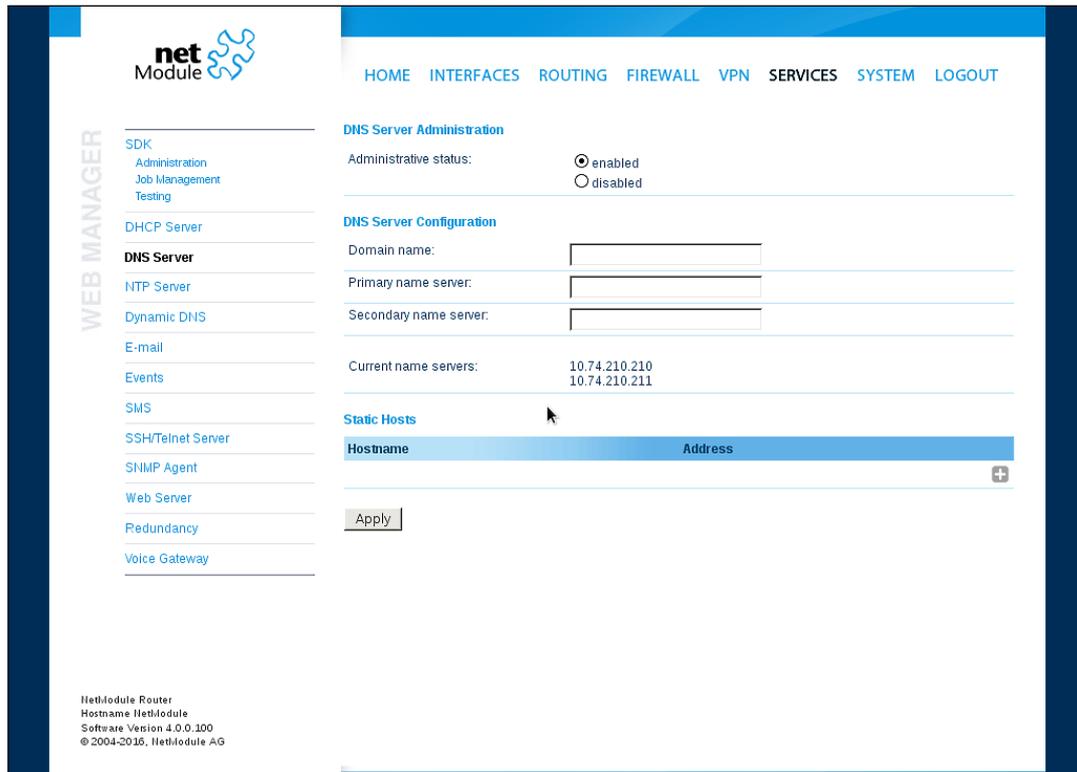


Figure 5.39.: DNS Server

The following settings can be applied:

Parameter	DNS Server Settings
Administrative status	Enables or disables the DNS server
Domain name	The domain name used for short name lookups
Primary name server	The primary default name server which will be used instead of negotiated name servers
Secondary name server	The secondary default name server which will be used instead of negotiated name servers

You may further configure static hosts for serving fixed IP addresses for various host names.

Parameter	DNS Static Hosts Settings
Address	The IP address of the static host
Hostname	The hostname of the static host

Please remember to point DNS lookups of local hosts to the router's address.

5.7.4. NTP Server

This section can be used to individually configure the Network Time Protocol (NTP) server function.

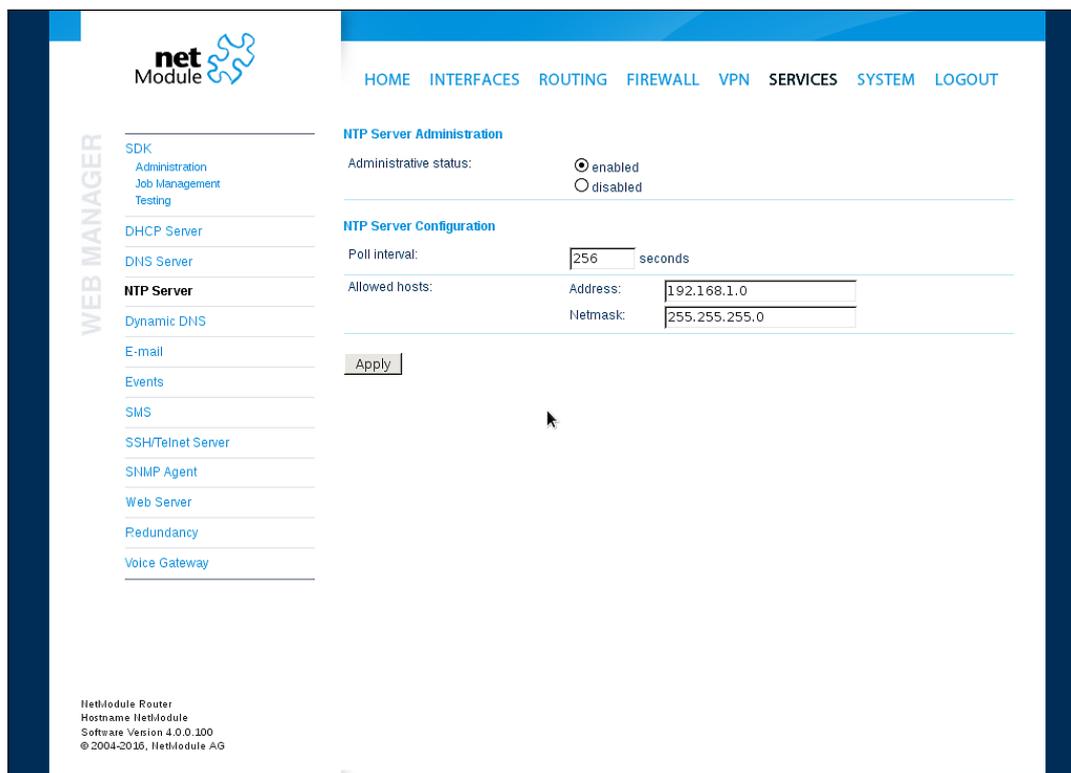


Figure 5.40.: NTP Server

The following settings for each interface can be applied then:

Parameter	NTP Server Settings
Administrative status	Specifies whether the NTP server is enabled or not
Poll interval	Defines the polling interval (64..2048 seconds) for synchronizing the time with the master clock servers
Allowed hosts	Defines the IP address range which is allowed to poll the NTP server

For setting the system time of the device see [5.8.1](#).

5.7.5. Dynamic DNS

The Dynamic DNS client can be used to tell one or multiple DynDNS providers the current IP address of your system. This address can be derived from the current hotlink interface or the outgoing interface which will be used when contacting the server. We further support to ask the CheckIP service at dyndns.org for obtaining the current Internet address which can be useful in NAT scenarios. The DynDNS client will be triggered whenever a WAN or VPN link comes up.

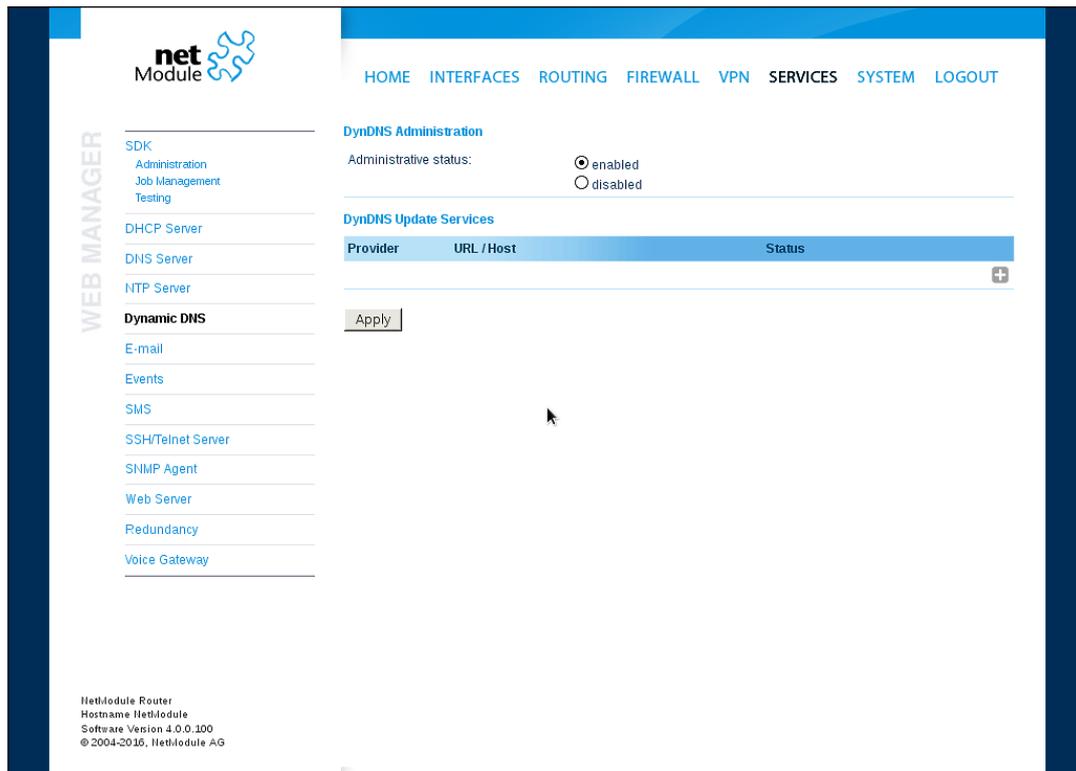


Figure 5.41.: Dynamic DNS Settings

We provide support for a bunch of common DynDNS operators but it is also possible to define a custom update URL.

Please note that your NetModule router can operate as DynDNS server on its own, provided that you have your hosts pointed to the DNS service of the router.

We can further operate the GnuDIP protocol and RFC2136-like dynamic DNS updates. The latter is in general secured by a TSIG key.

A DynDNS service can receive the following parameters:

Parameter	Dynamic DNS Settings
Provider	You can choose one of the listed providers or provide a custom URL
Dynamic address	Specifies whether the address is derived from the hot-link or via an external service
Hostname	The host-name provided by your DynDNS service (e.g. my-box.dyndns.org)
Port	The HTTP port of the service (typically 80)
Username	The user-name used for authenticating at the service
Password	The password used for authentication
Protocol	The protocol used for authentication (HTTP, HTTPS)
Server address	The address of the server which shall be updated
Server port	The port of the server which shall be updated
TSIG key name	The name of the TSIG key which is allowed to perform updates
TSIG key	The TSIG key encoded in base64

5.7.6. E-Mail

The E-Mail client can be used to send notifications to a particular E-Mail address upon certain events or by SDK scripts.

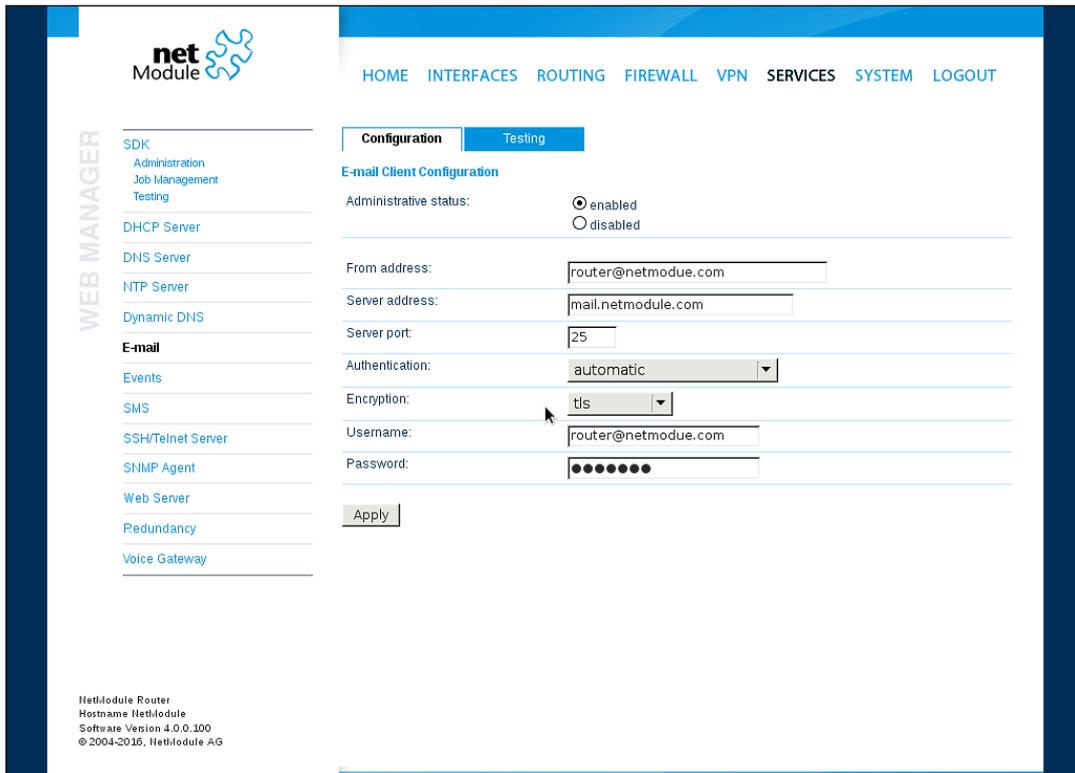


Figure 5.42.: E-Mail Settings

It can be enabled by applying the following settings.

Parameter	E-Mail Client Settings
E-mail client status	Administrative status of the E-Mail client
From e-mail address	E-Mail address of the sender
Server address	SMTP server address
Server port	SMTP server port (typically 25)
Authentication method	Select the required authentication method which will be used to authenticate against the SMTP server
Encryption	Select the encryption. Can be tls or none.
Username	User name used for authentication
Password	Password used for authentication

5.7.7. Events

By using the event manager you can notify remote systems about system events. A notification can be sent using E-Mail, SMS or SNMP traps.

Parameter	Event Notification Settings
E-Mail address	The E-Mail address to which the notification shall be sent (E-Mail client must be enabled)
Phone number	The phone number to which the notification shall be sent (SMS service must be enabled)
SNMP host	The SNMP host or address to which the trap shall be sent
SNMP port	The port of the remote SNMP service
Username	The username for accessing the remote SNMP service
Password	The password for accessing the remote SNMP service
Authentication	The authentication algorithm for accessing the remote SNMP service (MD5 or SHA)
Encryption	The encryption algorithm for accessing the remote SNMP service (DES or SHA)
Engine ID	The engine ID of the remote SNMP service

The messages will contain a description provided by you and a short system information. A list of all system events can be found in the appendix [A.2](#).

5.7.8. SMS

Administration

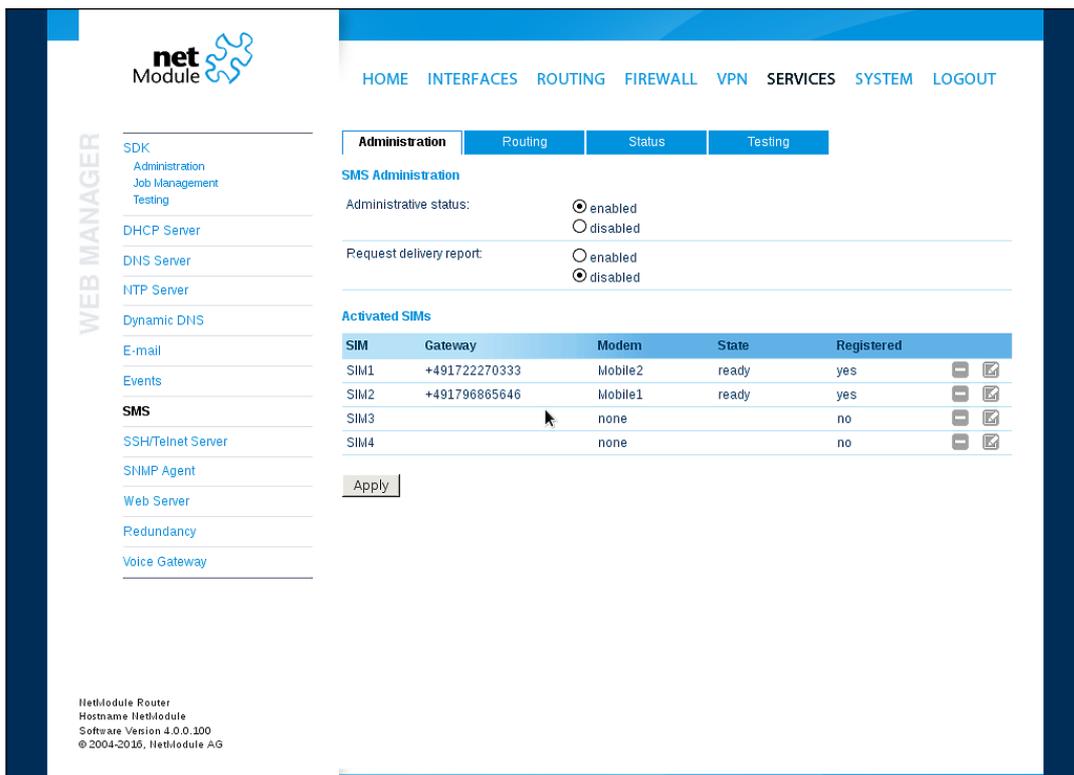
NetModule routers can receive or send short messages (SMS) if enabled by your SIM provider. Messages are received/sent by the modem which has been assigned to a SIM, so one has to properly configure a SMS-capable default modem as described in chapter 5.3.3.

Please note that the system may switch SIMs in case you are running multiple WWAN interfaces sharing the same SIM. Thus, it may happen that a different modem will be used for communication or, if the SIM is unassigned, any operation will even stop.

Please do not forget that modems might register roaming to foreign networks where other fees may apply. You can manually assign a fixed network (by LAI) in the Mobile SIMs section (see 5.3.3).

Sending messages heavily depends on the registration state of the modem and whether the provided SMS Center service works and may fail. You may use the `sms-report-received` event to figure out whether a message has been successfully sent.

Received messages are pulled from the SIMs and temporarily stored on the router but get cleared after a system reboot. Please consider to consult an SDK script in case you want to process or copy them.



The screenshot displays the NetModule Router web interface. The top navigation bar includes links for HOME, INTERFACES, ROUTING, FIREWALL, VPN, SERVICES, SYSTEM, and LOGOUT. The left sidebar, labeled WEB MANAGER, contains a tree view with categories like SDK, DHCP Server, DNS Server, NTP Server, Dynamic DNS, E-mail, Events, SMS, SSH/Telnet Server, SNMP Agent, Web Server, Redundancy, and Voice Gateway. The main content area has tabs for Administration, Routing, Status, and Testing. Under the Administration tab, there is an SMS Administration section with two radio button options: Administrative status (enabled/disabled) and Request delivery report (enabled/disabled). Below this is a table titled 'Activated SIMs' with columns: SIM, Gateway, Modem, State, and Registered. The table lists four SIMs: SIM1 (+491722270333, Mobile2, ready, yes), SIM2 (+491796865646, Mobile1, ready, yes), SIM3 (none, no), and SIM4 (none, no). Each row has minus and plus icons for expansion/collapse. An 'Apply' button is located below the table. At the bottom left, there is a footer with router information: NetModule Router, Hostname NetModule, Software Version 4.0.0.100, © 2004-2016, NetModule AG.

SIM	Gateway	Modem	State	Registered
SIM1	+491722270333	Mobile2	ready	yes
SIM2	+491796865646	Mobile1	ready	yes
SIM3		none		no
SIM4		none		no

Figure 5.43.: SMS Configuration

The relevant page can be used to enable the SMS service and specify on which it should operate. We identify SIMs based on their IMEI number and track their statistics in a non-volatile manner.

Parameter	SMS SIM Configuration
SMS gateway	The service center number for sending short messages. It is generally retrieved automatically from your SIM card but you may define a fix number here.

Routing & Filtering

By using SMS routing you can specify outbound rules which will be applied whenever message are sent. On the one hand, you can forward them to an enabled modem. For a particular number, you can for instance enforce messages being sent over a dedicated SIM. Phone numbers can also be specified by regular expressions, here are some examples:

Number	Result
+12345678	Specifies a fixed number
+1*	Specifies any numbers starting with +1
+1*9	Specifies any numbers starting with +1 and ending with 9
+[12]*	Specifies any numbers starting with either +1 or 2

Table 5.99.: SMS Number Expressions

Please note that numbers have to be entered in international format including a valid prefix. On the other hand, you can also define rules to drop outgoing messages, for instance, when you want to avoid using any expensive service or international numbers.

Both types of rules form a list will be processed by order, forwarding outgoing messages over the specified modem or dropping them. Messages which are not matching any of the rules below will be dispatched to the first available modem.

Filtering serves a concept of firewalling incoming messages, thus either dropping or allowing them on a per-modem basis. The created rules are processed by order and in case of matches will either drop or forward the incoming message before entering the system. All non-matching messages will be allowed.

Status

The status page can be used to the current modem status and get information about any sent or received messages. There is a small SMS inbox reader which can be used to view or delete the messages. Please note that the inbox will be cleared each midnight in case it exceeds 512 kBytes of flash usage.

Testing

This page can be used to test whether SMS sending in general or filtering/routing rules works. The maximum length per message part is limited to 160 characters, we also suggest to exclusively use characters which are supported by the GSM 7-bit alphabet.

5.7.9. SSH/Telnet Server

Apart from the Web Manager, the SSH and Telnet services can be used to log into the system. Valid users include *root* and *admin* as well as additional users as they can be created in the User Accounts section. Please note, that a regular system shell will only be provided for the *root* user, the CLI will be launched for any other user whereas normal users will only be able to view status values, the *admin* user will obtain privileges to modify the system.

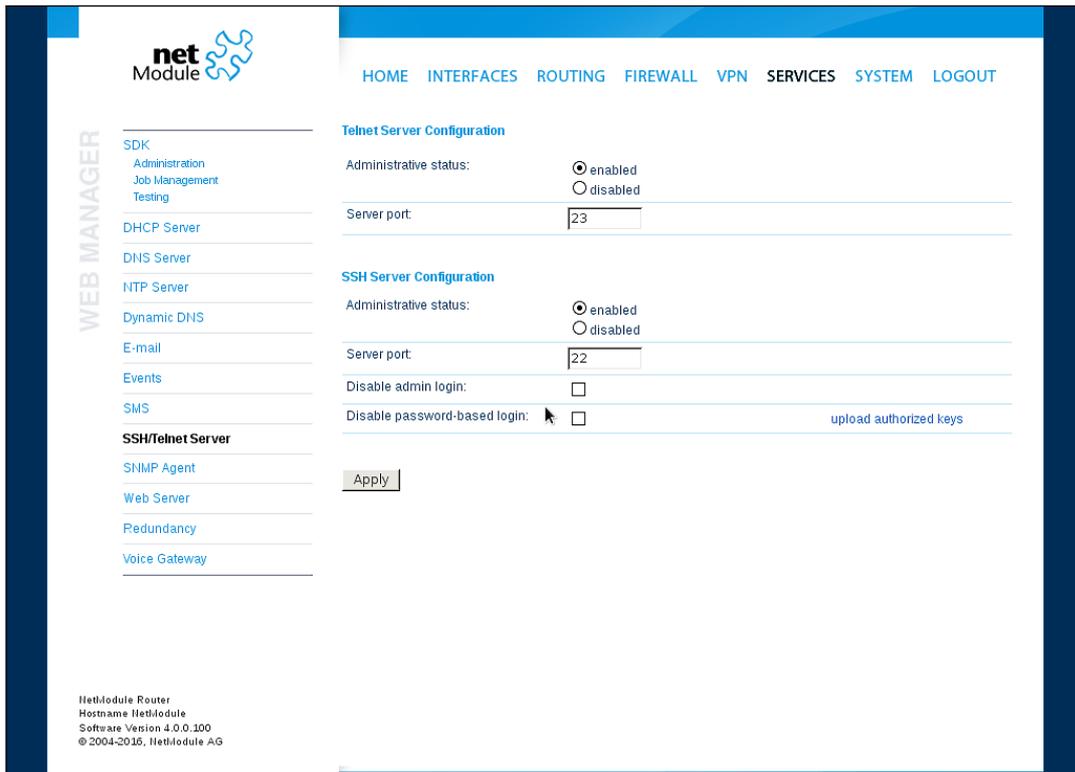


Figure 5.44.: SSH and Telnet Server

Please note that these services will be accessible from the WAN interface also. In doubt, please consider to disable or restrict access to them by applying applicable firewall rules. The following parameters can be applied to the Telnet service:

Parameter	Telnet Server Settings
Administrative status	Whether the Telnet service is enabled or disabled
Server port	The TCP port of the service (usually 23)

The following parameters can be applied to the SSH service:

Parameter	SSH Server Settings
Administrative status	Whether the SSH service is enabled or disabled
Server port	The TCP port of the service (usually 22)
Disable admin login	Disable login for admin users
Disable password-based login	By turning on this option, all users will have to authenticate by SSH keys which can be uploaded to the router.

5.7.10. SNMP Agent

NetModule routers are equipped with an SNMP daemon, supporting basic MIB tables (such as ifTable), plus additional enterprise MIBs to manage multiple systems.

Parameter	Supported MIBs
.1.3.6.1.2.1	MIB-II (RFC1213), SNMPv2-MIB (RFC3418)
.1.3.6.1.2.1.2.1	IF-MIB (RFC2863)
.1.3.6.1.2.1.4	IP-MIB (RFC1213)
.1.3.6.1.2.1.10.131	TUNNEL-MIB (RFC4087)
.1.3.6.1.2.25	HOST-RESOURCES-MIB (RFC2790)
.1.3.6.1.6.3.10	SNMP-FRAMEWORK-MIB
.1.3.6.1.6.3.11	SNMPv2-SMI (RFC2578)
.1.0.8802.1.1.2	LLDP-MIB
.1.0.8802.1.1.2.1.5.4795	LLDP-EXT-MED-MIB
.1.3.6.1.4.1.31496	VENDOR-MIB

The VENDOR-MIB tables offer some additional information over the system and its WWAN, GNSS and WLAN interfaces. They can be accessed over the following OIDs:

Parameter	Vendor MIB OID Assignment
NBAdminTable	.1.3.6.1.4.1.31496.10.40
NBWwanTable	.1.3.6.1.4.1.31496.10.50
NBGnssTable	.1.3.6.1.4.1.31496.10.51
NBDioTable	.1.3.6.1.4.1.31496.10.53
NBWlanTable	.1.3.6.1.4.1.31496.10.60
NBWanTable	.1.3.6.1.4.1.31496.10.22

They offer facilities for:

- rebooting the device
- updating to a new system software via FTP/TFTP/HTTP
- updating to a new system configuration via FTP/TFTP/HTTP
- getting WWAN/GNSS/WLAN/DIO information

Our VENDOR-MIB is listed in the appendix or can be downloaded directly from the router.

SNMP Configuration

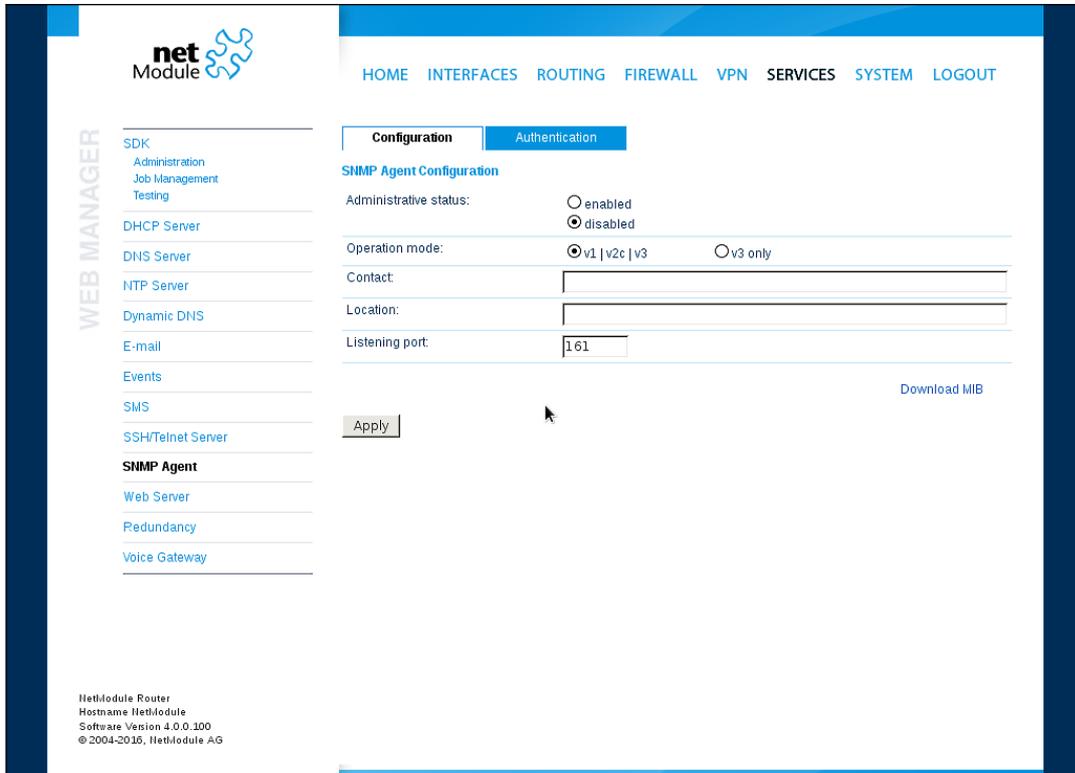


Figure 5.45.: SNMP Agent

The following parameters can be used to configure the SNMP agent:

Parameter	SNMP Configuration
Administrative status	Enable or disable the SNMP agent
Operation mode	Specifies if agent should run in compatibility mode or for SNMPv3 only
Contact	System maintainer or other contact information
Location	Location of the device
Listening Port	SNMP agent port

Once the SNMP agent is enabled, SNMP traps can be generated using SDK scripts.

SNMP Authentication

When running in SNMPv3, it is possible to configure the following authentication settings:

Parameter	SNMPv3 Authentication
Authentication	Defines the authentication (MD5 or SHA)
Encryption	Defines the privacy protocols to use (DES or AES)

In general, the admin user can read and write any values. Read access will be granted to any other system users.

There is no authentication/encryption in SNMPv1/v2c and should not be used to set any values. However, it is possible to define its communities and authoritative host which will be granted administrative access.

Parameter	SNMPv1/v2c Authentication
Read community	Defines the community name for read access
Admin community	Defines the community name for admin access
Allowed host	Defines the host which is allowed for admin access

Attention must be paid to the fact that SNMP passwords have to be more than 8 characters long. Shorter passwords will be doubled for SNMP (e.g. admin01 becomes admin01admin01). Please note that the SNMP daemon is also listening on WAN interfaces and it is therefore suggested to restrict the access with the firewall.

Typical SNMP Commands

Setting MIB values and triggering extensions is generally limited to the SNMPv3 admin user. It is possible to specify an administrative host for SNMP v1/2c.

The SNMP extensions can be read and triggered as follows:

Getting the software version of the system:

```
snmpget -v 3 -u admin -n "" -l authNoPriv -a MD5 -x DES -A admin01admin01
192.168.1.1 1.3.6.1.4.1.31496.10.40.1.0
```

Getting the kernel version:

```
snmpget -v 3 -u admin -n "" -l authNoPriv -a MD5 -x DES -A admin01admin01
192.168.1.1 1.3.6.1.4.1.31496.10.40.2.0
```

Getting the serial number:

```
snmpget -v 3 -u admin -n "" -l authNoPriv -a MD5 -x DES -A admin01admin01
192.168.1.1 1.3.6.1.4.1.31496.10.40.3.0
```

Getting the current config description:

```
snmpget -v 3 -u admin -n "" -l authNoPriv -a MD5 -x DES -A admin01admin01
192.168.1.1 1.3.6.1.4.1.31496.10.40.4.0
```

Getting the current config hash:

```
snmpset -v 3 -u admin -n "" -l authNoPriv -a MD5 -x DES -A admin01admin01
192.168.1.1 1.3.6.1.4.1.31496.10.40.5.0
```

Restarting the device:

```
snmpset -v 3 -u admin -n "" -l authNoPriv -a MD5 -x DES -A admin01admin01
192.168.1.1 1.3.6.1.4.1.31496.10.40.10.0 i 1
```

Running a configuration update:

```
snmpset -v 3 -u admin -n "" -l authNoPriv -a MD5 -x DES -A admin01admin01
192.168.1.1 1.3.6.1.4.1.31496.10.40.11.0 s "http://server/directory"
```

You can use TFTP, HTTP, HTTPS and FTP URLs (specifying a username/password or a port is not yet supported).

Please note that config updates expect a zip-file named <serial-number>.zip in the specified directory.

Getting the configuration update status:

```
snmpget -v 3 -u admin -n "" -l authNoPriv -a MD5 -x DES -A admin01admin01
192.168.1.1 1.3.6.1.4.1.31496.10.40.12.0
```

The return value can be one of: succeeded (1), failed (2), inprogress (3), notstarted (4).

Running a software update:

```
snmpset -v 3 -u admin -n "" -l authNoPriv -a MD5 -x DES -A admin01admin01
192.168.1.1 1.3.6.1.4.1.31496.10.40.13.0 s "http://server/directory"
```

Getting the software update status:

```
snmpget -v 3 -u admin -n "" -l authNoPriv -a MD5 -x DES -A admin01admin01
192.168.1.1 1.3.6.1.4.1.31496.10.40.14.0
```

The return value can be one of: succeeded (1), failed (2), inprogress (3), notstarted (4).

Setting the update operation:

```
snmpset -v 3 -u admin -n "" -l authNoPriv -a MD5 -x DES -A admin01admin01
192.168.1.1 1.3.6.1.4.1.31496.10.40.15.0 i 1
```

By default, the update operation is set to update (0) which results in an immediate update of software or configuration once triggered. One may also set the operation to store (1) which will only store the software or configuration package. It can be later activated using the following switch operators.

Switching to alternative software:

```
snmpset -v 3 -u admin -n "" -l authNoPriv -a MD5 -x DES -A admin01admin01
192.168.1.1 1.3.6.1.4.1.31496.10.40.16.0 i 0
```

The return value can be derived from the software update status.

Switching to alternative config:

```
snmpset -v 3 -u admin -n "" -l authNoPriv -a MD5 -x DES -A admin01admin01
192.168.1.1 1.3.6.1.4.1.31496.10.40.16.0 i 1
```

The return value can be derived from the config update status.

Getting the alternative config description:

```
snmpget -v 3 -u admin -n "" -l authNoPriv -a MD5 -x DES -A admin01admin01
192.168.1.1 1.3.6.1.4.1.31496.10.40.17.0
```

Getting the alternative config hash:

```
snmpget -v 3 -u admin -n "" -l authNoPriv -a MD5 -x DES -A admin01admin01
192.168.1.1 1.3.6.1.4.1.31496.10.40.18.0
```

Getting the alternative software version:

```
snmpget -v 3 -u admin -n "" -l authNoPriv -a MD5 -x DES -A admin01admin01
192.168.1.1 1.3.6.1.4.1.31496.10.40.19.0
```

Getting the alternative software hash:

```
snmpget -v 3 -u admin -n "" -l authNoPriv -a MD5 -x DES -A admin01admin01
192.168.1.1 1.3.6.1.4.1.31496.10.40.20.0
```

Setting digital OUT1:

```
snmpset -v 3 -u admin -n "" -l authNoPriv -a MD5 -x DES -A admin01admin01
192.168.1.1 .1.3.6.1.4.1.31496.10.53.10.0 i 0
snmpset -v 3 -u admin -n "" -l authNoPriv -a MD5 -x DES -A admin01admin01
192.168.1.1 .1.3.6.1.4.1.31496.10.53.10.0 i 1
```

Setting digital OUT2:

```
snmpset -v 3 -u admin -n "" -l authNoPriv -a MD5 -x DES -A admin01admin01
192.168.1.1 .1.3.6.1.4.1.31496.10.53.11.0 i 0
snmpset -v 3 -u admin -n "" -l authNoPriv -a MD5 -x DES -A admin01admin01
192.168.1.1 .1.3.6.1.4.1.31496.10.53.11.0 i 1
```

Listing discovered devices:

```
snmpget -v 3 -u admin -n "" -l authNoPriv -a MD5 -x DES -A admin01admin01
192.168.1.1 .1.0.8802.1.1
```

5.7.11. Web Server

This page can be used to configure different ports for accessing the Web Manager via HTTP/HTTPS. We strongly recommend to use HTTPS when accessing the web service via a WAN interface as the communication will be encrypted and thus avoids any misuse of the system.

In order to enable HTTPS you would need to generate or upload a server certificate in the section 5.8.8.

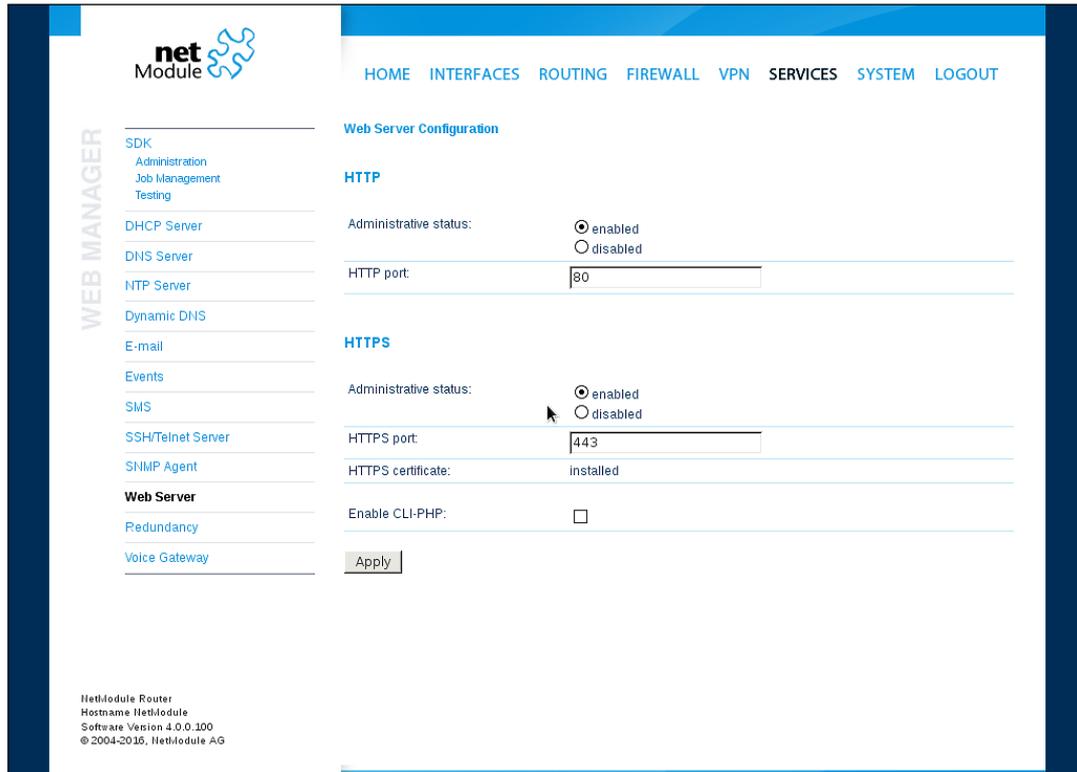


Figure 5.46.: Web Server

Parameter	Web Server Settings
Administrative Status	Enable or disable the Web server
HTTP port	Web server port for HTTP connections
HTTPS port	Web server port for HTTPS connections
Enable CLI-PHP	Enable CLI-PHP service (see chapter 6.17)

5.7.12. Discovery

This page can be used to enabled discovery protocols which can be used to discover and to get discovered by other hosts.

Parameter	Discovery Configuration
Administrative status	Administrative status
Enabled protocols	List of enabled discovery protocols

The following protocols are supported:

Parameter	Discovery Configuration
LLDP	Link Layer Discovery Protocol
CDP	Cisco Discovery Protocol
FDP	Foundry Discovery Protocol
SONMP	Nortel Discovery Protocol
EDP	Extreme Discovery Protocol
IRDP	ICMP Router Discovery Protocol

IRDP implements RFC1256 and can also inform locally connected hosts about the nexthop gateway. Any discovered hosts will be exposed to the LLDP-MIB and can be queried over SNMP or CLI/GUI.

5.7.13. Redundancy

This page can be used to set up a redundant pair of NetModule routers (or other systems) by running the Virtual Router Redundancy Protocol (VRRP) between them. A typical VRRP scenario defines a first host playing the master and another the backup device, they both define a virtual gateway IP address which will be distributed by gratuitous ARP messages for updating the ARP cache of all LAN hosts and thus redirecting the packets accordingly. A takeover will happen within approximately 3 seconds as soon as the partner is not reachable anymore (checked via multicast packets). This may happen when one device is rebooting or the Ethernet link went down. Same applies when the WAN link goes down.

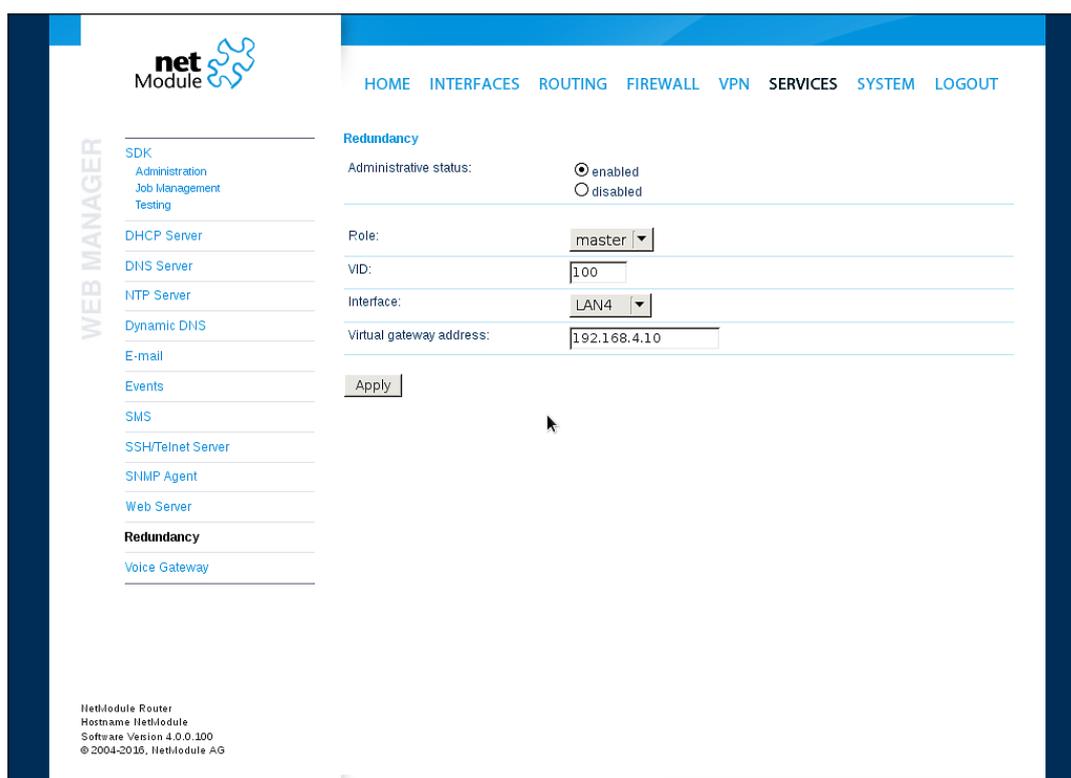


Figure 5.47.: VRRP Configuration

In case DHCP has been activated, please keep in mind that you will need to reconfigure the DHCP gateway address offered by the server and let them point to the virtual gateway address. In order to avoid conflicts you may turn off DHCP on the backup device or even better, split the DHCP lease range across both routers in order to prevent any lease duplication.

Parameter	Redundancy Configuration
Administrative status	Administrative status
Role	The role of this system (either master or backup)
VID	The Virtual Router ID (you can theoretically run multiple instances)

Parameter	Redundancy Configuration
Interface	Interface on which VRRP should be performed
Virtual gateway address	The virtual gateway address formed by the participating hosts

We assign a priority of 100 to the master and 1 to the backup router. Please adapt the priority of your third-party device appropriately.

5.7.14. Voice Gateway

Depending on your hardware, you can set up a voice gateway on the router which can be used to connect mobile calls to VoIP clients and vice versa.

Administration

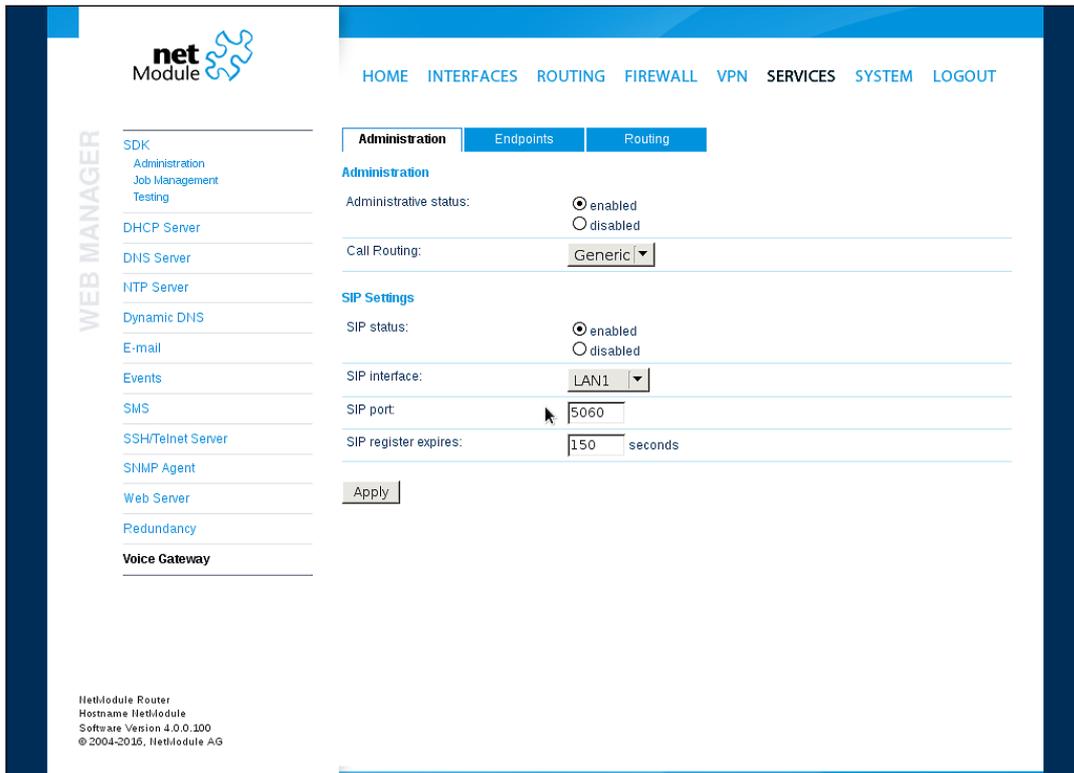


Figure 5.48.: Voice Gateway Administration

The following parameters can be used to set it up:

Parameter	Voice Gateway Administration Settings
Administrative status	Specifies whether the gateway shall be enabled or disabled
Call routing	Defines who will be responsible for call routing. If SDK has been specified you would need to install a script (see examples) which will be responsible for routing and accepting the calls. Otherwise the static routing configuration will be used.
SIP status	Specifies whether the SIP agent will be enabled or disabled
SIP interface	Specifies the interface (LAN or WAN) on which the agent should listen for incoming calls
SIP port	Specifies the agent's listening port

Parameter	Voice Gateway Administration Settings
SIP user name	Specifies the username used in from headers
SIP register expires	Specifies the registration interval in seconds

In case you are running multiple WWAN interfaces sharing the same SIM, please bear in mind that the system may switch SIMs during operation which will also result in different settings for voice communication.

Voice Endpoints

The screenshot shows the NetModule Router web interface. The main content area has three tabs: Administration, Endpoints, and Routing. The 'Endpoints' tab is selected, showing a table of Voice Endpoints. The table has three columns: Name, Type, and Using. There are two rows of data:

Name	Type	Using
Vom1	Voice-over-Mobile	Mobile1 (registered at Vodafone.de with SIM2)
Sip1	SIP (registrar)	Subscriber admin

Below the table is a 'Refresh' button. The sidebar on the left lists various services under 'WEB MANAGER', including SDK, DHCP Server, DNS Server, NTP Server, Dynamic DNS, E-mail, Events, SMS, SSH/Telnet Server, SNMP Agent, Web Server, Redundancy, and Voice Gateway.

Figure 5.49.: Voice Gateway Endpoint Configuration

On this page you can activate the endpoints used for voice communication, the following types are supported:

Parameter	Voice Gateway Endpoint Types
Voice-Over-Mobile	Endpoint for GSM/UMTS/LTE calls (can be used for calls to mobile or landline phones)
SIP (registrar)	SIP endpoint which can be a client registered to our registrar
SIP (direct)	Endpoint for calls directly routed to a SIP agent without registration

Parameter	Voice Gateway Endpoint Types
SIP (user-agent)	Endpoint acting as SIP user agent towards a remote registrar

Based on your equipment, we recommend to adjust the modem’s audio profile for a better sound experience. The following profiles are available:

Parameter	Voice-Over-Mobile Audio Profiles
Handset	Provides a mild echo, short delay (less than 16-ms dispersion). This mode is intended for use with a well-designed handset, where the Echo Return Loss (ERL) is generally high. Full-duplex performance is easiest to achieve in this mode.
Headset	Provides a moderate echo, short delay (less than 16-ms dispersion). This mode is intended for use in situations where the echo may be loud but low in delay. There are a variety of different headsets available with a wide variety of echo characteristics and noise pickup. Although the echo delay is typically short (< 16 ms) with all headsets, the echo return loss characteristics can vary significantly and are not well known a priori to the handset designer. This mode is more robust and more aggressive at echo cancellation.
Speakerphone	Handle situations of loud echo with extreme acoustic distortion. This mode is intended for use with a car kit or speakerphone applications with high volume and high distortion. Acoustic echo in this situation has negative ERL and is impossible to cancel completely. It operates in a half-duplex manner and will be very aggressive in muting the entire signal to prevent any echo blips from being heard.
Bluetooth	Provides moderate echo, long delay (up to 64-ms dispersion). This mode is intended for bluetooth headsets and carkits which may have DSP processing on board and could give added delay to the system.

Parameter	Endpoint Settings Voice-Over-Mobile
Modem	Specifies the modem which will be used for voice-over-mobile calls
Audio profile	Specifies the modem’s audio profile
Volume level	Specifies the modem’s volume level - 1 = low

Parameter	Endpoint Settings SIP (registrar)
Subscriber	The subscriber name for a registering SIP client
Username	The username for a registering SIP client
Password	The password for a registering SIP client

Parameter	Endpoint Settings SIP (direct)
Subscriber	The subscriber name of the SIP agent
Host	The IP address of the SIP agent
Port	The port of the SIP agent
Username	The username to authenticate at the SIP agent
Password	The password used for authentication

Parameter	Endpoint Settings SIP (user-agent)
Host	The IP address of the remote SIP registrar
Port	The port of the registrar
Domain	The domain name used at the registrar
Subscriber	The subscriber name used at the registrar
Username	The username to authenticate at the registrar
Password	The password used for authentication
Register	Selects whether the user-agent shall register at the registrar
Expires	The expiry time in seconds after registration will be triggered again

Voice Routing

This page can be used to configure generic voice routing between the endpoints.

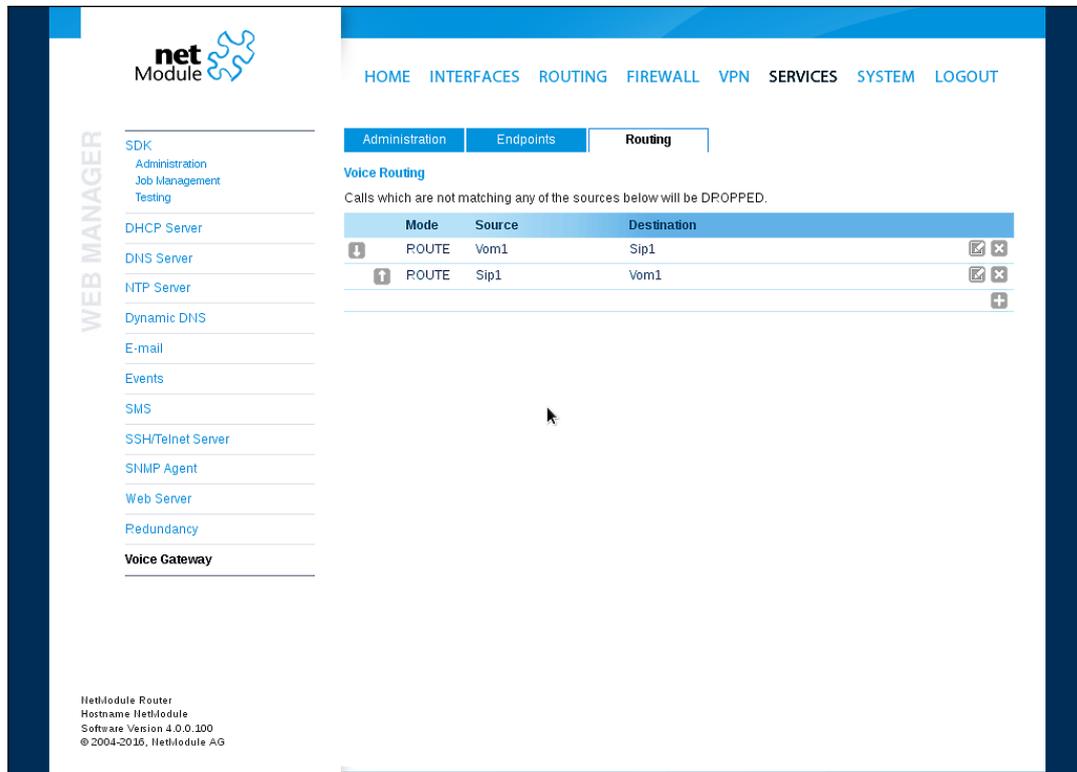


Figure 5.50.: Voice Gateway Routing Configuration

Enhanced routing facilities are provided via the SDK interface which is able to dispatch voice calls based on their attributes (such as phone number) and other system related status information (e.g. number/duration of calls per endpoint, registration status and so on). Using the SDK, you can also initiate or accept a call, adjust its volume level or do a hangup. Anyway, for simple scenarios the generic method should be sufficient and can be configured as follows:

Parameter	Voice Gateway Routing Settings
Source	Specifies the source endpoint (i.e. where the call comes in)
Mode	The type of action which shall be applied for the call: DROP will silently hangup the call, ROUTE will route the call to the specified endpoint.
Destination	Specifies the target endpoint (i.e. where to call is routed to)

Client Configuration

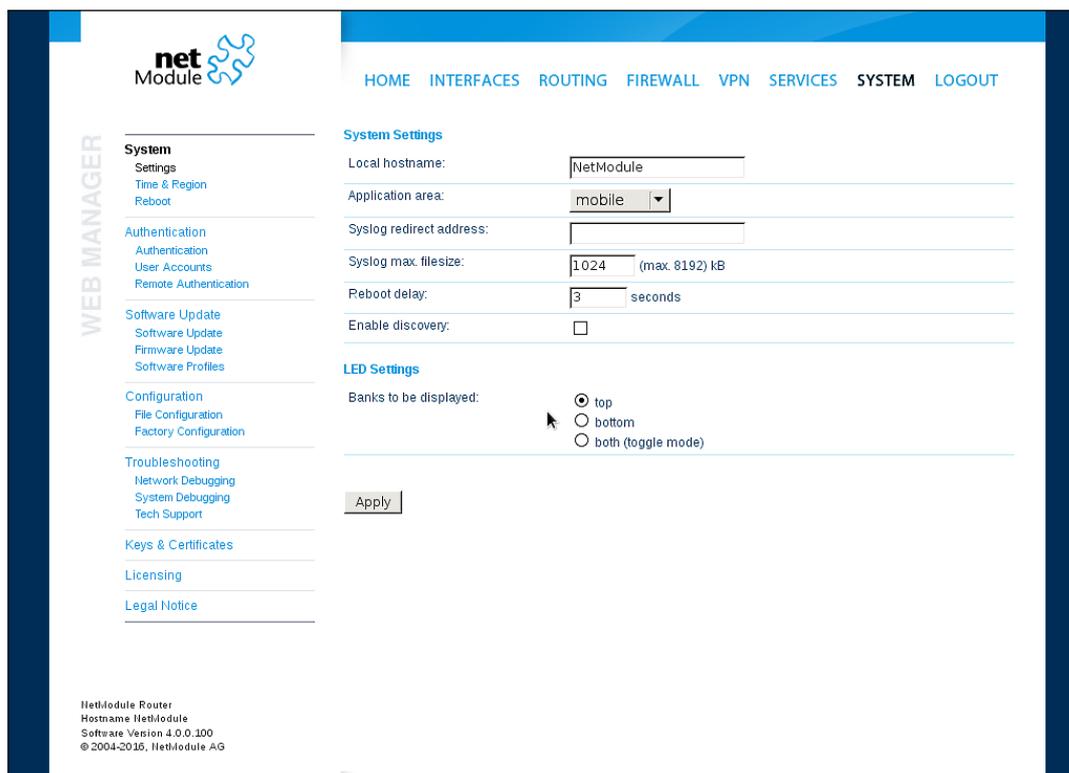
Any SIP client must be configured to use the router as its registrar/proxy.

Parameter	X-Lite Configuration
User ID	SIP username used in from headers (i.e. subscriber name)
Domain	SIP Domain used in from headers (optional)
Authorization name	Username used for authentication (i.e. subscriber name)
Password	Password used for authentication
Display name	Name to be displayed on the handset

5.8. SYSTEM

5.8.1. System

System Settings



The screenshot shows the NetModule WEB MANAGER interface. On the left is a vertical 'WEB MANAGER' sidebar with a navigation menu. The main content area is titled 'System Settings' and contains the following fields:

- Local hostname: NetModule
- Application area: mobile (dropdown menu)
- Syslog redirect address: (empty field)
- Syslog max. filesize: 1024 (max. 8192) kB
- Reboot delay: 3 seconds
- Enable discovery:

Below these are the 'LED Settings' section with radio buttons for 'Banks to be displayed':

- top
- bottom
- both (toggle mode)

An 'Apply' button is located at the bottom of the settings area. At the bottom left of the page, the footer reads: 'NetModule Router, Hostname: NetModule, Software Version 4.0.0.100, © 2004-2016, NetModule AG'.

Figure 5.51.: System

System

The following system parameters can be set:

Parameter	System Settings
Local hostname	The hostname of the system
Application area	The desired application area which influences the system behaviour such as registration timeouts or other adaptations when operating in mobile environments.
Reboot delay	The number of seconds which will be waited before regular system reboots (might be needed for system-rebooting events)

Parameter	System Settings
Enable TCP timestamps	Enable TCP timestamps for system wide TCP communication. This is needed for Protection Against Wrapped Sequence numbers (PAWS), but with these timestamps enabled a remote attacker can guess the uptime of the system. The uptime is a lower bound for the age of the main system components like the kernel. If the system has an uptime of 3 years it's unlikely that recent security patches were applied.

Syslog

The following syslog parameters can be set:

Parameter	Syslog Settings
Storage	The storage device on which log files shall be stored.
Max. filesize	The maximum size of the log files (in kB) until they will get rotated.
Redirect address	Specifies an IP address to which log messages should be redirected to. A tiny system log server for Windows is included in TFTP32 which can be downloaded from our website.

In general, the box comes with an internal flash device which can be used to store data. Depending on your model this can be extended by additional flash or USB disks. The following storage devices exist:

Parameter	Storage Devices
flash root	The root partition of the internal flash
flash data	The data partition of the internal flash
extended disk	An extended storage disk
USB disk	A storage disk connected to the external USB port

LEDs

The following LED parameters can be set:

Parameter	LED Settings
Banks to be displayed	You can configure the behavior of the status LEDs on the front panel of your device. They are usually divided into two banks (top/bottom) and are either indicating the connection status or the digital IO port status. You may configure toggle mode, so that the LEDs periodically cycle between the two states.

Bootloader

The following bootloader parameters can be set:

Parameter	Bootloader Settings
Password	The password used to unlock the bootloader. If empty, the admin password will be used.

Time & Region

This page can be used for setting the system time and configuring the time zone. You may further enable daylight saving changes for your specific time zone. NetModule routers can synchronize their system time by using one or more servers by the help of the Network Time Protocol (NTP) or via GNSS. If enabled, the time synchronization is usually triggered after a WAN link has come up but before starting any VPN connections. Further time synchronization cycles are scheduled in background.

Most routers don't have a battery backed clock (RTC). In this case the system time is set during boot to the last valide time, e.g. before power off.

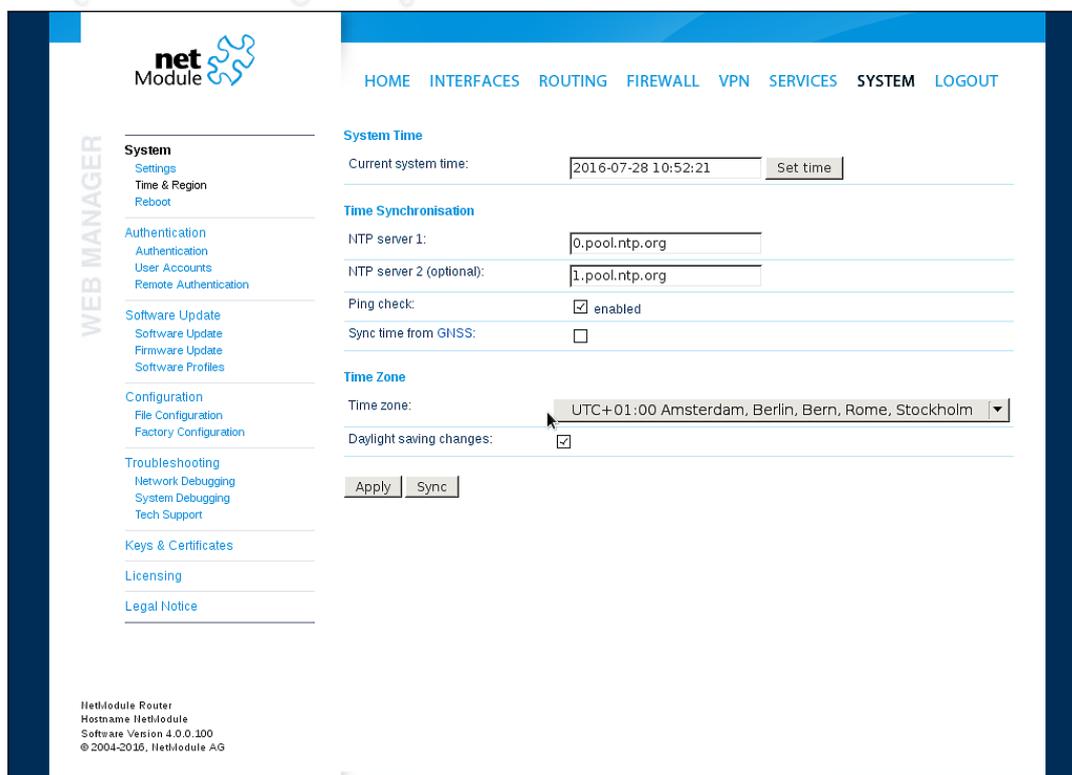


Figure 5.52.: Regional settings

Parameter	Time Synchronisation
NTP server	Address of the primary NTP server
NTP server 2	Optionally, the address of a second NTP server
Ping check	Uses an ICMP ping to check whether NTP servers are available when running initial time update
Sync time from GNSS	Derive time from first GNSS device (if enabled)

Parameter	Time Zone
Time Zone	Set the local time zone.
Daylight saving changes	Enable/disable daylight saving changes.

Virtualization

Virtualization techniques can be used to run multiple isolated guests on top of the host system. The netmodule routers use OS-level virtualization: A system is virtualized at the operating system level, enabling multiple isolated user-space instances called containers. The same operating system kernel is used to implement the guest environments, applications running in a guest environment view it as a stand-alone system.

General settings:

Parameter	Virtualization Settings
Administrative status	Defines whether virtualization is enabled or not

The following parameters can be used to configure a virtual guest:

Parameter	Guest Settings
Type	Defines which virtualization technique is being used
Description	The description of the guest
Storage	Specifies the storage device on which the root file system of the guest will be located

To Install a root-file-system you can set up a URL to load a the image from and trigger the installation:

Parameter	Install
URL	The URL to load the image from. The Image needs to be provided as XZ compressed TAR archive containing the files of a root FS compatible with our CPU architecture (armv7l). Different protocols may be used for the transaction like 'http://' or 'https://', 'ftp://' or 'tftp://'. If you uploaded the the image to the router in advance you can also use 'file://' followed by the local path of the file. We can provide various tailored Linux distribution images (such as Debian) on demand.
Install	If this trigger is set the image download will start on apply. Any existing root file system will be overwritten. This parameter will not be stored in the configuration. After the installation was proceeded the value will be reset and needs to be set again if a new image shall be installed.

Communication to and from the guest can be achieved by defining network interfaces which can be either routed towards the guest or bridged with a LAN interface:

Parameter	Guest Networking
Guest interface	The name of the interface inside the guest
Mode	The network mode for this interface (either routed or bridged)
Address	The IP address of the interface inside the guest
Netmask	The netmask of the interface inside the guest
Gateway	The gateway used inside the guest which is also set on the host interface

Parameter	Guest Networking
Bridge interface	The interface to which the guest interface shall be bridged

The guest devices parameter shows a list of devices (e.g bluetooth, CAN) which can be provided to the guest system.

Parameter	Guest Devices
Enable devices	Enable or disable device for the guest

In order to limit the resources for a guest, the following settings can be applied:

Parameter	Guest Limits
CPU	The number of CPUs used for the guest
Memory	The amount of memory available for the guest

Reboot

This page can be used to set up a periodic automatic reboot but also to trigger a manual reboot which will be issued immediately.

5.8.2. Authentication

This page can be used to define the access model for all management interfaces (e.g. GUI, SSH/telnet server).

Parameter	Authentication Methods
Authentication required	Users can login via HTTP/telnet if authentication succeeds
Secure authentication required	Users can only login via HTTPS/ssh
Secure authentication preferred	Users will be redirected to HTTPS but can sill login via HTTP/telnet

User Accounts

By using this page you can manage the user accounts on the system.

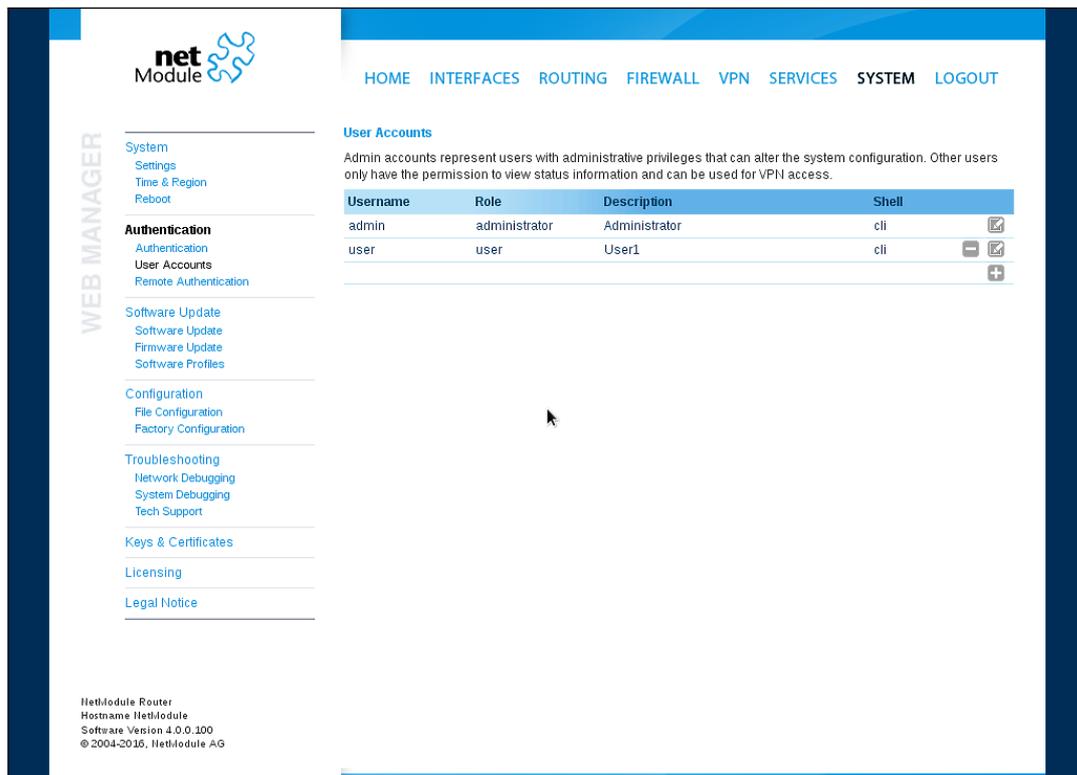


Figure 5.53.: User Accounts

The `admin` user is a built-in power user which represents the default administrator of the system. Please note that the `admin` password will be also applied to the `root` user which is able to enter a system shell. Further admin accounts with administrative privileges can be added, they can also alter the system configuration or perform administrative system tasks. Other users only have the permission to view status information. They can be also used for VPN access.

The Web Manager supports up to 5 concurrent users. Inactive users will be kicked after being idle for 30 minutes. If login was successful, any duplicate users from other remote hosts will be logged out. Remote hosts will be blocked for 5 minutes after 10 failed login attempts.

User accounts management	
Parameter	
Username	The name of the user
Role	Either admin or user
Old password	The old password of the user
New password	The new password of the user
Confirm new password	The confirmed new password of the user

Please note, when adding additional admin users you are required to provide the password of the default administrator.

Remote Authentication

A RADIUS server can be used for authenticating remote users. This applies for the Web Manager, the WLAN network and other services supporting and incorporating remote authentication.

The screenshot displays the 'Remote Authentication' configuration page in the NetModule Web Manager. The interface includes a left-hand navigation menu with categories like System, Authentication, Software Update, Configuration, Troubleshooting, and Keys & Certificates. The main content area has a top navigation bar with links for HOME, INTERFACES, ROUTING, FIREWALL, VPN, SERVICES, SYSTEM, and LOGOUT. Below this, the 'Remote Authentication' section is active, showing 'Administrative Status' as 'enabled' with a radio button. The 'RADIUS Configuration' section contains the following settings: 'Server address' set to 192.168.1.200, 'Secret' represented by a masked field of ten dots, 'Authentication port' set to 1812, 'Accounting port' set to 1813, and 'Use for login' checked with a checkbox. An 'Apply' button is located below the settings. The bottom left corner of the page displays version information: 'NetModule Router', 'Hostname: NetModule', 'Software Version: 4.0.0.100', and '©2004-2016, NetModule AG'.

Figure 5.54.: Remote Authentication

It can be configured as follows:

Parameter	Remote authentication settings
Administrative status	Defines whether a remote server should be used for authentication
RADIUS server	The RADIUS server address
RADIUS secret	The secret used to authenticate against the RADIUS server
Authentication port	The port used for authentication
Accounting port	The port used for accounting messages
Use for login	This option enables remotely-defined users to access the Web Manager, otherwise it is only used by services which have explicitly configured it (e.g. WLAN)

5.8.3. Software Update

Manual Software Update

This menu can be used to run a manual software update of the system.

Parameter	Manual Software Update
Update operation	The update operation method being used. You can upload the image, download it from an URL or use the latest version from our server
URL	The server URL where the software update image should be downloaded from

An Uniform Resource Locator (URL) can have the following format:

```
http://<username>:<password>@<host>:<port>/<path>
https://<username>:<password>@<host>:<port>/<path>
ftp://<username>:<password>@<host>:<port>/<path>
sftp://<username>:<password>@<host>:<port>/<path>
tftp://<host>/<path>
file:///<path>
```

When issuing a software update, the current configuration (including files like keys/certificates) will be backed up. Any other modifications to the filesystem will be erased.

The configuration is generally backward-compatible. We also apply forward compatibility when downgrading to a previous software within the same release line, which is accomplished by sorting out unknown configuration directives which actually may lead to loss of settings and features. Therefore, it's always a good idea to keep a copy of the working configuration.



Attention

In case you perform a major downgrade with a previous release line (e.g. 3.7.0 to 3.6.0), please ensure to always use the latest release of that branch (i.e. 3.6.0.X) as only those tend to be fully forward-compatible. Also keep in mind, that some hardware features may not work (e.g. if not implemented in that version). In doubt, please consult our support team.

A software image can be either uploaded via the Web Manager or retrieved from a specific URL. It will be unpacked and deployed to a spare partition which gets activated if the update completed successfully. The whole procedure is accompanied by all green LEDs flashing up, the subsequent system reboot gets denoted by a slowly blinking Status LED. The backed-up configuration will be applied at bootup and the Status LED will blink faster during this operation. Depending on your configuration, this may take a while.

Automatic Software Update

This menu can be used to run a automatic software update of the system.

Parameter	Automatic software update
Status	Enable/disable automatic software update
Time of day	Every day at this time the router will do a check for updates
Operation	Download latest image from the the server or specify the URL where the software update package should be downloaded from. Supported protocols are TFTP, HTTP, HTTPS, and FTP. Provide a URL like <code>protocol://server/path/file</code>

Remark: SSL certificates of HTTPS URLs will be only verified if a list of CA root certificates are provided under [5.8.8](#).

After the new software has been installed, the latest running configuration will be applied afterwards during bootup. This is indicated by a faster green blinking of the Status LED.

5.8.4. Module Firmware Update

This menu can be used to perform a firmware update of a specific module.

Parameter	Module Firmware Update
Update operation	The update operation method being used. You can either upload a firmware package or download it from a specifc URL.
Module	The module which shall be updated.
Storage	The temporary storage which shall be used for the update procedure. For boxes with limited amount of flash it is possible to use an USB stick which must be properly set up in the USB section and hold a proper filesystem such as ext4.
URL	The server URL where the firmware package should be downloaded from (e.g. <code>protocol://server/path/file</code>). Supported protocols are TFTP, HTTP, HTTPS, and FTP. For boxes with limited amount of flash you may also use <code>usb0://<path-to-firmware-package></code> .

A firmware package (ZIP) usually consists of a flash utility, an info file and the corresponding firmware files. Please follow <http://www.netmodule.com/support/supportform.aspx> in order to get the latest version.

5.8.5. Software Profiles

The system consists of two root partitions which can hold different software versions and this menu can be used to switch between them. By doing so, you can test a newer software version and simply switch-back if things go wrong.

5.8.6. Configuration

Configuration via the Web Manager becomes tedious for larger volumes of devices. The router therefore offers automatic and manual file-based configuration to automate things. Once you have successfully set up the system you can back up the configuration and restore the system with it afterwards. You can either upload a single configuration file (.cfg) or a complete package (.zip) containing the configuration file and a packed version of other essential files (such as certificates) in the root directory.

Manual File Configuration

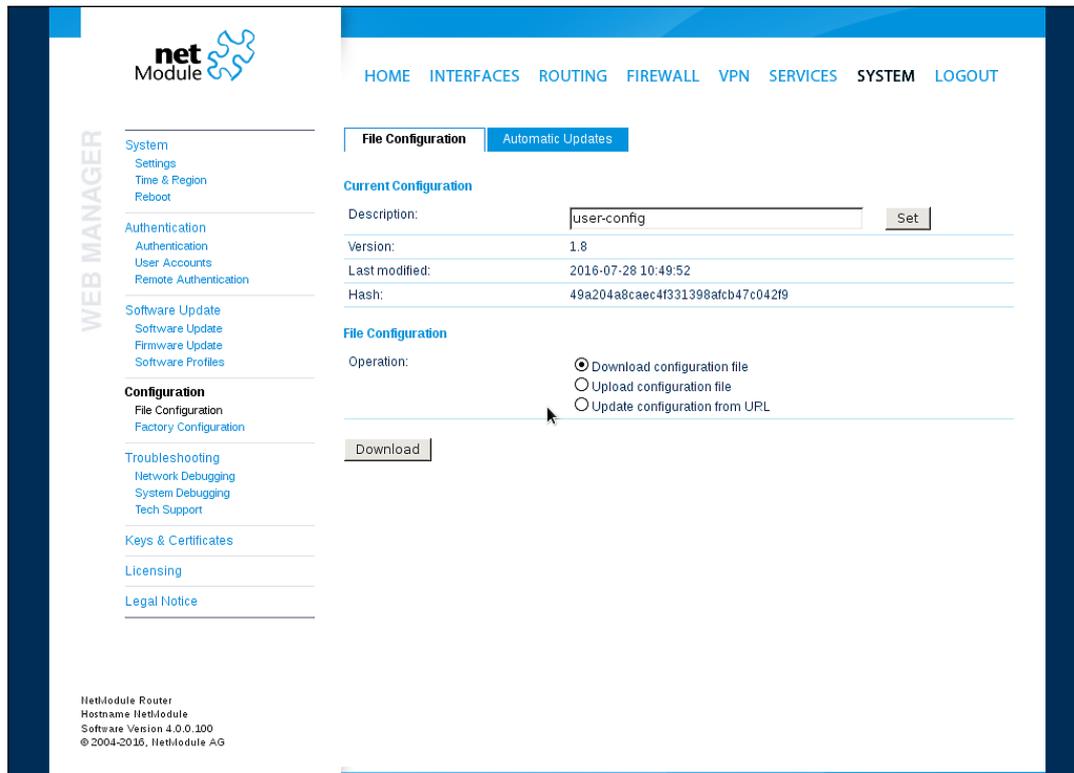


Figure 5.55.: Manual File Configuration

This section can be used to download the currently running system configuration (including essential files such as certificates). In order to restore a particular configuration you can upload a configuration previously downloaded. You can choose between missing configuration directives set to factory defaults or getting ignored, that means, potentially existing configuration directives will be kept at the system.

Automatic File Configuration

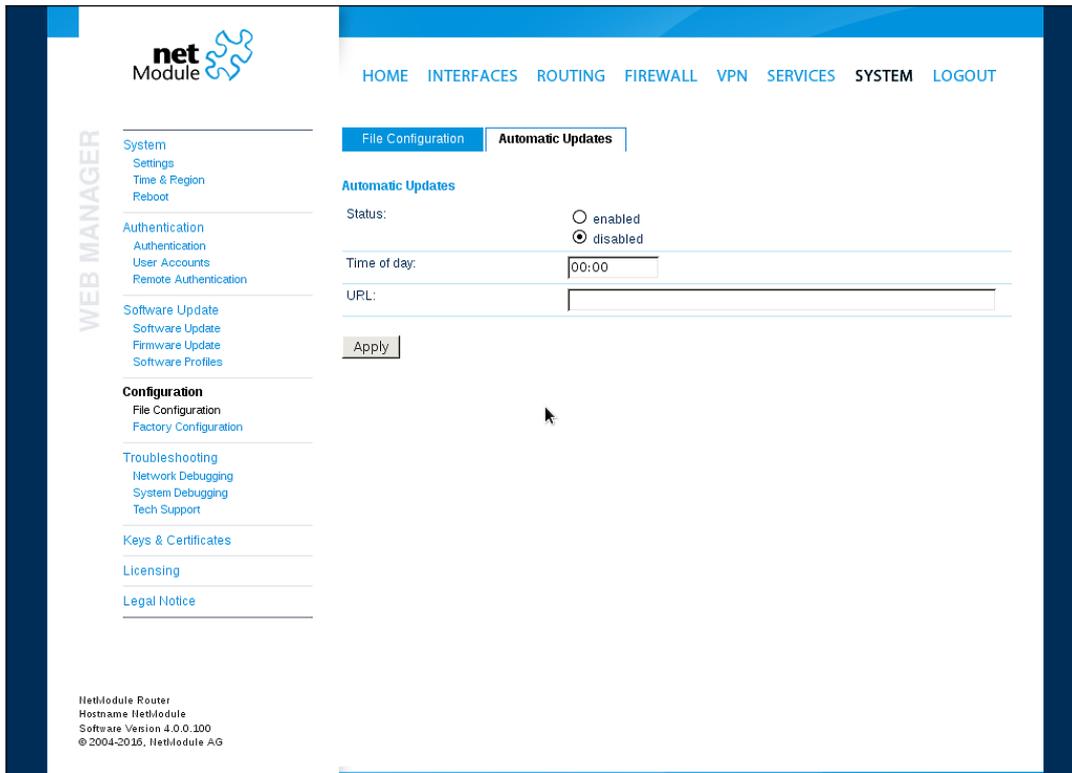


Figure 5.56.: Automatic File Configuration

This menu can be used to run an automatic configuration update of the system. It is configured as follows:

Parameter	Automatic File Configuration
Status	Enable/disable an automatic configuration update
Time of day	Time of day when the system should check for updates
URL	The URL where the configuration file should be retrieved from (supported protocols are HTTP, HTTPS, TFTP, FTP)

Factory Configuration

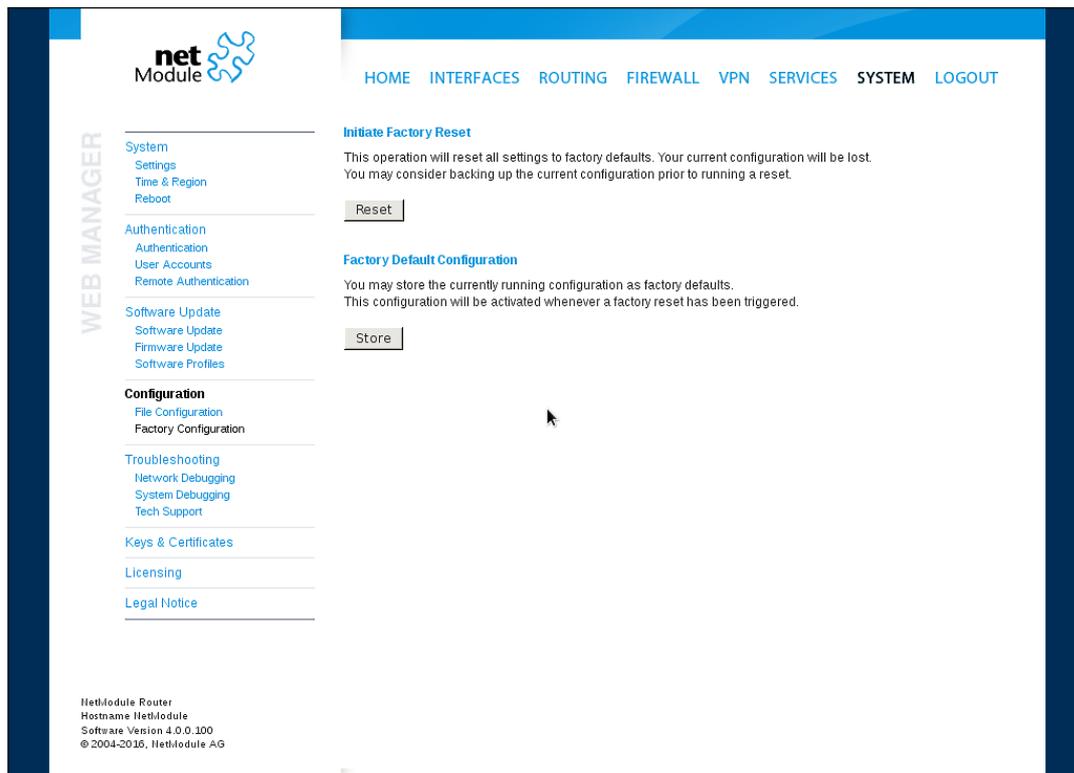


Figure 5.57.: Factory Configuration

This menu can be used to reset the device to factory defaults. Your current configuration will be lost. This procedure can also be initiated by pressing and holding the *Reset* button for at least five seconds. A successfully initiated factory reset can be noticed by all LEDs having been turned on. The factory reset will set the IP address of the first Ethernet interface back to 192.168.1.1. You will be able to communicate again with the device using the default network parameters. You may store the currently running configuration as factory defaults which will reside active even when a factory reset has been initiated (e.g. by your service staff).

Please ensure that this corresponds to a working configuration. A real factory reset to the default settings can be achieved by restoring the original factory configuration and initiating the factory reset again.

5.8.7. Troubleshooting

Network Debugging

There are several tools for network debugging like ping, traceroute, tcpdump and darkstat.

Parameter	Automatic software update
Ping	The ping utility can be used to verify whether a remote host can be reached via IP.
Time of day	The traceroute utility can be used to print the route packets trace to a remote host.
Tcpdump	The tcpdump utility generates a network capture (PCAP) of an interface which can be later analyzed with Wireshark.
Darkstat	The darkstat utility can be used to visualize your current network connections and traffic on a particular interface.

System Debugging

You can view the system log here by selection the option *Debug log* or if you are interested in the boot log select *Boot log*.

Another way to see what is going on on the box is opening a SSH or Telnet session as *root* and typing `tail -log`. Furthermore the system log can be redirected to a syslog server, see section 5.8.1.

The screenshot shows the NetModule web interface. On the left is a 'WEB MANAGER' sidebar with a navigation menu including System, Authentication, Software Update, Configuration, Troubleshooting, Keys & Certificates, Licensing, and Legal Notice. The main content area is titled 'System Debugging' and features a 'Log Viewer' tab. Below the tabs, there are dropdown menus to show 'all' logs of the 'system log'. The log viewer displays a list of system messages, including errors, debugs, and info messages, with timestamps and details about network link management and WAN configuration.

```
Jul 28 10:52:18 NetModule user.err qosd[3035]: cmd '/usr/sbin/tc class change dev  
wwan0 classid 1:1 htb rate 0.00200mbit' failed  
Jul 28 10:52:18 NetModule user.debug wwan-manager[2991]: wwan0: getting temp  
Jul 28 10:52:18 NetModule user.debug wwan-manager[2991]: wwan0: sending 'get temp'  
to wwannd  
Jul 28 10:52:18 NetModule user.info wwan-manager[2991]: wwan0: temp is '40.8'  
Jul 28 10:52:19 NetModule user.debug link-manager[32008]: triggering permanent link  
wanlink0  
Jul 28 10:52:19 NetModule user.debug link-manager[32008]: triggering permanent link  
wanlink1  
Jul 28 10:52:19 NetModule user.debug link-manager[32008]: current hotlink wanlink0  
is best  
Jul 28 10:52:19 NetModule user.info link-manager[32008]: wanlink0: supervising  
permanent link (state up, triggered 1, not suspended)  
Jul 28 10:52:19 NetModule user.debug link-manager[32008]: wanlink0: signal strength  
for card1 changed from -51 to -57  
Jul 28 10:52:19 NetModule user.info link-manager[32008]: wanlink1: supervising  
permanent link (state up, triggered 1, not suspended)  
Jul 28 10:52:19 NetModule user.info wwan-manager[2991]: wwan1: querying me909 and  
sim0  
Jul 28 10:52:19 NetModule user.debug wwan-manager[2991]: wwan1: getting regstate  
Jul 28 10:52:19 NetModule user.debug wwan-manager[2991]: wwan1: sending 'get  
regstate' to wwannd  
Jul 28 10:52:19 NetModule user.info wwan-manager[2991]: wwan1: regstate is  
'registeredInHomeNetwork'  
Jul 28 10:52:19 NetModule user.debug wwan-manager[2991]: wwan1: getting operator  
Jul 28 10:52:19 NetModule user.debug wwan-manager[2991]: wwan1: sending 'get  
operator' to wwannd  
Jul 28 10:52:20 NetModule user.err qosd[3035]: rawping[wwan0]: invalid gateway  
'100.z□□□.□□□.□□□.□□□'  
Jul 28 10:52:20 NetModule user.notice qosd[3035]: [wwan0] ping (mark=20) to 8.8.8.8  
failed. set rtt(20)=2000.00000 ms  
Jul 28 10:52:20 NetModule user.info wwan-manager[2991]: wwan1: operator is  
'Telekom.de, 26201, FFFE, 192C902'  
Jul 28 10:52:20 NetModule user.debug wwan-manager[2991]: wwan1: current network is  
'Telekom.de'  
Jul 28 10:52:20 NetModule user.debug wwan-manager[2991]: wwan1: current LAI is  
'26201'  
Jul 28 10:52:20 NetModule user.debug wwan-manager[2991]: wwan1: current LAC is 'FFFE'  
Jul 28 10:52:20 NetModule user.debug wwan-manager[2991]: wwan1: current CID is  
'192C902'
```

Figure 5.58.: Log Viewer

Tech Support

You can generate and download a tech support file here. We strongly recommend providing this file when getting in touch with our support team, either by e-mail or via our on-line support form, as it would significantly speed up the process of analyzing and resolving your problem. Log files can be viewed a downloaded and reset here. Please study them carefully in case of any issues. Various tools reside on this page for further analysis of potential configuration issues.

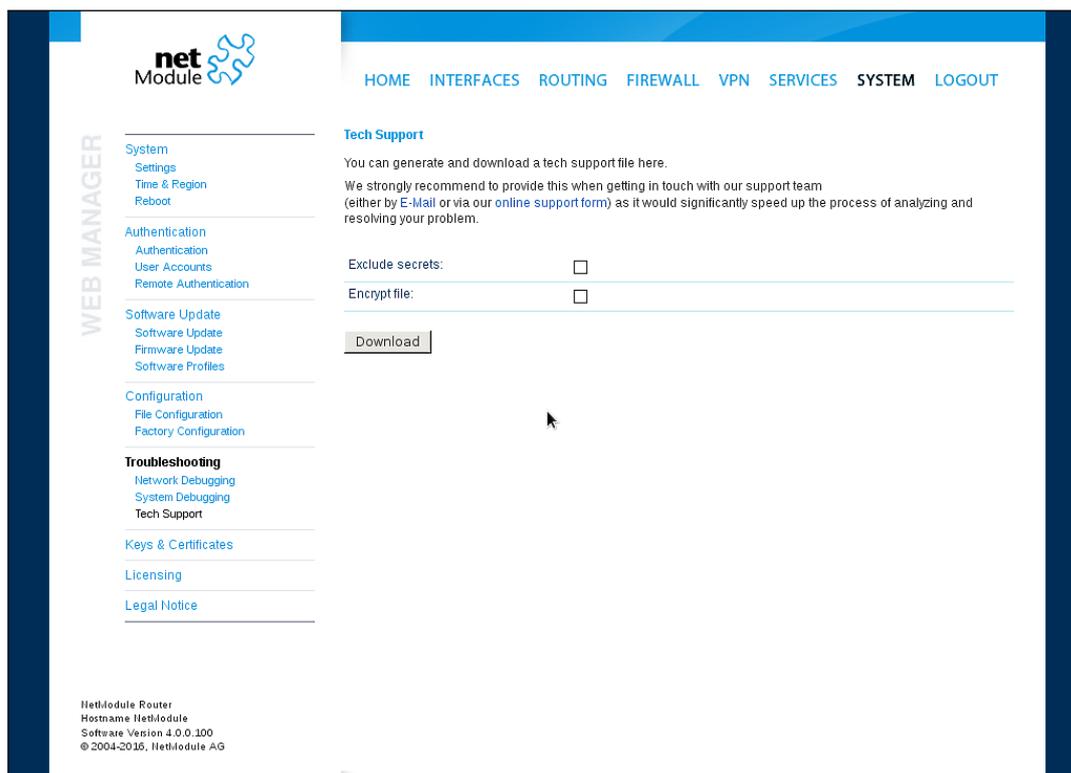


Figure 5.59.: Tech Support File

It is possible to trace any IP interface and inspect individual packet flows between hosts. This can be achieved by logging onto the box and start a network packet capture by using the tool *tcdump*. We recommend to use the `-n` switch to bypass name resolution (e.g. `tcpdump -n -i lan0`). You may also generate a dump in PCAP format using the Web Manager, download it to your computer and perform further inspections with Wireshark (available at www.wireshark.org).

5.8.8. Keys and Certificates

The key and certificate page lets you generate required files for securing your services (such as HTTP and SSH server) but also to implement authentication and encryption for certificate-based VPN tunnels and WLAN clients.

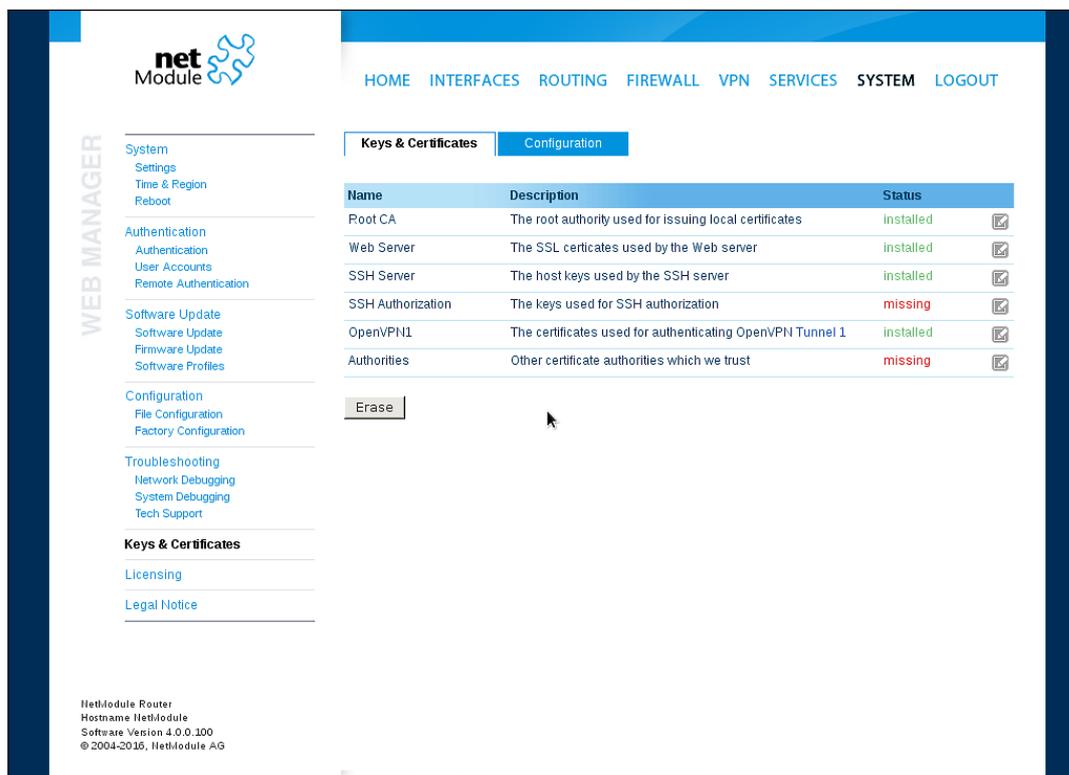


Figure 5.60.: Keys and certificates

The entry pages shows an overview about installed keys and certificates. The following sections may appear:

Type	Description
Root CA	The root Certificate Authority (CA) which issues certificates, its key can be used to certify it at trusted third party on other systems
Web Server	The certificates for the Web server required for running HTTP over SSL (HTTPS).
SSH Server	The DSS/DSA keys for the SSH server.
SSH Authorization	The keys used for SSH authorization.
OpenVPN	Server or client keys and certificates for running OpenVPN tunnels.

Type	Description
IPsec	Server or client keys and certificates for running IPsec tunnels.
WLAN	Keys and certificates for implementing certificate-based WLAN authentication (e.g. WPA-EAP-TLS).
Authorities	Other certificate authorities which we trust when establishing SSL client connections.

Table 5.141.: Certificate Sections

For each certificate section it is possible to perform the following operations:

Operation	Description
generate locally	Generate key and certificate locally on the box (see 5.8.8 for more options)
upload files	Key and certificate will be uploaded. We support files in PKCS12, PKCS7, PEM/DER format as well as RSA/DSS keys in OpenSSH or Dropbear format.
enroll via SCEP	Enroll key and certificate via SCEP (see 5.8.8 for more options)
download certificate	Download key and certificate in ZIP format (files will be encoded in PEM format)
create signing request	Generate key locally and create a signing request to retrieve a certificate signed by another authority
erase certificate	Erase all keys and certificates associated with this section

Table 5.142.: Certificate Operations

Configuration

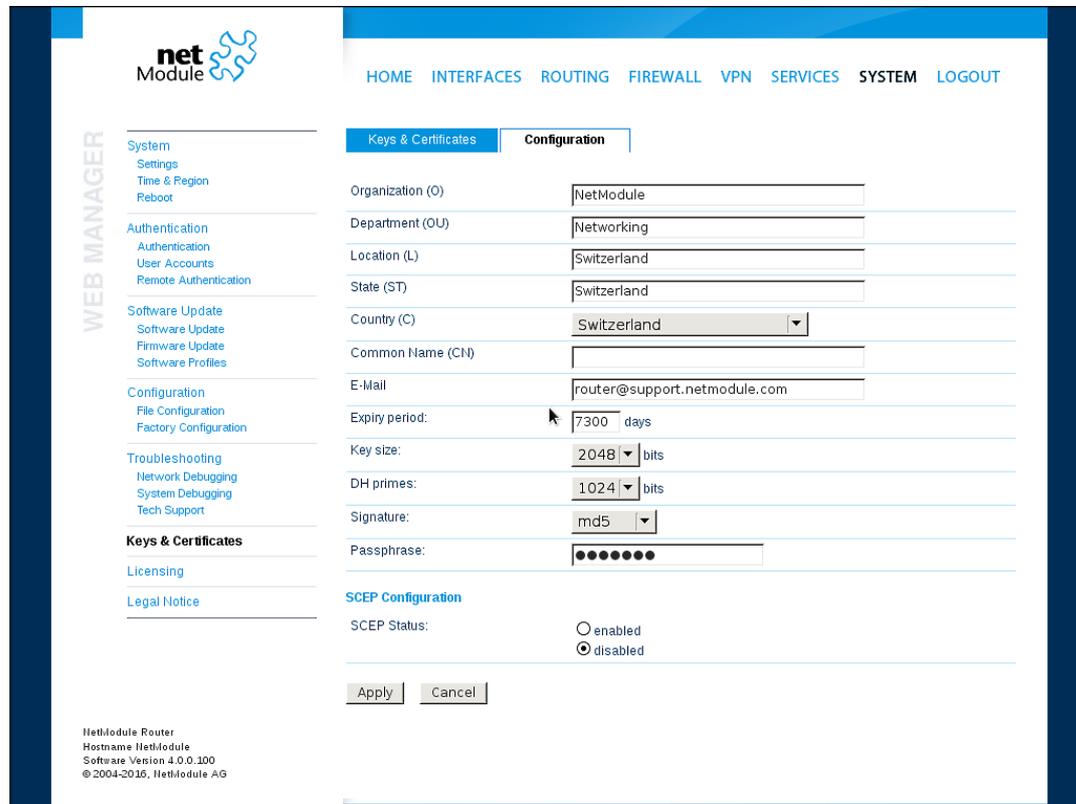


Figure 5.61.: Certificate Configuration

This page provides some general configuration options which will be applied when operating on keys and certificates.

If keys, certificates and signing requests are generated locally, the following settings will be taken into account:

Parameter	Certificate Configuration
Organisation (O)	The certificate owner’s organization
Department (OU)	The name of the organizational unit to which the certificate issuer belongs
Location (L)	The certificate owner’s location
State (ST)	The certificate owner’s state
Country (C)	The certificate owner’s country (usually a TLD abbreviation)
Common Name (CN)	The certificate owner’s common name, mainly used to identify a host
E-Mail	The certificate owner’s email address

Parameter	Certificate Configuration
Expiry period	The number of days a certificate will be valid from now on
Key size	The length of the private key in bits
DH primes	The number of bits for custom Diffie-Hellman primes
Signature	The signature algorithm when signing certificates
Passphrase	The passphrase for accessing/opening a private key

Please be aware of the fact, that the local random number generator (RNG) provides pretty good randomness for most applications. If stronger cryptography is mandatory, we suggest to create the keys at an external RNG device or manage all certificates completely on a remote certification server. Nevertheless, using a local certificate authority can issue and manage all required certificates and also run a certificate revokation list (CRL).

When importing keys, the certificate and key file can be uploaded individually encoded in PEM/DER or PKCS7 format. All files (CA certificate, certificate and private key) can also be uploaded in one stroke by using the container format PKCS12. RSA/DSS keys can be converted from OpenSSH or Dropbear formats. It is possible to specify the passphrase for opening the private key. Please note that the system will generally apply the system-wide certificate passphrase on a key when installing the certificate. Thus, changing the general passphrase will result in all local keys getting equipped with the new one.

SCEP Configuration

If certificates are getting enrolled by using the Simple Certificate Enrollment Protocol (SCEP) the following settings can be configured:

Parameter	SCEP Configuration
SCEP status	Specifies whether SCEP is enabled or not
URL	The SCEP URL, usually in the form <code>http://<host>/<path>/pkiclient.exe</code>
CA fingerprint	The fingerprint of the certificate used to identify the remote authority. If left empty, any CA will be trusted.
Fingerprint algorithm	The fingerprint algorithm for identifying the CA (MD5 or SHA1)
Poll interval	The polling interval in seconds for a certificate request
Request timeout	The max. polling time in seconds for a certificate request
ID type	Can be IP, Email or DNS
Password	The password for the scep server.

When enrolling certificates, the CA certificate will be initially fetched from the specified SCEP URL using the `getca` operation. It will be shown on the configuration page and it has to be verified that it belongs to the correct authority. Otherwise, the CA must be rejected. This part is essential when using SCEP as it builds up the chain of trust.

If a certificate enrollment request times out, it is possible to re-trigger the interrupted enrollment request and it will be resumed using the previously generated key. In case a request has been rejected, you are required to erase the certificate first and then start the enrollment process all over again.

Authorities

For SSL client connections (as used by SDK functions or when downloading configuration/software images) you might upload a list of CA certificates which are considered trusted.

To obtain the CA certificate from a particular site with Mozilla Firefox, the following steps will be required:

- Point the browser to the relevant HTTPS website
- Click the padlock in the address bar
- Click the **More Information** and the **View Certificate** button
- Select the **Details** tab press the **Export** button
- Choose a path for the file (e.g. website.pem)

Certificates from self-signed authorities can also be retrieved by running:

```
echo quit | \  
openssl s_client -showcerts -connect <host>:443 | \  
sed -ne '/-BEGIN CERTIFICATE-/,/-END CERTIFICATE-/p' > other.crt
```

The PEM-encoded X.509 certificate files can be edited and concatenated using a simple editor (if required) and then uploaded to the box. Once installed, an SSL client connection will terminate if verification with any of those CA certificates fails.

5.8.9. Licensing

Certain features of NetModule routers require a valid license to be present in the system, some of them also depend on the mounted modules. Please contact us for getting a valid license for available components and we will provide a license file based on your serial number which can be installed to the router afterwards.

The screenshot shows the NetModule web manager interface. On the left is a 'WEB MANAGER' sidebar with various menu items like System, Authentication, Software Update, Configuration, Troubleshooting, Keys & Certificates, Licensing, and Legal Notice. The main content area has a top navigation bar with links: HOME, INTERFACES, ROUTING, FIREWALL, VPN, SERVICES, SYSTEM, LOGOUT. Below this, the 'License Installation' section is active, showing options to 'Upload license file' (selected) or 'Download license from URL'. A 'License file:' field with a 'Browse...' button shows 'No file selected.' and an 'Install' button is present. The 'Licensing Status' section displays 'Serial number: 00112B0114FE' and 'License status: A valid license is installed.' Below this is a table:

Feature	Availability	Licensing Status
GPS	yes	licensed
GSM	yes	licensed
LTE	yes	licensed
MOBILEIP	yes	licensed
SERVER	yes	unlicensed
UMTS	yes	licensed
VIRT	no	unlicensed
VOICE	yes	licensed
WLAN	yes	licensed

At the bottom left, footer text reads: NetModule Router, Hostname: NetModule, Software Version 4.0.0.100, © 2004-2016, NetModule AG.

Figure 5.62.: Licensing

5.8.10. Legal Notice

OSS Notice

We inform you that NetModule products may contain in part open-source software. We are distributing such open-source software to you under the terms of GNU General Public License (GPL), GNU Lesser General Public License (LGPL) or other open-source licenses.

These licenses allow you to run, copy, distribute, study, change and improve any software covered by GPL, Lesser GPL, or other open-source licenses without any restrictions from us or our end user license agreement on what you may do with that software. Unless required by applicable law or agreed to in writing, software distributed under open-source licenses is distributed on an "AS IS" basis, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

To obtain the corresponding open source codes covered by these licenses, please contact our technical support at router@support.netmodule.com.

Acknowledgements

This product includes PHP, freely available from <http://www.php.net>.

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (<http://www.openssl.org/>).

This product includes cryptographic software written by Eric Young(eay@cryptsoft.com).

This product includes software written by Tim Hudson (tjh@cryptsoft.com).

This product includes software written Jean-loup Gailly and Mark Adler.

This product includes software MD5 Message-Digest Algorithm by RSA Data Security, Inc.

This product includes an implementation of the AES encryption algorithm based on code released by Dr Brian Gladman.

Multiple-precision arithmetic code originally written by David Ireland

Software from The FreeBSD Project (www.freebsd.org)

Copyright (C) 2020, NetModule. All rights reserved.

5.9. LOGOUT

Please use this menu to log out from the Web Manager.

6. Command Line Interface

The Command Line Interface (CLI) offers a generic control interface to the router and can be used to get/set configuration parameters, apply updates, restart services or perform other system tasks.

It will be started automatically in interactive mode when logging in as *admin* user or by running `cli -i`. However, the same syntax can be used when calling it from the system shell. A list of available commands can be displayed by running `cli -l`.

The CLI supports TAB completion, that is expanding entered words or fragments by hitting the TAB key at any time. This applies to commands but also to some arguments and generally offers a convenient way for working on the shell.

Please note that each CLI session will perform an automatic logout as soon as a certain time of inactivity (10 minutes by default) has been reached. It can be turned off by the command `no-autologout`.

6.1. General Usage

When operating the CLI in interactive mode, each entered command will be executed by the RETURN key. You can use the Left and Right keys to move the current point between entered characters or use the Up and Down keys to search the history of entered commands. Typing `exit` as well as pressing CTRL-c twice or CTRL-d on an empty command line will exit the CLI.

List of supported key sequences:

Key Sequence	Action
CTRL-a	Move to the start of the current line
CTRL-e	Move to the end of the line
CTRL-f	Move forward a character
CTRL-b	Move back a character
ALT-f	Move forward to the end of the next word
ALT-b	Move back to the start of the current or previous word
CTRL-l	Clear the screen leaving the current line at the top of the screen; with an argument given, refresh the current line without clearing the screen
CTRL-p	Fetch the previous command from the history list, moving back in the list
CTRL-n	Fetch the next command from the history list, moving forward in the list
ALT-<	Move to the first line in the history
ALT->	Move to the end of the input history

Key Sequence	Action
CTRL-r	Search backward starting at the current line and moving up through the history
CTRL-s	Freeze session
CTRL-q	Reactivate frozen session
CTRL-d	Delete character at point or exit CLI if at the beginning of the line
CTRL-t	Drag the character before point forward moving point forward as well; if point is at the end of the line, then this transposes the two characters before the point
ALT-t	Drag the word before point past the word after point, moving point over that word as well. If point is at the end of the line, this transposes the last two words on the line.
CTRL-k	Delete the text from point to the end of the line
CTRL-y	Yank the top of the deleted text into the buffer at point

Please note, that it can be required to apply quotes ("") when entering commands with arguments containing whitespaces.

6.2. Print Help

The `help` command can be used to get the list of available commands when called without arguments, otherwise it will print the usage of the specified command.

```
> help
```

```
Usage:
```

```
    help [<command>]
```

```
Available commands:
```

```

get          Get config parameters
set          Set config parameters
done         Check done
update       Update system facilities
cert         Manage keys and certificates
status       Get status information
scan         Scan networks
send         Send message, mail, techsupport or ussd
restart      Restart service
debug        Debug system
reset        Reset system to factory defaults
reboot       Reboot system
shell        Run shell command
help         Print help for command
no-autologout Turn off auto-logout
history      Show command history
```

```
exit
```

```
Exit
```

6.3. Getting Config Parameters

The `get` command can be used to get configuration values.

```
> get -h
```

Usage:

```
get [-hsvfc] <parameter> [<parameter>..]
```

Options:

```
-s          generate sourceable output
-v          validate config parameter
-f          get factory default rather than current value
-c          show configuration sections
```

6.4. Setting Config Parameters

The `set` command can be used to set configuration values.

```
> set -h
```

Usage:

```
set [-hv] <parameter>=<value> [<parameter>=<value>..]
```

Options:

```
-v          validate config parameter
```

6.5. Checking Config Completed

The `done` command can be used to check if all modify scripts have completed after a config change.

```
> done -h
```

Usage:

```
done [-h]
```

6.6. Getting Status Information

The `status` command can be used to get various status information of the system.

```
> status -h
```

Usage:

```
status [-hs] <section>
```

Options:

`-s` generate sourceable output

Available sections:

<code>summary</code>	Short status summary
<code>info</code>	System and config information
<code>config</code>	Current configuration
<code>system</code>	System information
<code>configuration</code>	Configuration information
<code>license</code>	License information
<code>wwan</code>	WWAN module status
<code>wlan</code>	WLAN module status
<code>gnss</code>	GNSS (GPS) module status
<code>eth</code>	Ethernet interface status
<code>lan</code>	LAN interface status
<code>wan</code>	WAN interface status
<code>openvpn</code>	OpenVPN connection status
<code>ipsec</code>	IPsec connection status
<code>pptp</code>	PPTP connection status
<code>gre</code>	GRE connection status
<code>dialin</code>	Dial-In connection status
<code>mobileip</code>	MobileIP status
<code>dio</code>	Digital IO status
<code>audio</code>	Audio module status
<code>can</code>	CAN module status
<code>uart</code>	UART module status
<code>ibis</code>	IBIS module status
<code>redundancy</code>	Redundancy status
<code>sms</code>	SMS status
<code>firewall</code>	Firewall status
<code>qos</code>	QoS status
<code>neigh</code>	Neighborhood status
<code>location</code>	Current Location

6.7. Scanning Networks

The `scan` command can be used to scan for available WWAN and WLAN networks.

```
> scan -h
```

Usage:

```
scan [-hs] <interface>
```

Options:

`-s` generate sourceable output

6.8. Sending E-Mail or SMS

The `send` command can be used to send a message via E-Mail/SMS to the specified address or phone number.

```
> send -h
```

Usage:

```
send [-h] <type> <dest> <msg>
```

Options:

```
<type>      type of message to be sent (mail, sms, techsupport,
ussd)
<dest>      destination of message (mail-address, phone-number or
index)
<msg>       message to be sent
```

6.9. Updating System Facilities

The update command can be used to perform various system updates.

```
> update -h
```

Usage:

```
update [-hfrsn] <software|config|license|sshkeys> <URL>
```

Options:

```
-r          reboot after update
-f          force update
-n          don't reset missing config values with factory defaults
-s          show update status
```

Available update targets:

```
software      Perform software update
firmware      Perform module firmware update
config        Update configuration
license       Update licenses
sshkeys       Install SSH authorized keys
```

You may also run 'update software latest' to install the latest version from our server.

6.10. Manage keys and certificates

The cert command can be used to manage keys and certificates.

```
> cert -h
```

Usage:

```
cert [-h] [-p phrase] <operation> <cert> [<url>]
```

Possible operations:

```
install      install a certificate from specified URL
create       create a certificate locally
```

enroll	enroll a certificate via SCEP
erase	erase an installed certificate
view	view an installed certificate

6.11. Restarting Services

The restart command can be used to restart system services.

```
> restart -h
```

Usage:

```
restart [-h] <service>
```

Available services:

configd	Configuration daemon
dnsmasq	DNS/DHCP server
dropbear	SSH server
firewall	Firewall and NAT
gpsd	GPS daemon
gre	GRE connections
ipsec	IPsec connections
lighttpd	HTTP server
link-manager	WAN links
network	Networking
openvpn	OpenVPN connections
pptp	PPTP connections
qos	QoS daemon
smsd	SMS daemon
snmpd	SNMP daemon
surveyor	Supervision daemon
syslog	Syslog daemon
telnet	Telnet server
usbipd	USB/IP daemon
voiced	Voice daemon
vrrpd	VRRP daemon
wlan	WLAN interfaces
wwan-manager	WWAN manager

6.12. Debug System

The debug command can be used to obtain debug/log messages.

```
> debug -h
```

Usage:

```
debug [-h] <target>
```

Available debug targets:

```
configd
```

```
event-manager  
home-agent  
led-manager  
link-manager  
mobile-node  
qmid  
qosd  
scripts  
sdkhost  
ser2net  
smsd  
surveyor  
swupdate  
system  
voiced  
watchdog  
wwan-manager  
wwanmd
```

6.13. Resetting System

The `reset` command can be used to reset the router back to factory defaults.

```
> reset -h  
Usage:  
    reset [-h]
```

6.14. Rebooting System

The `reboot` command can be used to reboot the router.

```
> reboot -h  
Usage:  
    reboot [-h]
```

6.15. Running Shell Commands

The `shell` command can be used to execute a system shell and run any arbitrary application or script.

```
> shell -h  
Usage:  
    shell [-h] [<cmd>]
```

6.16. Working with History

The `history` command will print the list of entered commands on a per-user basis.

```
> history -h
Usage:
  history [-c]
```

It can be cleared by `history -c`.

6.17. CLI-PHP

CLI-PHP, the HTTP frontend to the CLI application, can be used to configure and control the router remotely. It is enabled in factory configuration, thus can be used for deployment purposes, but disabled as soon as the admin account has been set up.

The service can later be turned on/off by setting the `cliphp.status` configuration parameter:

```
cliphp.status=0      Service is disabled
cliphp.status=1      Service is enabled
```

This section describes the CLI-PHP interface for Version 2. It accepts POST and GET requests. Running with GET requests, the general usage is defined as follows:

Usage:

```
http(s)://cli.php?<key1>=<value1>&<key2>=<value2>..<keyN>=<valueN>
```

Available keys:

```
output      Output format (html, plain)
usr         Username to be used for authentication
pwd         Password to be used for authentication
command     Command to be executed
arg0..arg31 Arguments passed to commands
```

Notes:

The commands correspond to CLI commands as seen by '`cli -l`', the arguments (`arg0..arg31`) will be directly passed to `cli`.

Thus, an URL containing the following sequence:

```
command=get&arg0=admin.password&arg1=admin.debug&arg2=admin.access
```

will lead to `cli` being called as:

```
cli get "admin.password" "admin.debug" "admin.access"
```

It supports whitespaces but please be aware that any special characters in

the URL must be specified according to RFC1738 (usually done by common clients such as `wget`, `lynx`, `curl`).

Response:

The returned response will always contain a status line in the format:

```
<return>: <msg>
```

with return values of `OK` if succeeded and `ERROR` if failed. Any output from the commands will be appended.

Examples:

```
OK: status command successful
ERROR: authentication failed
```

status - Display status information

Key usage:

```
command=status[&arg0=<section>]
```

Notes:

Available sections can be retrieved by running
`command=status&arg0=-h`.

Please note that the status summary can be displayed without authentication.

Examples:

```
http://192.168.1.1/cli.php?version=2&output=html&usr=admin&pwd=admin01&
command=status&arg0=-h
```

```
http://192.168.1.1/cli.php?version=2&output=html&usr=admin&pwd=admin01&
command=status&arg0=summary
```

```
http://192.168.1.1/cli.php?version=2&output=html&command=status
```

get - Get configuration parameter

Key usage:

```
command=get&arg0=<config-key>[&arg1=<config-key>..]
```

Examples:

```
http://192.168.1.1/cli.php?version=2&output=html&usr=admin&pwd=admin01&
command=get&arg0=config.version
```

```
http://192.168.1.1/cli.php?version=2&output=html&usr=admin&pwd=admin01&
command=get&arg0=openvpn.status&arg1=snmp.status&arg2=ipsec.status
```

set - Set configuration parameter

Key usage:

```
command=set&arg0=<config-key>&arg1=<config-value>[&arg2=<config-key>&
arg3=<config-value>..]
```

Notes:

In contrast to the other commands, this command requires a set of tuples because of the reserved '=' char, i.e.
 [arg0=key0, arg1=val0], [arg2=key1, arg3=val1], [arg4=key2, arg5=val2], etc

Examples:

```
http://192.168.1.1/cli.php?version=2&output=html&usr=admin&pwd=admin01&
command=set&arg0=snmp.status&arg1=1
```

```
http://192.168.1.1/cli.php?version=2&output=html&usr=admin&pwd=admin01&
command=set&arg0=snmp.status&arg1=0&arg2=openvpn.status&arg3=1
```

restart - Restart a system service

Key usage:

```
command=restart&arg0=<service>
```

Notes:

Available services can be retrieved by running 'command=restart&arg0=-h'

Examples:

```
http://192.168.1.1/cli.php?version=2&output=html&usr=admin&pwd=admin01&
command=restart&arg0=-h
```

```
http://192.168.1.1/cli.php?version=2&output=html&usr=admin&pwd=admin01&
command=restart&arg0=link-manager
```

reboot - Trigger system reboot

Key usage:

```
command=reboot
```

Examples:

```
http://192.168.1.1/cli.php?version=2&output=html&usr=admin&pwd=admin01&
command=reboot
```

reset - Run factory reset

Key usage:

```
command=reset
```

Examples:

```
http://192.168.1.1/cli.php?version=2&output=html&usr=admin&pwd=admin01&
command=reset
```

update - Update system facilities

Key usage:

```
command=update&arg0=<facility>&arg1=<URL>
```

Notes:

Available facilities can be retrieved by running 'command=update&arg0=-h'

Examples:

```
http://192.168.1.1/cli.php?version=2&output=html&usr=admin&pwd=admin01&
command=update&arg0=software&arg1=tftp://192.168.1.254/latest
```

```
http://192.168.1.1/cli.php?version=2&output=html&usr=admin&pwd=admin01&
command=update&arg0=config&arg1=tftp://192.168.1.254/user-config.zip
```

```
http://192.168.1.1/cli.php?version=2&output=html&usr=admin&pwd=admin01&
command=update&arg0=license&arg1=http://192.168.1.254/xxx.lic
```

```
http://192.168.1.1/cli.php?version=2&output=html&usr=admin&pwd=admin01&
command=update&arg0=firmware&arg1=wwan0&arg2=tftp://192.168.1.254/
firmware
```

send - Send SMS

Key usage:

```
command=send&arg0=sms&arg1=<number>&arg2=<text>
```

Notes:

The phone number has to be specified in international format such as +123456789 including a leading plus sign (which can be encoded with %2B). The SMS daemon must be properly configured prior to using that function.

Examples:

```
http://192.168.1.1/cli.php?version=2&output=html&usr=admin&pwd=admin01&
command=send&arg0=sms&arg1=%2B123456789&arg2=test
```

send - Send E-Mail

Key usage:

```
command=send&arg0=mail&arg1=<address>&arg2=<text>
```

Notes:

The address has to be a valid E-Mail address such as abc@abc.com (the at-sign can be encoded with %40). The E-Mail client must be properly configured prior to using that function.

Examples:

```
http://192.168.1.1/cli.php?version=2&output=html&usr=admin&pwd=admin01&command=send&arg0=mail&arg1=abc%40abc.com&arg2=test
```

send - Send TechSupport

Key usage:

```
command=send&arg0=techsupport&arg1=stdout  
command=send&arg0=techsupport&arg1=<address>&arg2=<subject>
```

Notes:

The address has to be a valid E-Mail address such as abc@abc.com (the at-sign can be encoded with %40). The E-Mail client must be properly configured prior to using that function.

In case of stdout, the downloaded techsupport file will be called 'download'.

Examples:

```
http://192.168.1.1/cli.php?version=2&output=mime&usr=admin&pwd=admin01&command=send&arg0=techsupport&arg1=stdout  
http://192.168.1.1/cli.php?version=2&output=html&usr=admin&pwd=admin01&command=send&arg0=techsupport&arg1=abc%40abc.com&arg2=subject
```

send - Send USSD code

Key usage:

```
command=send&arg0=ussd&arg1=<card>&arg2=<code>
```

Notes:

The argument card specifies the card module index (e.g. 0 for wwan0). The USSD code can consist of digits, plus signs, asterisks (can be encoded with \%2A) and dashes (can be encoded with \%23).

Examples:

```
http://192.168.1.1/cli.php?version=2&output=html&usr=admin&pwd=admin01&command=send&arg0=ussd&arg1=0&arg2=%2A100%23
```

A. Appendix

A.1. Abbreviations

Parameter	Description
ETH _x	Corresponds to Ethernet interfaces (either single or switched ones)
LAN _x	LAN interfaces which are generally based on Ethernet interfaces (including bridges)
WLAN _x	Refers to a Wireless LAN interface which will be represented as additional LAN interface when configured as access point
WWAN _x	Refers to a Wireless Wide Area Network (2G/3G/4G) connection
TUN _x	Specifies an OpenVPN tunnel interface (based on TUN)
TAP _x	Specifies an OpenVPN tunnel interface (based on TAP)
PPTP _x	Specifies a PPTP tunnel interface
MOBILEIP _x	Refers to a Mobile IP tunnel interface
SIM _x	Specifies the SIM slot as seen on the front panel
GNSS _x	Specifies a Global Navigation Satellite System module
Mobile _x	Identifies a WWAN modem
SERIAL _x	Identifies a serial port
OUT _x	Specifies a digital I/O output port (DO _x)
IN _x	Specifies a digital I/O input port (DI _x)
ANY	Generally includes all options offered by the current section
APN	Access Point Name
CID	A Cell ID is a generally unique number used to identify each Base Transceiver Station (BTS).
LAC	The Location Area Code corresponds to an identifier of a set of base stations that are grouped together to optimize signaling
LAI	The Location Area Identity is a globally unique number that identifies the country, network provider and location area
MSS	Maximum Segment Size
MTU	Maximum Transmission Unit
DNS	Domain Name System
NAPT	Network Address and Port Translation

Parameter	Description
DHCP	Dynamic Host Configuration Protocol
SDK	Script Development Kit which can be used to program applications
CLI	Command Line Interface, a generic interface to query the router or perform system tasks
SIM	Subscriber Identity Module
SMS	Short Message Service
SSID	Service Set Identifiers, can be used to define multiple WLAN networks on a module
STP	Spanning Tree Protocol
USSD	Unstructured Supplementary Service Data
VRRP	Virtual Router Redundancy Protocol
VPN	Virtual Private Network
WAN	WAN links include all Wide Area Network interfaces which are currently activated in the system
FQDN	Fully qualified domain name
ASU	Arbitrary Strength Unit
RSRP	Referenz Signal Received Power
RSRQ	Reference Signal Received Quality
LAI	Location Area Identification
LAC	Location Area Code
MCC	Mobile Country Code
MNC	Mobile Network Code
CID	Cell-ID
MSISDN	Mobile Subscriber Integrated Services Digital Network Number
ICCID	Integrated Circuit Card Identifier
MEID	Mobile Equipment Identifier
IMSI	International Mobile Subscriber Identity
IMEI	International Mobile Station Equipment Identity

Table A.1.: Abbreviations

In general, internal interfaces are written lower-case and may have a different naming. Their index starts from zero, whereas interfaces seen by the user will be written in capital letters

starting from one.

A.2. System Events

ID	Event	Description
101	wan-up	WAN link came up
102	wan-down	WAN link went down
201	dio-in1-on	DIO IN1 turned on
202	dio-in1-off	DIO IN1 turned off
203	dio-in2-on	DIO IN2 turned on
204	dio-in2-off	DIO IN2 turned off
205	dio-out1-on	DIO OUT1 turned on
206	dio-out1-off	DIO OUT1 turned off
207	dio-out2-on	DIO OUT2 turned on
208	dio-out2-off	DIO OUT2 turned off
301	gps-up	GPS signal is available
302	gps-down	GPS signal is not available
401	openvpn-up	OpenVPN connection came up
402	openvpn-down	OpenVPN connection went down
403	ipsec-up	IPsec connection came up
404	ipsec-down	IPsec connection went down
406	pptp-up	PPTP connection came up
407	pptp-down	PPTP connection went down
408	dialin-up	Dial-In connection came up
409	dialin-down	Dial-In connection went down
410	mobileip-up	Mobile IP connection came up
411	mobileip-down	Mobile IP connection went down
412	gre-up	GRE connection came up
413	gre-down	GRE connection went down
501	system-login-failed	User login failed
502	system-login-succeeded	User login succeeded
503	system-logout	User logged out

ID	Event	Description
504	system-rebooting	System reboot has been triggered
505	system-startup	System has been started
506	test	test event
507	sdk-startup	SDK has been started
508	system-time-updated	System time has been updated
509	system-poweroff	System poweroff has been triggered
510	system-error	System is in error state
511	system-no-error	System left error state
601	sms-sent	SMS has been sent
602	sms-notsent	SMS has not been sent
603	sms-received	SMS has been received
604	sms-report-received	SMS report has been received
701	call-incoming	A voice call is coming in
702	call-outgoing	Outgoing voice call is being established
801	ddns-update-succeeded	Dynamic DNS update succeeded
802	ddns-update-failed	Dynamic DNS update failed
901	usb-storage-added	USB storage device has been added
902	usb-storage-removed	USB storage device has been removed
903	usb-eth-added	USB Ethernet device has been added
904	usb-eth-removed	USB Ethernet device has been removed
905	usb-serial-added	USB serial device has been added
906	usb-serial-removed	USB serial device has been removed
1001	redundancy-master	System is now master router
1002	redundancy-backup	System is now backup router

Table A.2.: System Events

A.3. Factory Configuration

The factory configuration including default values for any configuration parameter can be derived from the file `/etc/config/factory-config.cfg` on the router. You may also call `cli get -f <parameter>` for obtaining a specific default value.

A.4. SNMP VENDOR MIB

```

-- *****
-- NetModule AG VENDOR MIB
--
--
-- (c) COPYRIGHT 2020 by NetModule AG, Switzerland
-- All rights reserved.
--
-- *****

NB-MIB DEFINITIONS ::= BEGIN

-- *****
-- imports
-- *****

IMPORTS
    MODULE-IDENTITY, OBJECT-TYPE, NOTIFICATION-TYPE,
    Integer32, Counter32, Gauge32,
    Counter64, TimeTicks                         FROM SNMPv2-SMI
    TEXTUAL-CONVENTION, DisplayString,
    PhysAddress, TruthValue, RowStatus, DateAndTime,
    TimeStamp, AutonomousType, TestAndIncr     FROM SNMPv2-TC
    MODULE-COMPLIANCE, OBJECT-GROUP           FROM SNMPv2-CONF
    snmpTraps                                 FROM SNMPv2-MIB
    URLString                                 FROM NETWORK-SERVICES-MIB
    enterprises                               FROM SNMPv2-SMI;

-- *****
-- module definition
-- *****

nb MODULE-IDENTITY
    LAST-UPDATED "201806261330Z"
    ORGANIZATION "NetModule AG"
    CONTACT-INFO
        "NetModule AG, Switzerland"
    DESCRIPTION
        "MIB module which defines the NB router specific entities"

    REVISION "201806261330Z"
    DESCRIPTION
        "MIB for software release 4.1"

    REVISION "201610181200Z"
    DESCRIPTION
        "MIB for software release 4.0"

    REVISION "201607121200Z"
    DESCRIPTION
        "MIB for software release 4.0"

    REVISION "201603021200Z"
    DESCRIPTION
        "MIB for software release 3.9"

    REVISION "201411241000Z"
    DESCRIPTION
        "MIB for software release 3.8"

    REVISION "201405091000Z"
    DESCRIPTION
        "MIB for software release 3.7"

    REVISION "201212191000Z"
    DESCRIPTION
        "MIB for software release 3.6"
    ::= { netmodule 10 }

-- *****
-- root anchor
-- *****

netmodule OBJECT IDENTIFIER ::= { enterprises 31496 }

-- *****
-- table definitions
-- *****

system OBJECT IDENTIFIER ::= { nb 1 }
products OBJECT IDENTIFIER ::= { nb 10 }
admin OBJECT IDENTIFIER ::= { nb 40 }

```

```

dio          OBJECT IDENTIFIER ::= { nb 53 }
sdk          OBJECT IDENTIFIER ::= { nb 90 }
traps       OBJECT IDENTIFIER ::= { nb 100 }

-- *****

nb1600      OBJECT IDENTIFIER ::= { products 46 }
nb2700      OBJECT IDENTIFIER ::= { products 47 }
nb3700      OBJECT IDENTIFIER ::= { products 48 }
nb2710      OBJECT IDENTIFIER ::= { products 51 }
nb3710      OBJECT IDENTIFIER ::= { products 52 }
nb3720      OBJECT IDENTIFIER ::= { products 53 }
nb2800      OBJECT IDENTIFIER ::= { products 54 }
nb3701      OBJECT IDENTIFIER ::= { products 55 }
nb3711      OBJECT IDENTIFIER ::= { products 56 }
nb3800      OBJECT IDENTIFIER ::= { products 57 }
nb800       OBJECT IDENTIFIER ::= { products 58 }

-- *****
-- NAdminTable
-- *****

swVersion OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The currently installed system software version"
    ::= { admin 1 }

kernelVersion OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The currently installed kernel version"
    ::= { admin 2 }

serialNumber OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The serial number of the device"
    ::= { admin 3 }

configDesc OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The description of the current configuration"
    ::= { admin 4 }

configHash OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The hash of the current configuration"
    ::= { admin 5 }

softwareHash OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The hash of the current software"
    ::= { admin 6 }

systemStatus OBJECT-TYPE
    SYNTAX      INTEGER {
        ok (1),
        degraded (2),
        outoforder (3)
    }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The global system status"
    ::= { admin 7 }

systemError OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "A comma-separated list of services which are in error state"
    ::= { admin 8 }

```

```
systemDate OBJECT-TYPE
SYNTAX      DateAndTime
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The current local date and time of day."
 ::= { admin 9 }

deviceRestart OBJECT-TYPE
SYNTAX      INTEGER {
                restart (1)
            }
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
    "Force a device restart"
 ::= { admin 10 }

-- Update --

updateOperation OBJECT-TYPE
SYNTAX      INTEGER {
                update (0),
                store (1)
            }
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
    "The desired operation for configuration or software updates"
 ::= { admin 11 }

switchOperation OBJECT-TYPE
SYNTAX      INTEGER {
                software (0),
                config (1)
            }
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
    "The operation trigger to switch to alternative software or configuration"
 ::= { admin 12 }

softwareActivationDate OBJECT-TYPE
SYNTAX      DateAndTime
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
    "The date and time when the alternative software shall be activated"
 ::= { admin 13 }

configActivationDate OBJECT-TYPE
SYNTAX      DateAndTime
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
    "The date and time when the alternative configuration shall be activated"
 ::= { admin 14 }

softwareActivatedDate OBJECT-TYPE
SYNTAX      DateAndTime
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
    "The Date and Time when the current running software was booted the first time"
 ::= { admin 15 }

-- Configuration Update --

configUpdate OBJECT-TYPE
SYNTAX      URLString
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
    "Update the system configuration from the specified URL,
     the URL must be preceded by a valid prefix (e.g. tftp://, sftp://, ftp://, https:// or http://)
     and either point to the update package or to a server directory which
     contains a file named <serial-number>.zip"
 ::= { admin 20 }

configUpdateStatus OBJECT-TYPE
SYNTAX      INTEGER {
                stored (0),
                succeeded (1),
                failed (2),
                inprogress (3),
                notstarted (4)
            }
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
```

```

        "The status of the last configuration update cycle"
    ::= { admin 21 }

configUpdateError OBJECT-TYPE
    SYNTAX      Integer32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The error code of the last configuration update"
    ::= { admin 22 }

configUpdated OBJECT-TYPE
    SYNTAX      DateAndTime
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The date of the last configuration update"
    ::= { admin 23 }

configUpdateMode OBJECT-TYPE
    SYNTAX      INTEGER {
        full (0),
        partial (1)
    }
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "The desired system configuration update mode (full or partial)"
    ::= { admin 24 }

-- Software Update --

softwareUpdate OBJECT-TYPE
    SYNTAX      URLString
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "Update the system software from the specified URL,
         the URL must be preceded by a valid prefix (e.g. tftp://, sftp://, ftp://, https:// or http://)
         and point to the to be installed image"
    ::= { admin 25 }

softwareUpdateStatus OBJECT-TYPE
    SYNTAX      INTEGER {
        stored (0),
        succeeded (1),
        failed (2),
        inprogress (3),
        notstarted (4)
    }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The status of the last software update cycle"
    ::= { admin 26 }

softwareUpdateError OBJECT-TYPE
    SYNTAX      Integer32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The error code of the last software update"
    ::= { admin 27 }

softwareUpdated OBJECT-TYPE
    SYNTAX      DateAndTime
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The date of the last software update"
    ::= { admin 28 }

-- Alternative Configuration --

altConfigDesc OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The description of the alternative configuration"
    ::= { admin 30 }

altConfigHash OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The hash of the alternative configuration"
    ::= { admin 31 }

```

```
altConfigUpdated OBJECT-TYPE
  SYNTAX      DateAndTime
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
    "The date of the last alternative configuration update"
  ::= { admin 32 }

-- Alternative Software --

altSoftwareVersion OBJECT-TYPE
  SYNTAX      DisplayString
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
    "The version of the alternative software"
  ::= { admin 35 }

altSoftwareHash OBJECT-TYPE
  SYNTAX      DisplayString
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
    "The hash of the alternative software"
  ::= { admin 36 }

altSoftwareUpdated OBJECT-TYPE
  SYNTAX      DateAndTime
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
    "The date of the last alternative software update"
  ::= { admin 37 }

-- Upload Syslog --

syslogUpload OBJECT-TYPE
  SYNTAX      URLString
  MAX-ACCESS  read-write
  STATUS      current
  DESCRIPTION
    "Upload the current system logs to the specified URL,
     the URL must be preceded by a valid prefix (e.g. tftp://, sftp://, ftp://, https:// or http://)
     and point to the path where the system log shall be stored."
  ::= { admin 40 }

syslogUploadStatus OBJECT-TYPE
  SYNTAX      INTEGER {
    succeeded (1),
    failed (2),
    inprogress (3),
    notstarted (4)
  }
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
    "The status of the last syslog upload cycle"
  ::= { admin 41 }

-- Upload Config --

configUpload OBJECT-TYPE
  SYNTAX      URLString
  MAX-ACCESS  read-write
  STATUS      current
  DESCRIPTION
    "Upload the current configuration to the specified URL,
     the URL must be preceded by a valid prefix (e.g. tftp://, sftp://, ftp://, https:// or http://)
     and point to the path where the config shall be stored."
  ::= { admin 42 }

configUploadStatus OBJECT-TYPE
  SYNTAX      INTEGER {
    succeeded (1),
    failed (2),
    inprogress (3),
    notstarted (4)
  }
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
    "The status of the last config upload cycle"
  ::= { admin 43 }

-- *****
-- NBWwanTable
-- *****

nbWwanTable OBJECT-TYPE
  SYNTAX      SEQUENCE OF NBWwanEntry
```

```

MAX-ACCESS not-accessible
STATUS current
DESCRIPTION "The table describing any WWAN modems and their current settings"
 ::= { nb 50 }

nbWwanEntry OBJECT-TYPE
SYNTAX NBWwanEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION "An entry describing a WWAN modem and its current settings"
INDEX { wwanModemIndex }
 ::= { nbWwanTable 1 }

NBWwanEntry ::= SEQUENCE {
  wwanModemIndex Integer32,
  wwanModemName DisplayString,
  wwanModemType DisplayString,
  wwanServiceType DisplayString,
  wwanRegistrationState DisplayString,
  wwanSignalStrength Integer32,
  wwanNetworkName DisplayString,
  wwanLocalAreaIdentification DisplayString,
  wwanLocalAreaCode DisplayString,
  wwanCellId DisplayString,
  wwanTemperature DisplayString,
  wwanLccid DisplayString,
  wwanRSRP DisplayString,
  wwanRSRQ DisplayString,
  wwanSINR DisplayString,
  wwanRSCP DisplayString,
  wwanECIO DisplayString,
  wwanSignalLevel Integer32,
  wwanSignalQuality DisplayString
}

wwanModemIndex OBJECT-TYPE
SYNTAX Integer32(0..254)
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"WWAN modem index"
 ::= { nbWwanEntry 1 }

wwanModemName OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"WWAN modem name"
 ::= { nbWwanEntry 2 }

wwanModemType OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"WWAN modem type"
 ::= { nbWwanEntry 3 }

wwanServiceType OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The current service type of the WWAN modem"
 ::= { nbWwanEntry 4 }

wwanRegistrationState OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The current registration state of the WWAN modem"
 ::= { nbWwanEntry 5 }

wwanSignalStrength OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The current signal strength of the WWAN modem (-999 means unknown)"
 ::= { nbWwanEntry 6 }

wwanNetworkName OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The network name to which the WWAN modem is currently registered"
 ::= { nbWwanEntry 7 }

```

```

wanLocalAreaIdentification OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The Local Area Identification (LAI) to which the WWAN modem is currently registered"
    ::= { nbWwanEntry 8 }

wanLocalAreaCode OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The Local Area Code (LAC) to which the WWAN modem is currently registered"
    ::= { nbWwanEntry 9 }

wanCellId OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The Cell ID (CID) to which the WWAN modem is currently registered"
    ::= { nbWwanEntry 10 }

wanTemperature OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The current temperature of the WWAN modem"
    ::= { nbWwanEntry 11 }

wanIccid OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The Integrated Circuit Card Identifier (ICCID) of the SIM connected to the WWAN modem"
    ::= { nbWwanEntry 12 }

wanRSRP OBJECT-TYPE
    SYNTAX      DisplayString
    UNITS       "dBm"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The current Reference Signal Received Power (LTE) of the WWAN modem"
    ::= { nbWwanEntry 13 }

wanRSRQ OBJECT-TYPE
    SYNTAX      DisplayString
    UNITS       "dB"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The current Reference Signal Received Quality (LTE) of the WWAN modem"
    ::= { nbWwanEntry 14 }

wanSINR OBJECT-TYPE
    SYNTAX      DisplayString
    UNITS       "dB"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The current Signal to interference plus noise ratio (LTE) of the WWAN modem"
    ::= { nbWwanEntry 15 }

wanRSCP OBJECT-TYPE
    SYNTAX      DisplayString
    UNITS       "dBm"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The current Received Signal Code Power (UMTS) of the WWAN modem"
    ::= { nbWwanEntry 16 }

wanECIO OBJECT-TYPE
    SYNTAX      DisplayString
    UNITS       "dB"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The current ratio of Received power of the carrier to the all over Noise (UMTS) of the WWAN modem←"
    ::= { nbWwanEntry 17 }

wanSignalLevel OBJECT-TYPE
    SYNTAX      Integer32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The current signal level of the WWAN modem"
    ::= { nbWwanEntry 18 }

wanSignalQuality OBJECT-TYPE
    
```

```

SYNTAX      DisplayString
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION "The current signal quality of the WWAN modem"
 ::= { nbWwanEntry 19 }

-- *****
-- NBGnssTable
-- *****

nbGnssTable OBJECT-TYPE
SYNTAX      SEQUENCE OF NBGnssEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
 "The table describing any GNSS devices and their current settings"
 ::= { nb 51 }

nbGnssEntry OBJECT-TYPE
SYNTAX      NBGnssEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
 "An entry describing a GNSS device and its current settings"
INDEX       { gnssIndex }
 ::= { nbGnssTable 1 }

NBGnssEntry ::= SEQUENCE {
    gnssIndex Integer32,
    gnssName  DisplayString,
    gnssSystem DisplayString,
    gnssLat   DisplayString,
    gnssLon   DisplayString,
    gnssAlt   DisplayString,
    gnssNumSat Integer32,
    gnssNumSatUsed Integer32,
    gnssHorizontalSpeed DisplayString,
    gnssVerticalSpeed DisplayString,
    gnssTrackAngle DisplayString
}

gnssIndex OBJECT-TYPE
SYNTAX      Integer32(0..254)
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
 "GNSS device index"
 ::= { nbGnssEntry 1 }

gnssName OBJECT-TYPE
SYNTAX      DisplayString
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
 "GNSS device name"
 ::= { nbGnssEntry 2 }

gnssSystem OBJECT-TYPE
SYNTAX      DisplayString
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
 "GNSS system used by the device"
 ::= { nbGnssEntry 3 }

gnssLat OBJECT-TYPE
SYNTAX      DisplayString
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
 "The current latitude value received by the GNSS device"
 ::= { nbGnssEntry 4 }

gnssLon OBJECT-TYPE
SYNTAX      DisplayString
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
 "The current longitude value received by the GNSS device"
 ::= { nbGnssEntry 5 }

gnssAlt OBJECT-TYPE
SYNTAX      DisplayString
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
 "The current altitude value received by the GNSS device"
 ::= { nbGnssEntry 6 }

gnssNumSat OBJECT-TYPE

```

```

SYNTAX      Integer32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The current number of satellites in view for the GNSS device"
 ::= { nbGnssEntry 7 }

gnssNumSatUsed OBJECT-TYPE
SYNTAX      Integer32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The current number of used satellites for the GNSS device"
 ::= { nbGnssEntry 8 }

gnssHorizontalSpeed OBJECT-TYPE
SYNTAX      DisplayString
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The current horizontal speed over the ground value in meter per second received by the GNSS device"
 ::= { nbGnssEntry 9 }

gnssVerticalSpeed OBJECT-TYPE
SYNTAX      DisplayString
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The current vertical speed value in meter per second received by the GNSS device"
 ::= { nbGnssEntry 10 }

gnssTrackAngle OBJECT-TYPE
SYNTAX      DisplayString
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The current track angle value in degrees received by the GNSS device"
 ::= { nbGnssEntry 11 }

-- *****
-- NBWlanTable
-- *****

nbWlanTable OBJECT-TYPE
SYNTAX      SEQUENCE OF NBWlanEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
    "A table describing any WLAN modems and their current settings."
 ::= { nb 60 }

nbWlanEntry OBJECT-TYPE
SYNTAX      NBWlanEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
    "An entry describing a WLAN modem and its current settings."
INDEX      { wlanModuleIndex }
 ::= { nbWlanTable 1 }

NBWlanEntry ::= SEQUENCE {
    wlanModuleIndex Integer32,
    wlanModuleName DisplayString,
    wlanModuleType DisplayString,
    wlanNumClients Integer32,
    wlanModuleChannel Integer32,
    wlanModuleFrequency Integer32,
    wlanSignalStrength Integer32
}

wlanModuleIndex OBJECT-TYPE
SYNTAX      Integer32(0..254)
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
    "WLAN module index"
 ::= { nbWlanEntry 1 }

wlanModuleName OBJECT-TYPE
SYNTAX      DisplayString
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "WLAN module name"
 ::= { nbWlanEntry 2 }

wlanModuleType OBJECT-TYPE
SYNTAX      DisplayString
MAX-ACCESS  read-only

```

```

STATUS        current
DESCRIPTION
    "WLAN module type"
 ::= { nbWlanEntry 3 }

wlanNumClients OBJECT-TYPE
SYNTAX        Integer32
MAX-ACCESS    read-only
STATUS        current
DESCRIPTION
    "Current number of clients connected to the WLAN module in access-point mode"
 ::= { nbWlanEntry 4 }

wlanModuleChannel OBJECT-TYPE
SYNTAX        Integer32
MAX-ACCESS    read-only
STATUS        current
DESCRIPTION
    "Current channel of the WLAN module"
 ::= { nbWlanEntry 5 }

wlanModuleFrequency OBJECT-TYPE
SYNTAX        Integer32
UNITS         "MHz"
MAX-ACCESS    read-only
STATUS        current
DESCRIPTION
    "Current frequency of the WLAN module"
 ::= { nbWlanEntry 6 }

wlanSignalStrength OBJECT-TYPE
SYNTAX        Integer32
UNITS         "dBm"
MAX-ACCESS    read-only
STATUS        current
DESCRIPTION
    "Current signal strength of the WLAN module in client mode"
 ::= { nbWlanEntry 7 }

-- *****
-- NBWlanStationTable
-- *****

nbWlanStationTable OBJECT-TYPE
SYNTAX        SEQUENCE OF NBWlanStationEntry
MAX-ACCESS    not-accessible
STATUS        current
DESCRIPTION
    "A table shows current connected clients "
 ::= { nb 61 }

nbWlanStationEntry OBJECT-TYPE
SYNTAX        NBWlanStationEntry
MAX-ACCESS    not-accessible
STATUS        current
DESCRIPTION
    "An entry describes one connected client"
INDEX         { wlanStationIndex }
 ::= { nbWlanStationTable 1 }

NBWlanStationEntry ::= SEQUENCE {
    wlanStationIndex Integer32,
    wlanStationInterface DisplayString,
    wlanStationMac DisplayString,
    wlanStationSignalStrength Integer32,
    wlanStationBitrate Integer32,
    wlanStationRxBytes Counter64,
    wlanStationTxBytes Counter64,
    wlanStationInactive Integer32
}

wlanStationIndex OBJECT-TYPE
SYNTAX        Integer32(0..254)
MAX-ACCESS    not-accessible
STATUS        current
DESCRIPTION
    "WLAN station index"
 ::= { nbWlanStationEntry 1 }

wlanStationInterface OBJECT-TYPE
SYNTAX        DisplayString
MAX-ACCESS    read-only
STATUS        current
DESCRIPTION
    "The WLAN interface name"
 ::= { nbWlanStationEntry 2 }

wlanStationMac OBJECT-TYPE
SYNTAX        DisplayString
MAX-ACCESS    read-only

```

```

STATUS      current
DESCRIPTION
    "The MAC address of a connected station"
 ::= { nbWlanStationEntry 3 }

wlanStationSignalStrength OBJECT-TYPE
SYNTAX      Integer32
UNITS       "dBm"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The signal strength of a connected station"
 ::= { nbWlanStationEntry 4 }

wlanStationBitrate OBJECT-TYPE
SYNTAX      Integer32
UNITS       "Mbit/s"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The bitrate of a connected station"
 ::= { nbWlanStationEntry 5 }

wlanStationRxBytes OBJECT-TYPE
SYNTAX      Counter64
UNITS       "bytes"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The number of received bytes of a connected station"
 ::= { nbWlanStationEntry 6 }

wlanStationTxBytes OBJECT-TYPE
SYNTAX      Counter64
UNITS       "bytes"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The number of transmitted bytes of a connected station"
 ::= { nbWlanStationEntry 7 }

wlanStationInactive OBJECT-TYPE
SYNTAX      Integer32
UNITS       "ms"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The inactivity time of a connected station"
 ::= { nbWlanStationEntry 8 }

-- *****
-- NBWanTable
-- *****

nbHotLink OBJECT-TYPE
SYNTAX      DisplayString
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The active WAN link"
 ::= { nb 70 }

nbWanTable OBJECT-TYPE
SYNTAX      SEQUENCE OF NBWanEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION "The table describing any WAN link and their current status"
 ::= { nb 71 }

nbWanEntry OBJECT-TYPE
SYNTAX      NBWanEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION "An entry describing a WAN link and its current status"
INDEX       { wanLinkIndex }
 ::= { nbWanTable 1 }

NBWanEntry ::= SEQUENCE {
    wanLinkIndex Integer32,
    wanLinkName  DisplayString,
    wanLinkState DisplayString,
    wanLinkSince DisplayString,
    wanLinkType  DisplayString,
    wanLinkInterface DisplayString,
    wanLinkAddress DisplayString,
    wanLinkGateway DisplayString,
    wanLinkNetmask DisplayString,
    wanDialAttempts Integer32,
    wanDialSuccess Integer32,

```

```

wanDialFailures Integer32,
wanDataDownloaded Counter64,
wanDataUploaded Counter64,
wanDownloadRate Integer32,
wanUploadRate Integer32,
wanDataDownloadedRoaming Counter64,
wanDataUploadedRoaming Counter64
}

wanLinkIndex OBJECT-TYPE
    SYNTAX      Integer32(0..254)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "WAN link index"
    ::= { nbWanEntry 1 }

wanLinkName OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "WAN link name"
    ::= { nbWanEntry 2 }

wanLinkState OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "WAN link state"
    ::= { nbWanEntry 3 }

wanLinkSince OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "WAN link since up"
    ::= { nbWanEntry 4 }

wanLinkType OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "WAN link type"
    ::= { nbWanEntry 5 }

wanLinkInterface OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "WAN link interface"
    ::= { nbWanEntry 6 }

wanLinkAddress OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "WAN link address"
    ::= { nbWanEntry 7 }

wanLinkGateway OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "WAN link gateway"
    ::= { nbWanEntry 8 }

wanLinkNetmask OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "WAN link netmask"
    ::= { nbWanEntry 9 }

wanDialAttempts OBJECT-TYPE
    SYNTAX      Integer32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "WAN link dial attempts"
    ::= { nbWanEntry 10 }

wanDialSuccess OBJECT-TYPE

```

```

SYNTAX      Integer32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "WAN link dial success"
 ::= { nbWanEntry 11 }

wanDialFailures OBJECT-TYPE
SYNTAX      Integer32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "WAN link dial failures"
 ::= { nbWanEntry 12 }

wanDataDownloaded OBJECT-TYPE
SYNTAX      Counter64
UNITS       "bytes"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "WAN link data downloaded"
 ::= { nbWanEntry 13 }

wanDataUploaded OBJECT-TYPE
SYNTAX      Counter64
UNITS       "bytes"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "WAN link data uploaded"
 ::= { nbWanEntry 14 }

wanDownloadRate OBJECT-TYPE
SYNTAX      Integer32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "WAN link download rate"
 ::= { nbWanEntry 15 }

wanUploadRate OBJECT-TYPE
SYNTAX      Integer32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "WAN link upload rate"
 ::= { nbWanEntry 16 }

wanDataDownloadedRoaming OBJECT-TYPE
SYNTAX      Counter64
UNITS       "bytes"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "WAN link data downloaded during roaming"
 ::= { nbWanEntry 17 }

wanDataUploadedRoaming OBJECT-TYPE
SYNTAX      Counter64
UNITS       "bytes"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "WAN link data uploaded during roaming"
 ::= { nbWanEntry 18 }

-- *****
-- NBDioTable
-- *****

dioStatusIn1 OBJECT-TYPE
SYNTAX      INTEGER {
                off (0),
                on (1)
            }
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The current value of digital I/O port IN1"
 ::= { dio 1 }

dioStatusIn2 OBJECT-TYPE
SYNTAX      INTEGER {
                off (0),

```

```

        }
        on (1)
    }
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The current value of digital I/O port IN2"
    ::= { dio 2 }

dioStatusOut1 OBJECT-TYPE
    SYNTAX INTEGER {
        off (0),
        on (1)
    }
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The current value of digital I/O port OUT1"
    ::= { dio 3 }

dioStatusOut2 OBJECT-TYPE
    SYNTAX INTEGER {
        off (0),
        on (1)
    }
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The current value of digital I/O port OUT2"
    ::= { dio 4 }

dioSetOUT1 OBJECT-TYPE
    SYNTAX INTEGER {
        off (0),
        on (1)
    }
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "The update value for digital I/O port OUT1"
    ::= { dio 10 }

dioSetOUT2 OBJECT-TYPE
    SYNTAX INTEGER {
        off (0),
        on (1)
    }
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "The update value for digital I/O port OUT2"
    ::= { dio 11 }

-- *****
-- NBSerialTable
-- *****

nbSerialTable OBJECT-TYPE
    SYNTAX SEQUENCE OF NBSerialEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "The table describing any serial ports and their current statistics"
    ::= { nb 54 }

nbSerialEntry OBJECT-TYPE
    SYNTAX NBSerialEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "An entry describing a serial port and its current statistics"
    INDEX { serialIndex }
    ::= { nbSerialTable 1 }

NBSerialEntry ::= SEQUENCE {
    serialIndex Integer32,
    serialName DisplayString,
    serialState Integer32,
    serialRxBytes Integer32,
    serialTxBytes Integer32,
    serialFrameErrors Integer32,
    serialOverrunErrors Integer32,
    serialParityErrors Integer32,
    serialBrkErrors Integer32,
    serialBufferOverrunErrors Integer32
}

serialIndex OBJECT-TYPE
    SYNTAX Integer32(0..254)
    MAX-ACCESS not-accessible

```

```

STATUS      current
DESCRIPTION
  "Serial port index"
 ::= { nbSerialEntry 1 }

serialName OBJECT-TYPE
SYNTAX      DisplayString
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
  "Serial port name"
 ::= { nbSerialEntry 2 }

serialState OBJECT-TYPE
SYNTAX      Integer32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
  "The current state of the serial port"
 ::= { nbSerialEntry 3 }

serialRxBytes OBJECT-TYPE
SYNTAX      Integer32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
  "The number of bytes received on the serial port"
 ::= { nbSerialEntry 4 }

serialTxBytes OBJECT-TYPE
SYNTAX      Integer32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
  "The number of bytes transmitted on the serial port"
 ::= { nbSerialEntry 5 }

serialFrameErrors OBJECT-TYPE
SYNTAX      Integer32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
  "The number of frame errors on the serial port"
 ::= { nbSerialEntry 6 }

serialOverrunErrors OBJECT-TYPE
SYNTAX      Integer32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
  "The number of overrun errors on the serial port"
 ::= { nbSerialEntry 7 }

serialParityErrors OBJECT-TYPE
SYNTAX      Integer32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
  "The number of parity errors on the serial port"
 ::= { nbSerialEntry 8 }

serialBrkErrors OBJECT-TYPE
SYNTAX      Integer32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
  "The number of BRK errors on the serial port"
 ::= { nbSerialEntry 9 }

serialBufferOverrunErrors OBJECT-TYPE
SYNTAX      Integer32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
  "The number of buffer overrun errors on the serial port"
 ::= { nbSerialEntry 10 }

-- *****
-- NBTrapHistoryTable
-- *****

nbTrapHistoryTable OBJECT-TYPE
SYNTAX      SEQUENCE OF NBTrapHistoryEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
  "This table shows the last SNMP Traps"
 ::= { nb 80 }

```

```

nbTrapHistoryEntry OBJECT-TYPE
    SYNTAX      NBTrapHistoryEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An entry describing an occurred SNMP trap"
    INDEX       { trapHistoryIndex }
    ::= { nbTrapHistoryTable 1 }

NBTrapHistoryEntry ::= SEQUENCE {
    trapHistoryIndex Integer32,
    trapHistoryTimestamp Counter64,
    trapHistoryUptime Counter64,
    trapHistoryEvent Integer32
}

trapHistoryIndex OBJECT-TYPE
    SYNTAX      Integer32(0..254)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "trap history index"
    ::= { nbTrapHistoryEntry 1 }

trapHistoryTimestamp OBJECT-TYPE
    SYNTAX      Counter64
    UNITS       "seconds"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The timestamp when the SNMP occurred"
    ::= { nbTrapHistoryEntry 2 }

trapHistoryUptime OBJECT-TYPE
    SYNTAX      Counter64
    UNITS       "seconds"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The uptime of the router when the SNMP trap occurred"
    ::= { nbTrapHistoryEntry 3 }

trapHistoryEvent OBJECT-TYPE
    SYNTAX      Integer32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The event type of the SNMP trap"
    ::= { nbTrapHistoryEntry 4 }

-- *****
-- trap objects
-- *****

events          OBJECT IDENTIFIER ::= { traps 0 }

sdk-trap NOTIFICATION-TYPE
    STATUS      current
    DESCRIPTION "SDK trap"
    ::= { events 1 }

wan-up NOTIFICATION-TYPE
    STATUS      current
    DESCRIPTION "WAN link came up"
    ::= { events 101 }

wan-down NOTIFICATION-TYPE
    STATUS      current
    DESCRIPTION "WAN link went down"
    ::= { events 102 }

dio-in1-on NOTIFICATION-TYPE
    STATUS      current
    DESCRIPTION "DIO IN1 turned on"
    ::= { events 201 }

dio-in1-off NOTIFICATION-TYPE
    STATUS      current
    DESCRIPTION "DIO IN1 turned off"
    ::= { events 202 }

dio-in2-on NOTIFICATION-TYPE
    STATUS      current
    DESCRIPTION "DIO IN2 turned on"
    ::= { events 203 }

dio-in2-off NOTIFICATION-TYPE
    STATUS      current
    DESCRIPTION "DIO IN2 turned off"
    ::= { events 204 }

```

```
dio-out1-on NOTIFICATION-TYPE
STATUS current
DESCRIPTION "DIO OUT1 turned on"
::= { events 205 }

dio-out1-off NOTIFICATION-TYPE
STATUS current
DESCRIPTION "DIO OUT1 turned off"
::= { events 206 }

dio-out2-on NOTIFICATION-TYPE
STATUS current
DESCRIPTION "DIO OUT2 turned on"
::= { events 207 }

dio-out2-off NOTIFICATION-TYPE
STATUS current
DESCRIPTION "DIO OUT2 turned off"
::= { events 208 }

gps-up NOTIFICATION-TYPE
STATUS current
DESCRIPTION "GPS signal is available"
::= { events 301 }

gps-down NOTIFICATION-TYPE
STATUS current
DESCRIPTION "GPS signal is not available"
::= { events 302 }

openvpn-up NOTIFICATION-TYPE
STATUS current
DESCRIPTION "OpenVPN connection came up"
::= { events 401 }

openvpn-down NOTIFICATION-TYPE
STATUS current
DESCRIPTION "OpenVPN connection went down"
::= { events 402 }

ipsec-up NOTIFICATION-TYPE
STATUS current
DESCRIPTION "IPsec connection came up"
::= { events 403 }

ipsec-down NOTIFICATION-TYPE
STATUS current
DESCRIPTION "IPsec connection went down"
::= { events 404 }

pptp-up NOTIFICATION-TYPE
STATUS current
DESCRIPTION "PPTP connection came up"
::= { events 406 }

pptp-down NOTIFICATION-TYPE
STATUS current
DESCRIPTION "PPTP connection went down"
::= { events 407 }

dialin-up NOTIFICATION-TYPE
STATUS current
DESCRIPTION "Dial-In connection came up"
::= { events 408 }

dialin-down NOTIFICATION-TYPE
STATUS current
DESCRIPTION "Dial-In connection went down"
::= { events 409 }

mobileip-up NOTIFICATION-TYPE
STATUS current
DESCRIPTION "Mobile IP connection came up"
::= { events 410 }

mobileip-down NOTIFICATION-TYPE
STATUS current
DESCRIPTION "Mobile IP connection went down"
::= { events 411 }

gre-up NOTIFICATION-TYPE
STATUS current
DESCRIPTION "GRE connection came up"
::= { events 412 }

gre-down NOTIFICATION-TYPE
STATUS current
DESCRIPTION "GRE connection went down"
::= { events 413 }
```

```

system-login-failed NOTIFICATION-TYPE
STATUS current
DESCRIPTION "User login failed"
::= { events 501 }

system-login-succeeded NOTIFICATION-TYPE
STATUS current
DESCRIPTION "User login succeeded"
::= { events 502 }

system-logout NOTIFICATION-TYPE
STATUS current
DESCRIPTION "User logged out"
::= { events 503 }

system-rebooting NOTIFICATION-TYPE
STATUS current
DESCRIPTION "System reboot has been triggered"
::= { events 504 }

system-startup NOTIFICATION-TYPE
STATUS current
DESCRIPTION "System has been started"
::= { events 505 }

test NOTIFICATION-TYPE
STATUS current
DESCRIPTION "test event"
::= { events 506 }

sdk-startup NOTIFICATION-TYPE
STATUS current
DESCRIPTION "SDK has been started"
::= { events 507 }

system-time-updated NOTIFICATION-TYPE
STATUS current
DESCRIPTION "System time has been updated"
::= { events 508 }

system-poweroff NOTIFICATION-TYPE
STATUS current
DESCRIPTION "System poweroff has been triggered"
::= { events 509 }

system-error NOTIFICATION-TYPE
STATUS current
DESCRIPTION "System is in error state"
::= { events 510 }

system-no-error NOTIFICATION-TYPE
STATUS current
DESCRIPTION "System left error state"
::= { events 511 }

sms-sent NOTIFICATION-TYPE
STATUS current
DESCRIPTION "SMS has been sent"
::= { events 601 }

sms-notsent NOTIFICATION-TYPE
STATUS current
DESCRIPTION "SMS has not been sent"
::= { events 602 }

sms-received NOTIFICATION-TYPE
STATUS current
DESCRIPTION "SMS has been received"
::= { events 603 }

sms-report-received NOTIFICATION-TYPE
STATUS current
DESCRIPTION "SMS report has been received"
::= { events 604 }

call-incoming NOTIFICATION-TYPE
STATUS current
DESCRIPTION "A voice call is coming in"
::= { events 701 }

call-outgoing NOTIFICATION-TYPE
STATUS current
DESCRIPTION "Outgoing voice call is being established"
::= { events 702 }

ddns-update-succeeded NOTIFICATION-TYPE
STATUS current
DESCRIPTION "Dynamic DNS update succeeded"
::= { events 801 }

```

```
ddns-update-failed NOTIFICATION-TYPE
STATUS current
DESCRIPTION "Dynamic DNS update failed"
::= { events 802 }

usb-storage-added NOTIFICATION-TYPE
STATUS current
DESCRIPTION "USB storage device has been added"
::= { events 901 }

usb-storage-removed NOTIFICATION-TYPE
STATUS current
DESCRIPTION "USB storage device has been removed"
::= { events 902 }

usb-eth-added NOTIFICATION-TYPE
STATUS current
DESCRIPTION "USB Ethernet device has been added"
::= { events 903 }

usb-eth-removed NOTIFICATION-TYPE
STATUS current
DESCRIPTION "USB Ethernet device has been removed"
::= { events 904 }

usb-serial-added NOTIFICATION-TYPE
STATUS current
DESCRIPTION "USB serial device has been added"
::= { events 905 }

usb-serial-removed NOTIFICATION-TYPE
STATUS current
DESCRIPTION "USB serial device has been removed"
::= { events 906 }

redundancy-master NOTIFICATION-TYPE
STATUS current
DESCRIPTION "System is now master router"
::= { events 1001 }

redundancy-backup NOTIFICATION-TYPE
STATUS current
DESCRIPTION "System is now backup router"
::= { events 1002 }

END
```

A.5. SDK Examples

Event	Description
best-operator.are	This script will scan for operators on startup and choose the one with the best signal
candump.are	This script can be used to receive CAN messages
config-summary.are	This script shows a summary of the currently running configuration.
dio-monitor.are	This script monitors the DIO ports and sends a SMS to the specified phone number.
dio-server.are	This script implements a TCP server which can be used to control the DIO ports.
dio.are	This script can be used to set a digital output port.
dynamic-operator.are	This script will scan Mobile2 and dial the appropriate SIM on Mobile1
email-to-sms.are	This script implements a lightweight SMTP server which is able to receive mail and forward them as SMS to a phone number.
etherwake.are	This script can be used to wake up a sleeping host (WakeOn-Lan)
gps-broadcast.are	This script sends the local GPS NMEA stream to a remote UDP server (incl. device identity).
gps-monitor.are	A script for activating WLAN as soon as GPS position (lat,lon) is within a specified range.
gps-udp-client-compat.are	This script sends the local GPS NMEA stream (incl. serial/checksum) to a remote UDP server.
gps-udp-client.are	This script sends the local GPS NMEA stream to a remote UDP server.
led.are	This script can be used to set a LED
modbus-rtu-master.are	This script can be used to read messages from the serial port.
modbus-rtu-slave.are	This script implements a modbus slave server
modbus-tcp-rtu-gateway.are	This script implements a Modbus TCP RTU gateway
mount-media.are	This script can be used to mount an USB storage stick.
opcua-browse.are	This script will browse for nodes at a remote OPC-UA server.
opcua-json.are	This script polls any temperature nodes of an OPC-UA server and sends them JSON-encoded to a remote server.
opcua-read.are	This script will read the node value at a OPC-UA server.

Event	Description
opcua-write.are	This script will write a new value to a node at a OPC-UA server.
ping-supervision.are	This script will supervise a specified host.
read-config.are	This script can be used to read a configuration parameter.
remote-mail.are	This script reads and sends mails from a remote IMAP/POP3/SMTP server
scan-mobile.are	This script can be used to switch the Mobile LAI according to available networks
scan-wlan.are	This script can be used to switch the WLAN client network according to availability
send-mail.are	This script will send an E-Mail to the specified address.
send-sms.are	This script will send an SMS to the specified phone number.
send-techsupport.are	This script will generate a techsupport and send it to the specified E-Mail address.
serial-read.are	This script can be used to read messages from the serial port.
serial-readwrite.are	This script will write to and read from the serial port.
serial-tcp-broadcast.are	This script reads messages coming from the serial port and forwards them via TCP to remote hosts (and vice versa).
serial-tcsetattr.are	This script can be used to set/get the attributes of the serial port.
serial-udp-server.are	This script reads messages coming from the serial port and forwards them via UDP to a remote host (and vice versa).
serial-write.are	This script can be used to write a message to the serial port.
set-ipsec-route.are	set route to IPSEC server depending on active WWAN / WLAN network
sms-confirm.are	This script will send out a message and confirm its delivery.
sms-control.are	This script will execute commands received by SMS.
sms-delete-inbox.are	This script can be used to flush the SMS inbox.
sms-read-inbox.are	This script can be used to read the SMS inbox.
sms-to-email.are	This script will forward incoming SMS messages to a given E-mail address.
sms-to-serial.are	This script can be used to write a received SMS to the serial port.
snmp-agent.are	This script extends MIB entries of the SNMP agent
snmp-cmd.are	This script issues SNMP set/get commands

Event	Description
snmp-trap.are	This script can be used to send SNMP traps
status.are	This script can be used to display all status variables
syslog.are	Throw a simple syslog message.
tcpclient.are	This script sends a message to a TCP server.
tcpserver.are	This script implements a TCP server which is able to receive messages.
techsupport.are	This transfers a techsupport to a remote FTP server
transfer-file.are	This scripts archives a remote file
transfer.are	This scripts stores the latest GNSS positions in a remote FTP file
udp-msg-server.are	This script will run an UDP server which is able to receive messages and forward them as SMS/E-Mail.
udpclient.are	This script sends a message to a remote UDP server.
udpserver.are	This script implements an UDP server which is able to receive messages.
update-config.are	This script can be used to perform a configuration update
voice-dispatcher-audio.are	This script implements an audio voice dispatcher
webpage.are	This script will generate a page which can be viewed in the Web Manager
write-config.are	This script can be used to set a configuration parameter.

Table A.3.: SDK Examples