www.westermo.com



WeConfig 2.5

User Guide

weCONFIG

Table of Contents

1. WeConfig User Guide 5
   1.1. Getting Started 6
2. Requirements 8
   2.1. Installation 8
   2.2. Device Requirements 12
3. Terminology 13
   3.1. CIDR Notation 13
   3.2. Project Gold File 14
   3.3. Link Local Address 15
   3.4. Projects 16
   3.5. Support Files 18
   3.6. Tasks 19
4. Interface 20
   4.1. Title Bar 21
   4.2. Document Panel 22
   4.3. Context Menu 25
   4.4. Navigation 29
      4.4.1. Device discovery 31
      Network Visualization
      4.4.2. Physical Network 36
      4.4.3. Devices 45
      Configuration
      4.4.4. Basic Setup 48
      4.4.5. Accounts 49
      4.4.6. Virtual Local Area Networks (VLAN) 52
      4.4.7. Network Address-Name Resolution 59
      Configuration ▷ Layer 2 Redundancy
      4.4.8. Fast Reconfiguring Network Topology (FRNT) 60
      4.4.9. High-available Seamless Redundancy / Parallel Redundancy Protocol (HSR/PRP) 63
      4.4.10. Media Redundancy Protocol (MRP) 69
      4.4.11. Rapid Spanning Tree Protocol (RSTP) 71
      4.4.12. Ring Coupling (RiCo) 72
      4.4.13. Firewall 75
      Configuration ▷ System
      4.4.14. Simple network Management Protocol (SNMP) 103
      4.4.15. CPU 110
      4.4.16. Date/Time 111
      4.4.17. Logging 112
      Configuration ▷ Routing
      4.4.18. OSPF 121
      4.4.19. RIP 124
      4.4.20. Static routing 125
      4.4.21. Virtual Router Redundancy Protocol (VRRP) 127
      Configuration ▷ Ports

4.4.22. Ethernet 132
4.4.23. Port Monitoring 134
4.4.24. SHDSL 136
4.4.25. 802.1X 138
4.4.26. Licensing 139
4.4.27. Alarm 141
4.4.28. WeConnect 152
4.4.29. Aggregates 156
4.4.30. Access Point 158
Per-Device Views
4.4.31. Route Table 160
4.4.32. Properties 161
4.4.33. Attachments 163
4.4.34. Configuration Files 164
4.4.35. Communication Summary 165
4.4.36. Cellular 166
Maintenance
4.4.37. Backup 169
4.4.38. Firmware Upgrade 172
4.4.39. Attachments 179
4.4.40. Device Access Settings 180
4.4.41. Clone or Replace Device 185
4.4.42. CLI Scripts 189
Diagnostics
4.4.43. Diagnostics 191
4.4.44. Syslog 193
4.4.45. Traps 194
Support
4.4.46. General Log 195
4.4.47. Issues 196
4.4.48. Reports 197
4.4.49. Security 199
4.4.49.1. Management Hardening 199
4.4.49.2. Port Protection 202
4.4.49.3. Routing Hardening 204
Tools
4.4.50. Traceroute 206
4.4.51. Subnet Calculator 207
4.4.52. SHDSL Reach Calculator 208
4.4.53. SHA256 Hash Calculator 209
4.4.54. Ping 210
4.4.55. Maximum MTU Discovery 212
4.4.56. Project Template 213
4.4.56.1. Import Devices from CSV File 213
4.4.56.2. Export from Current Project 215
4.4.56.3. Build Network from Template 217

4.4.57. Refresh 222
4.5. File Menu 223
4.5.1. About 224
4.5.2. Application Settings 225
4.5.3. Portable Installations 231
4.6. Components 232
4.6.1. CIDR Address box 232
4.6.2. IPv4 Address box 233
4.6.3. Log box 234
4.7. Notifications 235
5. Algorithms 236
5.1. Recursive Discovery 236
5.2. Network Order 240

1. WeConfig User Guide

About

WeConfig is a powerful and user-friendly network configuration tool designed to simplify the management and deployment of industrial networks. Developed by Westermo, WeConfig provides a comprehensive suite of features that enable users to efficiently configure, monitor, and maintain their network infrastructure.

Key Features

- Network Discovery: Quickly identify and visualize all devices within your network.
- Centralized Management: Manage all network devices from a single interface, reducing the complexity of network administration.
- Configuration Templates: Apply consistent configurations across multiple devices with ease.
- Diagnostic Monitoring: Monitor network performance to identify and remediate network issues.
- Firmware Management: Easily update firmware on multiple devices simultaneously.

WeConfig is designed to support a wide range of Westermo devices, ensuring seamless integration and optimal performance. Whether you are managing a small network or a large-scale industrial system, WeConfig provides the tools you need to keep your network running smoothly and efficiently.

Legal Information

The contents of this document are provided “as is”. Except as required by applicable law, no warranties of any kind, either express or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose, are made in relation to the accuracy and reliability or contents of this document. Westermo reserves the right to revise this document or withdraw it at any time without prior notice.

Under no circumstances shall Westermo be responsible for any loss of data or income or any special, incidental, and consequential or indirect damages howsoever caused. More information about Westermo can be found at https://www.westermo.com[1]

[1] https://www.westermo.com

1.1. Getting Started

In order to get started using WeConfig, you will need to install the application. This is done by downloading the installer (Autoupdating[1] / MSI[2]) and running it.

Once the program is up and running, make sure that the machine running WeConfig is connected to the devices you want to configure. This can be done directly via cable or over a VPN connection, but WeConfig is best utilized over direct connections to the devices, in order to have a better understanding of the network topology at play.

Now, you can start using WeConfig. The following sections will guide you through the interface and its components, as well as how to use them. As a recommended start, it is suggested you visit the Topology and Devices panels, as they are the main views of WeConfig and will give you a good overview of the network you are working with.

Secondly, consult the device discovery section, as it will help you discover the devices in your network and add them to WeConfig's knowledge of the network, as well as allow you to align your own IP Address to match what the devices may need for communication.

How to: Set basic properties on a factory-defaulted WeOS 5 device.

Prerequisites

- This guide assumes you have administrator rights on your Windows PC
- The PC has an Ethernet NIC configured to acquire its IP address via DHCP
- You have the latest version of WeConfig installed
- The WeOS 5 device is factory-defaulted and has not been configured yet
- The WeOS 5 device is connected to the PC via an Ethernet cable

Step 1

Open WeConfig and make sure the device is connected to the PC via an Ethernet cable. The device should be powered on and in factory-default state.

Step 2

Navigate to the device discovery task and select the NIC that the device is connected to and press "Start Scan", wait for the operation to finish. This should discover a device with a Link-local IP address and identify at as a WeOS 5 device.

> *i* Info
>
> Before leaving this task, consider editing the NIC and adding an IP address that exists within the address-space of the IP address you will be changing the devices IP address to, such as `192.168.123.121`. This will allow you to communicate with the device after the IP address change.
>
> This is not a requirement, but it is a good practice to ensure that you can communicate with the device after the IP address change.

[1] https://weconfigservices.westermo.com/api/setup
[2] https://weconfigservices.westermo.com/api/msi

Step 3

Navigate to the Topology panel and click to select the device. There should be a visible link between the device and the PC in this view.

Step 4

Navigate to the Basic Setup task. Therein you will find the device selected, fill in a new IP Address, such as `192.168.123.1`, and a new netmask, such as `255.255.255.0`. Optionally you may also fill out a hostname, such as `myWeOSDevice`, a location, such as `myTestBench`, and a new gateway, such as `192.168.1.254`. Press apply and wait for the operation to finish. This will change the IP address of the device to the new IP address and netmask, as well as set the hostname, location and gateway if they were filled out.

Step 5

Right click the device in the Topology panel and click “Refresh Selected”. This will trigger a rescan of the device. If you have set a matching IP on your machine’s NIC, you should still be able to communicate with the device. If you have not, WeConfig should fail to communicate with the device, and indicate such in the Device List panel. If this is the case, you will need to set a matching IP address on your machine’s NIC in order to communicate with the device again.

2. Requirements

2.1. Installation

Prerequisites

Hardware & Software requirements

Type	Minimum	Recommended
Processor	64-bit (x86-64) compatible	> 2.0 GHz, > 4 Cores
Memory	2 GB RAM	16 GB RAM
Graphics	DirectX 9	DirectX 12
Display	1920x1080 screen resolution	2560x1440 screen resolution
Operating System	Windows 10 version 1607 or .AppImage compatible Linux distro	Windows 11 version 24H2 or later or Ubuntu 24.04 or higher or other .AppImage compatible Linux distro

Network Packet Capture

- One of:
  - WinPcap: Version 4.1.3 or later.
  - Npcap: Version 1.6 or later.

> **Note**
>
> If you decide to use Npcap, ensure you have a valid license. Licensing details are available at Npcap Licensing[1].

> **Note**
>
> WeConfig may offer to install WinPcap if it detects the need.
>
> If so, a notification will appear in the user interface with a link to install WinPcap.

> **Info**
>
> If neither WinPcap nor Npcap is installed, WeConfig will operate in a reduced mode with limited functionality.

> **Warning**
>
> WeConfig may not detect the connection between the computer and the network if the Network Interface Card (NIC) discards LLDP frames. This issue is common with low-end USB NICs.

[1] https://npcap.com

Installation Packages

Standard Windows Installation Package

- Executable Name: WeConfigSetup.exe[1]
- Usage: Recommended for general use.
- Installation Process:
  - Installs WeConfig in the user's directories.
  - WeConfig, once installed, will be able to automatically update to newer versions if configured to do so, see Application Settings
  - Administrative access rights are not required for this installation.

MSI Windows Installation Package

- Executable Name: WeConfig.Msi.(Version).msi[2]
- Usage: Recommended for strict IT environments where automatic software updates are prohibited.
- Installation Process:
  - Installs WeConfig into a desired location, typically C:\Program Files\ or similar.
  - WeConfig, once installed, will not be able to automatically update itself, but will attempt to notify the user when an update becomes available.
  - Administrative access rights are required for this installation

Linux AppImage Installation Package

- Executable Name: WeConfig.AppImage[3]
- Usage: Running WeConfig on Linux
- Installation Process:
  - Download the AppImage file.
  - Make the file executable using `chmod +x WeConfig.AppImage`.
  - Run the AppImage to start WeConfig.

> **Note**
>
> On Linux, the AppImage allows running WeConfig without requiring installation privileges. However, unless ran with elevated permissions (e.g., using `sudo -E`), certain operations may be restricted:
> - IpConfig based scanning will not work
> - WeConfig will be unable to receive SNMP Traps or Syslog messages
> - Ping/Traceroute/MTU tools will not work.

Portable Installation Package

- Executable Name: Irrelevant, generated as a full directory with all relevant files
- Usage: Recommended for airgapped systems.
- Installation Process:
  - Generated from an existing installation of WeConfig
  - Will not attempt to check for updates

[1] https://weconfigservices.westermo.com/api/setup
[2] https://weconfigservices.westermo.com/api/msi
[3] https://weconfigservices.westermo.com/api/appimage

- Any shortcuts and similar handling must be set up manually.
- See Portable Installations for more information.

> 💡 Can I have multiple versions of WeConfig installed?
>
> Yes, you can have multiple versions of WeConfig installed on the same machine so long as they arise from different installation packages: Standard, MSI, or Portable.
>
> Each of these packages will operate completely independently, so can be used side-by-side for different versions.

Usage requirements
Certain functionality in WeConfig requires an elevated level of access. These requirements exist due to the functional requirements of WeConfig as a network configuration manager, which implies a certain level of desired access, as detail below.

Requirement 1: Manipulation of network adapter addressing
In order to function properly against local network devices that lack a DHCP server, which is often the case in offline networks, WeConfig needs to be able to change the the IP address of the network adapter which is being used to connect to these devices. Alternatively, the user running the program can, by themselves, alter their IP address on said adapter, but doing so requires a similar level of permission be granted to the user.

Requirement 2: Creating firewall rules
In order to receive certain information from the network, WeConfig needs certain ports to be open. WeConfig comes with the built-in ability to make these firewall adjustments at the users behest and with their confirmation, but needs authorization to actually execute these changes. Alternatively, if the required changes are deployed to the managing device ahead of time, this functionality becomes unnecessary. The rules that WeConfig may attempt to create are as follows:

Rule name	Protocol	Port	Profiles
Snmp Trap WeConfig	UDP	162	Domain & Public & Private
IpConfig WeConfig	UDP	5098	Domain & Public & Private
Syslog WeConfig	UDP	514	Domain & Public & Private
Mdns WeConfig	UDP	5353	Domain & Public & Private

Online activity
Some functionality of WeConfig relies on occasional internet activity in order to retrieve additional information and data.

The following functionality requires internet connectivity to function properly:

Functionality	Explanation	Offline fallback
Device definitions	Model names, icons and port maps, used to render the network topology map	Keeps any previously retrieved entries

Functionality	Explanation	Offline fallback
MAC OUIs	Organizational Unique Identifier database, used to help identify unknown devices	Keeps any previously retrieved entries
Firmware Rules	Rules specifying in what order firmwares must be installed	Keeps any previously retrieved entries
Licenses	Limits or enables extended functionality in WeConfig	Cached license will be kept until expiry
Firmwares	Download specific versions of firmware for devices	Previously downloaded firmware is kept
Application Usage	Gathers usage and performance statistics, see Statistics Gathering	Do not send

Statistics Gathering

WeConfig will by default send usage statistics to a centralized database over the Internet. The information sent does not contain any identifying information, nor any sensitive network information. It is used to measure performance, how the application is used, and unanticipated problems.

Information will be sent unless the user opts out. Every time WeConfig starts, WeConfig will ask the user whether they consent to this or not until a choice is made. It is possible at any time to revert any choice in the Application Settings.

2.2. Device Requirements

WeConfig is designed for Westermo devices with WeOS version 4.28 or later, WeOS version 5.15 or later, and devices in the Ibex, MRD, and Merlin families. WeConfig will however find and try to display some information about other types of devices too. Earlier WeOS versions might have functional features, but they are however not supported.

For ideal compatibility with WeConfig, devices should ensure that the following services are configured, if available:

Functionality	Description	Reasoning
IPConfig	A Westermo specific protocol available on WeOS 4	Used for discovery of WeOS 4 devices, as well as configuration of factory defaulted WeOS 4 devices prior to WeOS 4.33
HTTPS	Standard web protocol	WeConfig must be provided with relevant credentials via the Device Access interface. Used for backup, restore, firmware, and bootloader upgrade.
SNMP	Simple Network Management Protocol	WeConfig must be provided with relevant credentials via the Device Access interface. For full trap-reception functionality, MS Windows Trap Host server needs to be disabled. WeConfig has its own built-in trap host server. Used for diagnostics and discovery, primarily
LLDP	Link Local Discovery Protocol	Used by WeConfig to establish the network topology map, as well as a means of recursive discovery
SSH	Secure Shell	Used by WeConfig for nearly all forms of configuration as well as diagnostics and discovery

When launching interactive SSH sessions to the devices (e.g., via context menu), WeConfig will start the OS built-in SSH client command in a terminal window. If it has not been installed as part of the OS, please do so.

3. Terminology

3.1. CIDR Notation

WeConfig frequently makes use of so called CIDR notation, which is a way of writing an IP address coupled with a netmask in the format IP/BitCount, where IP is the full IP address (either IPv4 or IPv6) and BitCount is the number of bits from the start of the netmask that are 1.

Examples

CIDR Notation	IP Address	Netmask	Netmask (Bitwise)	Subnet
10.12.13.14/8	10.12.13.14	255.0.0.0	11111111 00000000 00000000 00000000	10.0.0.0
172.123.234.1/16	172.123.234.1	255.255.0.0	11111111 11111111 00000000 00000000	172.123.0.0
192.168.1.2/24	192.168.1.2	255.255.255.0	11111111 11111111 11111111 00000000	192.168.1.0
1.2.3.4/32	1.2.3.4	255.255.255.255	11111111 11111111 11111111 11111111	1.2.3.4
13.13.13.13/13	13.13.13.13	255.248.0.0	11111111 11111000 00000000 00000000	13.8.0.0
1.10.20.40/17	1.10.20.40	255.255.128.0	11111111 11111111 10000000 00000000	1.10.0.0

3.2. Project Gold File

Project gold file is a template file which represents an entire network with the devices and all their connections and settings. This gold file can be used to setup new networks on network topologies that are exactly the same regarding the number of devices, model and physical connections.

3.3. Link Local Address

A link-local address is a type of IP address that is used for communication within a single network segment or link. These addresses are not routable and are used for local network communication only. They are automatically configured on most network switches that support them and do not require manual configuration or a DHCP server.

In IPv4, link-local addresses are in the range `169.254.0.0` to `169.254.255.255`.

In IPv6, link-local addresses start with the prefix `fe80::/10`.

3.4. Projects

In WeConfig, a project refers to a collection of items representing an observed network. A project can be saved to the file system, and uses the `.nprj` (Network project) extension by default when saved. The actual file itself is a ZIP-archive and can be opened/inspected using software like 7z.

Contents

A project file contains the following information:

- A network specification
- A data directory containing
  - Cached user notifications and their state.
  - Project-wide attachments
  - A device-attachments directory containing
    - A subdirectory per device that has attachments containing
      - Any device specific attachments
- A configuration file directory containing
  - A subdirectory per device that has backups containing
    - any backups associated with the device

Network specification

The network specification, called `Project.xml` within the project file, is an XML-file that details the topology of the network, along with any properties attached to the device. This XML file has some specific properties of note:

Project version

The first element of the XML-File looks roughly as follows

```
<Project Version="3.1" xmlns="http://westermo.com/weconfig">
```

Take note of the version attribute, which is metadata for WeConfig that states which version of the project structure is in use. WeConfig 2.5 writes version 3.1 and can read both 3.0 (used by WeConfig 1.21) and 3.1.

Physical Network

Under the `<PhysicalNetwork>` element, two primary sub-elements exists, `<Nodes>` and `<Connections>`. Nodes contains the set of devices present in the network, and Connections contains a connection map of how the different devices are linked together.

Device element

Under the `<Nodes>` element, several `<Device>` elements are usually found. This element is a serialization of WeConfigs knowledge of the device, and contains information including, but not limited to:

- Model
- Firmware version
- Management IP Address
- Mac Address
- Ports
- VLAN
- Routes

- And more…

There also usually exists a special element under `<Nodes>` called the `<WeConfigPC>` node, this corresponds to the computer running WeConfig that discovered the devices.

Connection element

Under the `<Connections>` element, a variety of connection elements exist, these tend to be one of:

- `<AggregatePortConnection>`: indicating a n-to-n port connection between devices where all ports involved are known.
- `<WeConfigConnection>`: indicating a 1-to-1 port connection between devices where at least 1 port involved is unknown.
- `<RoutedConnection>`: indicating a 1-to-1 connection between devices where neither port is known, typically as a result of being discovered via tracing routes.

Together, these connection elements span all known connections between devices.

Attachments

Attachments are generic files that have been associated either with the project itself or with a specific device. They can be of any file type and with any content. For more details, see Attachments and Device Attachments.

Backups

A backup is a copy of a devices configuration file at a certain date and time, possibly with some attached metadata used by WeConfig. For more information, see Backups

3.5. Support Files

Support files are used to generate information that is useful for developers of WeConfig to help solve identified problems. They are ZIP archives containing the following information:

- The log files associated with WeConfig
- The database-synchronization status
- The exact files that make up the instance of WeConfig that generated the support file.
- The processor architecture of the computer that generated the support file
- The operating system architecture of the computer that generated the support file.
- The operating system version of the computer that generated the support file.

The support file does not contain any references to the actual topology used at the time of generation.

3.6. Tasks

A task in the context of WeConfig refers to a graphical user interface element that is mutually exclusive with all other tasks. In other words, only a singular task can be active at any one time. Attempting to navigate to a new task will close any prior task, although a warning may be issued if attempting to navigate away from a task that is currently running.

Tasks are identified in the user interface by two common components. Firstly, the tab hosting the task will always be named “Current Task:” followed by the name of the task at hand.

Secondly, all tasks have a button at the lower end of their interface, which executes said task. This button often contains the text “Apply” but may have other content, depending on the particular task at hand.

Typical elements of WeConfig that are handled as Tasks are operations that change the network state, may temporarily alter device behavior, or discover devices. As such, most elements under the “Configuration” section, as well as the “Device Discovery” section of the Navigation are considered tasks. Other notable tasks are Firmware Upgrade, Backup, Clone Device and CLI Scripts

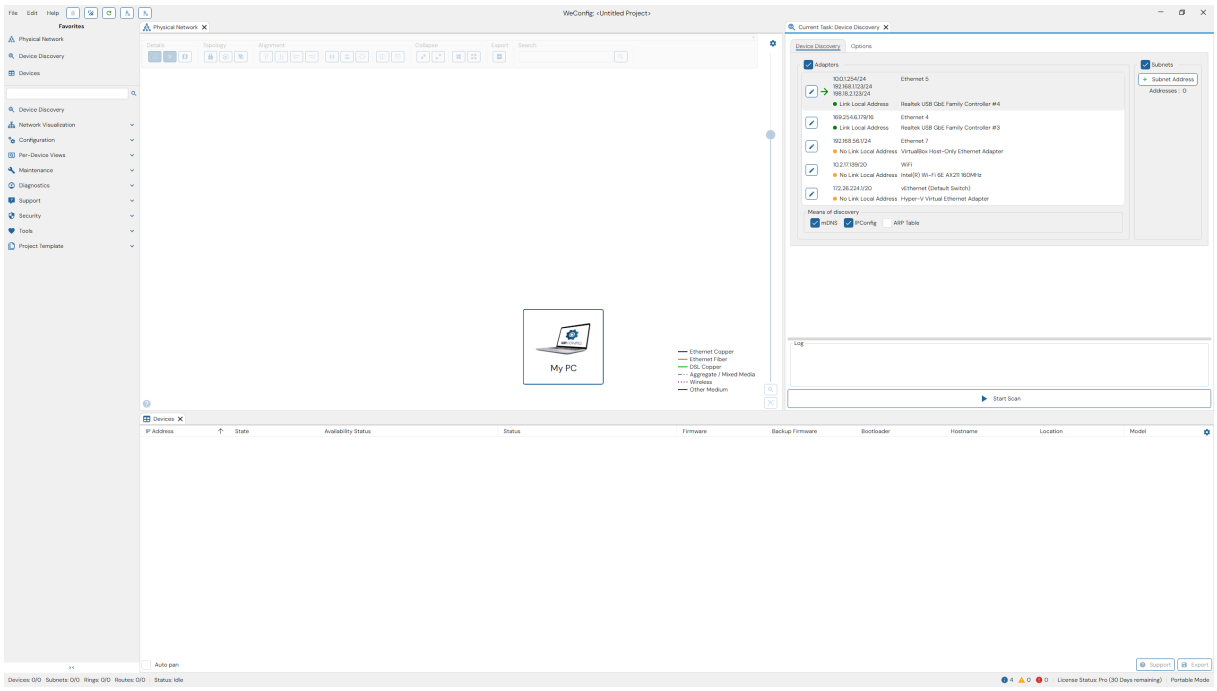
If the task requires any form of device selection, that selection will mirror the current selection in the Network Topology and/or Device List

Staged Tasks

Some tasks may have several stages, as indicated by a step-by-step progress bar at the top of the user interface indicating the stages present. You can always return to a previous stage in a staged task by clicking on the associated stage step in the progress bar.

4. Interface

WeConfig's user interface is divided into three primary areas:



1. The Document panel
2. The Navigation menu
3. The Title bar

Zoom In/Out

The entire user interface may be magnified, or zoomed in/out, just like web browsers.

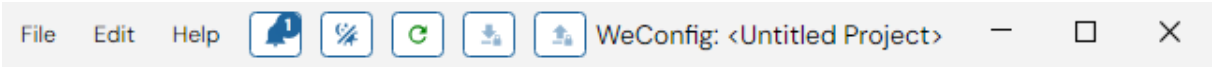
To make the user interface elements bigger (zoom in), press Ctrl+Plus or Ctrl+Mouse Wheel Up.

To make the user interface elements smaller (zoom out), press Ctrl+Minus or Ctrl+Mouse Wheel Down.

To reset any zoom, press Ctrl+Zero. The status bar will also show the current magnification if it differs from normal.

4.1. Title Bar

At the top of WeConfig sits the titlebar, which displays the application name and provides window controls along with a few other options.



File menu

Leftmost you will find the File menu, which provides several commands for file / project management as well as application settings.

Quick Action Buttons

To the right of the file menu, you will find various "quick action" buttons that provide easy access to frequently used features and settings. These buttons are, as depicted above, in left-to-right order:

Button	Description	Hotkey
Notifications	Opens the notifications panel	N/A
Theming	Toggles the theming options	Ctrl+Shift+T
Refresh	Trigger a refresh of all devices	F5
Commit All	Commits all pending running changes to startup configuration on applicable devices	F7
Revert All	Reverts all pending running changes to startup configuration on applicable devices	F8

Application/Project name

Next, you will find the application name (WeConfig) following by the name of the currently open project (or <Untitled Project> if no saved project is open).

Window controls

Finally, to the far right you will find the window controls, which allow you to minimize, maximize, or close the application window.

4.2. Document Panel

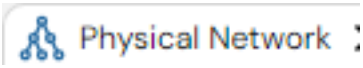
The document panel is the central piece of the WeConfig user interface, and is a free-form docking space where different panels can be arranged to your desire and specification. Any panel can also be undocked entirely as a free-floating window to be placed externally, useful for running WeConfig in a multi-screen setup.

The layout is saved in-between program runs on a per-screen-setup basis. As such, WeConfig will remember where you have placed its different panels after it is closed and reopened.

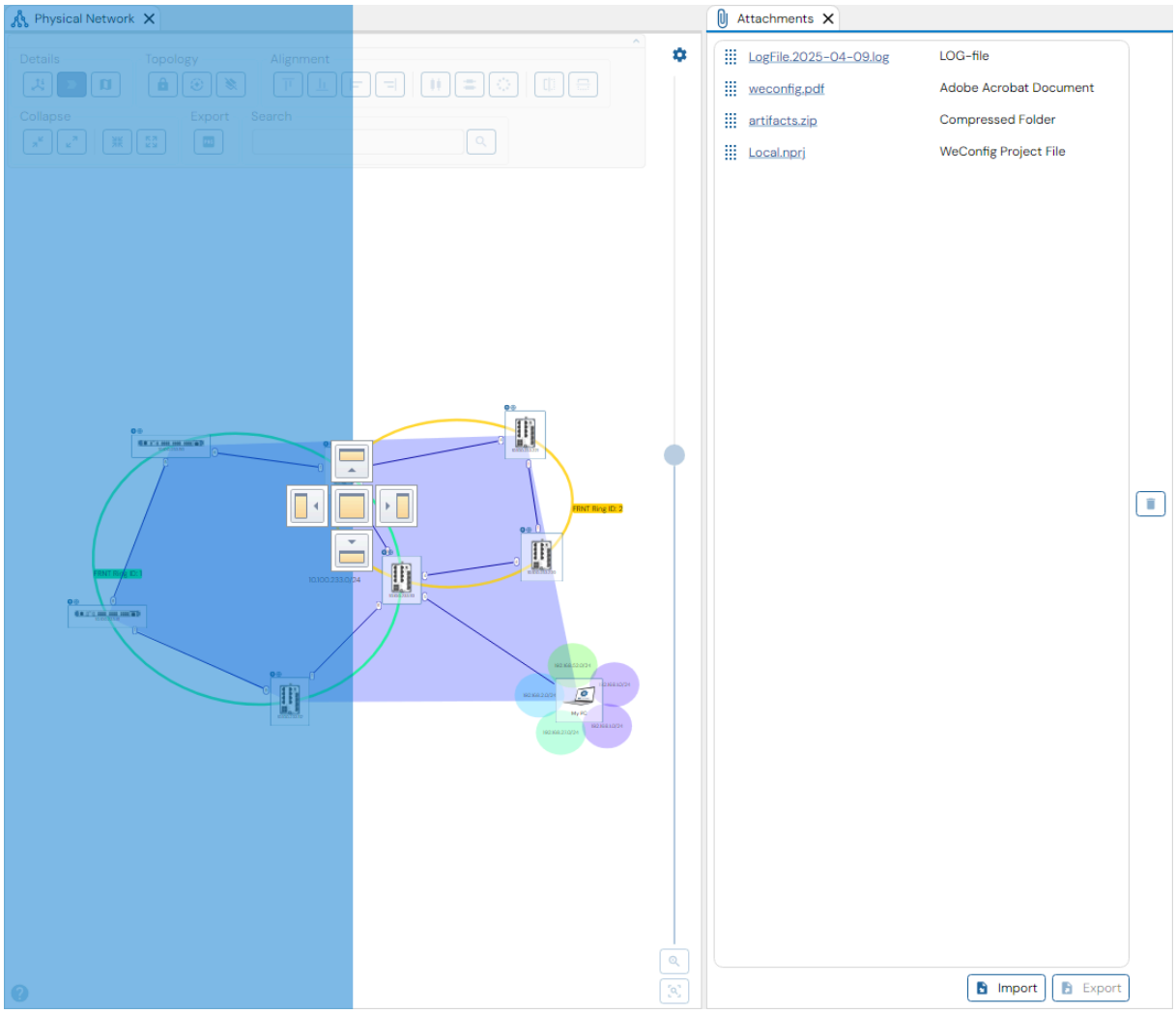
> **Tip**
>
> If you have lost track of a panel or want to go back to the default view, you can always reset the layout

Docking

In order to dock a panel somewhere, start by left-clicking and dragging the panel header, as depicted below:



WeConfig will then display a number of dock anchors, as seen below:



Where the one highlighted in a brighter way than the other anchors is the currently selected docking point (in this case, the inner left one in the middle of the 'Physical Network' panel), drag the mouse over another anchor to select a different one. A highlight area, seen here to the left in the 'Physical Network' panel provides a preview of where the docked window will end up. In order to complete the dock, simply release the left mouse button over one of the anchors.

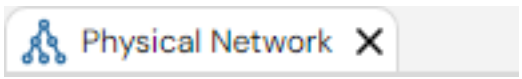
Floating panels

In order to put a panel into floating mode, start the procedure to dock it, but do not select any of the anchors, upon release, the panel will then be put into floating mode.

Any panel can also be put into float mode by the panel headers context menu, selecting the "Float" option. This context menu also allows you to re-dock a floating window.

Closing panels

Panels can be closed either via the panel headers context menu, selecting the "Close" option, or by clicking on the panel header with the middle mouse button, or via the 'X' button located to the rightmost part of the panel header, as seen below



Status bar

At the very bottom of the document panel you will find the status bar:



This bar communicates several pieces of information, from left to right:

Selection indicator

Leftmost you will find the selection indicator, which displays the number of devices, subnets, and routes in the network, in the format Selected/Total. In the above example, 1 device is selected out of a total of 15 devices in the network.

Status indicator

Next is the status indicator, this contains a short summary of what WeConfig is currently doing. When WeConfig is not doing anything of interest, the status will be listed as “Idle”.

Issue indicator

Next, to the right hand side of the screen, you will find the issue indicator, this will let you know if there are any issues that WeConfig would like to suggest you take care of, their count and severity. With the above example indicating there are 5 issues warranting inspection, 4 are ranked as “informatory” and 1 as “warning”.

> **Tip**
>
> Clicking the issue indicator will open up the issue panel

License indicator

Next, conditionally, is the license indicator, which displays the current licensing status of the software.

> **Tip**
>
> Interested in acquiring a licensed version of WeConfig? Contact Westermo sales.

License indicator context menu

The license indicator also has a context menu with two options.

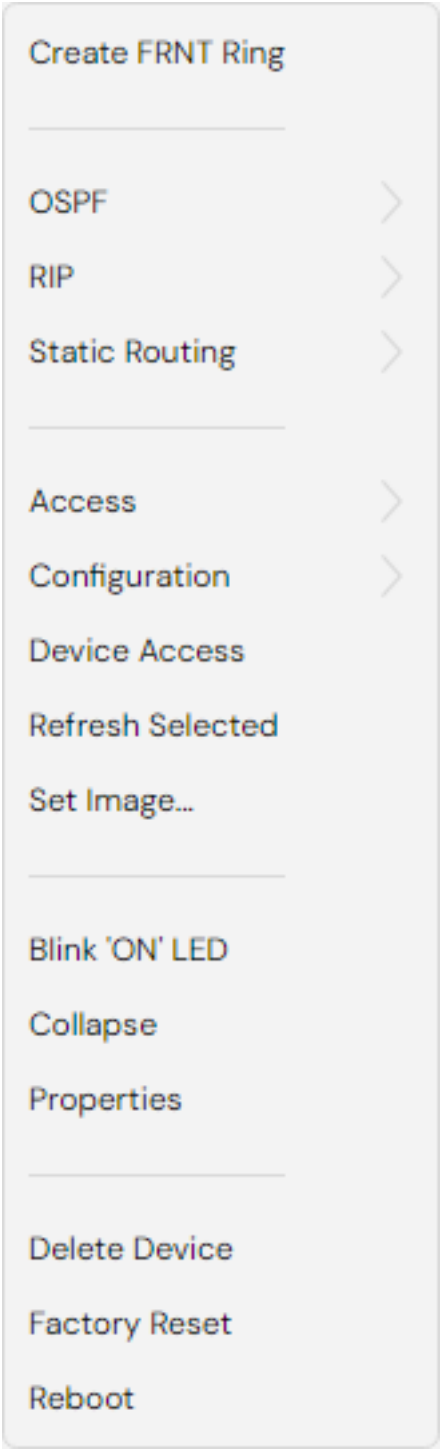
- Create temporary license: This context menu will allow you to create a finite amount of temporary “trial” licenses that are valid for one day, perfect for trying out the expanded features that a licensed version of WeConfig will offer.
- Activate license key: This context menu will allow you to activate a license key for the full Pro version of WeConfig.

Portable mode indicator

Finally, to the rightmost part of the status bar, is the portable mode indicator, this is only visible if, and an indicator that, WeConfig is running in portable mode.

4.3. Context Menu

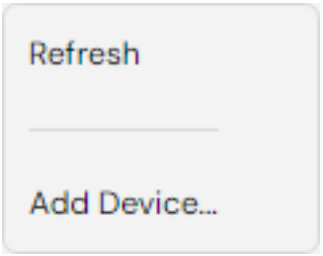
The context menu is one of the primary ways to interact with the network from the topology view and the device list. It can be accessed by right-clicking anywhere in the topology, or on a selection of one or more devices in the device list.



The exact contents of the context menu depends on, well, context. For example, WeConfig will only display context menu options related to OSPF when the current device selection supports any relevant OSPF-based action.

There are a lot of items that may populate this menu, some are listed here below:

Add Device



Add device allows you to add devices ad-hoc to the topology. WeConfig will ask for details pertaining to the device, which may vary for the type of device you are attempting to create, but will always include:

- Management IP address
- Hostname.
- Location, optional.

If attempting to add a type of device that WeConfig is not aware you, you may also add a manually selected image to represent it in this view.

> Note
>
> This option is only visible if no devices are selected.

Set Image

This context menu option allows you to set a custom image for a device that is not recognized by WeConfig, i.e not a known Westermo switch or similar.

> Note
>
> This option is only visible when a singular, unknown device is selected.

Blink “ON”-LED

Using this context menu option option will start cause devices’ “ON”-LED to blink which make it easier to identify the device visually. The device will keep blinking as long as it is selected.

> Note
>
> This option is only visible when a singular device that supports LED-blinking is selected.

Access

This context menu contains three choices to access the selected device, either through HTTP, HTTPS or SSH/CLI. Click on the associated protocol to open up an external access attempt to the targeted device If SSH/CLI is selected, the configured SSH client is used. If HTTP or HTTPS is selected, an attempt is made to access the device via the default system browser.

> **Warning**
>
> This feature is provided as best-effort attempt, particularly when using it on unknown devices that WeConfig has no awareness of regarding support for HTTP / HTTPS / SSH.

> **Info**
>
> If the device has a known configured Public-key authentication based account, WeConfig may try and authenticate the SSH access session using that account.

> **Note**
>
> This option is only visible when a singular device is selected.

Add Connection
This context menu option is used to set the connection between two devices manually. If one or more of the devices is an unknown devices, the user must specify a port name to connect to manually by typing the name of the port in the provided field for the unknown device(s).

> **Note**
>
> This option is only visible when a single pair of devices is selected.

Delete Devices
This context menu option removes the selected devices and all their connections from the topology.

> **Note**
>
> This option is only visible when one or more devices are selected

> **Info**
>
> This option is also hotkeyed to the 'Delete' keyboard button.

Reboot
This context menu option reboots one or more selected device.

> **Note**
>
> This option is only visible when all selected devices support being rebooted in a manner known to WeConfig.

Factory Reset
Factory Reset resets the selected devices to the factory configuration.

> **Note**
>
> This option is only visible when one or more of the selected devices support factory reset in a manner known to WeConfig

> **Warning**
>
> Factory reseting devices may lead to loss of connection, use with caution.

Disable/Enable SNMP
This context menu option enabled or disables SNMP on MRD / BRD devices.

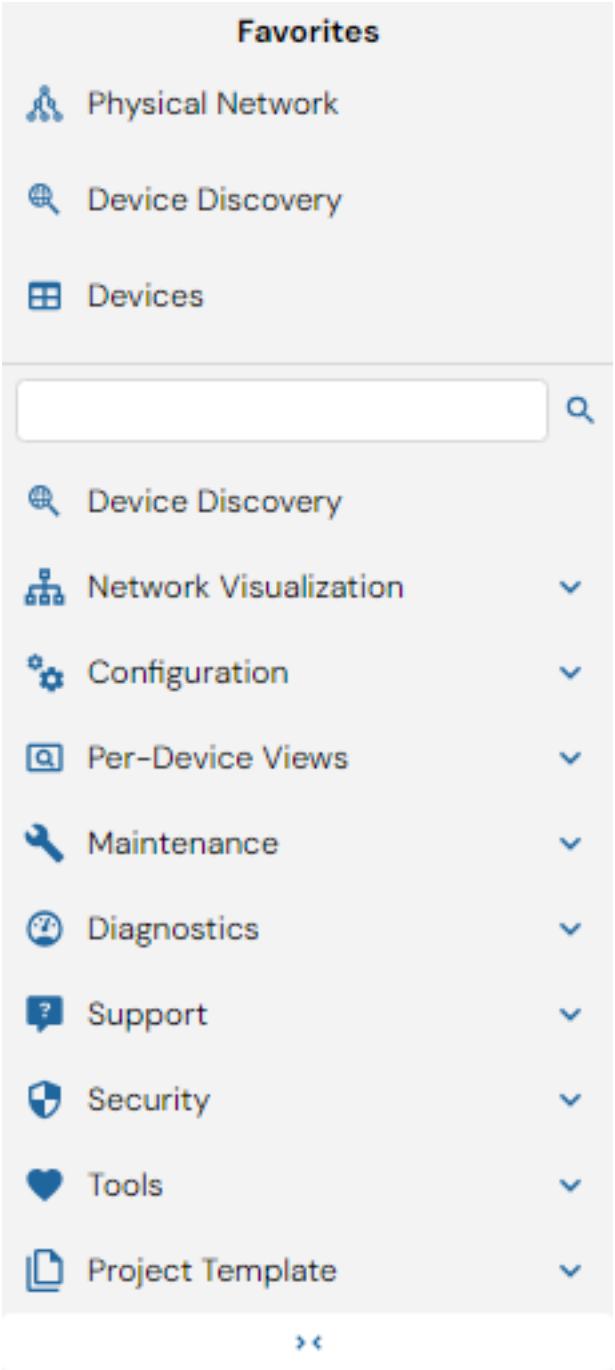
> **Note**
>
> This option is only visible when MRD or BRD devices are selected.

Refresh Selected
This context menu option triggers a refresh operation on the selected devices.

> **Note**
>
> This option is only visible when one or more devices are selected

4.4. Navigation

On the left-hand side of WeConfigs user interface, you will find the navigation menu, depicted below:



From top to bottom, the navigation panel contains a search field, the navigation tree, and Favorites.

Search and navigation tree

At the top of the panel, the search field filters the navigation tree. The tree contains the panels available to open, organized into categories that can be expanded or collapsed.

Left-clicking a non-category item opens the corresponding panel in the document panel. Entries for views unavailable under the current license are marked as unavailable.

To add a view to Favorites, right-click it and select “Pin to favorites” from the context menu.

Favorites
Favorites appears below the navigation tree and lists the panels pinned for quick access. Initially, it contains `Physical Network`, `Device Discovery`, and `Devices`.

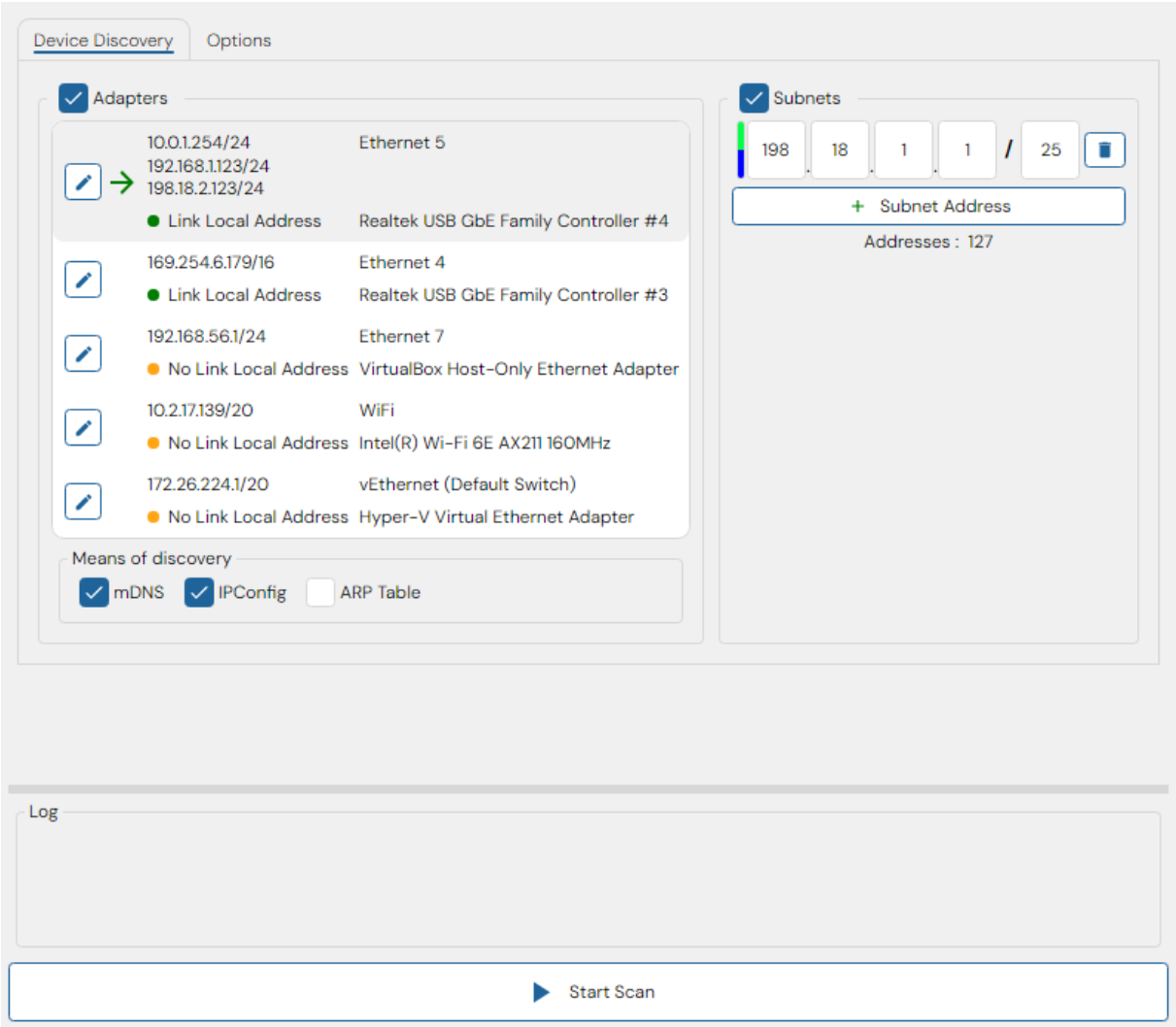
Left-clicking a favorite opens its panel in the document panel. To remove a favorite, right-click it and select “Unpin from favorites”.

At the bottom of the navigation menu, you will find a >< button, clicking this button collapses all elements of the navigation menu except for the top level buttons, if you need more space.

4.4.1. Device discovery

WeOS 4 WeOS 5 Ibex XRD Merlin other

The device discovery task exists to enable local and remote identification of network devices. The image below display the typical user interface for this task.



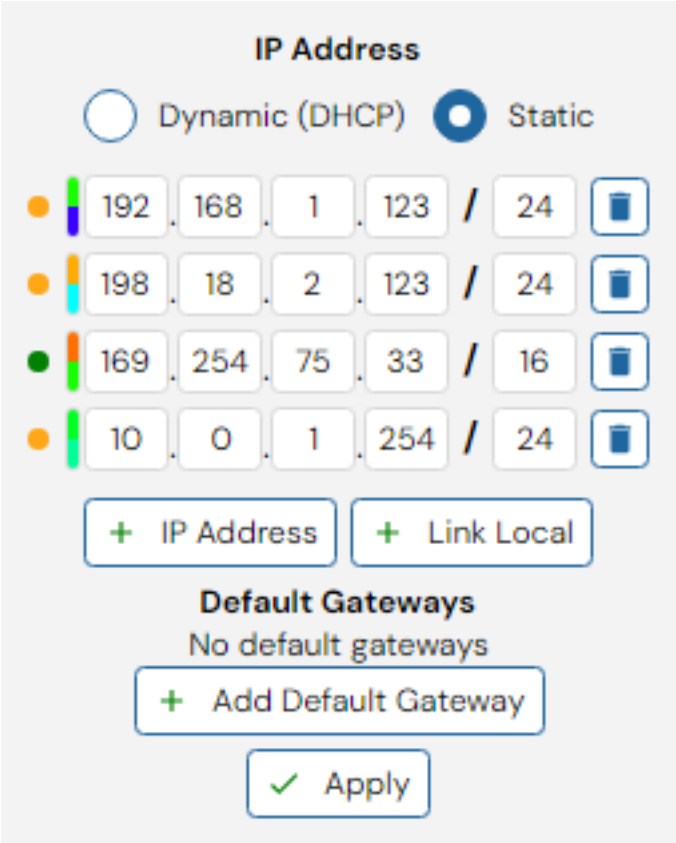
Interface Elements

Network Interface selector

This element lists the available network interfaces on the PC, with each entry containing the following data:

- A list of CIDR-notation network addresses associated with the interface.
- An indicator for whether or not the interface has a link local address.
- The interface name, seen as, for example `Ethernet 5` at the top entry in the image above.
- The device name associated with the interface, for example `Realtek USB GbE Family Controller #4` in the image above.
- An edit button.

Edit button
The edit button, attached to each individual network interface listed, allows a user to bring up an editor for that interface which appears roughly as per the image below:



This editor allows the user to specify either a number of static CIDR-notation addresses for the interface, or set DHCP assignment for the interface.

Additionally, the editor allows the user to configure a default gateway for the specific interface, in order to ease routing access.

> *i* Info
>
> On Windows, using the editor functionality requires at least `NET ADMIN` or `Administrator` privileges. WeConfig will request these privileges as needed.
>
> On Linux, when not running as `sudo` or as the root user, WeConfig will attempt to run `pkexec` to gain the necessary privileges for the editor operation

Interface Discovery Methods
Located beneath the network interface selector, this element contains a selection of discovery methods to be applied to the selected interfaces. The options are as follows:

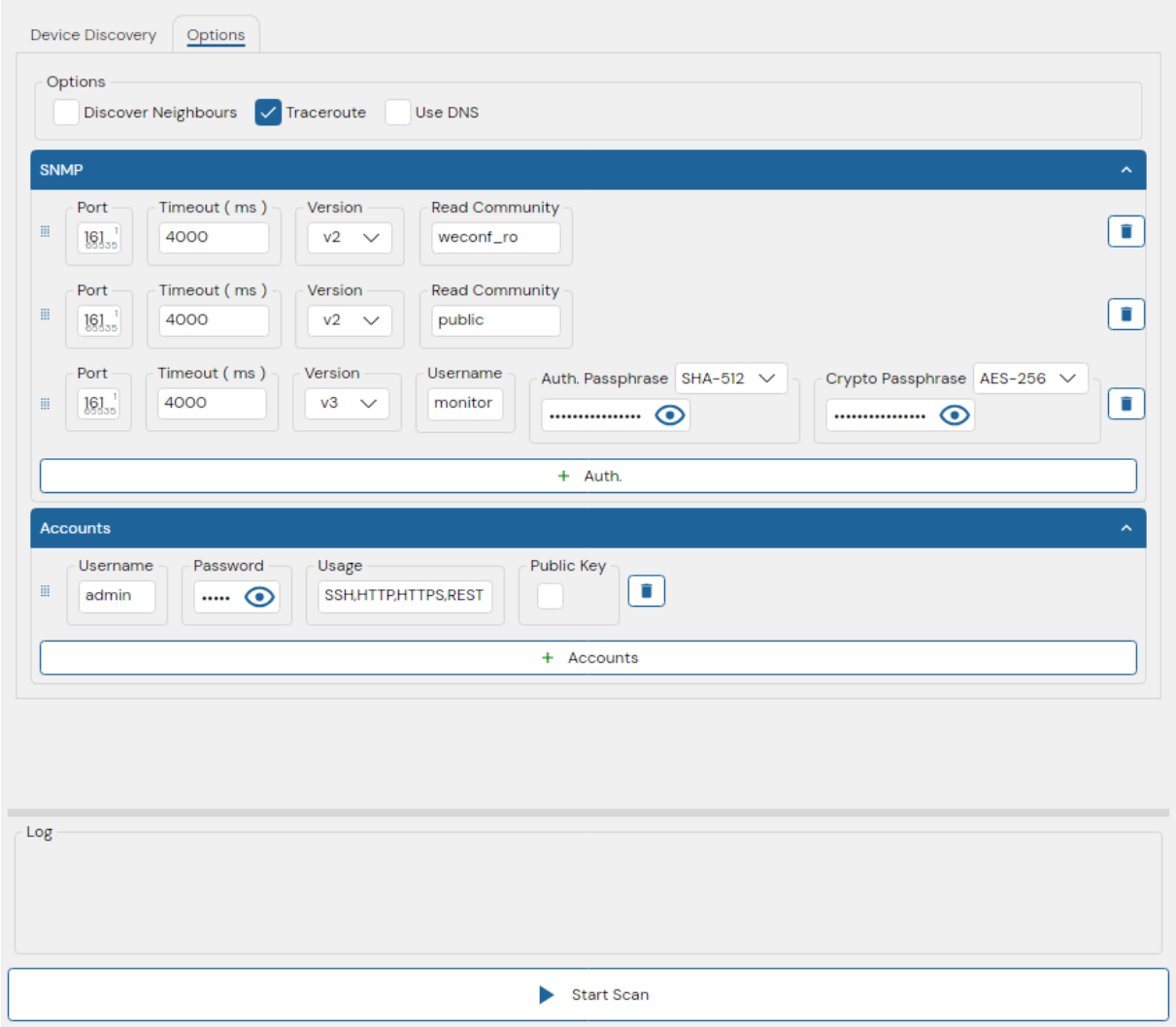
Option	Description	Limitations
mDNS	Multicast domain name resolution, a protocol that broadcasts a request for	mDNS calls do not jump across routes, and thus will only discover layer 2 connections.

Option	Description	Limitations
	devices to identify themselves on the specified interface.	
IPConfig	A Westermo specific protocol used to discover WeOS 4 units,	Provides a fairly limited amount of metadata, limited similarly to mDNS to only layer 2 connections.
ARP Table	Checks the PC's ARP-table for reachable IP addresses	Requires a populated ARP table on the local PC

Subnets

This element allows the user to specify a number of subnets to be scanned during the discovery process. These subnets will be scanned with ICMP ping in addition to any addresses discovered by interface-based methods. You may add any number of subnets to this list by clicking the + button, and remove them by clicking the trash button next to each subnet. Beneath the subnet list, you will see a count of the number of addresses that will be scanned during the discovery process.

Options



This tab contains a number of options that affect the behavior of the discovery task:

Option	Description
Discover Neighbours	When enabled, the discovery process will apply recursive discovery to all discovered devices. This allows for the discovery of devices that may not be directly reachable, and across routing boundaries.
Traceroute	When enabled, the discovery process will perform a traceroute to all discovered devices that do not have a known topological connection to the local PC. This allows for the discovery of intermediate routing devices.
Use DNS	When enabled, the discovery process will attempt to resolve the hostnames of discovered devices using DNS. This can provide more user-friendly names for devices, but may not always be successful depending on the network configuration.

> **Info**
>
> All of these options add significant time to the discovery process, especially when applied to large numbers of devices.

SNMP

Located beneath the Options group, this element allows the user to configure SNMP settings for the discovery process. These settings include:

- SNMP Port
- SNMP Timeout
- SNMP Version to be used (v2c or v3)
- SNMP Community String (for v2c) or User Credentials (for v3)

Multiple SNMP configurations can be created and selected for use during the discovery process.

Accounts

Beneath the SNMP settings lies the Accounts section, where the user can configure account credentials for the discovery process. These credentials will be used to authenticate with devices during discovery when the “Usage” field matches the type of account being requested.

The fields available for each account credential include:

- Username: The username for the account
- Password: The password for the account
- Usage: The type of usage this account is intended for (e.g., SSH, HTTP, HTTPS, REST)
- Public Key: Whether or not public-key authentication should be used for this account.

The following usage types are available for account credentials, at least one needs to be configured, and multiple can be configured as a comma-separated list:

Type	Explanation
SSH	Used for authenticating with devices over the SSH protocol
HTTP	Used for authenticating with devices over the HTTP protocol
HTTPS	Used for authenticating with devices over the HTTPS protocol

Type	Explanation
REST	Used for authenticating with devices over REST-based APIs

Warnings, Windows Only

The following actionable warnings may appear during the task, each with a remediation link that can be clicked:

Warning	Explanation
IPConfig is blocked	The PC's firewall does not allow IPConfig packets to be sent or received, which will prevent device discovery using this protocol
mDNS is blocked	The PC's firewall does not allow mDNS packets to be sent or received, which will prevent device discovery from using this protocol

> *i* Info
>
> All of the remediation links require escalated privileges to resolve, WeConfig will request these as necessary. `NET ADMIN` is sufficient to add link local addresses, but `Administrator` is required for any firewall editing.

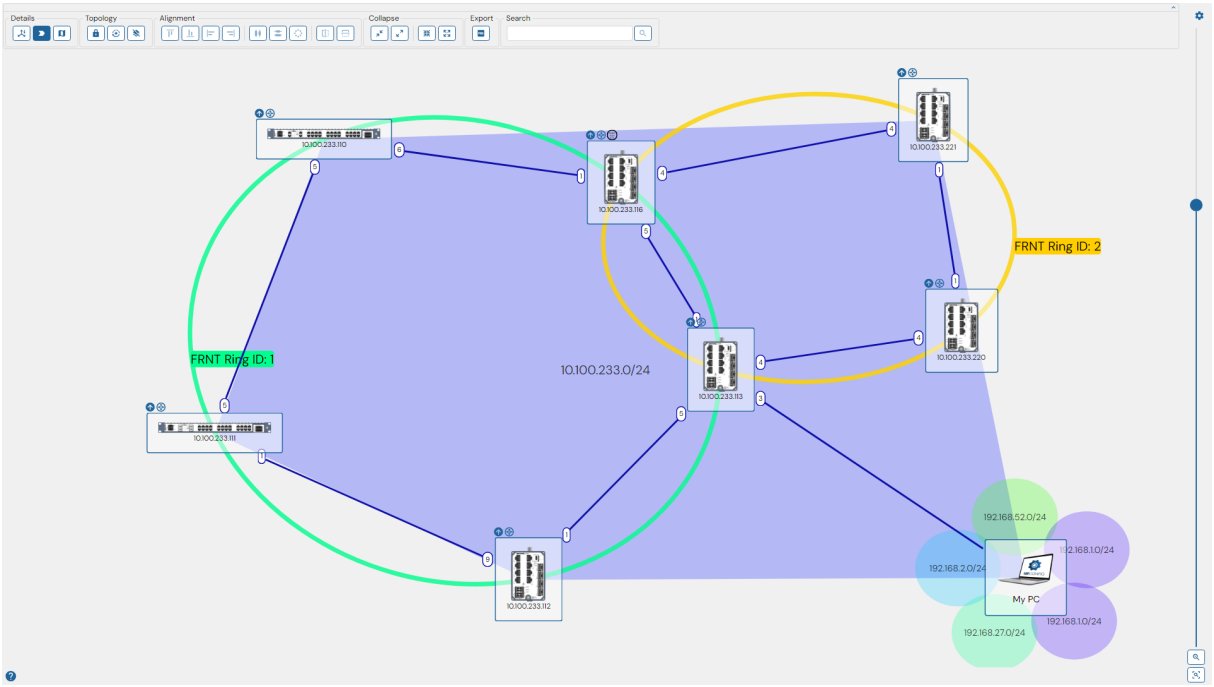
Log

This box contains a log window of the discovery process. The log will indicate which addresses respond to the different protocols used, the data about the devices as they are being discovered, as well as the current state of the discovery process.

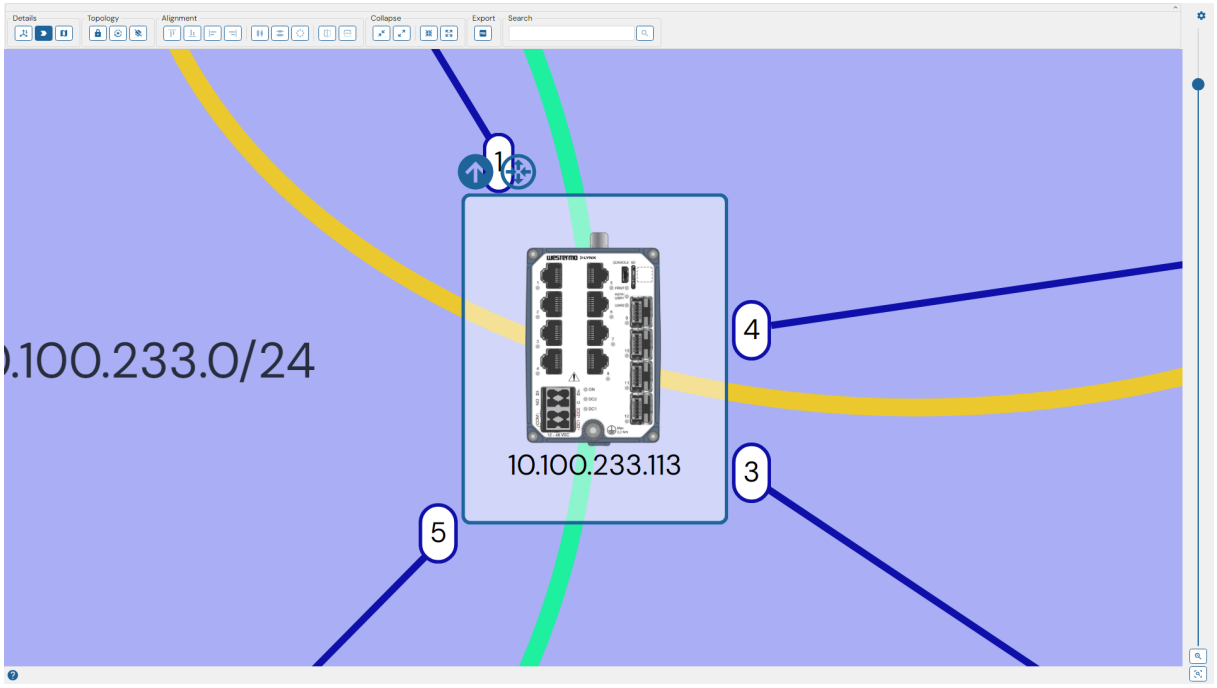
4.4.2. Physical Network

WeOS 4 WeOS 5 Ibex XRD Merlin other

The physical network panel is one of the main ways of visualizing the network in WeConfig, it will render the currently know network topology, including devices, connections between them, subnets and rings. Depicted below is an example topology with a single core subnet and 2 FRNT rings. Detected MRP rings are also displayed in the topology.



Wherein we can see the 7 different devices (plus the PC). These devices grouped together in the `10.100.233.0/24` subnet indicated by the semi-transparent blue polygon forming between them. We can also observe two rings, `Ring ID: 1` depicted in green and `Ring ID: 2` depicted in yellow. There are several more components to this topology. Let's zoom in on the device at IP Address 10.100.233.113:



Here we can observe a set of components.

Device border

Within the confines of the device border, there is a picture depicting the device, with its IP address listed beneath.

> **Tip**
>
> The contents displayed within the device border can be configured in the top right of the view

Node Icons

Above the device border, we see a list of icons, 2 in total in this example. These carry information and may serve as quick actions to undertake on the device. See the table below for what each icon means.

Icon description	Explanation	Clickable
A box with two dots & signal	Device can act as a router	No
Blue up arrow	There is an available firmware upgrade	Yes
Blue circle with an 'i'	There is at least one issue linked to this device	Yes
A yellow triangle with a '!' inside	Device has connectivity issues.	No
A red circle with a '!' inside	Device uses firmware that is not supported	No
Circle with "FRNT Focal Point"	Device is the focal point for a FRNT ring	No
Circle with "MRP Manager"	Device is an MRP manager	No

> **Tip**
>
> Hovering above each icon also gives additional information about its purpose.

Port labels

Only visible when toggled under the details section, these ellipses on the edge of the device border indicates the port associated with any connections going to and from this device. In the example above, we can see the the device has four connections, one through port 1, one through port 3, one through port 4 and one through port 5.

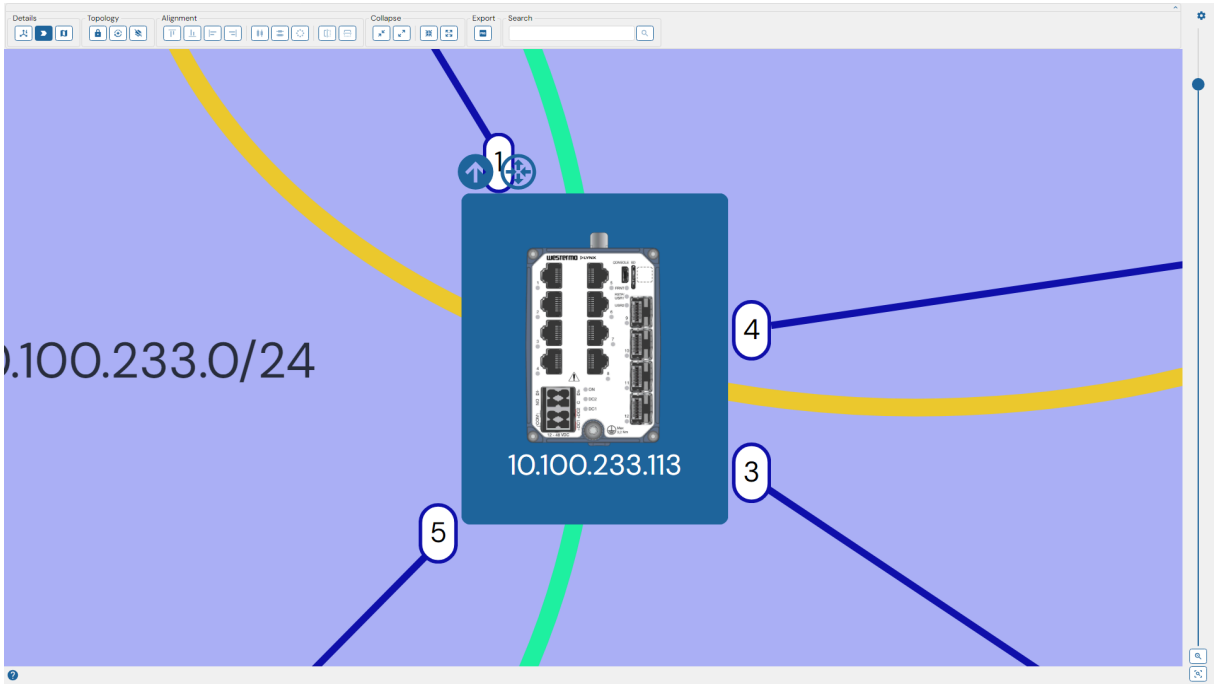
The edge color of these ellipses and their continuation in connection lines also serve to specify the physical medium across which the connection is carried, according to the following table:

Line	Medium
Straight, blue	Copper-based ethernet cable.
Straight, orange	Fiber-based ethernet cable.
Sinusodal, green	Copper-based DSL cable.
Dash-dot-dot, grey	Mixed media / aggregate link
Dotted, Purple	Wireless
Straight, White/Black	Unknown

> **Info**
>
> This legend is also available in the topology view itself if “Show connection information” is toggled under the details section

Selection

The physical network panel is one of the primary sources of selection in WeConfig, you may select a device by left-clicking on it. Any device that is selected will be highlighted, see image below:



Which depicted the previously mentioned device at `10.100.233.113` when it is selected.

By default, selecting a device deselects any previously selected devices. In order to append to the current selection instead of overriding it, hold down `Ctrl` while left clicking.

Subnet & Ring Selection

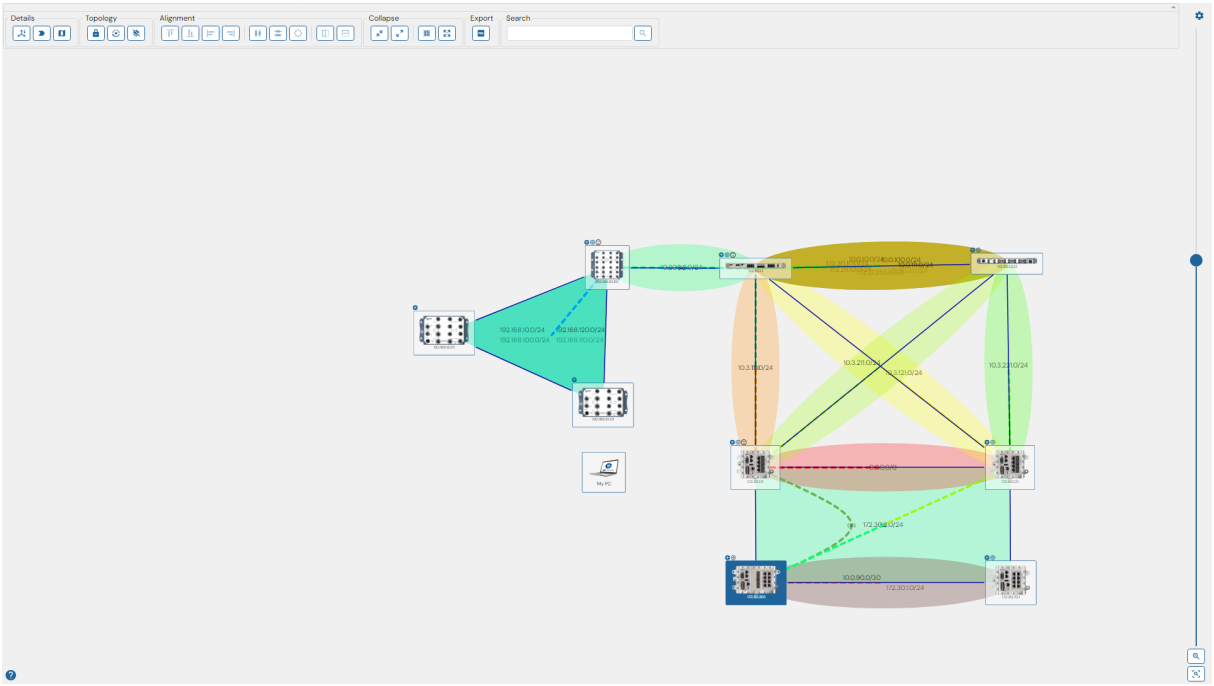
It is possible to select based on both subnets and rings present in the topology as well, to do so, simply click the ring or subnet you wish to select. Some specifications apply for subnet and ring selection:

- Selecting a ring or subnet selects all devices that belong to said ring or subnet.
- A ring or subnet is considered selected if, and only if, all of the devices that belong to said ring or device is selected.

> **Tip**
>
> Hovering a subnet or ring will highlight it slightly.

Route visualization

Additionally, whenever a device is selected, WeConfig will attempt to visualize its outgoing routes to any known subnets inside of the topology.



Where we can observe a set of curves going between, for example, the device `172.30.2.1`, with the color indicating the origin subnet and, jumping through relevant gateways, arriving at the target subnet.

Zoom Control

The zoom control is located to the right of the main topology view. It controls the zoom level within the physical network topology view. It consists of a slider indicating the current zoom level, as well as two buttons which indicate and alter the current zoom state, they are as follows:

Button	Purpose
Fill	Zoom the network topology to the minimum encompassing bounding box
1:1	Reset the zoom level to 1

> **Tip**
>
> The zoom level in the topology can also be controlled via the mouse-wheel.

Topology control section

At the very top of the physical network panel we will find a section containing a variety of options that allow us to further refine and modify the network topology and how we view it. These are divided into six section: Details, Topology, Alignment, Collapse, Export, and Search.

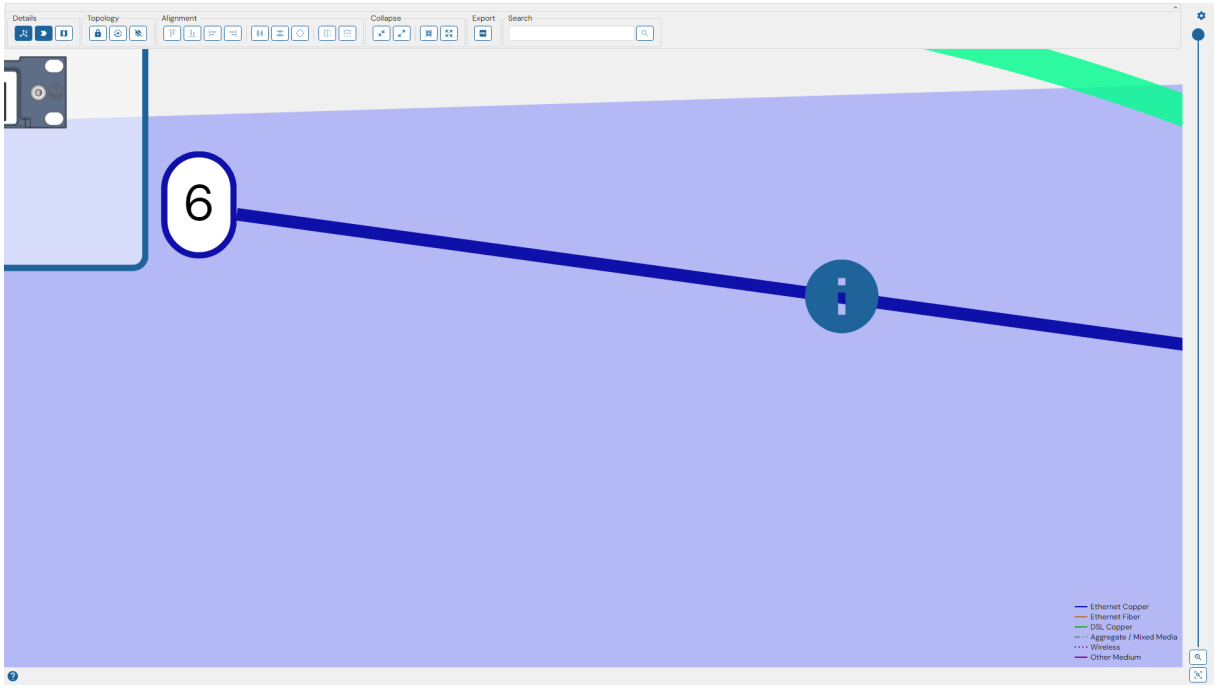
Details

The first section menu controls whether or not certain additional information is rendered, it consists of three toggles, which serve the following purpose:

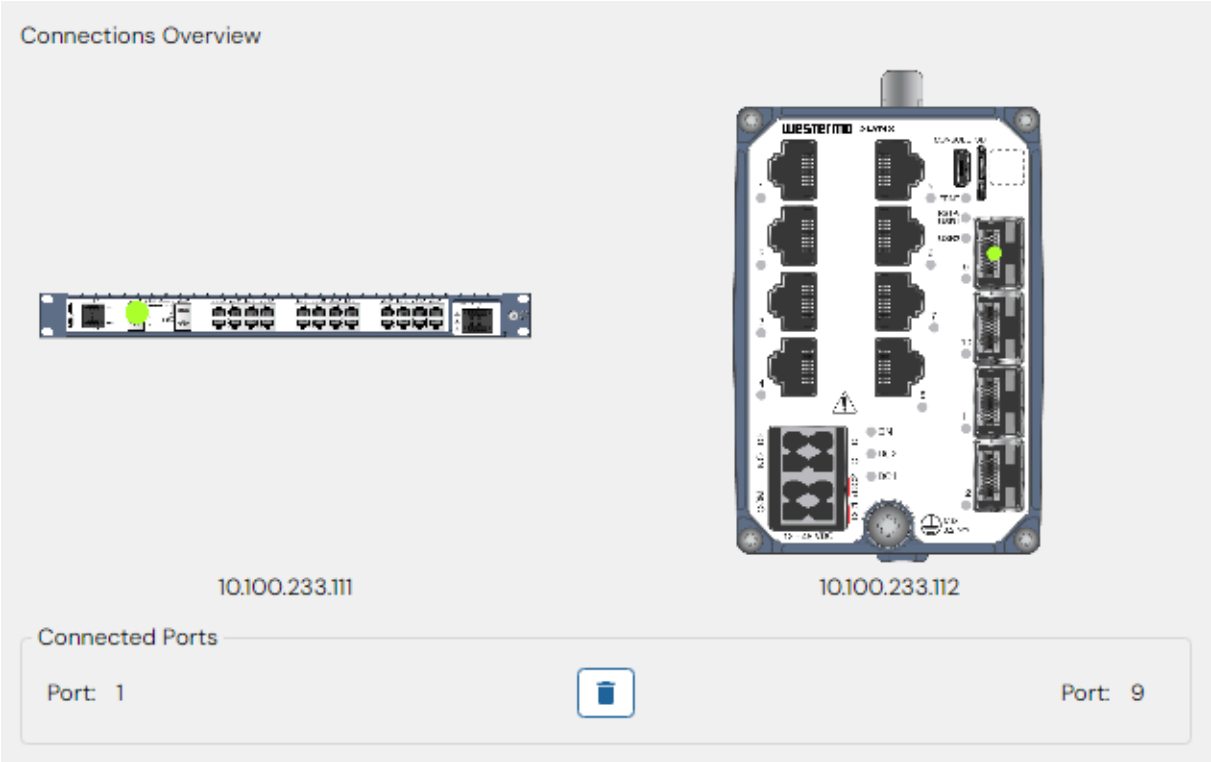
Toggle	Purpose
Show connection information	Controls the display of the connection information legend and the bottom right corner of the panel, as well as the clickable 'Connection Information' buttons that adorn the connections in the topology'
Show Port Labels	Controls the display of port labels adjacent to devices
Show/Hide Minimap	Controls the display of a network 'Minimap' at the top right corner of the panel

Connection information

When toggled in the details section, a button will appear on the centre of each connection in the topology, as depicted below:



Clicking this button opens up a connection information flyout, which contains additional information about the connection in question, see one example depicted below:



In this example, we can see that the device at `10.100.233.111` is connected to the device at `10.100.233.112` from port `1` to port `9`. The device images also indicate the physical location of these respective ports on the device itself with a green dot.

If for some reason, this connection is considered incorrect, the delete button positioned between the port indicators can be used to make WeConfig forget about this connection.

Topology

The second section menu controls the topology layout, it has three options, as depicted below:

Where the options are, from left to right:

Option	Explanation
Lock/Unlock Layout	When toggled, prevents any changes to the topology layout.
Auto Layout	Attempt to organize the topology according to the currently set layout algorithm
Clear Topology	Removes all devices from the topology.

Alignment

The third section menu controls the alignment of devices within the topology, it has several options, as depicted below:

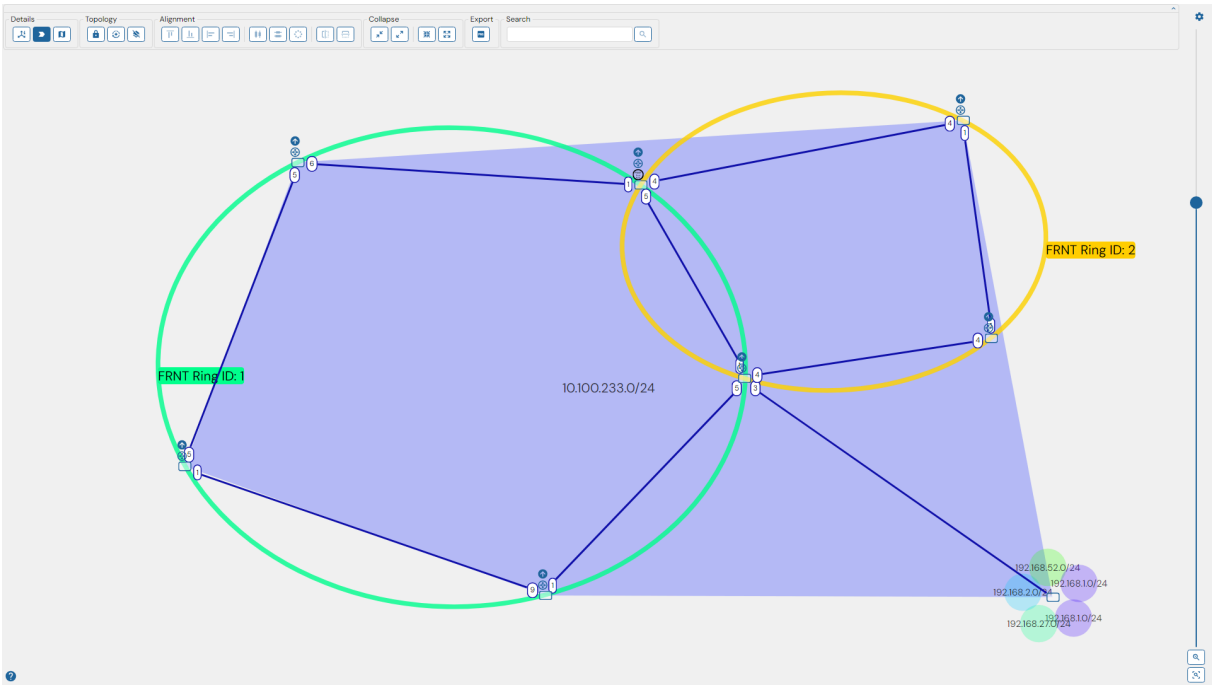
Where the options are, from left to right:

Option	Explanation
Align Top	Move the Y-coordinate of all selected devices to the Y-coordinate of the topmost selected device.

Option	Explanation
Align Bottom	Move the Y-coordinate of all selected devices to the Y-coordinate of the bottommost selected device.
Align Left	Move the X-coordinate of all selected devices to the X-coordinate of the leftmost selected device.
Align Right	Move the X-coordinate of all selected devices to the X-coordinate of the rightmost selected device.
Distribute Horizontally	Distribute the X-coordinate of the selected devices evenly on a scale between the leftmost and rightmost selected device
Distribute vertically	Distribute the Y-coordinate of the selected devices evenly on a scale between the topmost and bottommost device
Distribute Radially	Distribute the coordinates of the selected devices evenly in a circle around their average center
Flip Horizontally	Invert the X-coordinates of the selected devices around their average X-coordinate
Flip Vertically	Invert the Y-coordinates of he selected devices around their average Y-coordinate

Collapse

The fourth section menu contains controls for collapsing and expanding devices. With the first two buttons collapsing and expanding so called “Unknown” devices, or in other words those that WeConfig does not know how to work with, and the last two collapsing and expanding all devices. For example, the original topology depicted at the top of this document would appear like this when all devices are collapsed:



> 💡 Tip
>
> Devices can also be individually expanded or collapsed via the context menu

This allows for a variety of views of the network, for example, if one sets connection opacity to 0 and collapses all devices, one would have as close to a pure “layer 3” view as WeConfig offers today.

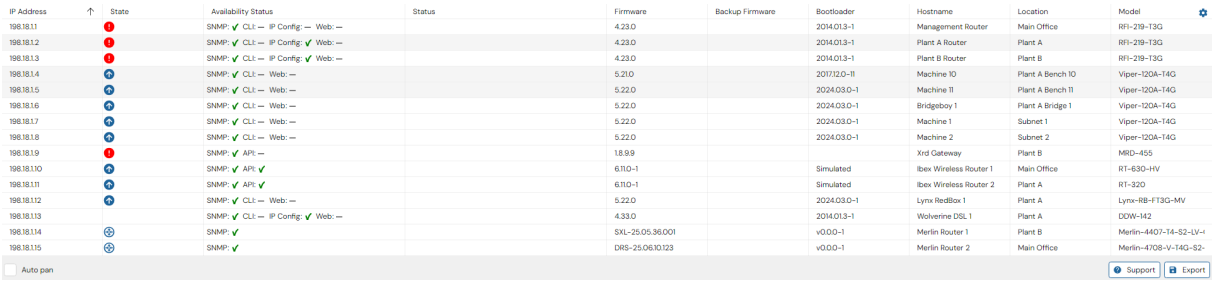
Export
The fifth section menu is used to export the current topology as a PNG image.

Search
The sixth and final section menu contains a search bar, which allows for searching for devices in the topology.

4.4.3. Devices

WeOS 4 WeOS 5 Ibex XRD Merlin other

The devices panel is a secondary way of visualizing the network in WeConfig, where devices are ordered in a grid-list with some properties on display. Depicted below is an example of a populate device list including WeOS, Ibex, Xrd and Merlin Devices.



Selection in this panel will be mirrored in the topology.

There are several details of note within this panel:

Columns

Headers

Some headers can be clicked, and the default behavior for doing so is to order the grid-list by the value of the columns associated with the header, if clicking the header does not sort the list, it indicates that the column does not contain sortable values. Additionally, hovering over a header may reveal a filter button, which can be used to filter the list according to desired properties.

IP Address

This column presents the primary / management IP address associated with the device.

Netmask

This column presents the subnet mask associated with the device's primary / management IP address.

MAC Address

This column presents the base MAC address associated with the device.

Serial Number

This column presents the device's serial number, when available.

State

This column is a mirror of the icons found above the devices in the topology panel

Availability status

This column presents a collection of status indicators:

Indicator	Meaning
Check mark	Good
Cross	Bad /Failed

Indicator	Meaning
-	Unknown

The following entries may appear:

Entry	Explanation	Applicable to
Web	The status of the latest attempt at communicating with the device via HTTP(S)	Any
SNMP	The status of the latest attempt at accessing the device via SNMP	Any
Ping	The status of the latest ICMP ping sent towards the associate management IP address	Any
CLI	The status of the latest attempt at accessing the device via SSH	WeOS
IP Config	The status of the latest attempt at communicating with the device via IP-config	WeOS 4
API	The status of the latest attempt at accessing the device via it's specific API	XRD, IBEX

Status
This column presents any current status messages associated with the device, and may indicate if a specific device has encountered an error.

Configuration hash
This column presents the hash of the current configuration detected on the device, if any.

Firmware
This column presents the current primary firmware version detected on the device, if any.

Backup Firmware
This column presents the current backup firmware version detected on the device, if any.

Bootloader
This column presents the current bootloader version detected on the device, if any.

FRNT
This column presents the current FRNT status of the device, if any.

Hostname
This column presents the known configured hostname, if any, associated with the device.

Location
This column presents the known configured location, if any, associated with the device.

Model
This column presents the identified specific model, if any, associated with the device.

Backup
This column presents the last backup time of the device, if any.

Family
This column presents the family to which the device belongs, if any.

Auto pan
Located at the bottom left of the panel, when checked, selection originating within this panel will send a request to the topology panel, if active, to move its focus to the selected devices.

Export
Located at the bottom right of the panel. Clicking this button will open up a dialog to export the grid-list into a .csv format file.

Support
Located to the left of the Export button. Clicking this button will download the tech support files from the selected devices.

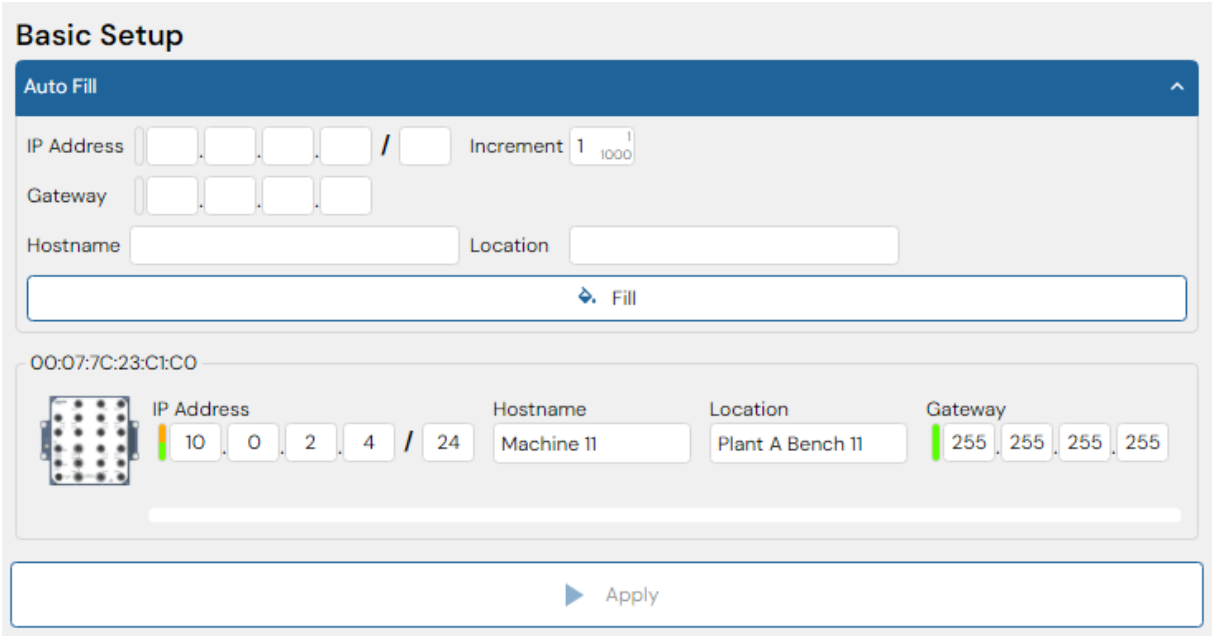
4.4.4. Basic Setup

WeOS 4 WeOS 5 Ibex XRD Merlin

Basic setup is a task that allows for configuration of the following basic device properties:

Property	Explanation	Required
IP Address	The primary IP address of the device, in CIDR notation	Yes
Host name	The host name tag of the device	No, will default to previous value if empty.
Location	The location tag of the device	No, will default to previous value if empty
Default gateway	The default gateway for routes	No

Interface Components



Fill pane
The fill pane allows the user to write properties once and then copy them to all devices. The increment box will operate on the IP Address field and increase the IP address handed to each device by the amount specified.

Any property left empty will be ignored when the “Fill” button is pressed.

Configuration list
Beneath the fill pane the list of individual device configurations can be found. This can be edited to set device-specific properties.

4.4.5. Accounts

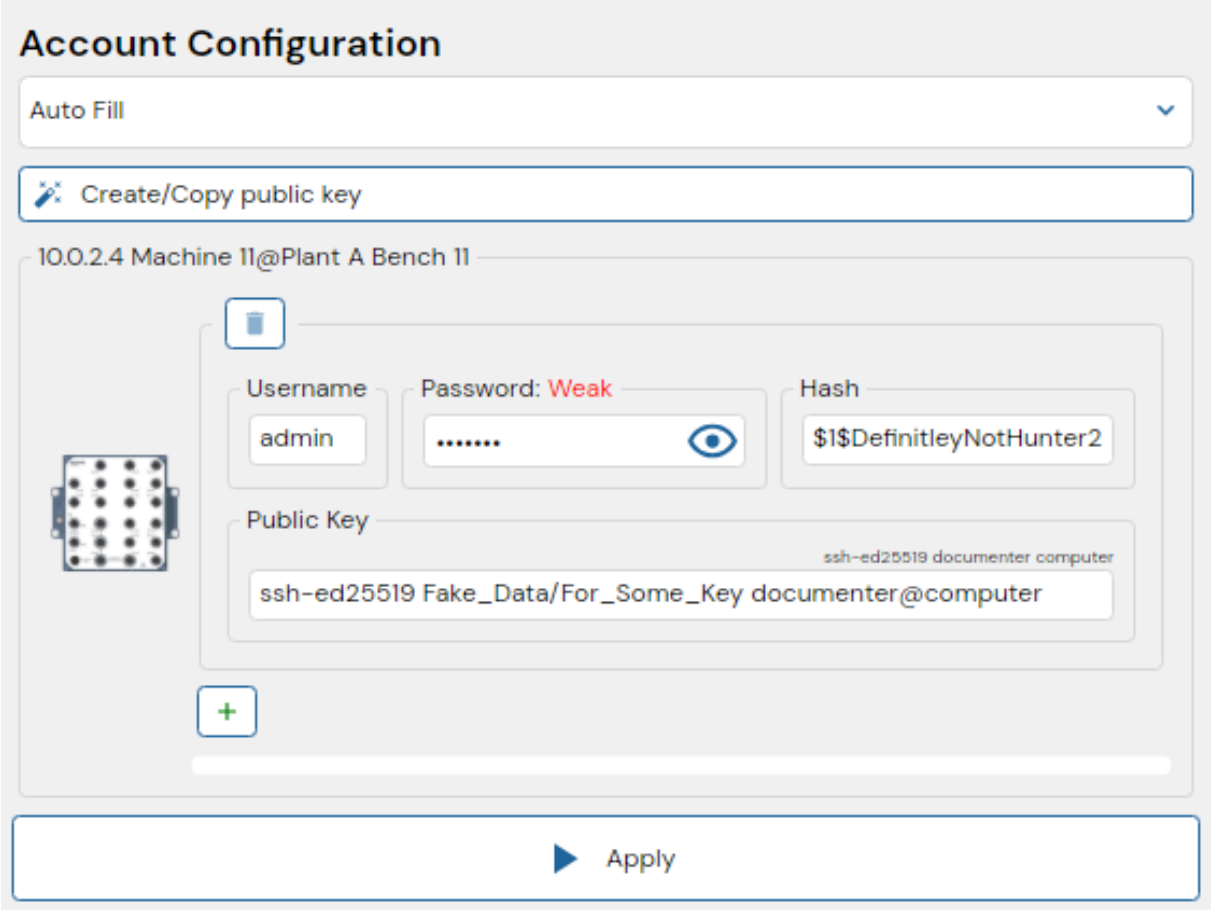
WeOS 4 WeOS 5 Ibex XRD Merlin

Accounts is a task that allows for configuration of user accounts on supported devices. The exact properties that can be configured varies between firmwares, as per the table below:

Firmware	Password	Username	Raw Hash	Public Keys
WeOS 4	Yes	Yes	Yes	Only for Admin
WeOS 5	Yes	Yes	Yes	Yes
Ibex	Yes	No	No	No
XRD	Yes	No	No	No

> **Note**
>
> Configuring the account setup for an XRD also sets up the API account associated with the device.

Interface Components



Autofill section

Collapsed by default under the label “Autofill”, this section of the interface allows for filling out a group of account that can then be copied to all selected devices.

Create/Copy public key
Located towards the top of the interface, this button is an easy short-hand for generating a public/private RSA key-pair associated with the running machine. When clicked, it will check whether or not an RSA key pair exists on the machine, and if not, create one. The public key of the RSA pair will then be copied to the users clipboard.

Device view
Picture above is the view for a singular device at the IP address `10.0.2.4`. This is a WeOS 5 device, and as such all the components are visible. They are as follows:

Username
This box specifies the username to associate with the account, it may be read-only if the device firmware does not support non-default usernames.

Password
This box specifies the password to associate with the account.

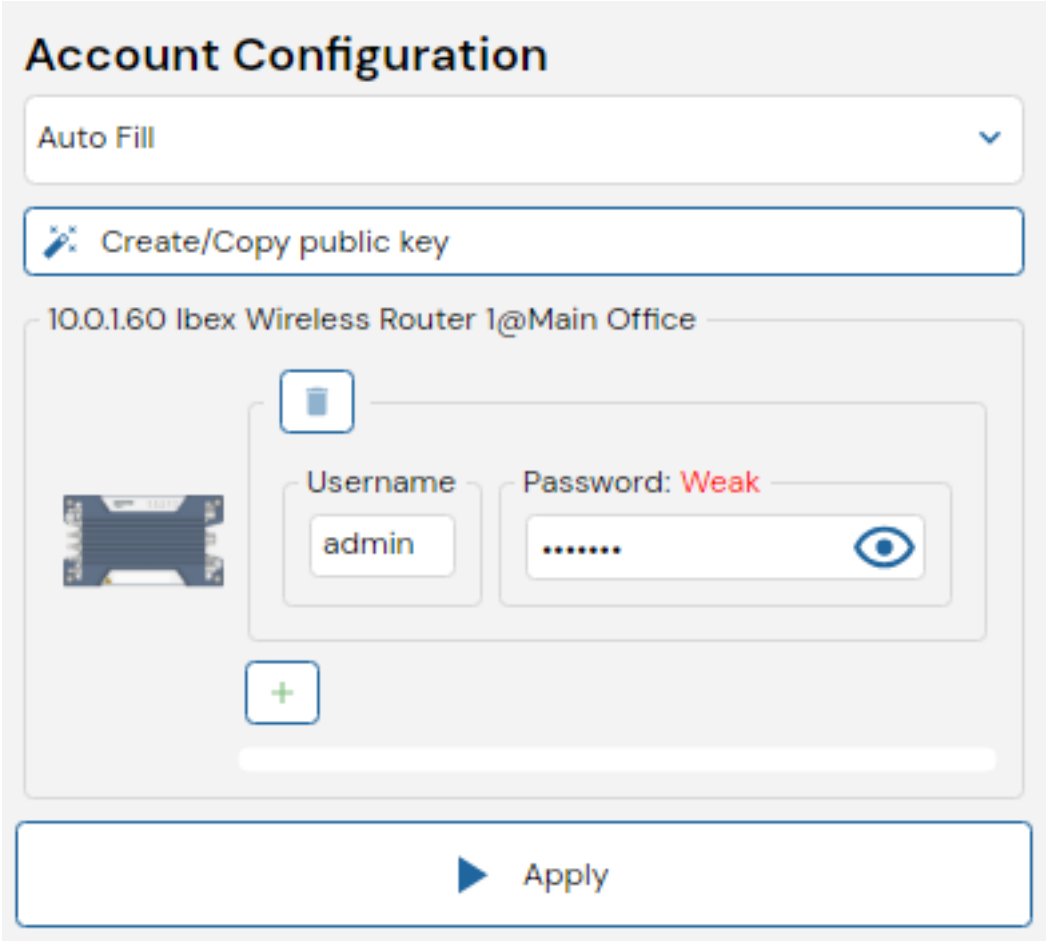
Hash
This box specifies the hashed version of the password, useful for copying authentication between devices without explicit knowledge of passwords.

> **Note**
>
> Any hash specified will be ignored in favor of the Password box if it is non-empty.

Public keys
This box specifies the public keys that can be used for authentication purposes. An account can usually have more than one public key associated with it. Filling out one line of the public key box will make another line appear beneath it, as long as there is space on the device for more public keys for the account.

> **Note**
>
> If the password box and hash box is left empty on an account that supports and is configured with public key authentication, the only way to access the account will be via public key authentication.

Reduced view
In comparison to the above full view for a WeOS 4 device, below is pictured the view for an Ibex device, which has less supported features in this view:



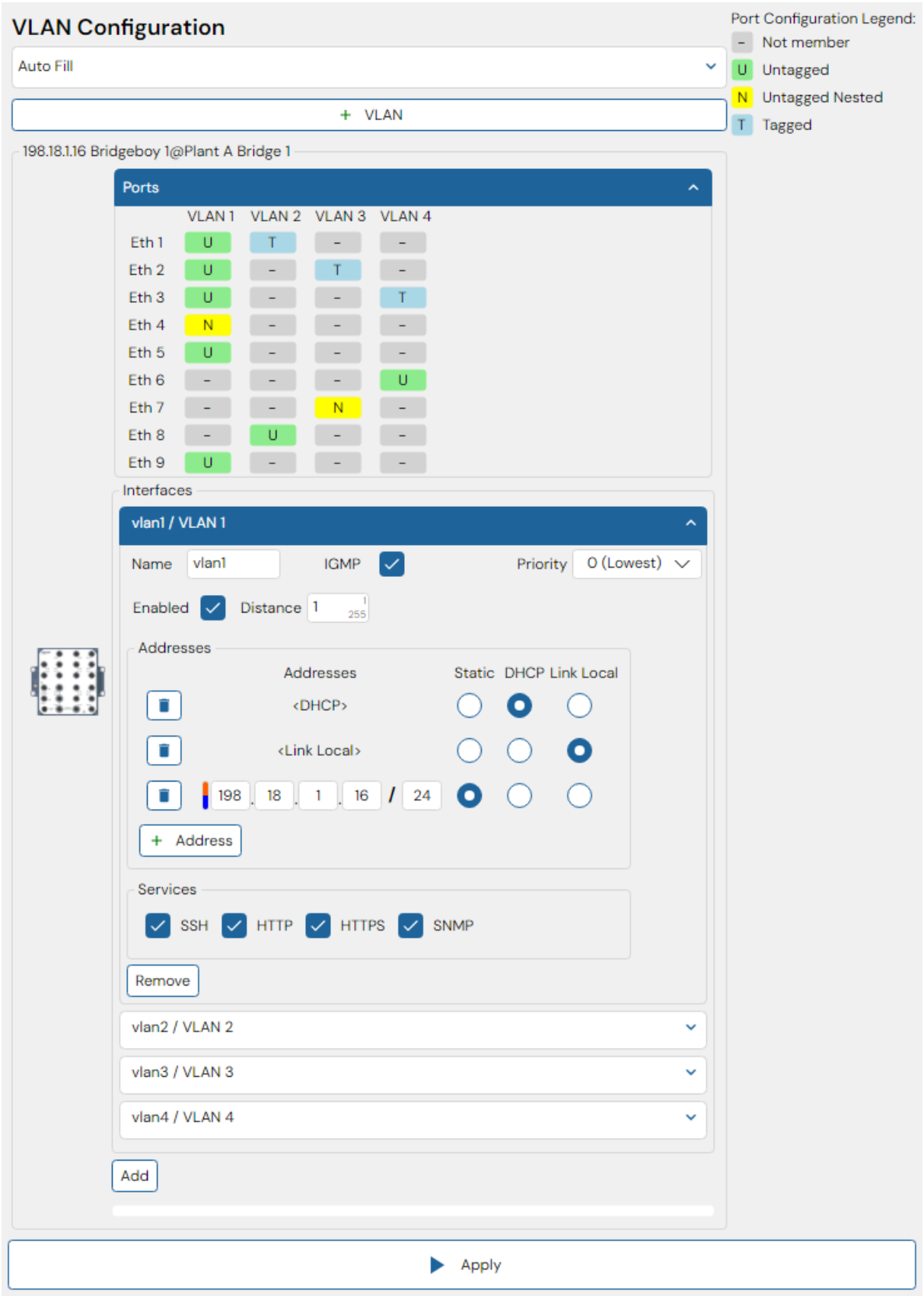
As can be seen, only the username and password box is visible, the username cannot be edited and no more accounts can be created.

4.4.6. Virtual Local Area Networks (VLAN)

WeOS 4 WeOS 5 Merlin

VLAN is a task that allows for configuration of virtual local area networks, VLANs for short, and their port associations.

Interface components



Legend
At the top right of the interface is a legend, depicting a shorthand assistance for reading the port association matrix.

Autofill
Just below the filter box exists an autofill section, which is collapsed by default. This allows for configuring a VLAN (without port association) once and copying that configuration to all selected devices.

Device view
The rest of the interface will contain a list of devices views, with a singular device in the example depicted above. Each device view is broken down into two sections, the port association matrix and the VLAN configuration list.

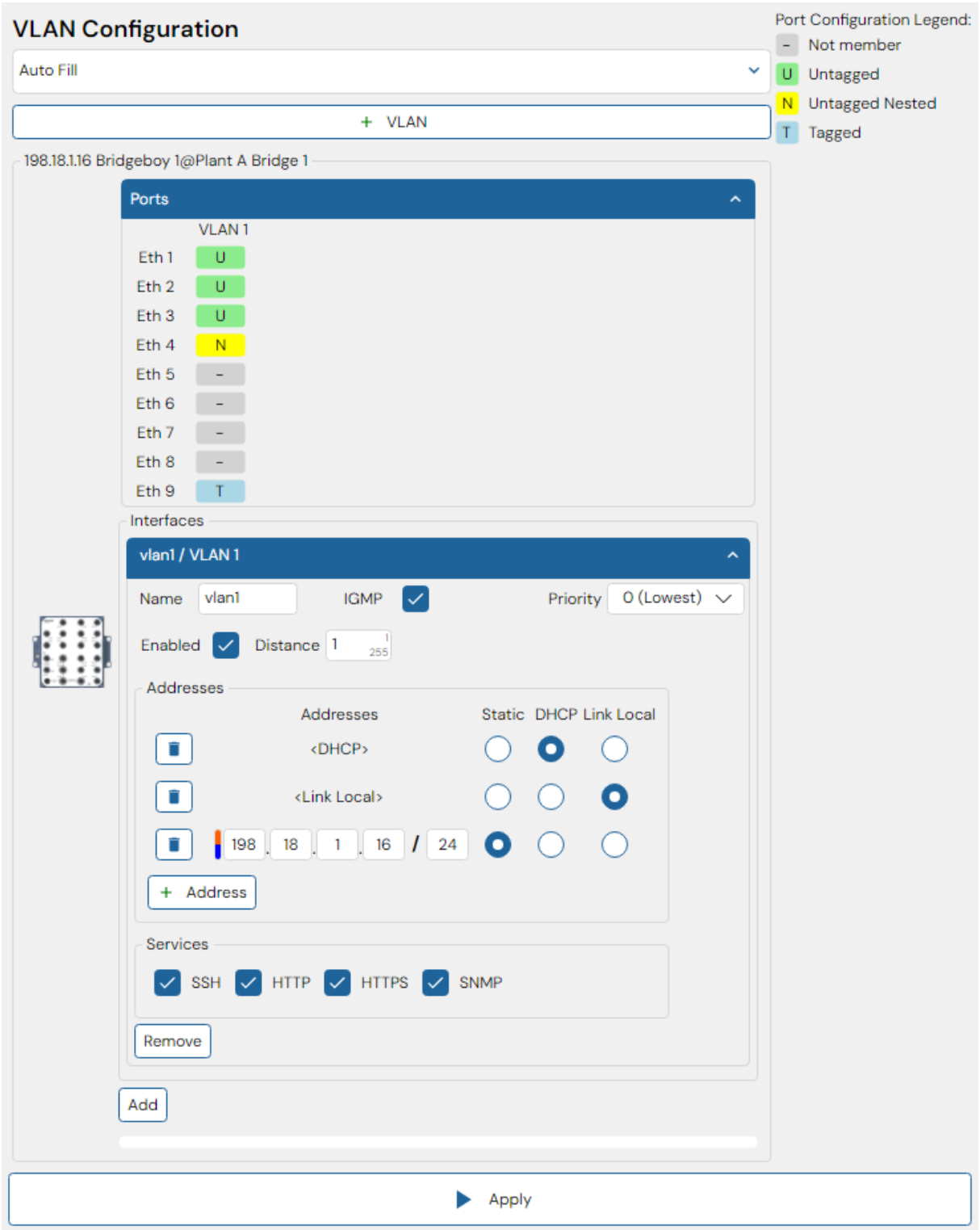
Port assocation matrix
The top of the device view is a collapsible port matrix, where the device ports are listed as rows and the existing VLANs are listed as columns. This allows specification of which VLANs are mapped to which port, and in which way they are mapped. The above example demonstrates the following setup:

- For VLAN 1, Ports Eth 1 through Eth 5 and Eth 9 are present but untagged. Port 4 has policy nesting enabled.
- For VLAN 2, Port Eth 1 is tagged (which does not conflict with untagged VLAN 1 traffic on the same port) and Port Eth 8 is untagged
- For VLAN 3, Port Eth 2 is tagged, and Port Eth 7 is untagged with policy nesting enabled.
- For VLAN 4, Port Eth 3 is tagged, and Port Eth 6 is untagged.

In order to change the port association, simply toggle the associated boxes

> *i* Info
>
> A port can only be Untagged on one VLAN

Interfaces
Under “Interfaces” you can configure the individual VLANs. What you can configure varies slightly depending on the WeOS version.


VLAN Configuration
Auto Fill
+ VLAN
198.18.1.16 Bridgeboy 1@Plant A Bridge 1
Ports
VLAN 1
Eth 1 U
Eth 2 U
Eth 3 U
Eth 4 N
Eth 5 -
Eth 6 -
Eth 7 -
Eth 8 -
Eth 9 T
Interfaces
vlan1 / VLAN 1
Name vlan1
IGMP
Priority 0 (Lowest)
Enabled
Distance 1
Addresses
Addresses
Static DHCP Link Local
<DHCP>
<Link Local>
198 . 18 . 1 . 16 / 24
+ Address
Services
SSH HTTP HTTPS SNMP
Remove
Add
Port Configuration Legend:
- Not member
U Untagged
N Untagged Nested
T Tagged
Apply

VLAN Configuration

Port Configuration Legend:
- – Not member
- U Untagged
- N Untagged Nested
- T Tagged

Auto Fill

+ VLAN

198.18.1.16 Plant A Router@Plant A

Ports

	VLAN 1
Eth 1/1	U
Eth 1/2	U
Eth 1/3	T
Eth 2/1	–
Eth 2/2	–
Eth 2/3	–
Eth 2/4	–
Eth 2/5	–
Eth 2/6	–
Eth 2/7	–
Eth 2/8	–
Eth 3/1	–
Eth 3/2	–
Eth 3/3	–
Eth 3/4	–
Eth 3/5	–
Eth 3/6	–
Eth 3/7	–
Eth 3/8	–



Interfaces

vlan1 / VLAN 1

Name vlan1 IGMP

Enabled Distance 1 (1–255)

Addresses

	Addresses	Static	DHCP	Primary
	<DHCP>	○	●	☐
	198.18.1.16 / 24	●	○	☑

+ Address

Services

SSH HTTP HTTPS IPConfig SNMP

Remove

Add

Apply

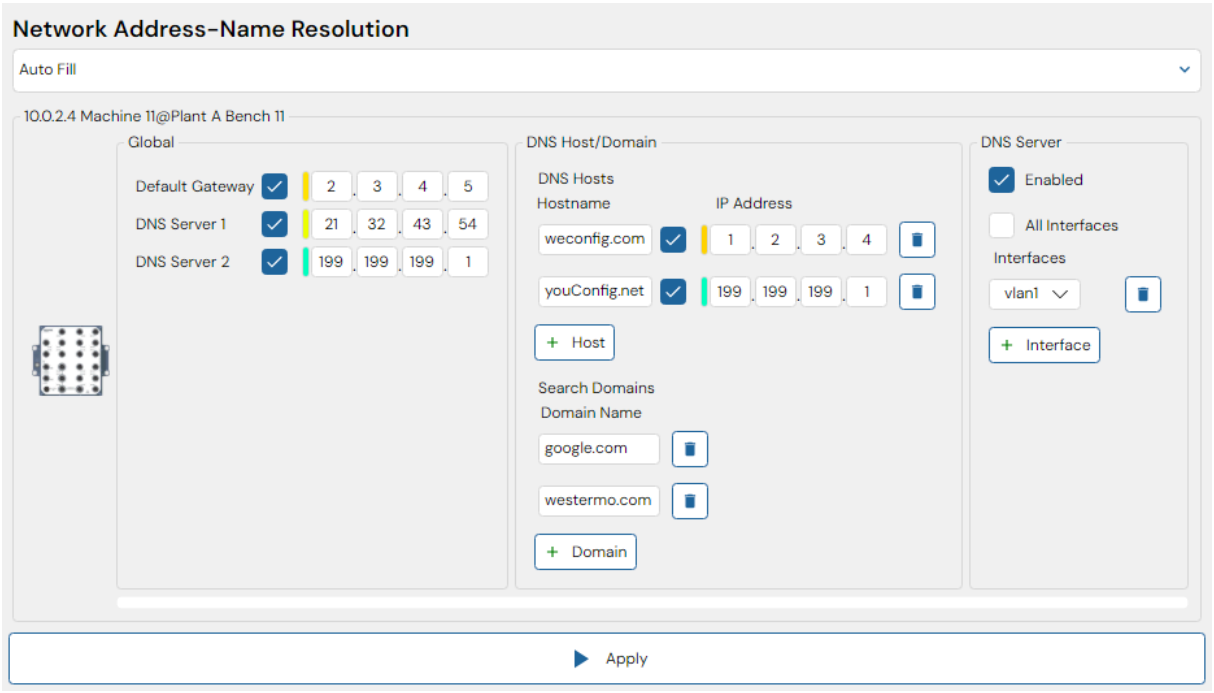
> **Info**
>
> On WeOS4, you must provide one (and only one) Primary address

4.4.7. Network Address-Name Resolution

WeOS 4 WeOS 5

Network Address-Name Resolution is a task that allows for configuration of default gateway, routing and DNS servers

Interface components



Autofill

This collapsed menu contains a group of options for quickly replicating configuration across all selected devices. Default gateway, routing and DNS server 1 / 2 can be set therein and then filled to all selected devices.

Device view

Below the autofill section a list of selected devices will appear, with their configuration options presented in three groups. The first group “Global” allows individual device configuration of the same properties as the Autofill section.

The second group “DNS Host/Domain” allows for device specific configuration of DNS Hosts and search domains to pass DNS request forwards to. Each device may have multiple of both options.

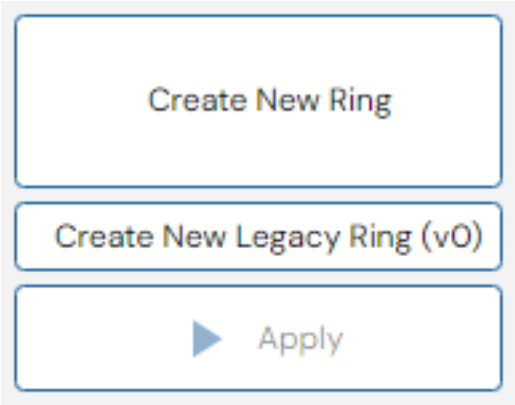
The third and final group “DNS Server” allows the user to configure the device to act as a DNS server, and to specify which interfaces that will be part of said server.

4.4.8. Fast Reconfiguring Network Topology (FRNT)

WeOS 4 WeOS 5

FRNT is a task that allows for configuration of FRNT rings in the network.

Interface Components



To create a new FRNT ring, click either Create New Ring or Create New Legacy Ring (v0). To edit an existing FRNT ring, select the ring in the drop down menu, and click Edit Ring.

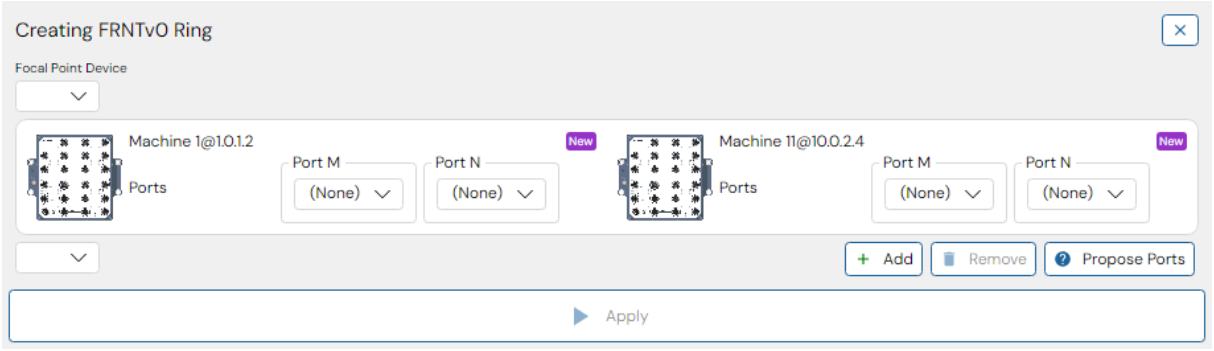
> **Note**
>
> RiCo (Ring Coupling) is not configured here. Please see Ring Coupling for documentation

When creating new rings, or editing existing ones:

- To add devices to a ring: select devices in the topology and click Add
- To remove devices from a ring: select devices in the list of devices in the right, and click Remove
- One device must be designated the role of focal point in the drop down menu Focal Point Device
- To apply changes: click Apply
- To propose ports based on current topology: click Propose Ports
- To manually specify ports: select ports per device
- To abort operation: click the button in the top right corner with a cross - that will take you back to the initial FRNT configuration tab.

WeConfig can also edit partially discovered FRNT rings. Select the detected ring and review the devices and ports before applying changes.

FRNT v0



Focal point device:

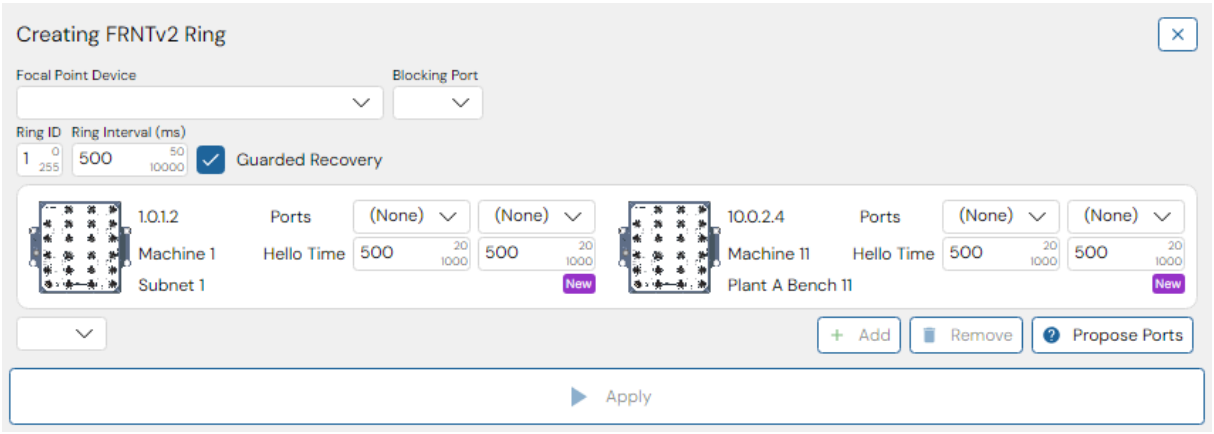
- No focal point options are applicable

Member devices:

- M port
- N port

Typically, all devices must have an M and and N port specified. The exception is so called “horse shoe” rings, where two end devices are allowed to have only one port configured.

FRNT v2



Focal point device:

- Ring ID must be specified, and it must be unique across all devices. WeConfig will emit an error if it detects that it is not unique. This field is not editable, it is only editable during creation. Once set, it cannot be changed. To change the ID, one must first delete the ring configuration across the entire device, and then create it again, but with the new ID.
- When editing a ring, the focal point device can be changed.
- Optionally specify the blocking port. Note that this is only possible once you’ve set the focal point’s ring ports. Until then, only Auto will be the only available option.
- Optionally change the ring interval
- Optionally enable guarded recovery

Member devices:

- First port
  - Optionally override the default hello time
- Second port
  - Optionally override the default hello time

Typically, all devices must have both ports specified. The exception is for so called “horse shoe” rings, where two end devices are allowed to have only one port configured

4.4.9. High-available Seamless Redundancy / Parallel Redundancy Protocol (HSR/PRP)

WeOS 5

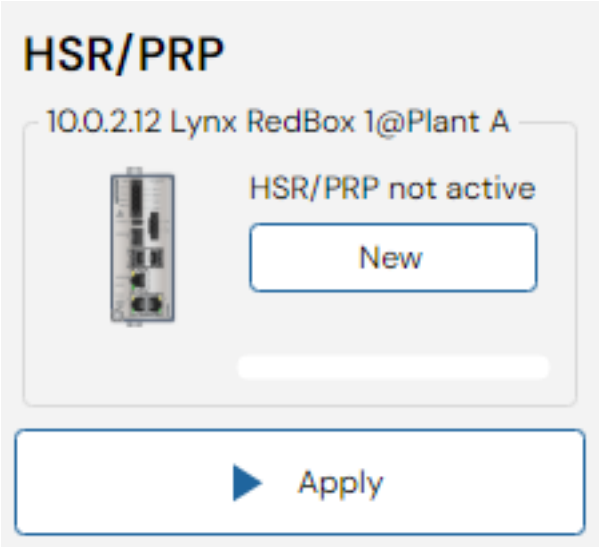
HSR/PRP is a task that allows for configuration of HSR/PRP redundancy in the network.

High-Availability Seamless Redundancy (HSR) and Parallel Redundancy Protocol (PRP) provides seamless failover on link failure. Both build on the same idea of duplicating traffic and can be coupled with each other.

Read more about HSR/PRP in the WeOS documentation[1].

> **Info**
>
> HSR/PRP is only available for RedBox

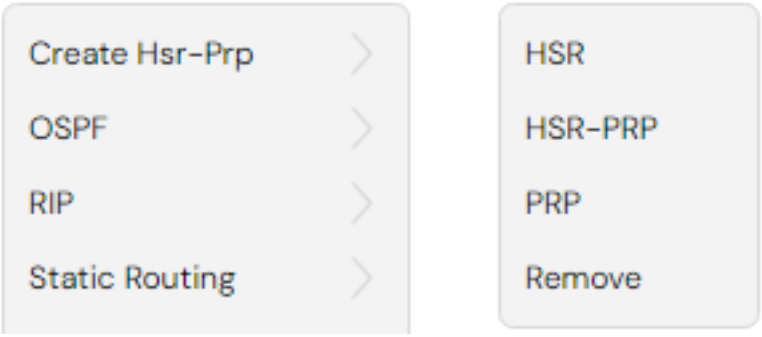
New



To create a new HSR/PRP ring, click the New button.

[1] https://docs.westermo.com/weos/weos-5/General/HSR_PRP.html

Context



You can also setup HSR/PRP from the context menu.

This will create a default HSR/PRP instance with the following properties:

> ⚠ Warning
>
> This will override any existing HSR/PRP configuration on the device.

HSR

Field	Value
Enabled	True
Type	HSR
Ring ID	1
Mode	H
Supervision Frames	True

PRP

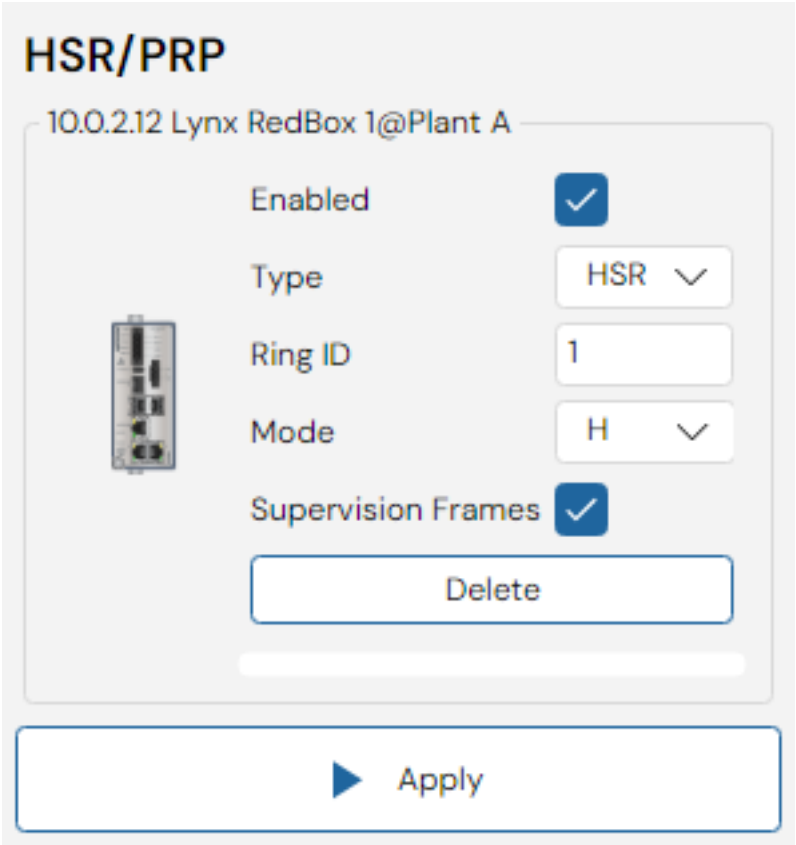
Field	Value
Enabled	True
Type	PRP
Ring ID	1
Supervision Frames	True

HSR-PRP

Field	Value
Enabled	True

Field	Value
Type	PRP
Ring ID	1
Mode	H
Supervision Frames	True
Pairing PRP Id	1
Pairing LAN Id	A

HSR



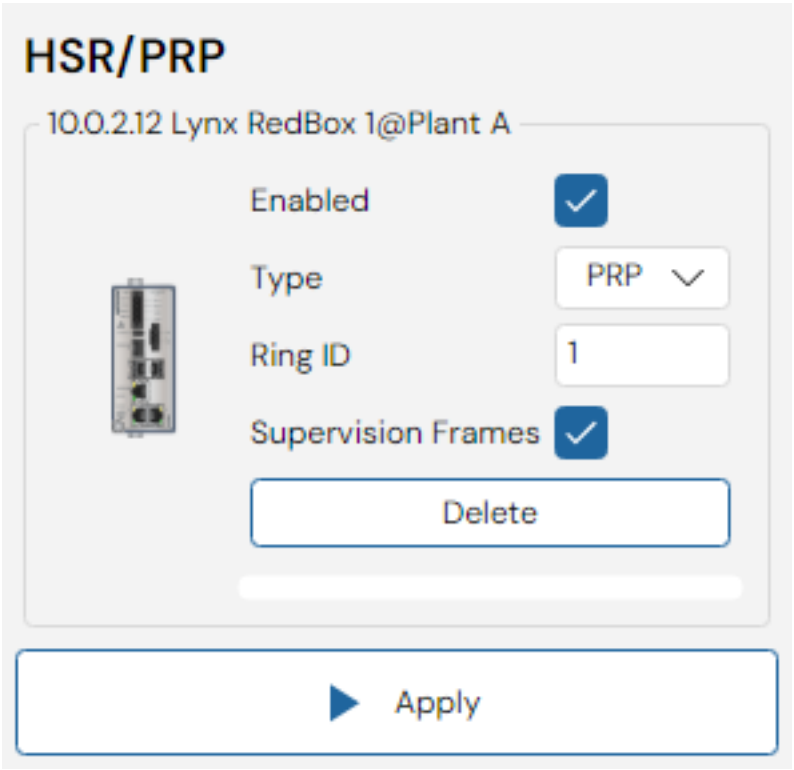
Field	Description
Enabled	Enable or disable this HSR instance
Type	Change the type of this instance, this will also change what fields are available
Ring ID	Id of this HSR instance
Mode	The operating mode for HSR. Can be either `H` or `N`
Supervision Frames	Enable/Disable supervision frames

Note

Modes:

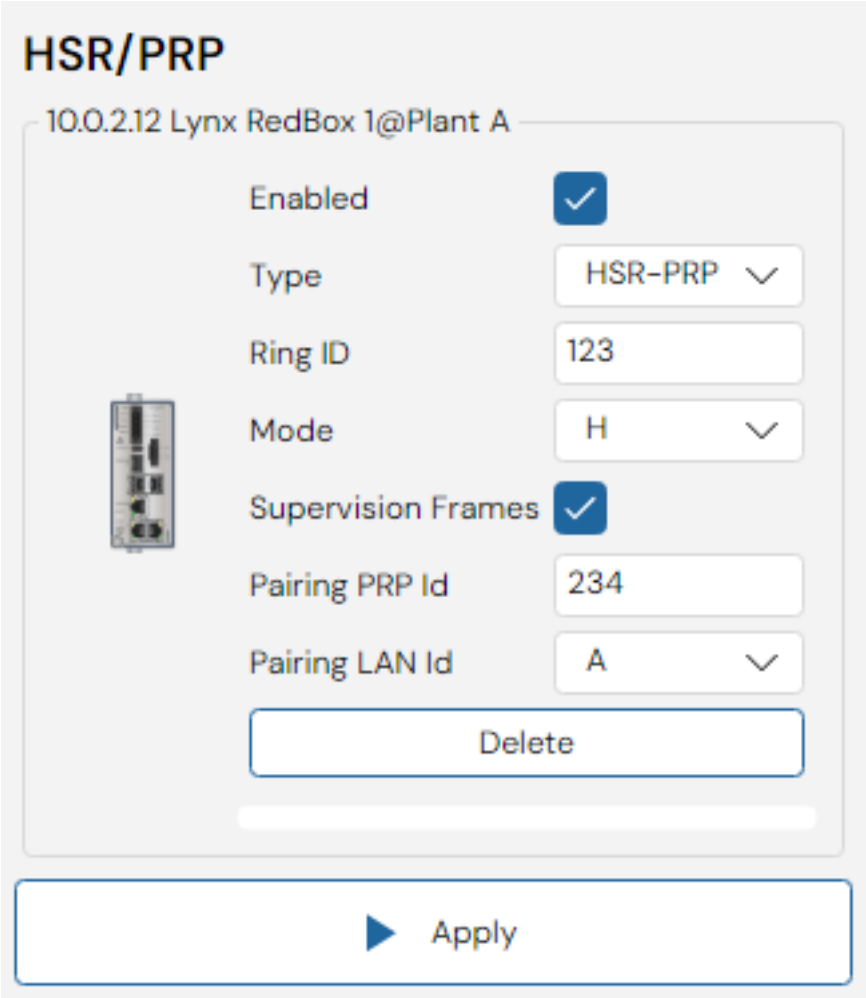
- H: HSR-tagged mode, the unit inserts HSR tag and forwards traffic.
- N: No forwarding operational mode, just like mode H, except that the unit does not forward traffic from port to port. Intended for testing purposes only.

PRP



Field	Description
Enabled	Enable or disable this PRP instance
Type	Change the type of this instance, this will also change what fields are available
Ring ID	Id of this PRP instance
Supervision Frames	Enable/Disable supervision frames

HSR/PRP



Field	Description
Enabled	Enable or disable this HSR/PRP instance
Type	Change the type of this instance, this will also change what fields are available
Ring ID	Id of this HSR/PRP instance
Mode	The operating mode for HSR. Can be either `H` or `N`
Supervision Frames	Enable/Disable supervision frames
Pairing PRP Id	NetId of the PRP network attached to the interlink.
Pairing LAN Id	LanId of the PRP network attached to the interlink.

Note

Modes:

- H: HSR-tagged mode, the unit inserts HSR tag and forwards traffic.
- N: No forwarding operational mode, just like mode H, except that the unit does not forward traffic from port to port. Intended for testing purposes only.

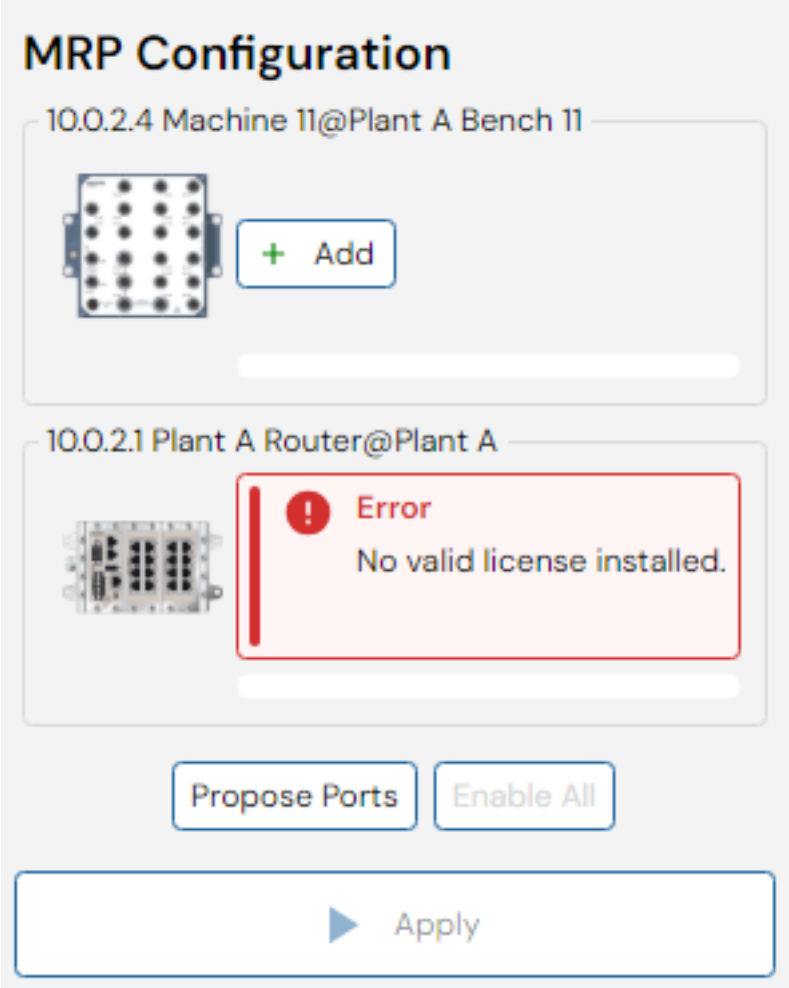
> **Info**
>
> RedBoxes connected to the same PRP network should have the same Pairing PRP Id (NetId)

4.4.10. Media Redundancy Protocol (MRP)

WeOS 4

MRP is a task that allows for configuration of Media Redundancy Protocol[1] rings in the network.

Interface Components



For each selected device, if the device supports / is licensed for MRP configuration, you will be presented with the following options:

Option	Description
ID	MRP Ring ID
Enabled	Whether or not this MRP ring is active on this device
Manager	Whether or not this device is the manager for this MRP ring
Port M	Which port on the device is considered the first element in the port-pair for this MRP ring
Port N	Which port on the device is considered the first element in the port-pair for this MRP ring
Profile Mode	The maximum recovery time profile for this MRP ring

[1] https://en.wikipedia.org/wiki/Media_Redundancy_Protocol

Option	Description
VLAN ID	Which VLAN to encapsulate the MRP rings signaling on, if any
React to link changes	Whether or not the MRP manager should react to link change frames
Open Ring detection	Whether or not the MRP manager should be more careful in detecting open rings

> **Info**
>
> For devices running WeOS 5.11 and newer, up to two ring configurations is supported. Although at most one may configure the device as an MRP client.

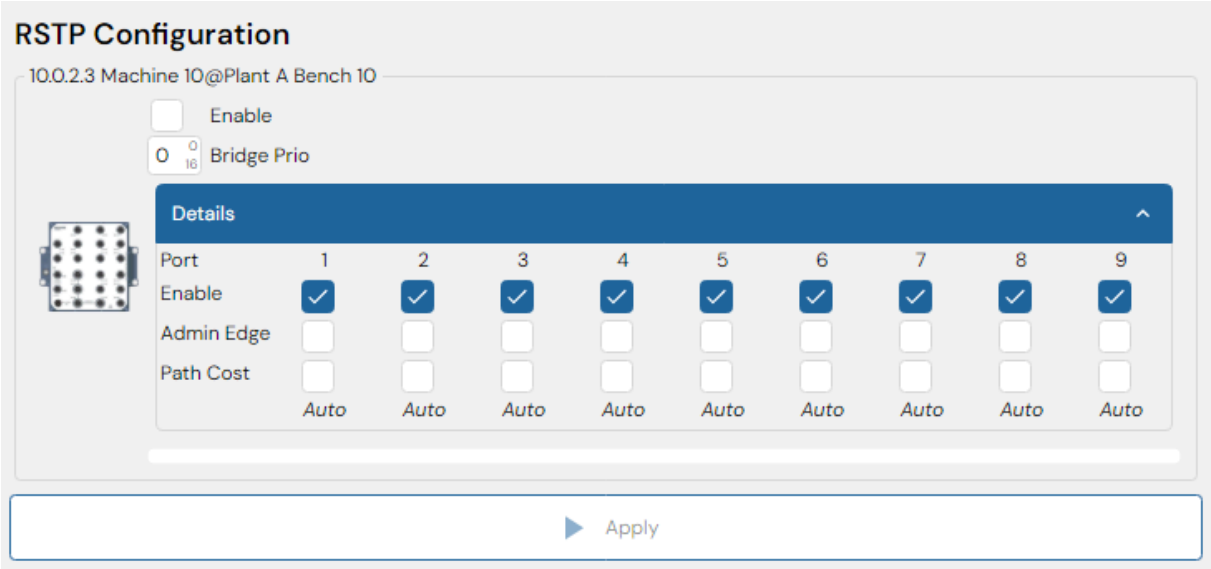
Additionally, you may click “Propose Ports” at the bottom of the interface to have suitable MRP ports suggested for you (if any exist).

4.4.11. Rapid Spanning Tree Protocol (RSTP)

WeOS 4 WeOS 5

RSTP is a task that allows for configuration of RSTP-meshes in the network.

Interface Components



Depicted above is the RSTP configuration interface, which is a list of selected devices, in this case one, with RSTP configuration details listed.

Edit the fields directly in the list. To enable RSTP or to set Admin Edge on port level click the “down-arrow” icon to the left of each device in the list.

> ⚠ Warning
>
> The network might become momentarily unstable when the configuration is applied, and the connection to devices may be lost.

4.4.12. Ring Coupling (RiCo)

WeOS 4 WeOS 5

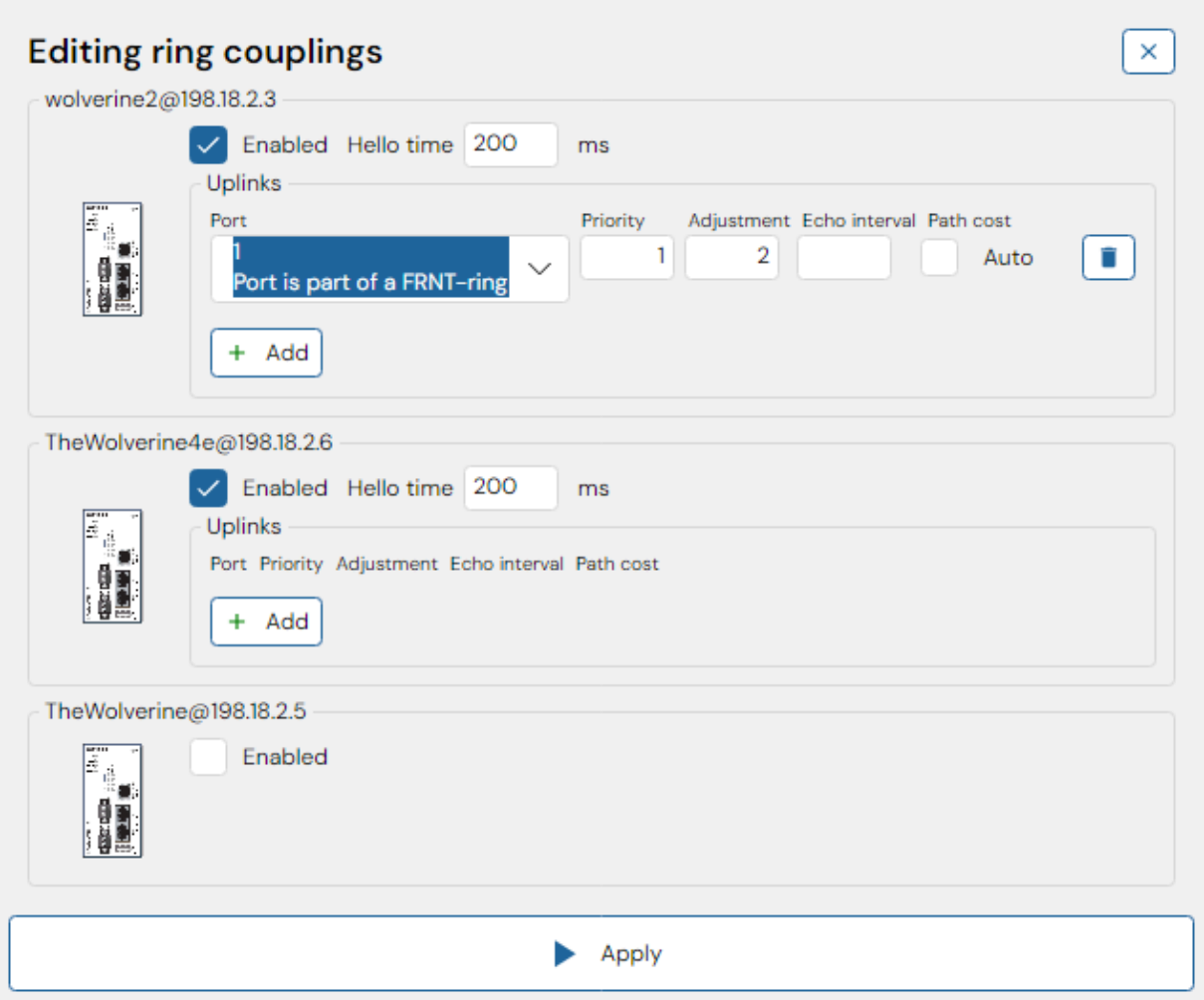
RiCo is a task that allows for configuration of ring coupling for rings in the network.

Interface Components



The initial view of this interface presents you with a dropdown of the detected FRNT rings in the current network topology. Choose a ring and click “Edit couplings” in order to navigate to the RiCo configuration menu. The menu may appear differently depending on the type of ring selected.

RiCo V0 / Ring V0



A list of member devices of the selected ring will be shown and RiCo can be enabled by checking the checkbox labeled Enabled on the desired device.

When RiCo is enabled Hello time can be set and Uplinks can be added.

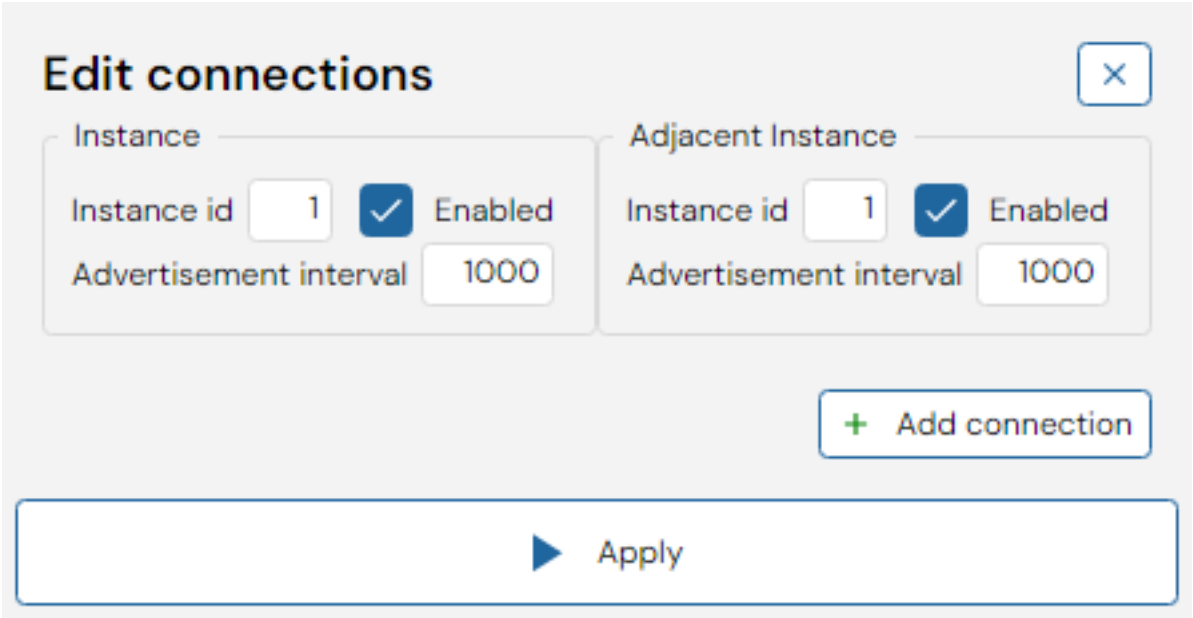
To add an Uplink click the Add button.
- Select a port in the Port dropdown
- Change Priority to the desired value, default is 128
- Change Adjustment to the desired value, default is 0
- Echo Interval is only available for WeOS 4.20 or newer, default value is 200
- Path cost is set default to Auto, To set it manually check the checkbox and fill in the desired value

To remove an uplink click the trashcan button on the row that should be removed.

To abort configuration: click the button in the top right corner with a cross - that will take you back to the initial RiCo configuration tab.

> **Info**
>
> RiCo V0 is only supported up until WeOS 5.15

RiCo V3 / Ring v2
When chosing to edit the RiCo configuration for an FRNTv2 based ring, you will be presented twith the following interface:



Where the left hand side corresponds to the selected ring instance, and the right hand side corresponds to any detected adjacent ring instance. Herein, you may specify:
- The instance ID
- Whether or not the RiCo instance is enabled
- The advertisment interval. Following the instance options, you may specify a number of connections, added via the Add connection button at the bottom right of the interface. In the

example depicted above, two connections have been added, each with a port, a priority, a hello interval and a preempt choice.

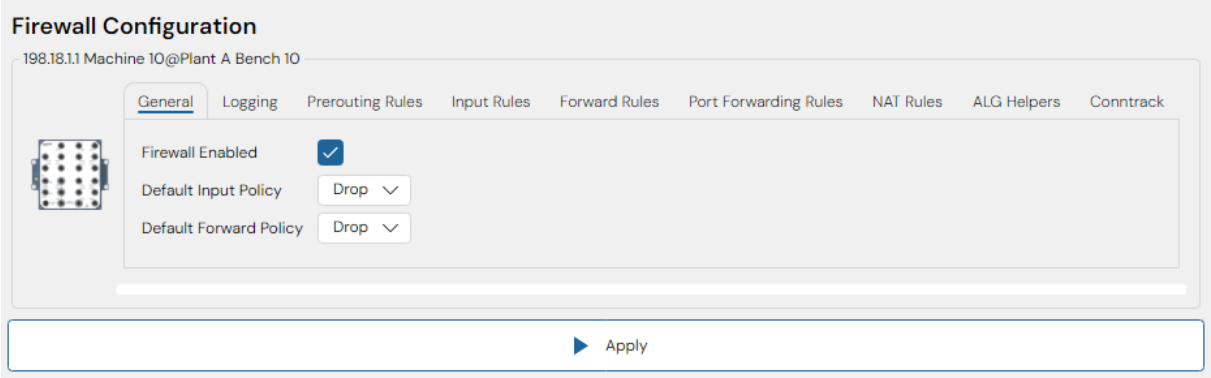
WeConfig will attempt to automatically detect the corresponding right hand side device whenever a left hand side device is selected, if it cannot do so, you may still specify the righthand side device manually.

4.4.13. Firewall

WeOS 4 WeOS 5

Firewall is a licensed task that allows for configuration of firewall rules on selected devices. The current set of devices support by this task is WeOS 4 and WeOS 5 devices.

Interface Components



Depicted above you can observe how the interface might appears for a selection of a singular WeOS 5 device. Where the configuration options for each device is divided into several tabs with the following contents:

What features are available depends on the WeOS version, and whether or not the device is an Extended device.

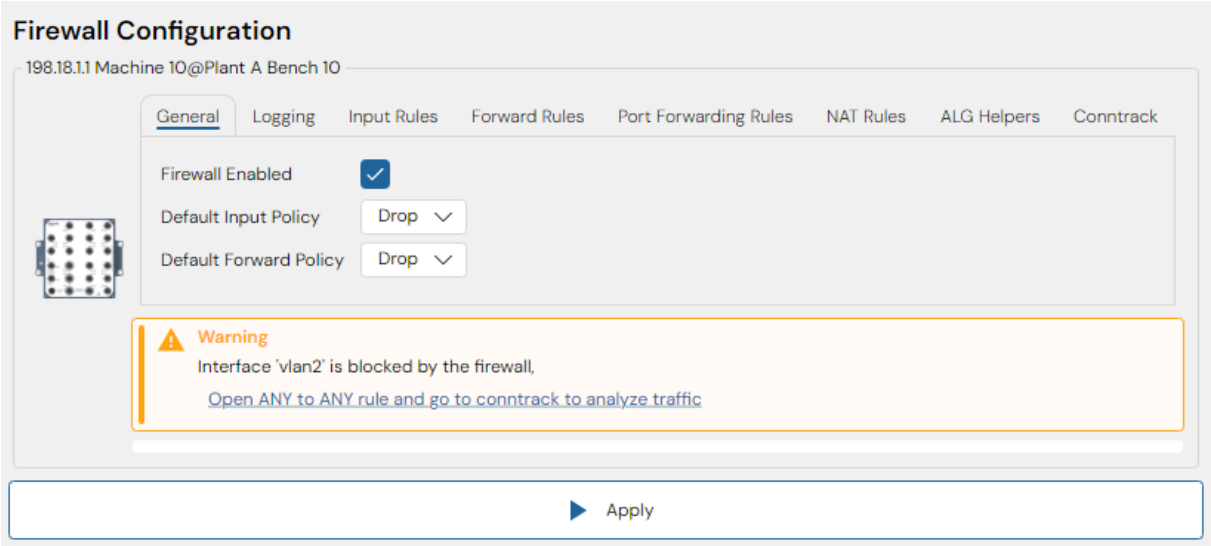
Tab	Description	WeOS 4/5 Standard	WeOS 4 Extended	WeOS 5 Extended
General	Enable / Disable firewall, and default policy	Yes	Yes	Yes
Logging	Enable / Disable logging, and configure rate limit on logs	Yes	Yes	Yes
Input Rules	Rules for incoming packets directed to the device	Yes	Yes	Yes
Forward Rules	Rules for incoming packets not directed to the device	No	Yes	Yes
DPI Filter Rules	Rules for incoming packets with a larger scope	No	Yes	No

Tab	Description	WeOS 4/5 Standard	WeOS 4 Extended	WeOS 5 Extended
Port Forwarding Rules	Expose internal devices via specific ports	No	Yes	Yes
NAT Rules	Rules for Network Address Translation	No	Yes	Yes
Modifier Rules	Rules for modifying the IP header in routed traffic	No	Yes	No
Prerouting Rules	Rules for marking traffic with DSCP before routing	WeOS 5.30+	No	WeOS 5.30+
ALG Helpers	Shorthand Rules for specific protocols	No	Yes	Yes
Conntrack	Maintains information about all incoming and outgoing connections	No	Yes	Yes

On WeOS 5.30 and later, Prerouting Rules are available on both standard and Extended devices.

For rule types with a Custom protocol option, enter the IANA protocol number (1–255). Port matching is available only for TCP and UDP.

General



The general tab allows for enabling or disabling the firewall on the device, and setting the default policy for the firewall. The default policy can be set to either `ACCEPT` or `DROP`. The default policy is the action that the firewall will take if no rules are matched.

For WeOS 5 devices, the firewall can be disabled and rules can still be set, but they will not be enforced until the firewall is enabled.

> **Info**
>
> Default Forward Policy is only available on Extended devices.

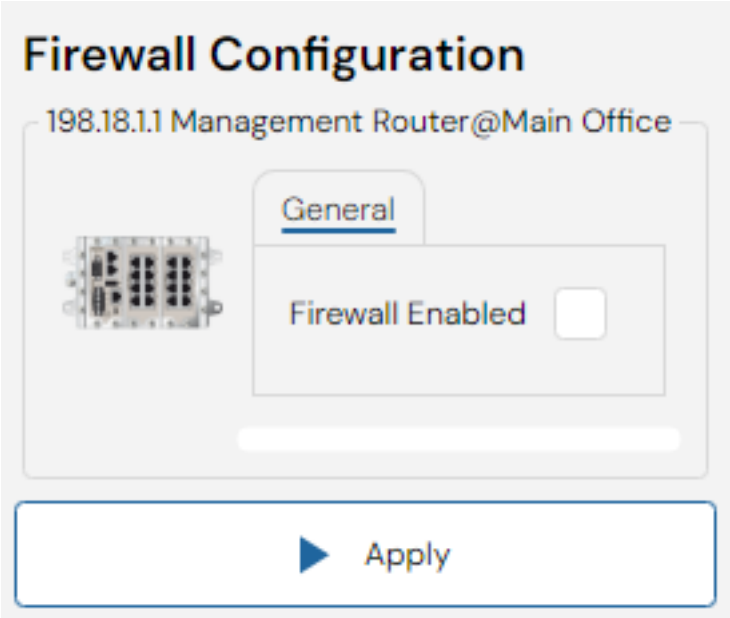
Admonitions

As depicted in the image above, the firewall configuration task may display certain admonitions to the user, depending on the current configuration of the device. These admonitions are hints or warnings that inform the user about potential issues with the current configuration, either in terms of functionality or security posture.

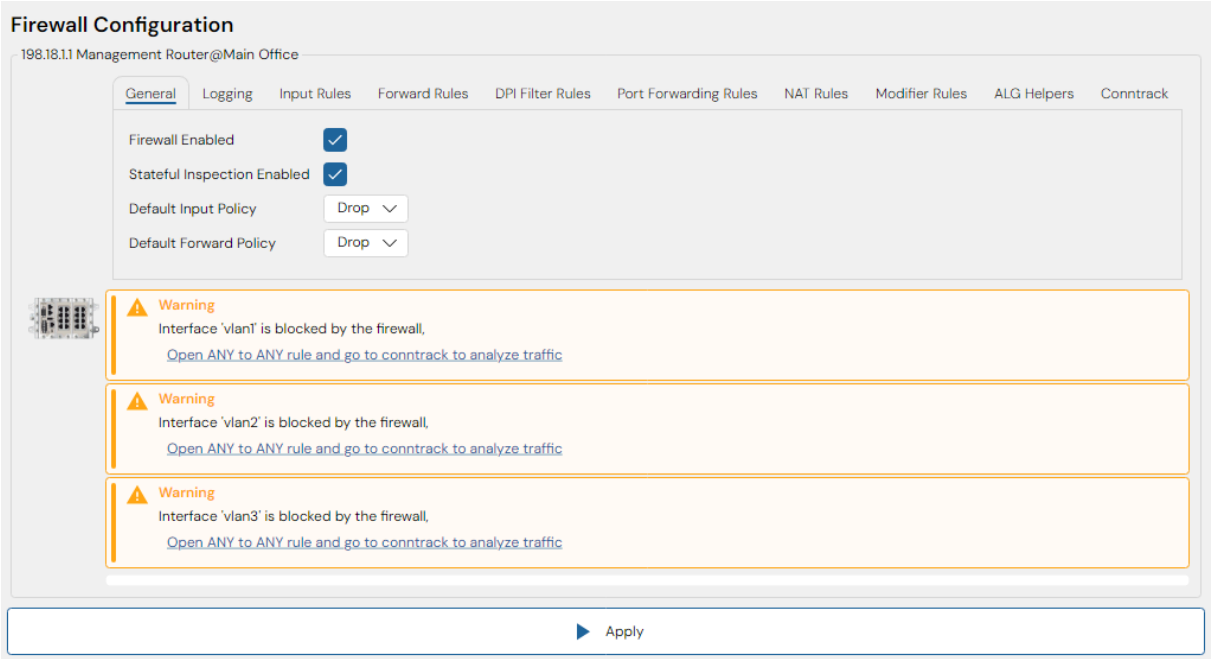
Some examples of admonitions that may appear are:

- Default policy issues: If the default policy is set to `ACCEPT`, an admonition may appear warning the user about the potential security risks of this configuration.
- Unlimited logging: If logging is enabled with no rate limit, an admonition may appear warning the user about the potential for denial-of-service attacks.
- VLAN X is blocked: If there are VLANs that are completely blocked by the current firewall rules, an admonition may appear listing those VLANs.
- Service Y is unreachable on VLAN X: If there are services on the device that are unreachable due to the current firewall rules, an admonition may appear listing those services.

WeOS 4

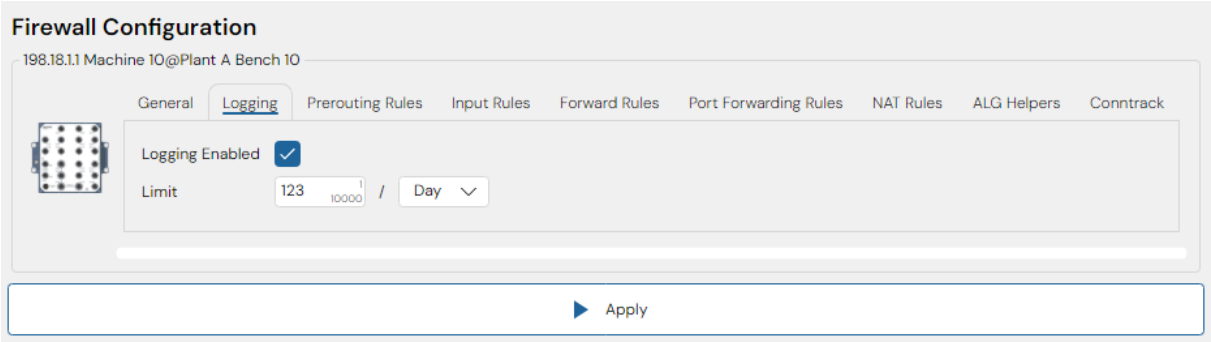


For WeOS 4 devices, the firewall must be enabled to be able to set any rules.



WeOS 4 devices also have the option to set Stateful Inspection, which is an ability to drop packets of invalid state.

Logging



The logging tab allows for enabling or disabling logging of packets that are matched by the firewall rules. The rate limit for logging can also be set, which is the maximum number of logs that can be generated per given time unit.

WeOS 5 devices defaults to unlimited logging, while WeOS 4 devices defaults to 5 logs per second.

> **Info**
>
> The limit should be in the range 1-10000, and the time unit can be set to either `second`, `minute`, `hour`, `day`, or `unlimited`.

> **Warning**
>
> Configuring logging with no rate limit may lower the security posture of the device by opening up for denial-of-service attacks.

Input Rules

The input rules tab allows for setting rules for incoming packets directed to the device. The rules can be set to either `ACCEPT` or `DROP` packets that match the rule.

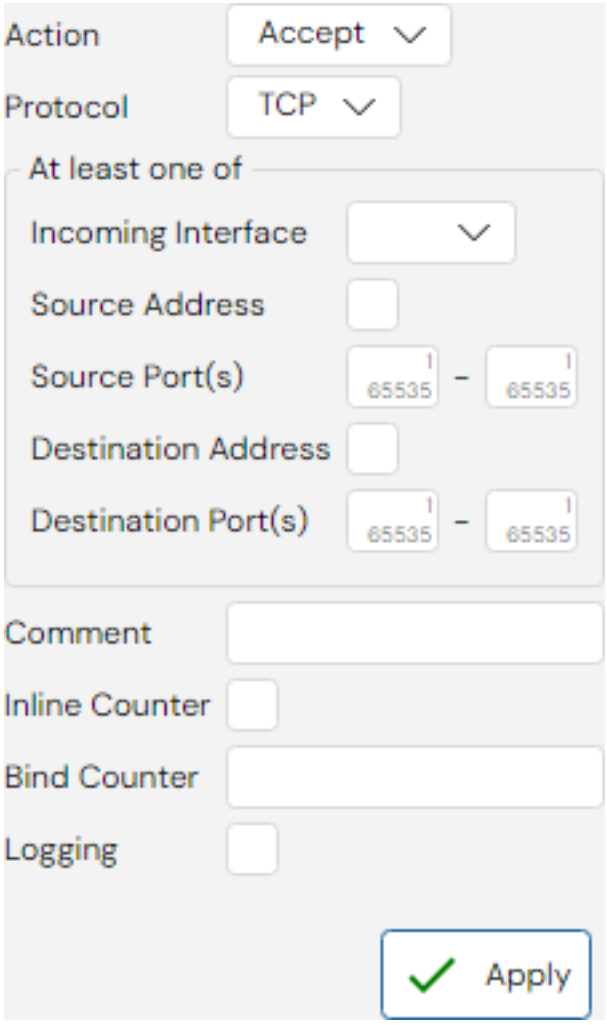
WeOS 5 defaults with no Input Rules, while WeOS 4 defaults with rules that allow ICMP access to the device.

> **Info**
>
> The input rules are checked after the routing decision and applies to packets that are directed to the local device and processes running on the device itself.

> **Tip**
>
> Drag the rows to reorder the rules.

Add Input Rule - WeOS 5

To add a rule to a WeOS 5 device, click the `Add` button. This will open a modal where the following fields can be set:



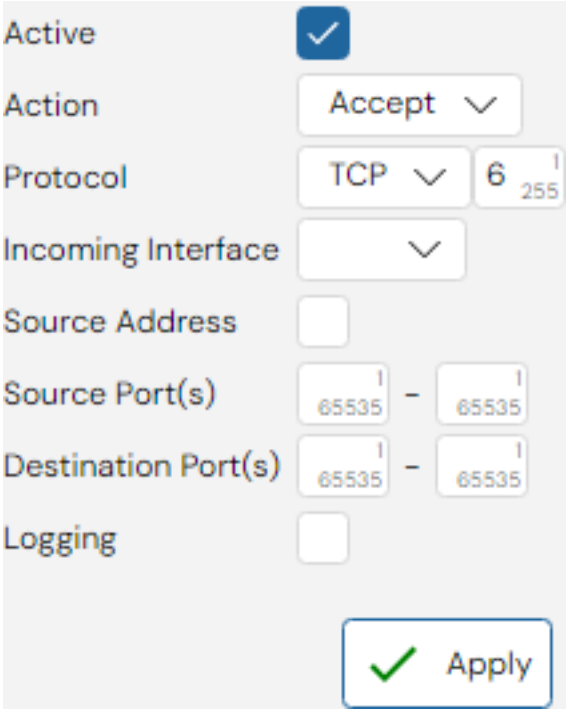
Field	Description	Required	Notes
Action	Accept or drop traffic matching the specified rule	Yes	-
Protocol	The protocol type of the IP payload. Typically TCP or UDP but the filtering can also be made to match other protocols such as ICMP and ESP.	Yes	Will affect what fields are available
Incoming Interface	The interface where the packet comes in	Yes*	See info section below
Source Address	The source IP address of the packet	Yes*	See info section below
Source Port(s)	The source port(s) of the packet	Yes*	Only available if Protocol is `TCP` or `UDP`. See info section below
Destination Address	The destination IP address of the packet	Yes*	See info section below

Field	Description	Required	Notes
Destination Port(s)	The destination port(s) of the packet	Yes*	Only available if Protocol is `TCP` or `UDP`. See info section below
Comment	A comment of the rule	No	-
Inline Counter	The rule uses a counter that only shows in the raw output. This counter is reset any time the firewall configuration is changed.	No	-
Bind Counter	The Counter the rule is assigned to	No	Can only be set if `Inline Counter` is not checked. If the Counter does not exist, one will be created.
Logging	Log traffic matching this rule	No	Logging must be enabled in the Logging tab

> *i* Info
>
> Yes* =
>
> If `Protocol` is `TCP` or `UDP`:
> - At least one of `Incoming Interface`, `Source Address`, `Destination Address`, `Source Port(s)`, or `Destination Port(s)` must be set.
>
> If `Protocol` is not `TCP` or `UDP`:
> - At least one of `Incoming Interface`, `Source Address`, or `Destination Address` must be set.

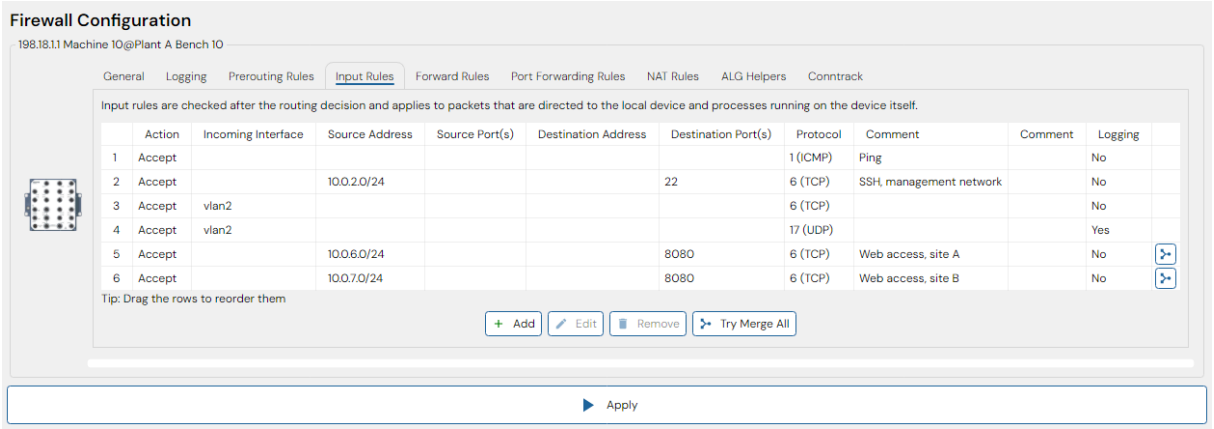
Add Input Rule - WeOS 4

To add a rule to a WeOS 4 device, click the `Add` button. This will open a modal where the following fields can be set:



Field	Description	Required	Notes
Active	Is this rule active?	Yes	-
Action	Accept or drop traffic matching the specified rule	Yes	-
Protocol	The protocol type of the IP payload. Typically TCP or UDP but the filtering can also be made to match other protocols such as ICMP and ESP.	Yes	Will affect what fields are available
Incoming Interface	The interface where the packet comes in	No	-
Source Address	The source IP address of the packet	No	-
Source Port(s)	The source port(s) of the packet	No	Only available if Protocol is `TCP` or `UDP`
Destination Port(s)	The destination port(s) of the packet	No	Only available if Protocol is `TCP` or `UDP`
Logging	Log traffic matching this rule	No	Logging must be enabled in the Logging tab

Merge Input Rules



Input rules can be merged. If two or more rules could be expressed as a single rule with a broader subnet range, then a button to merge those rules will appear to the right on the relevant rows. When hovering over that button, the rules that will be merged are highligted.

You can also click the “Try Merge All” button to merge all rules at once.

Prerouting Rules

On WeOS 5.30 and later, the Prerouting Rules tab configures DSCP marking for packets before the device routes them. Add and order rules by matching on an incoming interface, source or destination address, source or destination port, and protocol. Each rule sets a DSCP value from 0 to 63; ports can be used only with TCP or UDP. Counters, comments, and logging can also be configured where supported by the device.

Forward Rules

The forward rules tab allows for setting rules for incoming packets not directed to the device. The rules can be set to either `ACCEPT` or `DROP` packets that match the rule.

The default is no Forward Rules.

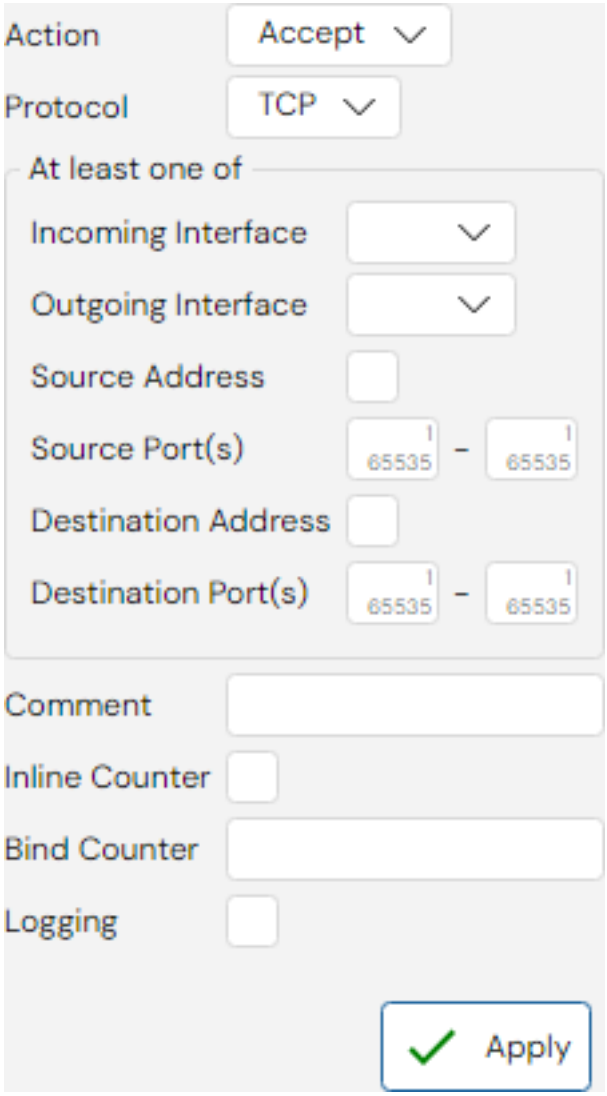
> **Info**
>
> Only available on Extended devices.

> **Info**
>
> The forward rules are checked after the routing decision and applies to packets that are not directed to the local device.

> **Tip**
>
> Drag the rows to reorder the rules.

Add Forward Rule - WeOS 5

To add a rule to a WeOS 5 device, click the `Add` button. This will open a modal where the following fields can be set:



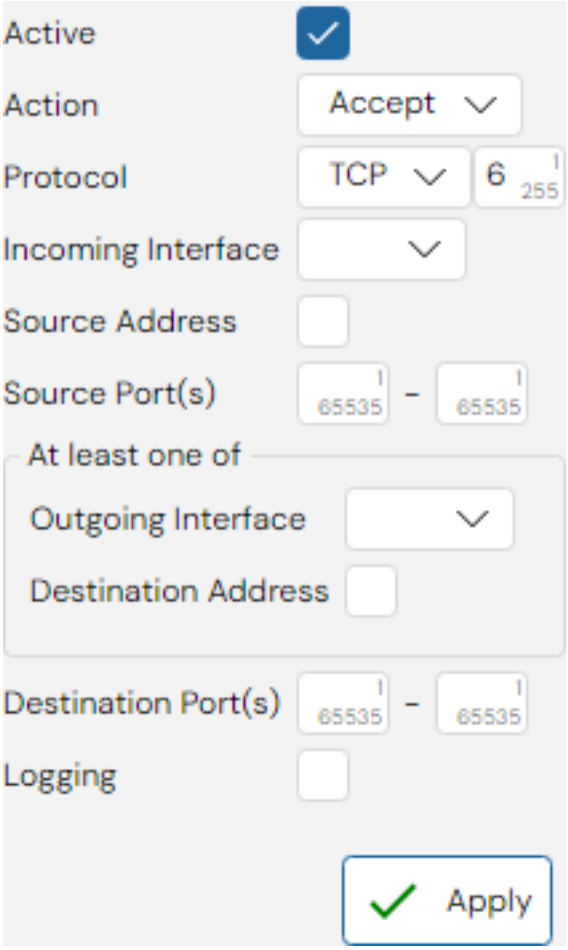
Field	Description	Required	Notes
Action	Accept or drop traffic matching the specified rule	Yes	-
Protocol	The protocol type of the IP payload. Typically TCP or UDP but the filtering can also be made to match other protocols such as ICMP and ESP.	Yes	Will affect what fields are available
Incoming Interface	The interface where the packet comes in	Yes*	See info section below
Outgoing Interface	The interface where the packet is sent out	Yes*	See info section below
Source Address	The source IP address of the packet	Yes*	See info section below

Field	Description	Required	Notes
Source Port(s)	The source port(s) of the packet	Yes*	Only available if Protocol is `TCP` or `UDP`. See info section below
Destination Address	The destination IP address of the packet	Yes*	See info section below
Destination Port(s)	The destination port(s) of the packet	Yes*	Only available if Protocol is `TCP` or `UDP`. See info section below
Comment	A comment of the rule	No	-
Inline Counter	The rule uses a counter that only shows in the raw output. This counter is reset any time the firewall configuration is changed.	No	-
Bind Counter	The Counter the rule is assigned to	No	Can only be set if `Inline Counter` is not checked. If the Counter does not exist, one will be created.
Logging	Log traffic matching this rule	No	Logging must be enabled in the Logging tab

> *i* Info
>
> Yes* =
>
> If `Protocol` is `TCP` or `UDP`:
> - At least one of `Incoming Interface`, `Outgoing Interface`, `Source Address`, `Destination Address`, `Source Port(s)`, or `Destination Port(s)` must be set.
>
> If `Protocol` is not `TCP` or `UDP`:
> - At least one of `Incoming Interface`, `Outgoing Interface`, `Source Address`, or `Destination Address` must be set.

Add Forward Rule - WeOS 4

To add a rule to a WeOS 4 device, click the `Add` button. This will open a modal where the following fields can be set:

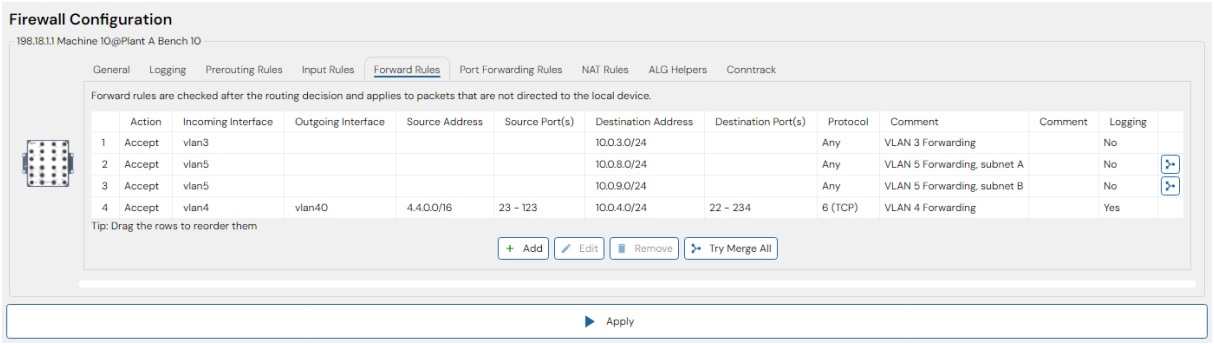


Field	Description	Required	Notes
Active	Is this rule active?	Yes	-
Action	Accept or drop traffic matching the specified rule	Yes	-
Protocol	The protocol type of the IP payload. Typically TCP or UDP but the filtering can also be made to match other protocols such as ICMP and ESP.	Yes	Will affect what fields are available
Incoming Interface	The interface where the packet comes in	No	-
Source Address	The source IP address of the packet	No	-
Source Port(s)	The source port(s) of the packet	No	Only available if Protocol is `TCP` or `UDP`
Outgoing Interface	The interface where the packet is sent out	Yes*	See info section below
Destination Address	The destination IP address of the packet	Yes*	See info section below

Field	Description	Required	Notes
Destination Port(s)	The destination port(s) of the packet	No	Only available if Protocol is `TCP` or `UDP`
Logging	Log traffic matching this rule	No	Logging must be enabled in the Logging tab

> **Info**
>
> Yes* = At least one of `Outgoing Interface` or `Destination Address` must be set.

Merge Forward Rules



Forward rules can be merged. If two or more rules could be expressed as a single rule with a broader subnet range, then a button to merge those rules will appear to the right on the relevant rows. When hovering over that button, the rules that will be merged are highligted.

You can also click the “Try Merge All” button to merge all rules at once.

DPI Filter Rules

Deep Packet Inspection (DPI) rules work similar to regular Packet Filter (Input/Forward) rules, but look further into the payload when deciding what packets to allow or drop.

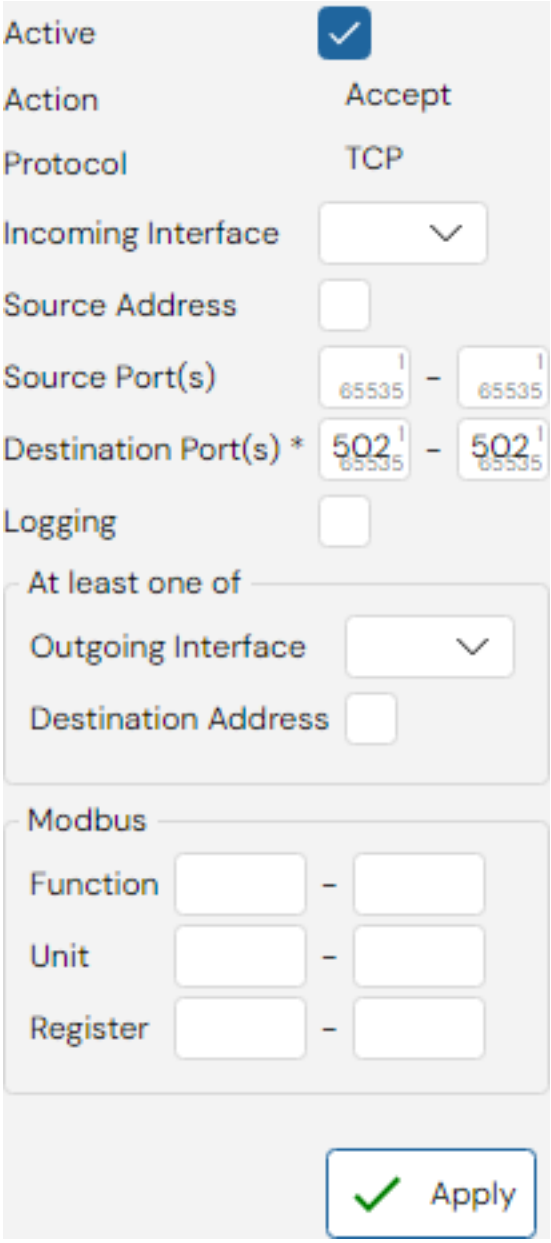
The default is no DPI Filter Rules.

> **Info**
>
> Only avaible for Extended WeOS 4 devices.

> **Tip**
>
> Drag the rows to reorder the rules.

Add DPI Filter Rule

To add a rule to a WeOS 4 device, click the `Add` button. This will open a modal where the following fields can be set:



Field	Description	Required	Notes
Active	Is this rule active?	Yes	-
Action	Accept or drop traffic matching the specified rule	Yes	Only Accept is allowed
Protocol	The protocol type of the IP payload	Yes	Only TCP is allowed
Incoming Interface	The interface where the packet comes in	No	-
Source Address	The source IP address of the packet	No	-
Source Port(s)	The source port(s) of the packet	No	-

Field	Description	Required	Notes
Destination Port(s)	The destination port(s) of the packet	No	Default is 502
Logging	Log traffic matching this rule	No	Logging must be enabled in the Logging tab
Outgoing Interface	The interface where the packet is sent out	Yes*	See info section below
Destination Address	The destination IP address of the packet	Yes*	See info section below
Modbus Function	DPI can filter on Modbus function codes. A range of codes can be specified.	No	-
Modbus Unit	DPI can filter on Modbus unit ID. More than one Modbus device may sit behind the same IP address, use this parameter to specify a single device.	No	-
Modbus Register	DPI can filter on Modbus register addresses. Note that the meaning of this filter varies depending on the function code.	No	-

> **Info**
>
> Yes* = At least one of `Outgoing Interface` or `Destination Address` must be set.

Port Forwarding Rules

The port forwarding rules tab allows for exposing internal devices via specific ports. The rules can be set to either `ACCEPT` or `DROP` packets that match the rule.

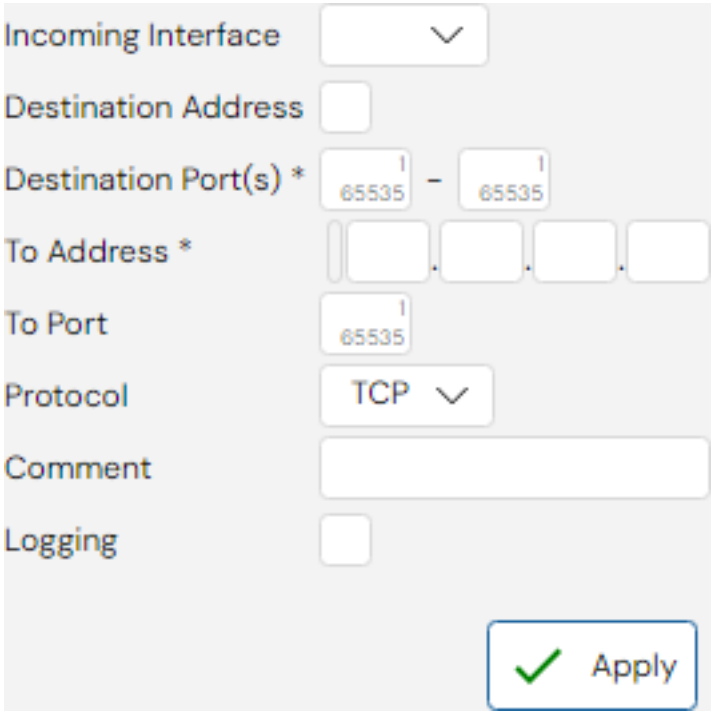
The default is no Port Forwarding Rules.

> **Info**
>
> Only available on Extended devices.

> **Tip**
>
> Drag the rows to reorder the rules.

Add Port Forwarding Rule - WeOS 5

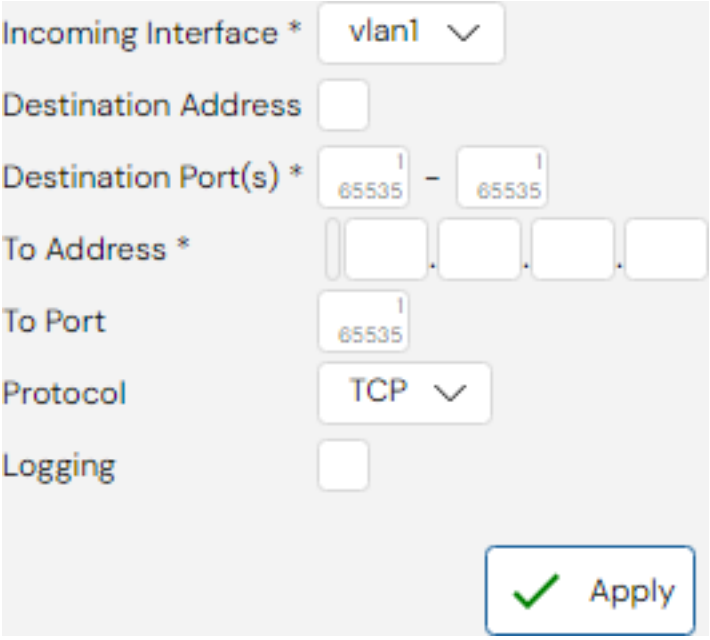
To add a rule to a WeOS 5 device, click the `Add` button. This will open a modal where the following fields can be set:



Field	Description	Required	Notes
Incoming Interface	The interface where the packet comes in	No	-
Destination Address	The destination IP address of the packet	No	-
Destination Port(s)	The destination port(s) of the packet	Yes	-
To Address	The destination address where the packets are to be forwarded	Yes	-
To Port	The destination port where the packets are to be forwarded	Yes	-
Protocol	The protocol type of the IP payload	Yes	TCP or UDP
Comment	A comment of the rule	No	-
Logging	Log traffic matching this rule	No	Logging must be enabled in the Logging tab

Add Port Forwarding Rule - WeOS 4

To add a rule to a WeOS 4 device, click the `Add` button. This will open a modal where the following fields can be set:



Field	Description	Required	Notes
Incoming Interface	The interface where the packet comes in	Yes	-
Destination Address	The destination IP address of the packet	No	-
Destination Port(s)	The destination port(s) of the packet	Yes	-
To Address	The destination address where the packets are to be forwarded	Yes	-
To Port	The destination port where the packets are to be forwarded	Yes	-
Protocol	The protocol type of the IP payload	Yes	TCP, UDP, or Any
Logging	Log traffic matching this rule	No	Logging must be enabled in the Logging tab

NAT Rules

Network Address Translation (NAT) can be used to hide private subnets behind a single public IP address.

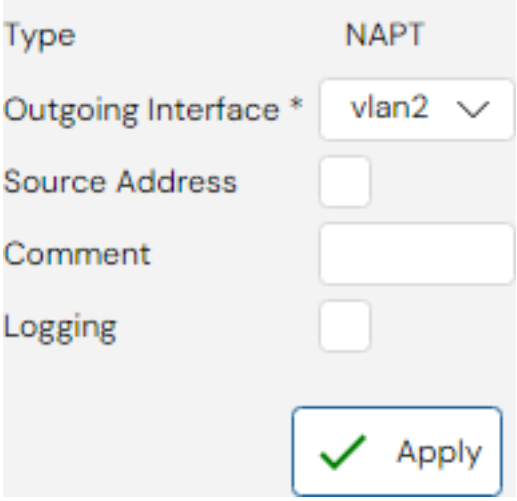
The default is no NAT Rules.

> **Info**
>
> Only available on Extended devices.

> **Tip**
>
> Drag the rows to reorder the rules.

Add NAT Rule - WeOS 5

To add a rule to a WeOS 5 device, click the `Add` button. This will open a modal where the following fields can be set:

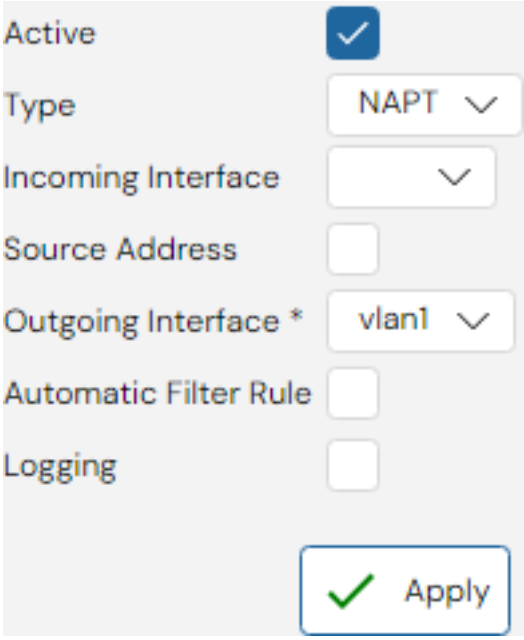


Field	Description	Required	Notes
Type	The type of NAT rule	Yes	Only `NAPT` is supported on WeOS 5
Outgoing Interface	The interface where the packet is sent out	Yes	-
Source Address	The source IP address of the packet	No	-
Comment	A comment of the rule	No	-
Logging	Log traffic matching this rule	No	Logging must be enabled in the Logging tab

Add NAT Rule - WeOS 4

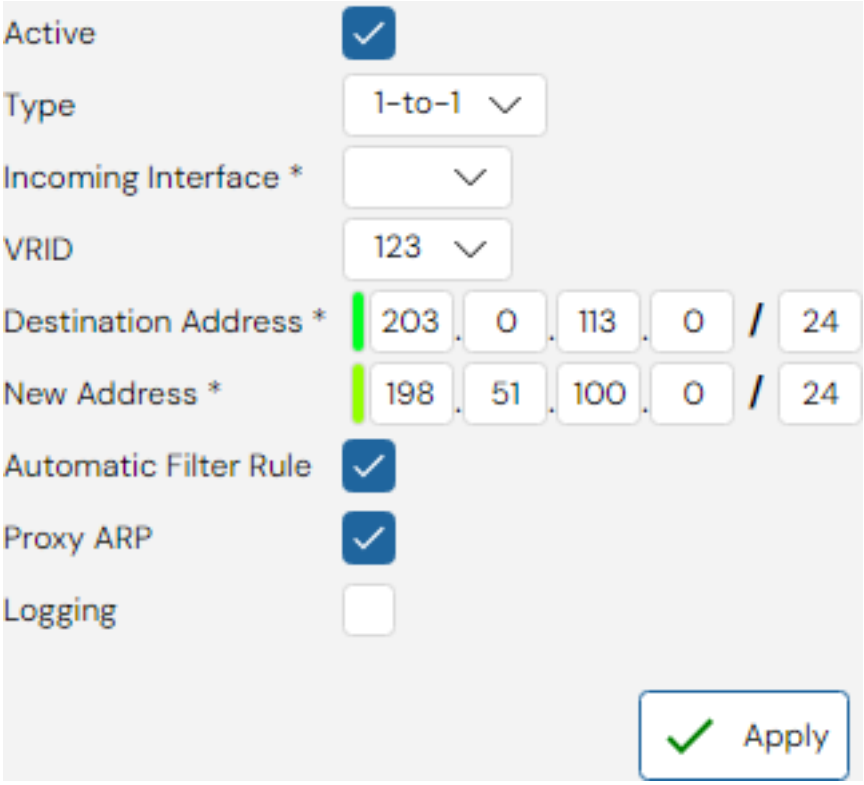
To add a rule to a WeOS 4 device, click the `Add` button. This will open a modal where the following fields can be set:

NAPT:



Field	Description	Required	Notes
Active	Is this rule active?	Yes	-
Type	The type of NAT rule	Yes	`NAPT` or `1:1`, will change what fields are available.
Incoming Interface	The interface where the packet comes in	No	-
Source Address	The source IP address of the packet	No	-
Outgoing Interface	The interface where the packet is sent out	Yes	-
Automatic Filter Rule	If set, an automatic (invisible) packet filter rule will be created in the forward filtering chain allowing packets matching this NAT rule. Do not set this option if you want to manage forwarding rules yourself.	No	-
Logging	Log traffic matching this rule	No	Logging must be enabled in the Logging tab

1-to-1:



Field	Description	Required	Notes
Active	Is this rule active?	Yes	-
Type	The type of NAT rule	Yes	`NAPT` or `1:1`, will change what fields are available.
Incoming Interface	The interface where the packet comes in	Yes	-
VRID	Virtual Router ID	No	Make sure the VRID exists in the device, or it will not 'stick' when applying the NAT Rule
Destination Address	Packets arriving on the inbound interface and has the IP destination within this subnet will be NATed	Yes	-
New Address	The new destination IP network for the NAT	Yes	The subnet size of `Destination Address` and `New Address` must be the same
Automatic Filter Rule	If set, an automatic (invisible) packet filter rule will be created in the forward filtering chain allowing packets matching this NAT rule. Do not set	No	-

Field	Description	Required	Notes
	this option if you want to manage forwarding rules yourself.		
Proxy ARP	WeOS 1-to-1 NAT includes a proxy ARP mechanism, which makes the WeOS unit answer on ARP requests for the external network (Destination Address). The router will only answer on ARP requests originating from the network connected to the Incoming Interface. This makes it possible to use 1-to-1 NAT to pick up traffic to a specific subnet from within a larger network without the need of explicit routing settings.	No	-
Logging	Log traffic matching this rule	No	Logging must be enabled in the Logging tab

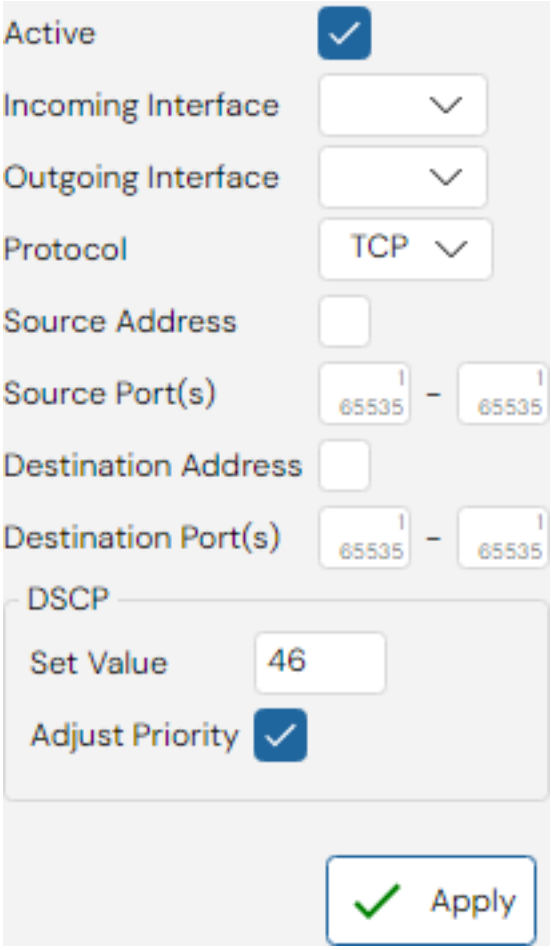
Modifier Rules
Changes the DSCP bits in the IP header for routed traffic.

The default is no Modifier Rules.

> **Info**
>
> Only avaible for Extended WeOS 4 devices.

> **Tip**
>
> Drag the rows to reorder the rules.

Add Modifier Rule
To add a rule to a WeOS 4 device, click the `Add` button. This will open a modal where the following fields can be set:



Field	Description	Required	Notes
Active	Is this rule active?	Yes	-
Incoming Interface	The interface where the packet comes in	No	-
Outgoing Interface	The interface where the packet is sent out	No	-
Protocol	The protocol type of the IP payload. Typically TCP or UDP but the filtering can also be made to match other protocols such as ICMP and ESP.	Yes	-
Source Address	The source IP address of the packet	No	-
Source Port(s)	The source port(s) of the packet	No	-
Destination Address	The destination IP address of the packet	No	-
Destination Port(s)	The destination port(s) of the packet	No	-
DSCP Set Value	The DSCP value to be set for packets matching this rule	Yes	Valid values 0-63
DSCP Adjust Priority	Indicates if the modified DSCP value should be used for switch internal	No	-

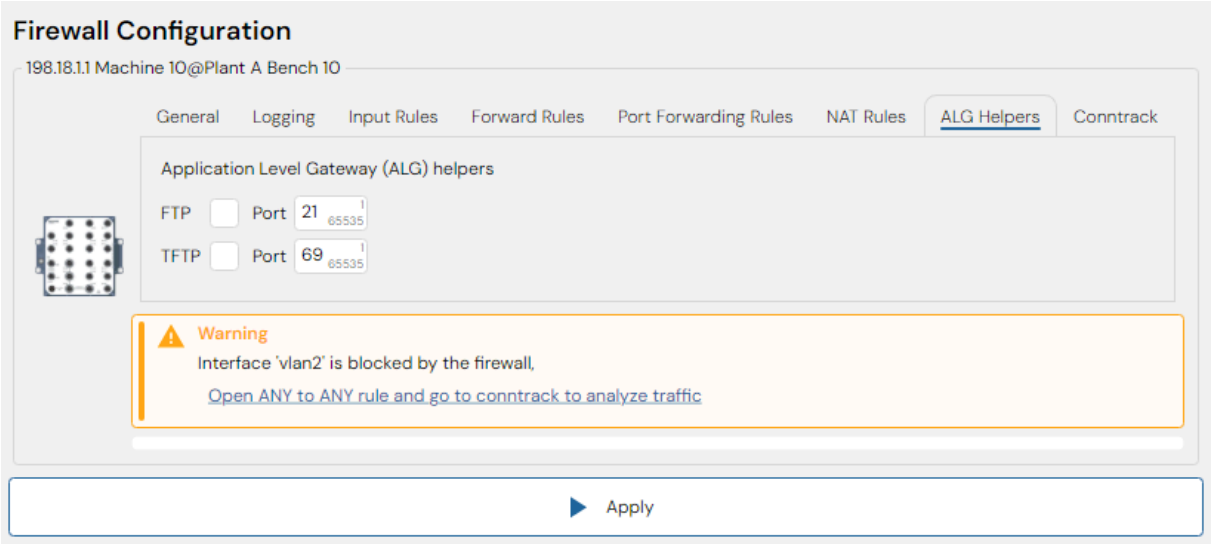
Field	Description	Required	Notes
	prioritising and applied to VLAN-priority on tagged packets		

ALG Helpers

Application Layer Gateway (ALG) helpers are shorthand rules for specific protocols.

The default is no ALG Helpers.

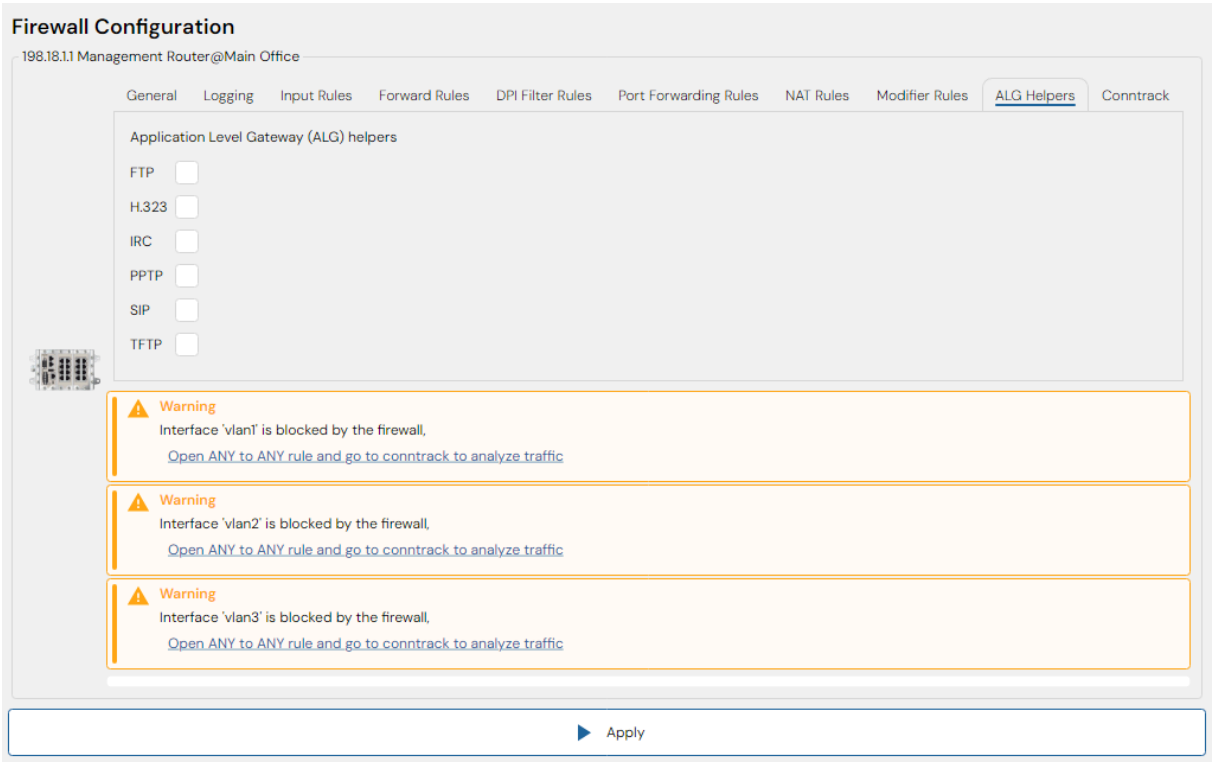
> **Info**
>
> Only available on Extended devices.



FTP and TFTP are the only ALG Helpers available for WeOS 5 devices.

> **Note**
>
> The ability to specify FTP/TFTP ports is available starting with WeOS 5.26 and upwards.

WeOS 4

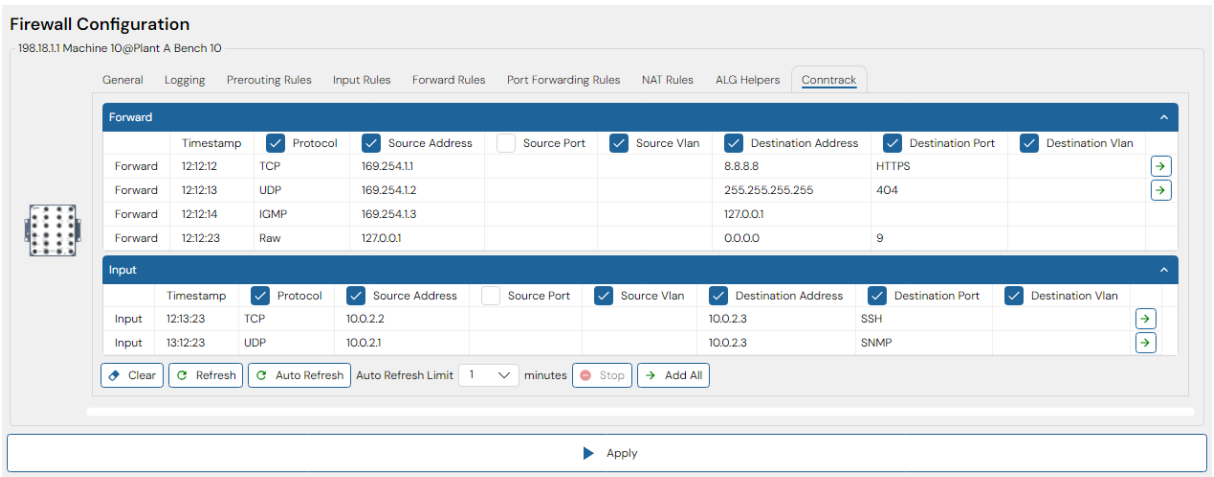


The following ALG Helpers are available for WeOS 4 devices:

- FTP
- H.323
- IRC
- PPTP
- SIP
- TFTP

Conntrack

Conntrack maintains information about all incoming and outgoing connections.



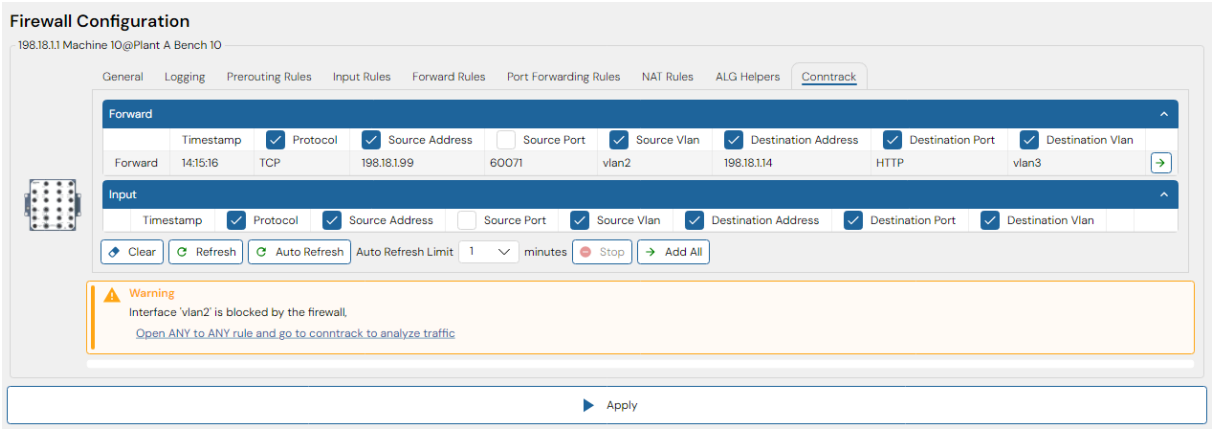
> **Info**
>
> For WeOS 5 devices, the firewall must be enabled to be able to access the conntrack.
>
> Enable the firewall in the General tab and press `Apply`.

> **Tip**
>
> Click on the column headers to sort the table by that column.

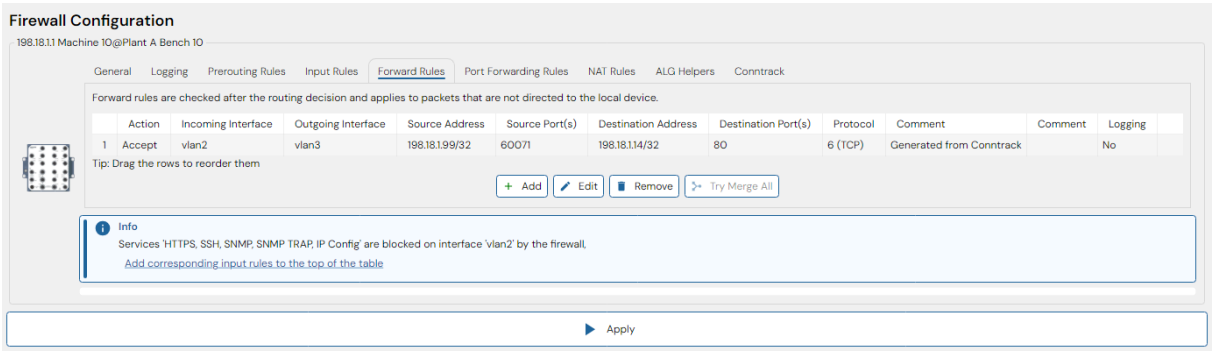
Field	Description
Timestamp	Timestamp of when the conntrack entry was read. This is NOT a timestamp of when the connection was made. Some entries stay in the conntrack for a long time.
Protocol	The IP protocol of the conntrack entry
Source Address	The source IP address, original direction
Source Port	The source port, original direction
Source Vlan	The source vlan, original direction
Destination Address	The destination IP address, original direction
Destination Port	The destination port, original direction
Destination Vlan	The destination vlan, original direction

> **Info**
>
> Vlan is only shown if WeConfig is able to derive it from the corresponding IP address (otherwise this field will be empty)

Create rule from contrack entry



Some conntrack entries has a green arrow next to them. Click that arrow to create a new rule based on the conntrack entry.



The generated rule will be added to the Input Rules or Forward Rules tab, depending on the direction of the conntrack entry. The contents of this rule will be copied from the conntrack entry, with the fields included being determined by the checked boxes in the table header. If a box is checked, the corresponding field will be included in the generated rule.

It is then up to you to edit the rule to your liking. The `Add All` button can be used to add all generated rules at once.

> **Note**
>
> The generated rule is not applied until you press to `Apply` the firewall configuration.

Reply

Conntrack entries also have “Reply” fields. These are commonly just the same as the original fields, but with the direction reversed.

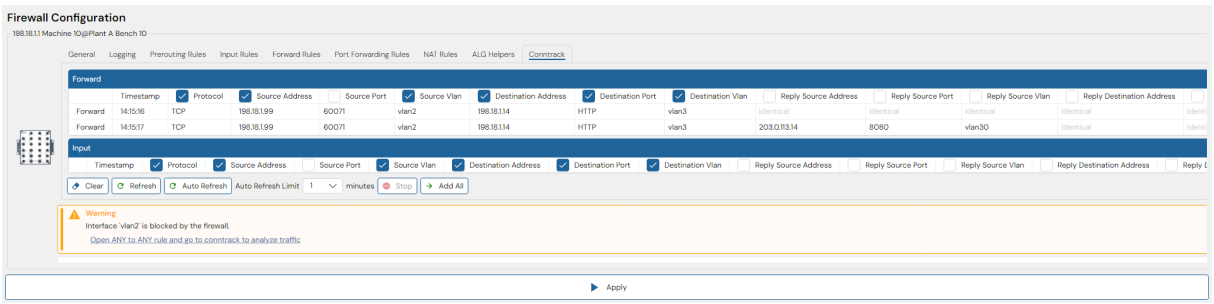
Example:

```
Original Source: 198.18.1.99:60071
Original Destination: 198.18.1.14:80

Reply Source: 198.18.1.14:80
Reply Destination: 198.18.1.99:60071
```

When the Reply is just a mirror of the original, it is not shown in the Conntrack table, to reduce clutter.

But when there are entries with different original and reply addresses/ports, the Reply fields are shown.



Field	Description
Reply Source Address	The source IP address, reply direction

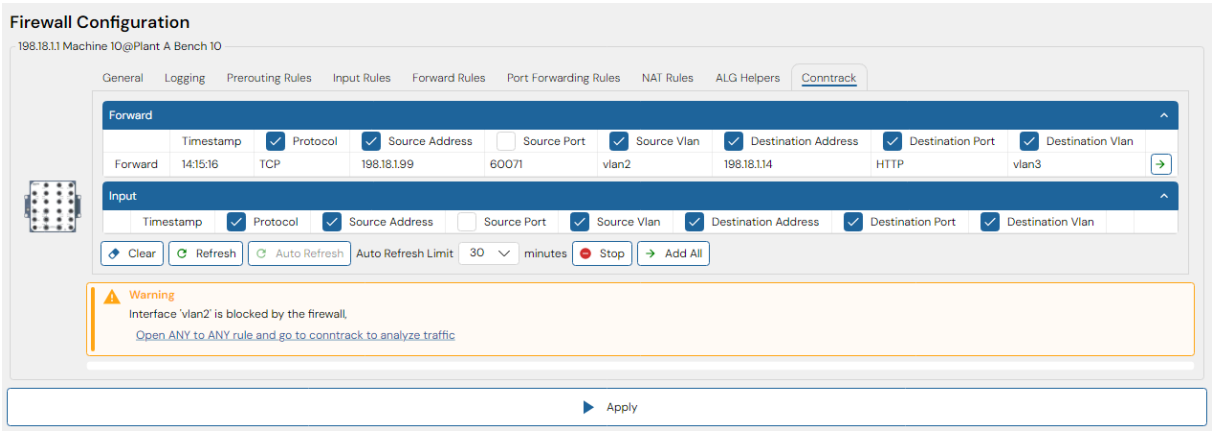
Field	Description
Reply Source Port	The source port, reply direction
Reply Source Vlan	The source vlan, reply direction
Reply Destination Address	The destination IP address, reply direction
Reply Destination Port	The destination port, reply direction
Reply Destination Vlan	The destination vlan, reply direction

If the Reply field is a mirror of the original, it is shown as “Identical”, this way you can quickly see which entries are mirrored and which are not.

> **Info**
>
> Vlan is only shown if WeConfig is able to derive it from the corresponding IP address (otherwise this field will be empty)

Refreshing the Conntrack

Below the Conntrack table, there are some buttons used to refresh the conntrack table.



Clear

The Clear button clears the conntrack table in WeConfig.

> **Info**
>
> Only clears the table in WeConfig, not on the device.

Refresh

The Refresh button refreshes the conntrack table in WeConfig. It will read the current conntrack table from the device and update the table in WeConfig.

Auto Refresh

The Auto Refresh button will automatically refresh the conntrack table in WeConfig every second until the given time limit is reached. This is useful for collecting the conntrack table over a period of time.

> ⚠ Warning
>
> If you deselect the device, or leave the Firewall tab, the Auto Refresh will stop.

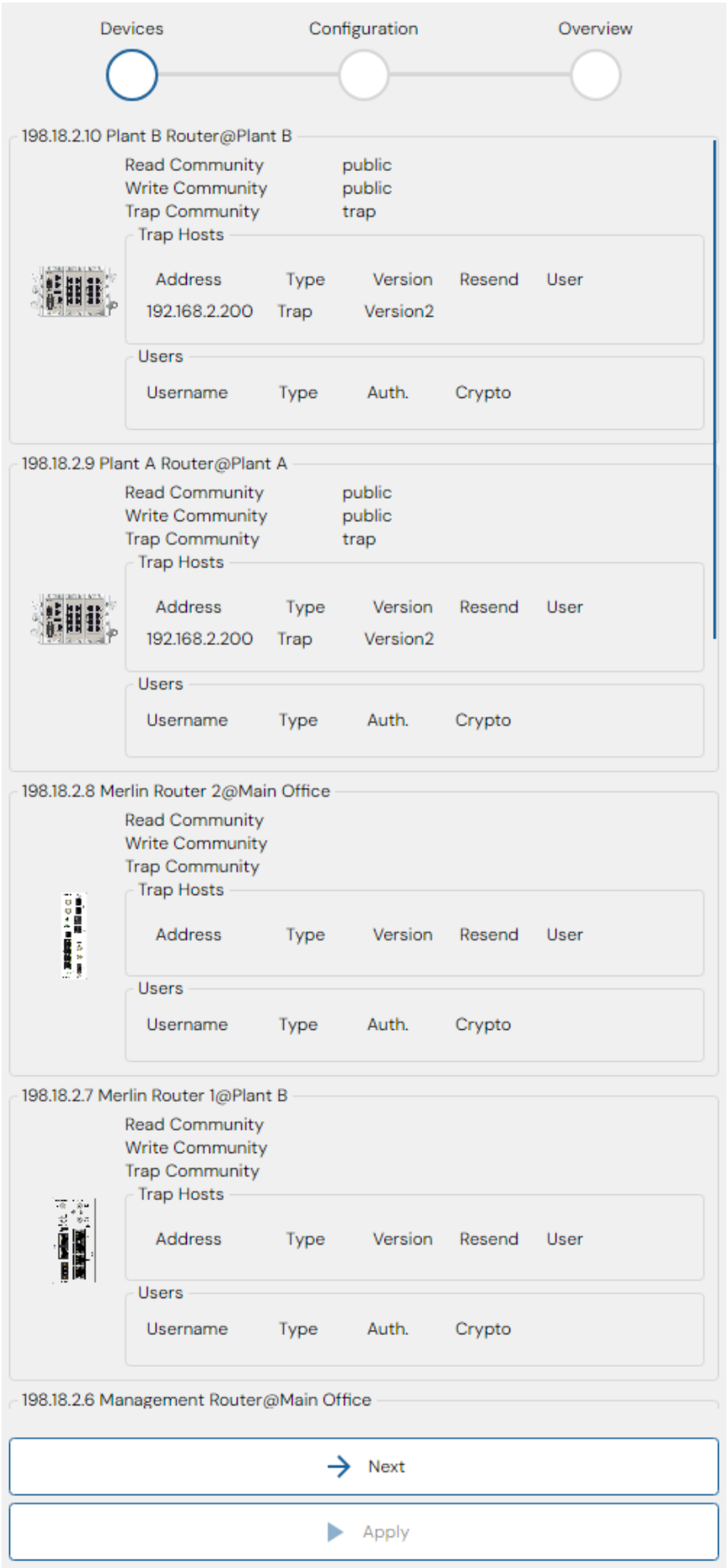
Stop
The Stop button will manually stop the Auto Refresh.

4.4.14. Simple network Management Protocol (SNMP)

WeOS 4 WeOS 5 Ibex XRD Merlin

SNMP configuration is a staged task that allows for configuring the SNMP accessibility on selected devices.

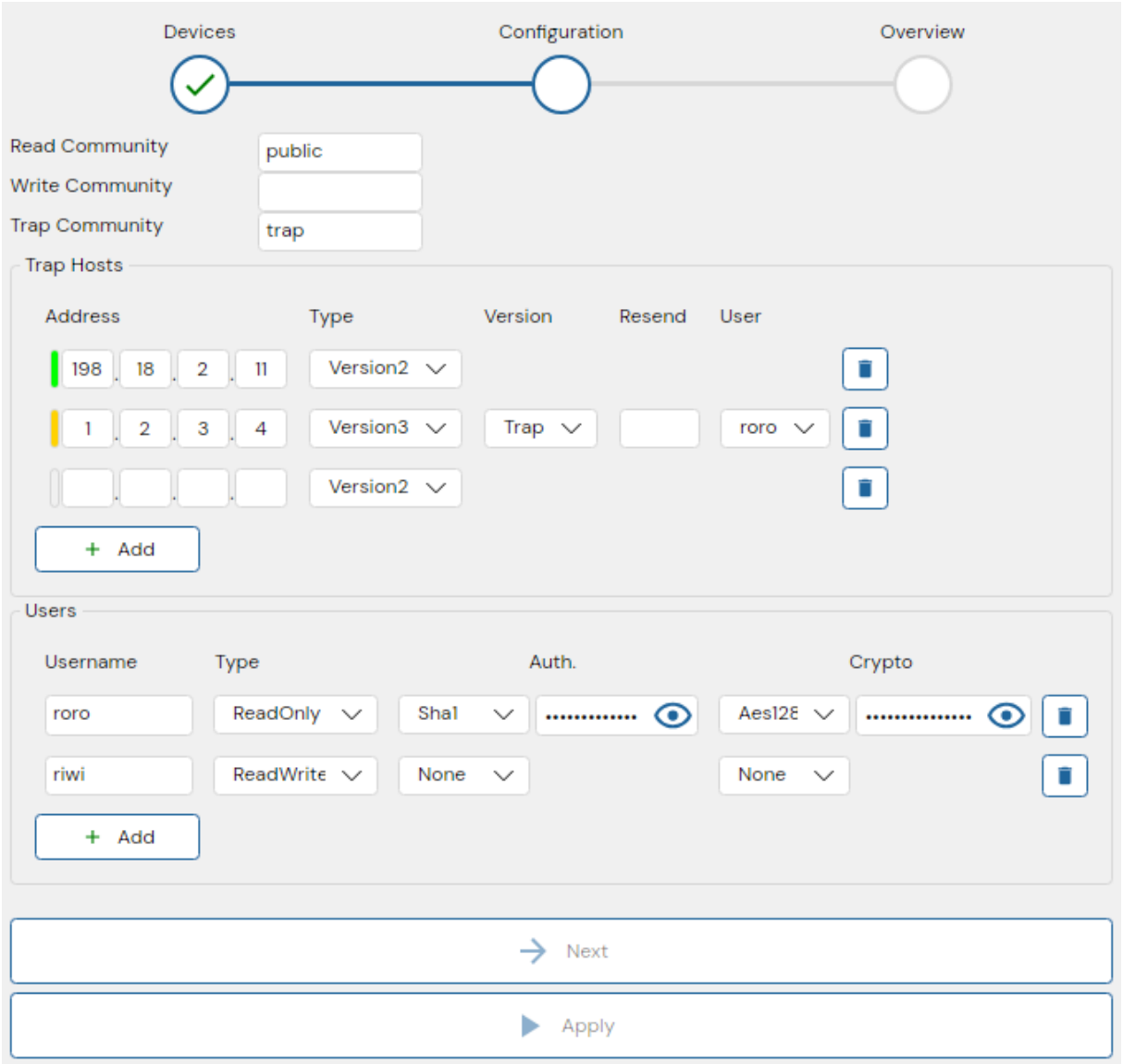
Interface Components
The task is divided into three stages: Devices, Configuration and Overview.


Devices
Configuration
Overview
198.18.2.10 Plant B Router@Plant B
Read Community public
Write Community public
Trap Community trap
Trap Hosts
Address Type Version Resend User
192.168.2.200 Trap Version2
Users
Username Type Auth. Crypto
198.18.2.9 Plant A Router@Plant A
Read Community public
Write Community public
Trap Community trap
Trap Hosts
Address Type Version Resend User
192.168.2.200 Trap Version2
Users
Username Type Auth. Crypto
198.18.2.8 Merlin Router 2@Main Office
Read Community
Write Community
Trap Community
Trap Hosts
Address Type Version Resend User
Users
Username Type Auth. Crypto
198.18.2.7 Merlin Router 1@Plant B
Read Community
Write Community
Trap Community
Trap Hosts
Address Type Version Resend User
Users
Username Type Auth. Crypto
198.18.2.6 Management Router@Main Office
Next
Apply

The first stage is called devices, and it will display the current selection of devices, as well as details regarding their current SNMP configuration. As can be seen above, this is presented as a list of devices, where their read, write and trap communities are listed, followed by a list of trap hosts and users.

To progress to the next stage, click “Confirm”.

Configuration



The second stage is called configuration, and is where the desired configuration for the devices selected in the prior stage is specified. Depicted above is a configuration with the SNMPv2 settings specified as:

- a read community called “public”
- no write community,
- a trap community called “trap”

> **Note**
>
> To disable the relevant SNMPv2 community, leave the field blank.

Additionally, two trap hosts have been configured:

- One version 2 trap host being sent to `198.18.2.11`
- One version 3 trap host being sent to `1.2.3.4` with the user `roro` specified.

Finally, there exist two SNMPv3 users:

- A readonly user called `roro` that does have both authentication and encryption configured.
- A read-write user called `riwi` that does not need any authentication.

> ⚠ Warning
>
> This is in no way a recommended configuration, and is only used as an example.

> Note
>
> If at least one compatible SNMPv3 user is configured on a device, WeConfig automatically selects one of the new users and updates the Device Access for the project to make use of a specific user.

To progress to the next stage, click “Confirm”.

Overview

The final stage is called “Overview” and contains a specification of the updates that will be made to the selected devices listed. Any values that are accepted can be seen in green.

However, not all configurations may be compatible with all devices; if any incompatibilities are found, they are listed on each relevant device, as depicted below:


Devices
Configuration
Overview
198.18.2.10 Plant B Router@Plant B
Read Community
public
Write Community
public
Trap Community
trap
Trap Hosts
Address Type Version Resend User
198.18.2.11 Trap Version2
0.0.0.0 Trap Version2
1.2.3.4 Trap Version3 roro
This device does not support version 3 trap hosts
Users
Username R/W Auth. Crypto
roro ReadOnly Sha1 Aes128
riwi ReadWrite None None
198.18.2.9 Plant A Router@Plant A
Read Community
public
Write Community
public
Trap Community
trap
Trap Hosts
Address Type Version Resend User
198.18.2.11 Trap Version2
0.0.0.0 Trap Version2
1.2.3.4 Trap Version3 roro
This device does not support version 3 trap hosts
Users
Username R/W Auth. Crypto
roro ReadOnly Sha1 Aes128
riwi ReadWrite None None
198.18.2.8 Merlin Router 2@Main Office
Read Community
public
Write Community
Trap Community
Apply

Where we can see that the device in question does not support version 3 of the SNMP trap host configuration.

With this in mind, the apply button is still available, and WeConfig will simply not configure anything unsupported on relevant devices.

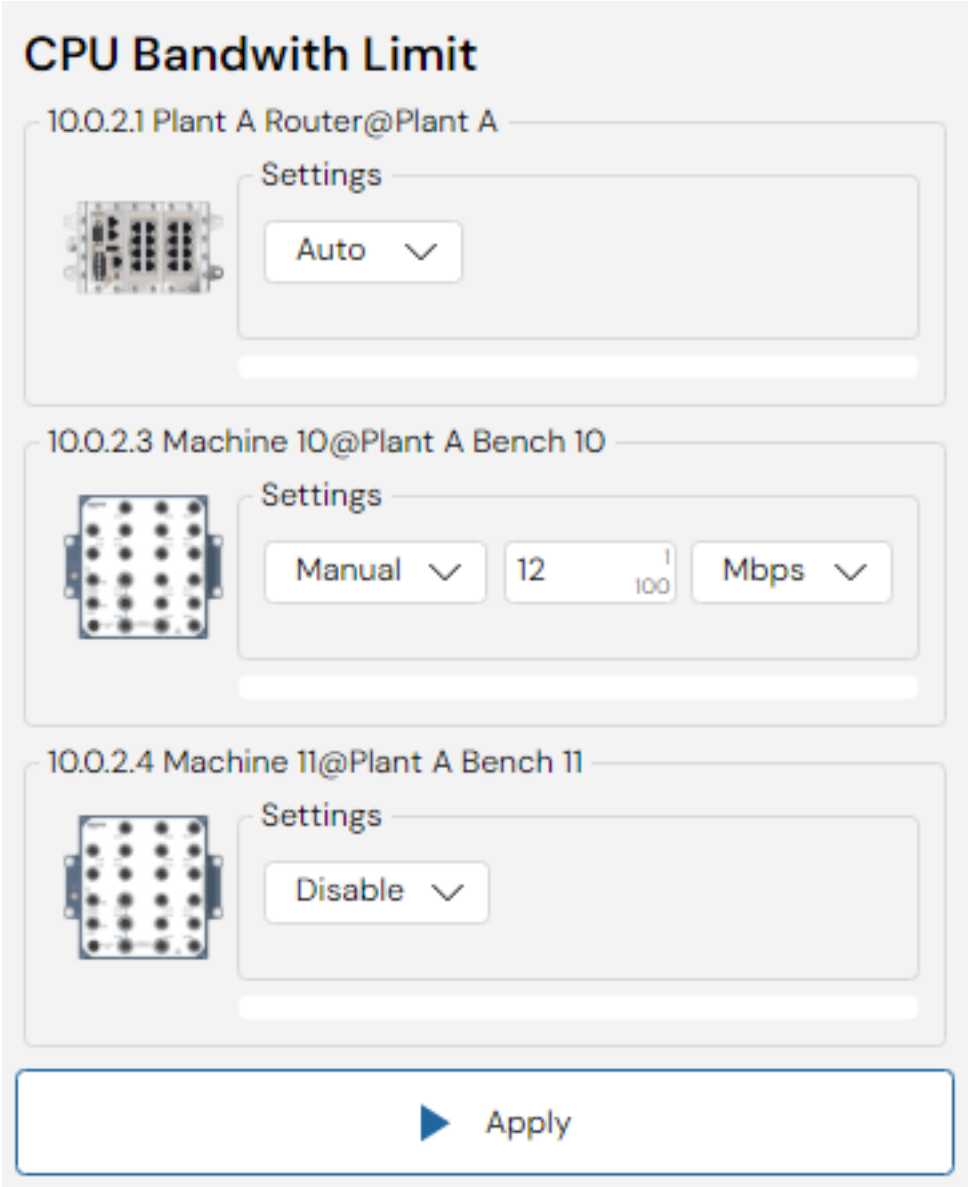
If the resulting configuration is considered unacceptable, simply head back to the prior stage and change it.

4.4.15. CPU

WeOS 4

CPU is a task that allows you to configure CPU bandwidth throttling.

Interface components



For each added device, you may choose the following parameters in the combo box:

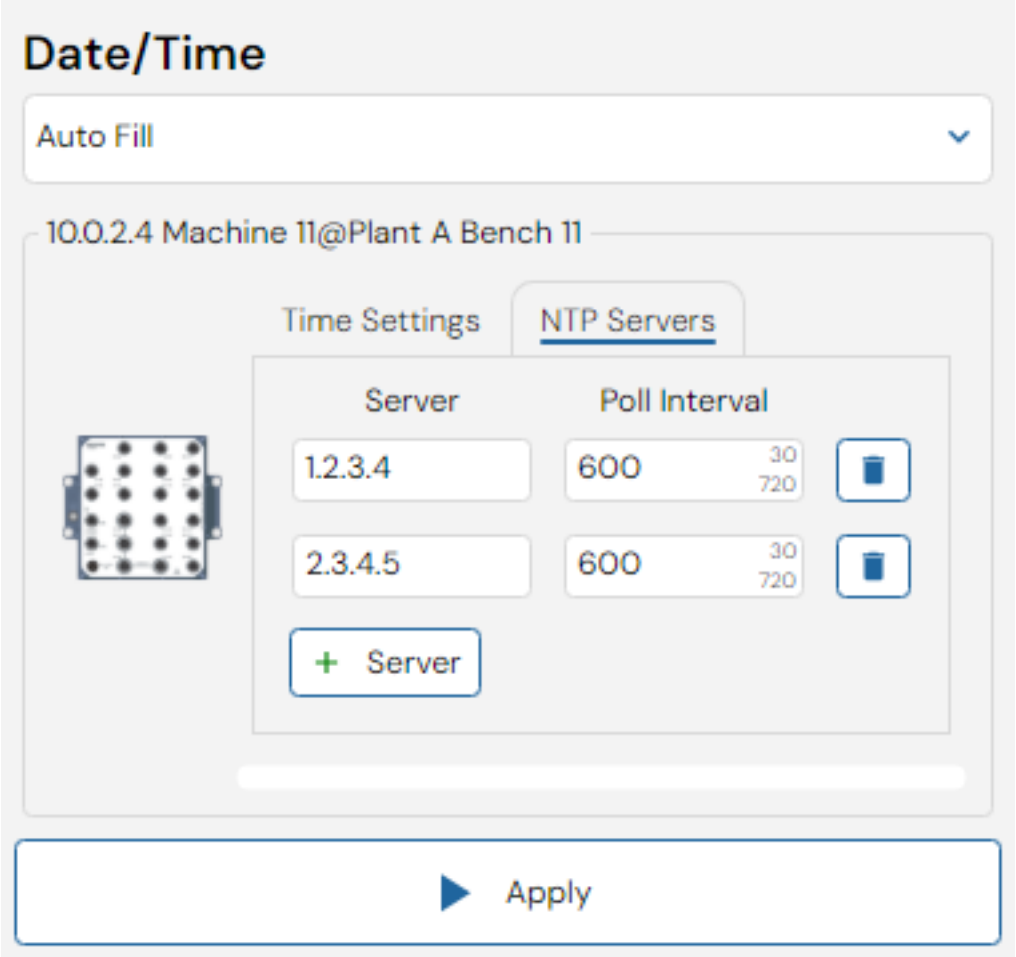
- Disable - CPU bandwidth will not be throttled
- Auto - WeOS will automatically throttle the CPU bandwidth as it sees fit
- Manual - enter a fixed value (expressed with a unit selected in the combo box to the right)

4.4.16. Date/Time

WeOS 4 WeOS 5

Date/Time is a task that allows you to configure current host time, time zone as well as NTP server addresses and poll intervals.

Interface components



As depicted above, this panel contains a list of selected devices where each device has two tabs, one for time settings and one for NTP servers. In the first tab, Time Settings, the host time can be set to match current time, and the time zone can be selected. In the second tab, NTP Servers, you can specify a number of NTP servers, together with their weights or intervals.

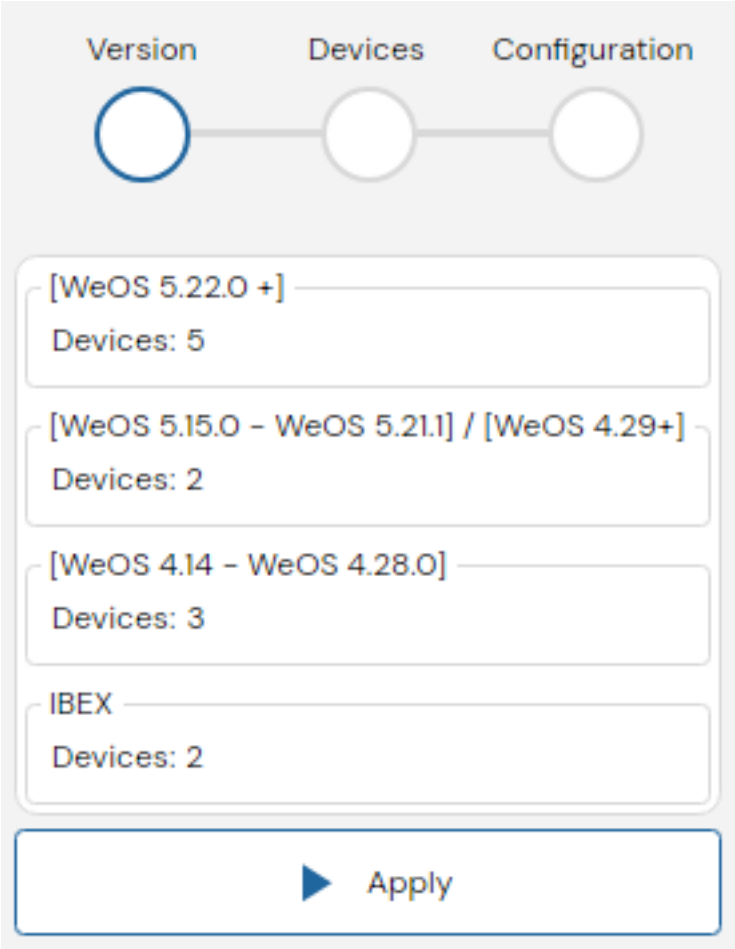
4.4.17. Logging

WeOS 4 WeOS 5 Ibex

Logging configuration is a staged task that allows for configuring syslog behavior on selected devices.

Interface components
The logging task is divided into three stages, Version selection, Devices and Configuration.

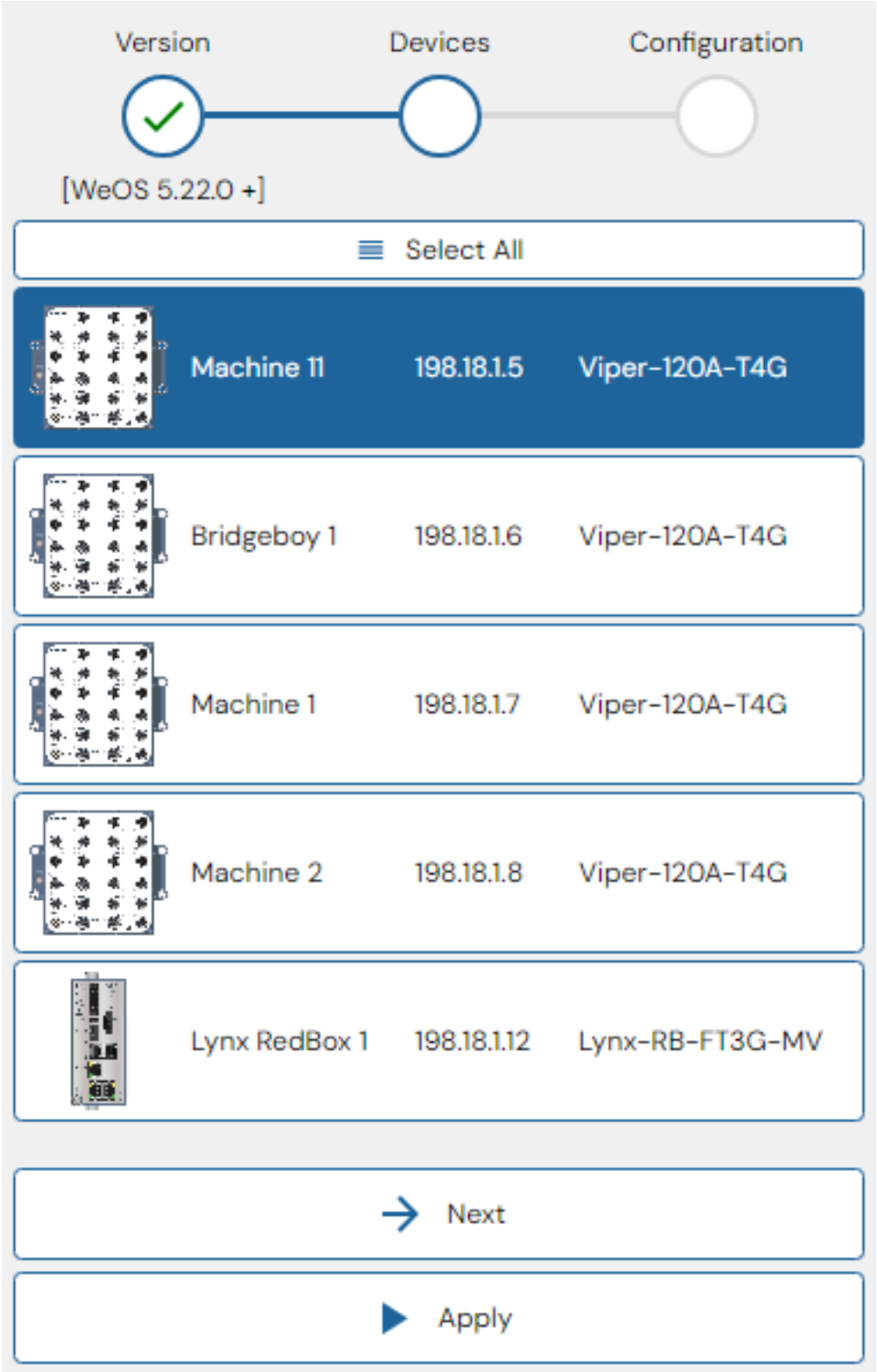
Version



The first stage is called Version, and presents you with a choice for which variant of syslog configuration, amongst the ones available in the current network, to set up. The reason for this view is that different device ranges and firmware versions support vastly different lines of configuration.

Click on the relevant version you want to configure to progress to the next stage.

Devices



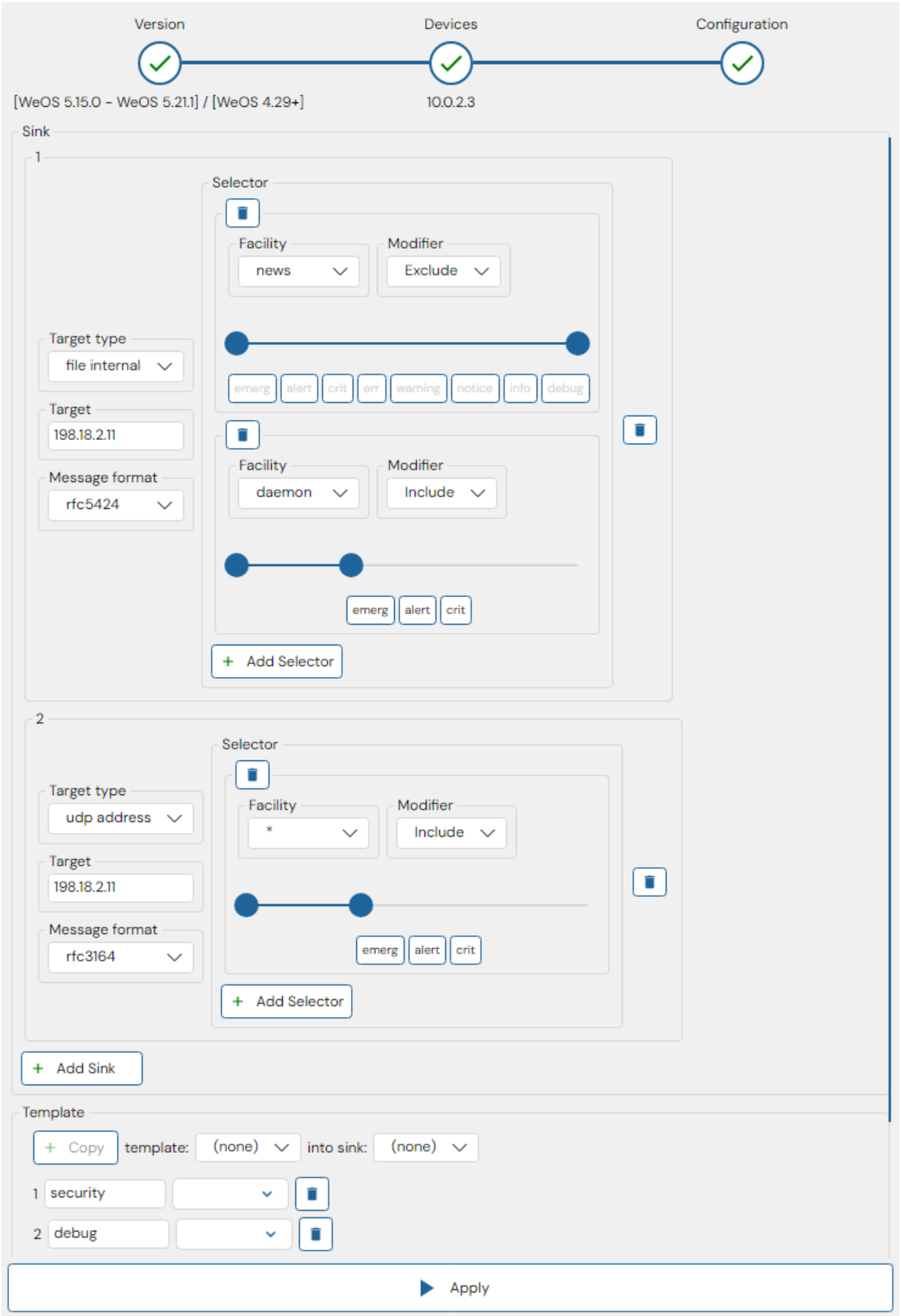
The second stage is called Devices, and presents a list of devices compatible with the selected version of syslog configuration. Any devices already selected in the topology will be preselected in this list, but the selection may be changed before procceeding.

Click on the button labeled “Confirm” to proceed.

Configuration

The final stage is called Configuration, and provides you with an interface to set up the relevant syslog configuration on the selected devices. How this view appears may drastically vary depending on the Version selection.

[WeOS 5.15.0 - WeOS 5.21.1] / [WeOS 4.29+]



When configuring logging for WeOS 5.15 and up to 5.21.1. the interface may appear as above. Therein a list of sinks serve as the primary means of configuration, which can be added to via the “Add Sink” button near the bottom left of the interface. Additionally, beneath the list of sinks is a collapsed-by-default template list, which allows you to quickly setup some common sink parameters on targeted sinks.

The configuration options for this version of logging is as follows:

Option	Description
Target Type	Where the log messages are sent, see Target types.
Target	Additional parameter depending on Target type.
Message format	The syslog format, either rfc3164[1], rfc5425[2] or bsd.
Selector	Which produced syslog messages are sent to this sink, see Selectors

Target Types

Four target types are available for WeOS 5.15 to 5.21.1:

Type	Description	Target parameter
udp address	A specific IP address is the target	Valid IPv4 Address, e.g `192.168.0.1`
udp dhcp	The target is determined via DHCP	N/A
file internal	The target is an internal on-device file	A valid file path, relative to the systems log file folder, e.g `my/logs/mysyslog`
file external	The target is an external media device	A valid file path, relative to the target media

> ⚠ Warning
>
> No two sinks can share the same target.

> ⚠ Warning
>
> When specifying file internal/external as the target, any parent directories must already exist on the device.

Selectors

A selector in the context of syslog configuration for WeOS 5.15 to WeOS 5.21.1 is a compination of three parameters. A facility, a modifier and a severity matcher. They can be observed in the image above under the “Selector” group for each sink.

Facilities

A facility specifies the area of origin for the message, and the selector will match only messages belonging to the specified facility.

[1] https://datatracker.ietf.org/doc/html/rfc3164
[2] https://datatracker.ietf.org/doc/html/rfc5424

WeOS 5.15 to WeOS 5.21.1 provides the following facilities:

Facility	Description	Facility	Description
kern	Kernel log messages	ntp	Time-protocol events
user	User-level messages	security	Log audit, for audit trail
mail	Unused	console	Log alert
daemon	System daemons	local0	Alarm sub-system
auth	Security and Authentication messages	local1	Unused
syslog	Unused	local2	PPP
lpr	Unused	local3	Unused
news	Unused	local4	OpenVPN, IPsec
uucp	Unused	local5	Reserved, OEM customer specific
cron	Unused	local6	Unused
authpriv	Unused	local7	Unused
ftp	Unused	*	Any facilities

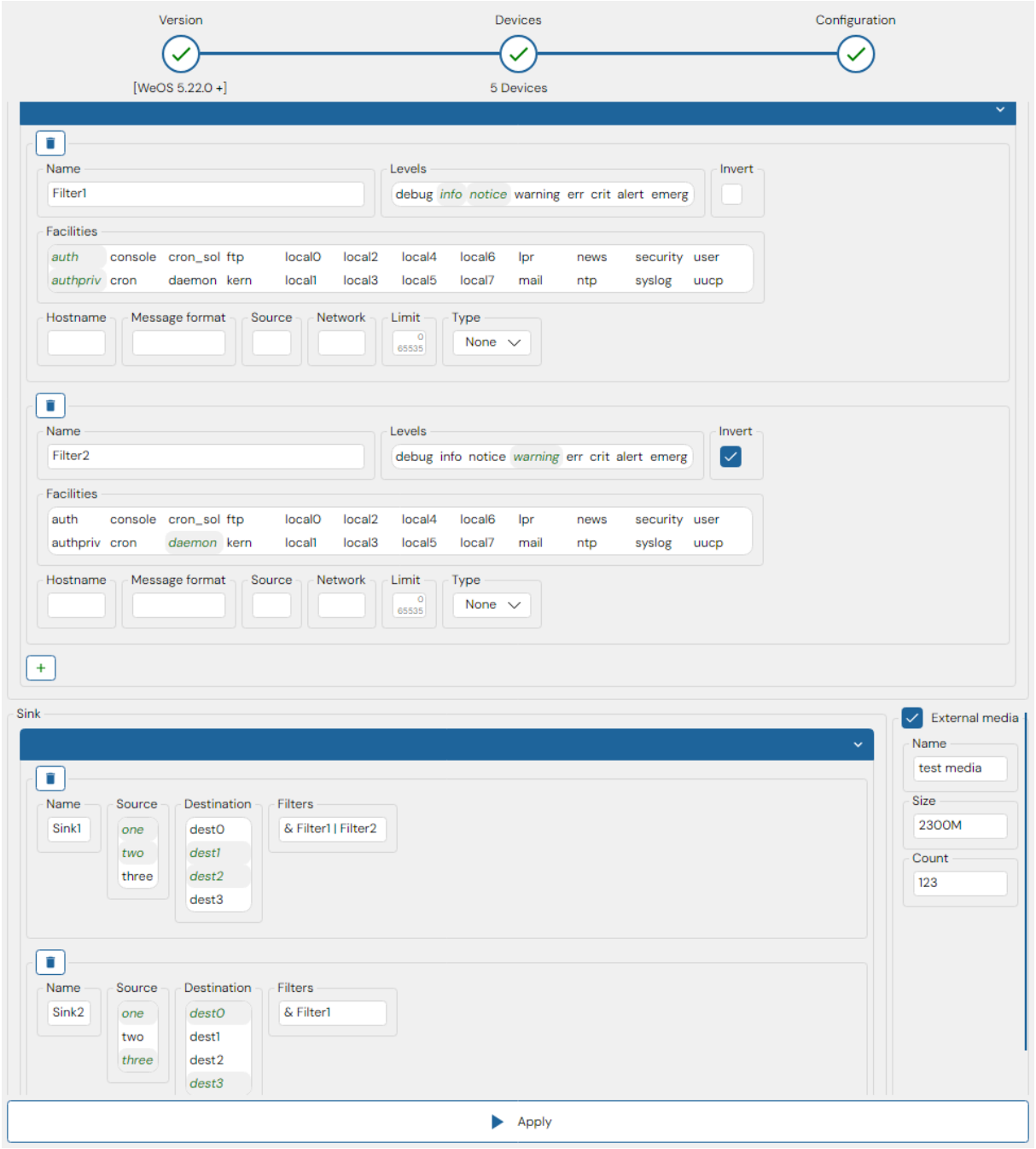
Modifier
A modifier is a simple boolean, which specifies the selector as either an inclusion filter or an exclusion filter. If specified as `Include` any syslog message that matches this selector will sent to the target sink. If specified as `Exclude` any syslog messages that matches this selector will not be sent to the target sink.

Severities
Lastly, a selector consists of a severity selector, which may be used to narrow the range of messages to only those of a certain severity by dragging the slider from the right or respectively.

The available severities are as follows:

Severity	Description
emerge	Emergency, System Level service only
alert	System level service only
crit	Critical, System level service only
err	Severe error, a daemon/service may restart
warning	Significant problems, such a failure to reach Radius servers
notice	General log messages, such as successful authentication
info	Informational, less important messages
debug	Developer/low-level debug messages

[WeOS 5.22.0 +]
From WeOS 5.22.0 and forwards, logging configuration has been expanded with several new options, with a subset depicted beneath:



Here, the interface is divided into five collapsable sections, Sinks, External Media, Sources, Destinations and Filters. In a slight deviation, let us start from the bottom:

Filters

Down at the last section you will find the filter specification, which is a list of zero-or-more filters specified for the syslog configuration. These function similarly to selectors in WeOS 5.15 to WeOS 5.21.1, but with the ability to specify a name for usage in sinks.

Destinations

In the second section from the bottom, you will find destination configuration, which sets up zero or more targets for sink composition. These share some similarity with target types in WeOS 5.15 to WeOS 5.21.1, as they dictate where generated syslog messages are delivered.

However, they contain different options, first and foremost a name. Secondly, there are now three main target types: Console, File and Remote.

Console destination
The simplest of the destinations to configure, as it requires no further details, when a sink uses this destination, messages from the sinks sources will be logged to the console.

File destination
This destination type is used to send syslog messages to a local on-device file or connected media, such as a USB or SD-card. The media name may be specified either as an external media configured on the device, or as `internal` which will consider the file name relative to the `/log/` folder.

> Note
>
> When using `internal` media, any log messages stored there will be lost upon reboot.

Additionally, you may specify a log rotation policy by setting the maximum size of each log file, the number of log files to retain, and the number of compressed log files to retain for these kinds of destinations.

Remote destinations
This destination type is used to send syslog messages to some external location, by specifying a host, port, protocol and format. Host may be specified either via host name, for DNS lookup, DHCP, a static IPv4, or a static IPv6 address. The protocol may be either UDP, TCP or TLS, where choosing TLS will allow you to specify certificates to use for trust. The formats configurable are the same as in WeOS 5.15 to WeOS 5.21.1.

> Warning
>
> When using UDP or TCP to send syslog messages, they will be sent in clear text and can be read by any interceptor.

Sources
Second from the top you will find the sources section, where the producers of syslog messages can be configured. These can either be local, on-device sources, and may include userspace or kernel messages if so, or they may be remote sources, in which case their port, protocol and format parameters must match the relevant destinations configured on other devices.

Sinks
Sinks reside at the very bottom of the interface, a sink is a named configuration that combines zero-or-more sources with zero-or-more destinations and zero-or-more filters. Any selected sources and destinations will be marked in green. And the sink depicted above as `Sink1` can be read as “receive syslog messages from the `one` and `two` sources, but not the `three` source, and if they match the filter specified by `Filter1` or `Filter2`, send them to the `dest1` and `dest2` destinations”. Of special note is perhaps the filters box, where applicable filters are combined together using either `&`, for and, or `|`, for or.

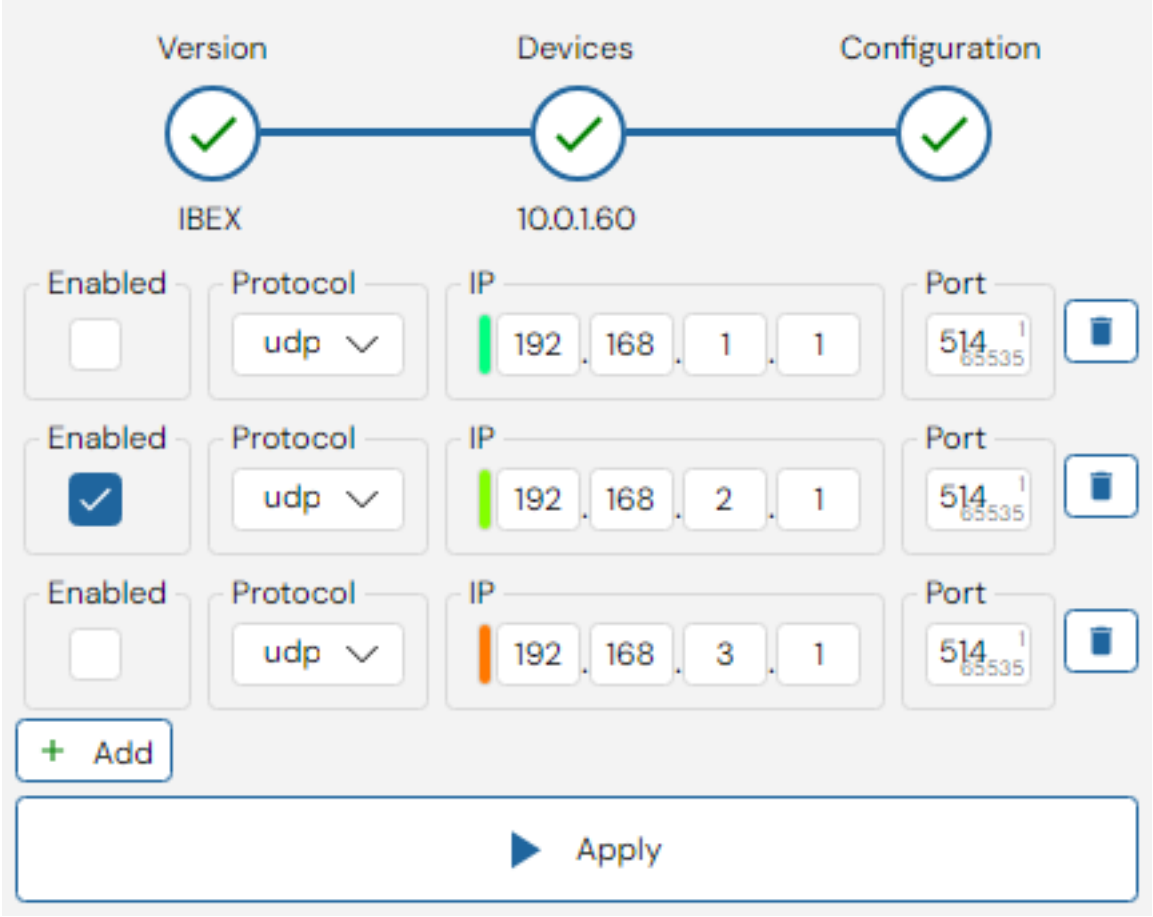
Note

Leaving the filter box empty is the equivalent to “any”

External media

To the right of Sinks you will find a small section dedicated to external media configuration, where you may configured how the log file is handled on the external media, filesize and count may be configured here.

Ibex



When configuring logging for Ibex, the interface may appear as above. Therein a simple list of sinks serve as the primary means of configuration, which can be added to via the “Add” button near the bottom of the interface.

4.4.18. OSPF

WeOS 4 WeOS 5

OSPF is a licensed task that allows for configuration of the OSPF protocol on selected devices. The current set of devices support by this task is WeOS 4 and WeOS 5 devices.

Context Menu Options

The presence of this panel adds a couple of options to the Context Menu under the top menu item “OSPF”, as specified below:

Activate default configuration

When, in the topology view, one-or-more devices that all support OSPF configuration are selected, this context menu item will appear. Clicking it will cause WeConfig to configure OSPF on all selected devices with the default configuration and all available networks exposed.

Redistribute

When, in the topology view, one-or-more devices that all support OSPF configuration are selected, this context menu will appear. It contains a number of redistribution options, clicking a specific option will cause WeConfig to configure the selected devices with the default form of redistribution for the specified route kind. The following redistrubtion options are available:

Option	Description
Connected	Redistributes directly connected routes through OSPF
Default	Redistributes default routes through OSPF
RIP	Redistributes RIP-generated routes through OSPF
Static	Redistributes static routes through OSFP

Remove

When, in the topology view, one-or-more number of devices that support OSPF configuration is selected, this context menu item will appear. Clicking it will strip the selected devices of any OSPF configuration.

Create OSPF Area

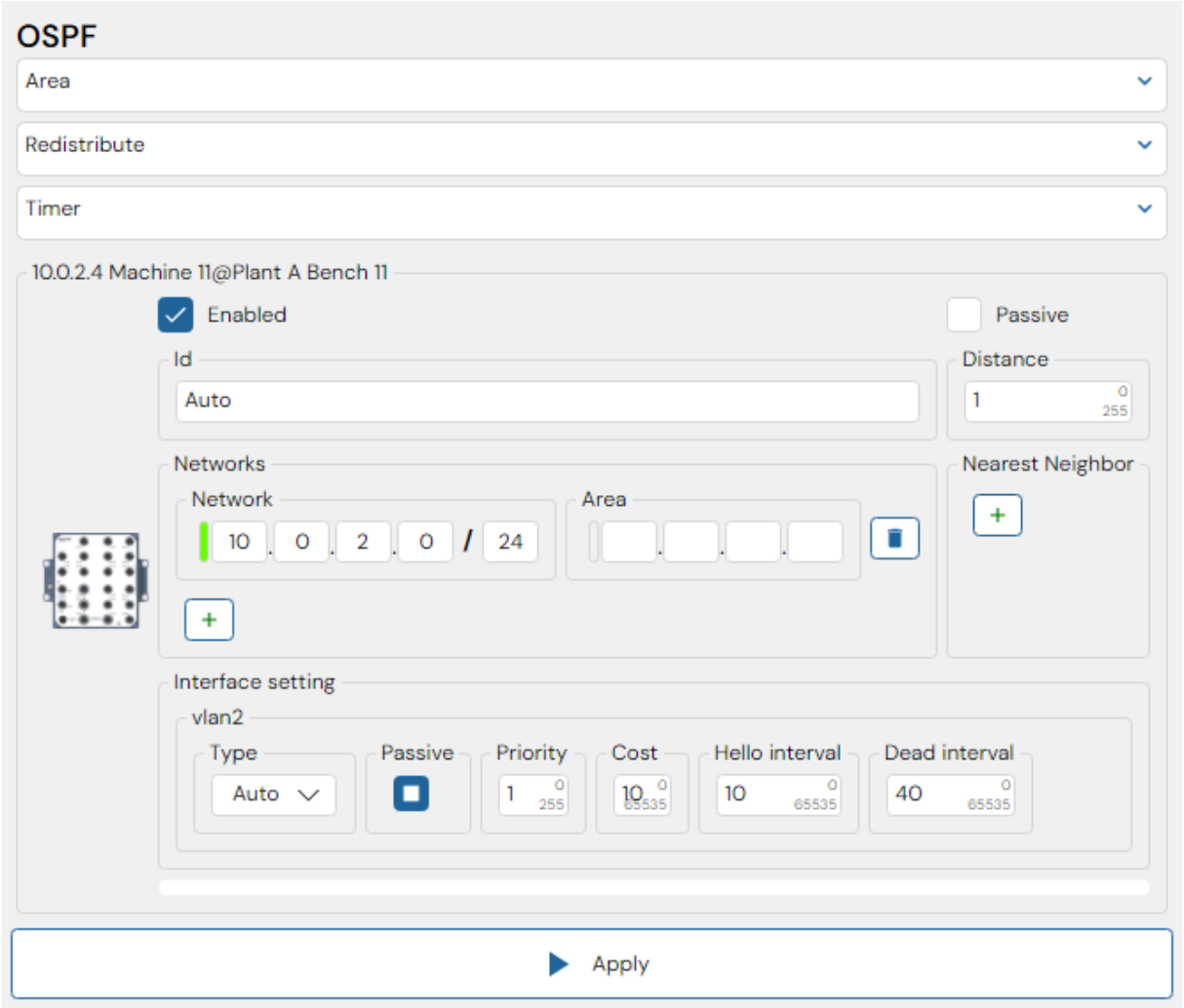
When, in the topology view, one-or-more subnets are selected that each contain at least one device supporting OSPF configuration, this context menu will appear. The menu will contain the five different varieties of OSPF area:

Type	Description
Regular	A standard OSPF area that accepts inter-area and external routes with no restrictions.
Stubby	An OSPF area that blocks external routes, replacing them with a default route.
Totally Stubby	An OSPF area that blocks both external and inter-area routes, replacing them with a default route.

Type	Description
Not so Stubby	An OSPF area that, like a Stubby area, blocks external routes from outside the area, but still allows external routes to be imported locally through an ASBR within the area.
Not so totally stubby	An OSPF area that combines the restrictions of a Totally Stubby area with the local external route import capability of a Not so Stubby area.

Selecting one of these area types will cause WeConfig to configure an OSPF area on the supported devices in the selected subnets.

Interface Components:



Depicted above is the panel interface for editting detailed OSPF settings. It is divided into four sections: Area, Redistribute, Timer and Device Settings

Area

This section allows for configuration of detected OSPF areas, it is recommended to first create theas areas using the Create OSPF Area context menu option and then edit the details of the area here as needed.

Redistribute
This section allows for detailed configuration of the protocol redistribution configured to the devices. Allowing you to set metrics and types for the respective redistributions, instead of the default ones set by the context menu option.

Timer
This section allows for detailed configuration of the timing configuration of OSPF, specifically for overiding the default configuration of the Link-state Advertisment, or LSA for short, as well as the Shortest Path First, or SPF for short, parameters.

Device Settings
This section contains a list of selected devices and their currently configured OSPF settings. There are seven categories of configuration that is contained within this section, and they are as follows:

- Enabled: Whether or not OSPF is enabled for this device.
- Passive: If all interfaces are considered Passive by default or not.
- Id: OSPF device id, Auto by default, or a valid IPv4 address otherwise.
- Distance: The administrative distance for OSPF-sourced routes.
- Networks: A list of CIDR Addresses and their associated OSPF area.
- Nearest Neighbors: A list of IPv4 specified known neighbors.
- Interface setting: A list of per-interface OSPF settings, where each interface can be configured with the following parameters:
  - Type: The OSPF network type, may be either Auto, Broadcast, Non-broadcast or Point-To-Point.
  - Passive: Override passive-interface settings for this specific interface
  - Cost: Specify the OSPF cost for the interface
  - Hello interval: Number of seconds inbetween hello packets
  - Dead interval: Number of seconds before considering neighbors down.

> **Info**
>
> If the OSPF network type is set to Non-broadcast, neighors must also be defined in order to have a useful OSPF configuration.

> **Warning**
>
> If you override the hello interval or dead interval on any given OSPF interface, ensure any neighbouring router configured with OSPF mirrors the same hello interval and dead interval configuration.

4.4.19. RIP

WeOS 4 WeOS 5

RIP is a licensed set of context menu options that allows for configuration of the RIP protocol on selected devices. The current set of devices support by this task is WeOS 4 and WeOS 5 devices.

Context Menu Options

RIP adds a couple of options to the Context Menu under the top menu item “RIP”, as specified below:

Activate default configuration

When, in the topology view, one-or-more devices that all support RIP configuration are selected, this context menu item will appear. Clicking it will cause WeConfig to configure RIP on all selected devices with the default configuration and all available networks exposed.

Redistribute

When, in the topology view, one-or-more devices that all support RIP configuration are selected, this context menu will appear. It contains a number of redistribution options, clicking a specific option will cause WeConfig to configure the selected devices with the default form of redistribution for the specified route kind. The following redistrubtion options are available:

Option	Description
Connected	Redistributes directly connected routes through RIP
Default	Redistributes default routes through RIP
OSPF	Redistributes OSPF-generated routes through RIP
Static	Redistributes static routes through OSFP

Remove

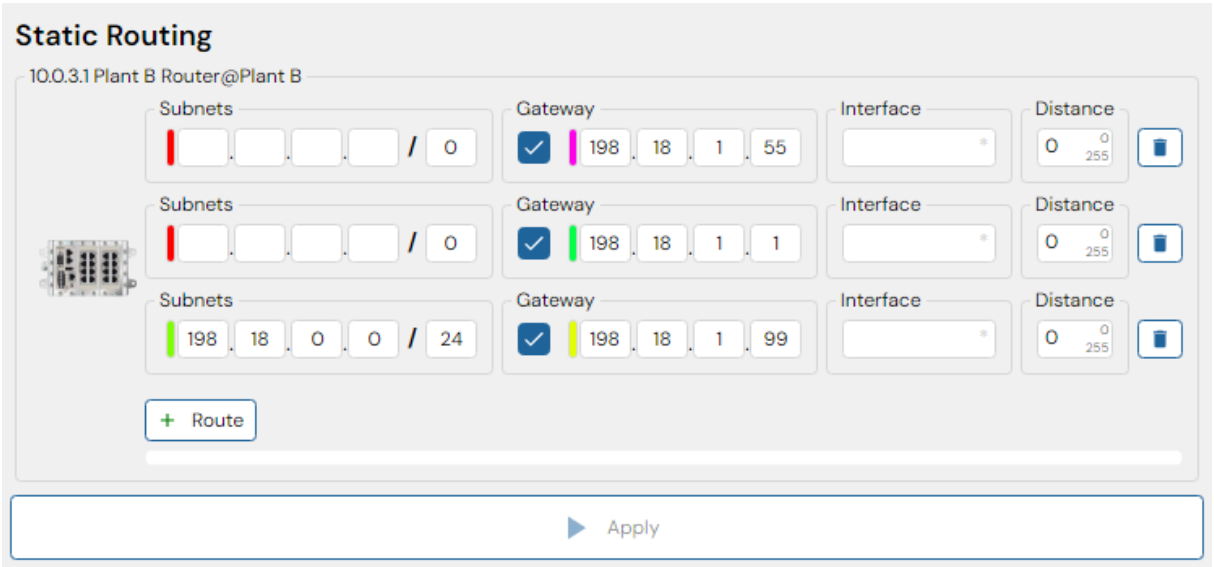
When, in the topology view, one-or-more number of devices that support RIP configuration is selected, this context menu item will appear. Clicking it will strip the selected devices of any RIP configuration.

4.4.20. Static routing

WeOS 4 WeOS 5

Static routing is a licensed task that allows for configuration of static routes on selected devices. The current set of devices support by this task is WeOS 4 and WeOS 5 devices.

Interface components



The user interface for this panel is a fairly straight-forward list of devices, each containing a list of configured routes. These routes may be added, deleted or modified. A route in this panel consists of the following parameters:

Parameter	Description
Subnets	The subnet address space targeted by the route in CIDR Notation
Gateway	The gateway address for the next hop along the route, an IPv4 address
Interface	Select a network interface from the list provided by the device. * means any interface, so only the gateway is used to determine the next hop.
Distance	The distance weight of the route, a value between 0-255

Context Menu Options

Additionally, the presence of this panel adds a couple of options to the Context Menu under the top menu item “Static Routing”, as specified below:

Route To...

When, in the topology view, a subnet A is selected that has at least 1 adjacent subnet and contains at least one device supporting static route configuration, this context menu will appear. The menu will contain a list of valid adjacent subnets, and clicking one of these specified subnets B in the context menu will cause WeConfig to set up static routes going from subnet A to subnet B.

Remove
When, in the topology view, one-or-more number of devices that support static route configuration is selected, this context menu item will appear. Clicking it will strip the selected devices of any configured static routes.

Create Routes
When, in the topology view, two or more subnets are selected that each contain at least one device supporting static route configuration, this context menu will appear. Clicking this context menu will attempt to set up all the static routes that traverse from the and to each pair of subnets in the selected set of subnets.

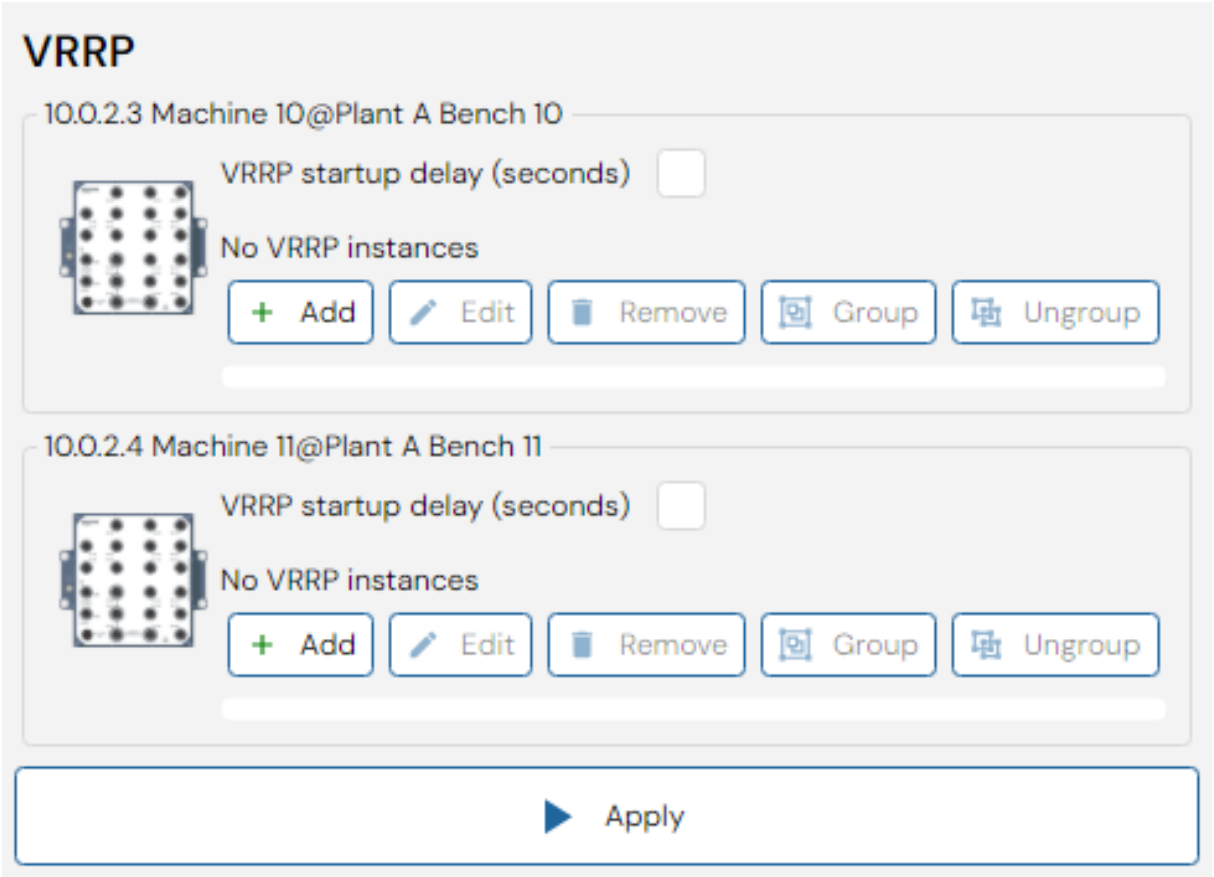
4.4.21. Virtual Router Redundancy Protocol (VRRP)

WeOS 4 WeOS 5

Virtual Router Redundancy Protocol (VRRP) is a licensed task that allows for configuration of VRRP on selected devices. VRRP is a standard protocol that is used to provide automatic assignment of available routers to its participating hosts. The current set of devices support by this task is WeOS 4 and WeOS 5 devices.

Read more about VRRP in the WeOS documentation[1].

Interface components



Add or Edit VRRP Instance

These are the common parameters for both WeOS4 and WeOS5:

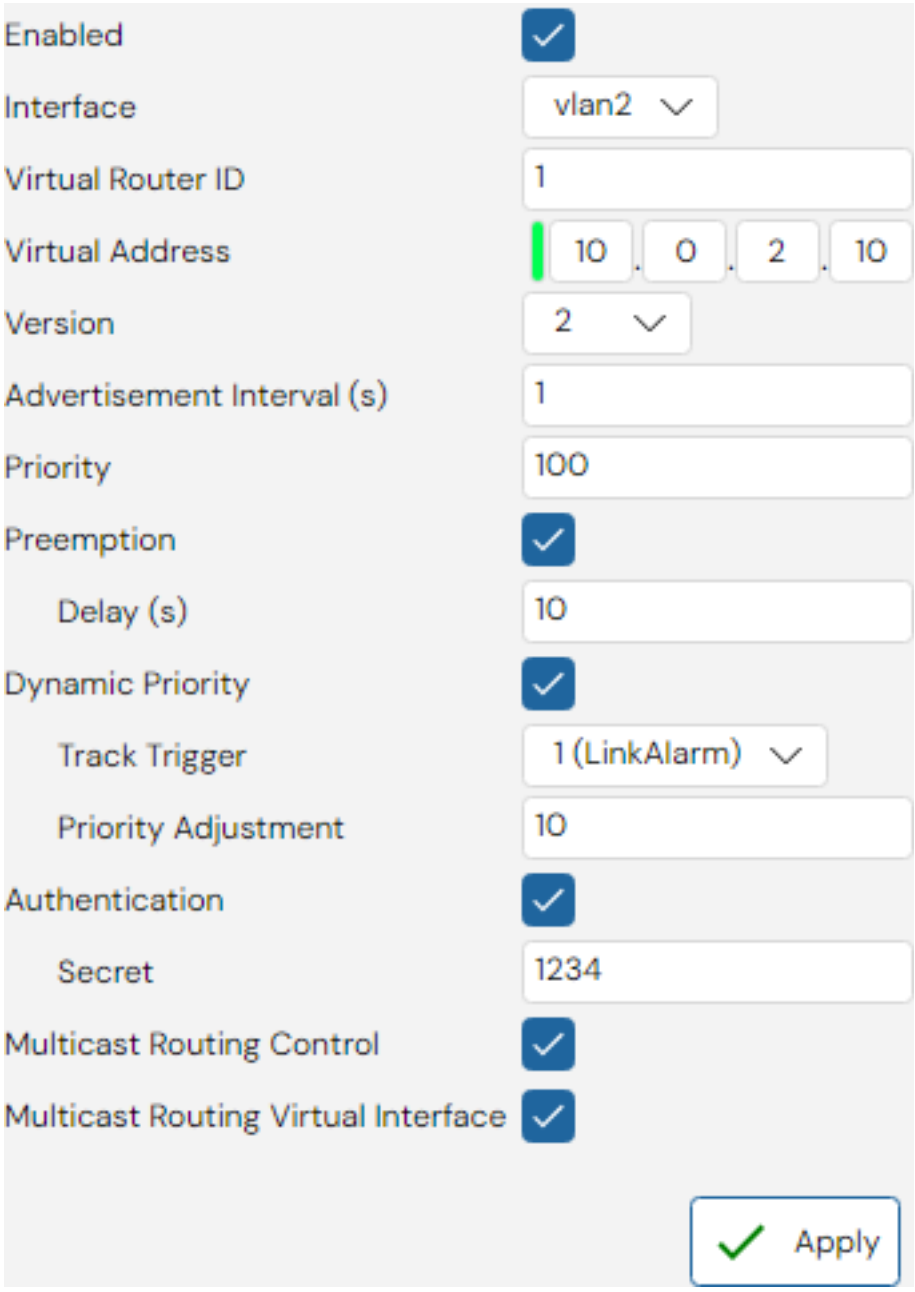
Parameter	Description	Allowed Values
Enabled	Should this VRRP instance be enabled?	
Interface	The interface on which to listen for VRRP information and act as gateway	

[1] https://docs.westermo.com/weos/weos-5/IP/VRRP.html

Parameter	Description	Allowed Values
Virtual Router ID	A unique ID common to those routers that will provide redundancy.	1-255
Virtual Address	A virtual address that the routers will use when providing the gateway support. The address should be in the same IP subnet as the regular IP address assigned to the interface	
Version	VRRP version to use (v2 or v3). Changing this will change what options are available.	
Advertisement Interval	The interval in seconds how often a VRRP advertisement message will be sent out.	v2: 1-255; v3 (WeOS4): 0.1-40; v3 (WeOS5): 0.01-40
Priority	A number used for election of current gateway. A higher number means a higher chance to become elected. If two routers has the same priority in an election, the router with the highest IP address will win. The value 255 should be used if (and only if) the router is also the owner of the virtual IP address.	1-255
Preemption	Preempt existing router of lower priority, with optional delay.	0-1000
Dynamic Priority	The alarm trigger selected will, if triggered, add the priority adjustment value to the router priority. Can not be used if the VRRP instance is part of a VRRP Group.	

VRRP Startup Delay is available on WeOS 5 and WeOS 4.35 or later. It delays starting all VRRP instances to allow interfaces or other services to become ready.

WeOS4

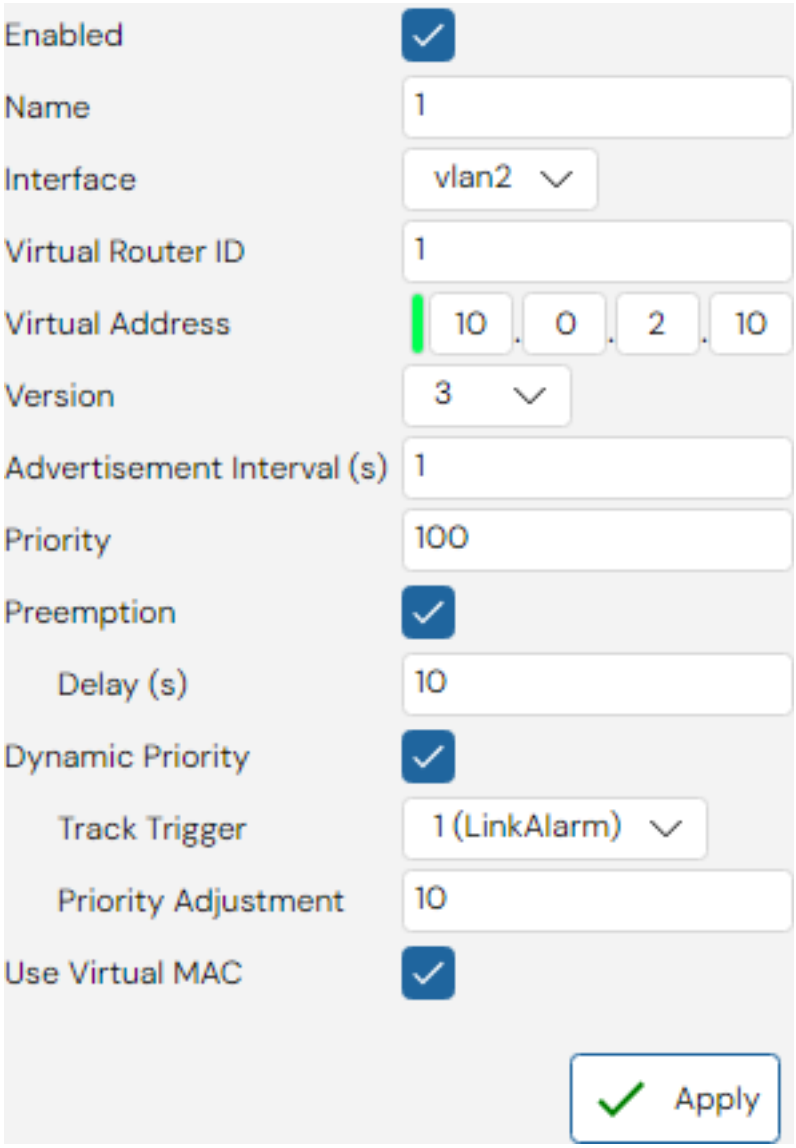


These parameters are only available for WeOS4 devices:

Parameter	Description	Allowed Values
Authentication	Simple clear-text authentication is supported for VRRP version 2.	4-7 characters. ASCII characters 33-126, except #
Multicast Routing Control	Disable multicast routing when in BACKUP state. Only one VRRP instance per interface may be configured for controlling multicast routing.	

Parameter	Description	Allowed Values
Multicast Routing Virtual Interface	Disable multicast routing for virtual VRRP interface in BACKUP state	

WeOS5



These parameters are only available for WeOS5 devices:

Parameter	Description	Allowed Values
Name	Unique Name of the VRRP instance.	Alpha-numerical (no spaces, no special characters)
Use Virtual MAC	Create and/or use a virtual interface for VRRP. Uses the VRRP MAC address.	

Grouping

VRRP instances can be grouped together to allow for synchronized failover. Select two instances (by Ctrl-clicking) and then click the `Group` button to create a new group.)

To ungroup, select one or more members of a group and then click the `Ungroup` button.

Context Menu Option

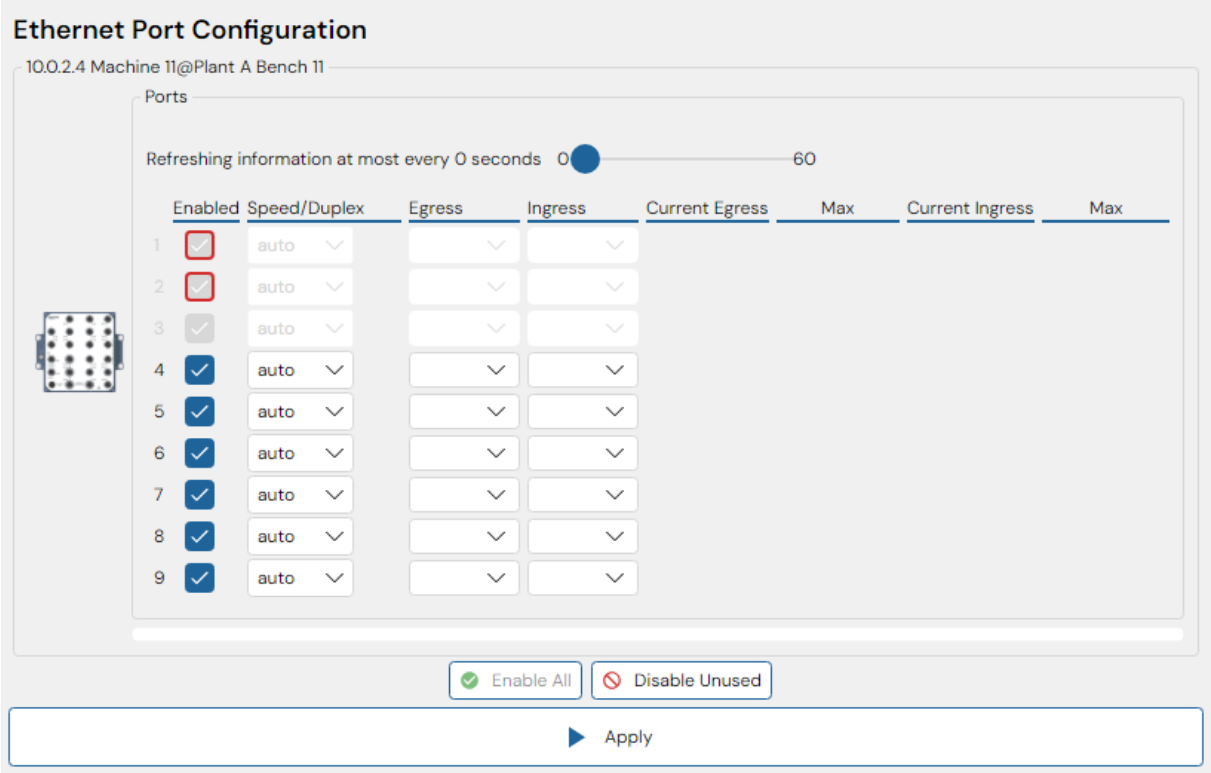
It is also possible to quickly setup VRRP from the Context Menu by selecting two routers and then selecting `Create Virtual Router`

4.4.22. Ethernet

WeOS 4 WeOS 5

Ethernet is a task that allows for configuration of Ethernet ports enabled/disabled status, speed/duplex as well as Egress and Ingress limitations.

Interface components



The user interface of this panel consist of a list of selected devices, where each device contains a list of ports. Additionally, above the port list of each device is a slider that will allow you to determine how often the data in the Current/Max Egress and Current/Max Ingress columnms is collected. The list of ports contains multiple columns, as follows:

Enabled
The first column indicates / allows you to configure whether the port is enabled or not.

> **Info**
>
> The button at the bottom of the interface with the content “Disable Unused” will disable all ports that are not currently electrically up.

> **Info**
>
> The button at the bottom of the interface with the content “Enable All” will enable all ports on all selected devices.

Speed/Duplex
On supported devices, this column will contain a combobox that allows you to selected the desired speed/duplex.

> **Note**
>
> For WeOS 5 devices several speed/duplex can be selected. To select Auto negotiation, just select this option in the same combo box.

Egress
This column allows you to specify an Egress limit from a prepared list of supported limits offered by the device.

Ingress
This column allows you to specify an Ingress limit from a prepared list of supported limits offered by the device. Additionally, selecting a limit here will show four additional checkboxes as seen in the image above. These are detailed in the table below:

Checkbox	Description
All	All Traffic
BC	MAC Broadcast `FF:FF:FF:FF:FF:FF`
MC	MAC Multicast `(2n+1):**:**:**:**:**`
U-Uni	MAC unicast `(2n):**:**:**:**:**`

Current/Max Egress & Ingress
Finally, the last four columns in the list are dedicated to a live observation of the data flowing through the ports. This data is collected at a rate determined by the slider above the port list, and operates by retrieving the number of bytes flowing through each port from each device, and calculating the average over the polling time. It also keeps track of the maximum observation recorded. This data acquistion will run for as long as the panel is open, and will cease if the slider is dragged down to 0.

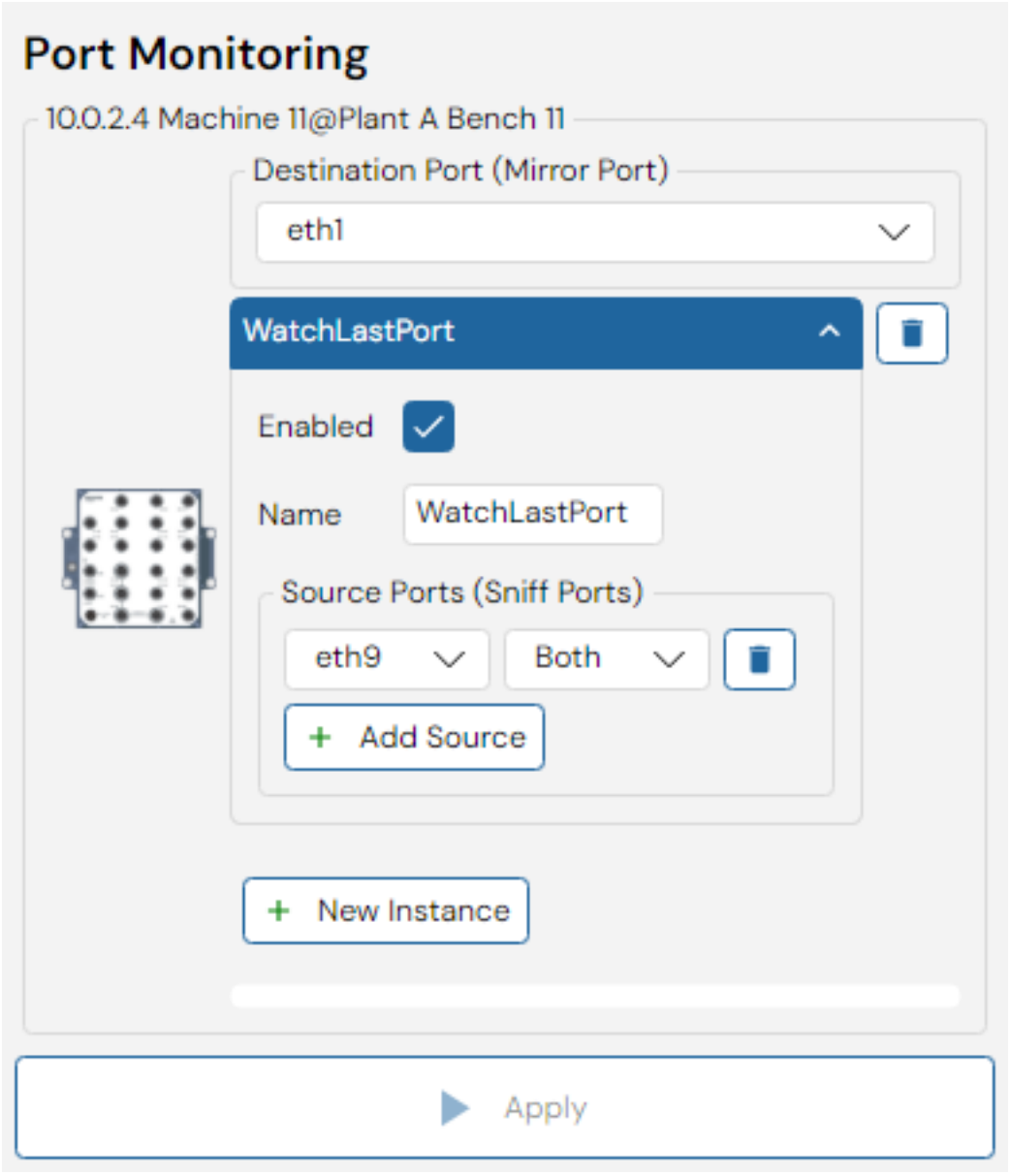
4.4.23. Port Monitoring

WeOS 4 WeOS 5

Port Monitoring is a task that allows for configuration of Port monitoring or mirroring; the act of observing traffic on a specified source port, or multiple source ports, and duplicating the traffic to another provided destination port.

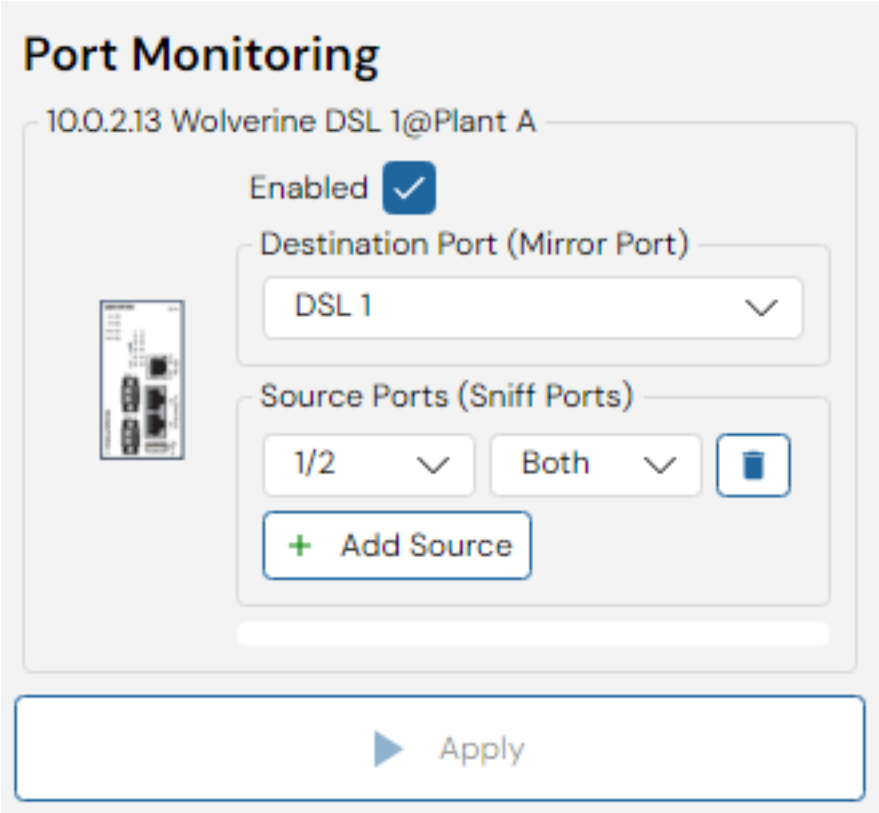
Interface components

WeOS 5



On WeOS 5, you can select one destination port (The Mirror Port; the port to which the traffic will be duplicated) and one or more instances. Each instance can be enabled or disabled, can be given a name, and can have one or more source ports (The Sniff Ports; the ports whose traffic will be duplicated to the Mirror Port) associated with it.

WeOS 4



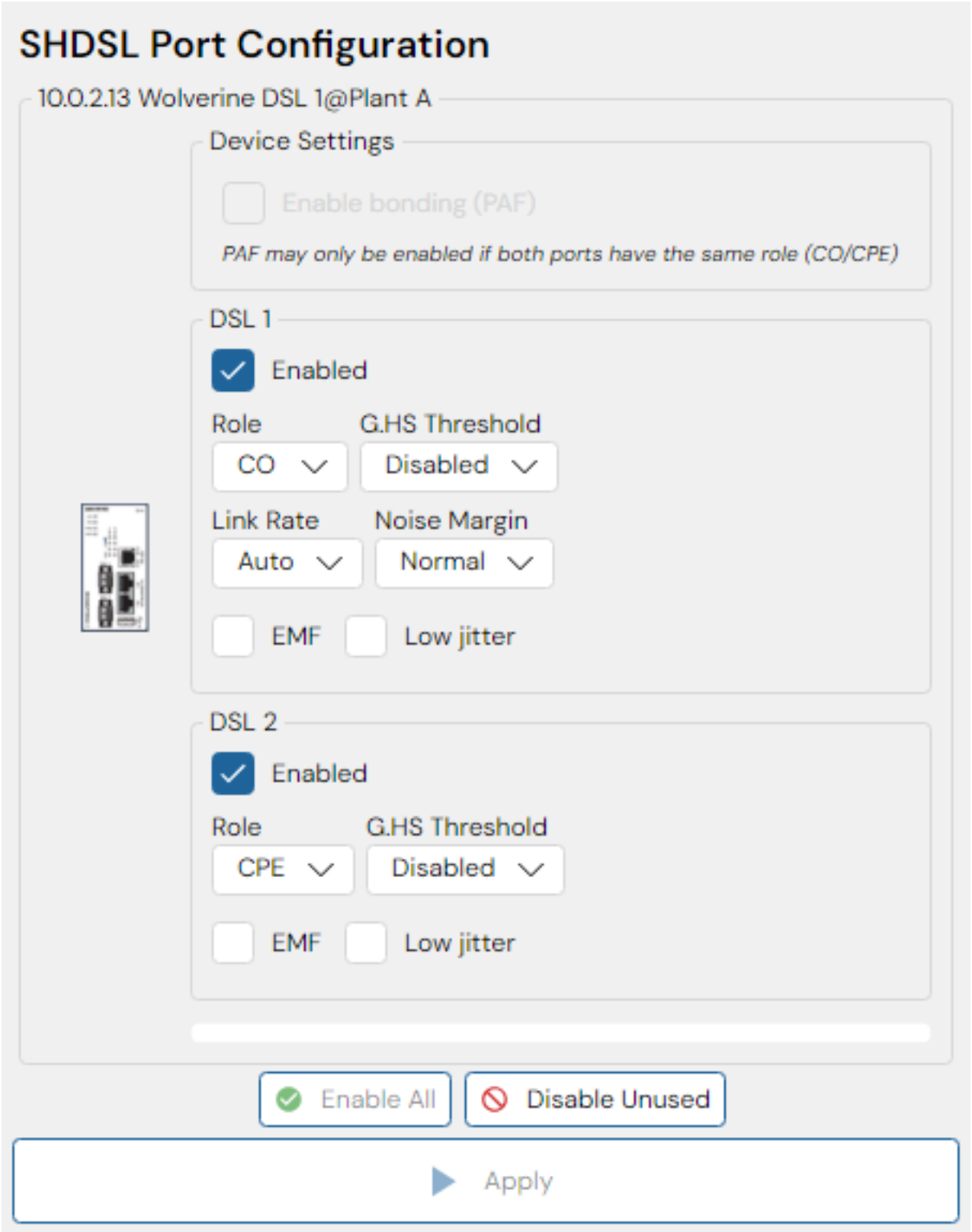
On WeOS 4, you only have one instance, which can be enabled or disabled. You can also select one destination port (The Mirror Port; the port to which the traffic will be duplicated) and one or more source ports (The Sniff Ports; the ports whose traffic will be duplicated to the Mirror Port) associated with it.

4.4.24. SHDSL

WeOS 4 WeOS 5

SHDSL is a task that allows for configuration of SHDSL port details pertaining to certain devices. If a selected devices does not have SHDSL ports, or does not support SHDSL port configuration, it will not appear in this panel.

Interface components



For each port, select Role (CO/CPE). When applicable, select G.HS threshold, link rate, EMF (emergency freeze), noise margin and low jitter. It will also be possible to select Pass. When applicable, it will be possible to select PAF (SHDSL bonding).

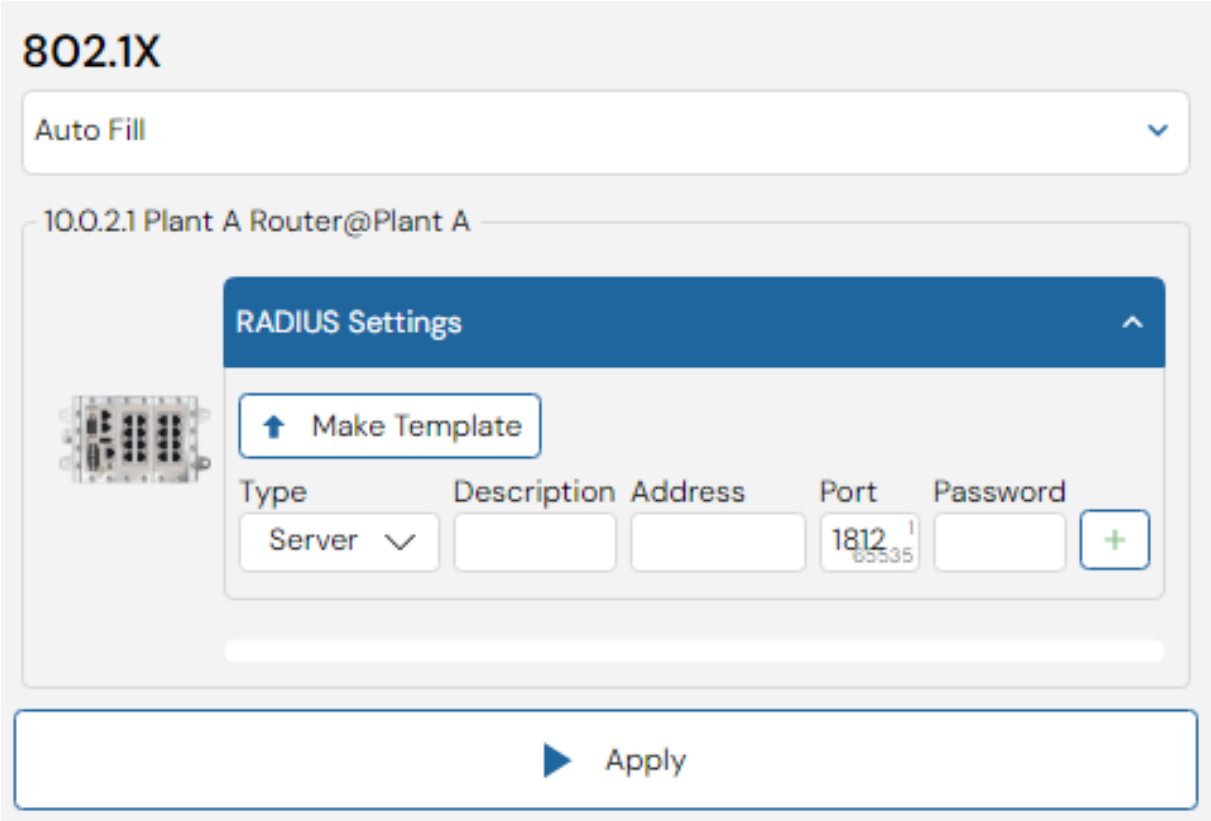
To ensure that a device is not configured so it is unreachable, WeConfig will detect if port pairs have incompatible configurations. This will only work if all connected SHDSL devices are added to the configuration panel.

4.4.25. 802.1X

WeOS 4 WeOS 5

802.1X is a task that allows for configuration of port authentication against RADIUS servers.

Interface components



For each device and VLAN that should be protected by 802.1X, click the desired Enabled checkbox. If any port on any device and VLAN should be excluded from 802.1X authentication, then click the desired port's checkbox in the Excluded ports area.

If this configuration is from scratch, consider using the RADIUS Settings Template feature, which allows for the configure of RADIUS settings in one place, and then propagate those settings to all devices added to the list with the Fill button.

To propagate the RADIUS settings from one device onto all devices, select the "master device's" RADIUS settings and click the Make template button. Now the template area has the same settings as the "master device". Then, click Fill to propagate to all devices. To add a RADIUS server, select Server in the Type combo box. Add a description, address (IP or DNS name), and service password. Click the button with a plus sign on it, and the entry will be added to the table above the input fields.

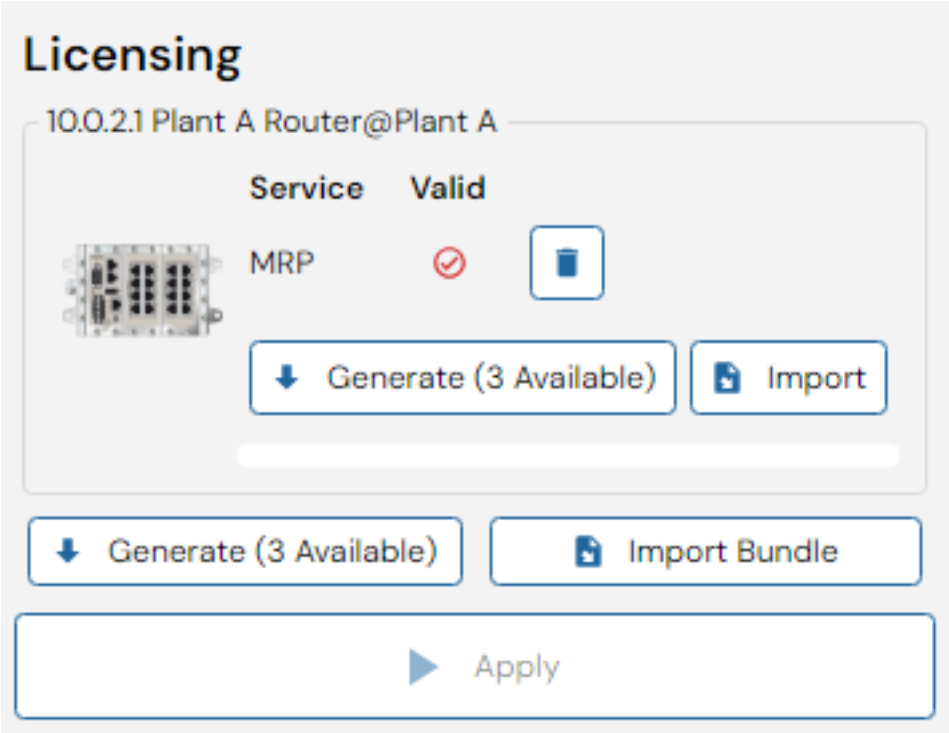
To add a RADIUS server group, first create one or more server entries. Then select Server group in the Type combo box and add a description. To link server entries to this group, type in the descriptions of the entries in the Server members text box, separated by a comma. Click the + button and the entry will be added to the table above the input fields. To select an entry in the RADIUS server/groups table as the entry to use for 802.1X authentication, click the checkbox on the correct row.

4.4.26. Licensing

WeOS 4

Licensing is a task that allows for importing or removing MRP licenses from WeOS devices.

Interface Components



Licenses can be managed for the selected devices, either separately by clicking the “Import” button or as a bundle by clicking “Import bundle”. A bundle contains licenses for multiple devices.

Additionally, licenses can be created through the self-service licensing feature by clicking the “Generate” button.

> **Note**
>
> Only WeOS 4.23 or above is supported.

MRP Self-Service Licensing

WeConfig can obtain MRP (Media Redundancy Protocol) licenses for devices directly through the Westermo web service, removing the need to manually download and import license files.

When initiating an MRP license request, WeConfig identifies the device by its serial number and article number, then contacts the Westermo licensing server to retrieve a valid license. A quota system controls how many licenses can be issued under a given license, with the remaining quota being displayed in the interface.

> **Info**
>
> An active internet connection is required for generating MRP licenses, as WeConfig communicates with the Westermo web service to obtain licenses.
>
> However, once a license has been generated, it is stored locally on the machine running WeConfig, as well as in the project file. This allows for offline installation of the license on devices, as WeConfig can apply the locally cached license file to the device without needing to contact the web service again.

Error Conditions
The following errors may occur during the licensing process:

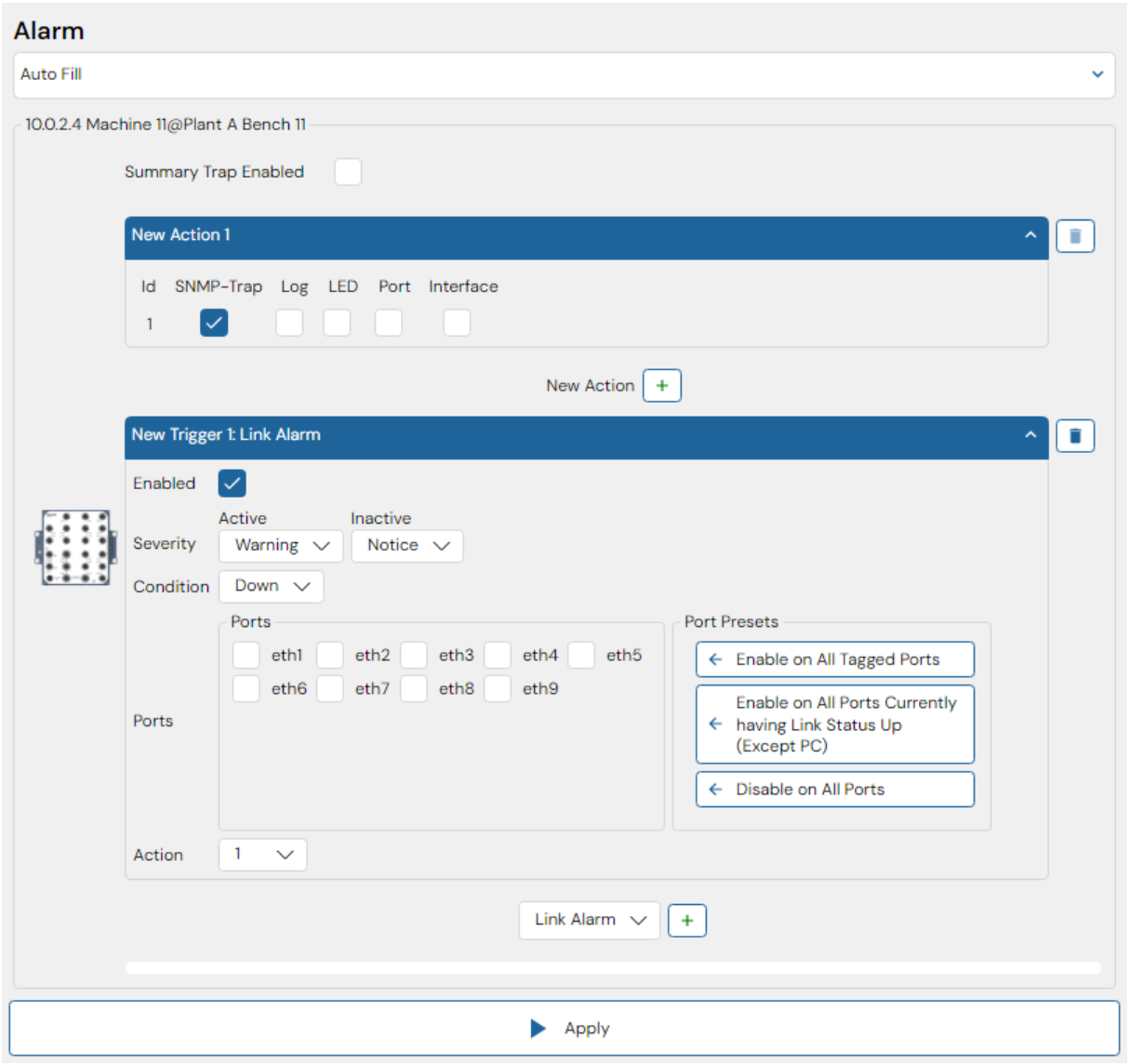
Error	Description
Serial number not found	WeConfig could not identify the serial number of the selected device.
Article number not found	WeConfig could not identify the article number of the selected device.
No local license	No cached MRP license was found on the device to derive a license from.
Insufficient quota	The quota for licenses under the current license has been exhausted.
No valid license returned	The web service did not return a valid license for what was presumed to be a valid request. Contact Westermo support.

4.4.27. Alarm

WeOS 4 WeOS 5

Alarm is a task that allows for configuration of alarms on selected devices. The current set of devices support by this task is WeOS 4 and WeOS 5 devices.

Interface components



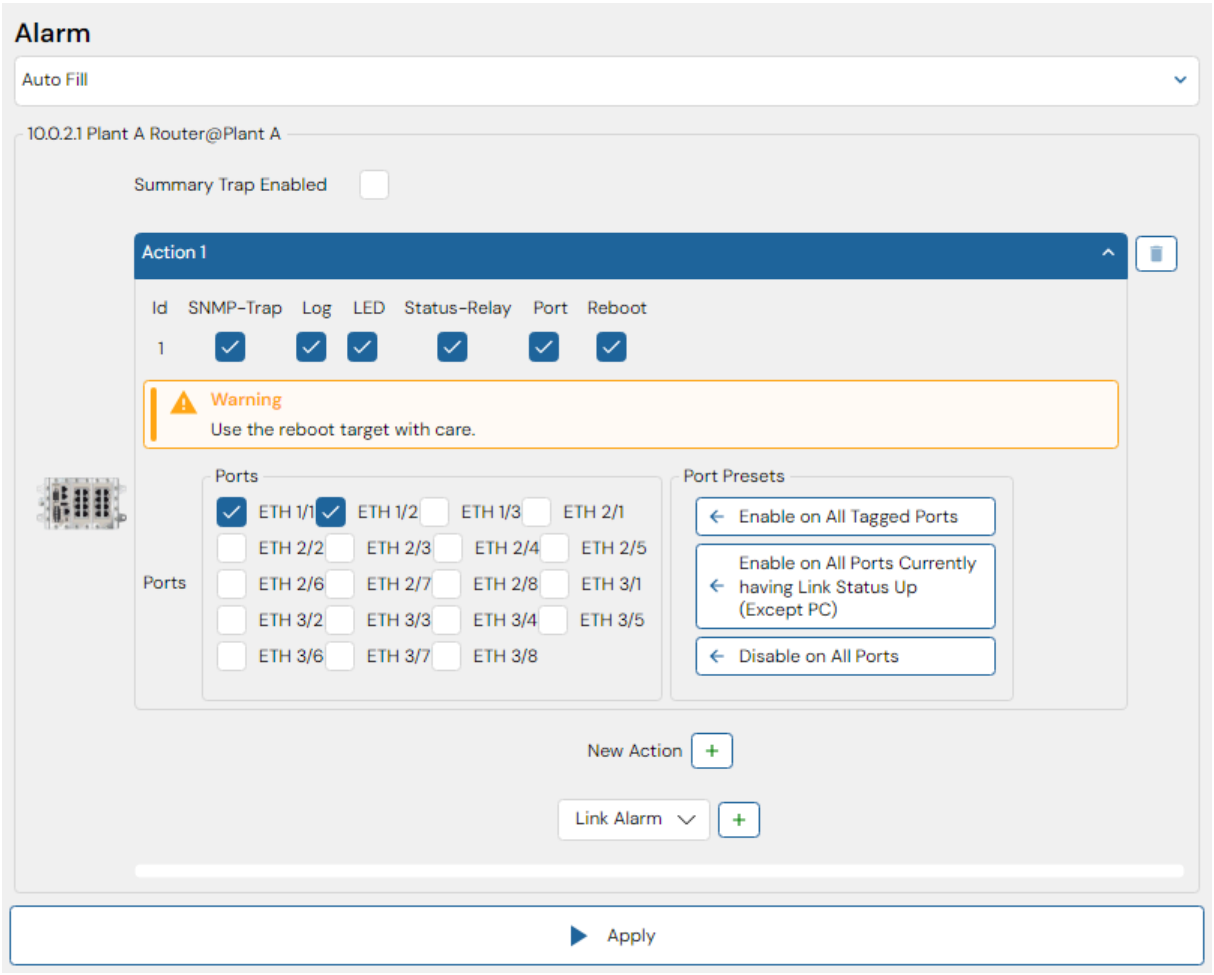
Each device can configure multiple actions and triggers, as well as enable or disable summary trap.

Actions

When an alarm event occurs, the operator can be informed in a number of ways. This is defined by a specific alarm action that is associated with any given alarm trigger. The alarm action in turn specify one or more alarm targets that will be used to notify the operator.

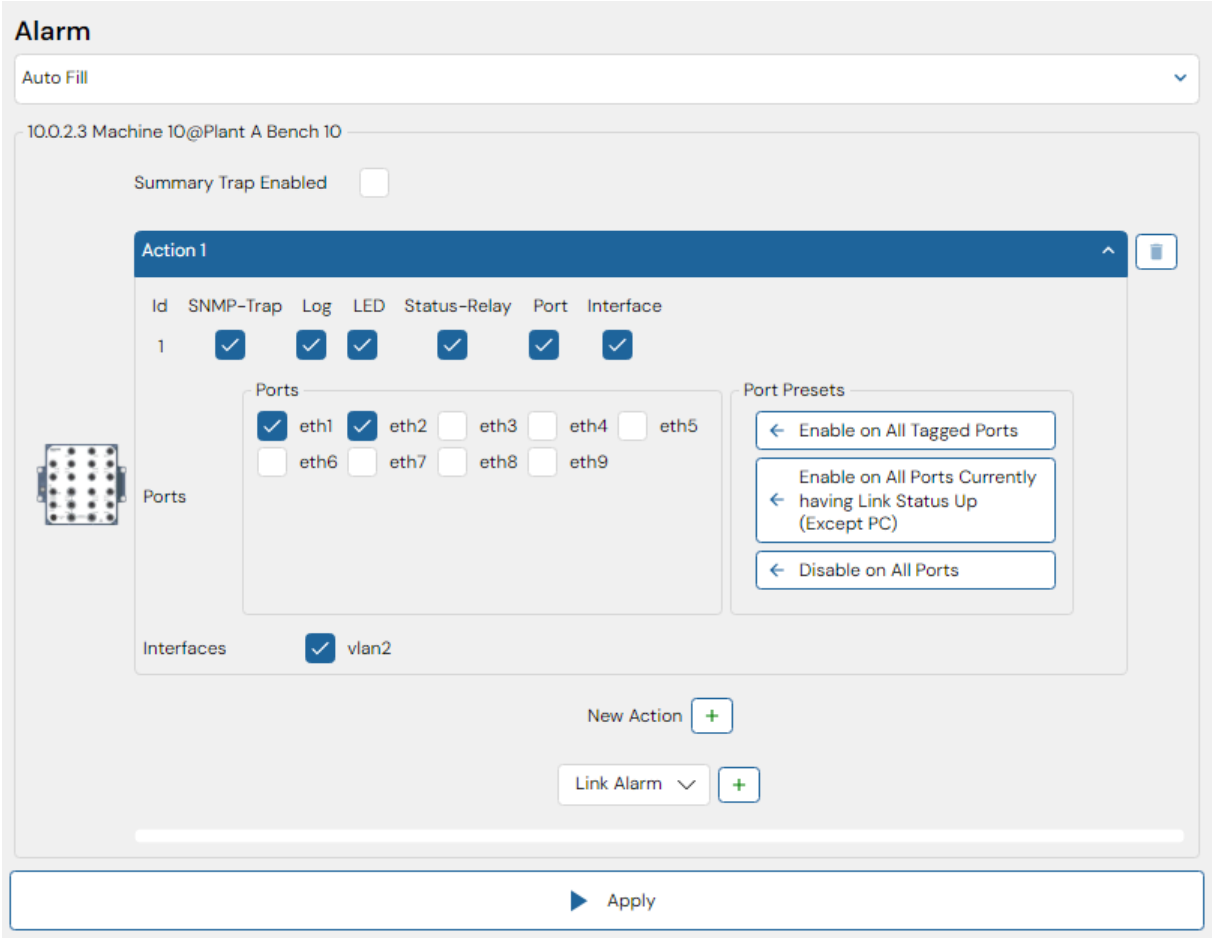
The available targets vary depending on the WeOS version

WeOS 4



Target	Description
SNMP-Trap	Generate an SNMP trap
Log	Log status change to syslog
LED	Set ON/Status LED
Status-Relay	Control the status relay (digital out)
Port	Control the admin status of a port.
Reboot	Reboot the unit. USE WITH CAUTION!

WeOS 5



Target	Description
SNMP-Trap	Generate an SNMP trap
Log	Log status change to syslog
LED	Set ON/Status LED
Status-Relay	Control the status relay (digital out)
Port	Control the admin status of a port.
Interface	Control the admin status of an interface.

Port selection

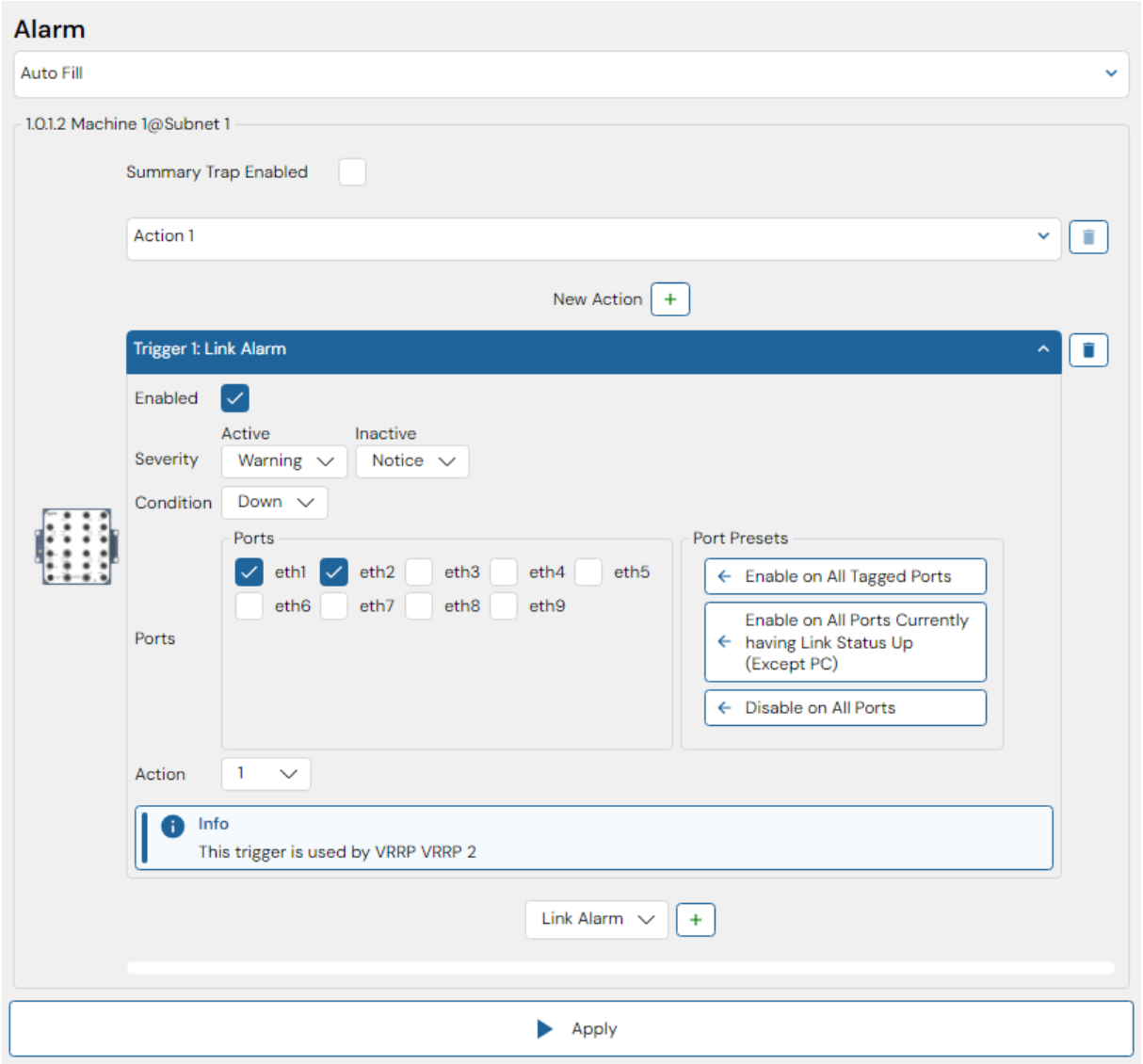
There are some presets available to make it easier to select the correct ports. The presets are as follows:

> **Info**
>
> Only presets that are applicable to the current device and configuration will be shown.

Preset	Description
Enable on All Tagged Ports	Selects all ports that are tagged in at least one VLAN.

Preset	Description
Enable on All Untagged Ports	Selects all ports that are untagged in at least one VLAN.
Enable on All FRNT Ports	Selects all ports that are configured to be part of at least one FRNT ring.
Enable on All MRP Ports	Selects all ports that are configured to be part of at least one MRP ring.
Enable on All RSTP Ports (Non Admin Edge)	Selects all ports that are configured for RSTP meshing, except those configured as an admin edge.
Enable on All Ports Currently having Link Status Up (Except PC)	Selects all ports that are currently up, except if that port connects directly to the configuring PC.
Enable on All RiCo uplink ports	Selects all ports that are configured to be RiCo uplink ports.
Enable on All Dual-Homing uplink ports	Selects all ports that are configured to be Dual-Homing ports.
Disable on all Ports	Deselects all ports

Triggers



An alarm trigger will monitor a provided alarm source and define the conditions for when specific alarm events occur, in other words, when the trigger becomes active (alarm situation) or inactive (normal operation).

Exactly which triggers are available depends on the device. The list you see in WeConfig is based on the list of supported trigger types the devices themselves provide.

These are all the triggers that WeConfig currently supports:

Common parameters
All triggers have a set of common parameters:

Parameter	Description
Enabled	Enables or disables the trigger.
Active Severity	Specify the severity level of active alarm events detected by this trigger.
Inactive Severity	Specify the severity level of inactive alarm events detected by this trigger.
Action	The action to be invoked when this trigger detects an alarm event.

Severity levels are defined as follows:

Severity Level	Description
Emergency	System is unusable
Alert	Action must be taken
Critical	Critical conditions
Error	Error conditions
Warning	Warning conditions
Notice	Significant condition
Info	Informational message
Debug	Debug-level message
None	No severity
*	

Address Conflict

> **Note**
>
> Only one address conflict trigger per interface can be configured.

Parameter	Description
Interface	A valid vlan interface
Interval	Preferred sample interval in seconds (1-300)

Ddm Bias Current

Parameter	Description
Condition	Define whether the `HIGH` or `LOW` trigger state should be considered the alarm state, while the other is considered the normal state.
Threshold	Alarm events will be generated when reaching the rising threshold on the way up, and the falling threshold on the way down.
Ports	The ddm port(s) to trigger alarm on

Ddm Rx Power

Parameter	Description
Condition	Define whether the `HIGH` or `LOW` trigger state should be considered the alarm state, while the other is considered the normal state.
Threshold	Alarm events will be generated when reaching the rising threshold on the way up, and the falling threshold on the way down.
Ports	The ddm port(s) to trigger alarm on

Ddm Temperature

Parameter	Description
Condition	Define whether the `HIGH` or `LOW` trigger state should be considered the alarm state, while the other is considered the normal state.
Threshold	Alarm events will be generated when reaching the rising threshold on the way up, and the falling threshold on the way down.
Ports	The ddm port(s) to trigger alarm on

Ddm Tx Power

Parameter	Description
Condition	Define whether the `HIGH` or `LOW` trigger state should be considered the alarm state, while the other is considered the normal state.
Threshold	Alarm events will be generated when reaching the rising threshold on the way up, and the falling threshold on the way down.
Ports	The ddm port(s) to trigger alarm on

Ddm Voltage

Parameter	Description
Condition	Define whether the `HIGH` or `LOW` trigger state should be considered the alarm state, while the other is considered the normal state.
Threshold	Alarm events will be generated when reaching the rising threshold on the way up, and the falling threshold on the way down.
Ports	The ddm port(s) to trigger alarm on

DigIn

Triggers on the presence of input voltage/current on the Digital-In pins of the Digital I/O connector.

Parameter	Description
Condition	Define whether the `HIGH` or `LOW` trigger state should be considered the alarm state, while the other is considered the normal state.
Sensors	Which sensor to trigger alarm on. If the device only have one sensor, then that sensor will be selected and this option is not visible

Frnt

Triggers when an FRNT ring is broken or healed (intact).

> ⚠ Warning
>
> As of WeOS 5.21: This is a deprecated trigger, use the Ring trigger instead!

> **Note**
>
> Only an FRNT focal point can determine the ring status with certainty.

Parameter	Description
Condition	Define whether the `UP` or `DOWN` trigger state should be considered the alarm state, while the other is considered the normal state.
Ring Id	Which FRNT ring id to trigger alarm on

Lff

Parameter	Description
Condition	Define whether the `UP` or `DOWN` trigger state should be considered the alarm state, while the other is considered the normal state.
Ports	The dsl port(s) to trigger alarm on

Link Alarm

Triggers when the status of a link changes (goes up or down).

Parameter	Description
Condition	Define whether the `UP` or `DOWN` trigger state should be considered the alarm state, while the other is considered the normal state.
Ports	The port(s) to trigger alarm on

Link Health

Available on supported WeOS 5 devices, this trigger reports link health events for the selected ports.

Parameter	Description
Ports	The port(s) to monitor

Media

Triggers if the external media device (USB or SD card) is available or unavailable.

Parameter	Description
Condition	Define whether the `Available` or `Unavailable` trigger state should be considered the alarm state, while the other is considered the normal state.
Interval	The interval (seconds) between checks

Media Threshold

Triggers if the used space on the media device (USB or SD card) exceeds the configured threshold.

Parameter	Description
Interval	The interval (seconds) between checks

Parameter	Description
Threshold	The threshold to trigger the alarm at. In Percent, Kilobytes, Megabytes, or Gigabytes

MicroLok

Parameter	Description
Condition	Define whether the `HIGH` or `LOW` trigger state should be considered the alarm state, while the other is considered the normal state.

Mrp

Parameter	Description
Condition	Define whether the `HIGH` or `LOW` trigger state should be considered the alarm state, while the other is considered the normal state.

Ping

A connectivity checker, sends an ICMP ping in a configurable interval.

Parameter	Description
Condition	Define whether the `UP` or `DOWN` trigger state should be considered the alarm state, while the other is considered the normal state.
Peer	The host to test the connectivity against.
Outbound	Use to force ping to use a specific interface, IP Address (if supported), or Default Gateway
Interval	The interval (seconds) between ICMP Pings
Trigger level	The number of ICMP ping that should be lost (or received) to determine if a host is unreachable (or reachable).
Initial state	The inital alarm state of the trigger

PoE

Triggers when the total power usage raises above (or falls below) a configured threshold.

Parameter	Description
Condition	Define whether the `HIGH` or `LOW` trigger state should be considered the alarm state, while the other is considered the normal state.
Utilization Threshold	Alarm events will be generated when reaching the rising threshold on the way up, and the falling threshold on the way down.

Power

Triggers when the temperature rises above (or falls below) some defined threshold.

Parameter	Description
Condition	Define whether the `UP` or `DOWN` trigger state should be considered the alarm state, while the other is considered the normal state.

Parameter	Description
Sensors	Which sensor to trigger alarm on. If the device only have one sensor, then that sensor will be selected and this option is not visible

Profinet

Parameter	Description
Condition	Define whether the `HIGH` or `LOW` trigger state should be considered the alarm state, while the other is considered the normal state.

RiCh

Parameter	Description
Condition	Define whether the `HIGH` or `LOW` trigger state should be considered the alarm state, while the other is considered the normal state.

RiCo Uplink

Triggers when the total power usage raises above (or falls below) a configured threshold.

Parameter	Description
Condition	Define whether the `UP` or `DOWN` trigger state should be considered the alarm state, while the other is considered the normal state.
Ports	The port(s) to trigger alarm on

Ring

Triggers when an ring is open or closed (intact)

Parameter	Description
Condition	Define whether the `UP` or `DOWN` trigger state should be considered the alarm state, while the other is considered the normal state.
Protocol	Which ring protocol to configure the trigger for (FRNT, MRP, HSR, PRP)
Ring Id	Which ring id to trigger alarm on

RioCgi

Parameter	Description
Condition	Define whether the `HIGH` or `LOW` trigger state should be considered the alarm state, while the other is considered the normal state.
Pattern	The pattern to match
Query	The query for trigger
Initial state	The inital alarm state of the trigger

Snr Margin

Parameter	Description
Condition	Define whether the `HIGH` or `LOW` trigger state should be considered the alarm state, while the other is considered the normal state.

Parameter	Description
Snr Threshold	Alarm events will be generated when reaching the rising threshold on the way up, and the falling threshold on the way down.
Ports	The dsl port(s) to trigger alarm on

Storm Detect

This is the existing `storm-detect` trigger; it is distinct from the WeOS 5 Storm Detection trigger below.

Parameter	Description
Threshold	Alarm events will be generated when reaching the rising threshold on the way up, and the falling threshold on the way down.
Persitent	Optional timeout (in seconds)
Traffic type(s)	Broadcast, Multicast, and/or Unicast
Ports	The port(s) to trigger alarm on

Storm Detection

Available on supported WeOS 5 devices, this trigger reports storm-detection events for the selected ports.

Parameter	Description
Ports	The port(s) to monitor

Temperature

Triggers when the temperature rises above (or falls below) some defined threshold.

Parameter	Description
Condition	Define whether the `HIGH` or `LOW` trigger state should be considered the alarm state, while the other is considered the normal state.
Temperature Threshold	Alarm events will be generated when reaching the rising threshold on the way up, and the falling threshold on the way down.
Sensors	Which sensor to trigger alarm on. If the device only have one sensor, then that sensor will be selected and this option is not visible

Timer

Goes off at given time interval.

Parameter	Description
Timeout	At what time during the day should the trigger go off?

4.4.28. WeConnect

WeOS 5

WeConnect is a task that provisions a secure VPN tunnel between a Westermo device and the WeConnect cloud platform. The wizard guides you through precondition checks, credential entry, interface selection, and automated provisioning of SSL tunnels, firewall rules, and RIP routing.

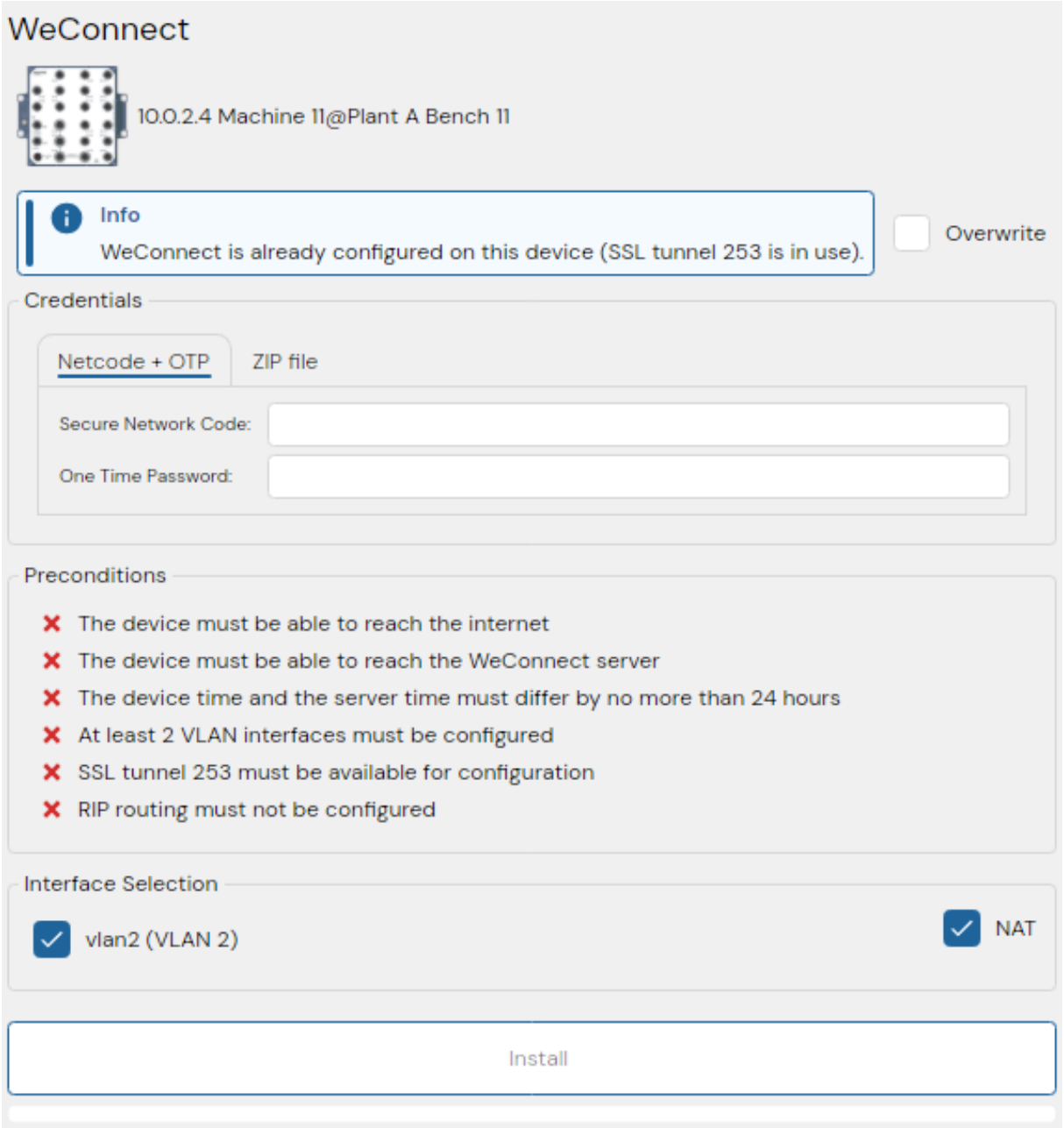
> **Info**
>
> To use WeConnect, the target device must have an active internet connection and have a DNS server capable of identifying the WeConnect server.

Preconditions
Before provisioning can begin, WeConfig automatically verifies that the device meets the following requirements:

Precondition	Description
VLAN interfaces	At least two VLAN interfaces must be configured on the device.
SSL tunnel 253	SSL tunnel 253 must be available (not already in use by another configuration).
RIP routing	RIP routing must not already be configured on the device.
Internet reachable	The device must be able to reach the internet.
WeConnect server reachable	The device must be able to reach the WeConnect server.
System clock	The device system clock must be within 24 hours of the server time.

If any precondition is not met, the wizard will indicate the issue and provisioning cannot proceed until it is resolved.

Interface Components



Credentials

WeConnect supports two credential modes for authenticating with the WeConnect service:

Mode	Description
Netcode + OTP	Enter a Secure Network Code and a One Time Password provided by Westermo.
ZIP file	Select a ZIP file containing the configuration bundle downloaded from the WeConnect portal.

Choose the appropriate mode and provide the required credentials to proceed.

Interface Selection

Select which VLAN interfaces on the device should be routed through the WeConnect tunnel. You may also configure NAT (Network Address Translation) for WeConnect traffic in this stage.

Provisioning

Once credentials and interfaces are confirmed, WeConfig automatically provisions the device. The following steps are performed:

1. Connectivity check — Verifies the device can reach the WeConnect server.
2. SSL tunnel configuration — Creates and configures SSL tunnel 253 with the provided credentials.
3. RIP routing — Configures RIP routing for the selected interfaces.
4. Firewall rules — Adds the necessary firewall rules to allow WeConnect traffic.

Progress is displayed in real time. Do not disconnect or power off the device during provisioning.

> ⚠ Warning
>
> Provisioning modifies the device's SSL tunnel, firewall, and RIP routing configuration. It is recommended that you have a backup of the device configuration before proceeding.

Managing Existing Configurations

If a device already has a WeConnect configuration, the wizard offers the following actions:

Action	Description
Reinstall	Re-provisions the device with a fresh WeConnect configuration, replacing the existing one.
Uninstall	Removes the WeConnect configuration from the device, including the SSL tunnel, firewall rules, and RIP routing entries.

Troubleshooting

The following errors may occur during WeConnect provisioning:

Error	Possible Cause	Resolution
Device not supported	WeConfig does not support WeConnect configuration on the selected device.	-
TLS certificate validation failed	The device could not validate the WeConnect server's TLS certificate.	Check the device system clock and ensure root certificates are up to date.
System clock too far off	The device clock differs from the server by more than 24 hours.	Synchronize the device clock via NTP or manual configuration.
Configuration bundle corrupt	The provided ZIP file is damaged or incomplete.	Re-download the configuration bundle from the WeConnect portal.

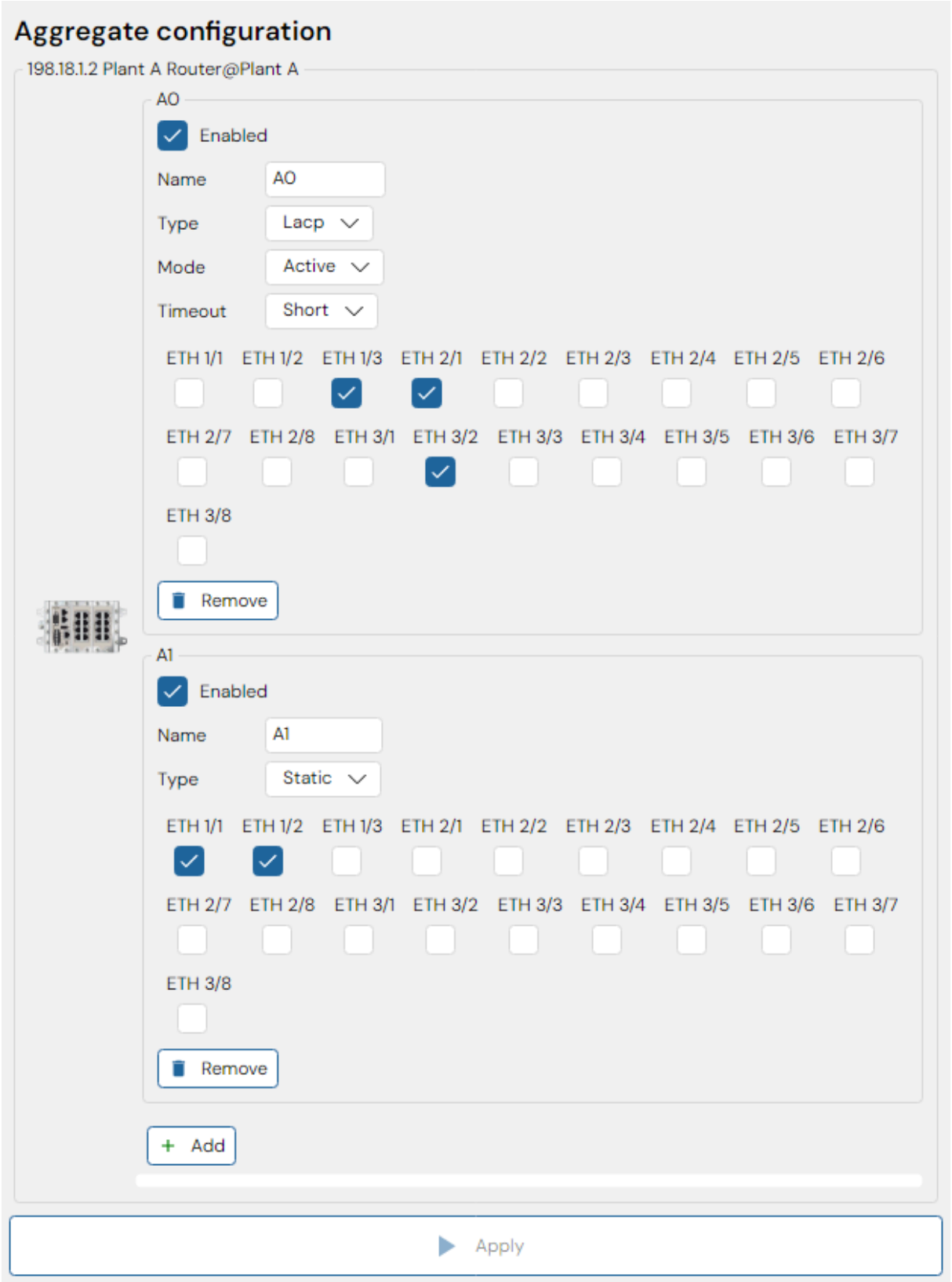
Error	Possible Cause	Resolution
Credentials rejected	The Secure Network Code or One Time Password was not accepted by the server.	Verify the credentials and ensure the OTP has not expired.
Network connectivity issue	The device cannot reach the internet or the WeConnect server.	Check the device's network configuration and upstream connectivity.
SSL tunnel import failed	The SSL tunnel configuration could not be applied to the device.	Ensure tunnel 253 is available and the device supports SSL tunnels.
RIP or firewall configuration failed	The RIP routing or firewall rules could not be applied.	Check for conflicting RIP or firewall configurations on the device.

4.4.29. Aggregates

WeOS 4 WeOS 5

Aggregates is a task that allows for configuration of static and LACP based port aggregates.

Interface components



As depicted above, the aggregate configuration interface consists of a list of devices, in this example a singular one, with a list of zero-or-more configured aggregates within. Each aggregate can be configured with a name, whether or not it's enabled, which type of aggregate it is and the ports that are included within it.

As can be seen above, when configuring LACP-type aggregates, options for configuring mode and timeout are also present.

Finally, aggregates that are no longer desired can be removed with the "Remove" button, and new aggregates can be created with the "Add" button beneath the list of each device.

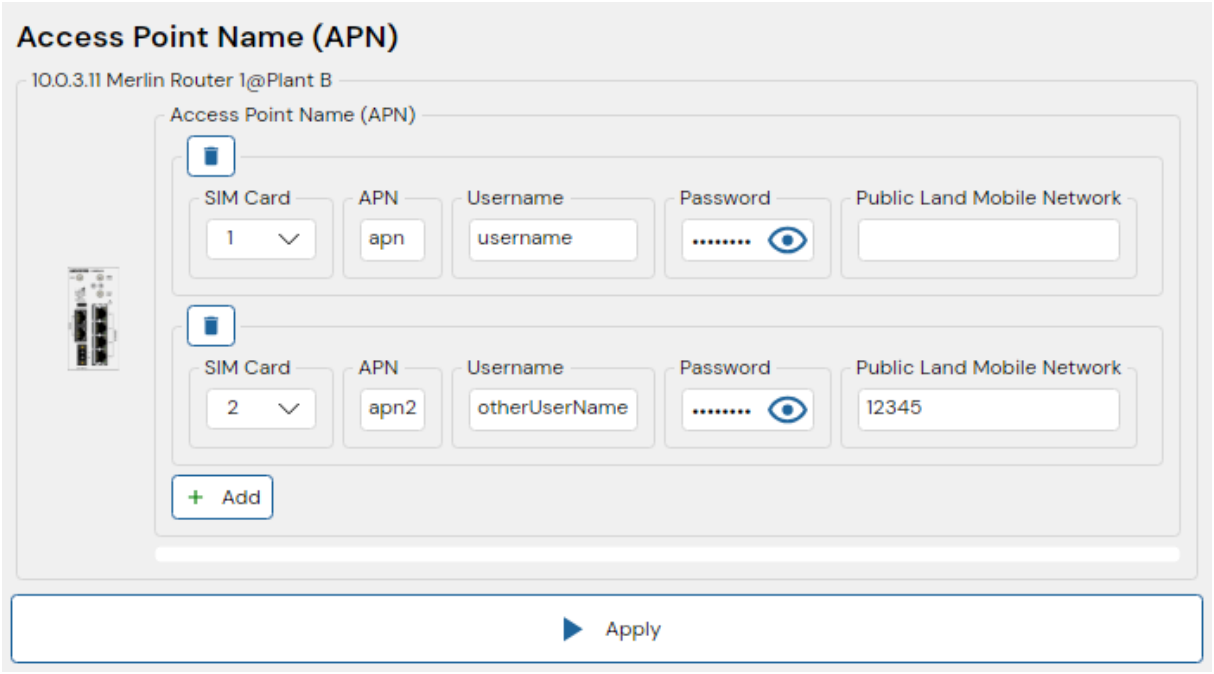
4.4.30. Access Point

Merlin

Access Point is a task that allows for configuration of Mobile Access Points, allowing wireless clients to connect to the network through it. Currently, only Merlin devices support this configuration feature.

Interface components

WeOS 5



The interface displays each selected device with its IP address and hostname. Within each device view, access point entries are presented in a grouped table where each row represents a single mobile access point configuration. If there exists the possibility of configuring more Access points on the device, the “Add” button will be enabled and clickable, otherwise, it will not be.

SIM Card
A dropdown selector that specifies which SIM card slot the access point entry is associated with. Available options correspond to the physical SIM slots on the device.

APN
The Access Point Name used to identify the packet data network the device should connect to.

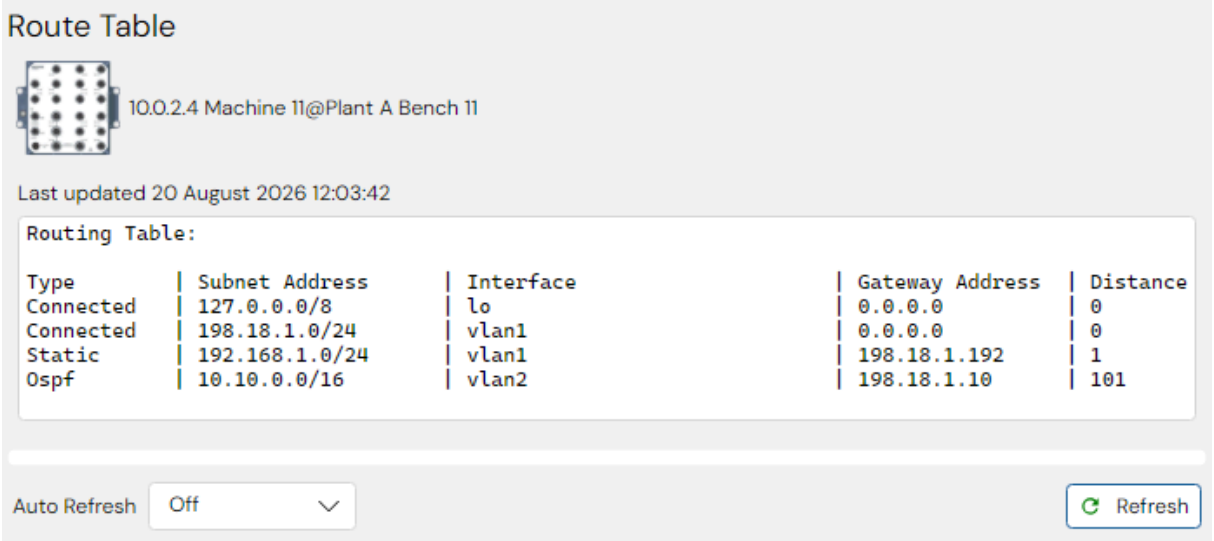
Username
The username credential required for authenticating with the mobile network, if applicable.

Password
The password credential for network authentication. The field is masked by default, with a visibility toggle button to reveal the value.

Public Land Mobile Network
An optional numeric identifier for the specific mobile network to connect to.

4.4.31. Route Table

WeOS 4 WeOS 5 Merlin



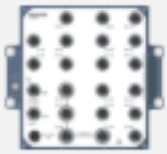
This panel shows the routing table of the selected device at the time of the last refresh. It also contains a Refresh for quick access to updating its relevant information.

Once the routing table has been fetched from the device, the user interface will show the routing table and the time when the information was received by WeConfig. The information is persisted in memory; the information is retained between device selections, and will be stored as part of the project file.

4.4.32. Properties

WeOS 4 WeOS 5 Ibex XRD Merlin other

Properties



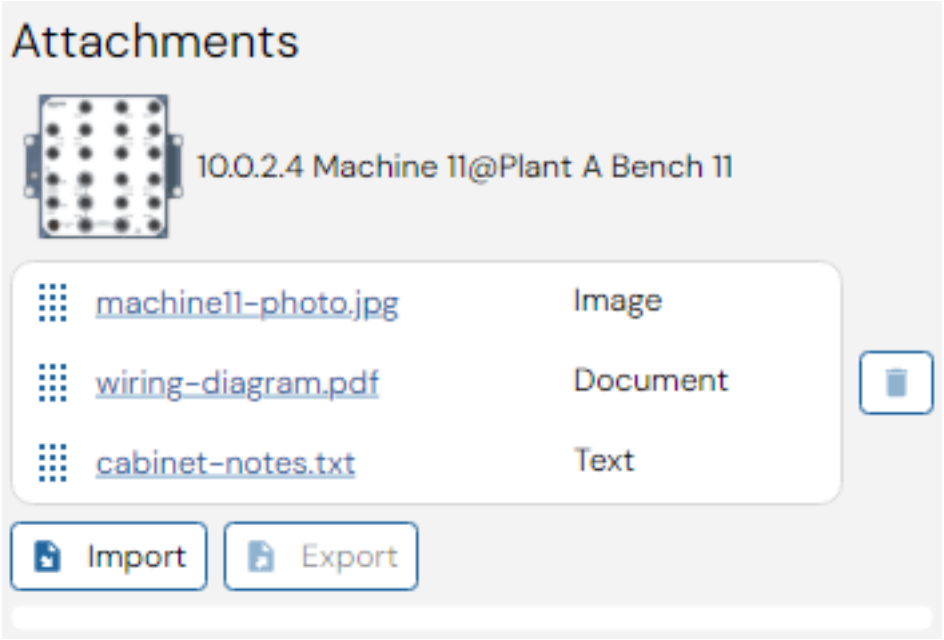
10.0.2.4 Machine 11@Plant A Bench 11

Status	
Hostname	Machine 11
Location	Plant A Bench 11
IP Address	10.0.2.4
Netmask	255.255.255.0
Default Gateway	
MAC	00:07:7C:23:C1:C0
Manufacturer	Westermo Network Technologies AB
Means of discovery	mDNS
Family	Viper
Model	Viper-120A-T4G
Firmware	5.22.0
Bootloader:	2024.03.0-1
Serial Number	1750
Host Key	Unknown
Thumbprint	thumbnail@10.0.2.4
Available Memory	81610 KB
FRNT	Member
Temperature	25 °C
Alarm	No
Power 1	High
Power 2	High
Art.No.	3641-4100

This panel contains a number of properties that WeConfig knows about the device. It is updated dynamically whenever WeConfig's knowledge of the specified properties changes. The following properties are available:

Property	Description
Status	Last reported status by WeConfig regarding this device
Hostname	System hostname of the device
Location	System location of the device
IP Address	Primary management IP address associated with the device
Netmask	The netmask associated with the primary management IP address
Default Gateway	Default gateway configured to the device, if any
MAC	MAC address associated with the device, or responding NIC, if no MAC address for the device itself is known
Manufacturer	Organization identified by the MAC address of the device / NIC
Means of Discovery	A note on how WeConfig discovered the device, typically listed either as a address plus protocol, i.e `169.254.145.82 [IPConfig]`, or just protocol, i.e as in the picture `mDNS` which, in the former case, indicates that it was discovered using the IPConfig protocol from the address 169.254.145.82.
Family	Identified device product family, if any
Model	Identified specific device model, if any
Firmware	The identified primary firmware version running on the device
Revision	The identified Hardware Revision, if any, primarily relevant for xRD devices
Backup firmware	The identified secondary firmware version on the device, if any
Bootloader	The identified bootloader version running on the device, if any
Serial Number	The serial number reported by the device
Host Key	SSH host key associated with trying to access the device, used to help detect MITM-attacks, and as a possible source of device identity
Thumbprint	TCP-connection certificate thumbprint identified, if any, used as a possible source of device identity
Available Memory	Last reported available memory on the device
Load Average	Average CPU load over the last minute, five minutes and fifteen minutes respectively
FRNT	FRNT ring status of the device, either nothing, member or focal point
Temperature	Last reported device Temperature
Uptime	Last reported device uptime.
Alarm	If the device has any outgoing alarms that WeConfig can detect
Power1	Last qualitative reporting on Power into slot 1 from the device
Power2	Last qualitative reporting on Power into slot 2 from the device
Art. No.	Reported Article number of the device

4.4.33. Attachments



This panel displays the file attachments associated with the currently last selected device as a list of links and format explanations.

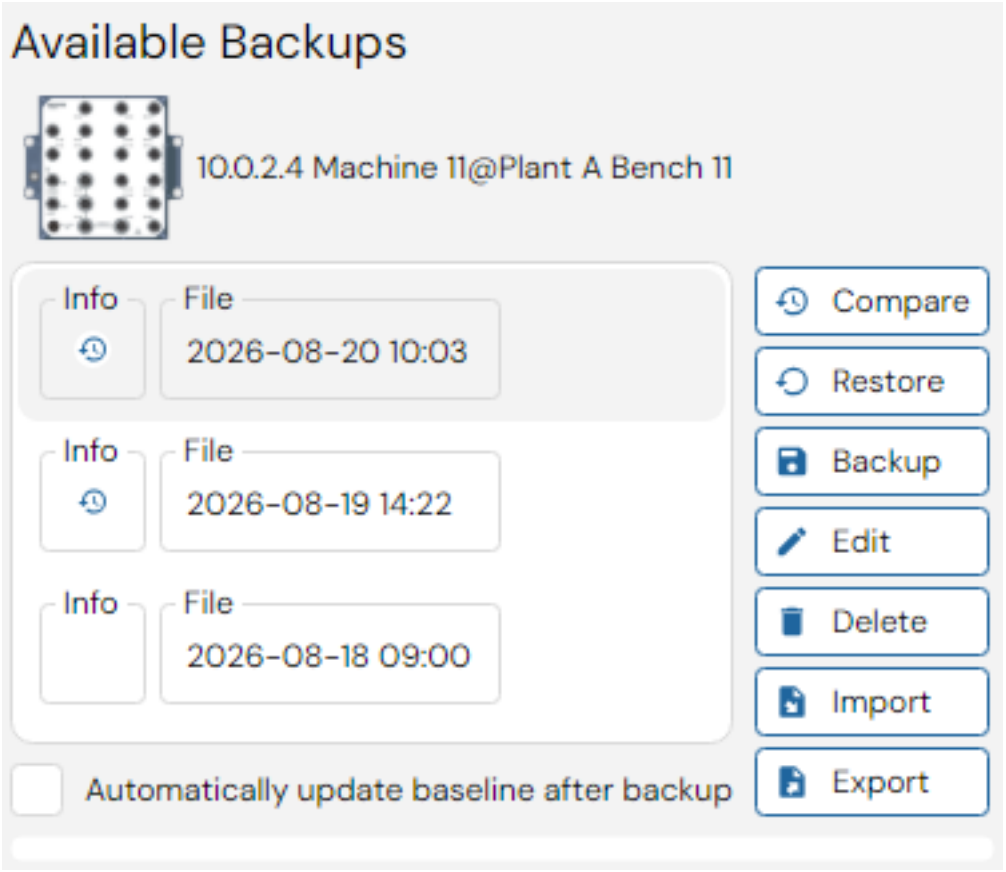
Additional attachments may be added to the device here by clicking the “Import” button, and may be exported unto the regular file system using the “Export” button with a specific attachment selected. Attachments can also be deleted by selecting them and pressing the Trashcan button to the right side of the interface.

Clicking on any of the links for any of the attachments will open the corresponding file using the running operating systems preferred method. When the process the operating system decided to open is closed, such as when the user completes editing a file, the attachment will be updated and packed back into the project.

4.4.34. Configuration Files

WeOS 4 WeOS 5 Ibex XRD Merlin

This panel displays the configuration files associated with the currently last selected device.



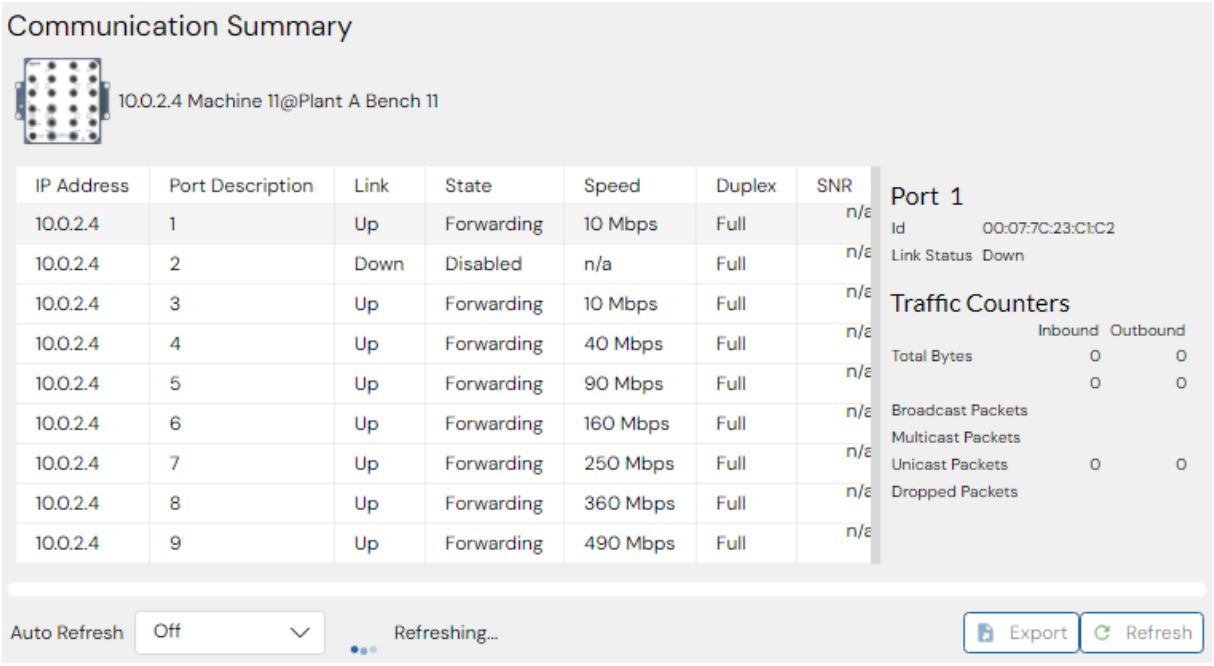
Select a device and currently available configuration file backups are listed (in local time order) in this panel. Configuration can be backed up, restored, edited, imported, exported, or deleted.

When selecting Automatically update baseline after backup the following backup will be used as new baseline.

When a listed configuration backup file differs from previous entry in the list, an icon is shown to the left of the entry. Click the icon to show actual file differences in a separate window. WeConfig uses an internal viewer that shows differences. This viewer can be changed to any other viewer via the settings.

4.4.35. Communication Summary

WeOS 4 WeOS 5 XRD Ibex



This panel lists a summary of communication information for ports on selected device.

Select a port in the list and detailed information will be available in the Communication Details section found to the right-hand side of the panel.

The communication information can be automatically updated every 5, 10, 30 or 60 seconds. Select an option in the drop-down found below the communication summary list. Click the Export button and the list is exported to a CSV file.

Communication Details

This part of the panel contains more details information regarding communcation on a specific, selected port. Such as the port MAC Id, its link status and various transmission history properties and statistics.

4.4.36. Cellular

Ibex XRD Merlin

Cellular



10.0.3.11 Merlin Router 1@Plant B

Service information

Self test	ok
Service ID	WEMO
Service name	Westermo
Primary SIM	SIM1
SIM status	ok
Modem status	online
ICCID	ICCDOM
Roaming	No
Cellular signal type	5G

Network

Registration	Singular
Provider	WEMOPRO
Cell location ID	Onboard
Cell ID	Docuphone

Packet network

Connection state	connected
Current session time	123
Total session time	234

Wireless CSD network

Connection state	connected
Sessions	3
Current session time	345
Total session time	567

Signal information

Type	5G NR
RSRQ	-3.3 db
RSRP	-44 dbm
RSSI	-51 dbm
SINR	29.3 db

Network information

wlan0

IP Address	169.254.0.0/16
APN	APN0
DNS 1	0.0.0.0
DNS 2	255.255.255.255
Gateway	127.0.0.1

LTE Information

Supported Bands	1, 2, 4, 8, 16
Active Band	2
Cell ID	123
Physical Cell ID	234
EPS Registration Status	53412

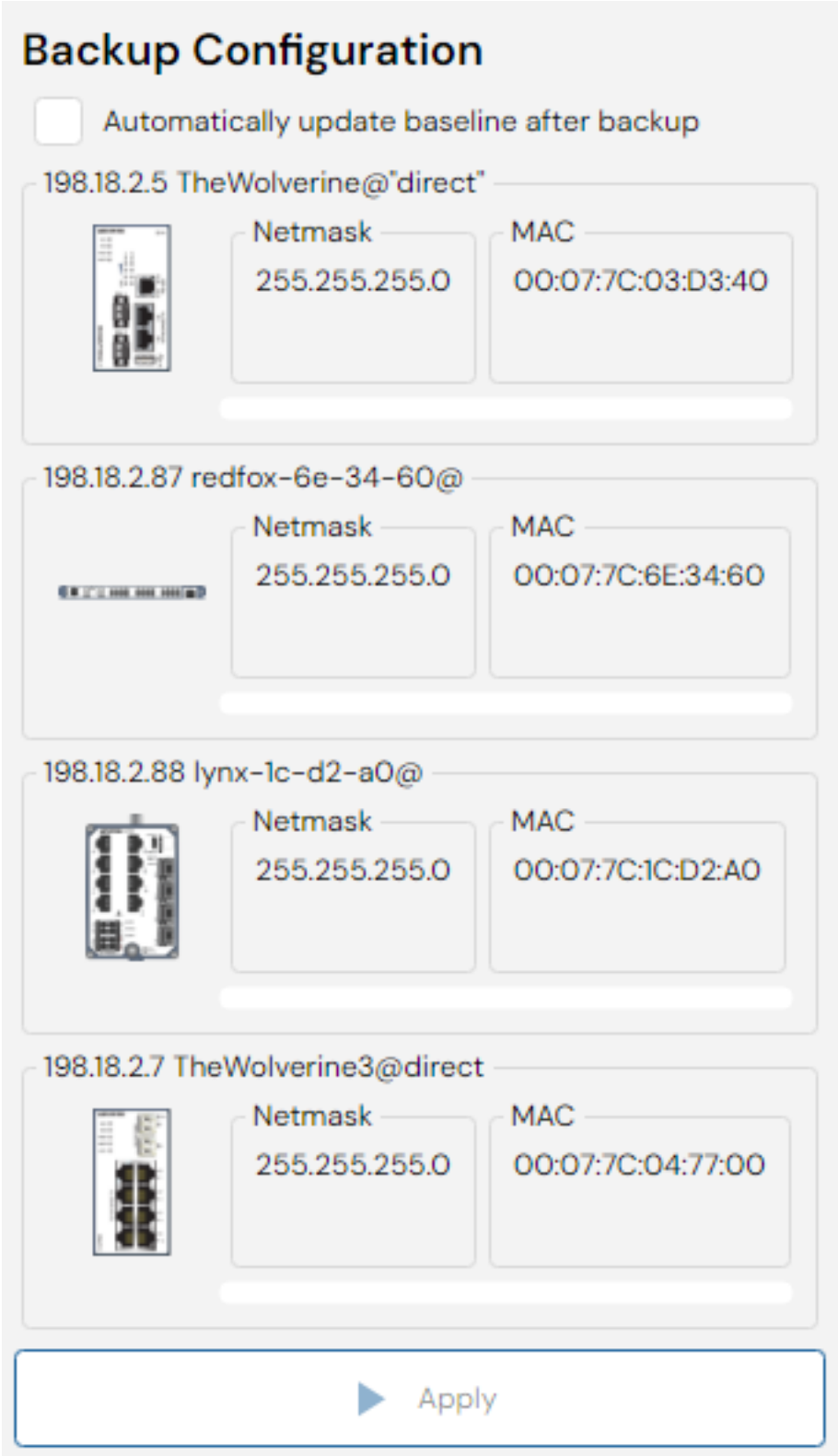
This panel displays information regarding the cellular status of the device, if applicable. The information held within primarily concerns xRD, Ibex, and Merlin devices. Depicted above is an example of a disconnected MRD-405, where it can be observed that, for example, its cellular self-test succeeded, it's operating in packet mode, but it has no network registration, and more.

4.4.37. Backup

WeOS 4 WeOS 5 Ibex XRD Merlin

Backup is a task that allows the user to perform multi-device backups in batches. Any backups generated with this functionality are saved with UTC time stamp.

Interface components



Depicted above is an example of how the user interface may appear for a set of selected devices. Each device is simply listed as an indicator for which devices will receive a backup command on task execution.

When selecting Automatically update baseline after backup at the top, then the following backup will be used as new baseline for the devices.

After a backup completes, the task displays an outcome summary for successful, failed, or partially successful runs. Each device also shows its own result.

4.4.38. Firmware Upgrade

WeOS 4 WeOS 5 Ibex XRD Merlin

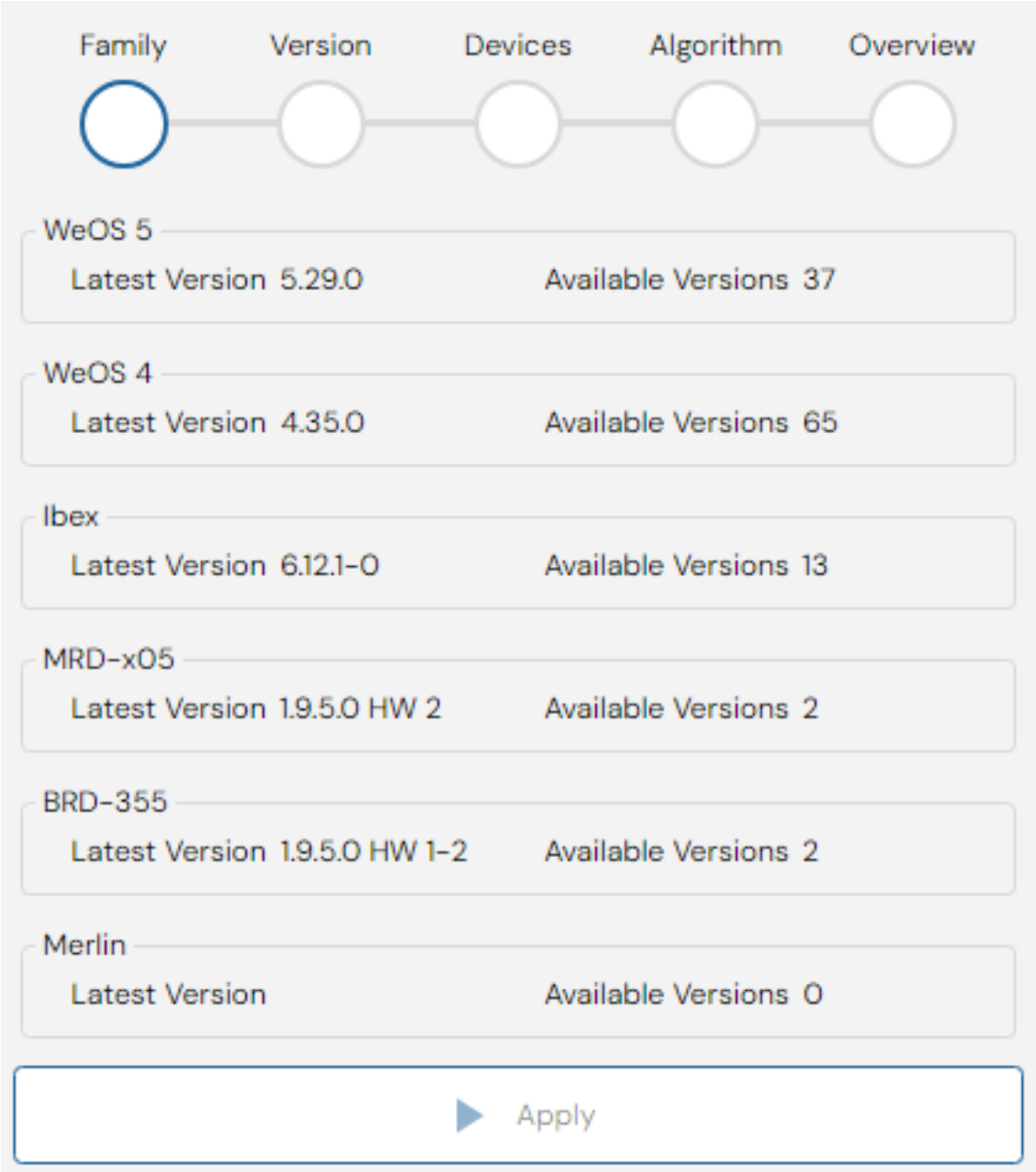
Firmware upgrade is a staged task that allows for upgrading or downgrading the running, backup and bootloader firmware on supported devices.

> **Info**
>
> To be able to use this feature, you must be connected to the internet, or have downloaded desired firmware ahead of time. If you have downloaded the firmware ahead of time, ensure it is placed in the firmware folder

Interface Components

The firmware upgrade interface is divided into five stages, Family, Version, Devices, Algorithm and Overview, where each is explained below:

Family

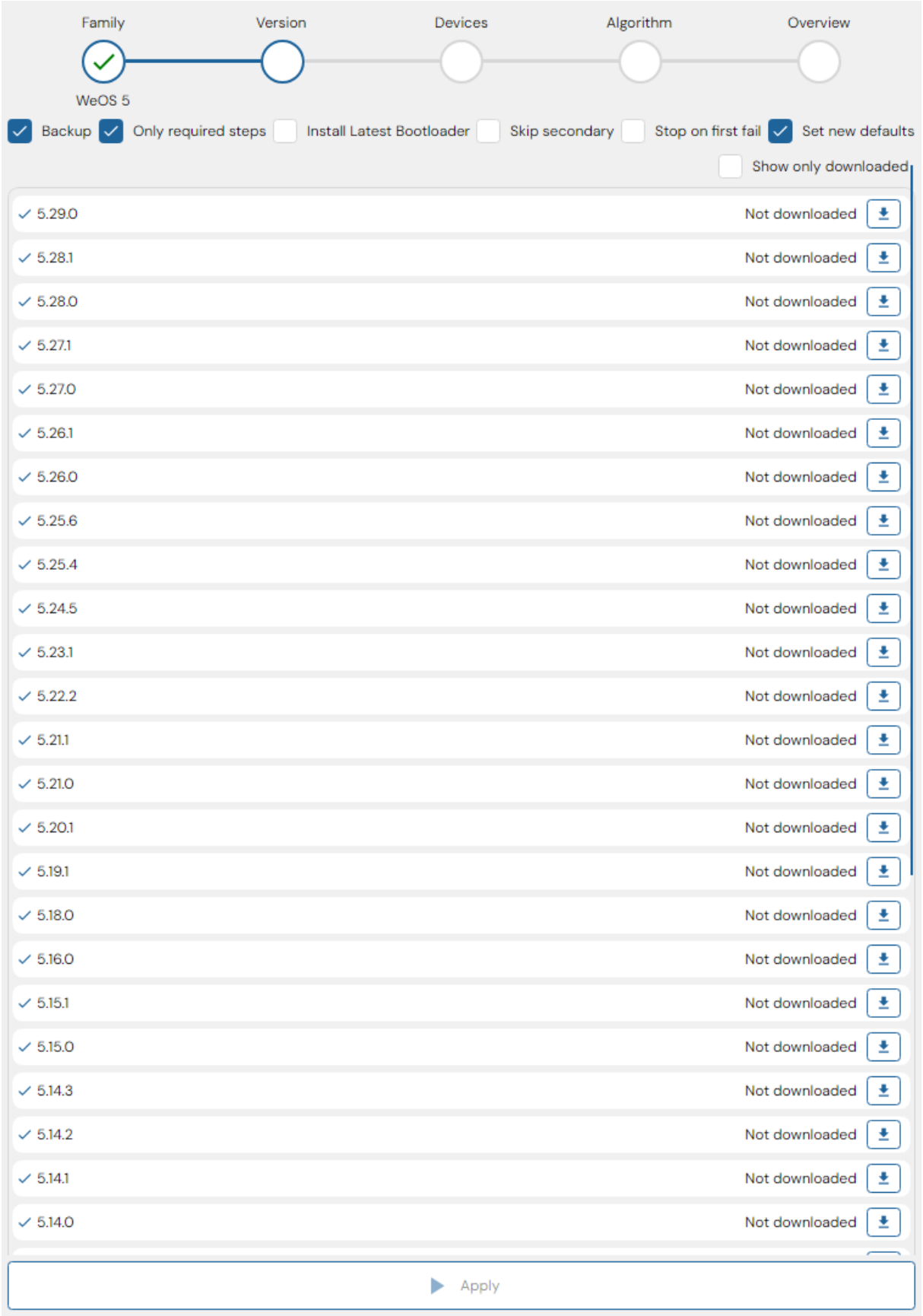


The first stage in firmware upgrade presents you with a choice of firmware families detected in the network. Where each family is listed as box with the title being the firmware family name, and the contents showcasing the latest version available within said family, as well as the total number of firmware versions present.

Click on the desired firmware family to proceed to the next stage.

> **Note**
>
> If only one firmware family currently exists in the network, this stage is skipped.

Version


Family
Version
Devices
Algorithm
Overview
WeOS 5
Backup
Only required steps
Install Latest Bootloader
Skip secondary
Stop on first fail
Set new defaults
Show only downloaded
5.29.0 Not downloaded
5.28.1 Not downloaded
5.28.0 Not downloaded
5.27.1 Not downloaded
5.27.0 Not downloaded
5.26.1 Not downloaded
5.26.0 Not downloaded
5.25.6 Not downloaded
5.25.4 Not downloaded
5.24.5 Not downloaded
5.23.1 Not downloaded
5.22.2 Not downloaded
5.21.1 Not downloaded
5.21.0 Not downloaded
5.20.1 Not downloaded
5.19.1 Not downloaded
5.18.0 Not downloaded
5.16.0 Not downloaded
5.15.1 Not downloaded
5.15.0 Not downloaded
5.14.3 Not downloaded
5.14.2 Not downloaded
5.14.1 Not downloaded
5.14.0 Not downloaded
Apply

The second stage in firmware upgrade presents a list of firmwares based on the selected family in the prior stage. These may either be marked as Validated, via a green checkmark, or as Unvalidated, via a yellow triangle with an exclamation point. A firmware version is considered Unvalidated when it originates from the users local disk, instead of being downloaded by WeConfig.

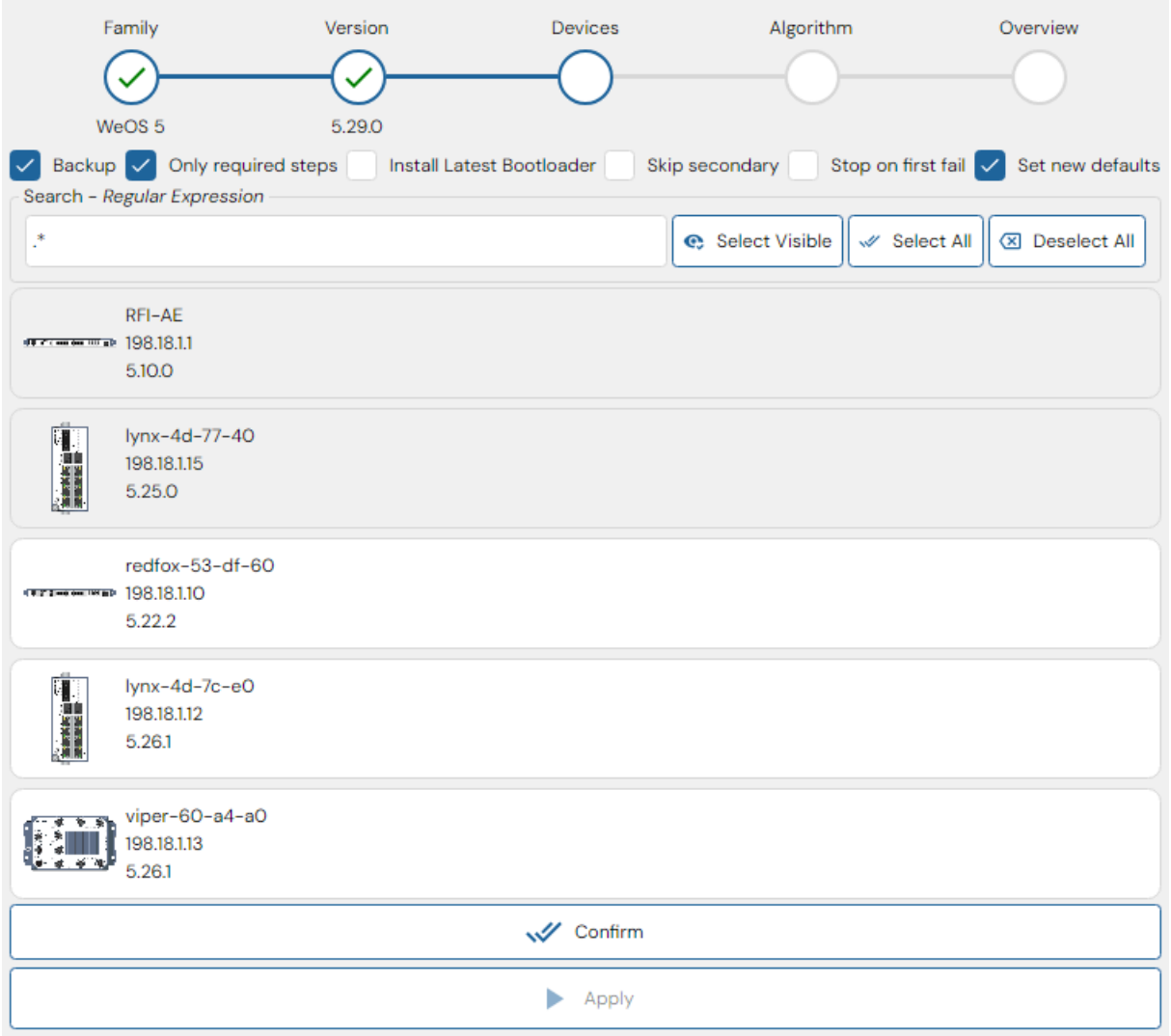
WeConfig makes no guarantees about the functionality or safety of Unvalidated firmware.

Additionally, this view will also indicate the download status of each firmware in the list. If the firmware is not already downloaded, as indicated in the picture above by the “Not downloaded” text

Click on the desired firmware version to proceed to the next stage.

> **Note**
>
> If only one firmware version exists in the selected family, this stage is skipped.

Devices



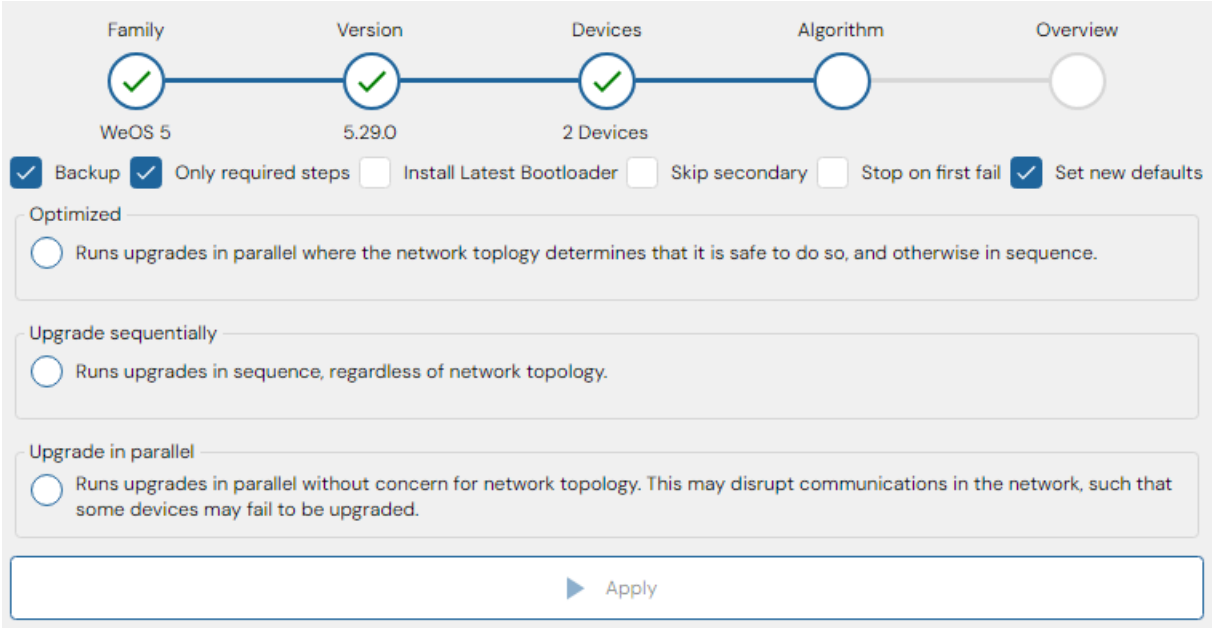
The third stage in firmware upgrade presents a list of devices compatible with the selected firmware version, where any matching devices already selected in the topology are

preselected. The list will also display any possible issues or warnings related to firmware upgrade. At the top of the interface a search box is provided to filter the list of devices, this box accepts regular expressions for advanced filtering, and matches against the following properties:

- IP Address
- Hostname
- Location
- Firmware Version
- Model Name

Select the desired devices (either manually or using select all) and click “Confirm” to proceed to the next stage.

Algorithm



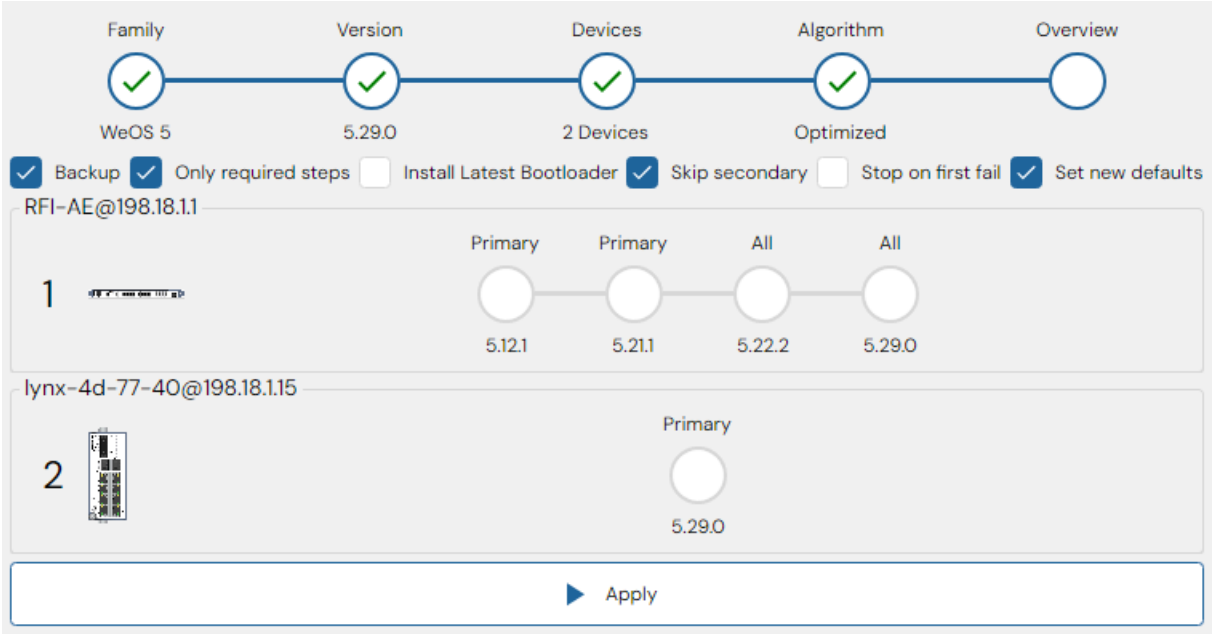
The fourth stage in firmware upgrade presents three options for the order-of-execution of the firmware upgrade sequence, the options are as follows:

Option	Description
Optimized	Parallelizes firmware upgrade when possible according to network order.
Upgrade sequentially	Do not parallelize firmware upgrade, run them in displayed top-down order.
Upgrade in parallel	Attempt to do all firmware upgrades in parallel

Click on the chosen algorithm option to proceed.

> **Note**
>
> Unless a specific use-case demands otherwise, it is suggested to use the Optimized algorithm when you have a fully detailed connection map of the topology.

Overview



The fifth and final stage in firmware upgrade presents you with a list of planned actions for WeConfig to undertake. As can be observed in the example above, to take `198.18.1.1` to `5.29.0`, WeConfig needs to upgrade it through the required versions of `5.12.1`, `5.21.1`, and `5.22.2` before finally upgrading to `5.29.0`.

Additionally, any remaining potential issues are also displayed on the devices here.

The number to the left of the device is its Upgrade Order. This is based on the algorithm you chose in the previous step. Devices with the same number will be upgraded in parallel.

If any planned upgrade step requires TFTP, the wizard shows a TFTP server address field. Enter the IP address of a TFTP server reachable by the devices before starting the upgrade.

Global options

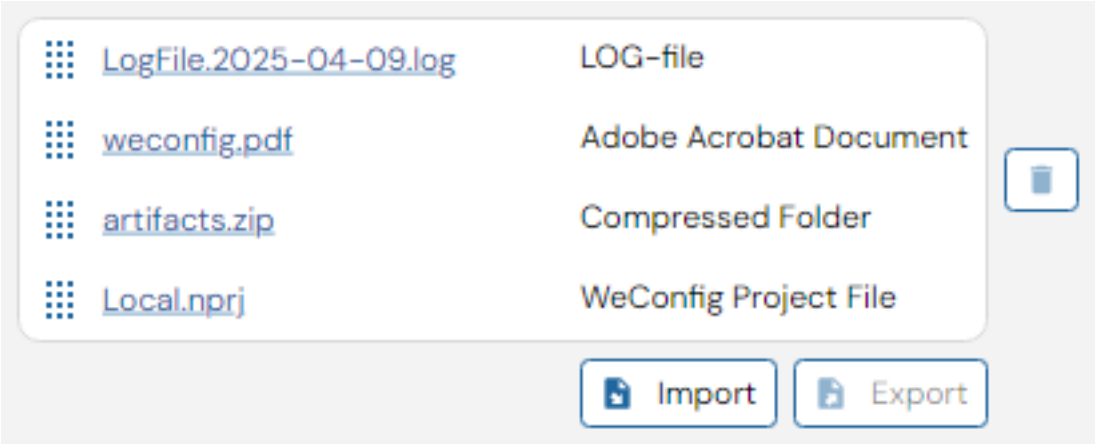
Atop the interface in stage 2-5 several checkboxes may appear, which slightly change the firmware upgrade procedure when checked, they are as follows:

Option	Description
Backup	When checked, WeConfig will take a backup of the current configuration file before performing an upgrade, checked by default
Only required steps	When unchecked, WeConfig will move through every intermediary version between the source and target version, when checked, WeConfig will only move through the versions considered necessary
Install latest bootloader	When checked, WeConfig will always attempt to upgrade the bootloader if a newer one exists in the provided firmware package, otherwise, WeConfig will only upgrade the bootloader when necessary

Option	Description
Skip Secondary	When checked, WeConfig will not upgrade the backup firmware on any of the devices, otherwise, it will upgrade the backup firmware as well.
Stop on first fail	When checked, WeConfig will not attempt further upgrades after the first failure occurs.
Set new defaults	When checked, WeConfig applies factory default parameters to each device after the firmware upgrade completes. This resets device configuration to the new firmware version's defaults

> *i* Info
>
> For firmware versions composed of multiple files (such as WeOS 5.26+), WeConfig handles the multi-file firmware package automatically. No additional user action is required — select the firmware package as usual, and WeConfig manages the individual files transparently during the upgrade process.

4.4.39. Attachments



This panel displays the file attachments associated with the current project as a list of links and format explanations.

Additional attachments may be added to the device here by clicking the “Import” button, and may be exported unto the regular file system using the “Export” button with a specific attachment selected. Attachments can also be deleted by selecting them and pressing the Trashcan button to the right side of the interface.

Clicking on any of the links for any of the attachments will open the corresponding file using the running operating systems preferred method. When the process the operating system decided to open is closed, such as when the user completes editing a file, the attachment will be updated and packed back into the project.

4.4.40. Device Access Settings

WeOS 4 | WeOS 5 | Ibex | XRD | Merlin | other

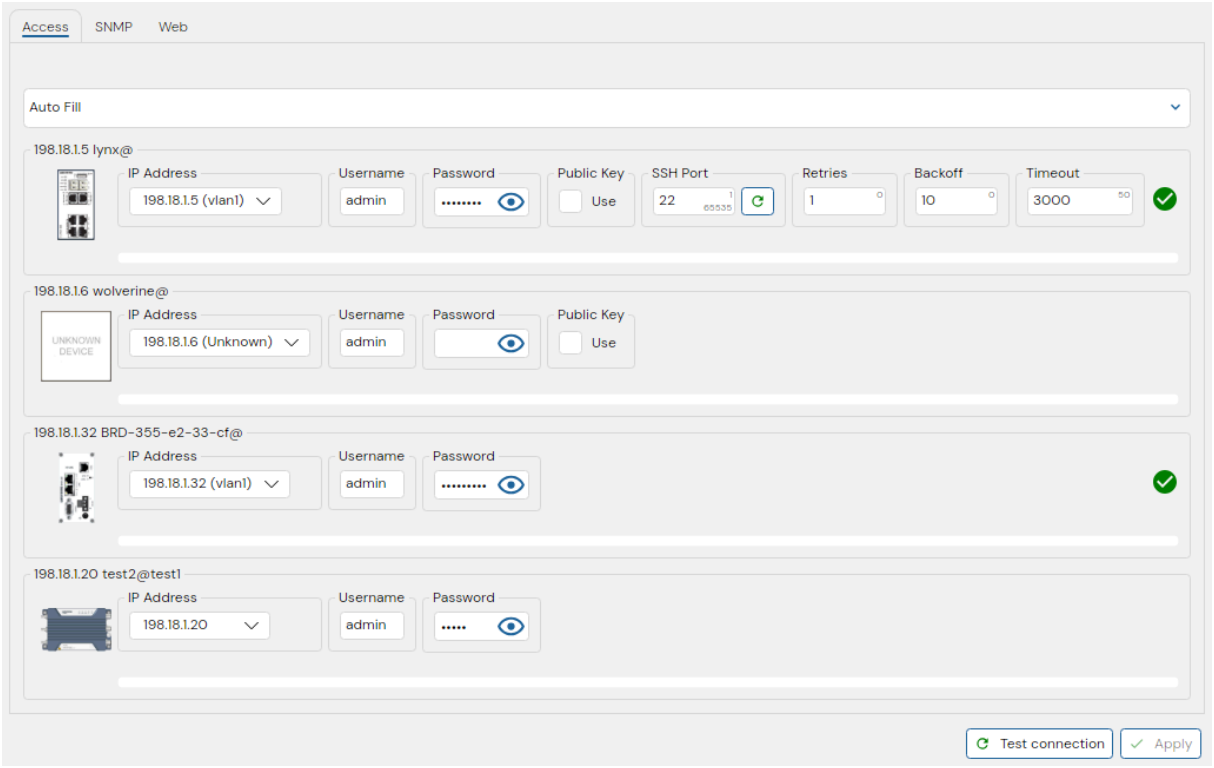
Device access settings is a panel dedicated to controlling how WeConfig authenticates and communicates with the devices in the network.

Interface components

The user interface of this panel is split into three tabs, Access, SNMP and Web, where each tab will contain a list of selected devices as well as an autofill section for quickly replicating settings across multiple devices. Below the three tabs two buttons are located, as follows:

- Test Connection: Check whether or not authentication/access succeeds given the current parameters in the panel. Success or failure will be indicated with either a green circle with a checkmark for success or a red circle with an exclamation point for failure attached to each relevant parameter.
- Apply: Sets the current parameters in the panel as the used authentication / access setting that WeConfig will associate with selected devices, then runs a Refresh on the selected devices.

Access Tab



The first of the three tabs, labeled Access, corresponds to device specific settings for how to access the primary means of device communication / configuration, which varies from firmware category to firmware category.

For WeOS devices, this primarily refers to the account used for SSH communication, but is also used for HTTP(s) communication in certain specific scenarios such as firmware upgrade. In contrast, for Ibex devices, this primarily refers to the account used to talk to the REST API on the device, and so on.

The exact parameters presented per device varies in accordance with what it represents. On WeOS 4/5, you will be presented with the following options:

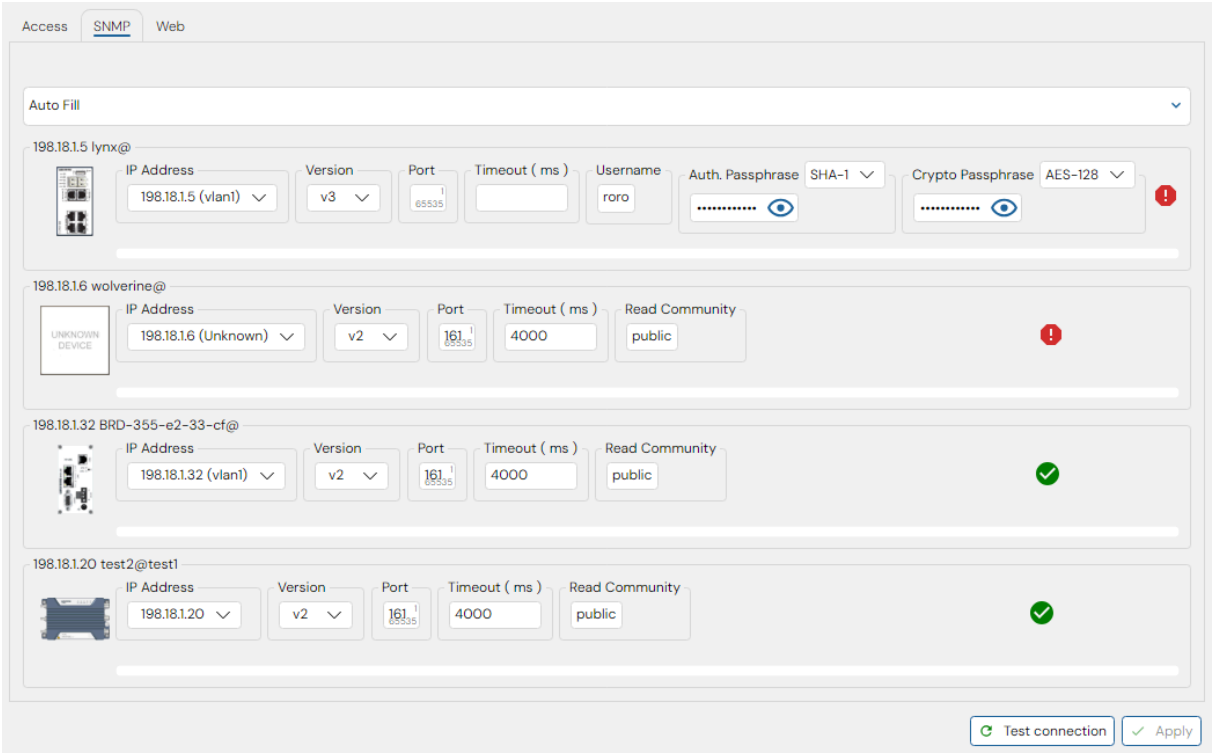
- IP Address: The management IP address used to reach the device, it is a dropdown list containing all known network interfaces / addresses that WeConfig has associated with the device.
- Username: The account username that WeConfig will try and authenticate with over SSH / Web.
- Password: The account password that WeConfig till try and authenticate with over SSH / Web, if any.
- Public Key: A checkbox indicating whether or not WeConfig should try to authenticate using the PC's public key store.
- SSH Port: Which port WeConfig should try and access the device on to reach an SSH session.

Similarly, it can be observed that for xRD's like the `BRD-355` depicted in the example above, only the password box will be editable, as that is the only form of authentication parameter that can be handled by that firmware at the time of writing.

Additionally, as depicted in the example above, when "Test Connection" has been run, the success of the configured parameters is indicated by checkmarks or exclamation points to the righthand side of each device parameter set.

> **Info**
>
> The contents of the Access tab may be automatically updated when configuring accounts.

SNMP Tab



The second of the three tabs, label SNMP, corresponds to device specific settings for how WeConfig communicates with devices using the SNMP protocol. It, similarly to the access tab,

has a dropdown box for selecting the target management IP address amongst the devices known IP addresses.

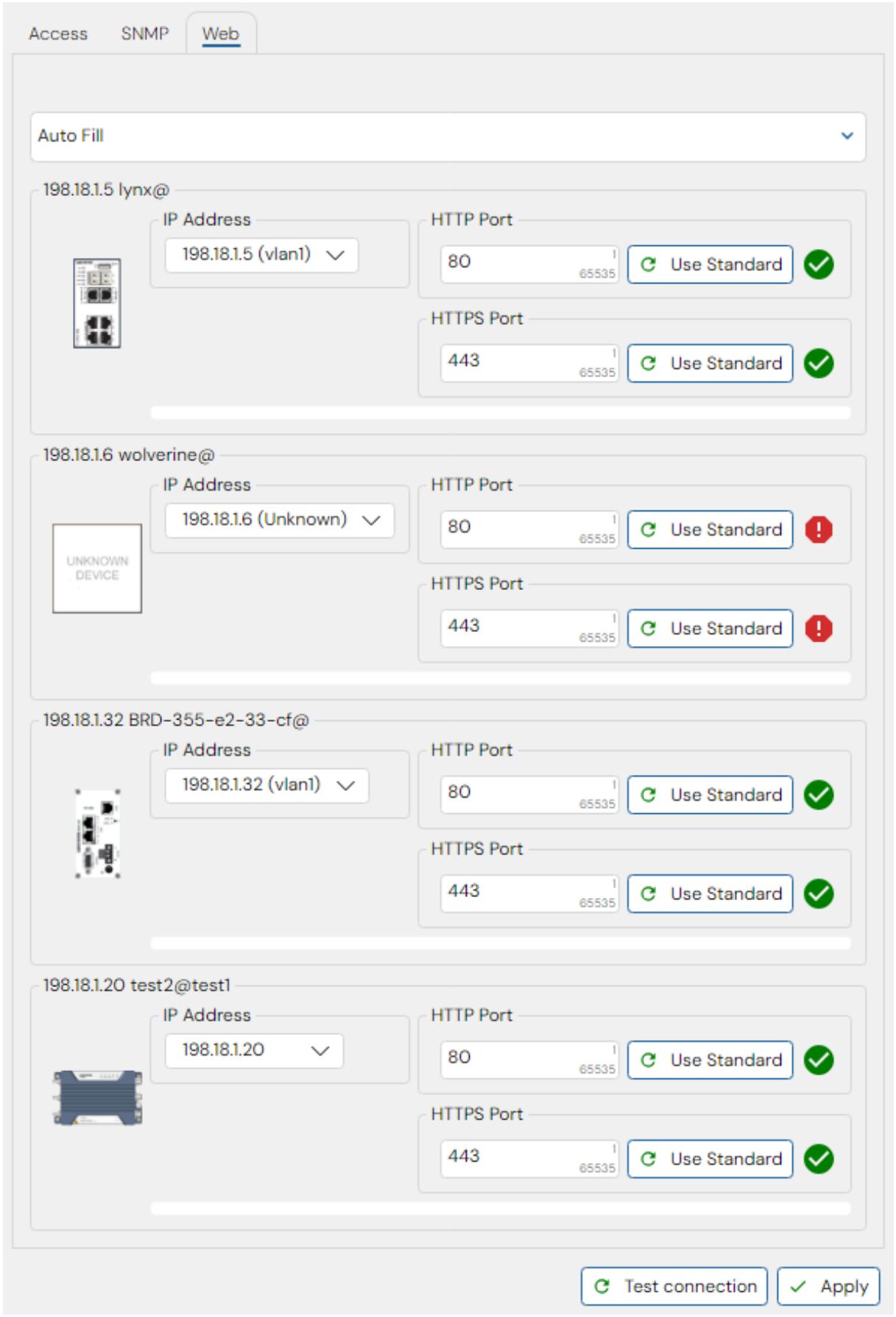
Secondly, each device contains a selector between SNMP v2 and SNMP v3, with the corresponding settable parameters adjusted based on your selection. Both versions have configurable port and timeout values. For SNMP v2, set the read community; for SNMP v3, set a username and, optionally, an authentication digest and password, as well as a privacy protocol and password.

The autofill section has separate checkboxes for each setting. Select only the fields to copy to the other selected devices. For SNMP v2, the fields include port, timeout, and read community; SNMP v3 also provides autofill for its username and authentication and privacy settings.

Similarly to the Access tab, the success of using these parameters to communicate with the device is indicating to the righthand side as either a checkmark or an exclamation point.

> *i* Info
>
> The contents of the SNMP tab may be automatically updated when configuring SNMP

Web Tab


Access
SNMP
Web
Auto Fill
198.18.1.5 lynx@
IP Address
198.18.1.5 (vlan1)
HTTP Port
80
65535
Use Standard
HTTPS Port
443
65535
Use Standard
198.18.1.6 wolverine@
IP Address
198.18.1.6 (Unknown)
UNKNOWN DEVICE
HTTP Port
80
65535
Use Standard
HTTPS Port
443
65535
Use Standard
198.18.1.32 BRD-355-e2-33-cf@
IP Address
198.18.1.32 (vlan1)
HTTP Port
80
65535
Use Standard
HTTPS Port
443
65535
Use Standard
198.18.1.20 test2@test1
IP Address
198.18.1.20
HTTP Port
80
65535
Use Standard
HTTPS Port
443
65535
Use Standard
Test connection
Apply

The third and final tab in this panel, labeled Web, corresponds to device specific settings for overriding the default HTTP and HTTPs ports. This interface is fairly straightforwards, and does not differ between device firmware families.

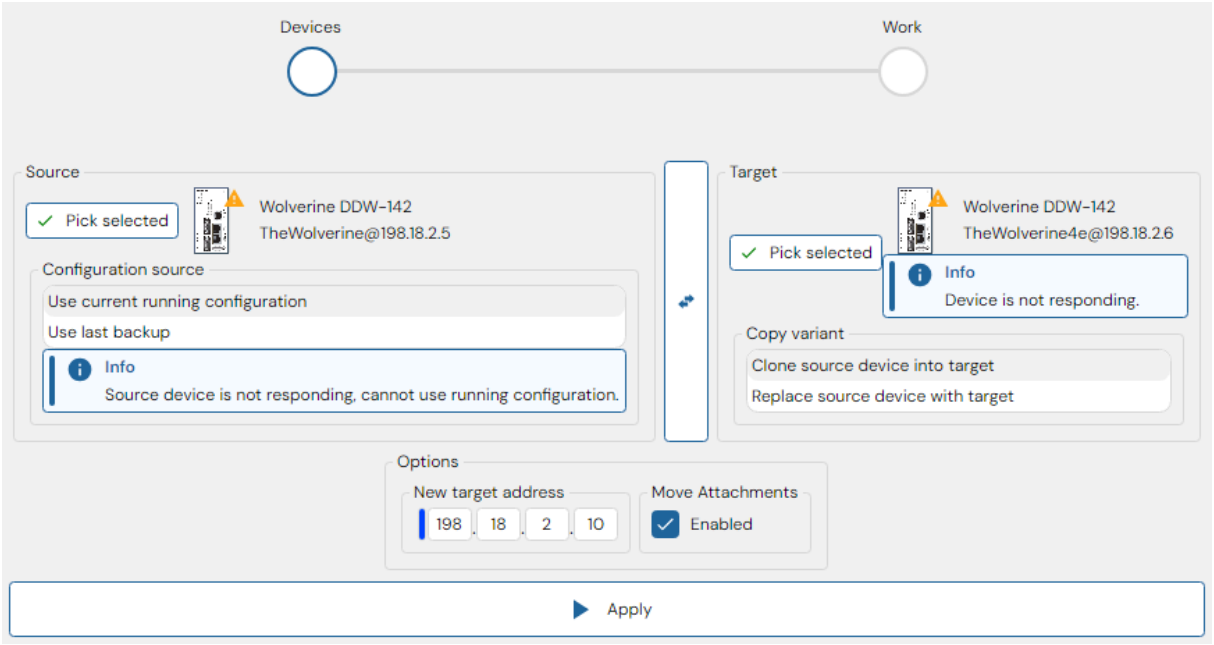
Similar to prior tabs, “Test connection” will also indicate whether the corresponding HTTP / HTTPS protocol is successfully reachable on the indicated port.

4.4.41. Clone or Replace Device

WeOS 4 WeOS 5 Ibex XRD Merlin

Clone or Replace device is a staged task for, as the name indicates, either copying WeConfig's configuration information from one device to another one, or to replace a device in the topology with a new one.

Interface components



The initial interface you will be met with when opening this panel is seen above. Here you may pick two devices using selection in either the Topology or Device list.

Secondly, for the "Source" device, you may select to either actively pull the current running configuration of that device, or use the last known backup (If WeConfig has one) for the copy operation.

Thirdly, for the "Target" device, select whether you are intending to Clone the device or in othe words create another device with almost the same configuration in the current network, or Replace the device, meaning that WeConfig will remove the original "Source" device from it's knowledge of the network once the operation has completed.

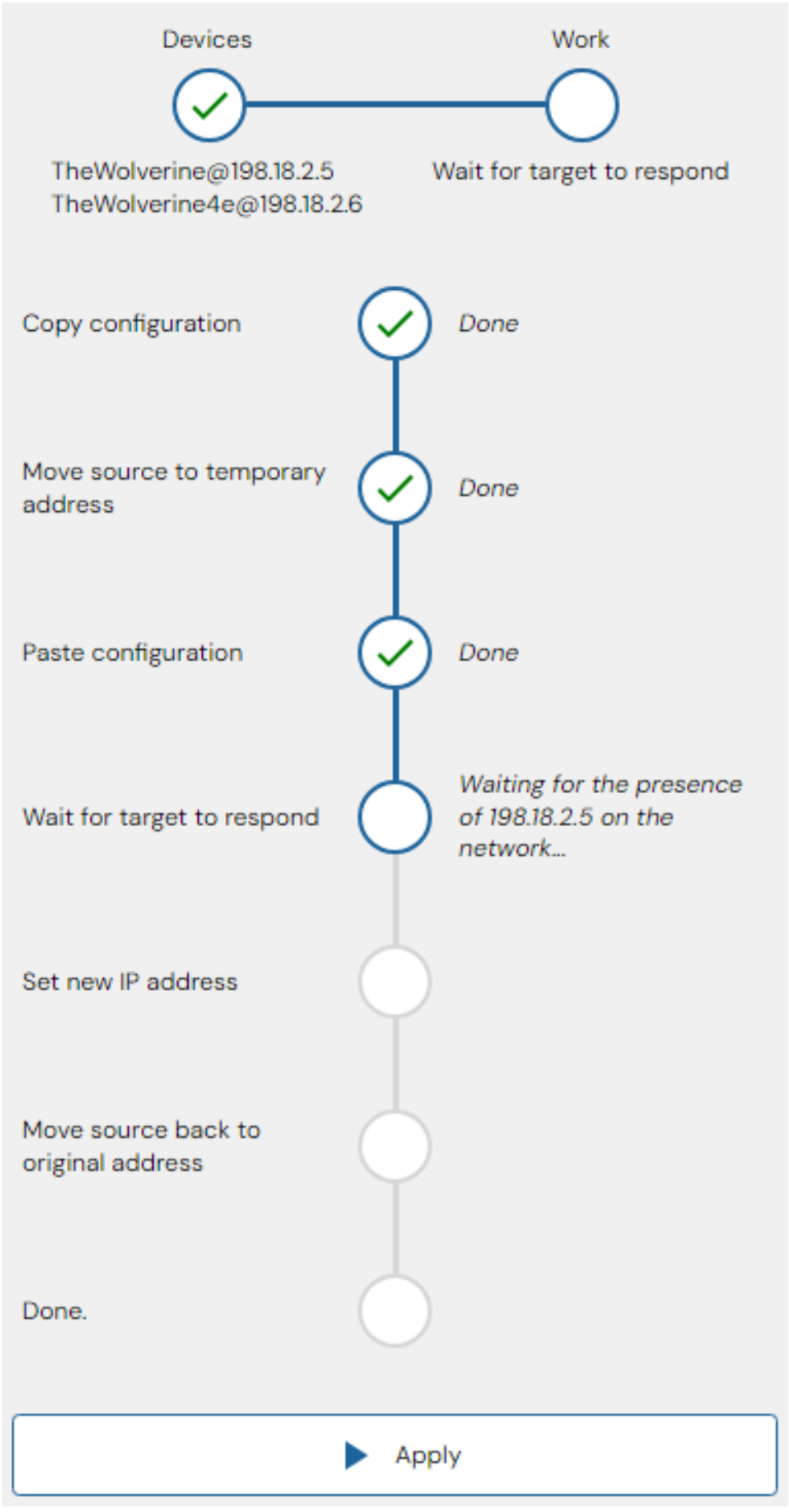
If you have selected the devices in the wrong order, and wish to swap the source and target of the operation, the button between the two selections allows you to do so trivially.

Finally, fill out relevant options in the "Options" group below the two selected devices. The exact content of this box may vary depending on the selected devices and operation type. But the following options may appear:

Option	Description
Select network adapter	When the target device requires knowledge of the PC's connected network adapter to successfully receive a new IP address, such as for older versions of WeOS 4, this option will appear and prompt the user to select the adapter that the device is connected to.

Option	Description
New target address	When the “Copy variant” selection is set to “Clone source device into target”, then this option will be present, allowing you to select the new IP address for the target device, to be assigned after configuration has been copied over
Move attachments	Always present, when checked, any Device attachments beloning to the source device will be moved to the target device at the end of the copy operation.

When you are content with your selection, press “Apply” to begin the copy operation. This will transition the interface into a view similar to the one seen below:


Devices
Work
TheWolverine@198.18.2.5
TheWolverine4e@198.18.2.6
Wait for target to respond
Copy configuration
Done
Move source to temporary address
Done
Paste configuration
Done
Wait for target to respond
Waiting for the presence of 198.18.2.5 on the network...
Set new IP address
Move source back to original address
Done.
Apply

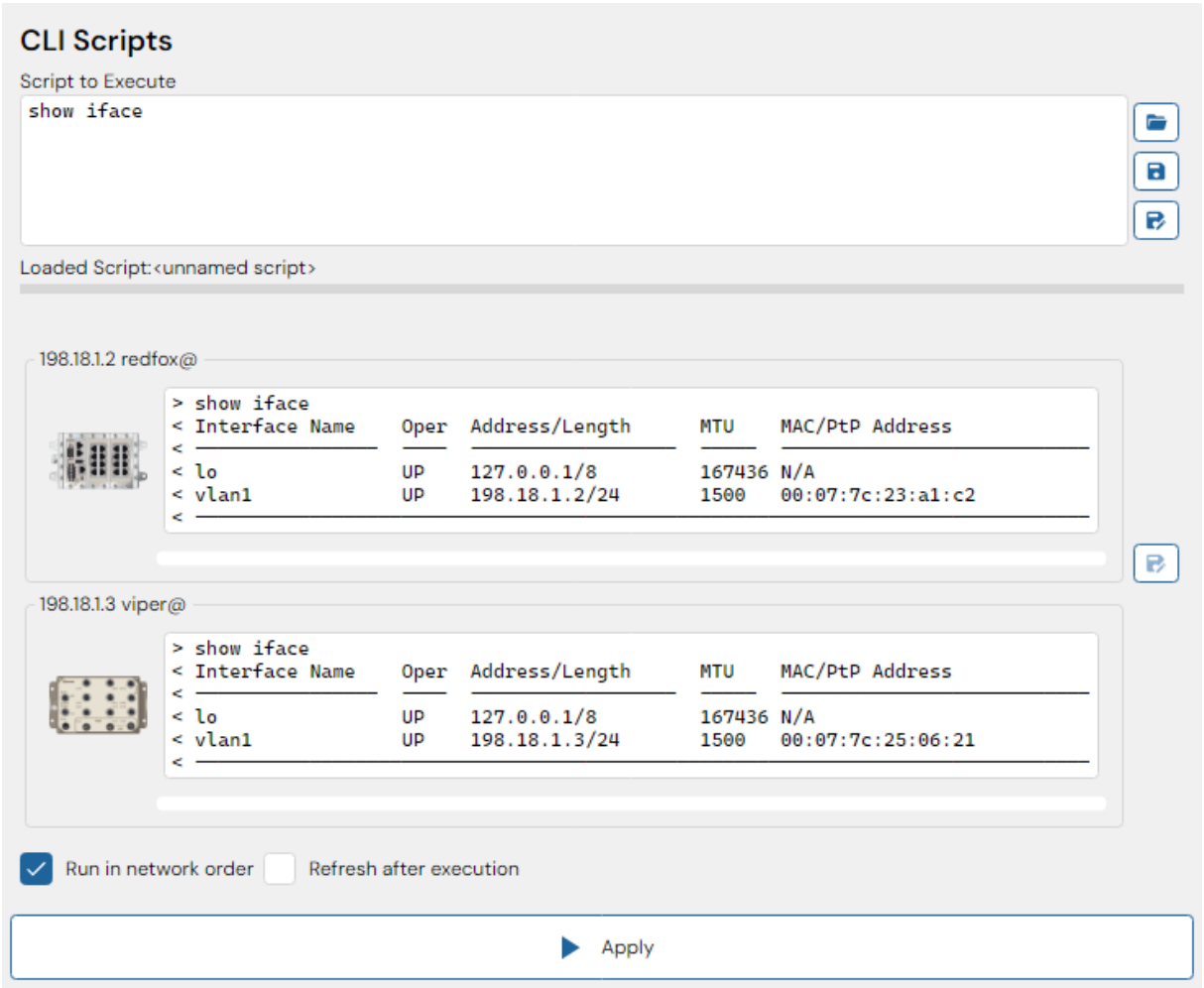
As can be seen, the copy operation is comprimised of several stages, which may vary slightly depending on the exact configuration chosen in the previous view. Some of these stages may require user interaction, and will in such case prompt the user for such.

4.4.42. CLI Scripts

WeOS 4 WeOS 5

CLI Scripts is a task that allows for the execution of custom scripts on multiple devices simultanously.

Interface Components



Script box
This box labled “Script to execute” is editable and contains the current script to execute on the selected devices. The script is a sequence of newline-seperated CLI commands that will be executed in order on all the selected devices.

Open button
The top of the three buttons to the right of the script box, this button will open up a file browser allowing the user to load a file to the script box.

Save buttons
The middle and bottom of the three buttons to the right of the script box, this button will save the current contents of the script box to the project or a user specified location, respectively.

Device list
Beneath the script box is a list of devices, where the contents of each list element is populated with the input and output of the last script executed, denoted by > and < respectively.

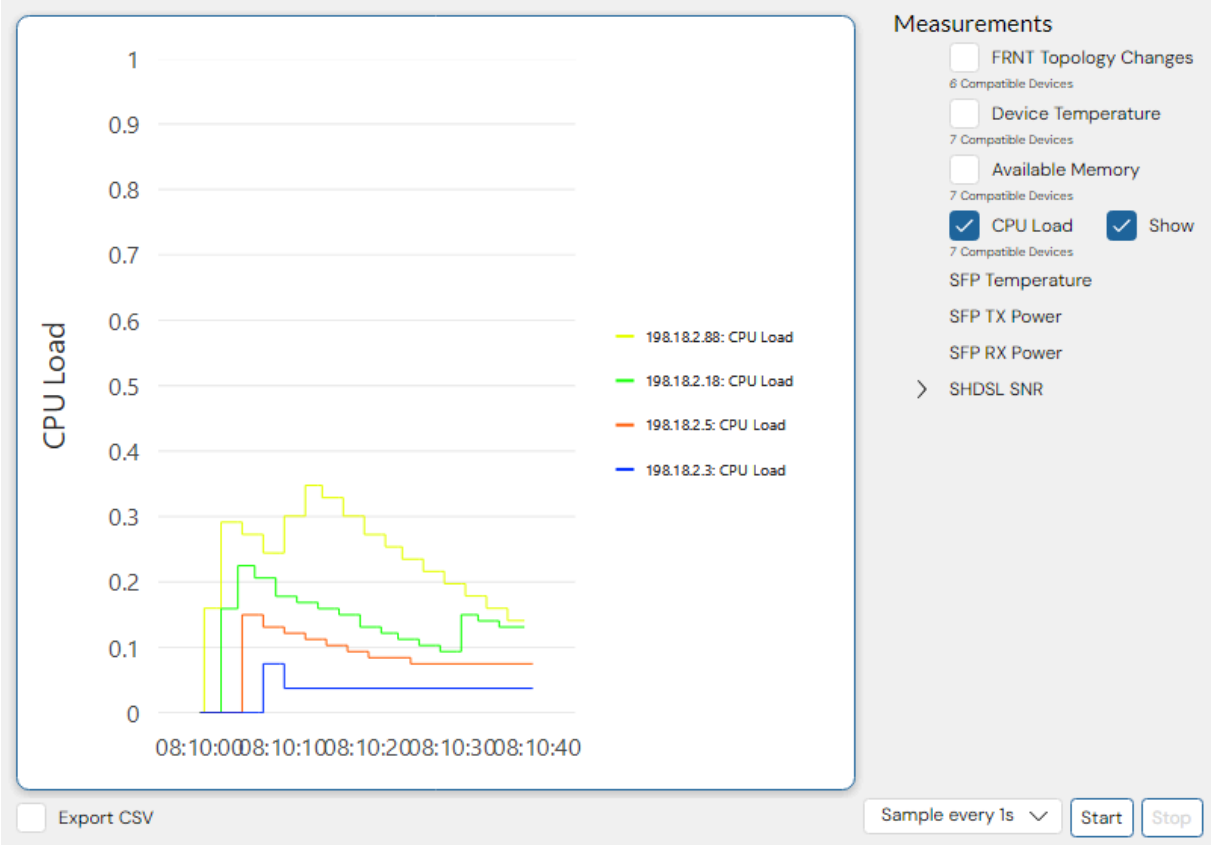
Save buttons
To the right of the device list is a button, this will save the input/output of the scripts per device to a user specified location.

Run in network order
This checkbox, when checked, will cause the script to run in network topology order, rather than in complete parallel.

Refresh after execution
This checkbox, when checked, will cause WeConfig to run a Refresh after all the devices have executed the provided script.

4.4.43. Diagnostics

WeOS 4 WeOS 5 Ibex XRD Merlin



In this Panel, a number of datapoint about the network can be observed and charted, such as:

- Available memory
- CPU load
- Device temperature
- FRNT change count
- PoE Power
- RSSI
- SFP Rx/Tx Power
- SFP Port temperature
- SHDSL SNR margin

The exact data sources present will vary depending on the devices selected in the topology.

To monitor the data sources, select them in the list to the righthand side of the interface, select a sampling rate from the dropdown menu at the bottom of the interface, and press start.

If you need to log the observed data for later analysis, make sure to check the “Export CSV” toggle in the bottom right corner just below the graph, and select a path to the target file.

It is possible to show and hide individual graphs at the sampling. Click the “Show” button in the list of monitored devices on the right side.

Additionally, you may hover any individual graph line to show a panel with the exact value at that point on the graph for all measured devices.

> **Note**
>
> When a monitor session is restarted, the graph is cleared. Data saved to CSV will not be lost. A new monitor session will add data to the CSV file, not replace it.

> **Warning**
>
> When you are exporting a measurement to CSV, do not open the CSV file in Excel or any other application at sample, as such a action may lock the file, preventing data from being written to it by WeConfig.

4.4.44. Syslog

WeOS 4 | WeOS 5 | Ibex | XRD | Merlin | other

When enabled in application settings, this panel contains a view of received syslog data.

Interface components



Any device that is configured to use WeConfig as Syslog server will display its Syslog messages in this tab, as can be seen in the example above. Click the corresponding device to switch view to that device's syslog messages.

The button to the lower right of the interface "Export" allows you to export the selected device's logged syslog messages to a file.

> **Info**
>
> In order to receive Syslog messages, the running machine's firewall must be configured to allow said syslog messages, which arrive on Port 514, see usage for more information.

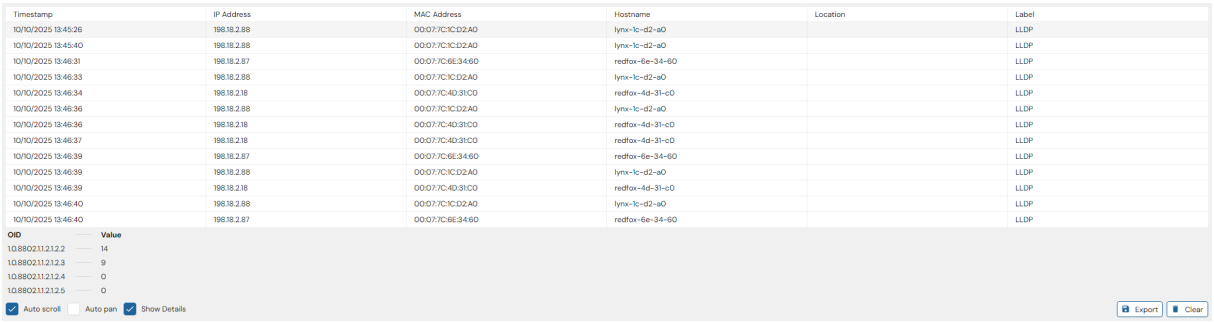
4.4.45. Traps

WeOS 4 | WeOS 5 | Ibex | XRD | Merlin | other

This panel contains a tabular view of received SNMP traps.

> **Info**
>
> In order to receive traps, the machine running WeConfig must have been configured as a trap host on the origin device.

Interface components



As depicted in the example above, arriving SNMP traps will be listed in this panel, and can be sorted or filtered within. Severe traps may be color-coded based on its severity level, making it easier to identify critical events at a glance.

Additionally, the raw SNMP content of the trap can be viewed for a selected trap while the “Show Details” toggle is checked, as depicted above.

If the “Auto scroll” toggle is checked, the list will automatically scroll to the bottom of the interface whenever a new trap arrives.

If the “Auto pan” toggle is checked, selecting a trap in the list will attempt to make the topology zoom to the corresponding device.

Click the Export button to export the list to a CSV file.

Click the Clear button to clear the list.

> **Info**
>
> On Windows, WeConfig uses Windows trap host when enabled.
>
> For full functionality, the Windows trap host must be disabled, as that will permit WeConfigs internal trap host to function.

> **Info**
>
> On Linux, WeConfig will not be able to receive traps unless ran in elevated permissions (e.g., using `sudo -E`).

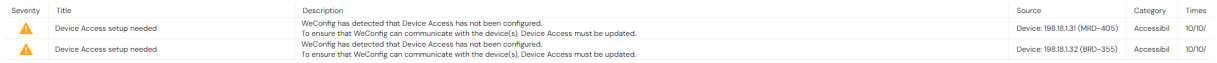
4.4.46. General Log



This panel provides a running view of the WeConfig log file. Its primary purpose is for developers and power-users, but it is generally available when configured in application settings for those that desire to see it. Depicted above is an example of how the log may appear when set to capture everything up to debug messages, which are marked with `[DBG]` and somewhat greyed out in the example above.

> **Info**
>
> This view also provides a reasonable view into what telemetry is being collected by WeConfig, when it is allowed to do so.

4.4.47. Issues



This panel contains a summary of any diagnostic issues WeConfig has encountered, either with devices in the network or with the application environment itself. Each issue has the following attributes:

Attribute	Description
Severity	How severe the issue is considered, may be either Informational, Warning or Error,
Title	A title of the diagnostic
Description	A more detailed explanation of what the issue entails
Source	Wherefrom the issue arose, if it is associated with a specific device or not
Category	The kind of issue being presented
Timestamp	The date at which the issue was generated.

Each of these issues has a context menu that, if there is an offered way to remediate the issue, contains a “Remediate” option that will cause WeConfig to do what it can to resolve the underlying problem causing the issue.

Any issue that is not a concern to the user can also be hidden with the “Hide” context menu option.

> **Info**
>
> Informational indicates issues that may improve the network from non-functional aspects, such as security

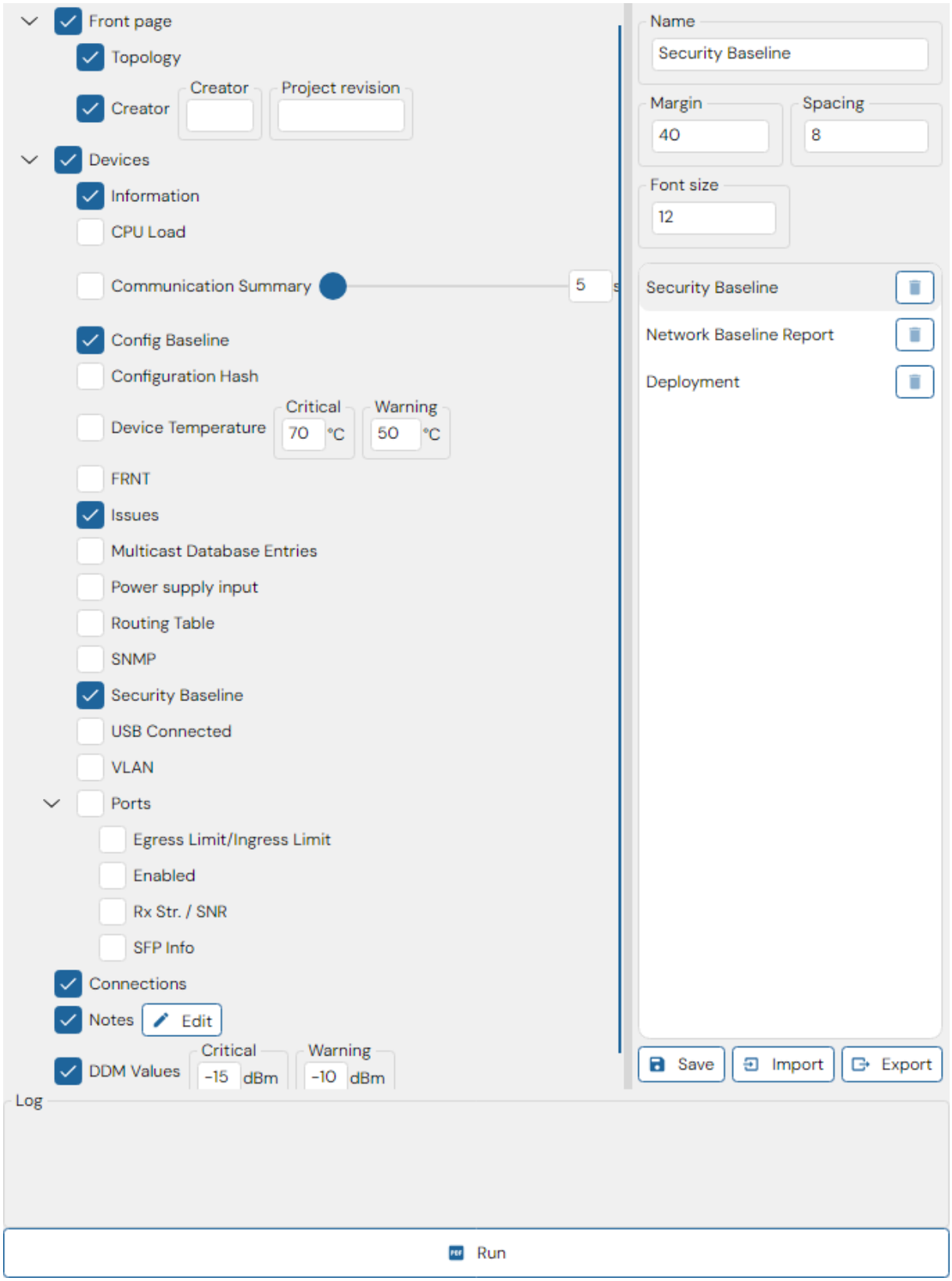
> **Warning**
>
> Warning indicates an issue that reduces functionality, such as depicted above, where certain devices report that they do not have correct authentication.

> **Danger**
>
> Error indicates an issue that prevents proper functionality, such as a blocking firewall.

4.4.48. Reports

WeOS 4 WeOS 5 Ibex XRD Merlin

This licensed panel allows for the generation of PDF reports about the state of the current project. The image below depicts the typical user interface for this panel.



Interface components

Content Tree
On the left side of the interface, you will find a tree listing the contents that will be included in the report. This tree allows you to select and deselect specific items to be included in the report. Some of these components, such as `Creator` and `Notes`, allow you to provide specific information to be included in the report.

Options
In the top right of the interface, you will find a number of options that affect the behavior and appearance of the generated report. These options include:

Option	Description
Name	The name of the report.
Margin	The page margins.
Spacing	The default spacing between elements in the report.

Report Templates
Beneath the options section, you will find a list of available report templates. These templates are predefined sets of tree structures that you may select by clicking on them, which will update the content tree to reflect the selected template. You may also create your own report templates for later use, which can be done by adjusting the content tree and clicking the `Save` button, this will create a template with the name specified in the `Name` option above.

Log
During report generation, a log will be populated by the current progress. Once report generation has finished, the resulting PDF will be opened in your default PDF viewer, and the file will additionally be stored in the current project file.

> ⚠ Warning
>
> In order to successfully generate any report, WeConfig must have a network connection to the devices involved.

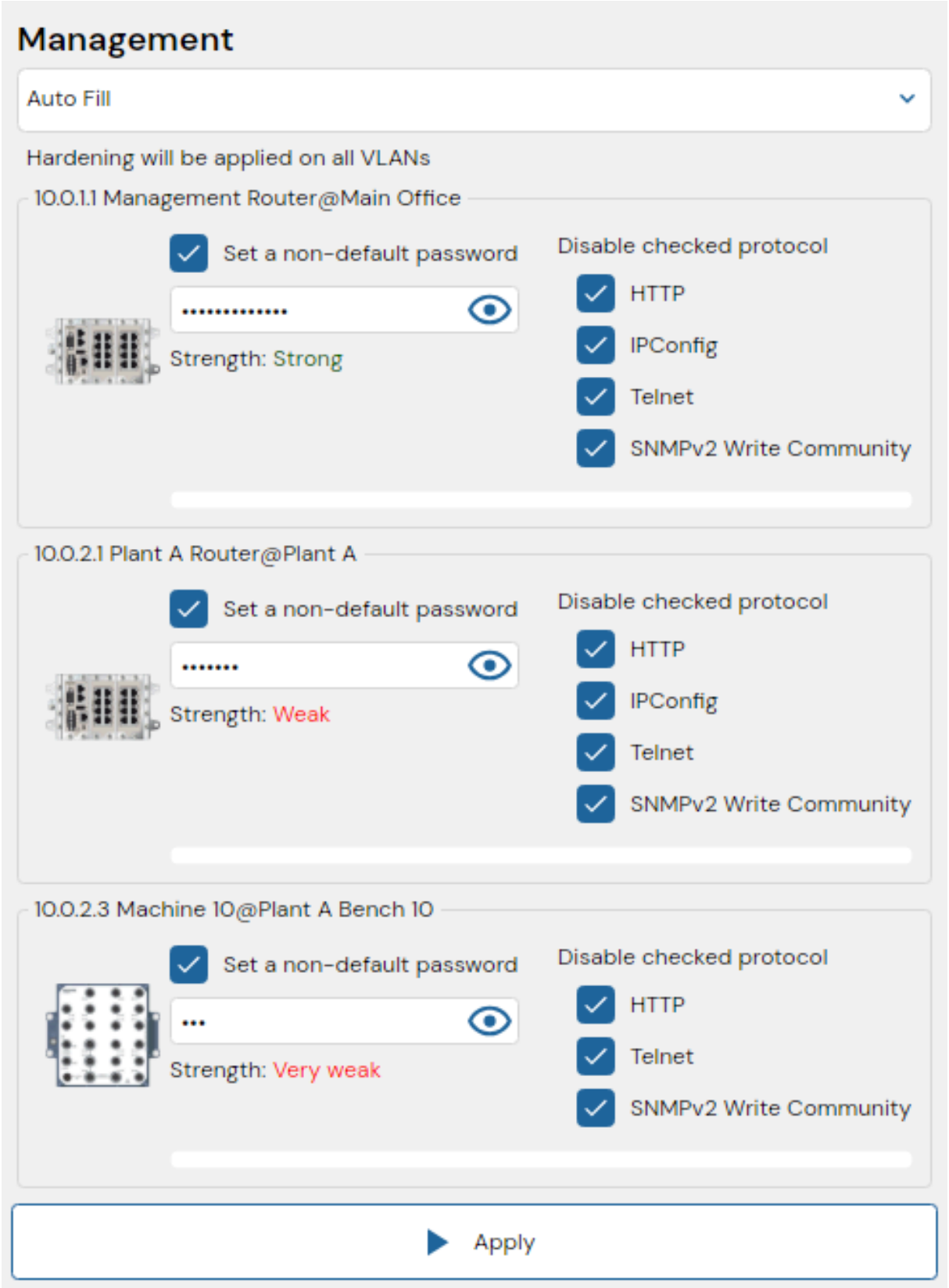
4.4.49. Security

4.4.49.1. Management Hardening

WeOS 4 WeOS 5

Management hardening is a task that allows for detection of some security issues related to device management.

Interface components



Use this panel to scan all or the selected devices in the project for known management hardening issues. These include the use of:

- HTTP for the web service
- SNMPv2 write community

- The default admin password
- IPConfig
- Telnet

When Scan is clicked, each device in the project will be interrogated for any of these issues. When the scan is finished, WeConfig will list all devices and their found issues.

WeConfig will by default suggest removing all issues. If the default admin password has been used on any device, it will not be possible to apply the fixes until the password has been changed.

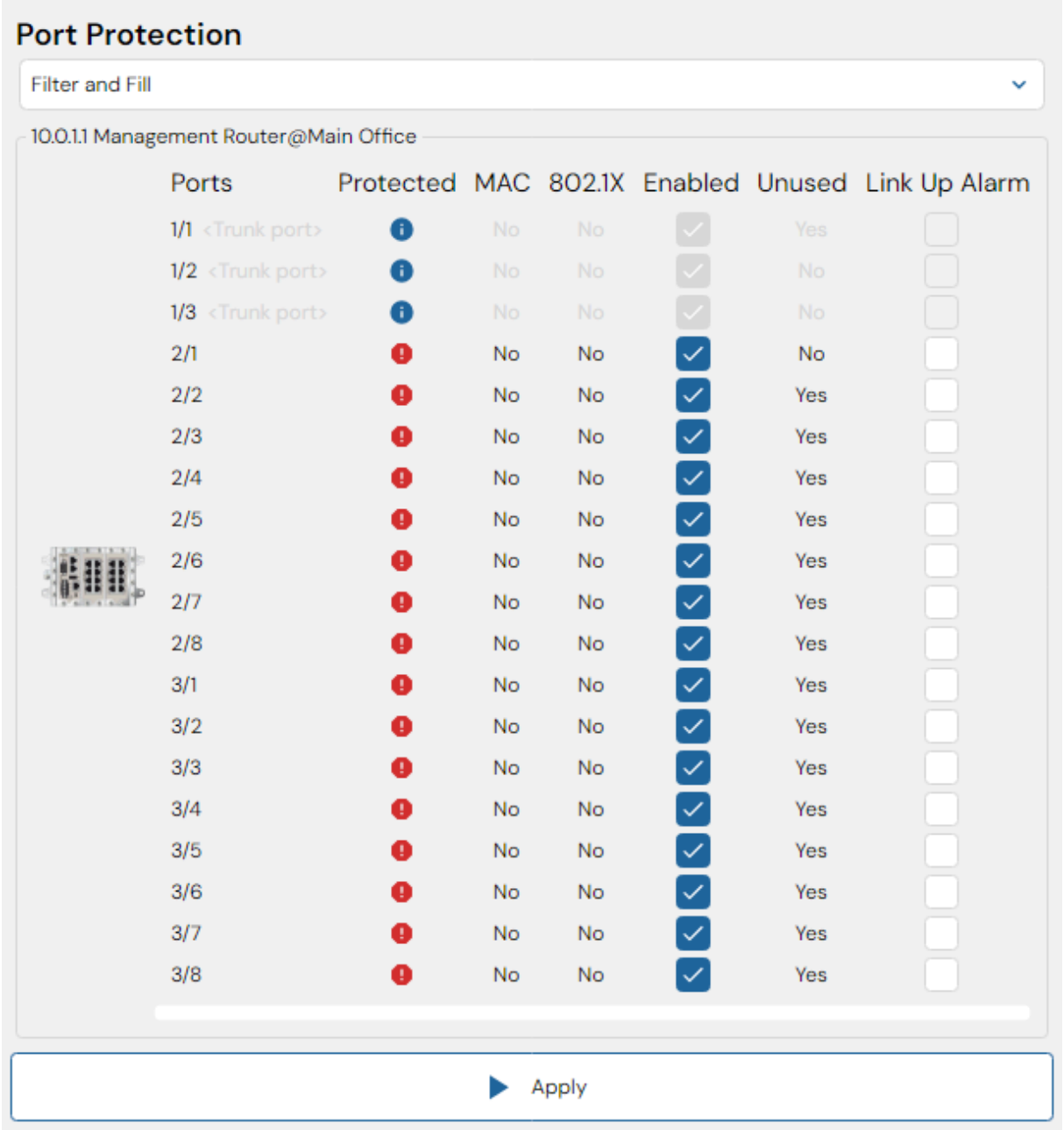
If any of the known issues are ignored, it is necessary to be explicit and uncheck the issue. This can be done easily from the Autofill section.

4.4.49.2. Port Protection

WeOS 4 WeOS 5

Port protection is is a task for reviewing MAC filter / 802.1X port protection status and enablding/disabling ports as well as configuring link-up alarm triggers.

Interface components



As can be seen in the interface depicted above, this panel will present a list of selected devices, where each device contains a matrix of ports with 7 columns, as follows:

Column	Description
Ports	The Port name, as well as an indicator if it is a trunk port or not
Protected	Whether or not this port is considered protected

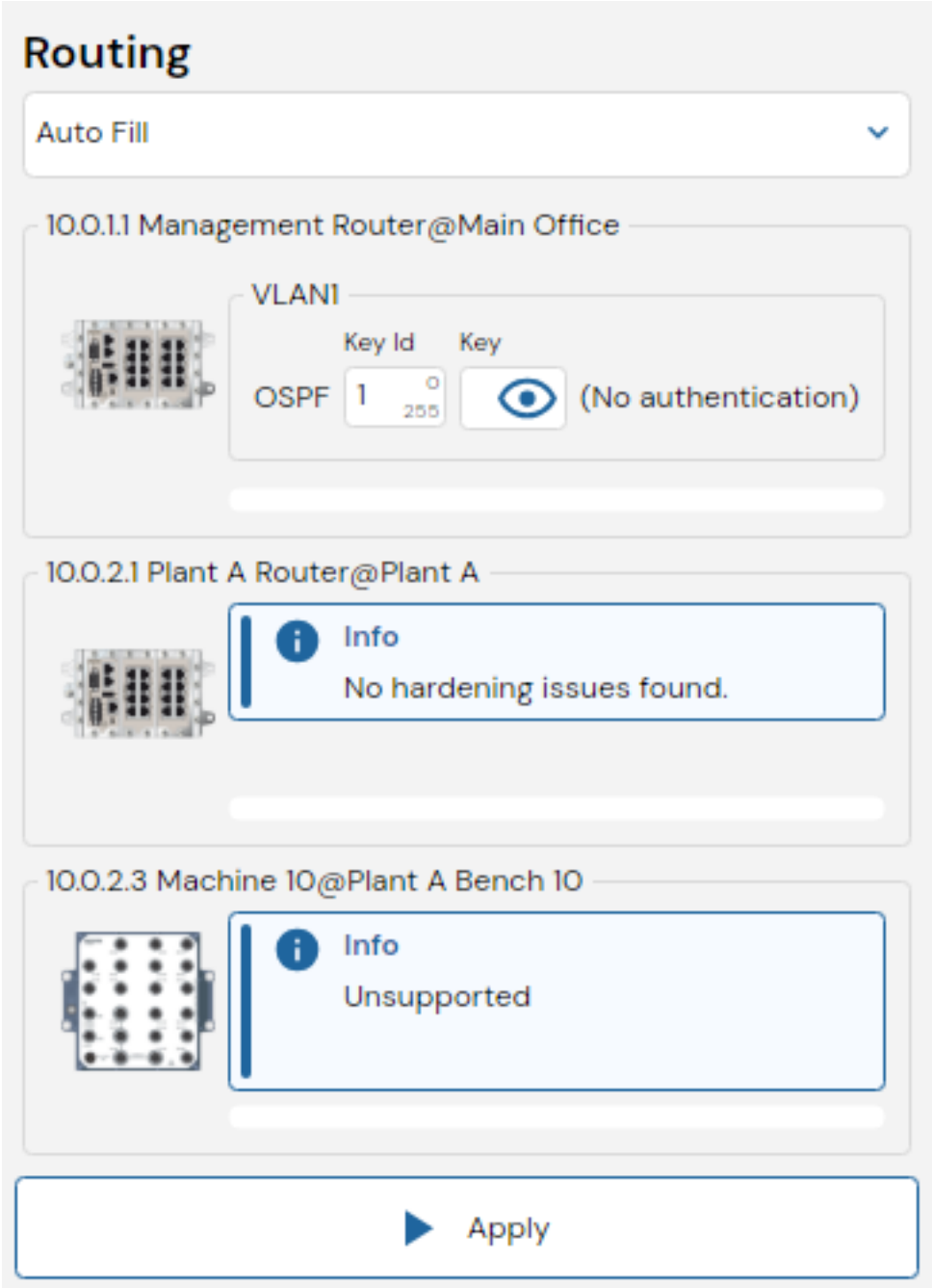
Column	Description
MAC	Whether this port has any MAC filter configured
802.1X	Whether any 802.1X port protection has been configured on this port
Enabled	Whether or not this port is enabled
Unused	Whether or not this port is currently in use
Link Up Alarm	If a Link Up alarm is configured on this port

4.4.49.3. Routing Hardening

WeOS 4 WeOS 5

Routing hardening is a task that allows for detection of some security issues related to routing.

Interface components



With this panel it is possible to scan all or the selected devices in the project that are configured to be routers. It detects OSPF or RIP router settings that do not use MD5- HMAC to sign routing traffic.

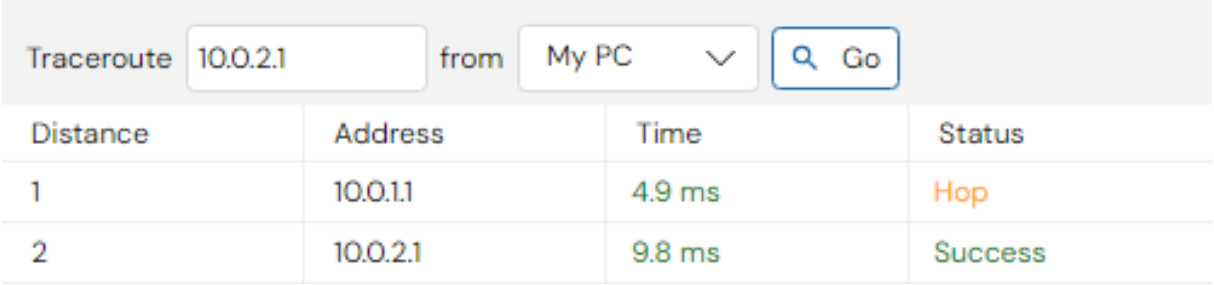
When Scan is clicked, each device in the project will be interrogated to see whether there are router configurations that do not use MD5-HMAC signatures.

A presentation of each device with an issue, all VLANs and all routing protocols that do not use MD5- HMAC.

Then it is possible to enter the key ID and key for each device/VLAN/protocol combination. The Autofill section can be used to great effect for this, if there are many devices.

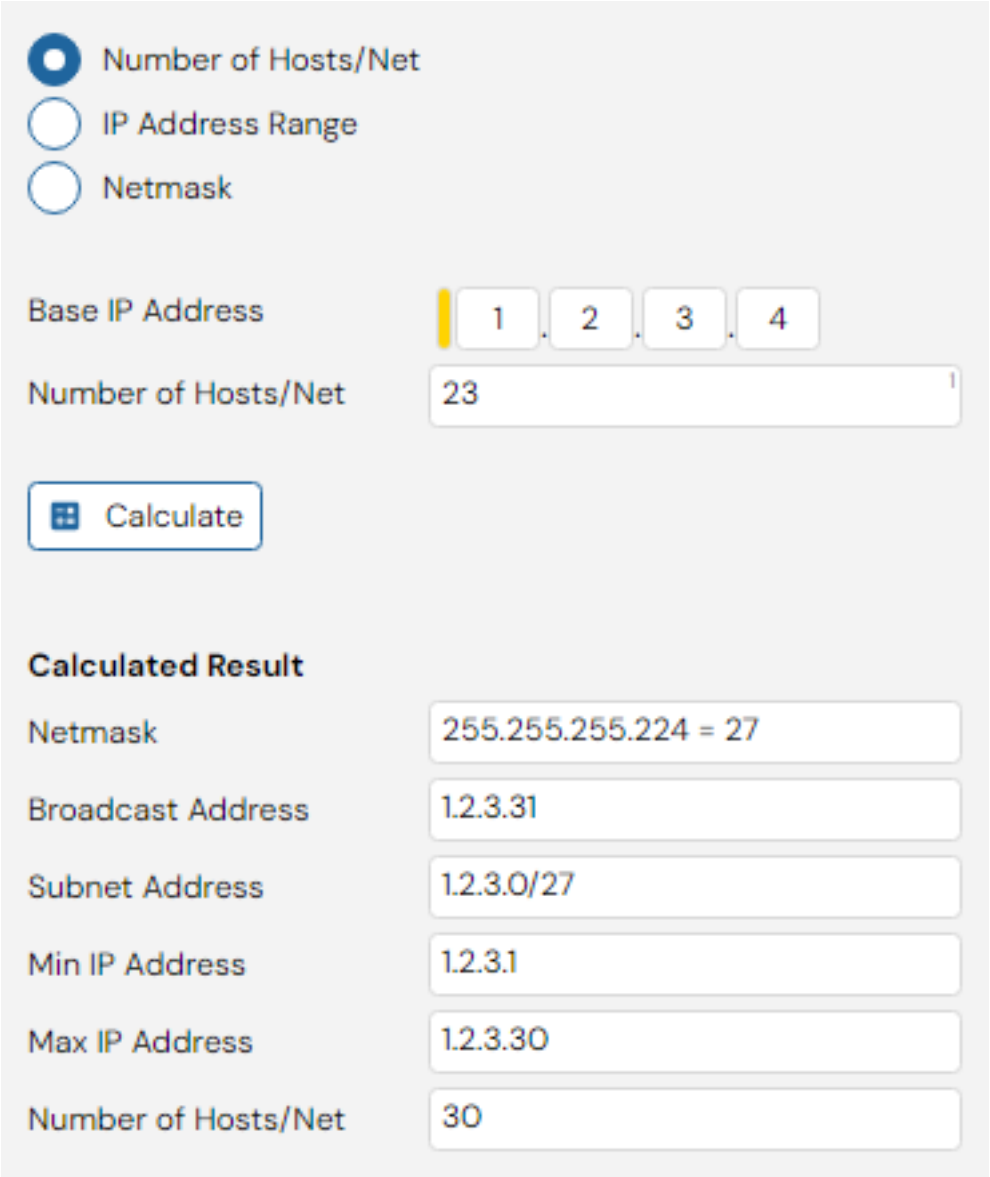
4.4.50. Traceroute

WeOS 4 WeOS 5 Merlin



This panel allows the user to either trace the route between devices. In order to run a traceroute, fill in the target IP address or url in the leftmost box and select the source from the dropdown box to it's right, which may either by the system running WeConfig, or any WeOS device in the network. To trace the route between two devices, the originating device must be a WeOS device.

4.4.51. Subnet Calculator



This panel is a tool to help the user calculate a subnet based on one of the following:

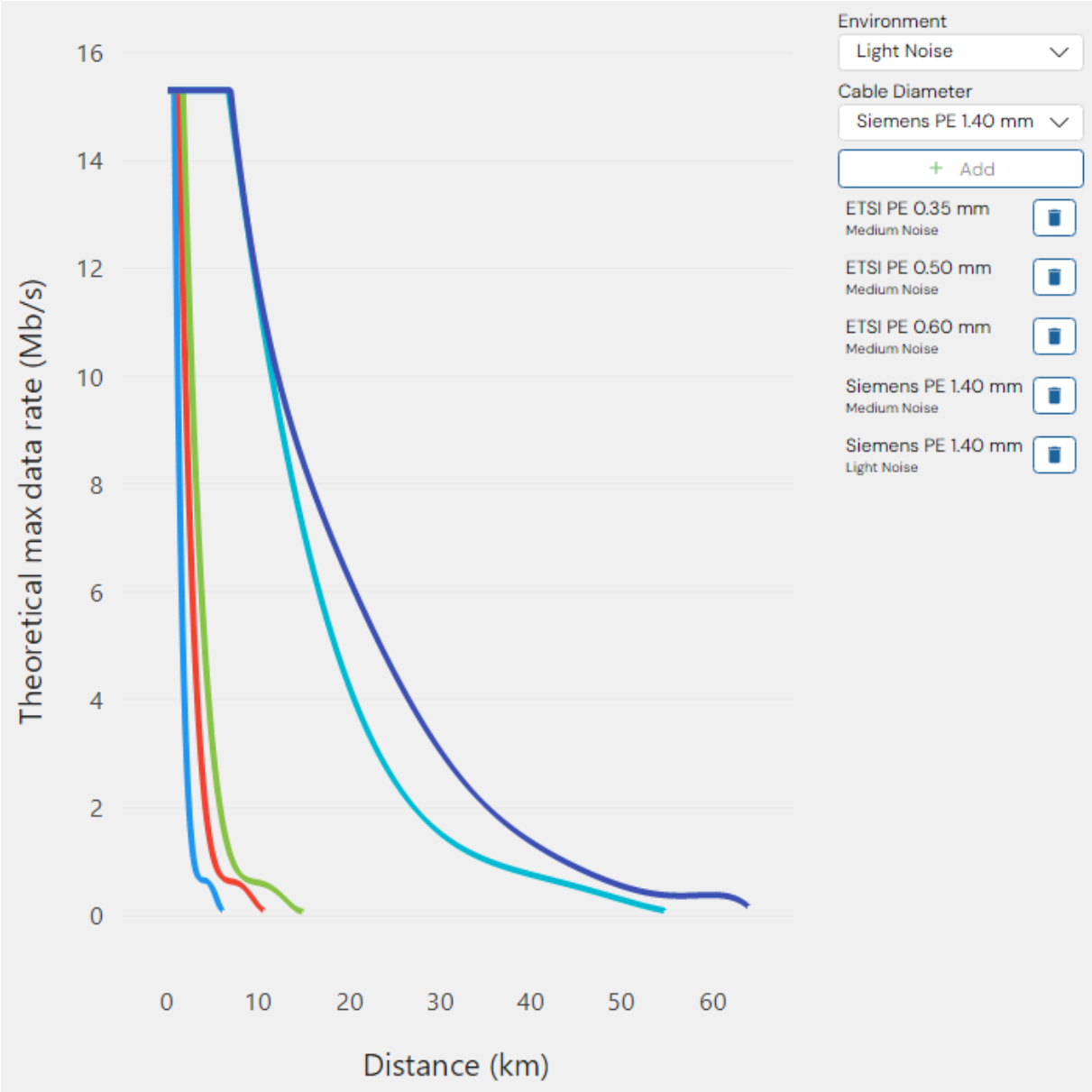
- IP address and Number of wanted hosts/net
- IP range
- IP address and a given netmask.

In order to utilize this panel, select the calculation variant you desire to use from the radiobutton list at the top of the panel, fill in the necessary data in the boxes below, and then press the “Calculate” button.

You will then be presented with the following data:

- Netmask / Bit count (255.255.255.224 and 27 respectively in the example above)
- Broadcast Address (1.2.3.31 in the example above)
- Subnet address (1.2.3.0/27 in the example above)
- Minimum IP address (1.2.3.1 in the example above)
- Maximum IP address (1.2.3.30 in the example above)
- Number of hosts/nets (30 in the example above)

4.4.52. SHDSL Reach Calculator

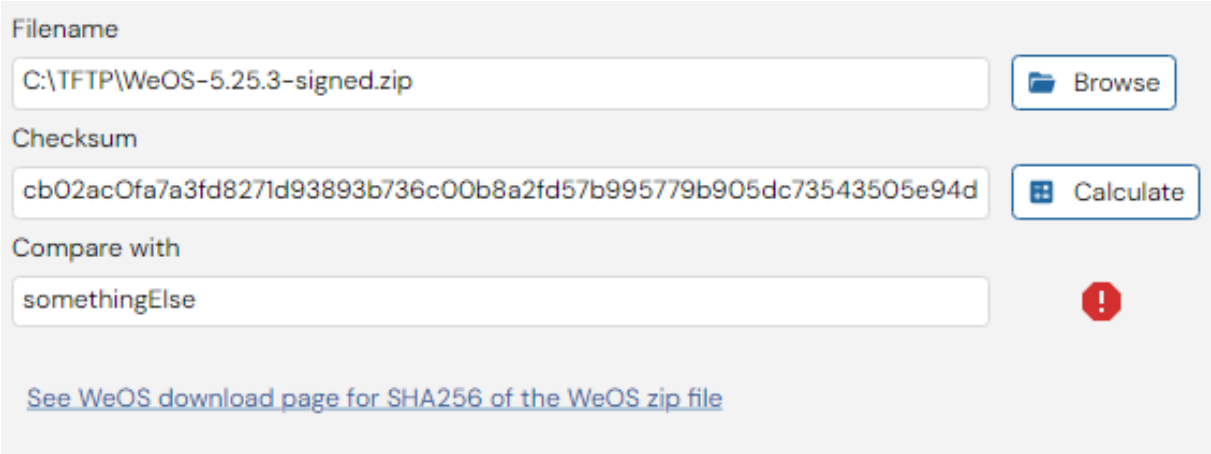


This panel allows you to explore indicative signal attenuation and data rates for the two parameters, environment and cable.

Select a combination of environment and cable parameters in the dropdown menu to the right and click Add. The combination will then be charted in the main graph of the panel.

The Y axis represents a theoretical maximum data rate in Mb/s, and the X axis represents distance in kilometres. To remove a graph line, click the associated Delete (Trashcan) button below the dropdown menus and add button.

4.4.53. SHA256 Hash Calculator



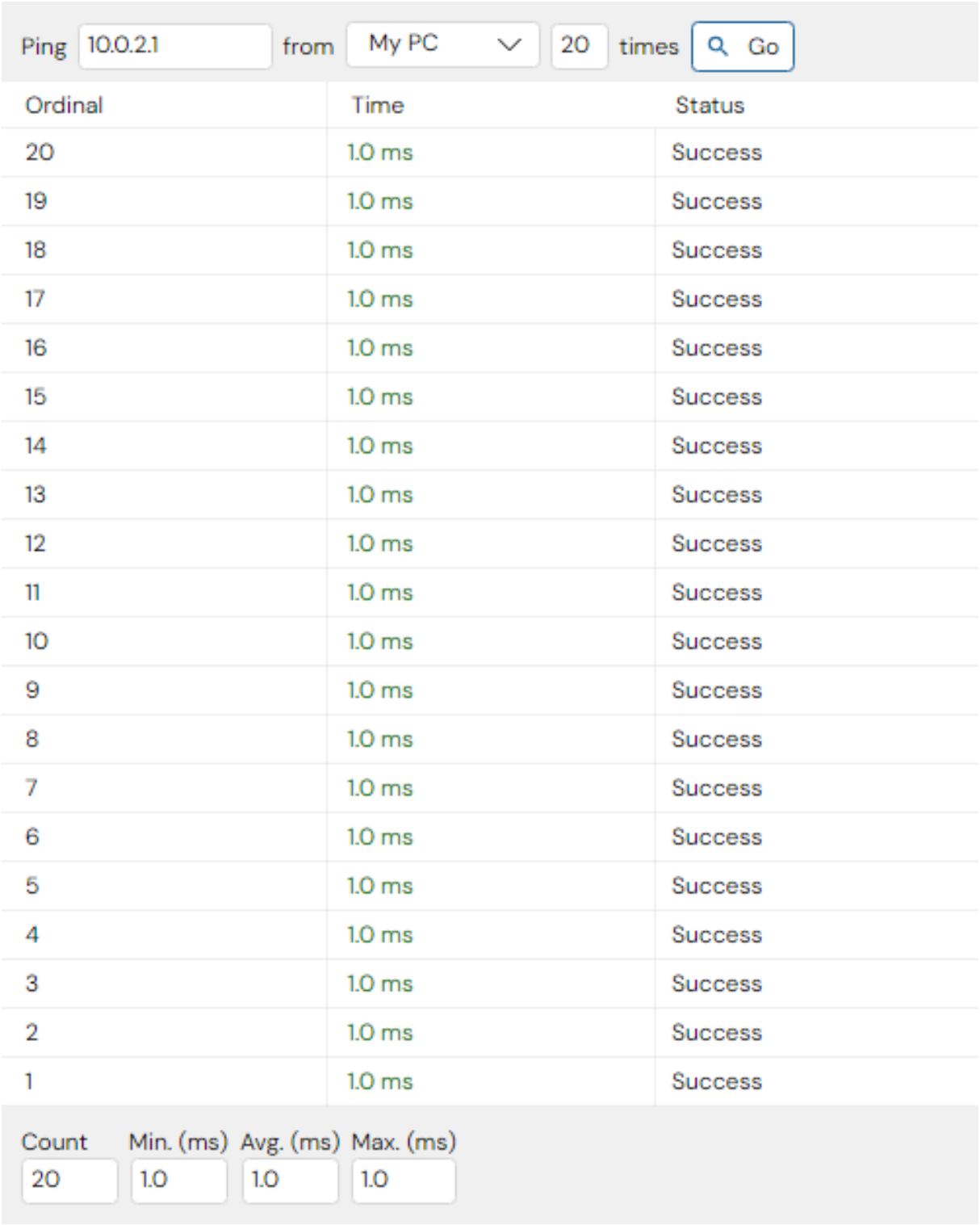
This panel is a tool to calculate SHA256 hash for a selected file, usually WeOS firmware. In order to calculate the SHA256 hash of a given file, user the browse button to select the desired file, or enter the path manually, and then press “Calculate”. The SHA256 hash will then be displayed in the box labeled “Checksum”. As seen in the above example, calculating the checksum of the file at `C:\TFTP\WeOS-5.25.3-signed.zip` yields a hash starting with `cd02ac...`

Additionally, a known SHA256 hash may be entered into the “Compare with” textbox to quickly check equality between it and the calculated SHA256 hash. As seen in the example above, a failed equality check between the calculated hash and the string `somethingElse` is indicated by a red circle containing an exclamation point.

The panel also contains a convenient link to a list of SHA256 hashes for various WeOS versions.

4.4.54. Ping

WeOS 4 WeOS 5 Merlin

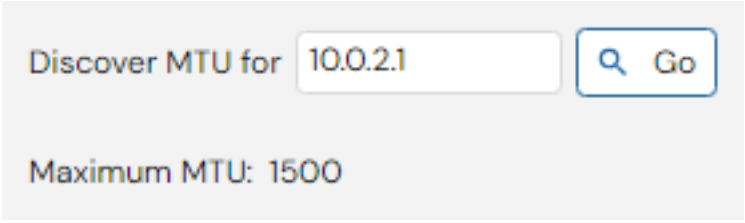


The Ping panel allows the user to ping a device from another device. In order to execute a ping, fill in the target address or url in the topmost leftmost box, and then selected the source from the dropdown list to its right, the source may be either the system running WeConfig or any known WeOS device in the network that is not of the Falcon model family. Finally, enter the number of times to ping the target and press “Go”.

The results will then be displayed in a table, as depicted in the example above, contain the ordinal, roundtrip time and status for each ping in the sequence.

The data is also summarized at the bottom of the panel by count, minimum, average and maximum roundtrip time.

4.4.55. Maximum MTU Discovery



This panel allows the user to find the maximum MTU size between the WeConfig PC and a target device. In order to discovery the maxmimum MTU size, enter the target device address or url into the indicated box and press “Go”.

The maximum MTU size will then be displayed below, as seen in the example above.

4.4.56. Project Template

4.4.56.1. Import Devices from CSV File

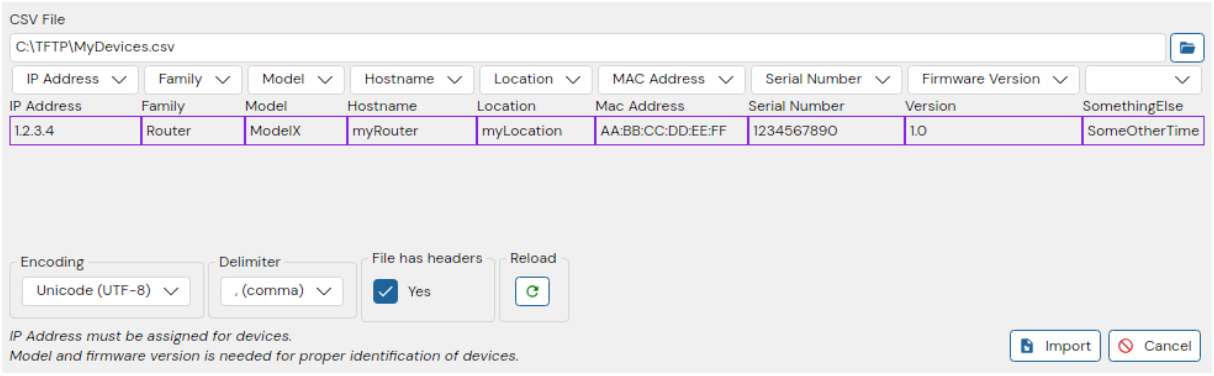
WeOS 4 WeOS 5 Ibex XRD Merlin other

Import Devices from CSV File allows for importing a project from a Comma Separated Value (CSV) file.

To import devices into the project from a CSV file browse for the CSV file and specify encoding, delimiter, and whether the file has headers or not.

The defaults are often good enough. Use the combo boxes to specify which column in the CSV file should map to which device attribute. Click Import to start.

> **Tip**
>
> If the CSV file has headers, WeConfig will attempt to map columns based on the header names. If no headers are present, the columns must be mapped manually.



The CSV file must have the following columns:

- IP Address (In the format of IPv4, e.g 1.2.3.4)
- Family
- Model

Other following columns are optional:

- Hostname
- Location
- Mac Address (in the format of `XX:XX:XX:XX:XX:XX` or `XX-XX-XX-XX-XX-XX` or `XXXXXXXXXXXX`)
- Serial Number
- Version

Additional columns are ignored.

WeConfig will attempt to map columns based on header names if the CSV file has headers. If no headers are present, the columns must be mapped manually.

Below are found a few examples of valid CSV files.

```
IP Address,Family,Model,Hostname,Location,Mac Address,Serial
Number,Version,SomethingElse
1.2.3.4,Router,ModelX,myRouter,myLocation,AA:BB:CC:DD:EE:FF,1234567890,1.0,SomeOtherTime
```

```
1.2.3.4,Router,ModelX,myRouter,myLocation,AA:BB:CC:DD:EE:FF,1234567890,1.0
```

```
1.2.3.4,Router,ModelX,,
1.2.54.25,Routeral,ModelY,,Name
12.2.3.4,Routerei,ModelZ,Home,

13.2.3.4;Router;ModelX;;
14.2.3.4;Routeral;ModelY;;Name
1.25.3.4;Routerei;ModelZ;Home;

111.2.3.4;Router;ModelX;;
1.222.3.4;Routeral;ModelY;;Name
1.2.32.4;Routerei;ModelZ;Home;

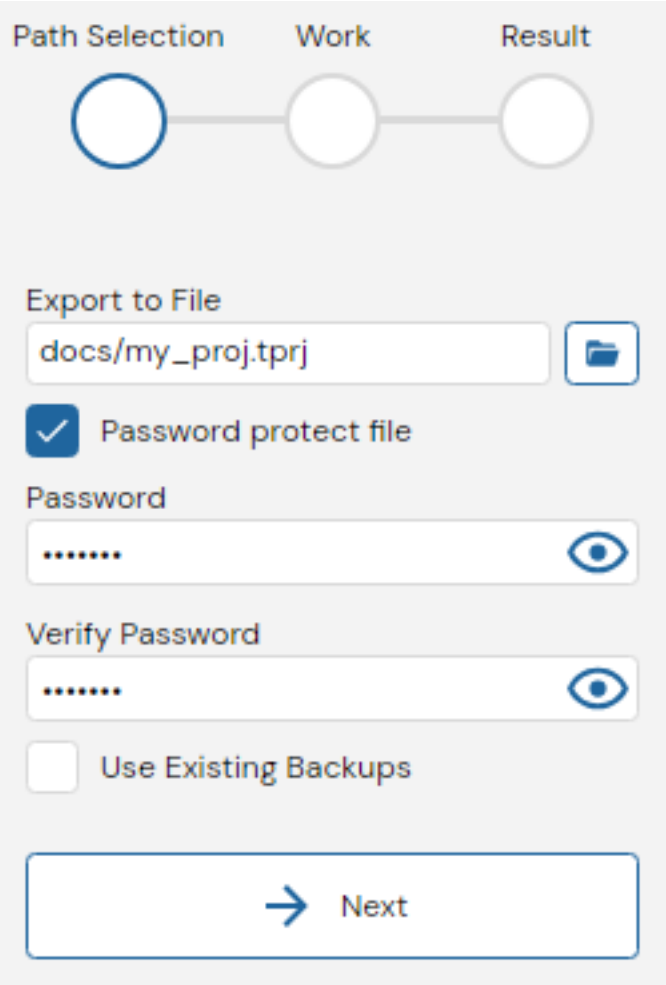
1.22.3.4 Router  ModelX      <empty>
1.2.33.4 Routeral ModelY     <empty> Name
1.2.3.45 Routerei ModelZ     Home
```

4.4.56.2. Export from Current Project

WeOS 4 WeOS 5

Export from Current Project allows for exporting the current project to a Gold File.

Path Selection



In the Path Selection step, you select the path where the project will be saved.

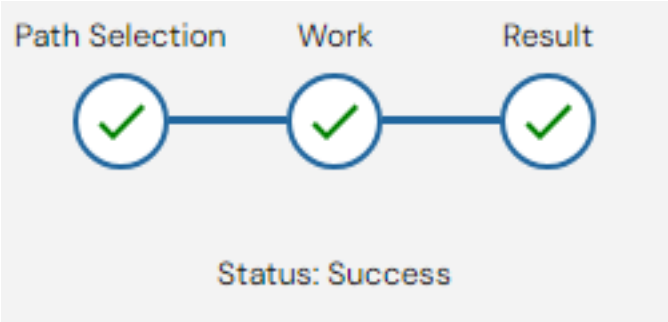
You also have the option to set a password to encrypt the project file.

Press “Next” to start the export process.

> **Note**
>
> To make a template of the current project, the following criteria must be met:
> - WeConfig’s connection in the topology is known
> - All devices must support the gold file functionality

> **Info**
>
> Currently, only WeOS devices are supported.

Result



In the Result step, you can see the progress of the export process.

If the export is successful, you will see a message indicating that the project has been exported successfully.

If the export fails, you will see an error message indicating the reason for the failure.

> Note
>
> The export process may take some time, depending on the size of the project and the number of devices in the topology.

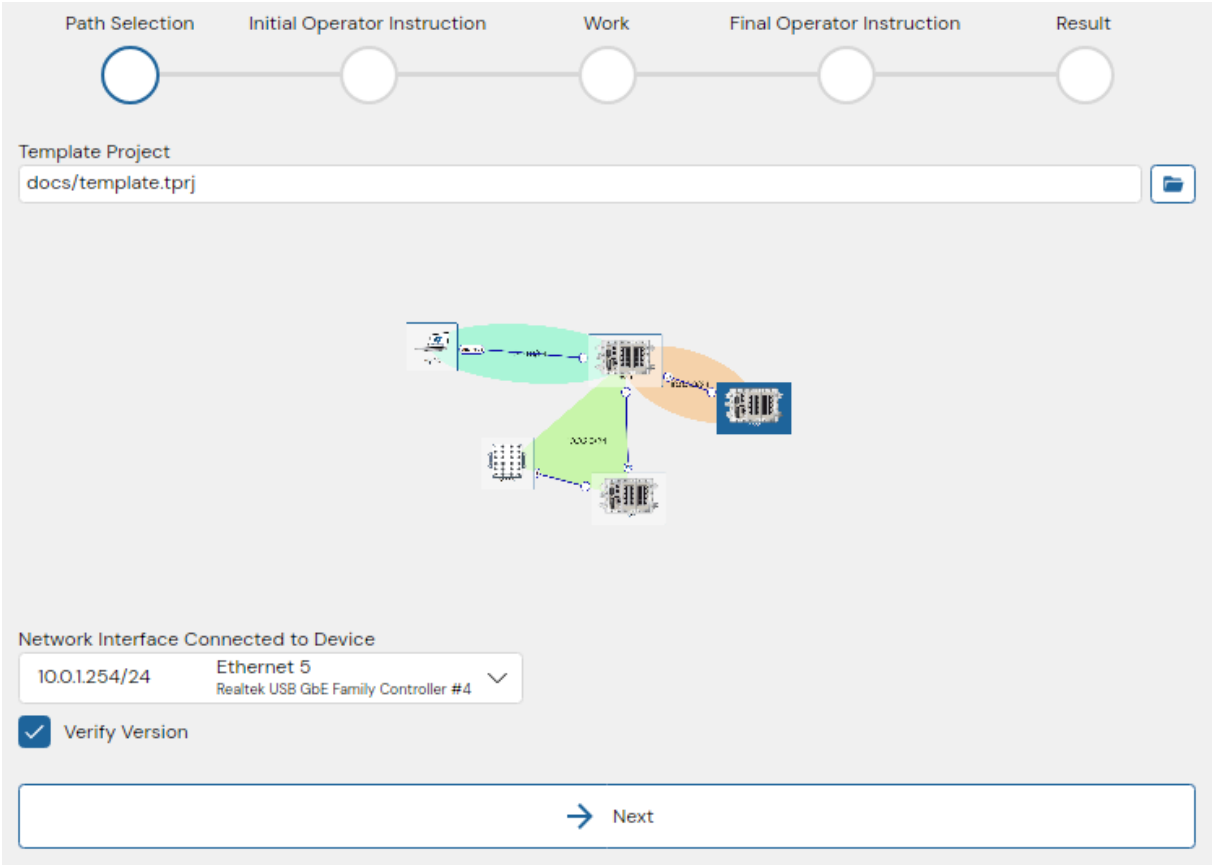
4.4.56.3. Build Network from Template

WeOS 4 WeOS 5

Build Network from Template allows for importing network configuration from a Gold File.

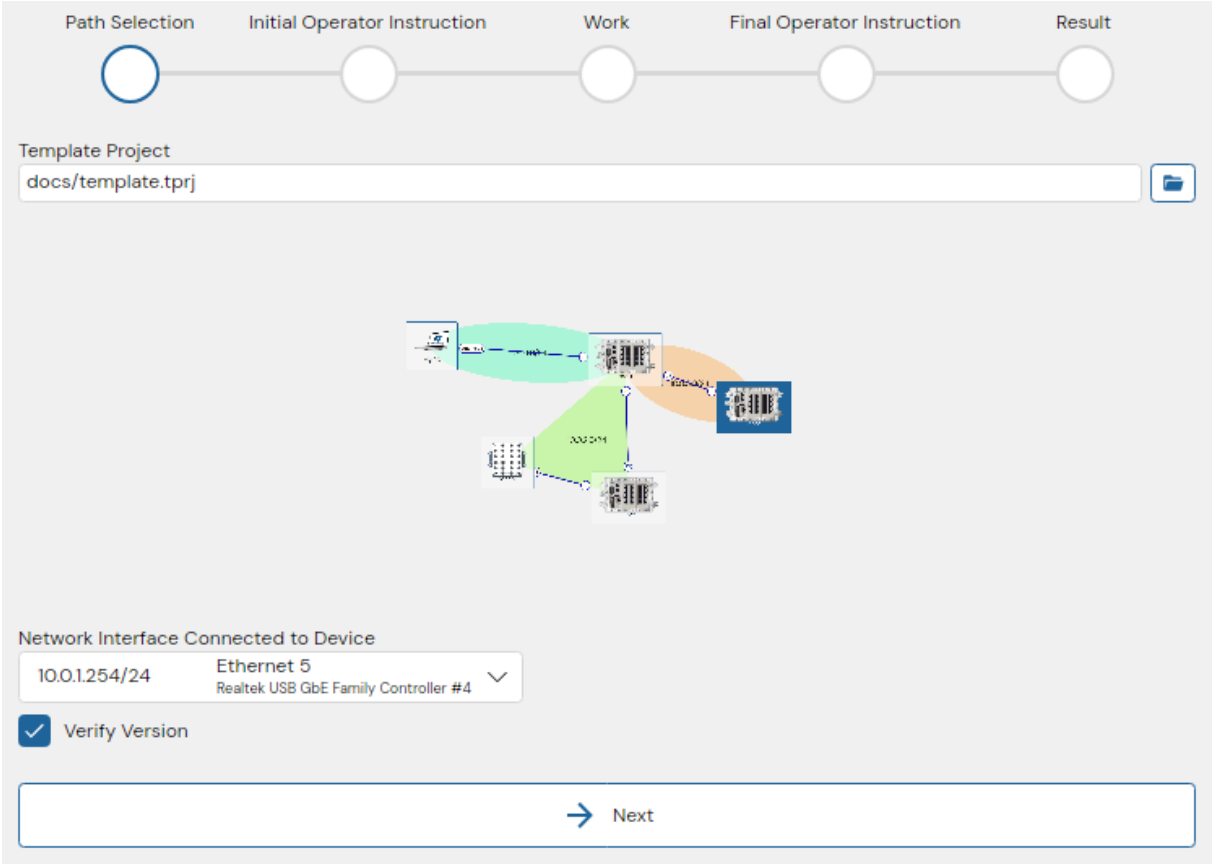
> **Info**
>
> A known limitation of the import process is that it requires there to be some unused IP Addresses.
>
> These temporary IP Addresses are used to avoid IP Address conflicts during the import process.
>
> The number of temporary IP Addresses required can vary depending on how many conflicts there are in the network. From zero to one for each device.

Interface Components



The build-network-process is wizard-based. It will guide and inform what to do to complete the operation.

Path Selection

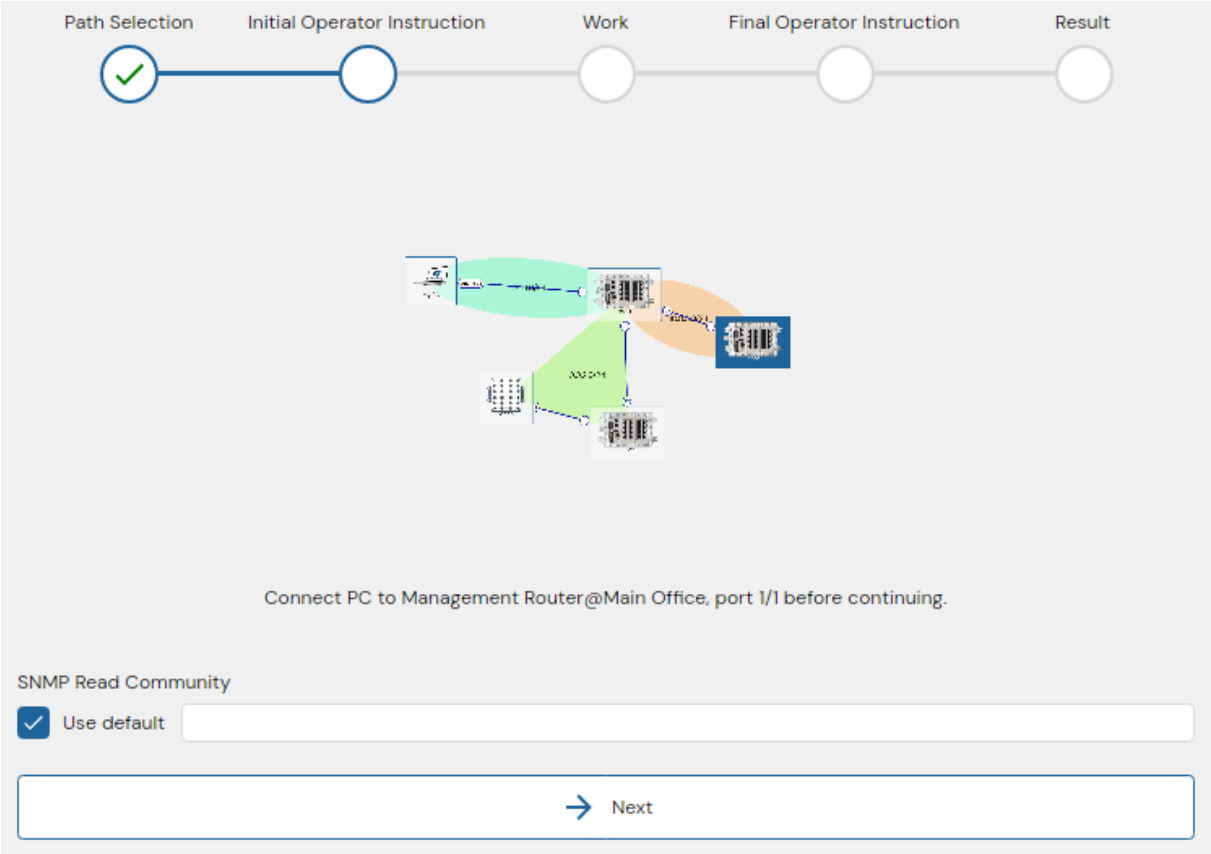


Browse and select the gold file template which will then be applied on the network.

Select the Network Interface that is connected to the device.

If the project template file is password protected, you will be asked to provide the password.

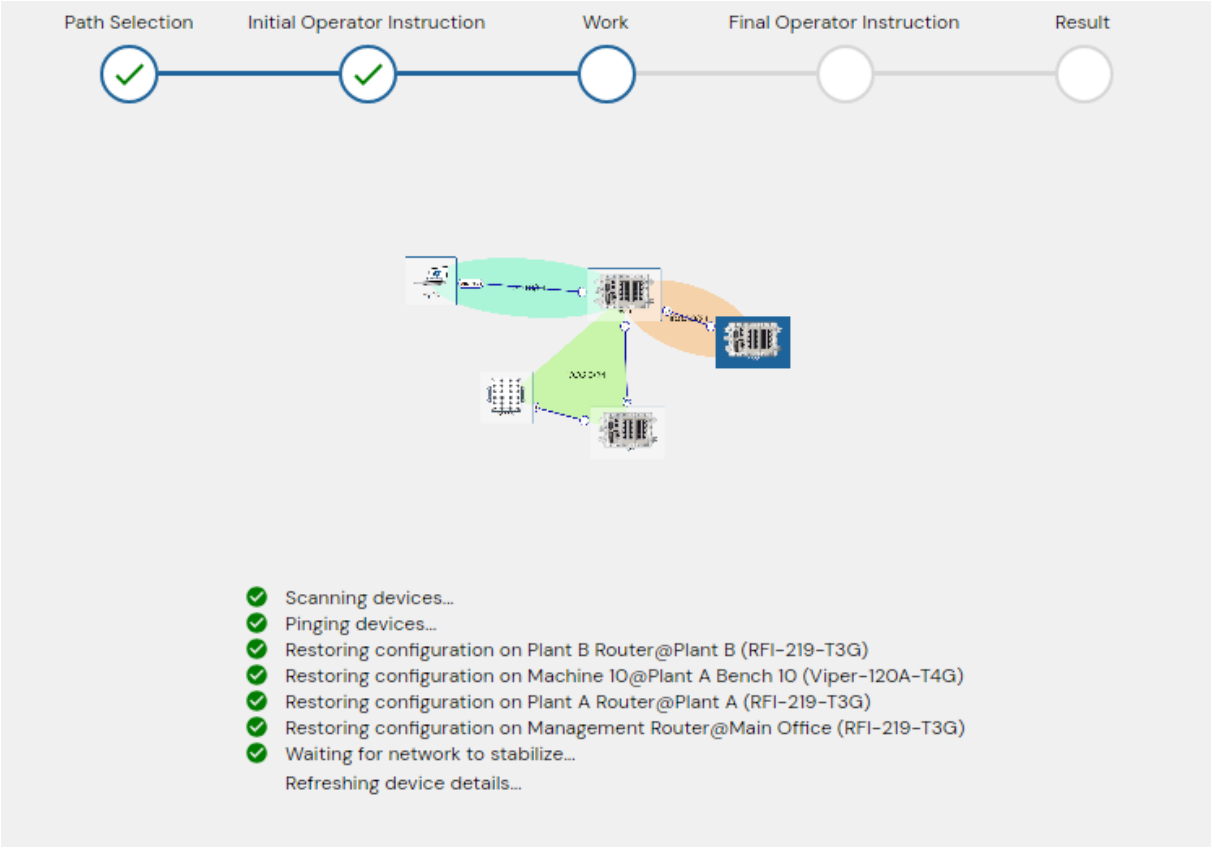
Initial Operator Instructions



This step will provide instructions you must follow before the import can start. Such as connecting the PC to the correct device on the correct port, and providing an SNMP Read Community.

> **Note**
>
> If there aren't any instructions, the step will be skipped.

Work

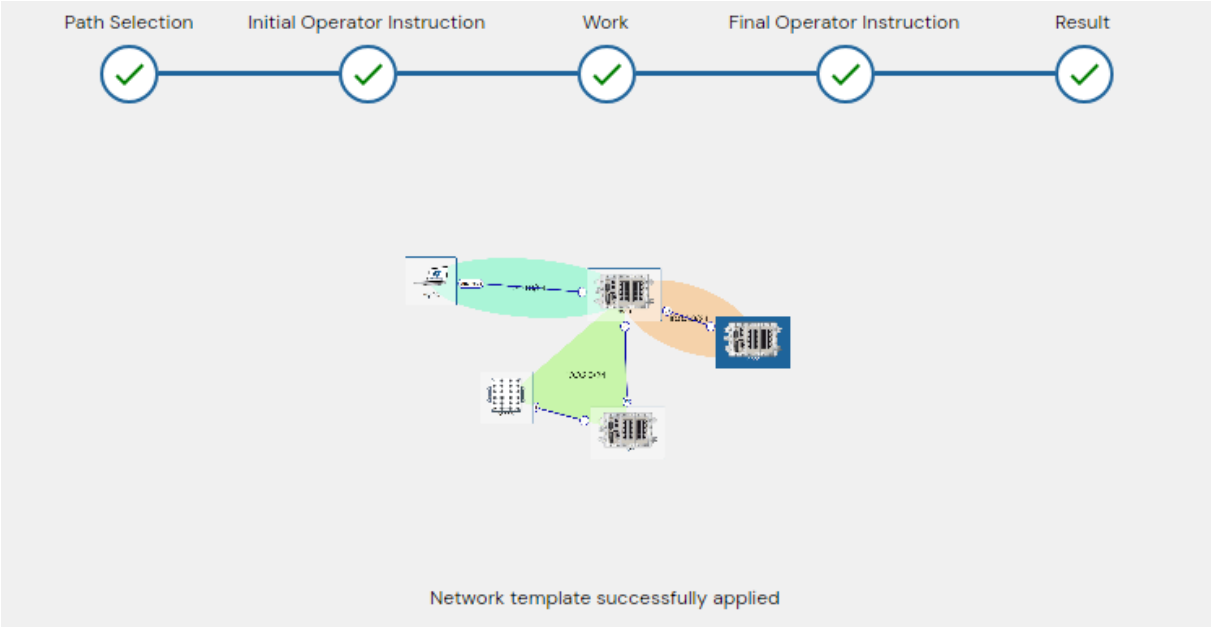


The work step will show the progress of the import process.

Final Operator Instructions
This step will provide instructions you must follow after the import is complete. Such as adding cables between devices.

> **Note**
>
> If there aren't any instructions, the step will be skipped.

Result



This step will show the result of the import process.

If the import is successful, you will see a message indicating that the project has been imported successfully.

If the import fails, you will see an error message indicating the reason for the failure.

4.4.57. Refresh

The third button at the top of the navigation menu runs what is called a Refresh operation on the entire network. A refresh operation entails that all affected devices are rescanned to update WeConfigs knowledge of their current state.

A refresh operation can also be triggered on individual devices via the Context Menu and through various other means through the software.

4.5. File Menu

The file menu contains a variety of commands, settings and options for WeConfig. It is divided into three submenus, File, Edit and Help:

File

The File menu contains commands related to project management, application settings, and exiting the application. It has the following commands:

Command	Hotkey	Explanation
New	Ctrl-N	Creates a new, empty project
Open	Ctrl-O	Opens an existing project from the file system
Recent Projects		Shows a list of recently opened projects, clicking a project opens it
Save	Ctrl-S	Saves the currently open project, requesting a save location if needed
Save As	Ctrl-Shift-S	Saves the currently open project, always requests a save location
About		Navigates to the about menu.
Exit	Alt-F4	Exits WeConfig

Edit

The Edit menu contains commands related to editing the current project or its contents. It has the following commands:

Command	Hotkey	Explanation
Create portable installation		Generate a portable installation of WeConfig
Settings		Navigates to the application settings menu.

Help

The Help menu contains commands related to obtaining help and information about WeConfig. It has the following commands:

Command	Hotkey	Explanation
User's Guide		Opens this user guide
Support file		Creates a support file to help developers identify the source of encountered issues
Reset Layout		Restores the document panel to a default, empty state.

4.5.1. About

This dialog contains additional information about WeConfig, it is divided into five sections, the about page, the licensing information, the EULA, the synchronized data status, and firmware information.

About page

This section indicates the running version of WeConfig.

Licensing

This section contains a document that lists the applicable licensing and copyright terms for all Free/Libre Open Source Software, FLOSS, included WeConfig.

EULA

This section contains a document detailing the End user license agreement (EULA) that applies to WeConfigs usage.

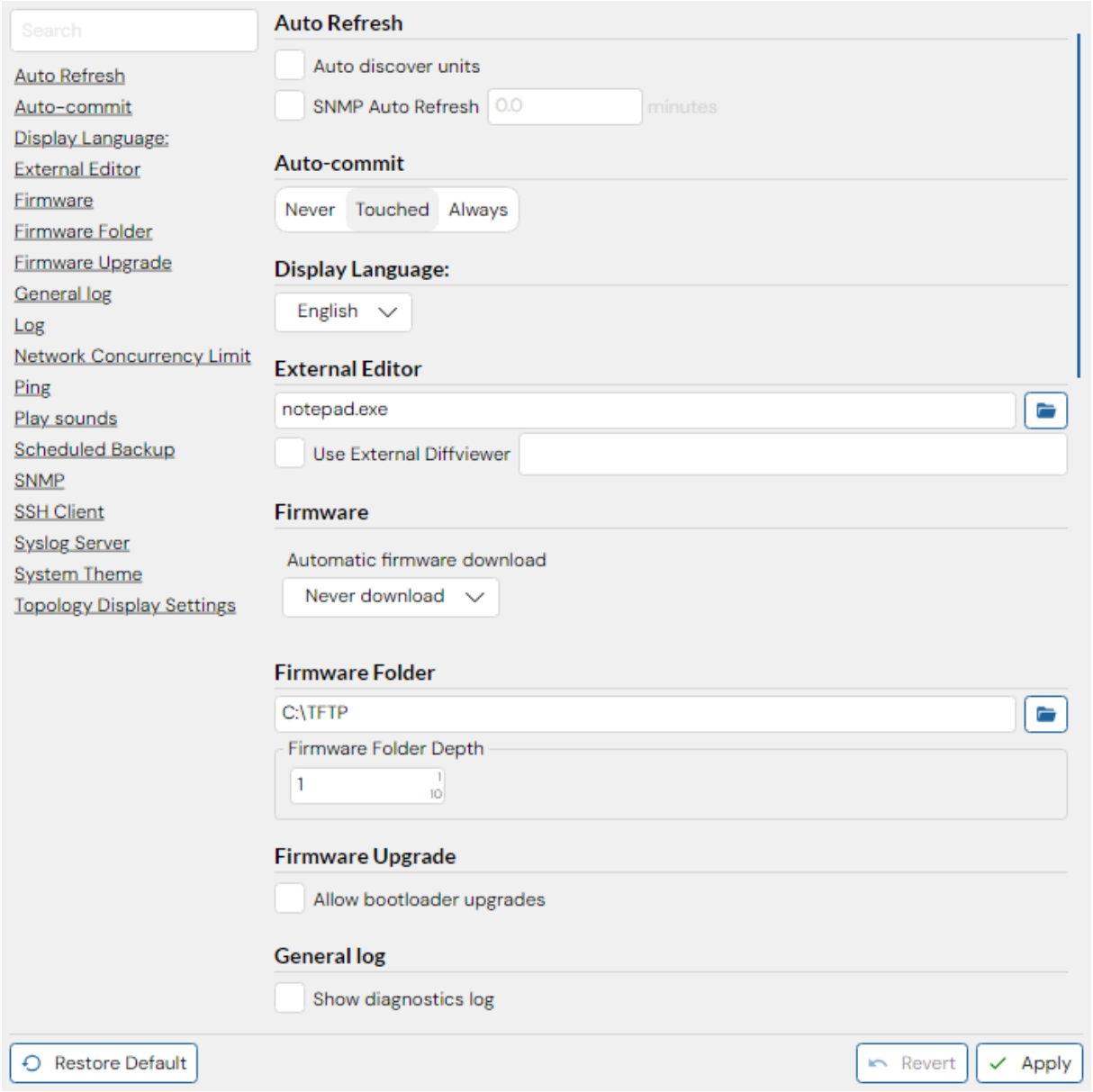
Sychronized data status

This section contains information regarding the current synchronization status with cloud resources such as device icons or OUI databases.

Firmware

This section contains information regarding the availability of device firmware versions.

4.5.2. Application Settings



This view contains a number of settings that affect the WeConfig application globally, regardless of the current working project. These settings are grouped for convenience into relevant sets where applicable. Any group that has a pending change is highlighted in the interface (as seen with “Syslog Server” in the image above), and changes are applied when the user clicks the Apply button at the bottom right of the interface.

Application Settings opens as a dockable tab and is also available from the navigation panel.

Auto-refresh

This group deals with options that allow WeConfig to execute certain network tasks in the background, automatically, and contains two options, Auto discover units and SNMP Auto Refresh.

Option	Description
Auto discover units	When checked, WeConfig will automatically attempt to discover devices that let themselves be detected via LLDP.
SNMP Auto Refresh	When checked, WeConfig will automatically, periodically according the configured value, run SNMP-based scanning on existing devices, and update their information accordingly.

Automatic update policy
This setting group controls how, if at all, Weconfig will update itself after installation. If enabled, when a new version is ready to run, a notification is shown in the user interface, prompting the user to restart with the new version.

The following options are available:

Option	Description
Automatic	Continually check for new versions, download, and automatically update the installation at first opportunity.
On Approval	Continually check for new versions, but let the user decide whether the update should be downloaded and installed.
Never	Never check for any new versions

> *i* Info
>
> Automatic and On Approval will require internet connectivity to function properly.

Auto-commit
This setting controls the behavior of “Auto-commit”, which controls when/if WeConfig automatically commits changes to the startup configuration on applicable devices without requiring manual intervention. The toggle has the following settings:

Setting	Description
Never	Auto-commit is never performed, and changes must always be manually committed.
Touched	Auto-commit is performed after WeConfig has made a configuration change to the device. I.e when the user has performed “Basic Setup” or “VLAN” configuration, the changes are automatically committed.
Always	Auto-commit is performed as soon as WeConfig detects a change between running and startup configuration on applicable devices. This includes any changes made through “Basic Setup” or “VLAN” configuration, as well as any other configuration changes detected by WeConfig, such as detecting a difference between running and startup at first discovery.

The default setting is “Touched”.

Display Language
This setting allows the user to change the user interface language. Supported languages:

- English (default)
- German
- French
- Chinese
- Spanish

External Editor

This setting group contains options related to external software that WeConfig may call. It has two options:

Option	Description
External editor	When WeConfig needs to open a text file, it will use the editor specified here. The default is `notepad.exe` on Windows. On Linux, WeConfig uses `$EDITOR`, then `$VISUAL`, and falls back to `xdg-open`.
External Diffviewer	When the user requests to see a difference between two configuration files, this option, when checked, allows the user to bypass the built in diff viewer functionality, and use custom one instead. Enter the path to the diff viewer application in External Diffviewer. WeConfig will call the application with two command line arguments. The first argument will be a path to the previous revision of the configuration file, and the second argument will be a path to the current revision.

Firmware

This group contains settings related to how WeConfig download and store device firmwares automatically on the user's local hard drive as they become available. By default this is turned off, but may be configured in the following way:

- No downloads at all (for any device type)
- All, as soon as possible (for all device types)
- Customized per device type
  - No downloads at all
  - All
  - Latest (WeOS only - downloads the latest patch version for each minor version)
  - Any firmware newer/published after than a specific date
  - Any firmware with a version higher than a specific version

Firmware Folder

This setting group contains options related to how WeConfig picks up user-specific firmware. It contains two options, the first is a path where user specified firmware can be found use during firmware upgrade. It can be anywhere, as long as the directory is accessible by the current user. The second is the Firmware folder depth

The default folder is `C:\TFTP` on Windows and `/tftpboot` on Linux.

Firmware Folder Depth

By default, WeConfig will only find the files found in the specified firmware folder. If you sort your firmwares in a directory structure, then increase the folder depth to let WeConfig find files in subfolders. A level of 1 is the firmware folder only, a level of 2 is the firmware folder, and all direct subfolders. A maximum of 10 levels is supported.

Firmware Upgrade

This setting group contains options related to how WeConfig executes firmware upgrade. WeOS firmware packages are typically delivered as one unit, and contains both firmware and bootloader software. WeConfig will upgrade the bootloader as necessary, during the firmware upgrade process. Some advanced scenarios require bootloader upgrades only without upgrading the firmware. By enabling this option, it is possible to perform bootloader upgrades only in the firmware upgrade process.

General Log

This setting group contains options related to the general log panel, which may be enabled by the checkbox at the top of the group. This panel is mainly for use by developers and power users, but is generally not available. The second option herein is a specifier for the length of the general log that will be kept and displayed.

Log

WeConfig logs its activity to files and, if enabled, as application telemetry. For installed Windows builds, logs are under `%LOCALAPPDATA%\Westermo\WeConfig\Logs\Application`; for installed Linux builds, they are under `~/.local/share/Westermo/WeConfig/Logs/Application`. Portable builds store logs in the application's `Data/Logs/Application` directory. The amount of information logged is determined by this slider.

The following options are available:

- Nothing
- Errors
- Errors and Warnings
- Errors, Warnings and Information (Default)
- Errors, Warnings, Information and Debug.

It is suggested to not change this setting from it's default value unless directed to do so by Westermo support.

Network concurrency limit

This setting allows the user to specify how many concurrent network operations WeConfig may execute at any given time. Increasing this value may speed up network tasks, but may also lead to network congestion and packet loss on slower networks. Decreasing this value may improve stability on slower networks, but will also slow down network tasks. The default value is 32 concurrent operations.

Ping settings

This setting group contains options related to how WeConfig performs ICMP ping operations. It contains the following options:

Option	Description	Default
Timeout	The amount of time, in milliseconds, to wait for a ping reply before considering the ping failed.	200 ms
Period	The amount of time, in seconds, WeConfig will wait between each background ping attempt.	30 s

Option	Description	Default
Period (After status change)	The amount of time, in seconds, WeConfig will wait between each background ping attempt after a device's ping response status has changed.	5 s

Play sounds
When enabled, WeConfig will play sounds to grab the user's attention.

Scheduled backup
These configuration options allows you to set up an automatic, periodic backup of all supported devices. You can configure the start time where the initial backup may be taken, as well as the number of days between backups.

> **Note**
>
> WeConfig must be running for the backups to occur.

SNMP Read Community
These settings provide the fallback SNMP read community, port, and timeout used for devices without specific SNMP settings in Device Access. Defaults are `public`, port `161`, and a timeout of `4000` milliseconds.

SSH Client and Parameters
When the user wants to access a device using an SSH client, WeConfig will use the client specified here. By default, WeConfig will use the SSH client built into the OS. To use another, select Custom, and enter the path to the executable.

When using the custom option, WeConfig must know how to specify connection options to the custom application. This is done in the SSH Parameters entry. Suppose a custom client is used like this to connect to a device: `my-ssh.exe user@ip-address`, then WeConfig must know how to replace `user` and `ip-address`. This is done by using variables. Variables are replaced by WeConfig with the actual values. The supported variables in WeConfig are:

Variable	Description
`%a`	The IP Address of the device
`%u`	The user name

In order to support the example client above (`my-ssh.exe user@ip-address`), the field SSH Parameters entry should read `%u@%a`.

Syslog Server
This setting group contains options related to WeConfigs interactions with the Syslog protocol. When enabled, WeConfig will act as a syslog server that can receive syslog encoded messages from the devices. WeConfig will keep a number of syslog messages up to the number of configured lines per device.

Topology Display Settings

This setting group contains options related to how the topology panel displays information. It contains four different options:

Auto Layout Algorithm

When the topology view is requested to auto layout devices and connections in the topology view, it will use the algorithm selected here. Supported algorithms:

Algorithm	Description
Adaptive Layout (default)	A stress-majorization layout solver that takes topology information such as ping distance and rings into account
ISOM	Isometric layout
Tree	Tree layout



Note

Using Auto layout functionality with the adaptive layout algorithm on very large networks (> 1000 devices) may take significant time.

Show Status In Topology

When enabled, the topology view will display current operation status in the device icons in the topology view.

4.5.3. Portable Installations

Located under the Edit menu, this command causes WeConfig to generated a portable replica of itself to the targeted file system, such as the local computer, or a USB drive. Once the portable replica has been generated in this way, it can be started as normal, or copied to a different destination (such as a different computer, in the case of a USB drive) and started there.

The primary use case of this functionality is to create WeConfig installations suitable for air-gapped systems.

Portable installations do not offer short cut icons in the start menu - one must either start the application by double clicking the executable WeConfig.Application.exe or create shortcuts manually.

Limitations

Certain functionalities of WeConfig are limited in portable mode due to it's air-gapped nature, of particular note are the following:

- Automatic or Manual firmware downloads should not be expected to work
- Cloud firmware lists will not be up-to-date.
- Device definitions and icons will not be up-to-date.
- Licenses will not be automatically renewed.

4.6. Components

4.6.1. CIDR Address box



The CIDR box is used in certain places around WeConfig, and is an input field specifically designed to make it quick to enter CIDR addresses. It will also indicate two colors, the lower is the color of the /32 IP address (from the IPv4 Address Box) and the upper is the color of the subnet specified by the CIDR notation.

This is a composite component that combines an IPv4 Address box with a subnet mask selector.

4.6.2. IPv4 Address box

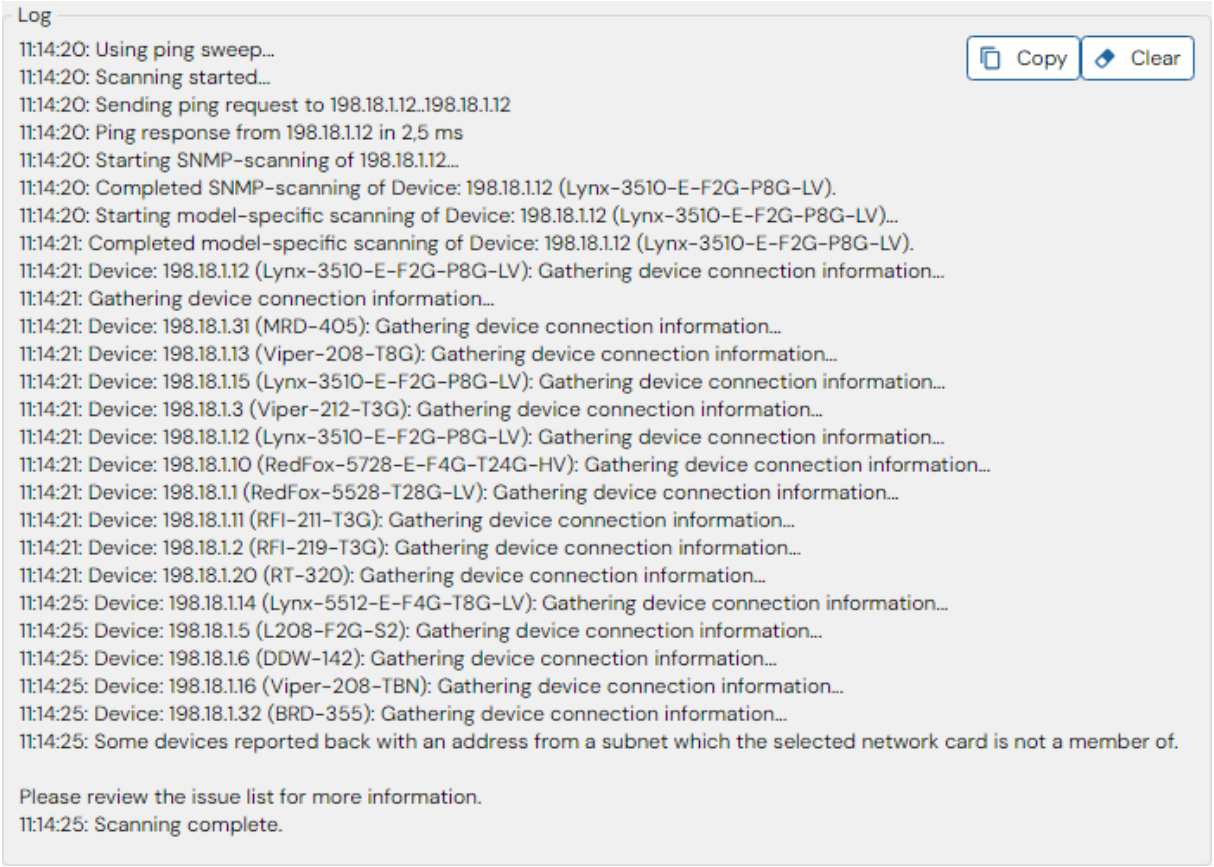


The IPv4 box is used in certain places around WeConfig, and is an input field specifically designed to make it quick to enter IPv4 addresses. It will also indicate a color, which corresponds to the color the corresponding /32 subnet would be rendered with in the topology and diagnostics.

It has a certain numbers of behaviors that are interesting to note:

- It will automatically jump forwards and backwards if it detects that the user has completed or erased the current value in a given segment. For example, typing `192` into the first box would automatically shift focus to the second box, but writing `19` would not, since another character would still be valid in the first box. As such, typing the following keys `12343541` would be sufficient to enter the value `123.43.54.1`
- Pressing `.`, or `Enter` will move focus to the next box.
- Pressing backspace on an empty box will move you to the previous box.
- Pressing Ctrl+C to copy will copy the entire box value in IPv4 format.
- Pressing Ctrl+V with a string in IPv4 format will populate the box.
- Leading zeros are allowed but will be removed. So `198.018.001.000` becomes `198.18.1.0`

4.6.3. Log box



The log box is a component used in WeConfig to display logs and messages. It is designed to show real-time updates and historical logs, making it easier for users to monitor network operations that WeConfig undertakes.

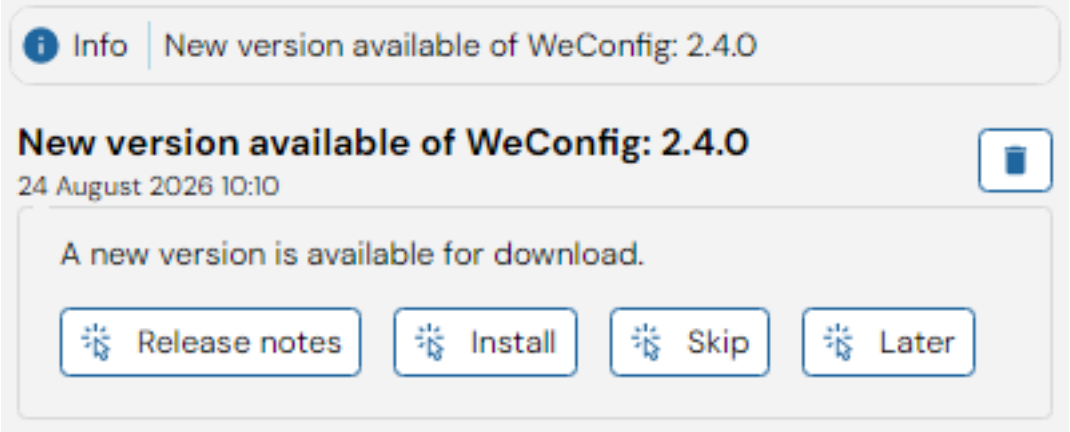
Each log box also comes with two buttons, visible when hovering the box:

- Copy logs: You may copy the contents of the log box to the clipboard by using the leftmost of the buttons in the top right.
- Clear logs: You may clear the log box via the ‘eraser’ icon in the top right of the log box.

4.7. Notifications

WeConfig has a notification system to alert to user of items that requires the users attention such as software updates. All current notifictions are available in the notifications flyout. Every notification can be deleted and most have actions specific to that notification available.

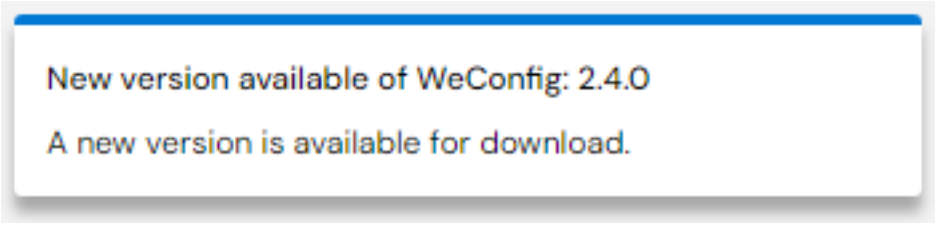
Backup outcomes are shown in the Backup task and are not retained in this notifications flyout. Toasts are brief alerts; dismissing a notification toast does not remove its notification from the flyout.



This menu can be accessed through the “Show/hide notifications” quick action in the title bar. The quick action button also indicates the current nofication count and highest severity.

	Severity	Color
	No notification	None
	Information	Blue
	Warning	Yellow
	Error	Red

When a new notification arrives, a toast appears in the bottom-right corner. Clicking the toast dismisses it, but the notification remains available through the notifications quick action.



5. Algorithms

5.1. Recursive Discovery

Recursive discovery is an algorithm for discovery used by WeConfig that is fundamentally relatively simple. Its general workings are described as follows:

- Given an initial set of devices `D`
- For each device `d` in `D`
  - Attempt to acquire the LLDP, ARP and Route table from `d` and collect all addresses present into a list `N`
    - For each address `n` in `N`
      - If the address does not belong to a known device, attempt to scan that address to produce a new device `m`. If `m` was successfully created, add `m` to `D` and continue iterating.

This algorithm, though simple, will allow WeConfig to reach out across different subnets and recursively discover the network topology for as long as it can comprehend the devices it encounters, and has permission to access them.

Example:
Consider an Discovery sequence with recursive discovery enabled run on `192.168.1.1/32` which so happens to be a WeOS device in this example.

Upon successfully scanning `192.168.1.1`, WeConfig's current picture of the network might look like this:

Step 1



WeConfig will then inspect its LLDP table (which, for this example, will yield the address `192.168.1.2` as connected on the port `Eth 2`), its ARP table (which will produce `192.168.1.2` and `192.168.1.3`) and its route table (which will indicate it has routes going via the gateway `192.168.1.4`).

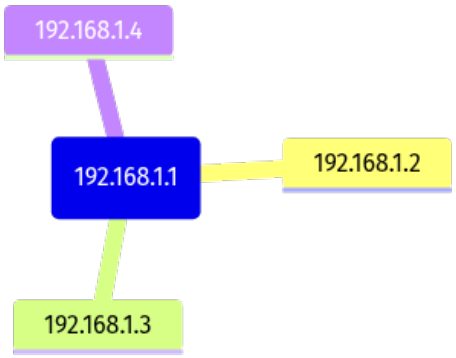
These addresses will then be collected into a distinct list:

- `192.168.1.2`
- `192.168.1.3`
- `192.168.1.4`

With the duplicate entry of `192.168.1.2` being removed.

WeConfig will then move to each of these devices and scan them, finding all of them to be another WeOS device, updating the network to:

Step 2



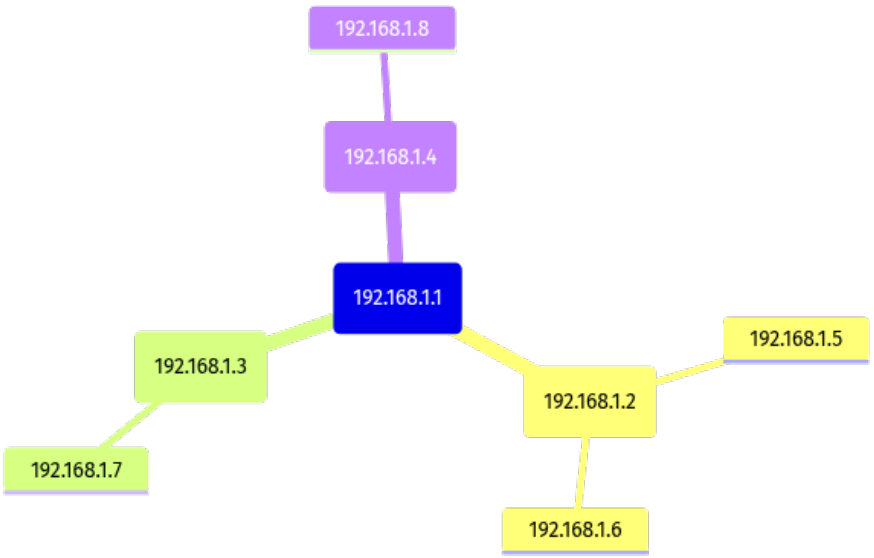
Where the rounded boxes indicate devices whose tables have been inspected.

A similar inspection of each of the new devices various tables yields the following addresses:

- `192.168.1.1`
- `192.168.1.3`
- `192.168.1.4`
- `192.168.1.5` (From `192.168.1.2`)
- `192.168.1.6` (From `192.168.1.2`)
- `192.168.1.7` (From `192.168.1.3`)
- `192.168.1.8` (From `192.168.1.4`)

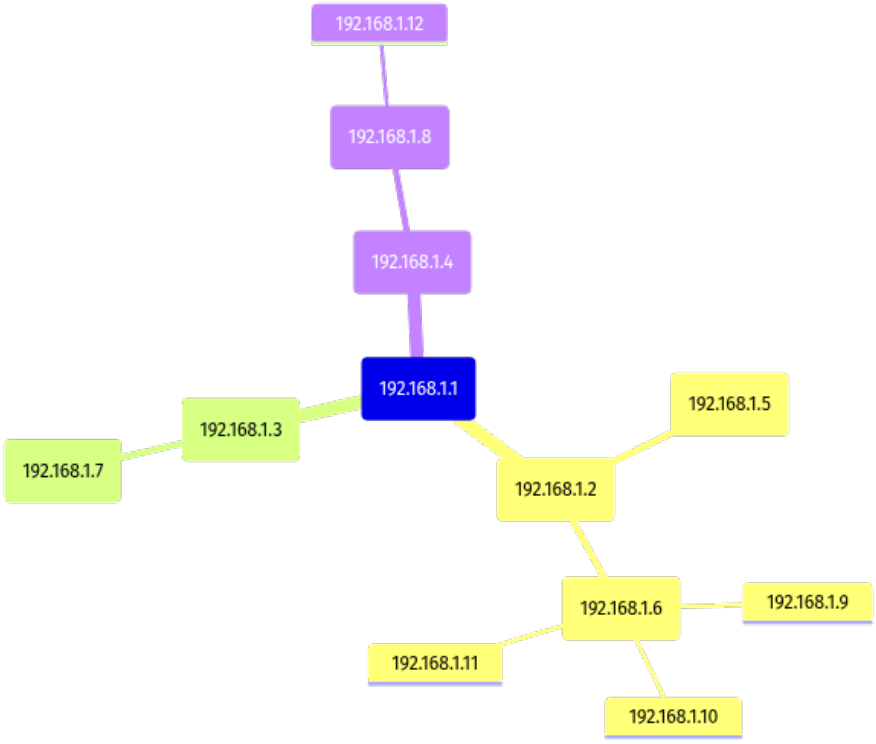
Since the three first addresses have already been found elsewhere, they will be ignored, the new ones will be scanned, and their tables checked, resulting in the network:

Step 3

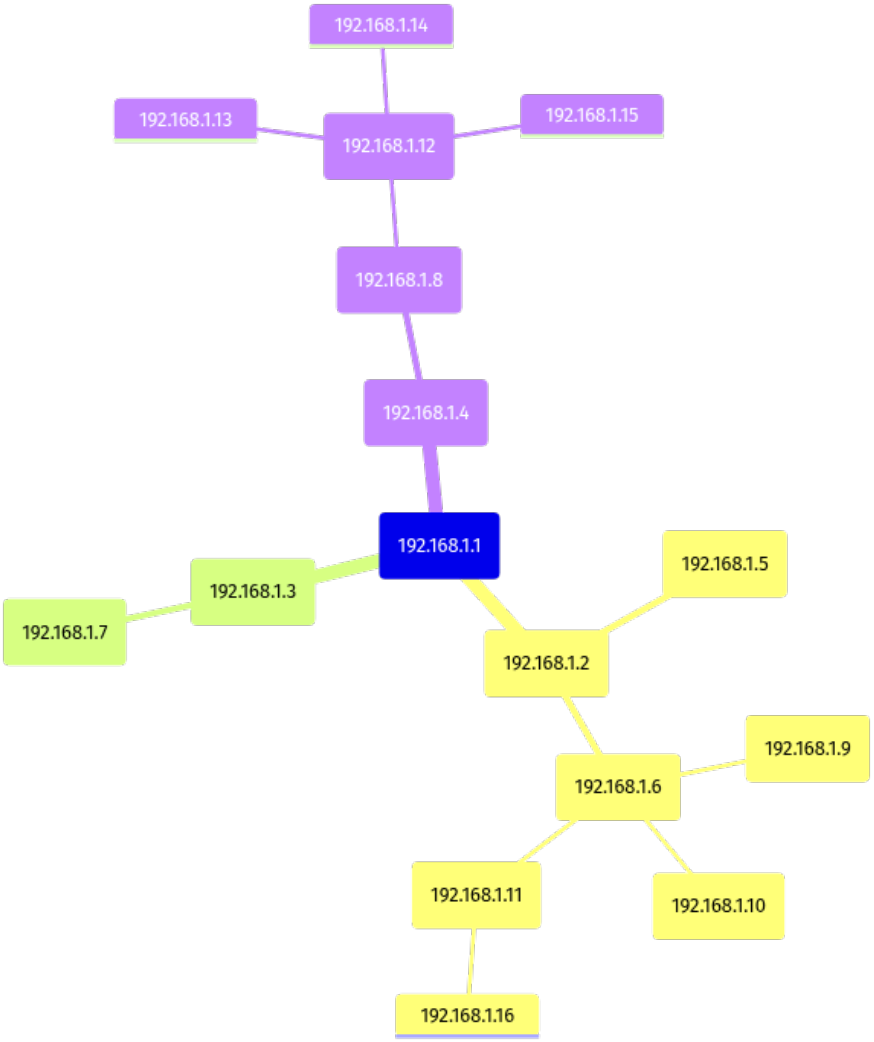


This procedure will then repeat until no new devices are found, with the operation hypothetically looking something like this.

Step 4


192.168.1.12
192.168.1.8
192.168.1.4
192.168.1.1
192.168.1.3
192.168.1.7
192.168.1.5
192.168.1.2
192.168.1.6
192.168.1.9
192.168.1.11
192.168.1.10

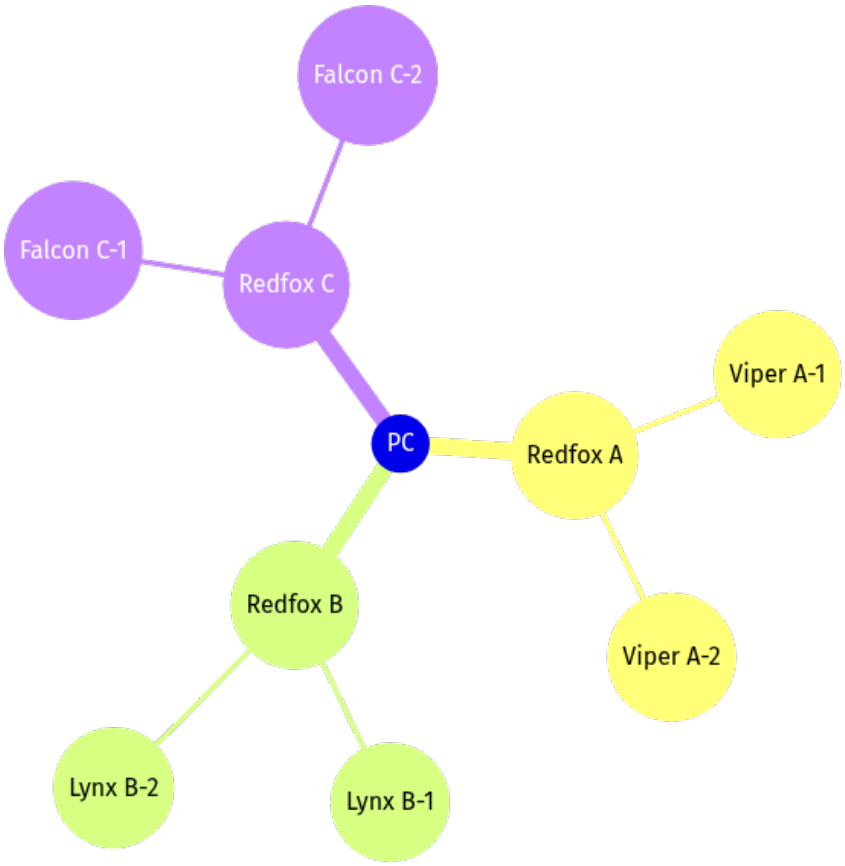
Step 5



And so on.

5.2. Network Order

Network order is a general term for an order-of-command-execution schema which aims to safely parallelize commands across a network in order to optimize execution time without risking execution failure. It works by traversing the network graph from the outside in, given a known position of the operating PC in the network. Consider the following network:



An operation executing in network order would be able to execute all command in two parallel batches, the first batch being the set:

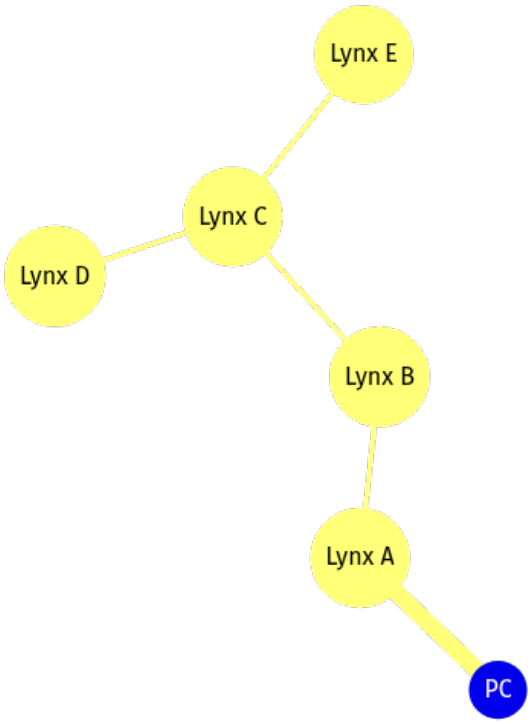
- Viper A-1
- Viper A-2
- Lynx B-1
- Lynx B-2
- Falcon C-1
- Falcon C-2

As they are all link-wise equidistant from the PC, the second batch being the set:

- Redfox A
- Redfox B
- Redfox C

Which then completes the network. This allows a safe parallelization across 9 devices in the time it'd take for (in an ideal world where all devices execute at the same time) 2 devices to execute the command.

Of course, in less parallel networks, such as the one below:



The benefit would be significantly reduced, since only Lynx E and Lynx D could be executed in parallel. But it is regardless never a performance loss over just executing the commands one-by-one in outside-in order.