www.westermo.com

WeConfig 2.0

User Guide

Table of Contents

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

[1] http://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 utilizied 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 reccomended start, it is suggested you visit the Topology and Devices panels, as it is the main views of WeConfig and will give you a good overview of the network you are working with.

Secondly, consult the Interface based 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 Interface based discovery task and select the NIC that the device is connected to and press "Apply", 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	Windows 11 version 24H2 or later

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.

Installation Packages

Standard Installation Package

- Executable Name: WeConfigSetup.exe[2]

[1] https://npcap.com

- 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 Installation Package

- Executable Name: WeConfig.Msi.(Version).msi[1]
- 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

Portable Installations

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

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.

Specifically, this is used by WeConfig to edit addresses and routes on selected network adapters, via the software known as `WeConfig.LinkLocal.Daemon.exe`, a small executable that uses the windows-native `IpHlpApi` to add linklocal addressing to a specific network adapter, and via `WeConfig.InterfaceEditor.Daemon.exe` in order to edit routes and other IP properties.

Requirement 2: Creating firewall rules

In order to recieve 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.

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

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

Alternatively, if the required changes are deployed to the managing device ahead of time, this functionality becomes unnessecary. 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 occassional 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 toplogy map	Keeps any previously retrieved entries
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[1]	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.

[1] #stats

2.2. Device Requirements

WeConfig is designed for Westermo devices with WeOS version 4.28 or later, and devices in the Ibex and MRD 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 compatability 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 Windows built-in SSH client command in a terminal window. If it has not been installed as part of Windows, please do so in Windows’s own Optional Features.

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	198.168.1.2	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	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[1]
- A data directory containing
  - Cached user notifications and their state.
  - Project-wide attachments[2]
  - A device-attachments directory containing
    - A subdirectory per device that has attachments[3] containing
      - Any device specific attachments[4]
- A configuration file directory containing
  - A subdirectory per device that has backups[5] 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 2.0 writes version 3.1, but 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[6] 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

[1] #xmlFile

[2] #attachments

[3] #attachments

[4] #attachments

[5] #backups

[6] #xmlFile-device

- 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

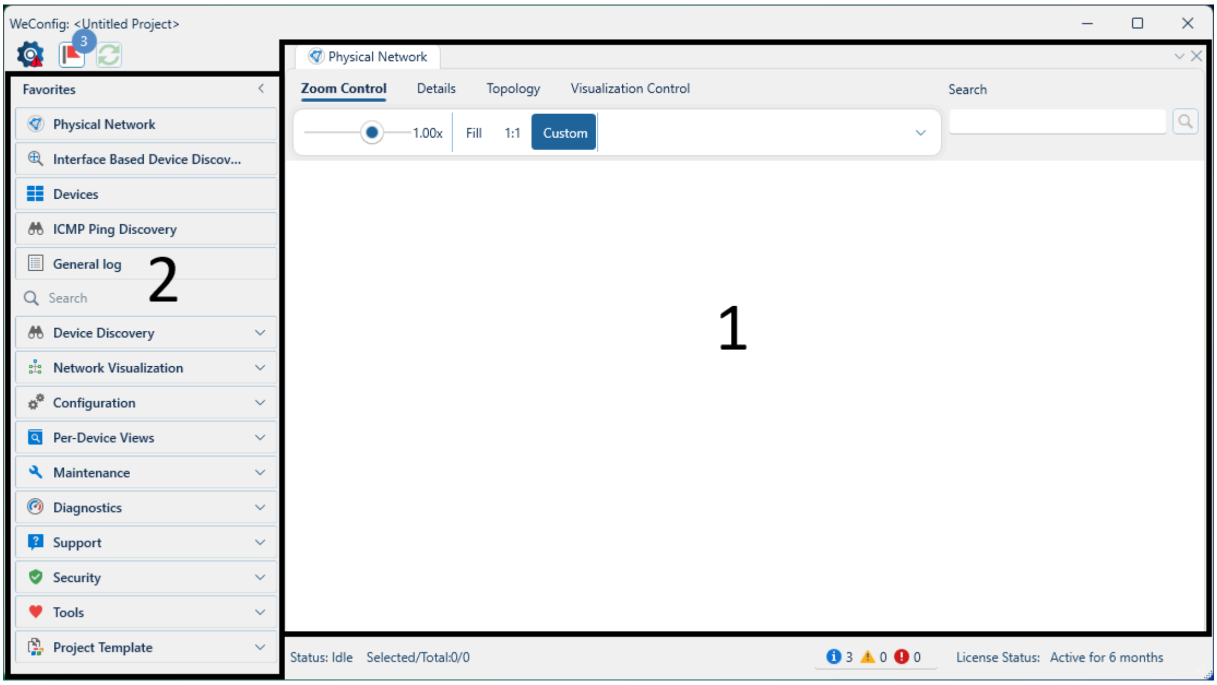
If the task requires any form of device selection, that selection will mirror the current selection in the Network Topology and/or Device List

Staged Tasks

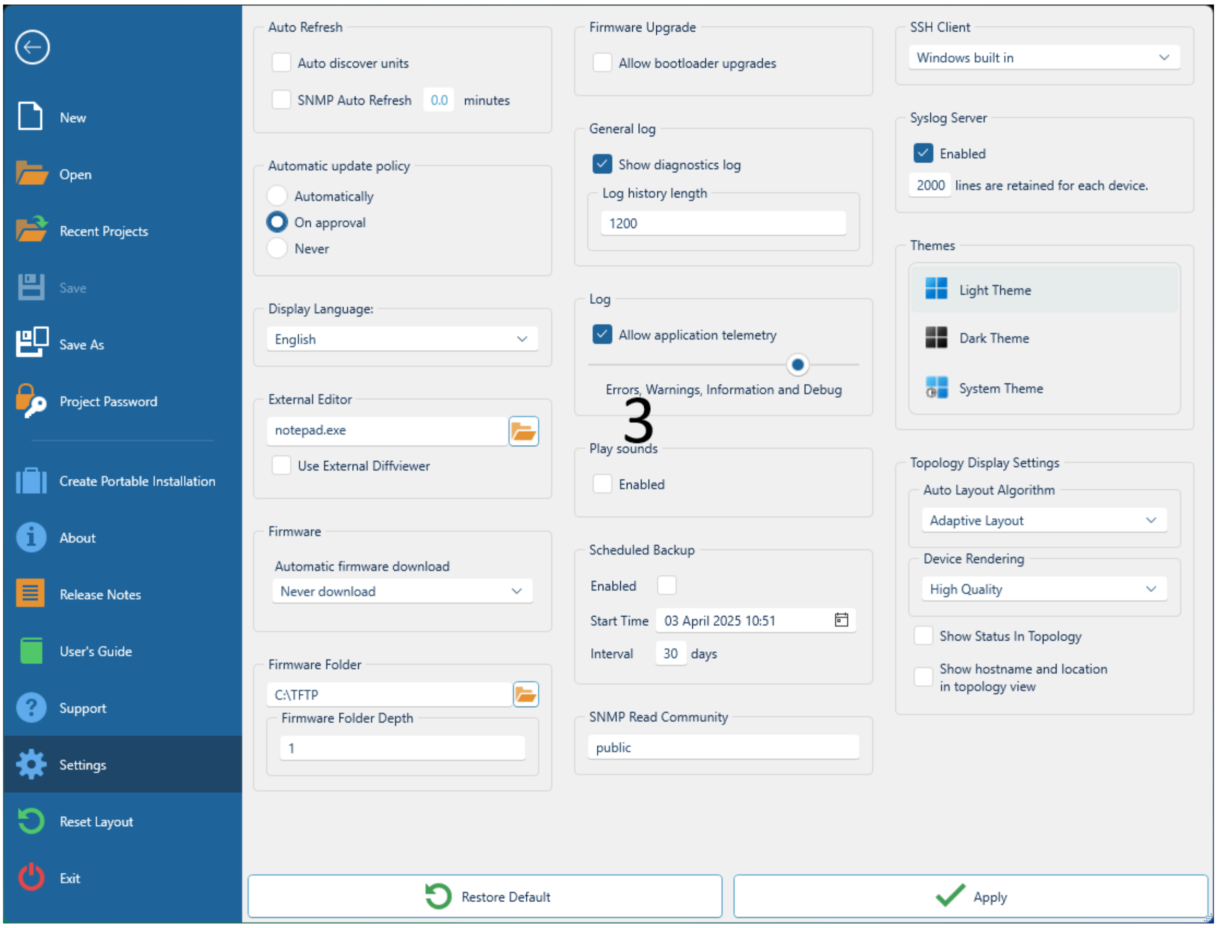
Some tasks may have several stages, as indicated by a step-by-step progress bar at the top of the user interface indicating the stages present. You can always return to a previous stage in a staged task by clicking on the associated stage step in the progress bar.

4. Interface

WeConfigs user interface is divided into three primary areas:



Status: Idle Selected/Total:0/0 ⓘ 3 ⚠ 0 ❗ 0 License Status: Active for 6 months



Auto Refresh
□ Auto discover units
□ SNMP Auto Refresh 0.0 minutes

Automatic update policy
○ Automatically
● On approval
○ Never

Display Language: English

External Editor: notepad.exe
□ Use External Diffviewer

Firmware
Automatic firmware download: Never download

Firmware Folder: C:\TFTP
Firmware Folder Depth: 1

Firmware Upgrade
□ Allow bootloader upgrades

General log
☑ Show diagnostics log
Log history length: 1200

Log
☑ Allow application telemetry
Errors, Warnings, Information and Debug

Play sounds
□ Enabled

Scheduled Backup
Enabled □
Start Time 03 April 2025 10:51
Interval 30 days

SNMP Read Community: public

SSH Client: Windows built in

Syslog Server
☑ Enabled
2000 lines are retained for each device.

Themes
Light Theme
Dark Theme
System Theme

Topology Display Settings
Auto Layout Algorithm: Adaptive Layout
Device Rendering: High Quality
□ Show Status In Topology
□ Show hostname and location in topology view

1. The Document panel
2. The Navigation menu

3. The Backstage menu, accessed by clicking the program icon at the top left

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. 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.

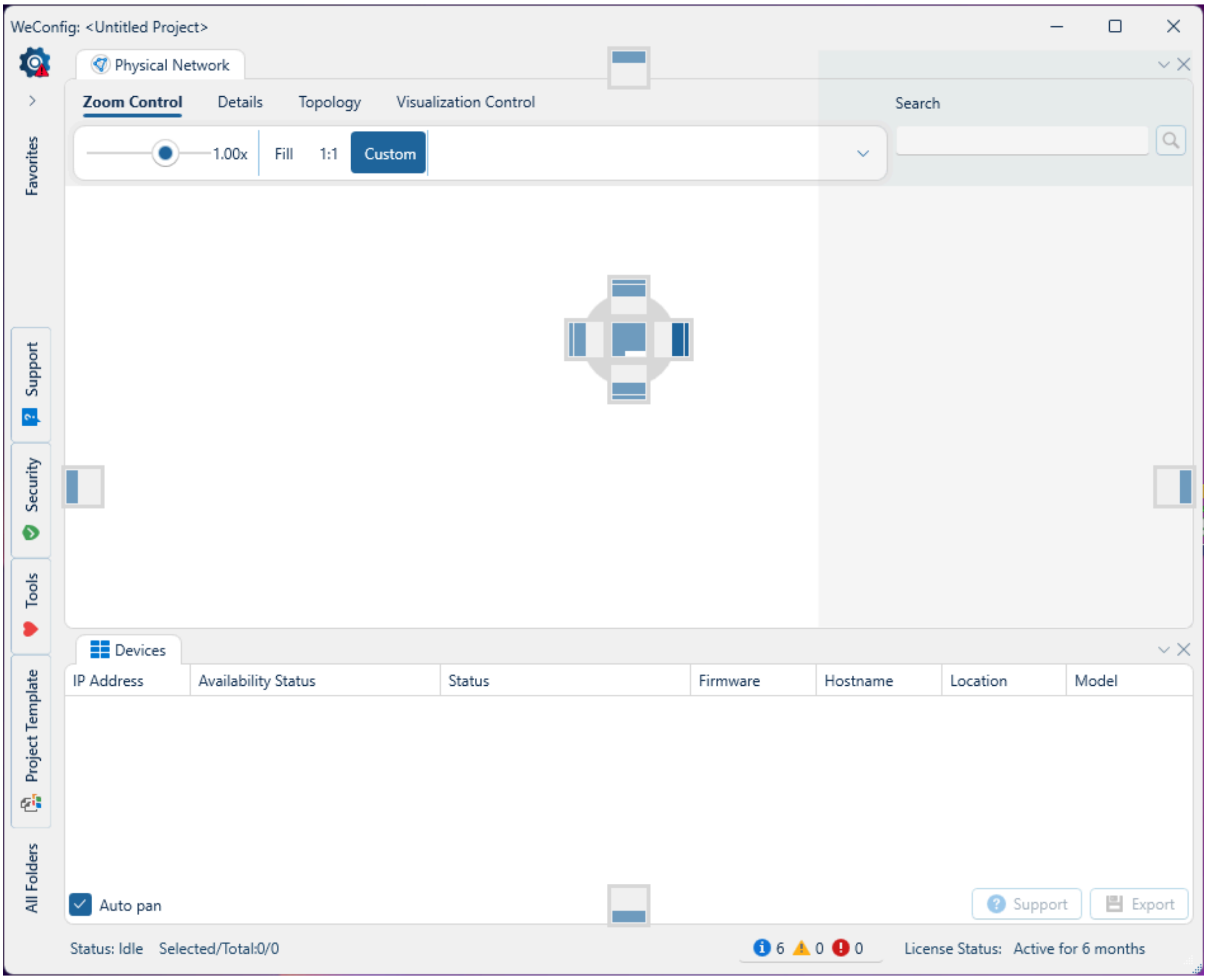
The layout is saved in-between program runs on a per-screen-setup basis. As such, WeConfig will remember where you have placed its different panels after it is closed and reopened.

> 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 between the 'Physical Network' panel and 'Attachments' 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.

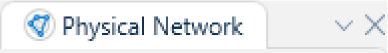
Floating panels

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.

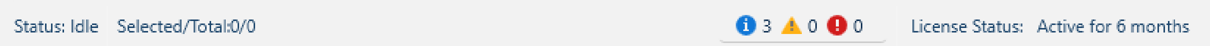
Closing panels

Panels can be closed either via the panel headers context menu, selecting the "Close" option, or via the 'X' button located to the top-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:


Status: Idle Selected/Total:0/0 ℹ 3 ⚠ 0 ❗ 0 License Status: Active for 6 months

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

Status indicator

Leftmost you will find 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".

Selection indicator

Next is the selection indicator, which displays both the current number of devices in the network, as well as how many of them are considered selected at the current moment.

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 3 issues warranting inspection, and all three are ranked as "informatory".

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

Portable mode indicator

Next, conditionally, is the portable mode indicator, this is only visible if, and an indicator that, WeConfig is running in portable mode.

License indicator

Finally, to the rightmost part of the status bar, is the license indicator, which displays the current licensing status of the software:

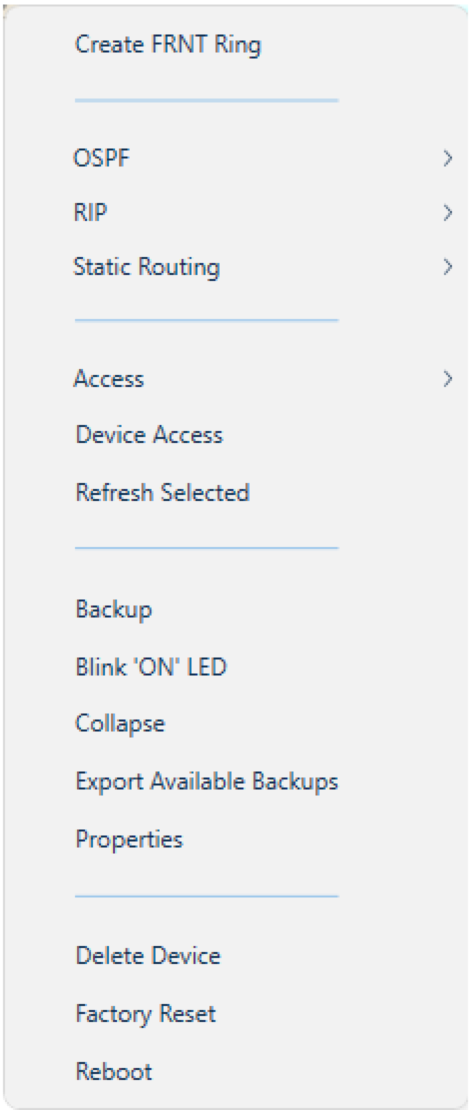
> 💡 Tip
>
> Interested in acquiring a licensed version of WeConfig? Please turn to your Westermo point of contact, or send us a request at https://www.westermo.com/contact/send-message[1].

> *i* Info
>
> This indicator also has a context menu. This context menu will allow you to do two things:
> - 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.
> - Open a dialog for activating license keys provided to you.

[1] https://www.westermo.com/contact/send-message

4.2. 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.

> *i* 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

> *i* 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

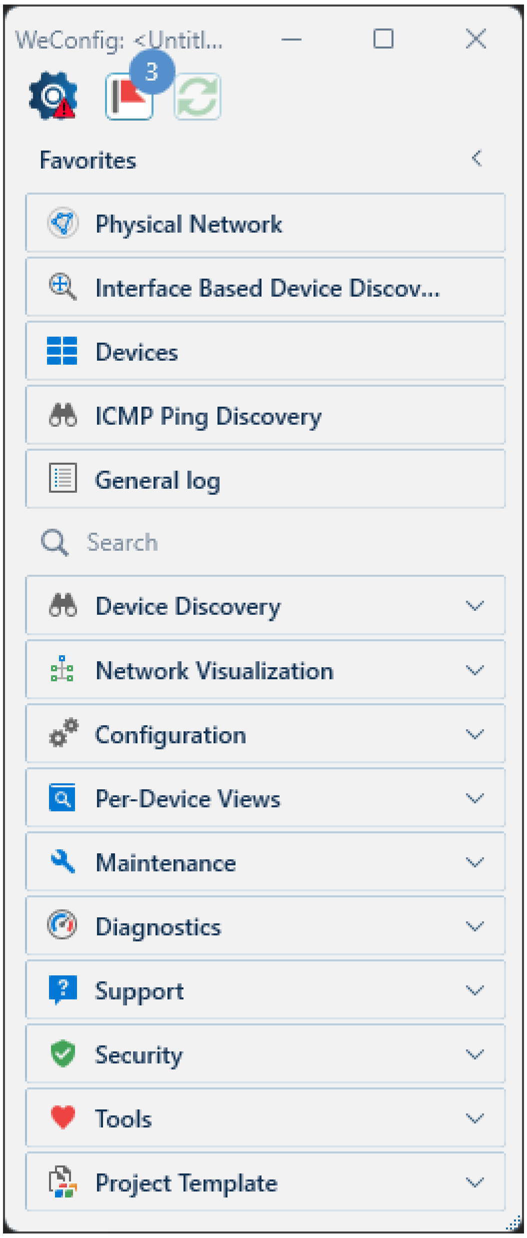
This context menu option triggers a refresh operation on the selected devices.

Note

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

4.3. Navigation

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



There are several components to this menu, which starting from the top to the bottom are:

Top buttons

At the very top to the navigation menu you will find three buttons, these are, from left to right:

- The backstage menu.
- The notifications list, decorated with the current number of notifications.
- The "Refresh all" button.

Favorites

Below the top level buttons you will find the favorites list, which lists the panels that the user wishes to have easiest access to. Initially, this menu is populated with the `Physical Network`, `Interface Based Device Discovery` and `Devices` panels, which correspond to the initially open panels on the first start of WeConfig.

Left-clicking an item in the favorites menu opens the corresponding panel in the document panel

In order to remove an item from the favorite list, right click it and select “Favorite” from the the provided context menu.

To the right of the header of the favorites list, 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.

Navigation tree

Below the favorites section is the rest of the navigation tree, which contains all panels that you can navigate to. This tree begins with a search box, which allows you to filter the tree in search of a particular panel.

These views are arranged into categories, indicated by the presence of an expander button to the right side of the category name. Left-clicking a category expands or collapses it, revealing or hiding the panels within that category.

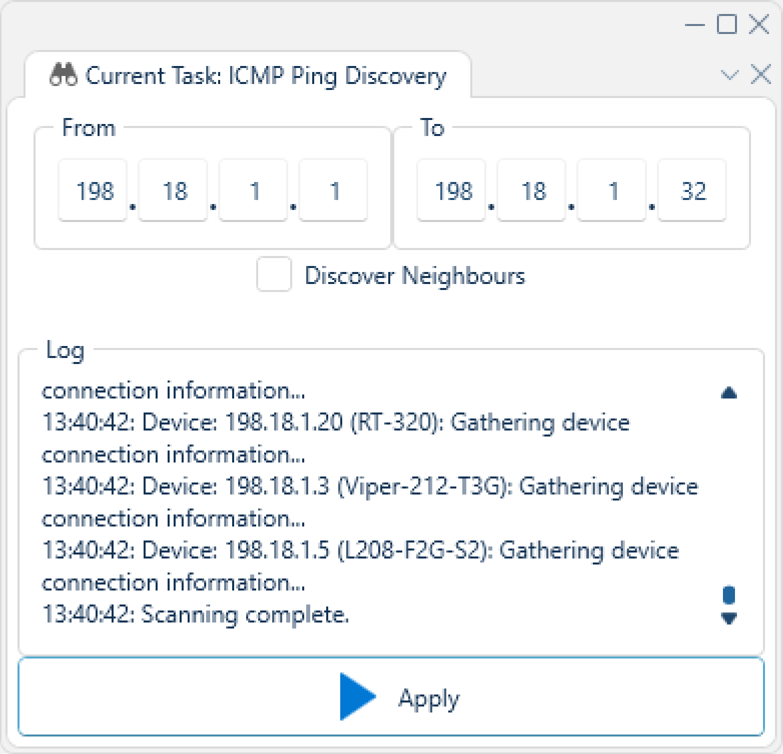
Left-clicking any non-category item within the tree opens the corresponding panel in the document panel.

In order to add an item from the navigation tree to the favorites list, right click it and select “Favorite” from the provided context menu.

4.3.1. Device Discovery

4.3.1.1. ICMP Ping Discovery

The ICMP Ping discovery task, or just Ping discovery for short, is a discovery method that scans a specified set of IPv4 addresses as a baseline to locate and identify devices. The image below display the typical user interface for this task.



```
connection information...
13:40:42: Device: 198.18.1.20 (RT-320): Gathering device
connection information...
13:40:42: Device: 198.18.1.3 (Viper-212-T3G): Gathering device
connection information...
13:40:42: Device: 198.18.1.5 (L208-F2G-S2): Gathering device
connection information...
13:40:42: Scanning complete.
```

Interface elements

From & To

These two IP-address boxes specify the range of IP addresses to scan as a linear range. I.e entering the values `192.168.0.1` and `192.168.0.5` respectively will attempt to find devices on the following addresses:

- `192.168.0.1`
- `192.168.0.2`
- `192.168.0.3`
- `192.168.0.4`
- `192.168.0.5`

Discover Neighbours

Checking this box will cause WeConfig to run recursive discovery on all devices that are successfully scanned in the ping range specified by From & To.

Warnings:

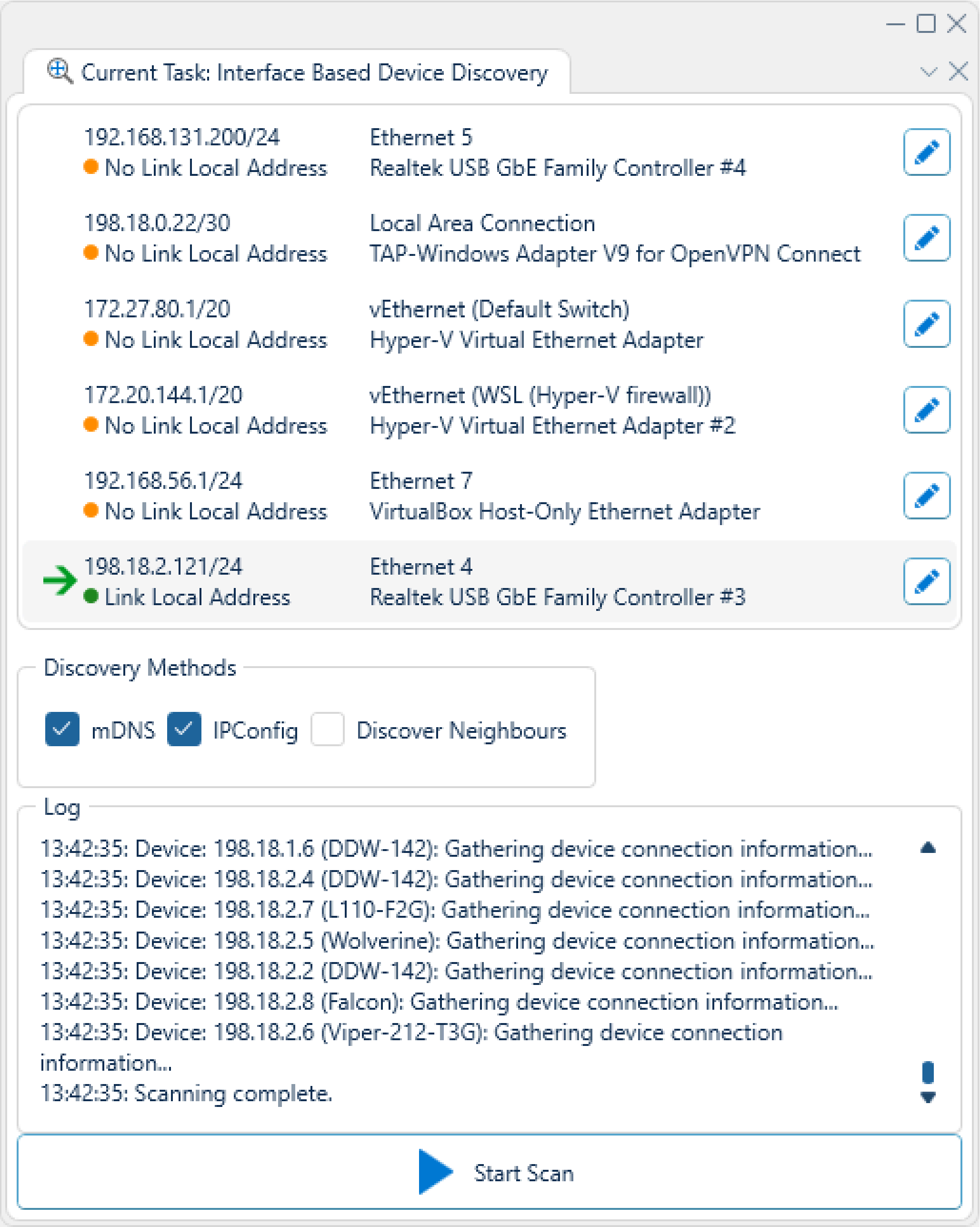
As seen in the interface figure, a warning may appear when attempting to scan a large number (>255) of addresses, the primary intent of this warning is to alert the user to potentially unintentional entries in the From & To box that would result in sending an excessive amount of pings.

Log

This box contains a log window of the discovery process. The log will indicate which addresses are being pinged, which have responded, and data about the devices as they are being discoverd, as well as the current state of the discovery process.

4.3.1.2. Interface based device discovery

The interface based device discovery task is a discovery method that attempts to discover devices connected to a specific PC-network interface. The image below display the typical user interface for this task.



192.168.131.200/24 ● No Link Local Address — Ethernet 5, Realtek USB GbE Family Controller #4

198.18.0.22/30 ● No Link Local Address — Local Area Connection, TAP-Windows Adapter V9 for OpenVPN Connect

172.27.80.1/20 ● No Link Local Address — vEthernet (Default Switch), Hyper-V Virtual Ethernet Adapter

172.20.144.1/20 ● No Link Local Address — vEthernet (WSL (Hyper-V firewall)), Hyper-V Virtual Ethernet Adapter #2

192.168.56.1/24 ● No Link Local Address — Ethernet 7, VirtualBox Host-Only Ethernet Adapter

→ 198.18.2.121/24 ● Link Local Address — Ethernet 4, Realtek USB GbE Family Controller #3

Discovery Methods

☑ mDNS ☑ IPConfig □ Discover Neighbours

Log

```
13:42:35: Device: 198.18.1.6 (DDW-142): Gathering device connection information...
13:42:35: Device: 198.18.2.4 (DDW-142): Gathering device connection information...
13:42:35: Device: 198.18.2.7 (L110-F2G): Gathering device connection information...
13:42:35: Device: 198.18.2.5 (Wolverine): Gathering device connection information...
13:42:35: Device: 198.18.2.2 (DDW-142): Gathering device connection information...
13:42:35: Device: 198.18.2.8 (Falcon): Gathering device connection information...
13:42:35: Device: 198.18.2.6 (Viper-212-T3G): Gathering device connection
information...
13:42:35: Scanning complete.
```

Interface Elements

Network Interface selector

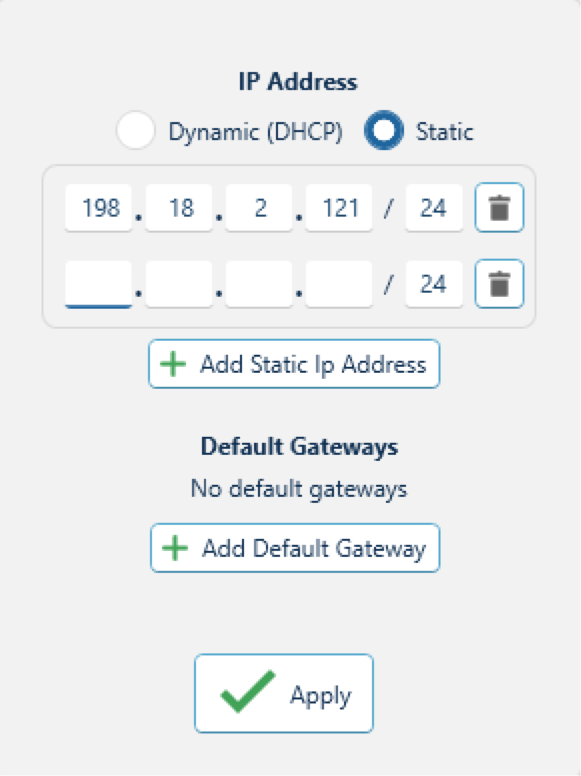
This element lists the available network interfaces on the PC, with each entry containing the following data:

- 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:



IP Address

○ Dynamic (DHCP) ● Static

198 . 18 . 2 . 121 / 24

. . . / 24

Default Gateways

No default gateways

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
>
> Using the editor functionality requires at least `NET ADMIN` or `Administrator` priviligies. WeConfig will request these priviligies as needed.

Discovery Methods

This element allows control over which methods of interface discovery that will be used in the discovery process. The options are as follows:

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

Option	Description	Limitations
	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.
Discover Neighbours	Checks the PC's ARP-table and runs recursive discovery based on that information, along with any information gathered from the other protocols.	Requires either a populated ARP table on the local PC, or at least one of the other two methods to be used.

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

Warning	Explanation
No Link local address	The selected interface does not have a link local address, which may hamper it's ability to communicate with identified devices
IPConfig is blocked	The PC's firewall does not allow IPConfig packets to be sent or recieved, which will prevent device discovery using this protocol
mDNS is blocked	The PC's fireall does not allow mDNS packets to be sent or recieved, which will prevent device discovery from using this protocol

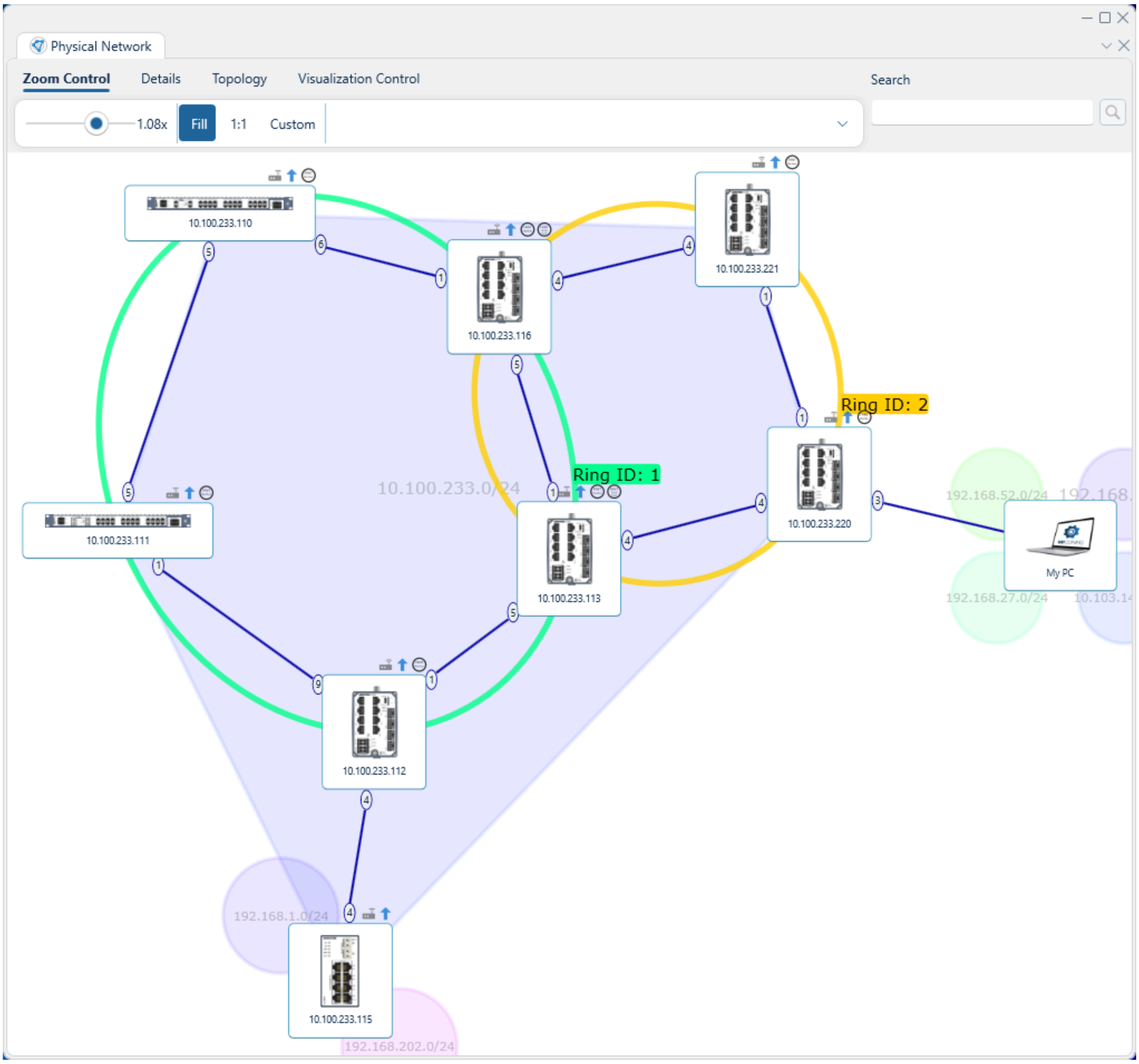
> *i* Info
>
> All of the remediation links require escalated priviligies to resolve, WeConfig will request these as nesssecary. `NET ADMIN` is sufficient to add link local addresses, but `Administrator` is required for any firewall editing.

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 discoverd, as well as the current state of the discovery process.

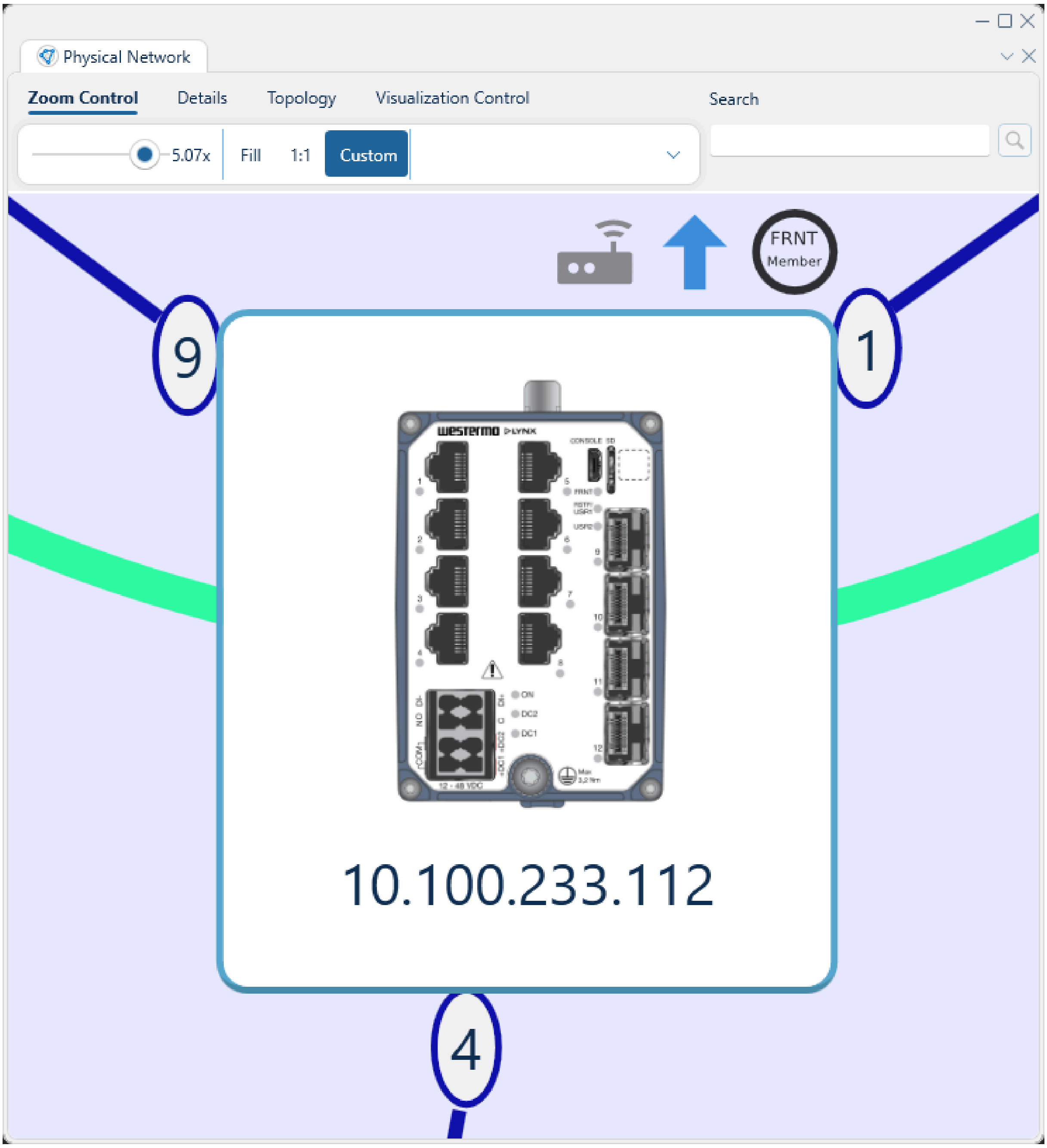
4.3.2. Network Visualization

4.3.2.1. Physical Network

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.



Wherein we can see the 8 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 roughly in the middle of the picture and `Ring ID: 2` depicted to the lower right. 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 and host name listed beneath.

> Tip
>
> The contents displayed within the device border can be configured in project settings.

Node Icons

Above the device border, we see a list of icons, 5 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 Member"	Device is part of at least 1 FRNT ring.	No
Circle with "FRNT Focal Point"	Device is the focal point for a FRNT ring	No
Circle with "MRP Client"	Device is an MRP client	No
Circle with "MP 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[1] ribbon, 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 three connections, one through port 1, 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

i Info

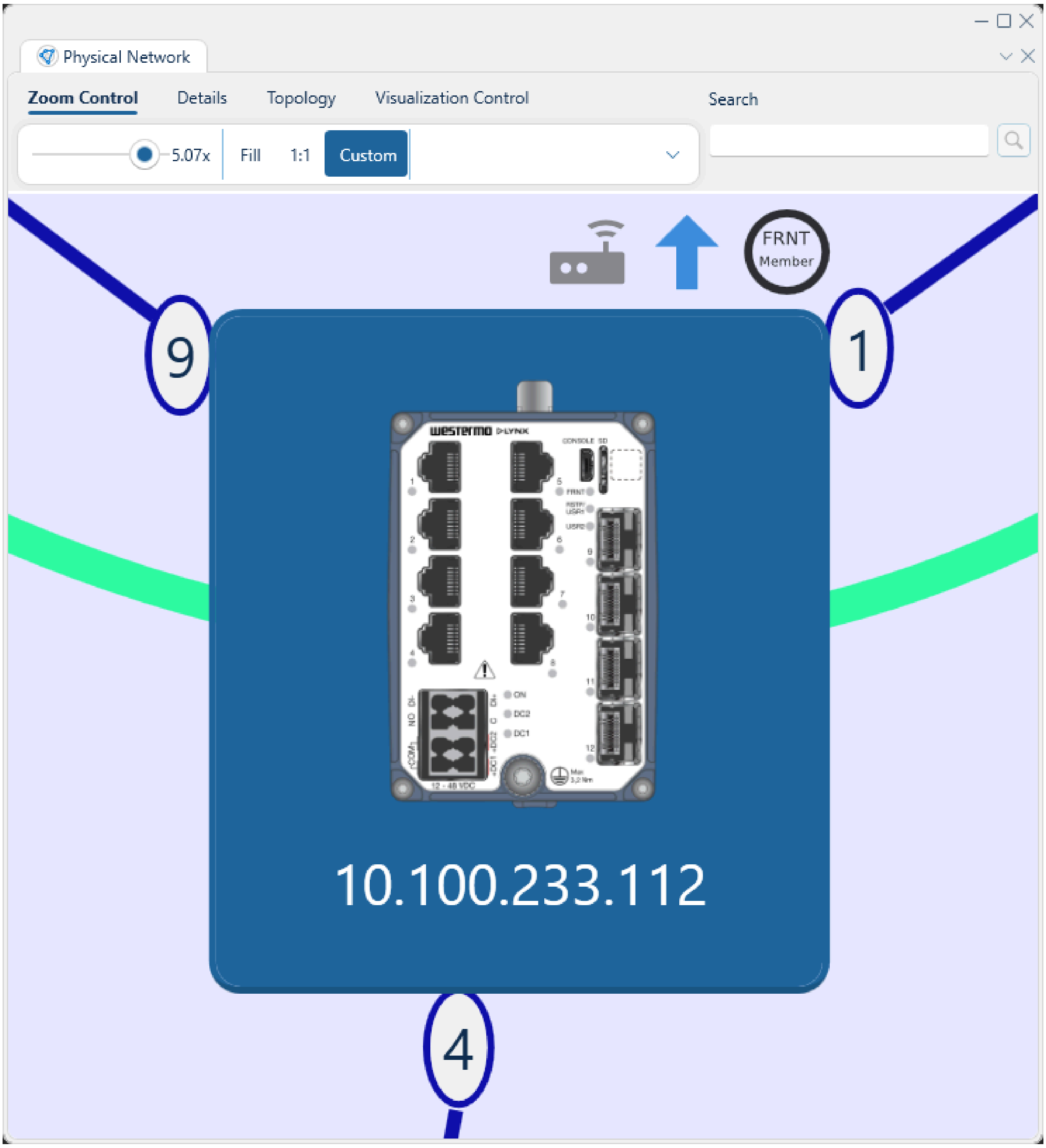
This legend is also available in the topology view itself if "Show connection information" is toggled under the details[2] ribbon

[1] #topology-details

[2] #topology-details

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 `10.100.233.133` 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

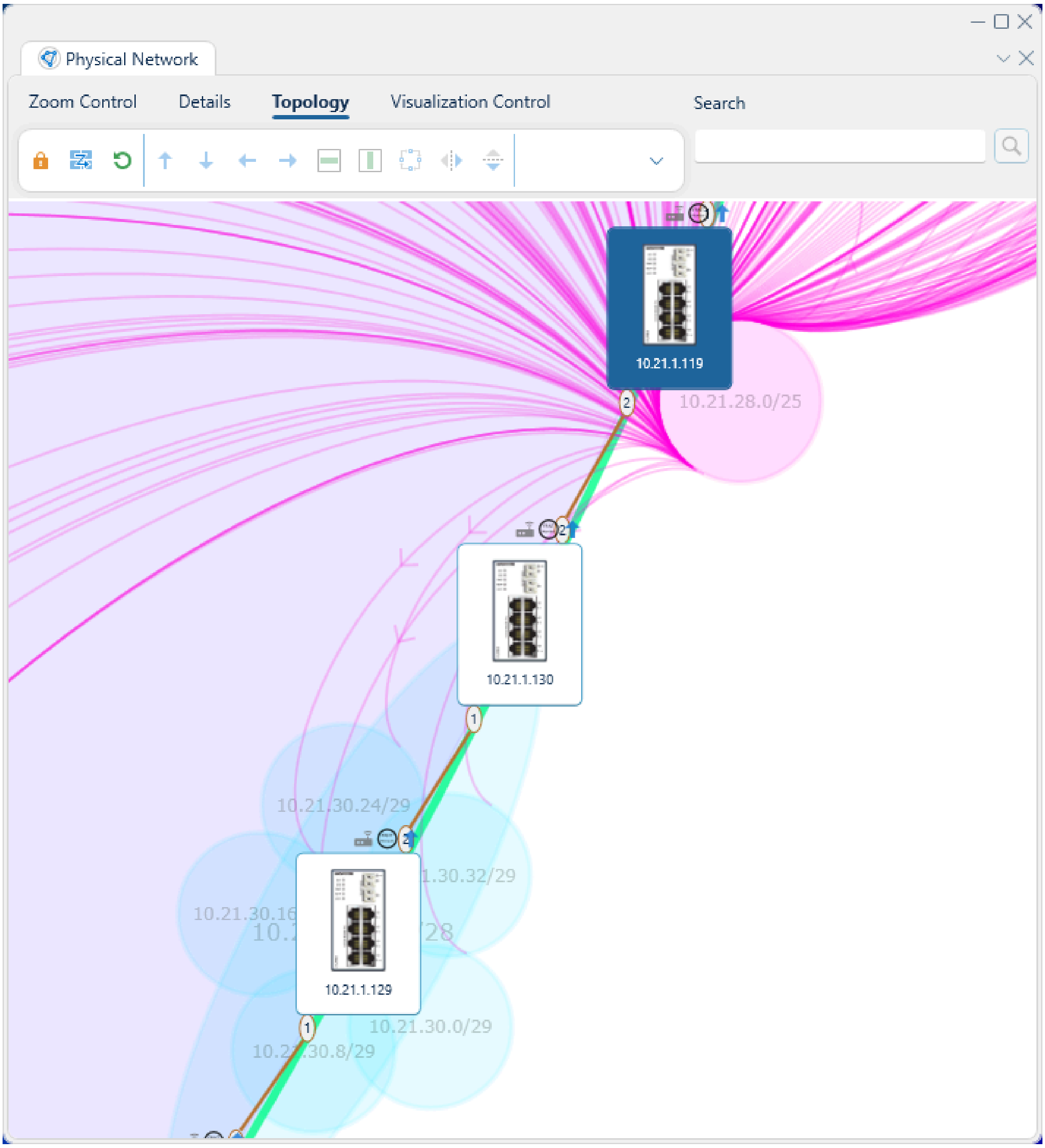
It is possible to select based on both subnets and rings present in the topology as well, to do so, simply click the ring or subnet you wish to select. Some specifications apply for subnet and ring selection:

- 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 subnet is selected, WeConfig will attempt to visualize it's outgoing routes to any other known subnets inside of the topology, see the image below where the subnet `10.21.59.0/26` is selected:

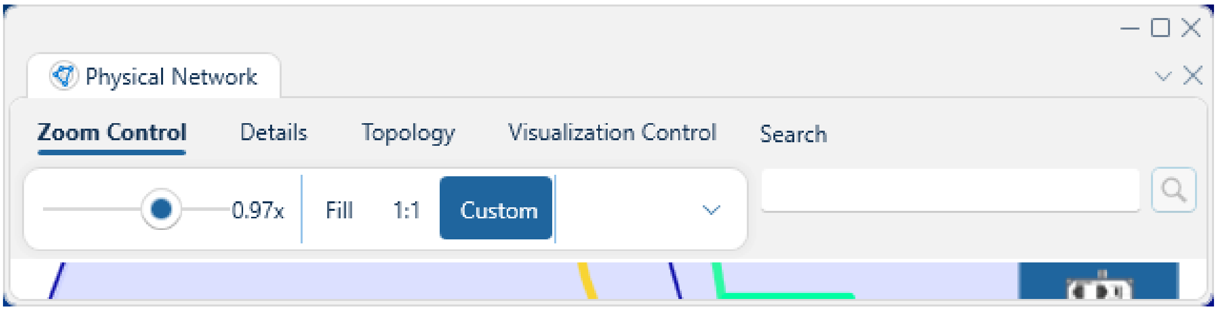


Where we can observe a set of curves going between, for example, the subnet `10.21.59.0/26` and `10.21.58.0/26`, with the color indicating the origin subnet and pointing to the target subnet, indicating that route exists from source to target.

Topology control ribbon

At the very top of the physical network panel we will find a ribbon 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 four section, Zoom control, Details, Topology and Visualization control.

Zoom Control



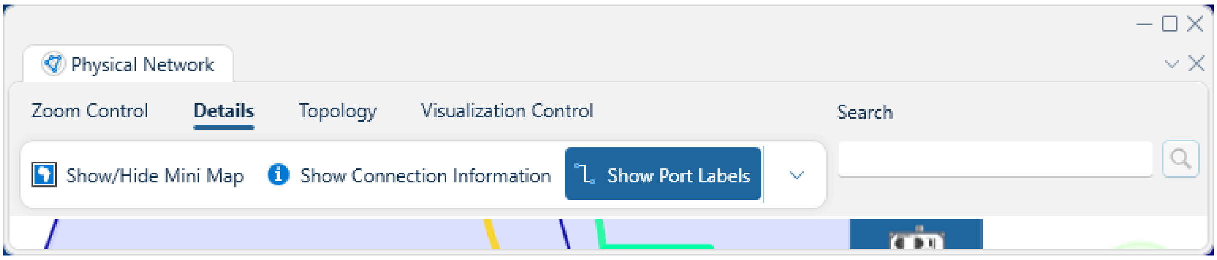
The first ribbon menu controls the zoom level within the physical network topology view. The ribbon contains a slider indicating the current zoom level, as well as a set of three buttons which indicate and alter the current zoom state, they are as follows:

Button	Purpose
Fill	Zoom the network topology to the minimum encompassing bounding box
1:1	Reset the zoom level to 1
Custom	Does nothing, but indicates that the current zoom state differs from the other options

Tip

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

Details



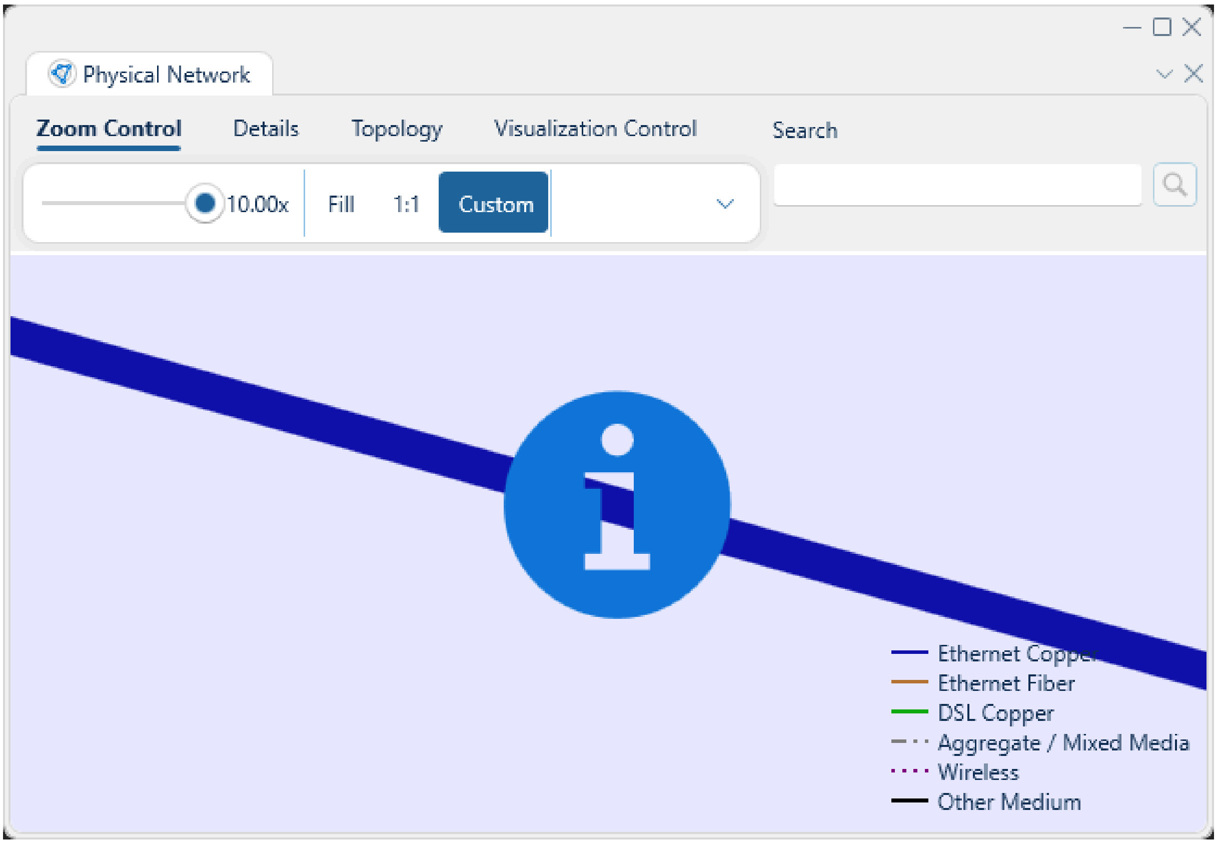
The second ribbon menu controls whether or not certain additional information is rendered, it consists of three toggles, which serve the following purpose:

Toggle	Purpose
Show/Hide Minimap	Controls the display of a network ‘Minimap’ at the bottom right corner of the panel

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[1] ' buttons that adorn the connections in the topology'
Show Port Labels	Controls the display of port labels adjacent to devices

Connection information

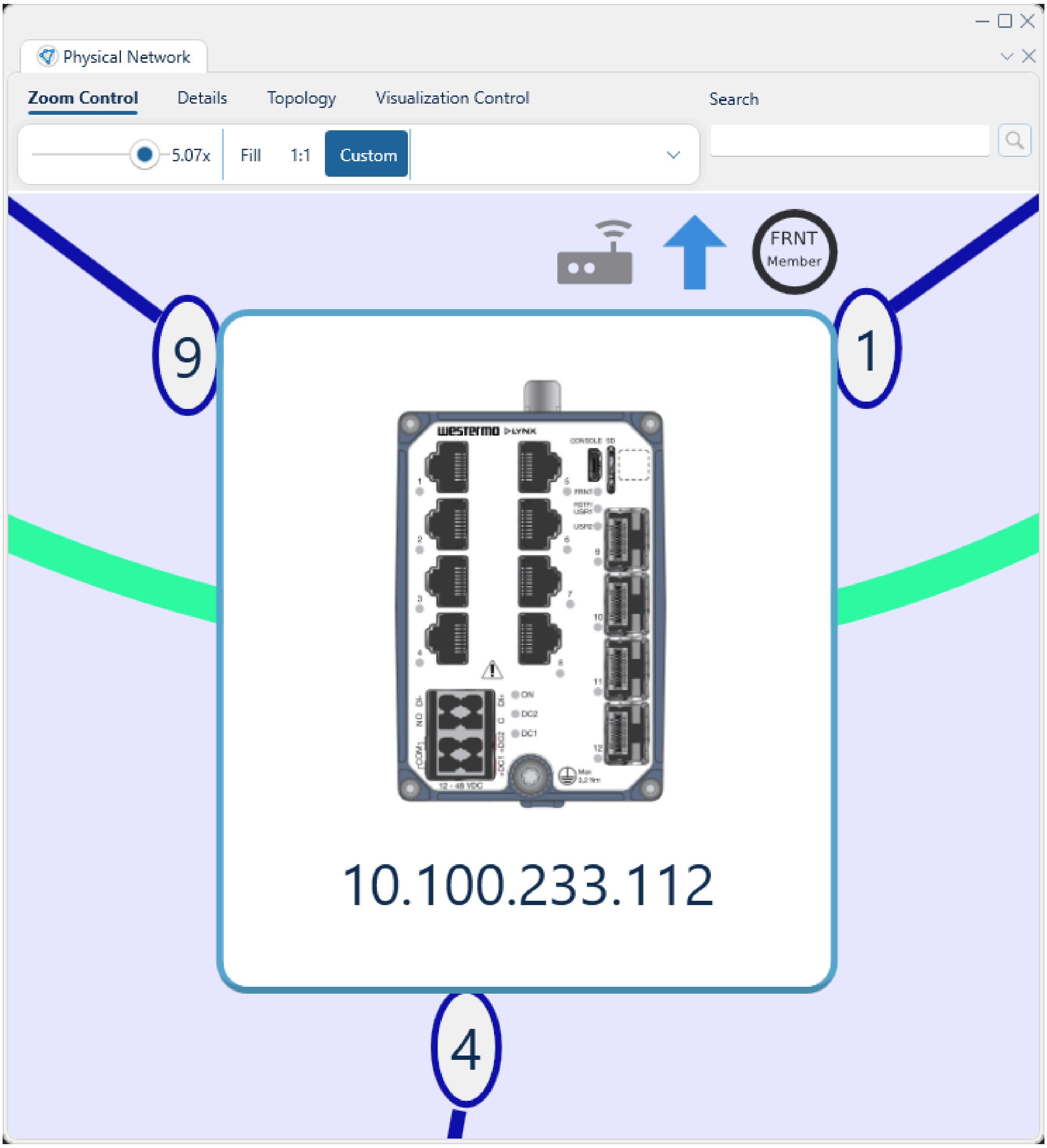
When toggled in the details[2] ribbon, a button will appear on the centre of each connection in the topology, as depicted below:



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

[1] #topology-connectionInfo

[2] #topology-details

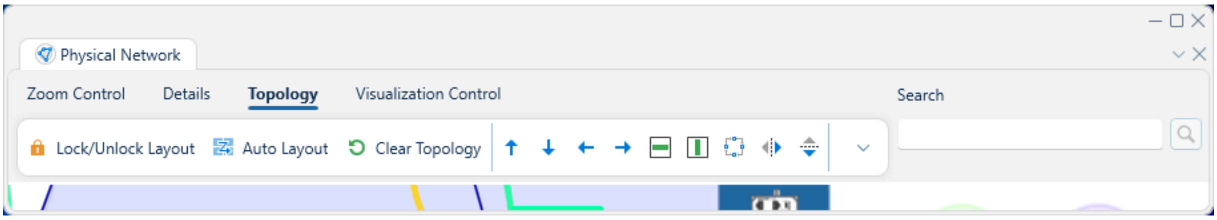


In this example, we can see that the device at `10.100.233.112` is connected to the device at `10.100.233.111` from port `9` to port `1`. The device images also indicate the physical location of these respective ports on the device itself with a green dot.

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 third ribbon menu controls the topology layout, it has several options, as depicted below:

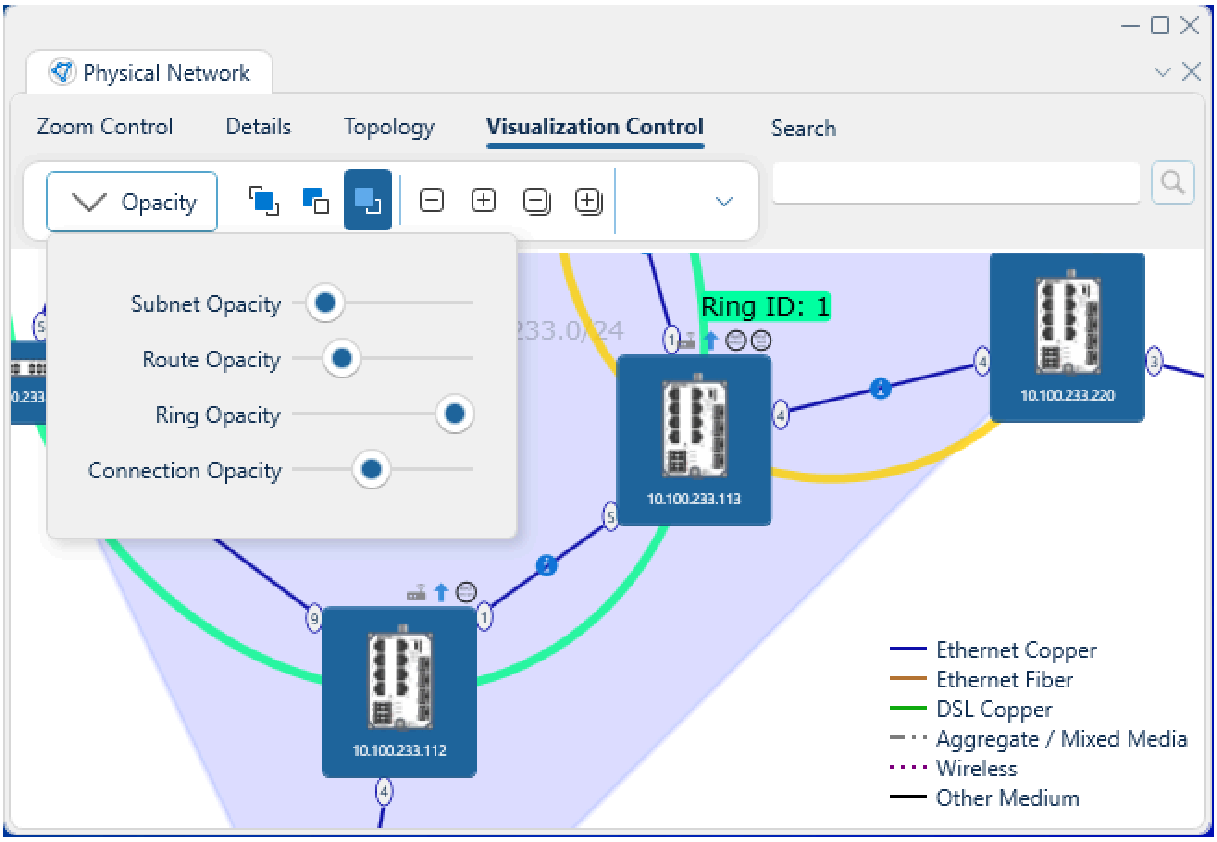


Where the options are, from left to right:

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

Visualization Control

The fourth and final ribbon menu contains additional control over topology visualization, as depicted below:

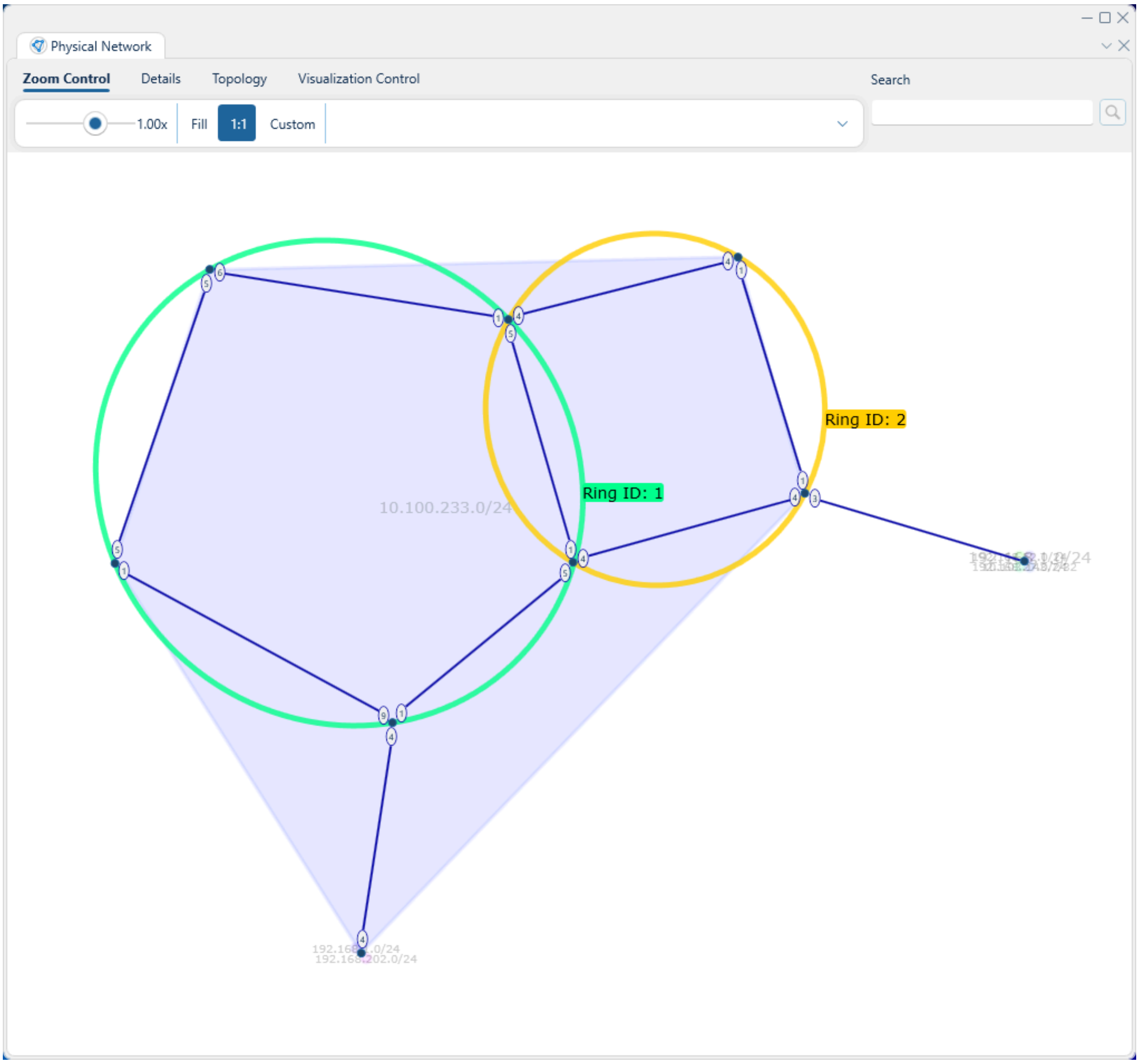


Where the dropdown-menu is toggleable via the “Opacity” button. The dropdown menu allows for control over the opacity of varius topology elements, including subnets, routes, rings and connections. Beyond that, the second set of toggles controls how subnets are colored, with the options being from left to right:

- By subnet address and OSPF area
- By OSPF area
- By subnet address

By default, subnets are colored by their address only.

Finally, the last four buttons allow 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:



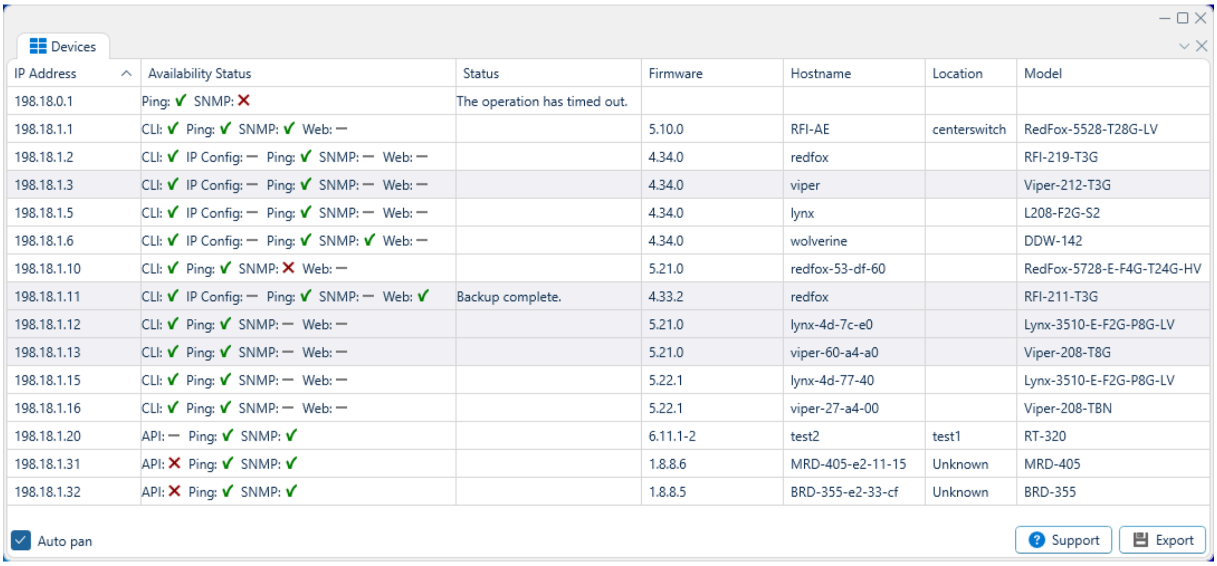
Tip

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.

4.3.2.2. Devices

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 multiple WeOS devices, an Ibex, a few xRDs and a singular unknown device.



IP Address	Availability Status	Status	Firmware	Hostname	Location	Model
198.18.0.1	Ping: ✓ SNMP: ✗	The operation has timed out.				
198.18.1.1	CLI: ✓ Ping: ✓ SNMP: ✓ Web: –		5.10.0	RFI-AE	centerswitch	RedFox-5528-T28G-LV
198.18.1.2	CLI: ✓ IP Config: – Ping: ✓ SNMP: – Web: –		4.34.0	redfox		RFI-219-T3G
198.18.1.3	CLI: ✓ IP Config: – Ping: ✓ SNMP: – Web: –		4.34.0	viper		Viper-212-T3G
198.18.1.5	CLI: ✓ IP Config: – Ping: ✓ SNMP: – Web: –		4.34.0	lynx		L208-F2G-S2
198.18.1.6	CLI: ✓ IP Config: – Ping: ✓ SNMP: ✓ Web: –		4.34.0	wolverine		DDW-142
198.18.1.10	CLI: ✓ Ping: ✓ SNMP: ✗ Web: –		5.21.0	redfox-53-df-60		RedFox-5728-E-F4G-T24G-HV
198.18.1.11	CLI: ✓ IP Config: – Ping: ✓ SNMP: – Web: ✓	Backup complete.	4.33.2	redfox		RFI-211-T3G
198.18.1.12	CLI: ✓ Ping: ✓ SNMP: – Web: –		5.21.0	lynx-4d-7c-e0		Lynx-3510-E-F2G-P8G-LV
198.18.1.13	CLI: ✓ Ping: ✓ SNMP: – Web: –		5.21.0	viper-60-a4-a0		Viper-208-T8G
198.18.1.15	CLI: ✓ Ping: ✓ SNMP: – Web: –		5.22.1	lynx-4d-77-40		Lynx-3510-E-F2G-P8G-LV
198.18.1.16	CLI: ✓ Ping: ✓ SNMP: – Web: –		5.22.1	viper-27-a4-00		Viper-208-TBN
198.18.1.20	API: – Ping: ✓ SNMP: ✓		6.11.1-2	test2	test1	RT-320
198.18.1.31	API: ✗ Ping: ✓ SNMP: ✓		1.8.8.6	MRD-405-e2-11-15	Unknown	MRD-405
198.18.1.32	API: ✗ Ping: ✓ SNMP: ✓		1.8.8.5	BRD-355-e2-33-cf	Unknown	BRD-355

☑ Auto pan

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.

Availability status

This column presents a collection of status indicators:

Indicator	Meaning
Check mark	Good
Cross	Bad /Failed
-	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.

Firmware
This column presents the current primary firmware version detected on 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 tthe identified specific model, if any, associated with the device.

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 the Export button. Clicking this button will download the tech support files from the selected devices.ce.

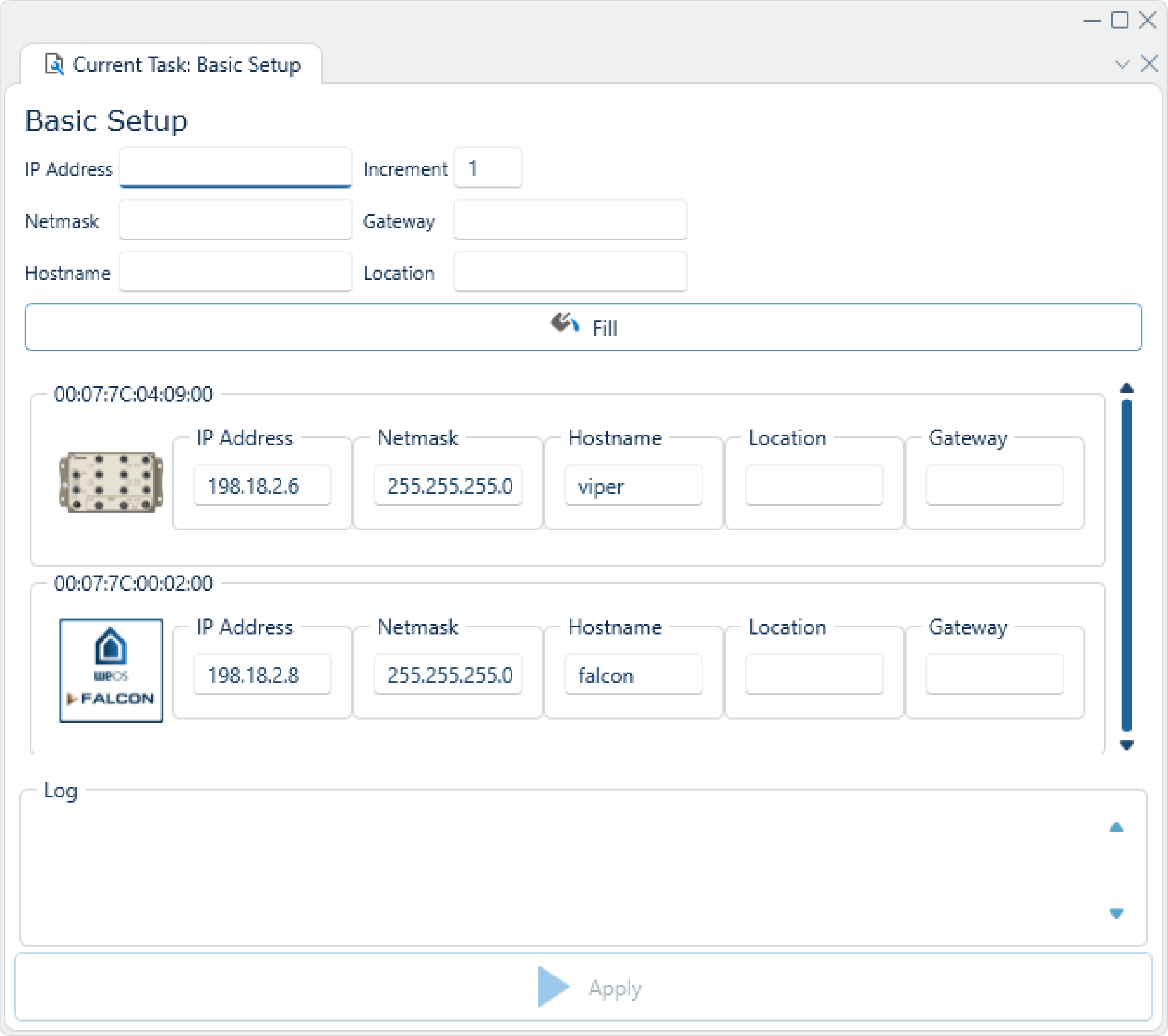
4.3.3. Configuration

4.3.3.1. Basic Setup

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 IP or CIDR notation	Yes
Netmask	The mask of the primary IP address, specifying it's subnet	Yes, unless IP Address is specified in CIDR notation
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



Increment: 1

00:07:7C:04:09:00

IP Address	Netmask	Hostname	Location	Gateway
198.18.2.6	255.255.255.0	viper		

00:07:7C:00:02:00

IP Address	Netmask	Hostname	Location	Gateway
198.18.2.8	255.255.255.0	falcon		

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 editted to set device-specific properties.

Log

This box contains a log window of the execution process. The log will the current state of the discovery process, as well as any errors

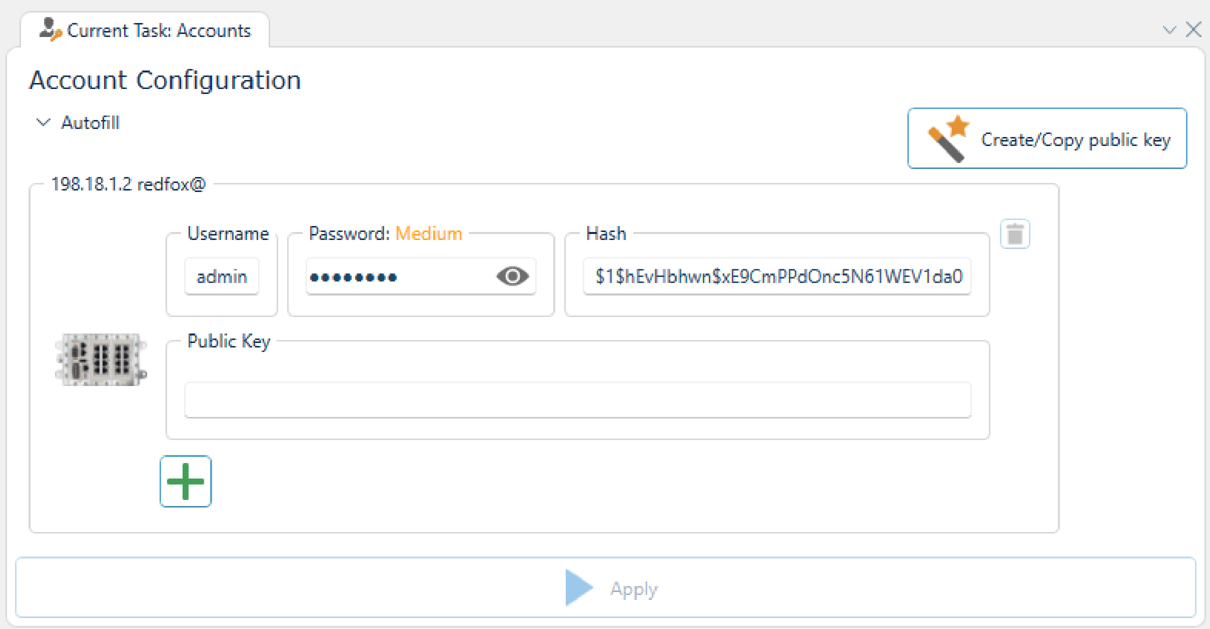
4.3.3.2. Accounts

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
Ibes	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



Username: admin

Password: Medium ••••••••

Hash: $1$hEvHbhwn$xE9CmPPdOnc5N61WEV1da0

Public Key:

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 to the top right 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, in this case a RFI-219-T3G with the hostname `redfox` at the IP address `198.18.1.2`. This is a WeOS 4 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.

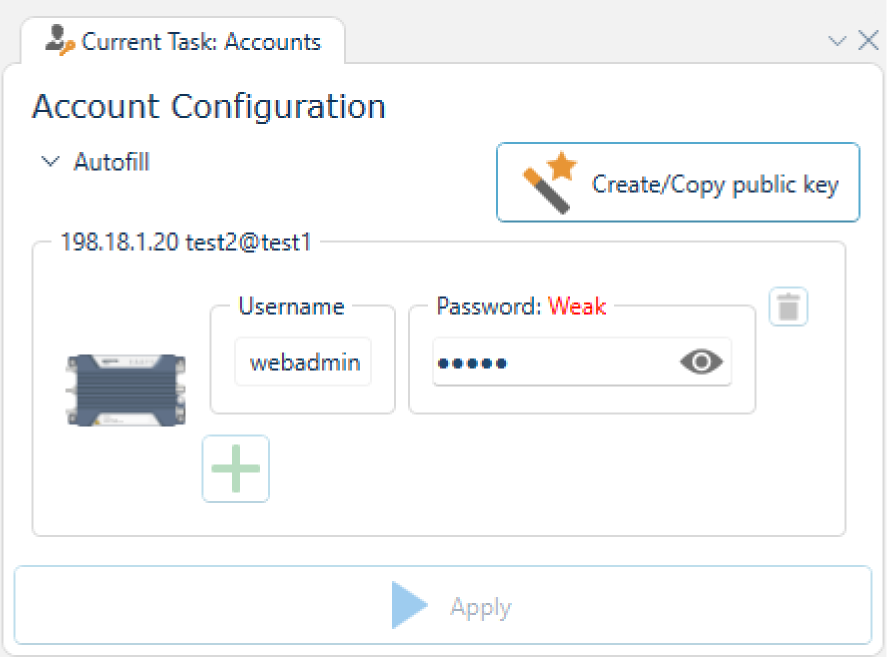
Hash
This box specifies the hashed version of the password, useful for copying authentication between devices without explicit knowledge of passwords.

> 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:



Current Task: Accounts

Account Configuration

Autofill

198.18.1.20 test2@test1

Username: webadmin

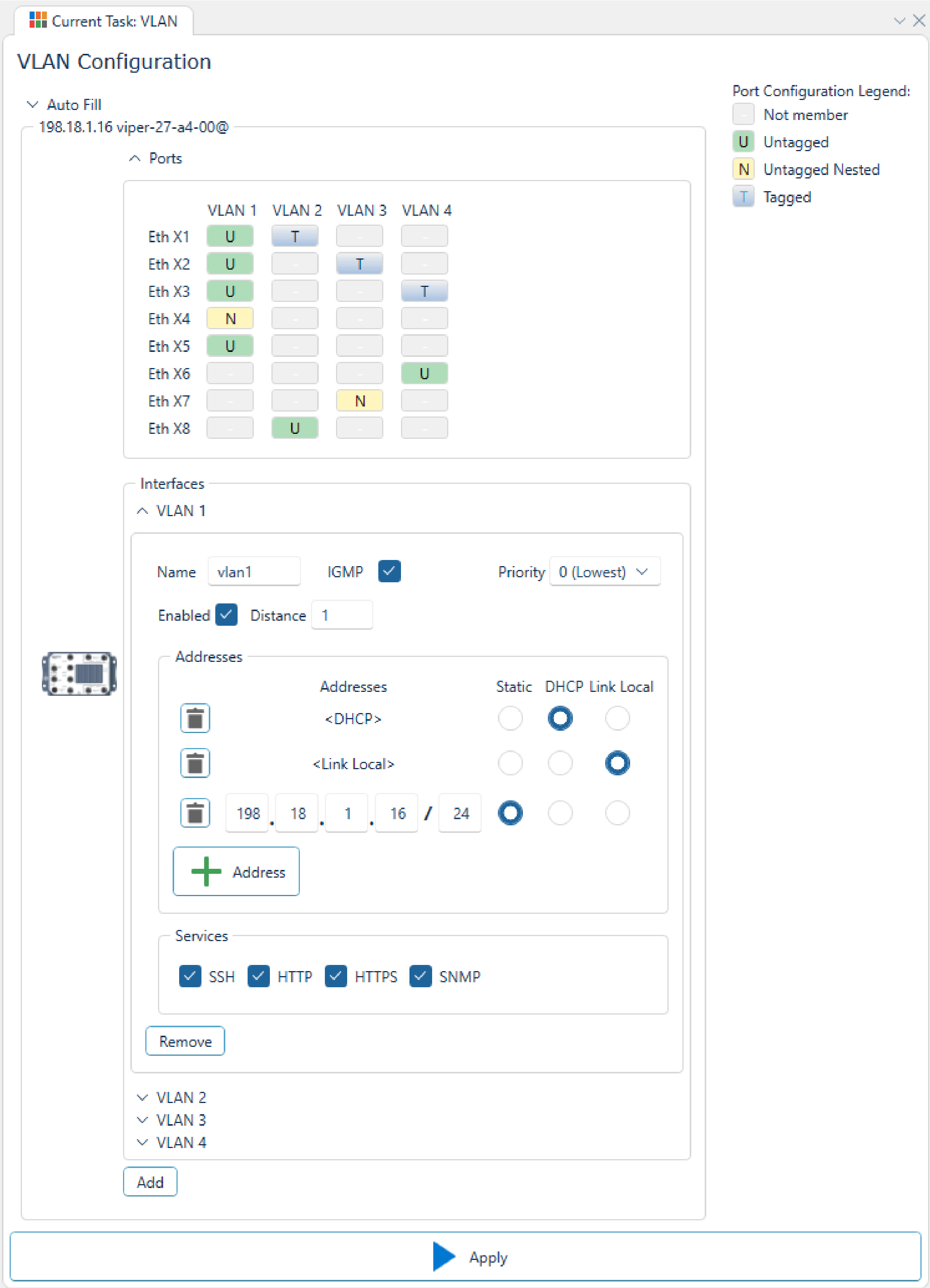
Password: Weak

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

4.3.3.3. Virtual Local Area Networks (VLAN)

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

Interface components



Current Task: VLAN

VLAN Configuration

Auto Fill

198.18.1.16 viper-27-a4-00@

Ports

	VLAN 1	VLAN 2	VLAN 3	VLAN 4
Eth X1	U	T	-	-
Eth X2	U	-	T	-
Eth X3	U	-	-	T
Eth X4	N	-	-	-
Eth X5	U	-	-	-
Eth X6	-	-	-	U
Eth X7	-	-	N	-
Eth X8	-	U	-	-

Port Configuration Legend:

□	Not member
U	Untagged
N	Untagged Nested
T	Tagged

Interfaces

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

VLAN 2

VLAN 3

VLAN 4

Add

▶ Apply

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 X1 through X5 are present but untagged. Port 4 has policy nesting enabled.
- For VLAN 2, Port X1 is tagged (which does not conflict with untagged VLAN 1 traffic on the same port) and Port X8 is untagged
- For VLAN 3, Port X2 is tagged, and Port X7 is untagged with policy nesting enabled.
- For VLAN 4, Port X3 is tagged, and Port X6 is untagged.

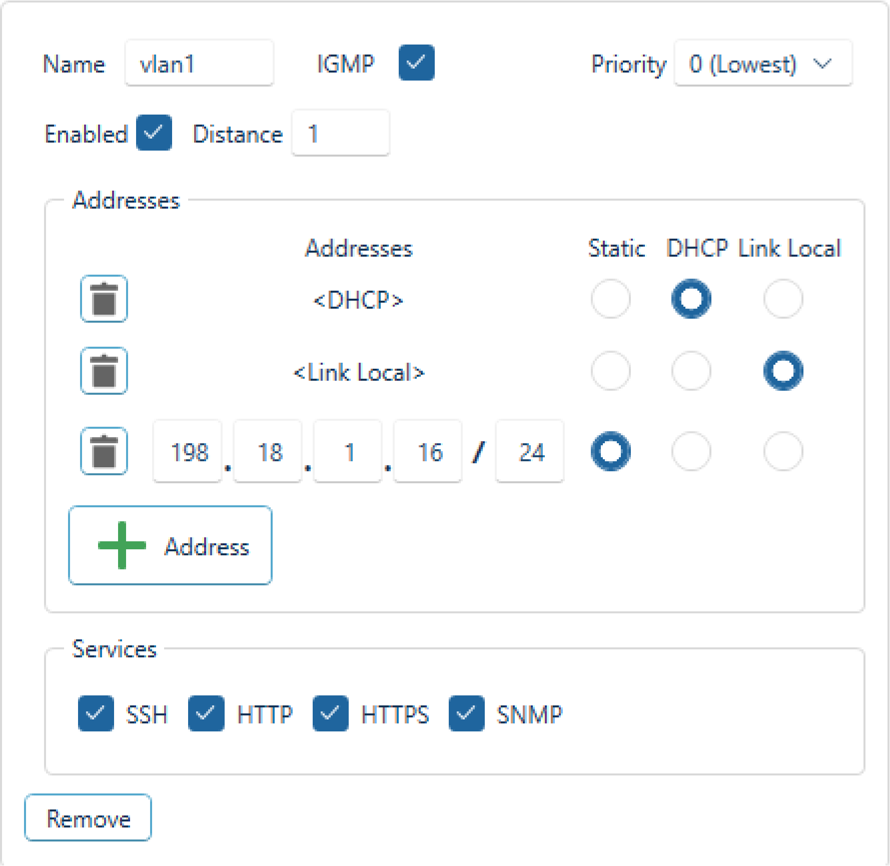
In order to change the port association, simply toggle the associated boxes

> *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.

WeOS 5



Name vlan1 IGMP ☑ Priority 0 (Lowest)

Enabled ☑ Distance 1

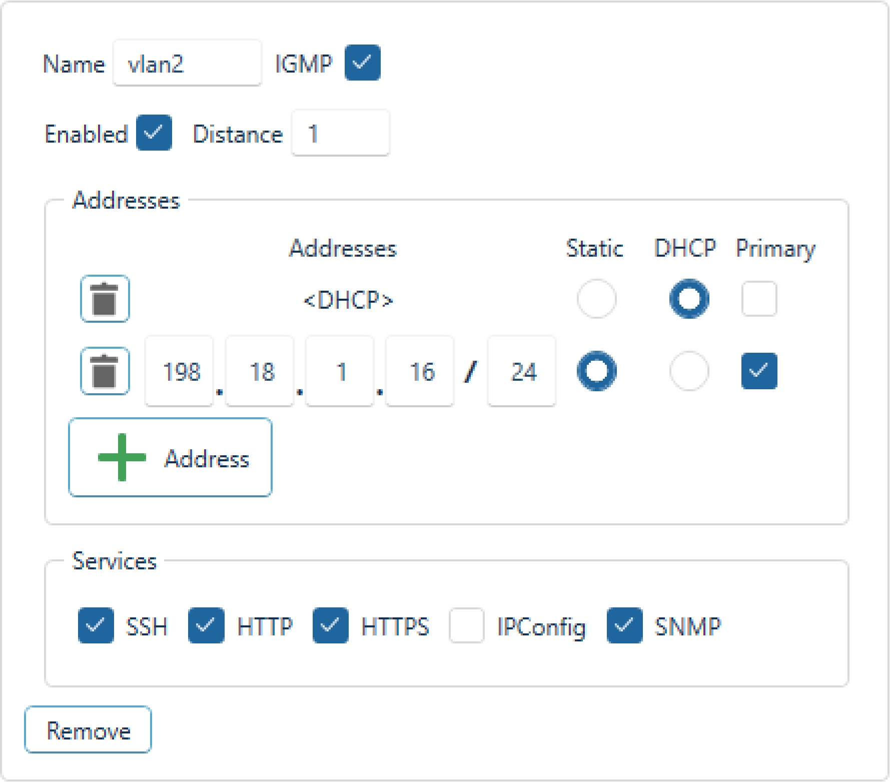
Addresses

Addresses	Static	DHCP	Link Local
<DHCP>	○	●	○
<Link Local>	○	○	●
198 . 18 . 1 . 16 / 24	●	○	○

Services

☑ SSH ☑ HTTP ☑ HTTPS ☑ SNMP

WeOS 4



Name vlan2 IGMP ☑

Enabled ☑ Distance 1

Addresses

Addresses	Static	DHCP	Primary
<DHCP>	○	●	□
198 . 18 . 1 . 16 / 24	●	○	☑

Services

☑ SSH ☑ HTTP ☑ HTTPS □ IPConfig ☑ SNMP

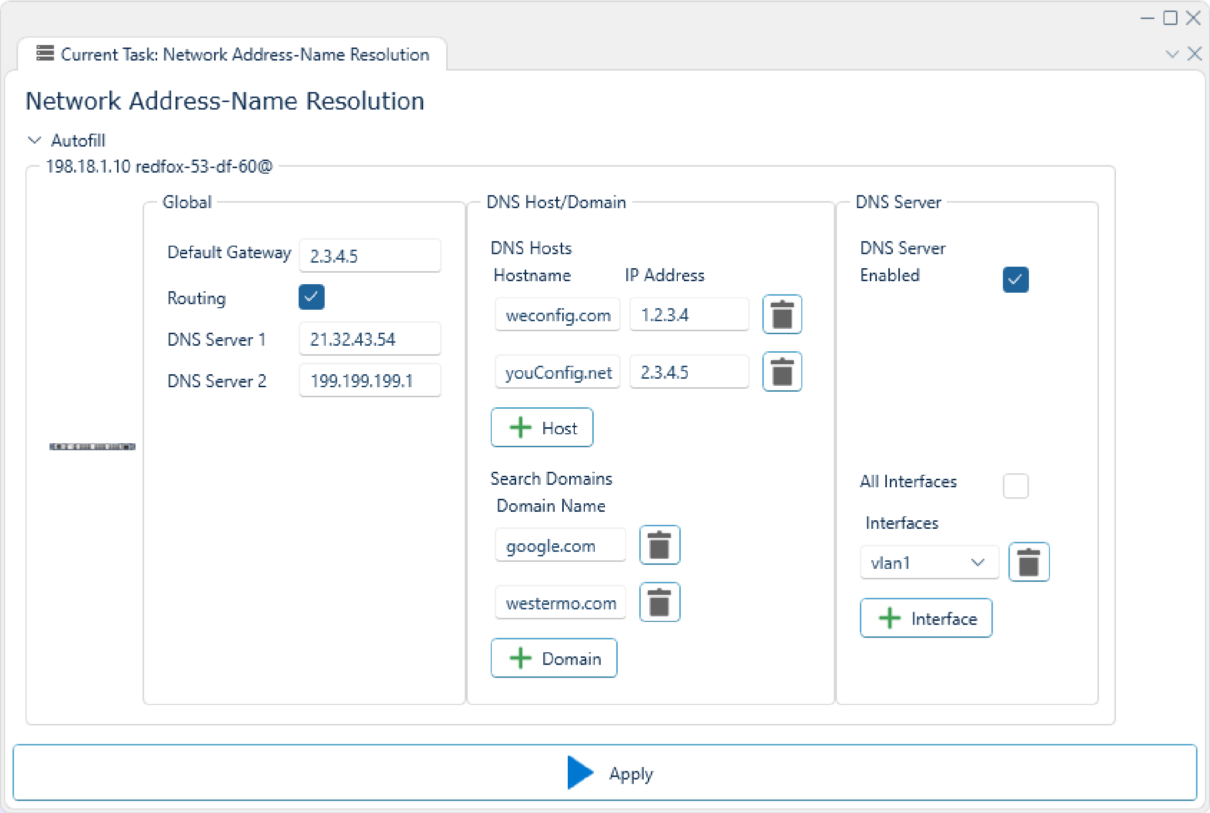
Info

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

4.3.3.4. Network Address-Name Resolution

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

Interface components


Current Task: Network Address-Name Resolution
Network Address-Name Resolution
Autofill
198.18.1.10 redfox-53-df-60@
Global
DNS Host/Domain
DNS Server
+ Host
+ Domain
+ Interface
Apply

Global

Default Gateway: 2.3.4.5

Routing: ☑

DNS Server 1: 21.32.43.54

DNS Server 2: 199.199.199.1

DNS Hosts

Hostname	IP Address
weconfig.com	1.2.3.4
youConfig.net	2.3.4.5

Search Domains

Domain Name
google.com
westermo.com

DNS Server Enabled: ☑

All Interfaces: ☐

Interfaces: vlan1

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.

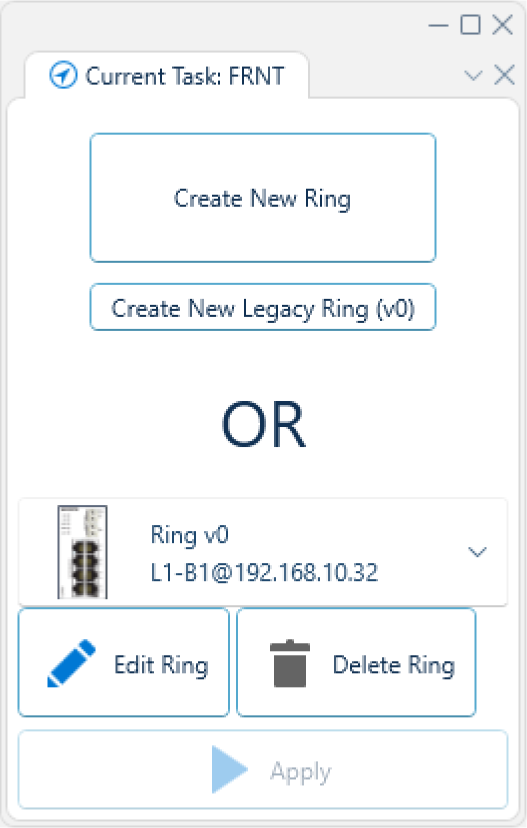
The third and final group "DNS Server" allows the user to configure the device to act as a DNS server, and to specify which interfaces that will be part of said server.

4.3.3.5. Layer 2 Redundancy

4.3.3.5.1. Fast Reconfiguring Network Topology (FRNT)

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

Interface Components



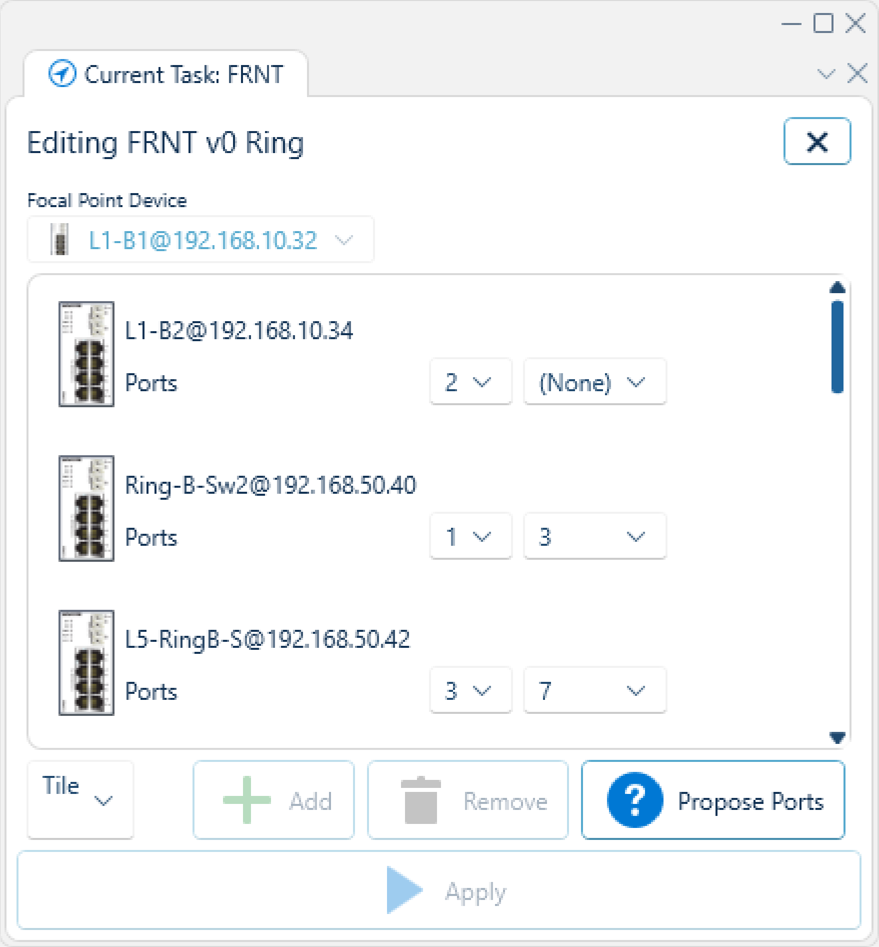
To create a new FRNT ring, click either Create New Ring or Create New Legacy Ring (v0). To edit an existing FRNT ring, select the ring in the drop down menu, and click Edit Ring.

> 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.

FRNT v0



Current Task: FRNT

Editing FRNT v0 Ring

Focal Point Device: L1-B1@192.168.10.32

L1-B2@192.168.10.34
Ports 2 (None)

Ring-B-Sw2@192.168.50.40
Ports 1 3

L5-RingB-S@192.168.50.42
Ports 3 7

Focal point device:

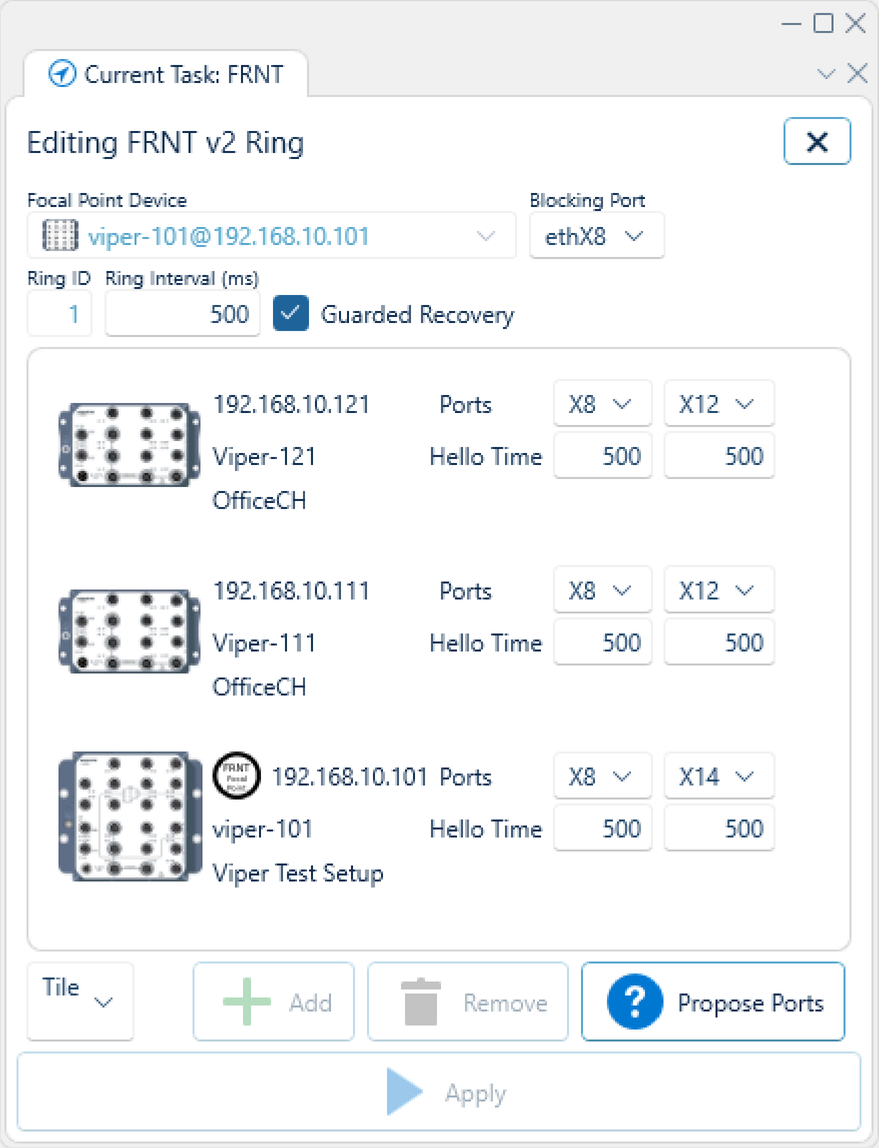
- No focal point options are applicable

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



Current Task: FRNT

Editing FRNT v2 Ring

Focal Point Device: viper-101@192.168.10.101

Blocking Port: ethX8

Ring ID: 1

Ring Interval (ms): 500

☑ Guarded Recovery

192.168.10.121	Ports	X8	X12
Viper-121	Hello Time	500	500
OfficeCH			
192.168.10.111	Ports	X8	X12
Viper-111	Hello Time	500	500
OfficeCH			
192.168.10.101	Ports	X8	X14
viper-101	Hello Time	500	500
Viper Test Setup			

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.
- 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:

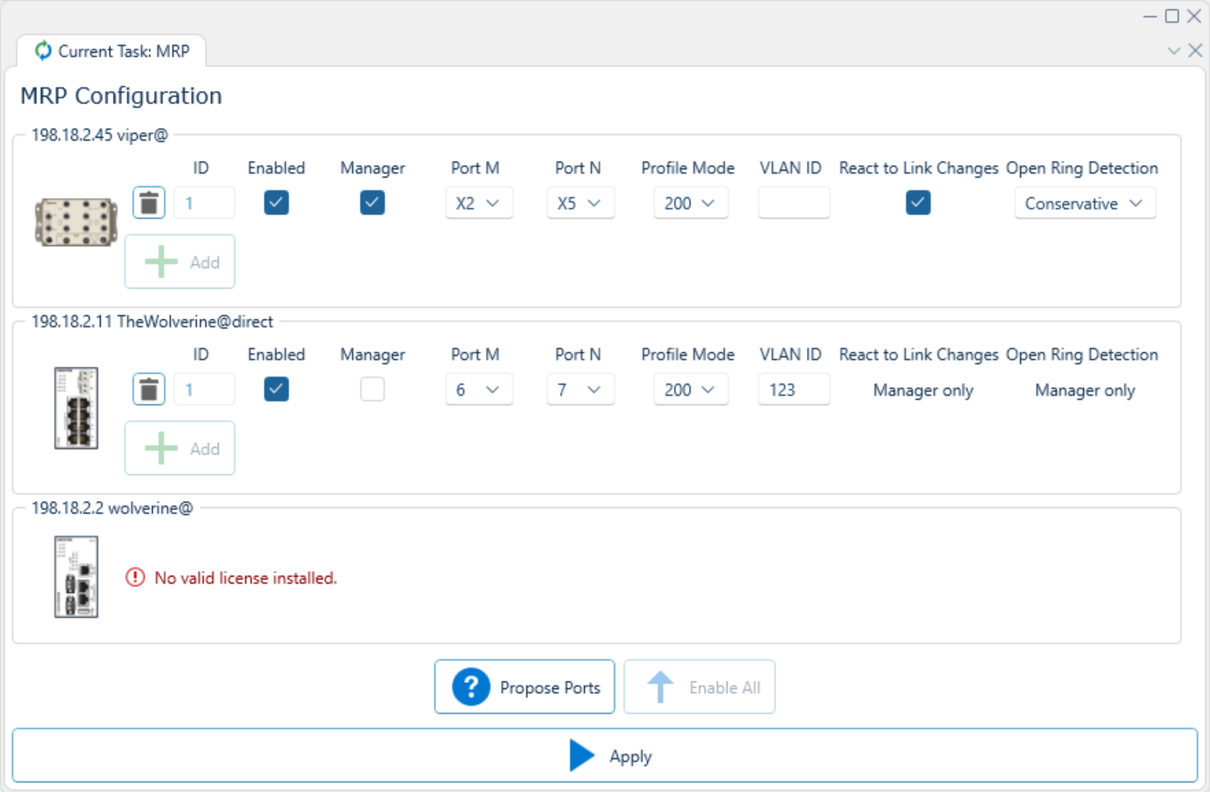
- First port
  - Optionally override the default hello time
- Second port
  - Optionally override the default hello time

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

4.3.3.5.2. Media Redundancy Protocol (MRP)

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

Interface Components



Current Task: MRP

MRP Configuration

198.18.2.45 viper@

ID	Enabled	Manager	Port M	Port N	Profile Mode	VLAN ID	React to Link Changes	Open Ring Detection
1	☑	☑	X2	X5	200		☑	Conservative

198.18.2.11 TheWolverine@direct

ID	Enabled	Manager	Port M	Port N	Profile Mode	VLAN ID	React to Link Changes	Open Ring Detection
1	☑	□	6	7	200	123	Manager only	Manager only

198.18.2.2 wolverine@

No valid license installed.

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
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

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

i Info

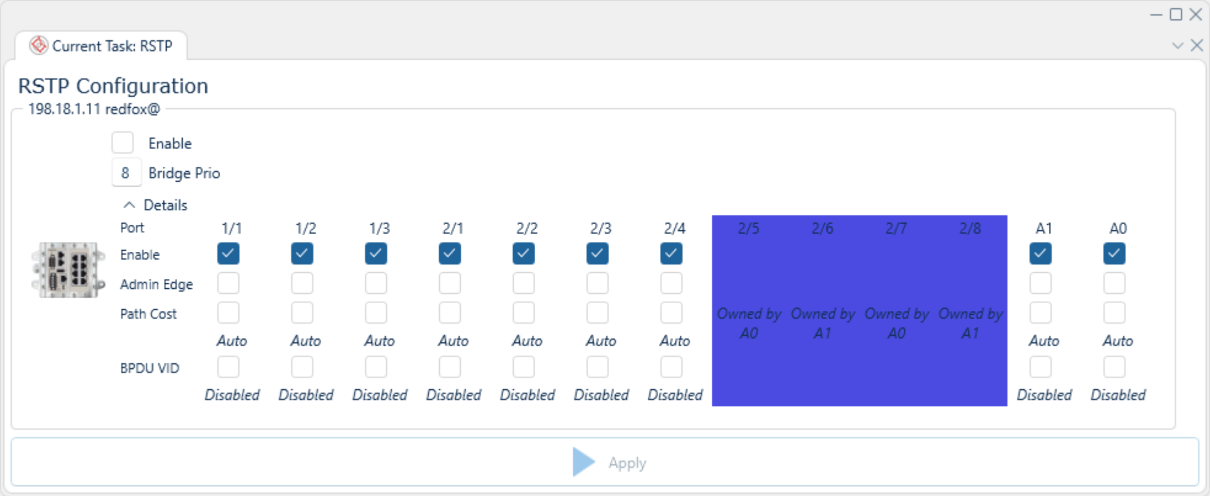
For devices running WeOS 5.11 and newer, up to two ring configurations is supported. Although at most one may configure the device as an MRP client.

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

4.3.3.5.3. Rapid Spanning Tree Protocol (RSTP)

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

Interface Components



Current Task: RSTP

RSTP Configuration

198.18.1.11 redfox@

□ Enable

8 Bridge Prio

Details

Port	1/1	1/2	1/3	2/1	2/2	2/3	2/4	2/5	2/6	2/7	2/8	A1	A0
Enable	☑	☑	☑	☑	☑	☑	☑	Owned by A0	Owned by A1	Owned by A0	Owned by A1	☑	☑
Admin Edge	□	□	□	□	□	□	□					□	□
Path Cost	□ Auto	□ Auto	□ Auto	□ Auto	□ Auto	□ Auto	□ Auto					□ Auto	□ Auto
BPDU VID	□ Disabled	□ Disabled	□ Disabled	□ Disabled	□ Disabled	□ Disabled	□ Disabled					□ Disabled	□ Disabled

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

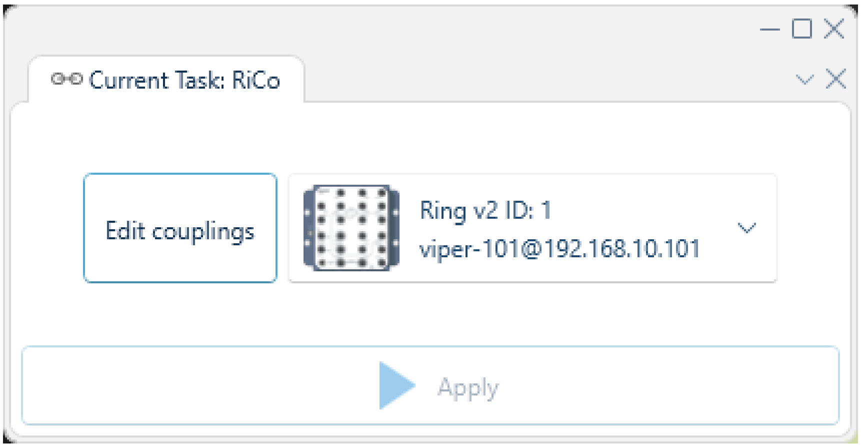
Edit the fields directly in the list. To enable RSTP or to set Admin Edge on port level click the “down-arrow” icon to the left of each device in the list.

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

4.3.3.5.4. Ring Coupling (RiCo)

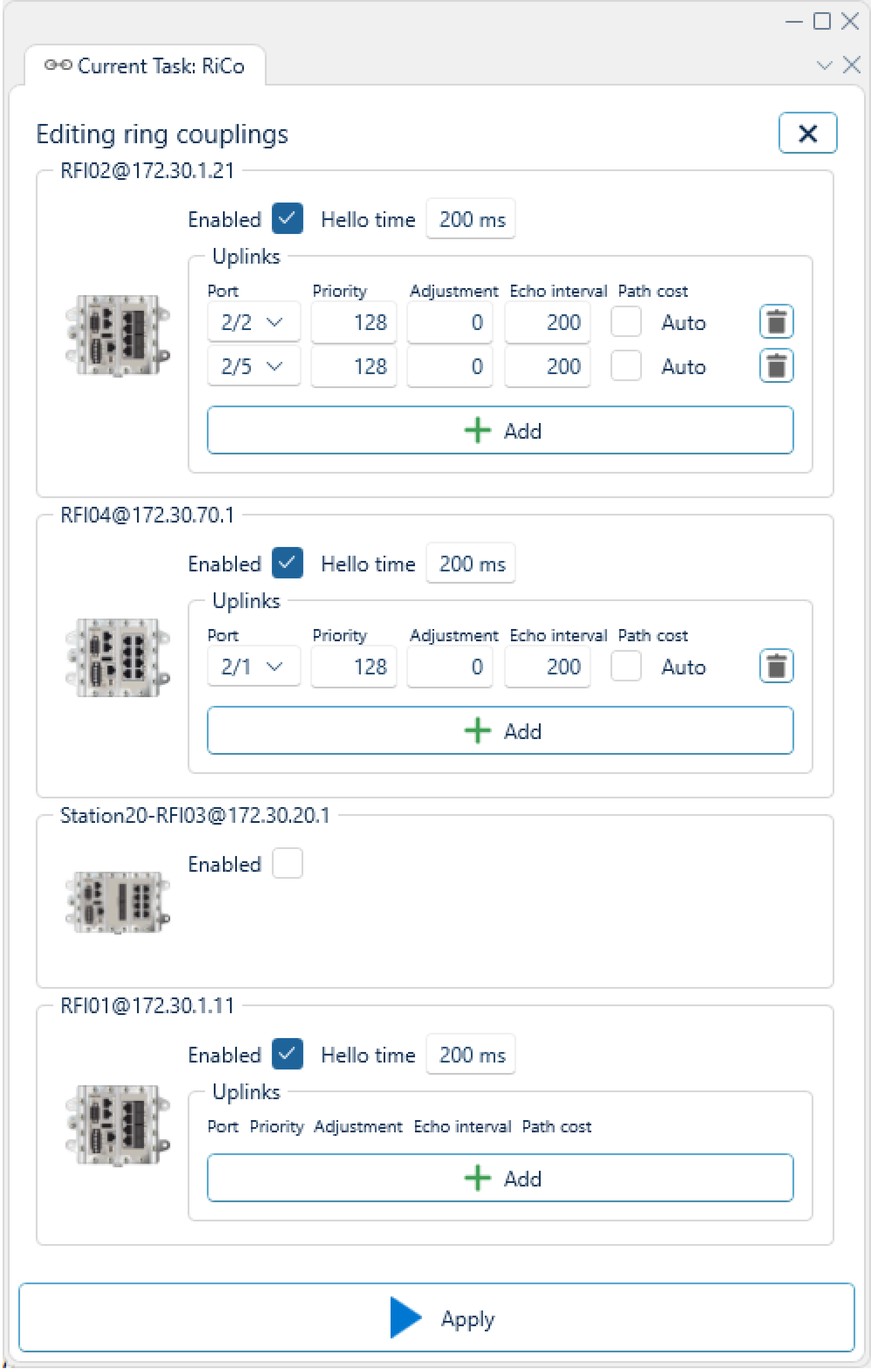
RiCo is a task that allows for configuration of ring coupling for rings in the network.

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



RFI02@172.30.1.21

Enabled ☑ Hello time 200 ms

Uplinks

Port	Priority	Adjustment	Echo interval	Path cost
2/2	128	0	200	□ Auto
2/5	128	0	200	□ Auto

RFI04@172.30.70.1

Enabled ☑ Hello time 200 ms

Uplinks

Port	Priority	Adjustment	Echo interval	Path cost
2/1	128	0	200	□ Auto

Station20-RFI03@172.30.20.1

Enabled □

RFI01@172.30.1.11

Enabled ☑ Hello time 200 ms

Uplinks

Port	Priority	Adjustment	Echo interval	Path cost

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

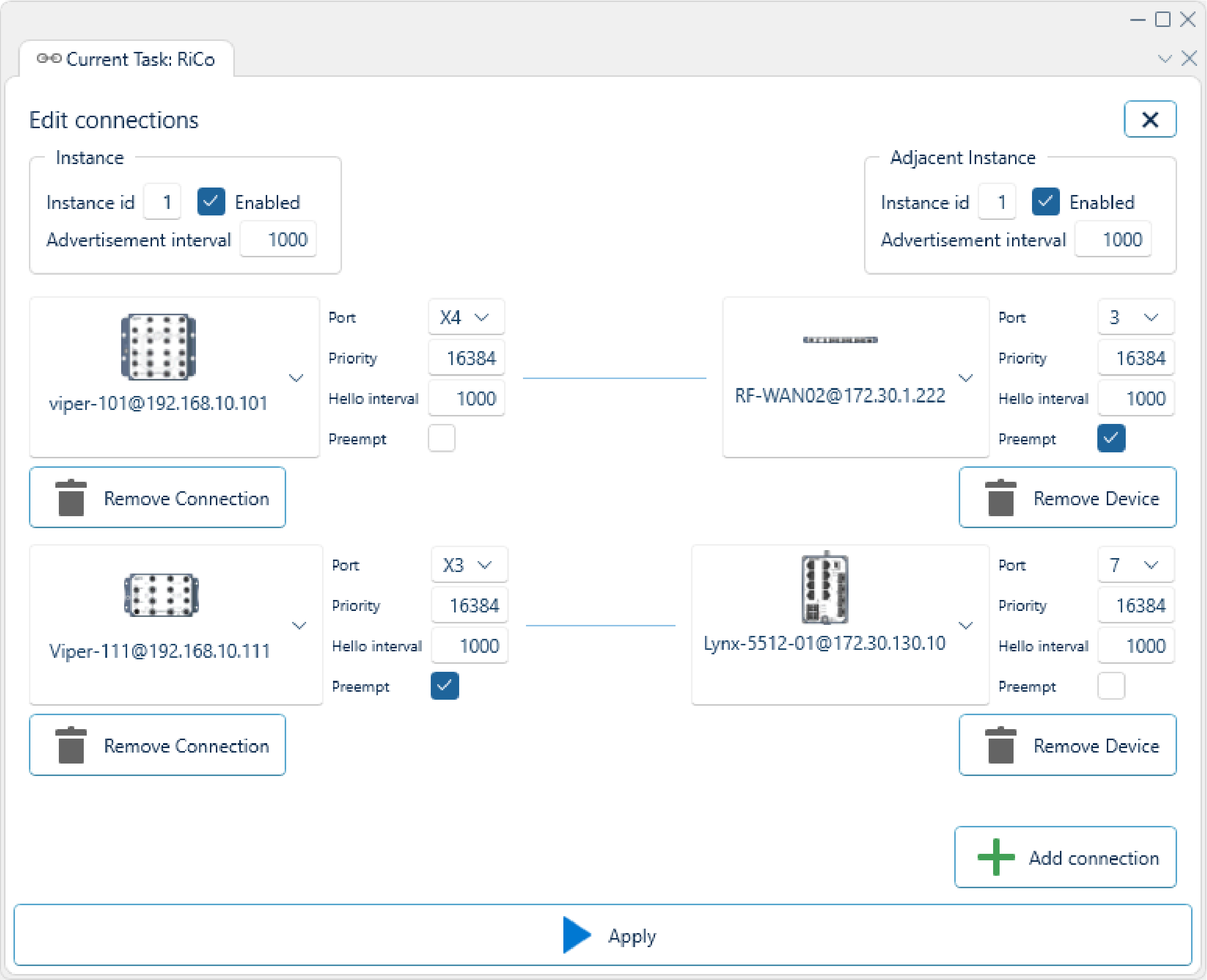
To remove an uplink click the trashcan button on the row that should be removed.

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.

> *i* 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:



Instance

Instance id 1 ☑ Enabled

Advertisement interval 1000

Adjacent Instance

Instance id 1 ☑ Enabled

Advertisement interval 1000

viper-101@192.168.10.101 — Port X4, Priority 16384, Hello interval 1000, Preempt □

RF-WAN02@172.30.1.222 — Port 3, Priority 16384, Hello interval 1000, Preempt ☑

Viper-111@192.168.10.111 — Port X3, Priority 16384, Hello interval 1000, Preempt ☑

Lynx-5512-01@172.30.130.10 — Port 7, Priority 16384, Hello interval 1000, Preempt □

Where the left hand side corresponds to the selected ring instance, and the righthand side corresponds to any detected adjacent ring instance. Herein, you may specify:

- The instance ID
- Whether or not the RiCo instance is enabled

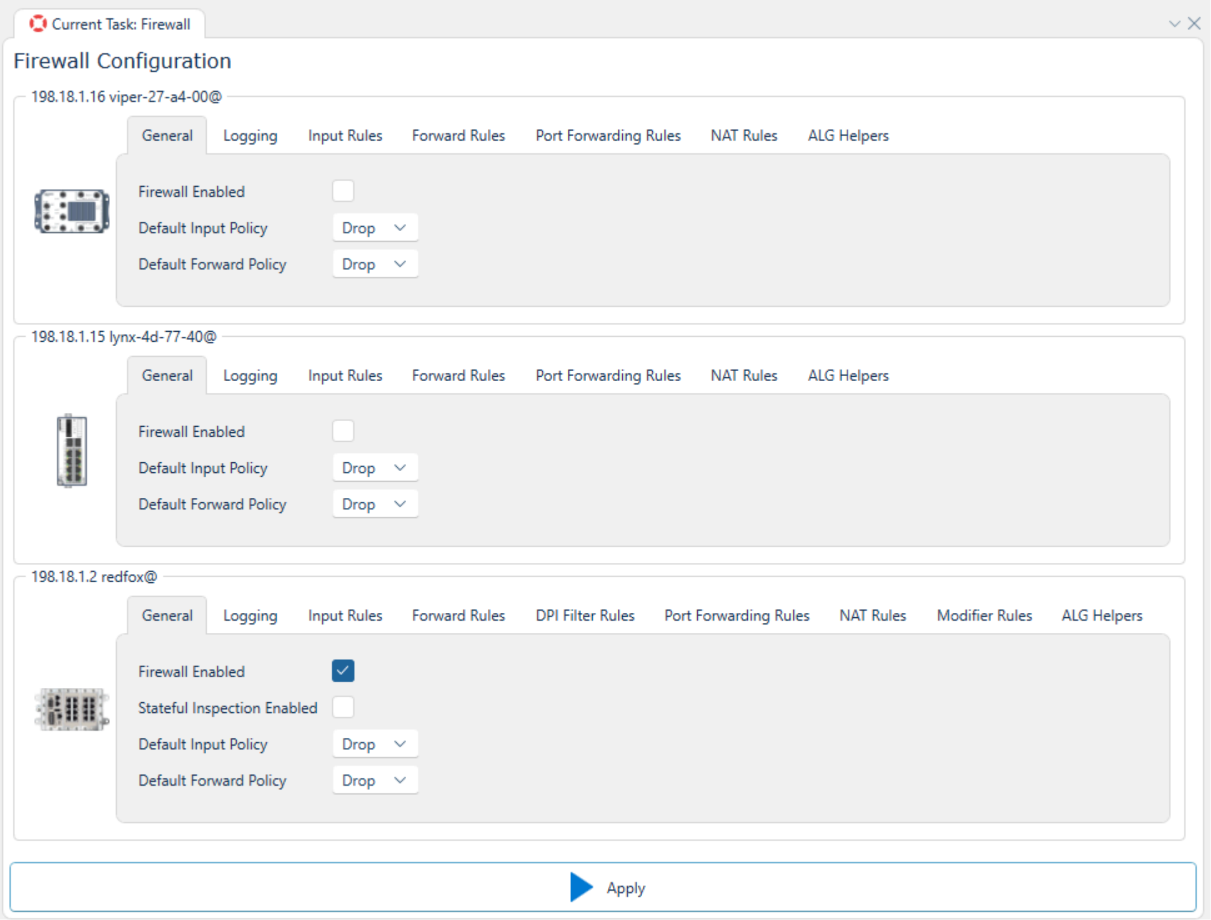
- The advertisment interval. Following the instance options, you may specify a number of connections, added via the Add connection button at the bottom right of the interface. In the example depicted above, two connections have been added, each with a port, a priority, a hello interval and a preempt 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.

4.3.3.6. Firewall

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



Firewall Configuration

198.18.1.16 viper-27-a4-00@

Firewall Enabled □
Default Input Policy: Drop
Default Forward Policy: Drop

198.18.1.15 lynx-4d-77-40@

Firewall Enabled □
Default Input Policy: Drop
Default Forward Policy: Drop

198.18.1.2 redfox@

Firewall Enabled ☑
Stateful Inspection Enabled □
Default Input Policy: Drop
Default Forward Policy: Drop

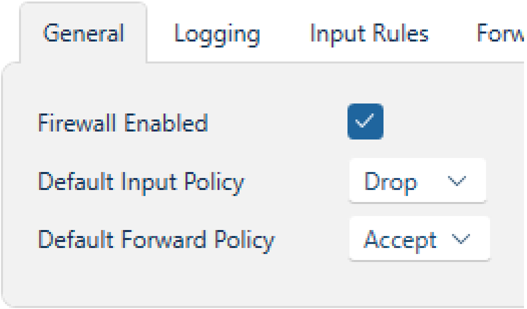
Depicted above you can observe how the interface might appears for a selection of two WeOS 5 devices and one WeOS 4 device. Where the configuration options for each device is divided into several tabs with the following contents:

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	Yes	Yes	Yes

Tab	Description	WeOS 4/5 Standard	WeOS 4 Extended	WeOS 5 Extended
	packets directed to the device			
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
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
ALG Helpers	Shorthand Rules for specific protocols	No	Yes	Yes

General



Firewall Enabled ☑

Default Input Policy Drop

Default Forward Policy Accept

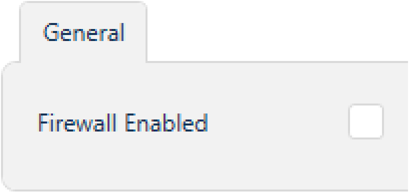
The general tab allows for enabling or disabling the firewall on the device, and setting the default policy for the firewall. The default policy can be set to either `ACCEPT` or `DROP`. The default policy is the action that the firewall will take if no rules are matched.

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

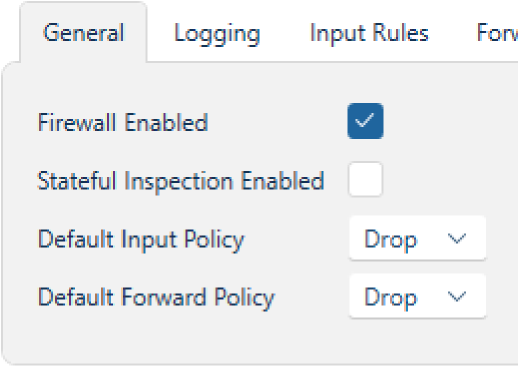
Default Forward Policy is only available on Extended devices.

WeOS 4



Firewall Enabled □

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



Firewall Enabled ☑

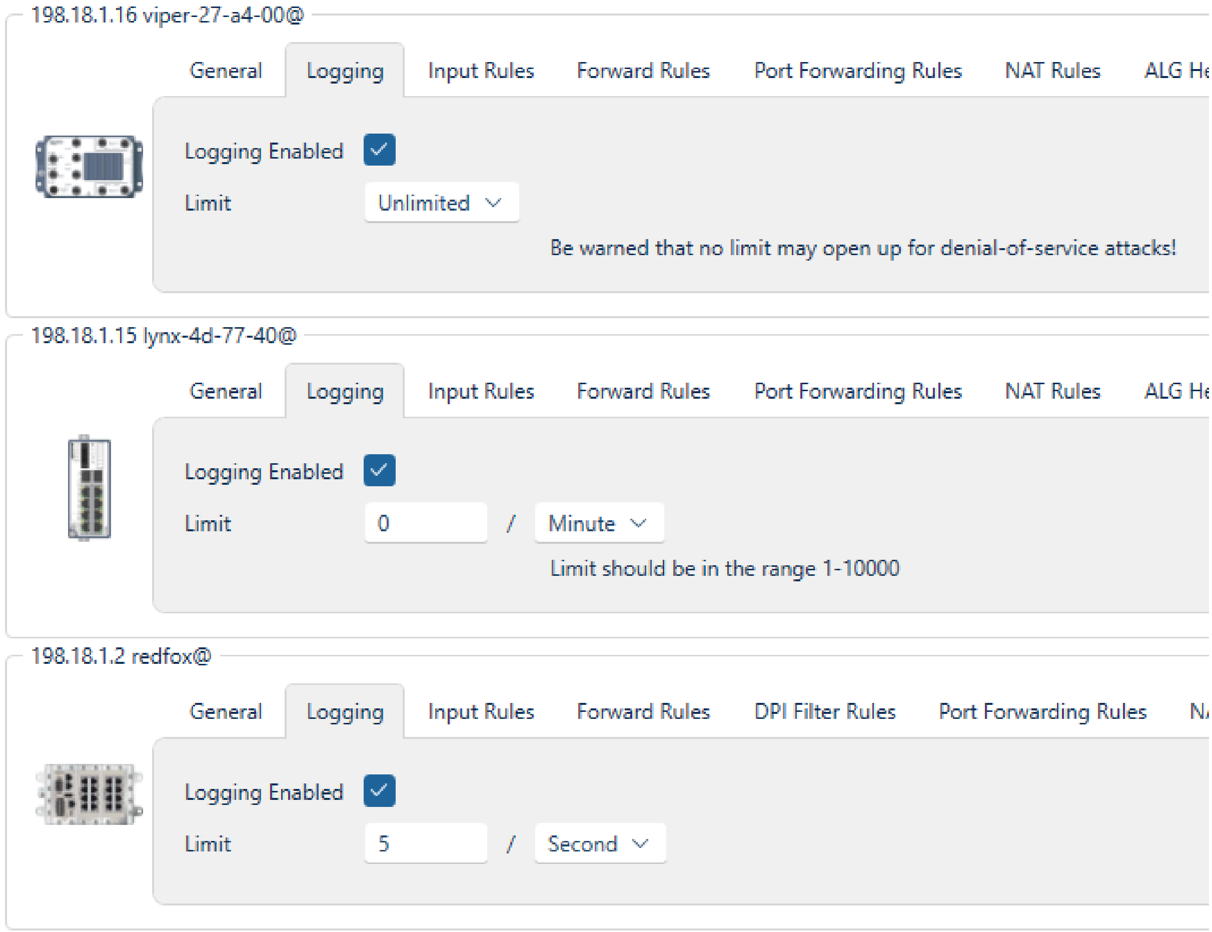
Stateful Inspection Enabled □

Default Input Policy: Drop

Default Forward Policy: Drop

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

Logging



198.18.1.16 viper-27-a4-00@

Logging Enabled ☑

Limit: Unlimited

Be warned that no limit may open up for denial-of-service attacks!

198.18.1.15 lynx-4d-77-40@

Logging Enabled ☑

Limit: 0 / Minute

Limit should be in the range 1-10000

198.18.1.2 redfox@

Logging Enabled ☑

Limit: 5 / Second

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.

> *i* 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.

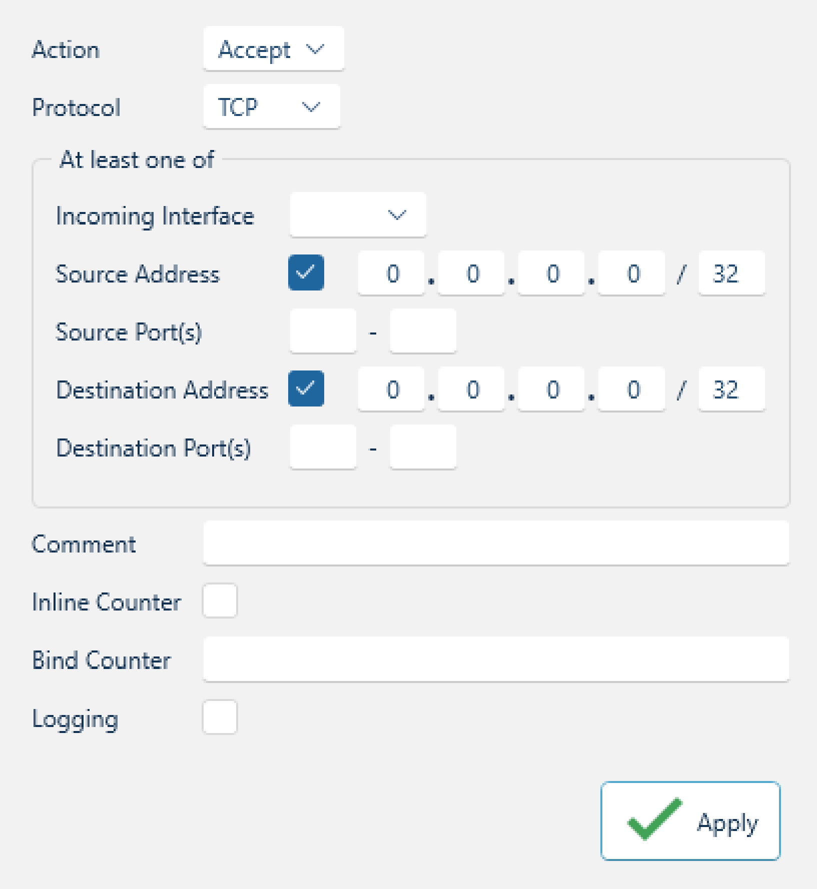
WeOS 5 defaults with no Input Rules, while WeOS 4 defaults with rules that allow ICMP access to the device.

> *i* 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 Inupt 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:



Action: Accept

Protocol: TCP

At least one of

Incoming Interface:

Source Address: ☑ 0 . 0 . 0 . 0 / 32

Source Port(s): -

Destination Address: ☑ 0 . 0 . 0 . 0 / 32

Destination Port(s): -

Comment:

Inline Counter: □

Bind Counter:

Logging: □

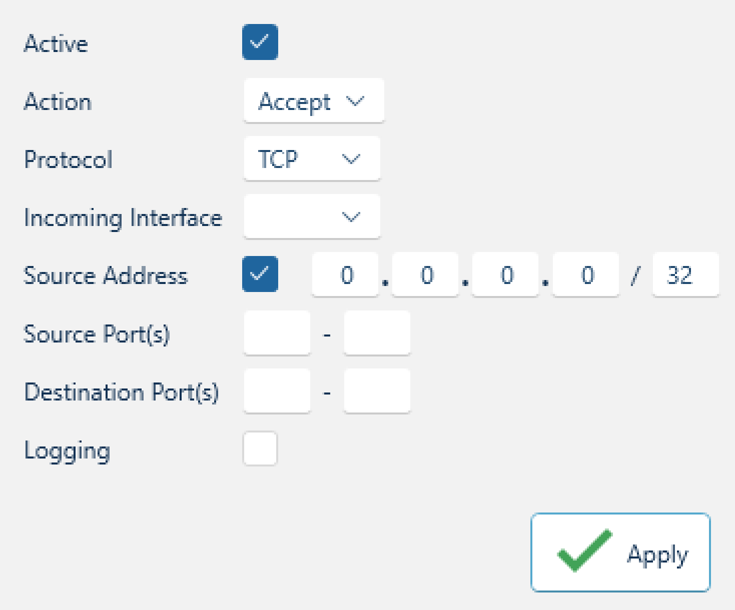
Field	Description	Required	Notes
Action	Accept or drop traffic matching the specified rule	Yes	-
Protocol	The protocol type of the IP payload. Typically TCP or UDP but the filtering can also be made to match other	Yes	Will affect what fields are available

Field	Description	Required	Notes
	protocols such as ICMP and ESP.		
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
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 Inupt 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:



Active	☑
Action	Accept
Protocol	TCP
Incoming Interface	
Source Address	☑ 0 . 0 . 0 . 0 / 32
Source Port(s)	-
Destination Port(s)	-
Logging	□

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

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

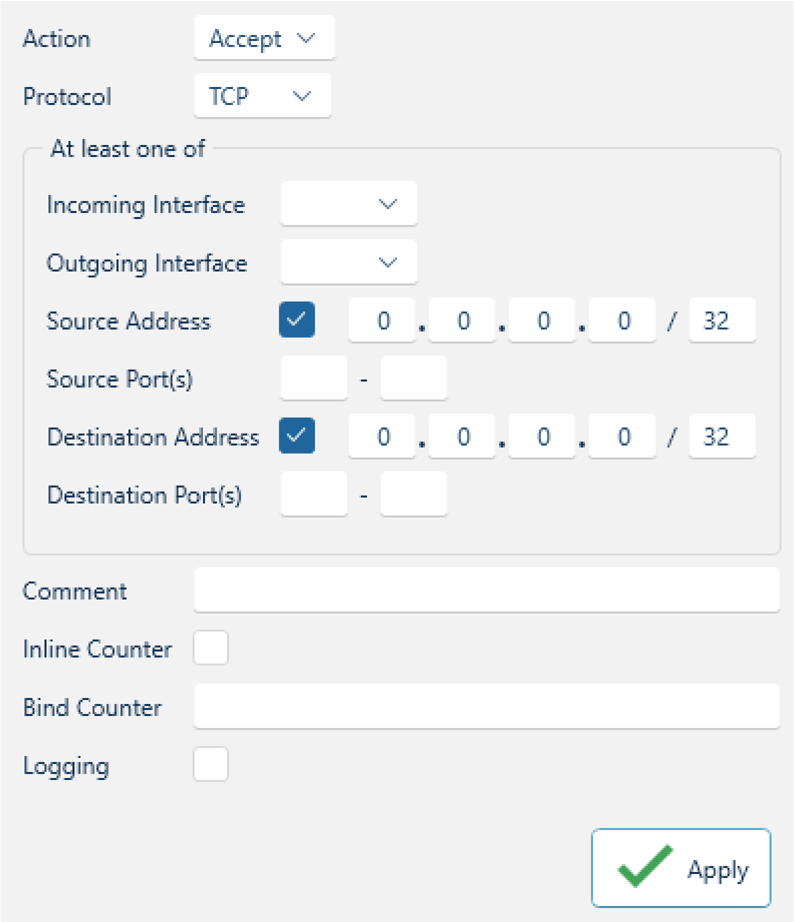
The forward rules are checked after the routing decision and applies to packets that are not directed to the local device.

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:



Action	Accept
Protocol	TCP

At least one of

Incoming Interface	
Outgoing Interface	
Source Address	☑ 0 . 0 . 0 . 0 / 32
Source Port(s)	-
Destination Address	☑ 0 . 0 . 0 . 0 / 32
Destination Port(s)	-

Comment	
Inline Counter	□
Bind Counter	
Logging	□

Field	Description	Required	Notes
Action	Accept or drop traffic matching the specified rule	Yes	-

Field	Description	Required	Notes
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
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`:

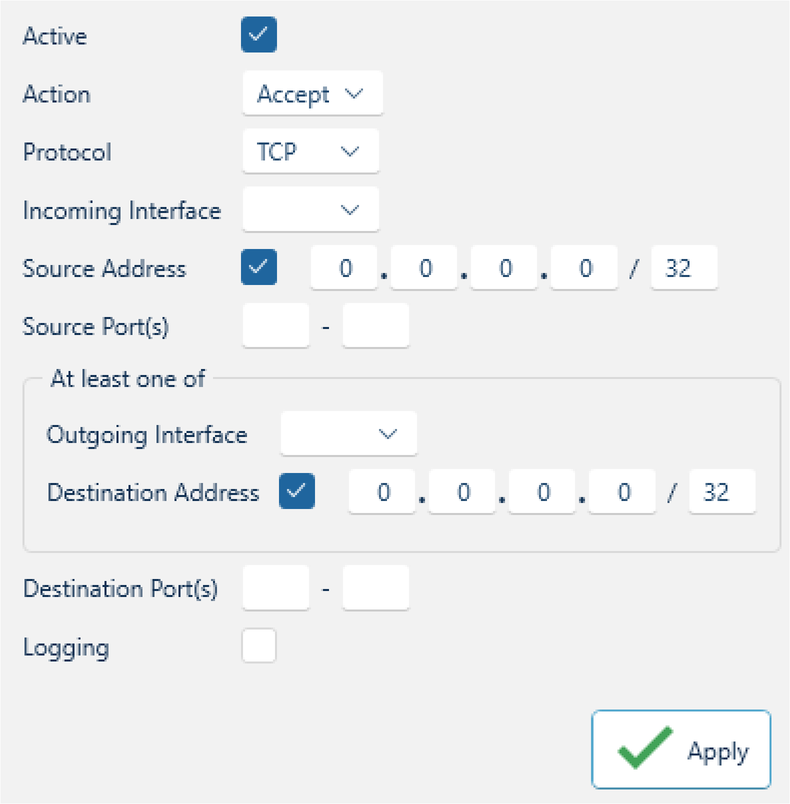
- At least one of `Incoming Interface`, `Outgoing Interface`, `Source Address`, `Destination Address`, `Source Port(s)`, or `Destination Port(s)` must be set.

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:



Active ☑

Action: Accept

Protocol: TCP

Incoming Interface:

Source Address ☑ 0 . 0 . 0 . 0 / 32

Source Port(s) -

At least one of

- Outgoing Interface:
- Destination Address ☑ 0 . 0 . 0 . 0 / 32

Destination Port(s) -

Logging □

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

ℹ Info

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

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.

ℹ Info

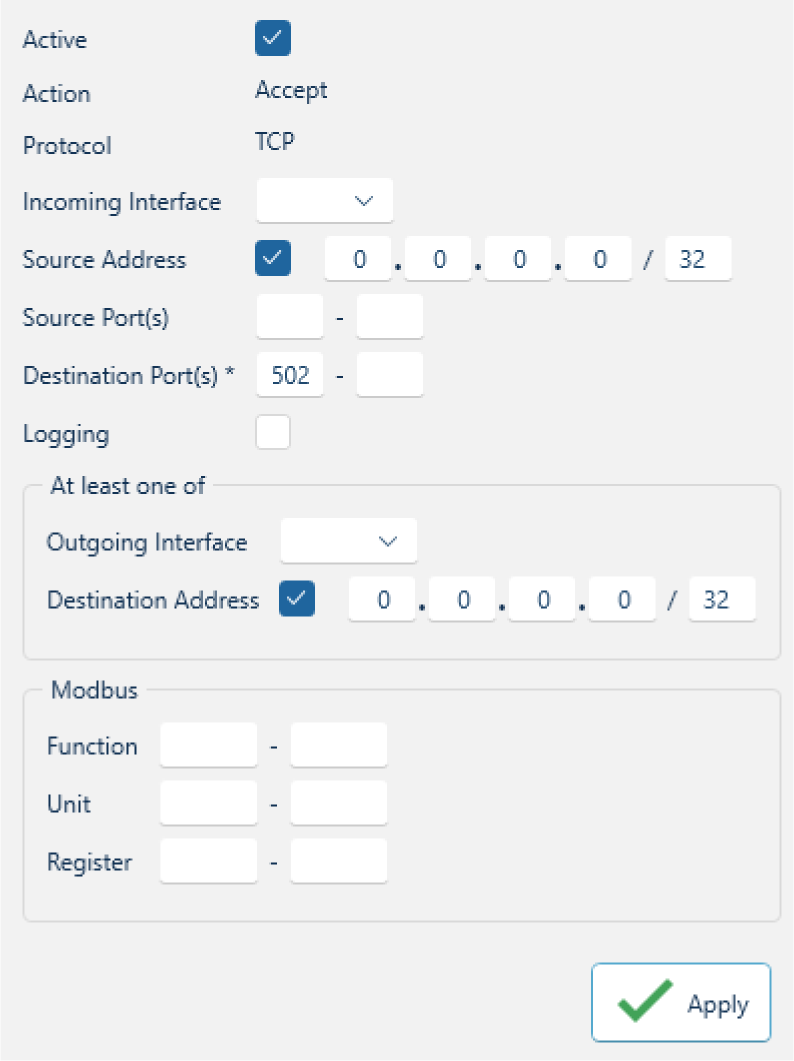
Only avaible for Extended WeOS 4 devices.

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:



Active ☑

Action Accept

Protocol TCP

Incoming Interface

Source Address ☑ 0 . 0 . 0 . 0 / 32

Source Port(s) -

Destination Port(s) * 502 -

Logging □

At least one of

Outgoing Interface

Destination Address ☑ 0 . 0 . 0 . 0 / 32

Modbus

Function -

Unit -

Register -

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	-
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

Field	Description	Required	Notes
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	-

> *i* 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.

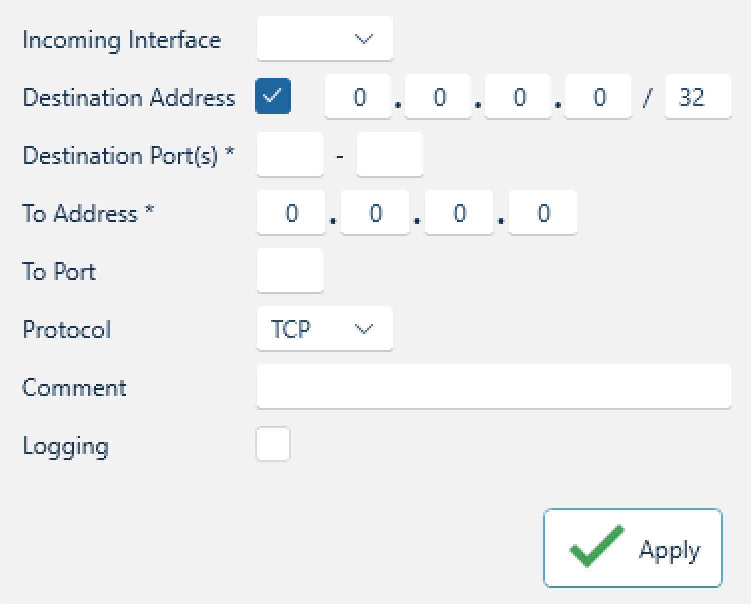
The default is no Port Forwarding Rules.

> *i* 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:

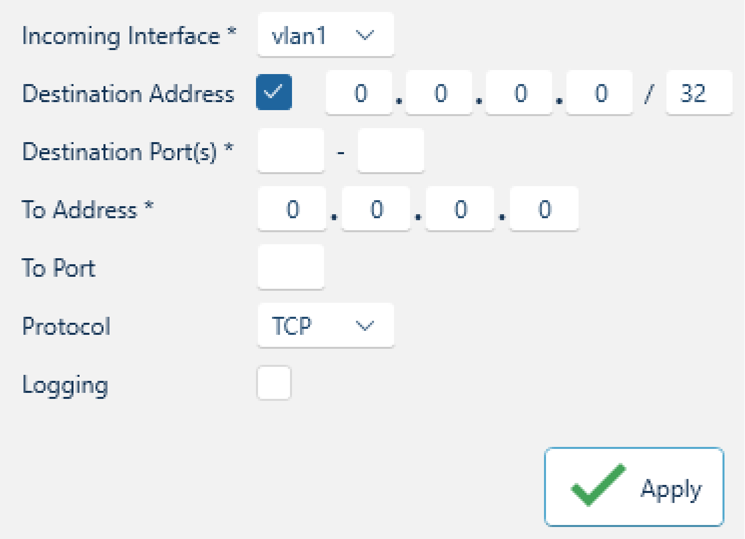


Incoming Interface	
Destination Address	☑ 0 . 0 . 0 . 0 / 32
Destination Port(s) *	-
To Address *	0 . 0 . 0 . 0
To Port	
Protocol	TCP
Comment	
Logging	□

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:



Incoming Interface *	vlan1
Destination Address	☑ 0 . 0 . 0 . 0 / 32
Destination Port(s) *	-
To Address *	0 . 0 . 0 . 0
To Port	
Protocol	TCP
Logging	□

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.

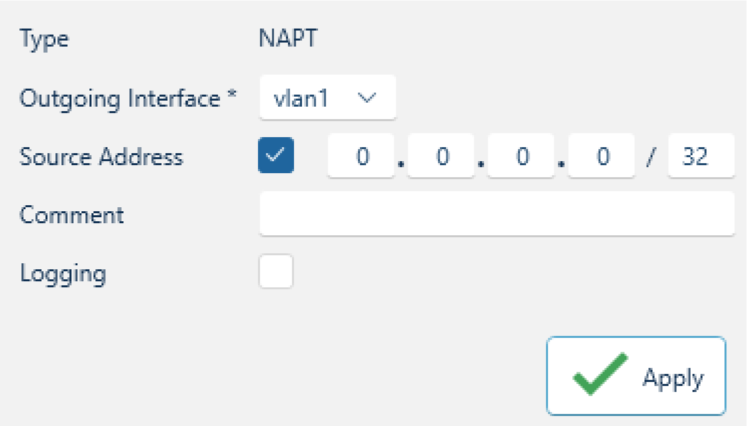
The default is no NAT Rules.

> *i* 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:



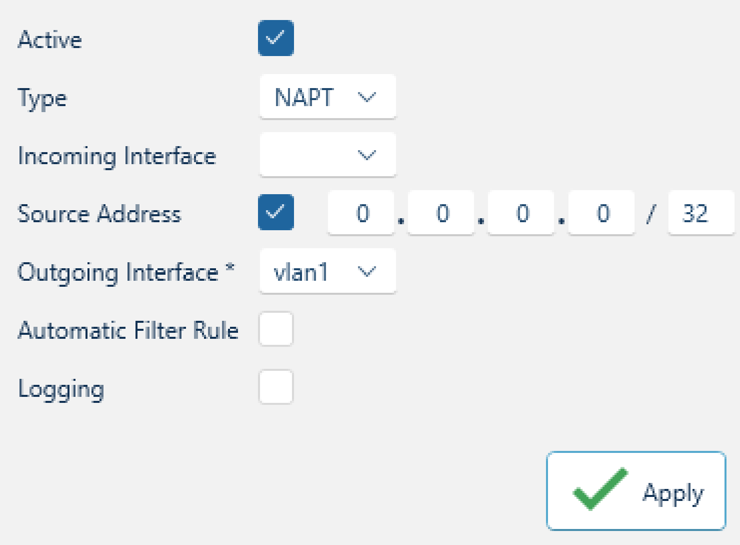
Type	NAPT
Outgoing Interface *	vlan1
Source Address	☑ 0 . 0 . 0 . 0 / 32
Comment	
Logging	□

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:

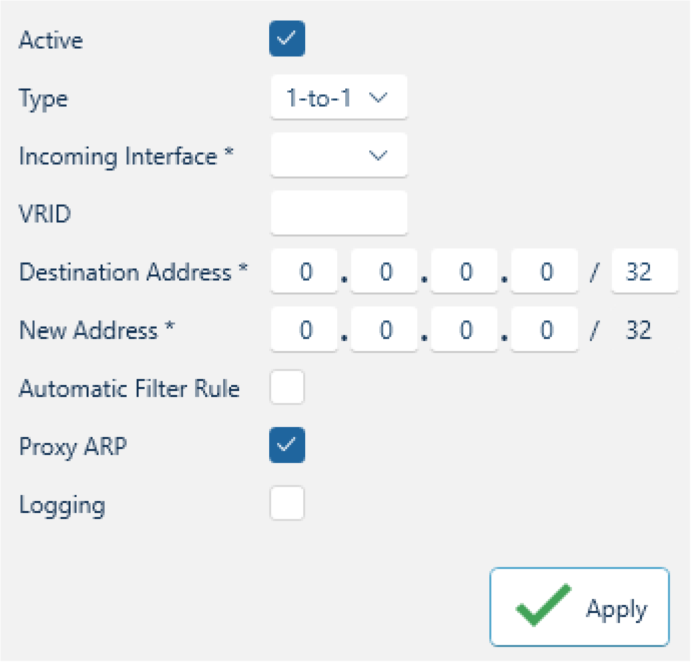
NAPT:



Active	☑
Type	NAPT
Incoming Interface	
Source Address	☑ 0 . 0 . 0 . 0 / 32
Outgoing Interface *	vlan1
Automatic Filter Rule	□
Logging	□

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:



Active	☑
Type	1-to-1
Incoming Interface *	
VRID	
Destination Address *	0 . 0 . 0 . 0 / 32
New Address *	0 . 0 . 0 . 0 / 32
Automatic Filter Rule	□
Proxy ARP	☑
Logging	□

Field	Description	Required	Notes
Active	Is this rule active?	Yes	-

Field	Description	Required	Notes
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 this option if you want to manage forwarding rules yourself.	No	-
Pryxy 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	-

Field	Description	Required	Notes
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.

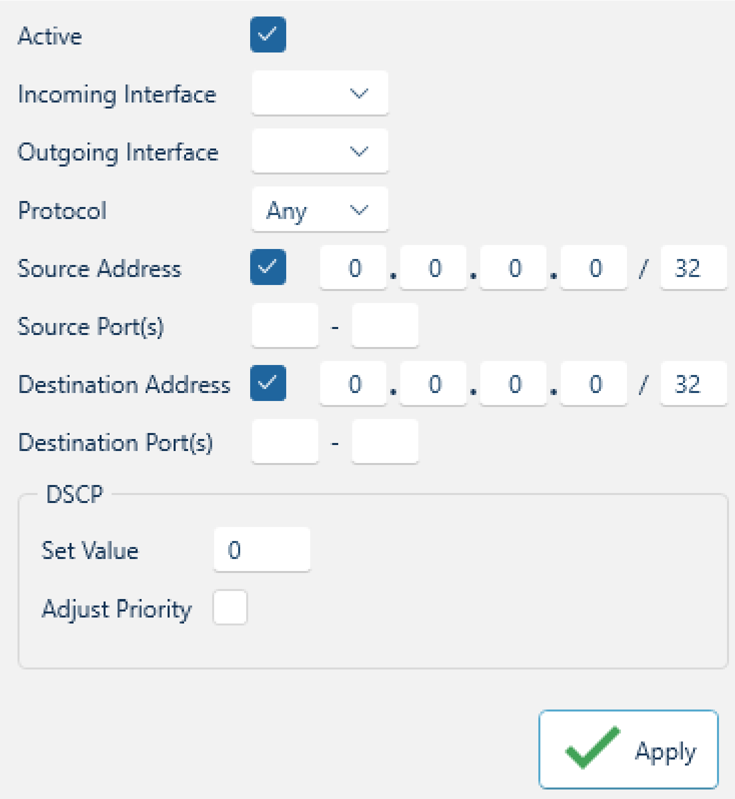
The default is no Modifier Rules.

> **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:



Active	☑
Incoming Interface	
Outgoing Interface	
Protocol	Any
Source Address	☑ 0 . 0 . 0 . 0 / 32
Source Port(s)	-
Destination Address	☑ 0 . 0 . 0 . 0 / 32
Destination Port(s)	-

DSCP

Set Value	0
Adjust Priority	□

Field	Description	Required	Notes
Active	Is this rule active?	Yes	-
Incoming Interface	The interface where the packet comes in	No	-

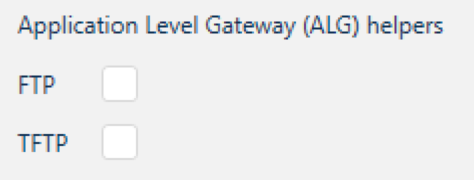
Field	Description	Required	Notes
Outgoing Interface	The interface where the packet is sent out	No	-
Protocol	The protocol type of the IP payload. Typically TCP or UDP but the filtering can also be made to match other protocols such as ICMP and ESP.	Yes	-
Source Address	The source IP address of the packet	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 prioritising and applied to VLAN-priority on tagged packets	No	-

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.



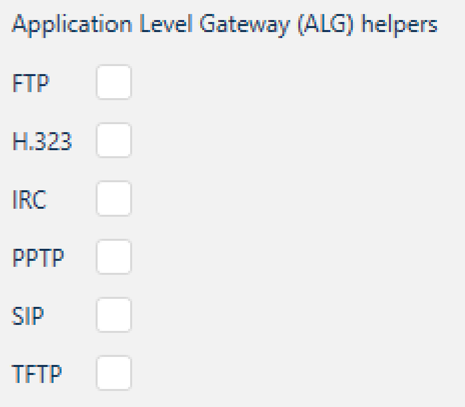
Application Level Gateway (ALG) helpers

FTP □

TFTP □

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

WeOS 4



Application Level Gateway (ALG) helpers

FTP □

H.323 □

IRC □

PPTP □

SIP □

TFTP □

The following ALG Helpers are available for WeOS 4 devices:

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

4.3.3.7. System

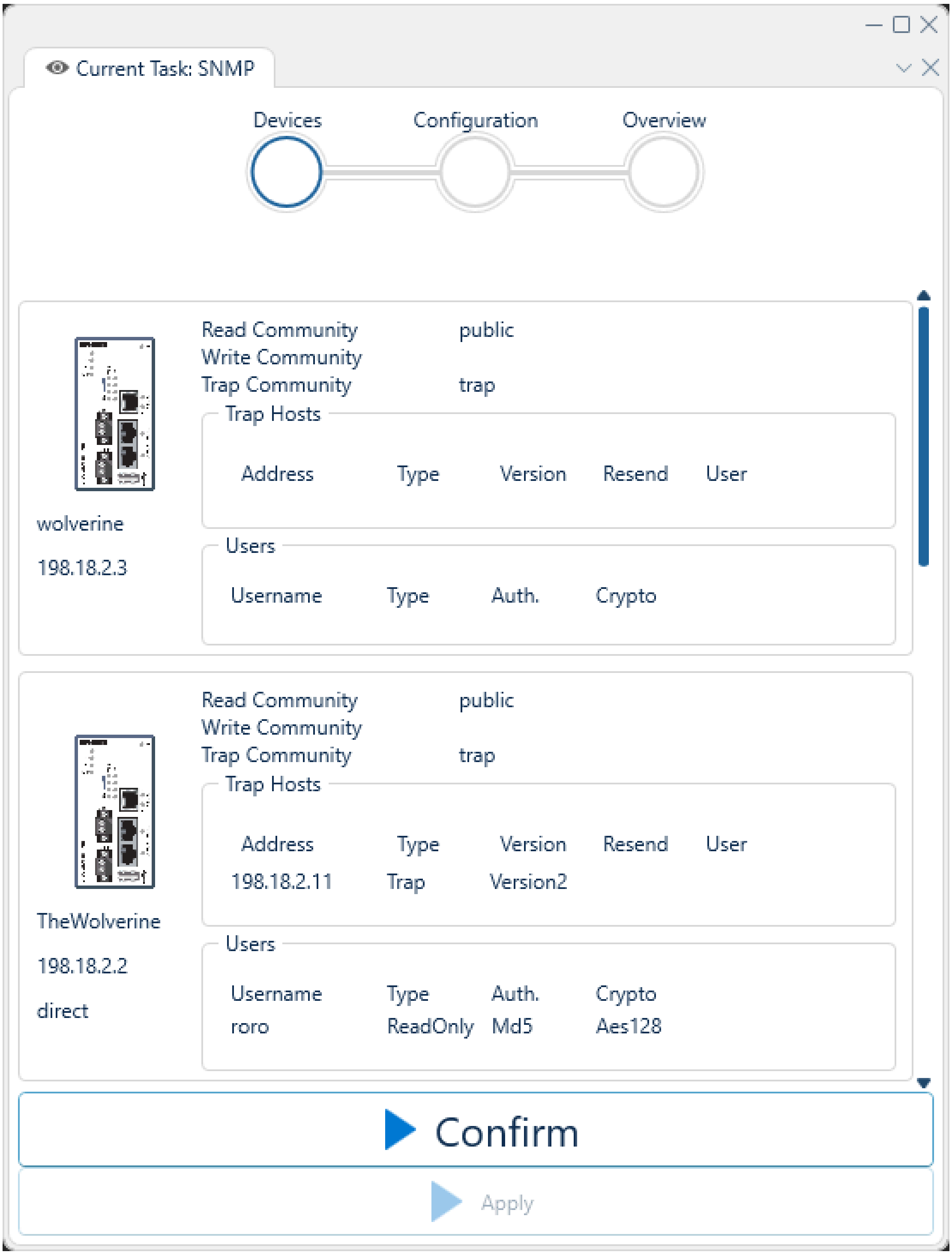
4.3.3.7.1. Simple network Management Protocol (SNMP)

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



wolverine

198.18.2.3

Read Community	public
Write Community	
Trap Community	trap

Trap Hosts

Address	Type	Version	Resend	User

Users

Username	Type	Auth.	Crypto

TheWolverine

198.18.2.2

direct

Read Community	public
Write Community	
Trap Community	trap

Trap Hosts

Address	Type	Version	Resend	User
198.18.2.11	Trap	Version2		

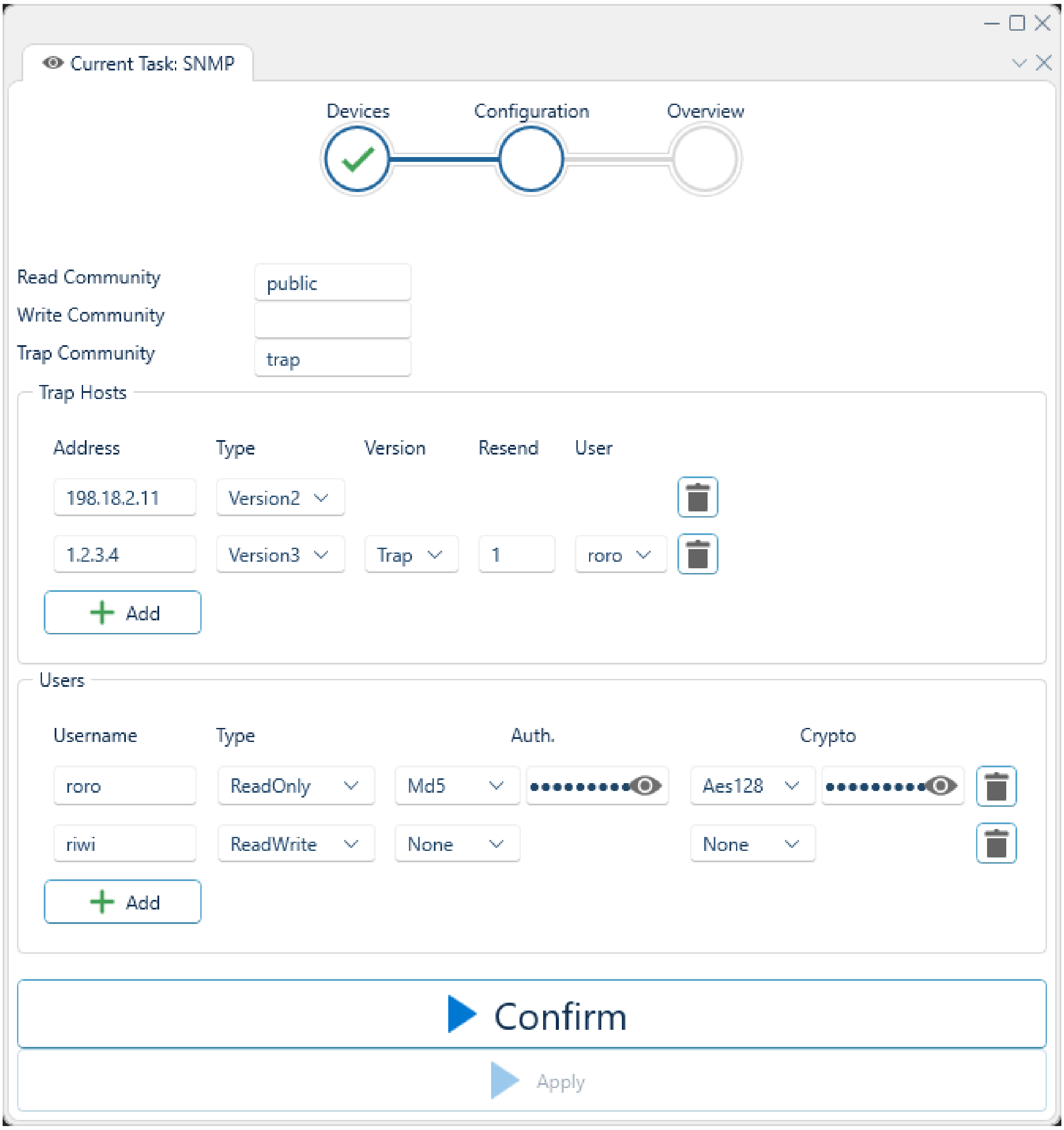
Users

Username	Type	Auth.	Crypto
roro	ReadOnly	Md5	Aes128

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



Read Community: public
Write Community:
Trap Community: trap

Trap Hosts

Address	Type	Version	Resend	User
198.18.2.11	Version2			
1.2.3.4	Version3	Trap	1	roro

Users

Username	Type	Auth.		Crypto	
roro	ReadOnly	Md5	•••••••••	Aes128	•••••••••
riwi	ReadWrite	None		None	

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

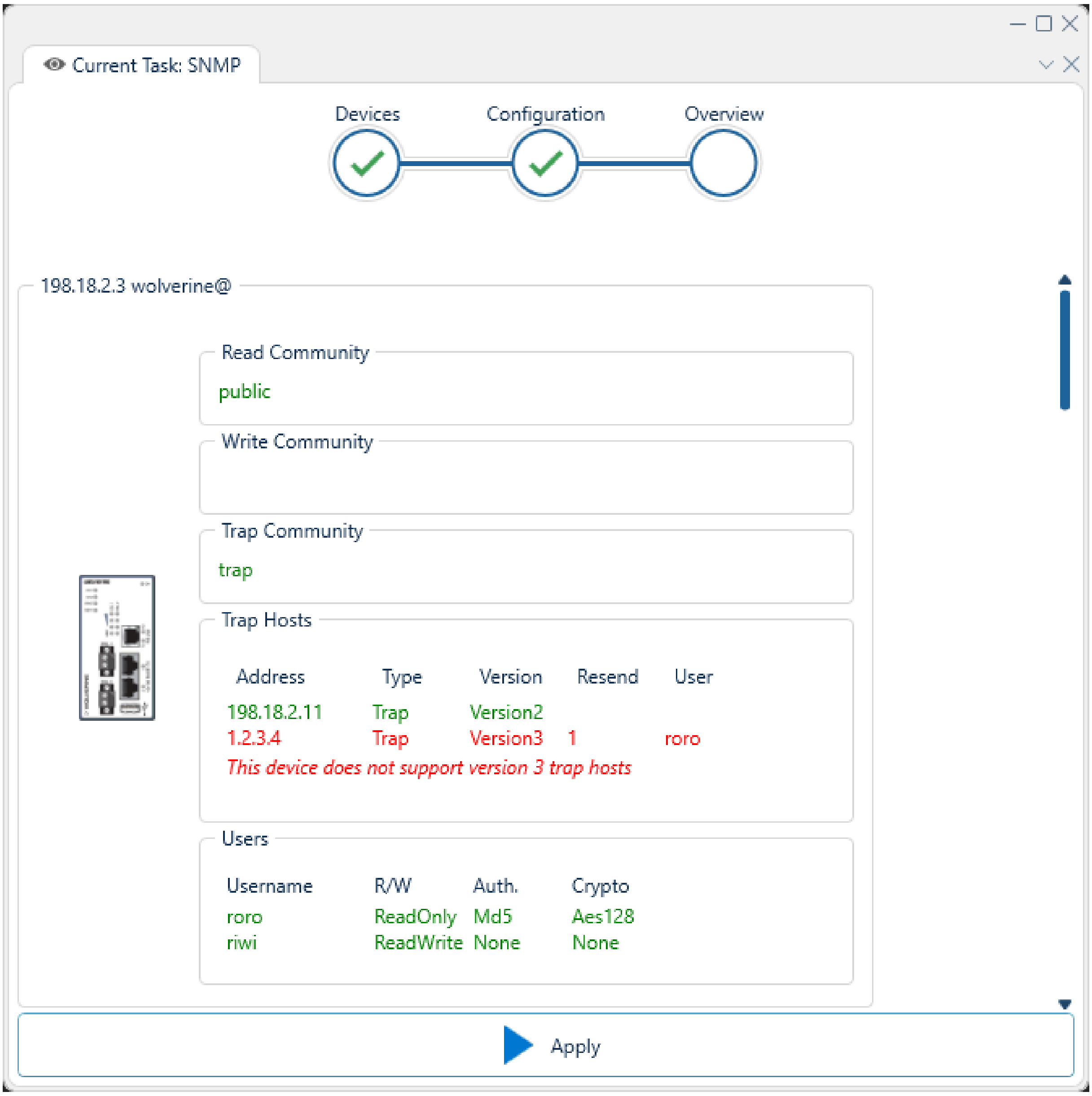
If at least one compatible SNMPv3 user is configured on a device, WeConfig automatically selects one of the new users and updates the Device Access for the project to make use of a specific user.

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:



198.18.2.3 wolverine@

Read Community
public

Write Community

Trap Community
trap

Trap Hosts

Address	Type	Version	Resend	User
198.18.2.11	Trap	Version2		
1.2.3.4	Trap	Version3	1	roro

This device does not support version 3 trap hosts

Users

Username	R/W	Auth.	Crypto
roro	ReadOnly	Md5	Aes128
riwi	ReadWrite	None	None

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

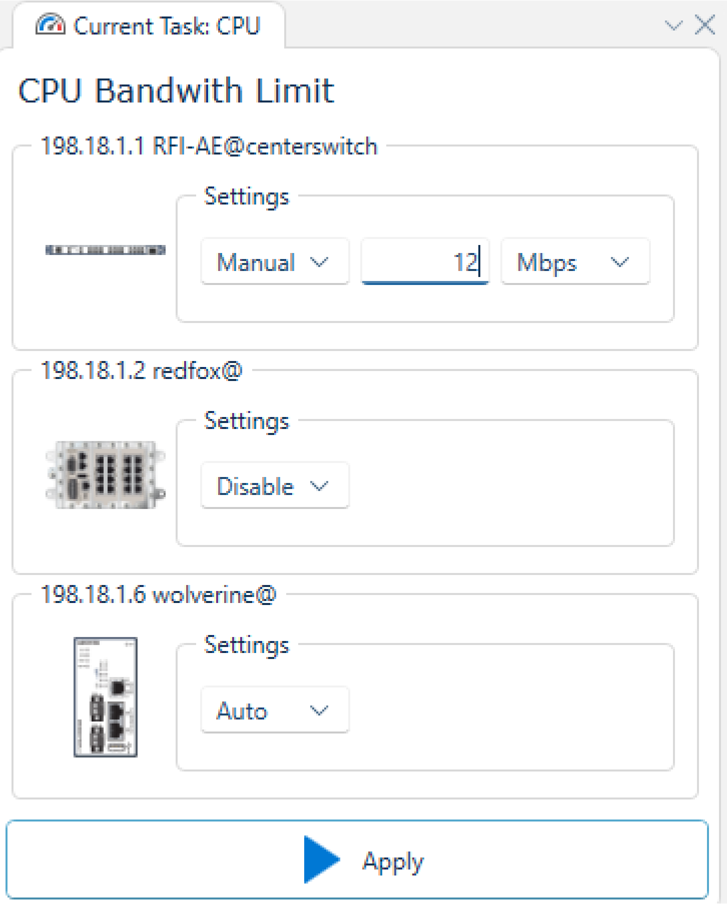
With this in mind, the apply button is still available, and WeConfig will simply not configure anything unsupported on relevant devices.

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

4.3.3.7.2. CPU

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

Interface components



Current Task: CPU

CPU Bandwith Limit

198.18.1.1 RFI-AE@centerswitch

Settings

Manual 12 Mbps

198.18.1.2 redfox@

Settings

Disable

198.18.1.6 wolverine@

Settings

Auto

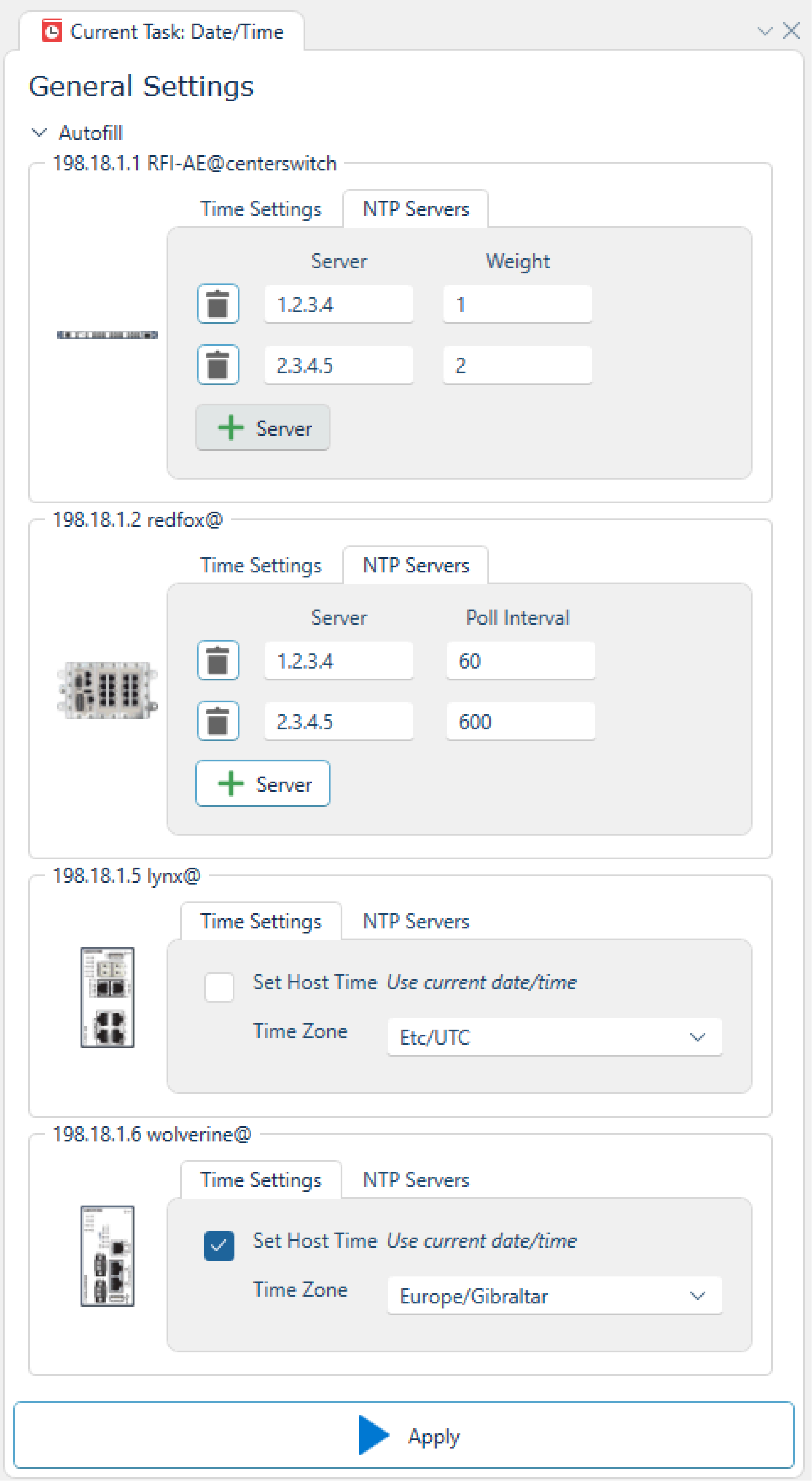
For each added device, you may choose the follow parameters in the combo box:

- 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)

4.3.3.7.3. Date/Time

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



General Settings

Autofill

198.18.1.1 RFI-AE@centerswitch

Server	Weight
1.2.3.4	1
2.3.4.5	2

198.18.1.2 redfox@

Server	Poll Interval
1.2.3.4	60
2.3.4.5	600

198.18.1.5 lynx@

□ Set Host Time *Use current date/time*

Time Zone Etc/UTC

198.18.1.6 wolverine@

☑ Set Host Time *Use current date/time*

Time Zone Europe/Gibraltar

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.

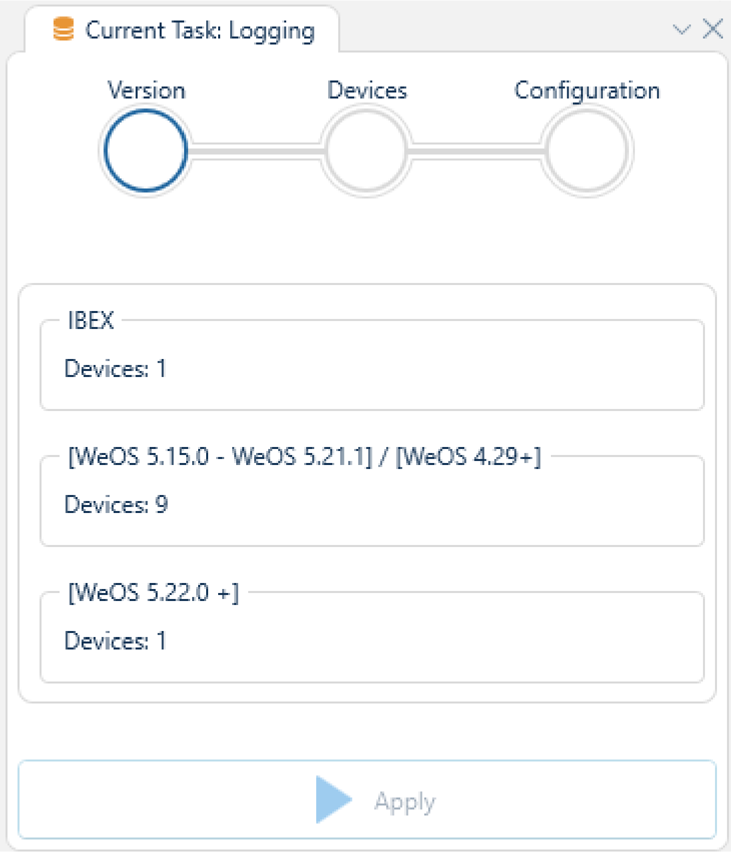
4.3.3.7.4. Logging

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



IBEX

Devices: 1

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

Devices: 9

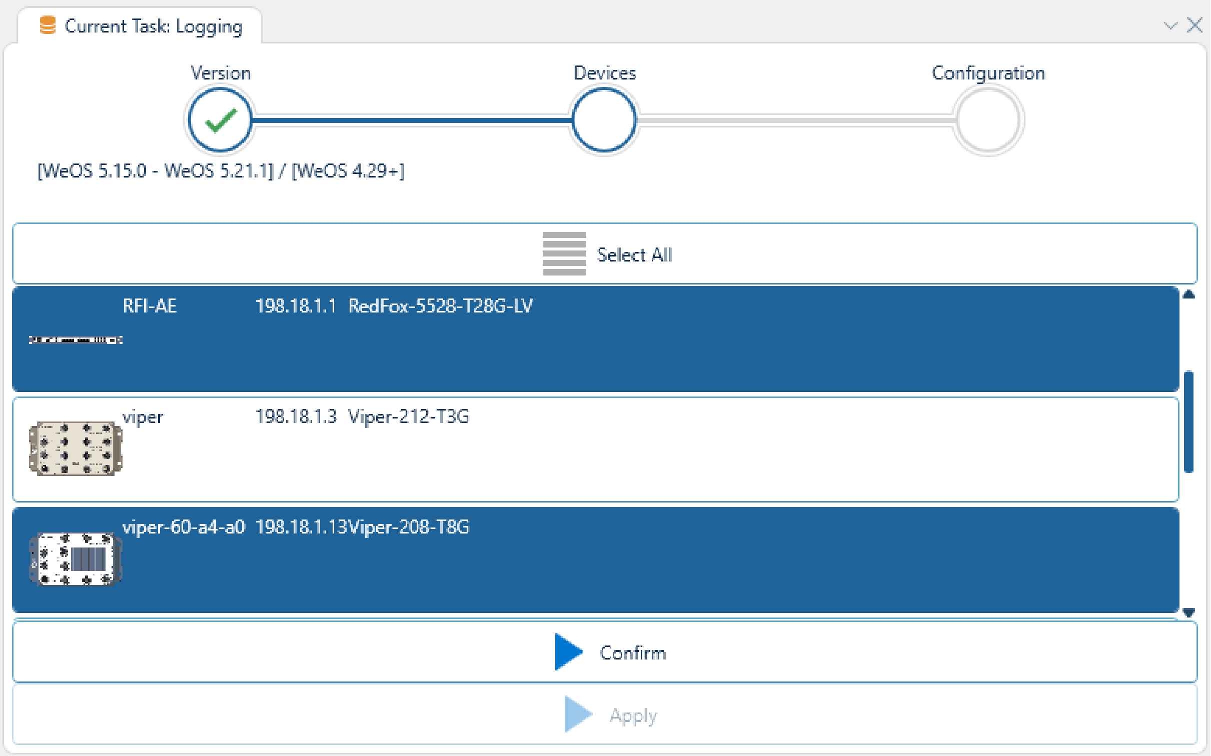
[WeOS 5.22.0 +]

Devices: 1

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



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

RFI-AE 198.18.1.1 RedFox-5528-T28G-LV

viper 198.18.1.3 Viper-212-T3G

viper-60-a4-a0 198.18.1.13Viper-208-T8G

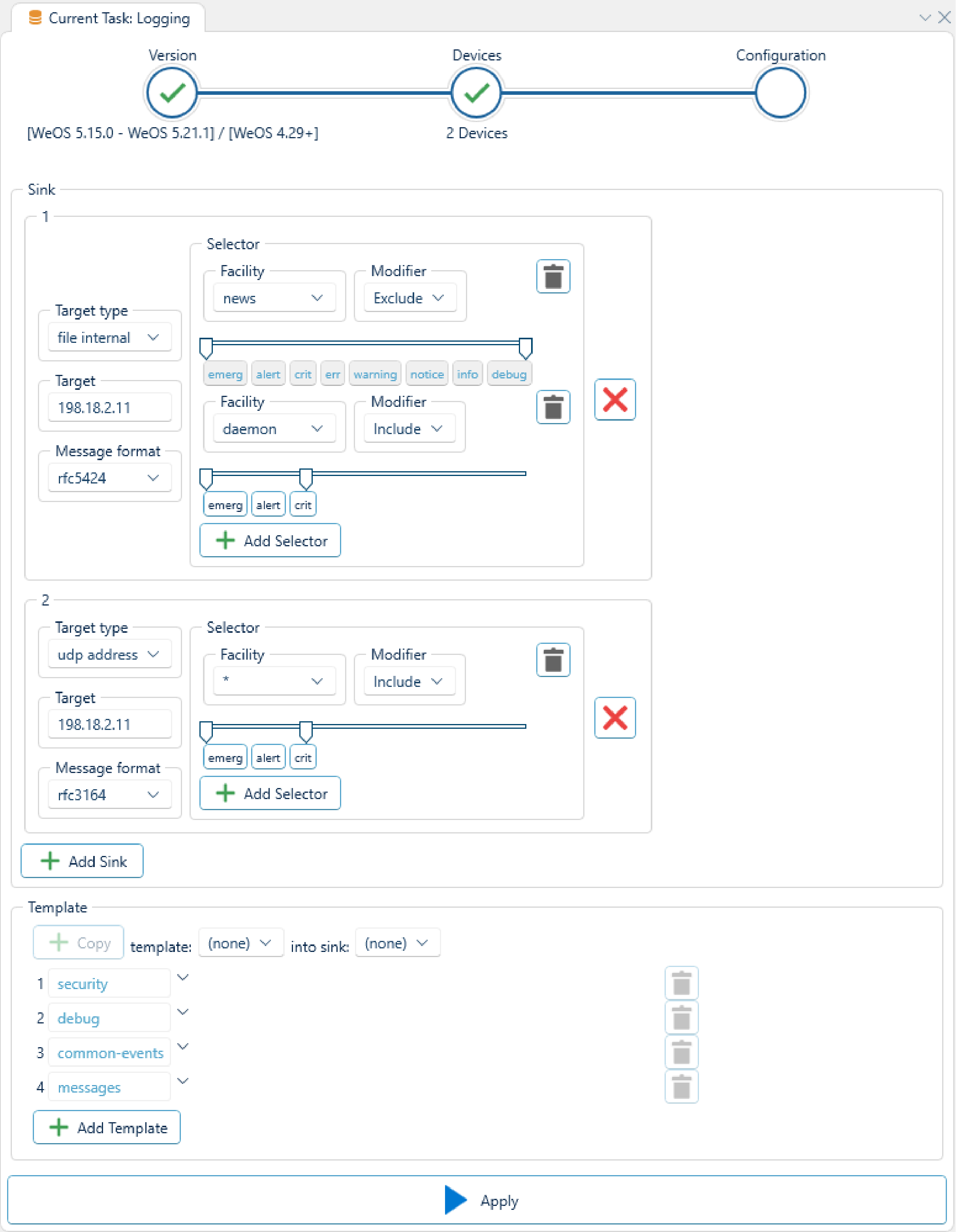
The second stage is called Devices, and presents a list of devices compatible with the selected version of syslog configuration. Any devices already selected in the topology will be preselected in this list, but the selection may be changed before procceeding.

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+]



Current Task: Logging

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

Devices: 2 Devices

Configuration

Sink

1

Target type: file internal

Target: 198.18.2.11

Message format: rfc5424

Selector

Facility: news

Modifier: Exclude

emerg alert crit err warning notice info debug

Facility: daemon

Modifier: Include

emerg alert crit

2

Target type: udp address

Target: 198.18.2.11

Message format: rfc3164

Selector

Facility: *

Modifier: Include

emerg alert crit

Template

template: (none) into sink: (none)

1 security

2 debug

3 common-events

4 messages

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[1] .
Target	Additional parameter depending on Target type[2] .
Message format	The syslog format, either rfc3164[3] , rfc5425[4] or bsd.
Selector	Which produced syslog messages are sent to this sink, see Selectors[5]

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.

WeOS 5.15 to WeOS 5.21.1 provides the following facilities:

[1] #iface-logging-v2-targets
[2] #iface-logging-v2-targets
[3] https://datatracker.ietf.org/doc/html/rfc3164
[4] https://datatracker.ietf.org/doc/html/rfc5424
[5] #iface-logging-v2-selectors

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

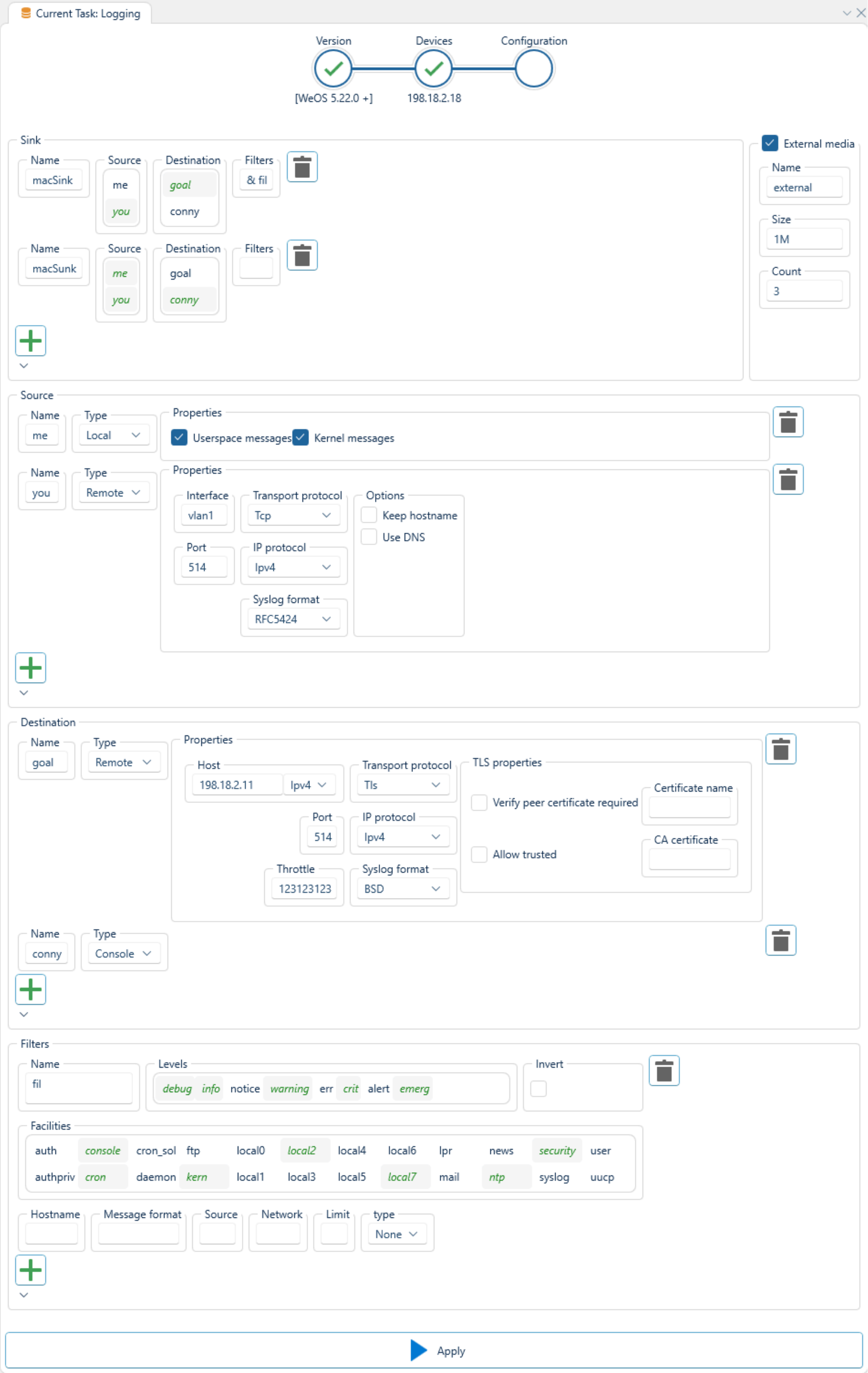
Lastly, a selector consists of a severity selector, which may be used to narrow the range of messages to only those of a certain severity by dragging the slider from the right or respectively.

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:



[WeOS 5.22.0 +]

198.18.2.18

Sink

Name: macSink | Source: me, *you* | Destination: *goal*, conny | Filters: & fil

Name: macSunk | Source: *me*, *you* | Destination: goal, *conny* | Filters:

☑ External media

Name: external

Size: 1M

Count: 3

Source

Name: me | Type: Local | Properties: ☑ Userspace messages ☑ Kernel messages

Name: you | Type: Remote | Properties:

- Interface: vlan1
- Transport protocol: Tcp
- Port: 514
- IP protocol: Ipv4
- Syslog format: RFC5424
- Options: □ Keep hostname □ Use DNS

Destination

Name: goal | Type: Remote | Properties:

- Host: 198.18.2.11 Ipv4
- Transport protocol: Tls
- Port: 514
- IP protocol: Ipv4
- Throttle: 123123123
- Syslog format: BSD
- TLS properties: □ Verify peer certificate required □ Allow trusted
- Certificate name:
- CA certificate:

Name: conny | Type: Console

Filters

Name: fil

Levels: *debug* *info* notice *warning* err *crit* alert *emerg*

Invert: □

Facilities:

auth *console* cron_sol ftp local0 *local2* local4 local6 lpr news *security* user

authpriv *cron* daemon *kern* local1 local3 local5 *local7* mail *ntp* syslog uucp

Hostname: | Message format: | Source: | Network: | Limit: | type: None

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[1] 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[2] 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[3].

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

[1] #iface-logging-v2-selectors
[2] #iface-logging-v2-targets
[3] #iface-logging-v2

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.

External media

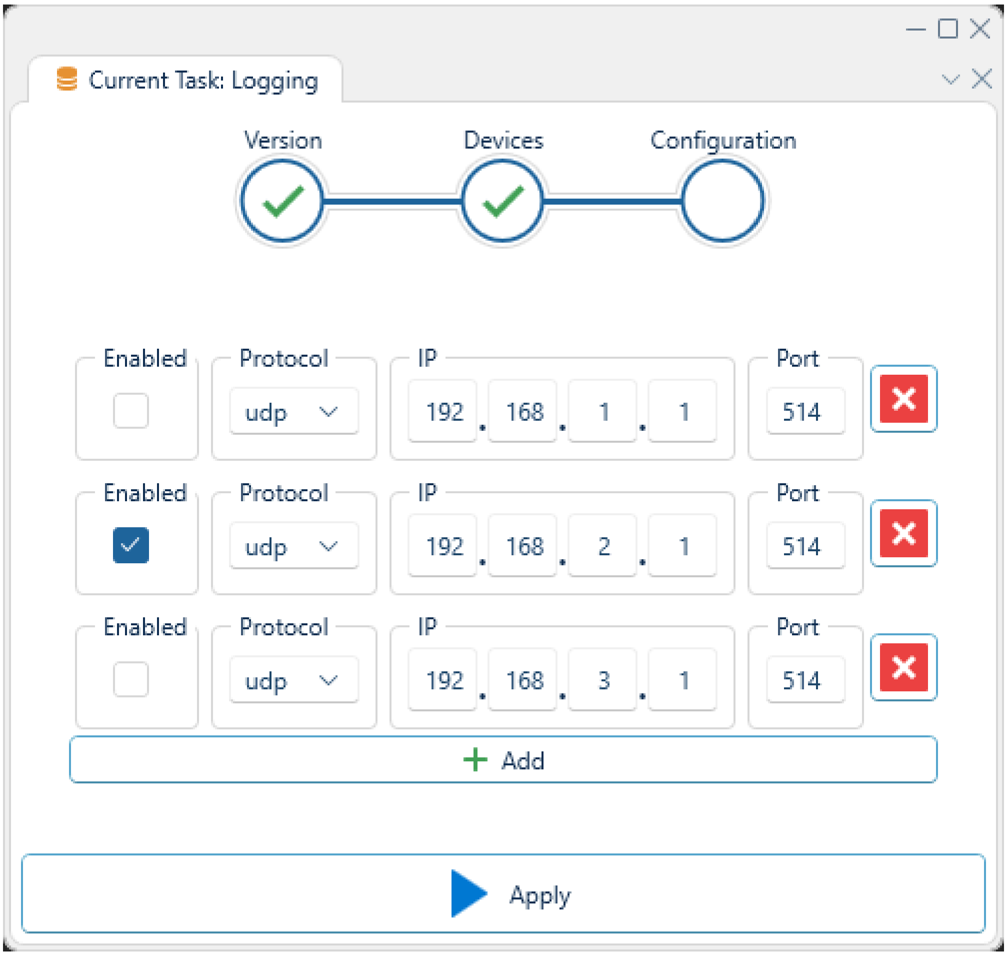
To the right at the top you will find a small section dedicated to external media configuration, where you may configured how the log file is handled on the external media, filesize and count may be configured here.

Sinks

Lastly, you will find Sinks at the very top 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 "recieve syslog messages from the `local` and `remote` source, but not the `other` source, and if they match the filter specified by `hideNonEssential`, send them to the `aggregator` destination". Of special note is perhaps the filters box, where applicable filters are combiend together using either `&`, for and, or `|`, for or.

> Note
>
> Leaving the filter box empty is the equivalent to "any"

Ibex



Enabled	Protocol	IP	Port
□	udp	192.168.1.1	514
☑	udp	192.168.2.1	514
□	udp	192.168.3.1	514

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.

4.3.3.8. Routing

4.3.3.8.1. OSPF

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.

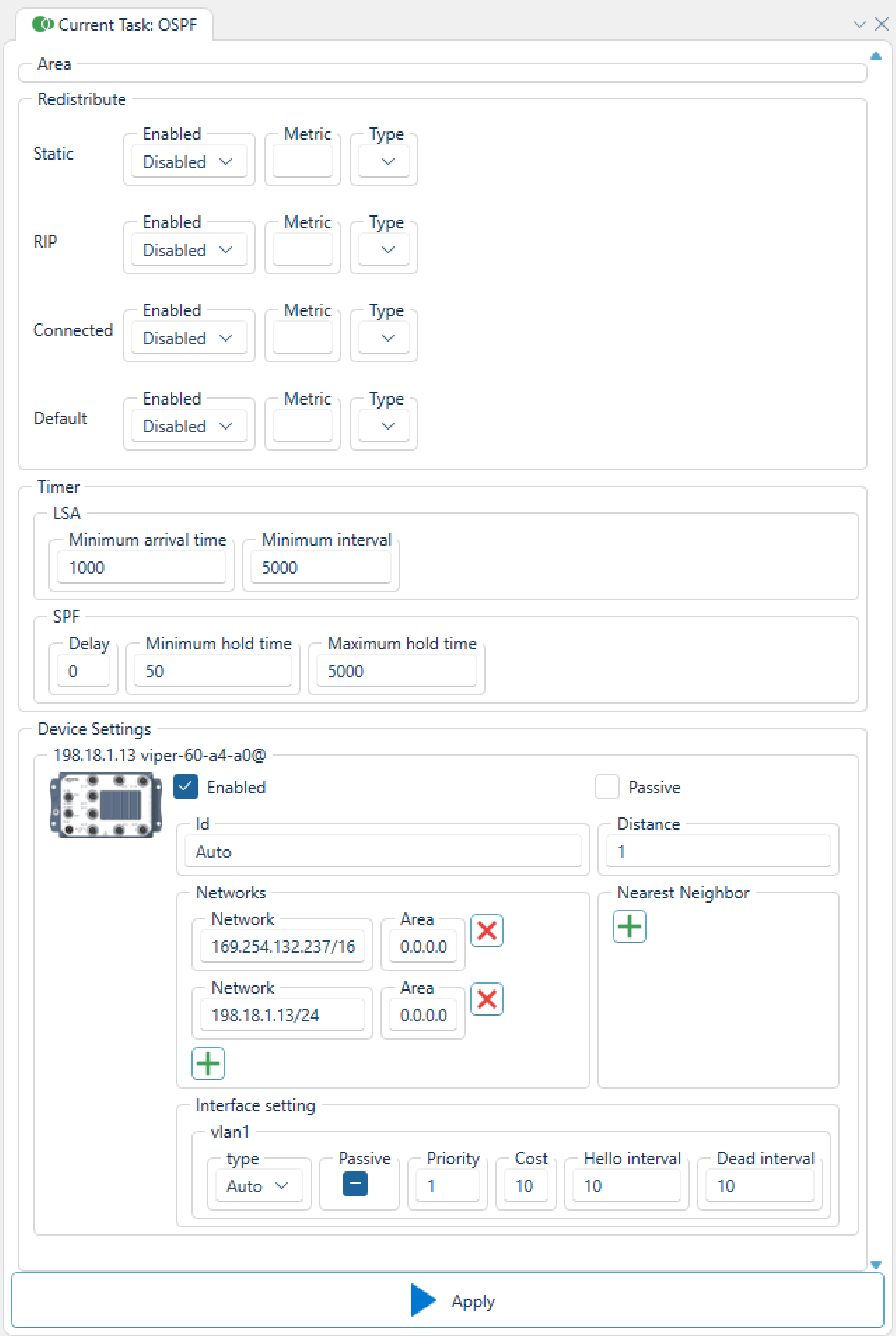
Create OSPF Area

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:

- Regular
- Stubby
- Totally Stubby
- Not so Stubby
- Not so totally stubby

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

Interface Components:



Current Task: OSPF

Area

Redistribute

	Enabled	Metric	Type
Static	Disabled		
RIP	Disabled		
Connected	Disabled		
Default	Disabled		

Timer

LSA

Minimum arrival time: 1000

Minimum interval: 5000

SPF

Delay: 0

Minimum hold time: 50

Maximum hold time: 5000

Device Settings

198.18.1.13 viper-60-a4-a0@

☑ Enabled

□ Passive

Id: Auto

Distance: 1

Networks

Network	Area
169.254.132.237/16	0.0.0.0
198.18.1.13/24	0.0.0.0

Nearest Neighbor

Interface setting

vlan1

type	Passive	Priority	Cost	Hello interval	Dead interval
Auto	−	1	10	10	10

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[1] 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[2].

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.

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

[1] #ospf-context-menu-create

[2] #ospf-context-menu-redistribute

⚠ 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.

4.3.3.8.2. RIP

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

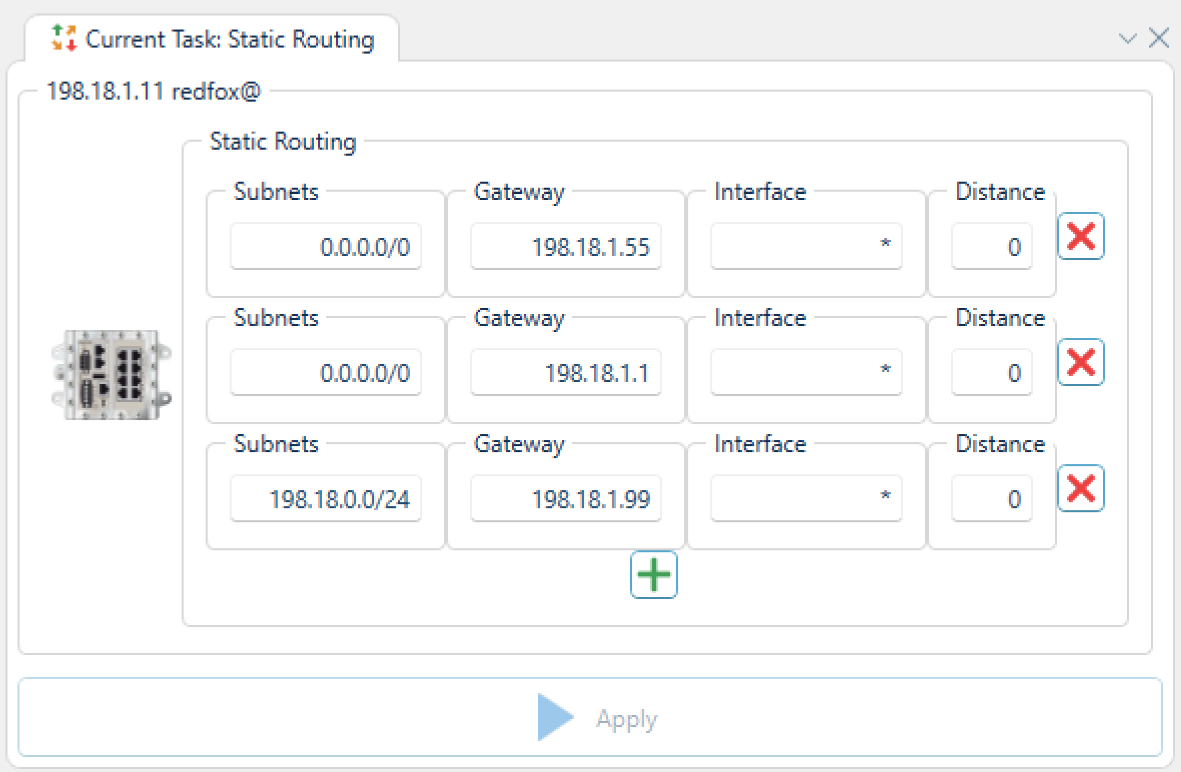
Remove

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.

4.3.3.8.3. Static routing

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



Current Task: Static Routing

198.18.1.11 redfox@

Static Routing

Subnets	Gateway	Interface	Distance
0.0.0.0/0	198.18.1.55	*	0
0.0.0.0/0	198.18.1.1	*	0
198.18.0.0/24	198.18.1.99	*	0

The user interfac 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	The network interface to send traffic matching the route on, `*` is used to mean any, meaning the gateway is the only targeting parameter, otherwise an interface such as `vlan1` may be specified
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

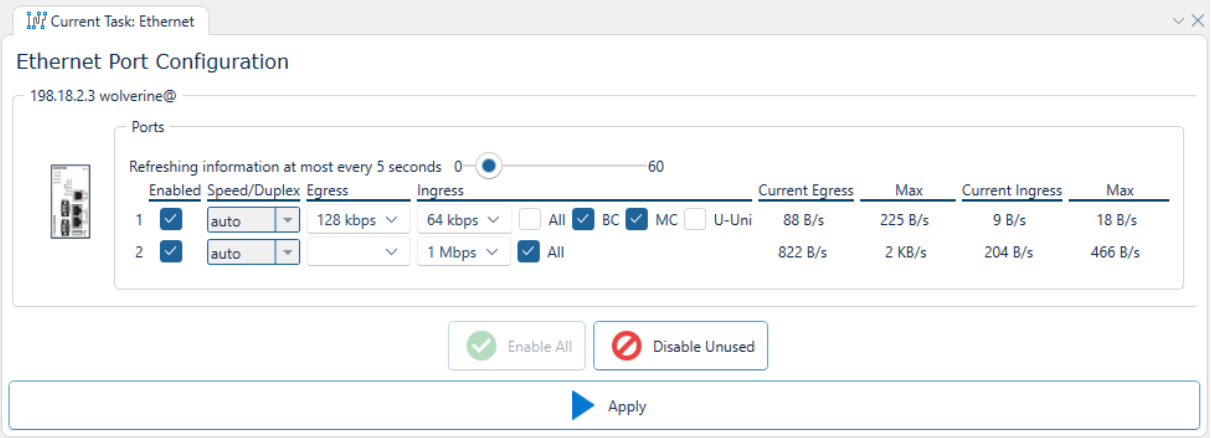
When, in the topology view, two or more subnets are selected that each contain at least one device supporting static route configuration, this context menu will appear. Clicking this context menu will attempt to set up all the static routes that traverse from the and to each pair of subnets in the selected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

4.3.3.9. Ports

4.3.3.9.1. Ethernet

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



Ethernet Port Configuration

198.18.2.3 wolverine@

Ports

Refreshing information at most every 5 seconds 0 60

	Enabled	Speed/Duplex	Egress	Ingress	Current Egress	Max	Current Ingress	Max
1	☑	auto	128 kbps	64 kbps □ All ☑ BC ☑ MC □ U-Uni	88 B/s	225 B/s	9 B/s	18 B/s
2	☑	auto		1 Mbps ☑ All	822 B/s	2 KB/s	204 B/s	466 B/s

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 buttom 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:

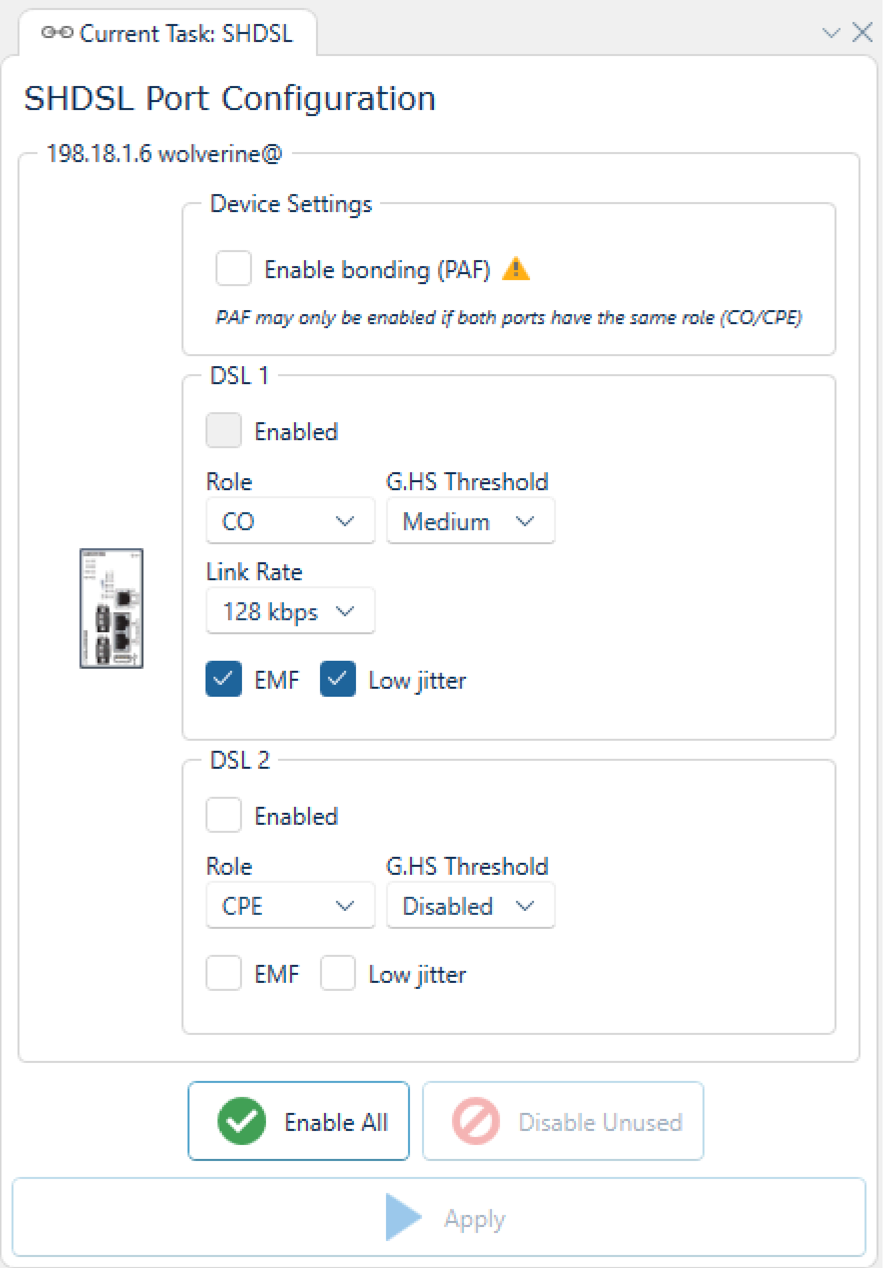
Checkbox	Description
All	All Traffic
BC	MAC Broadcast `FF:FF:FF:FF:FF:FF`
MC	MAC Multicast `(2n+1):**:**:**:**:**`
U-Uni	MAC unicast `(2n):**:**:**:**:**`

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.

4.3.3.9.2. SHDSL

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



Current Task: SHDSL

SHDSL Port Configuration

198.18.1.6 wolverine@

Device Settings

□ Enable bonding (PAF) ⚠

PAF may only be enabled if both ports have the same role (CO/CPE)

DSL 1

□ Enabled

Role: CO

G.HS Threshold: Medium

Link Rate: 128 kbps

☑ EMF ☑ Low jitter

DSL 2

□ Enabled

Role: CPE

G.HS Threshold: Disabled

□ EMF □ Low jitter

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.

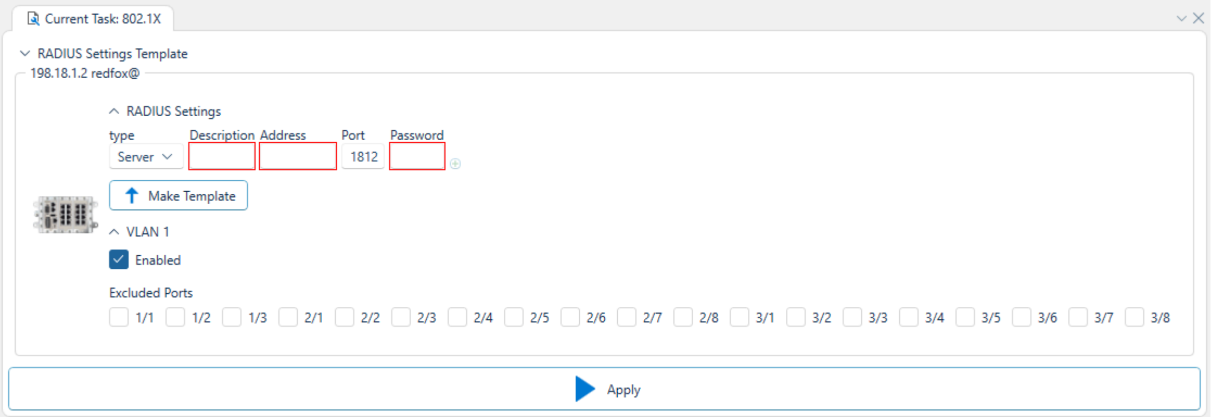
WeConfig will also remind you to click Propose Order before the use of new configurations.

Propose Order will order the devices in such a way that device configurations are applied in an order such that WeConfig is not locked out by unstable intermediate links. This function will only work if WeConfig has established its connection to the topology.

4.3.3.10. 802.1X

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

Interface components



Current Task: 802.1X

RADIUS Settings Template

198.18.1.2 redfox@

RADIUS Settings

type: Server

Description:

Address:

Port: 1812

Password:

VLAN 1

☑ Enabled

Excluded Ports

□ 1/1 □ 1/2 □ 1/3 □ 2/1 □ 2/2 □ 2/3 □ 2/4 □ 2/5 □ 2/6 □ 2/7 □ 2/8 □ 3/1 □ 3/2 □ 3/3 □ 3/4 □ 3/5 □ 3/6 □ 3/7 □ 3/8

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.

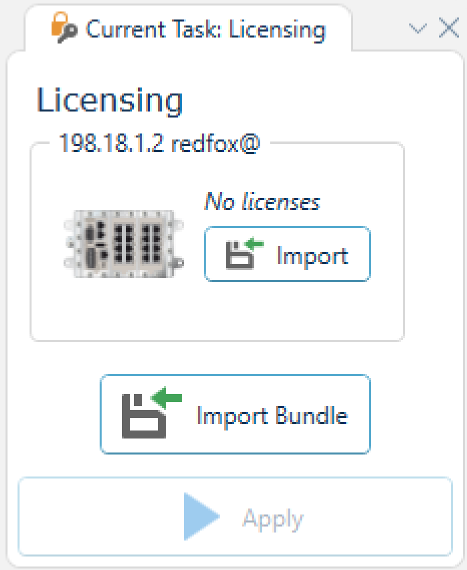
To propagate the RADIUS settings from one device onto all devices, select the “master device’s” RADIUS settings and click the Make template button. Now the template area has the same settings as the “master device”. Then, click Fill to propagate to all devices. To add a RADIUS server, select Server in 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.

4.3.3.11. Licensing

Licesning is a task that allows for importing or removing 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.

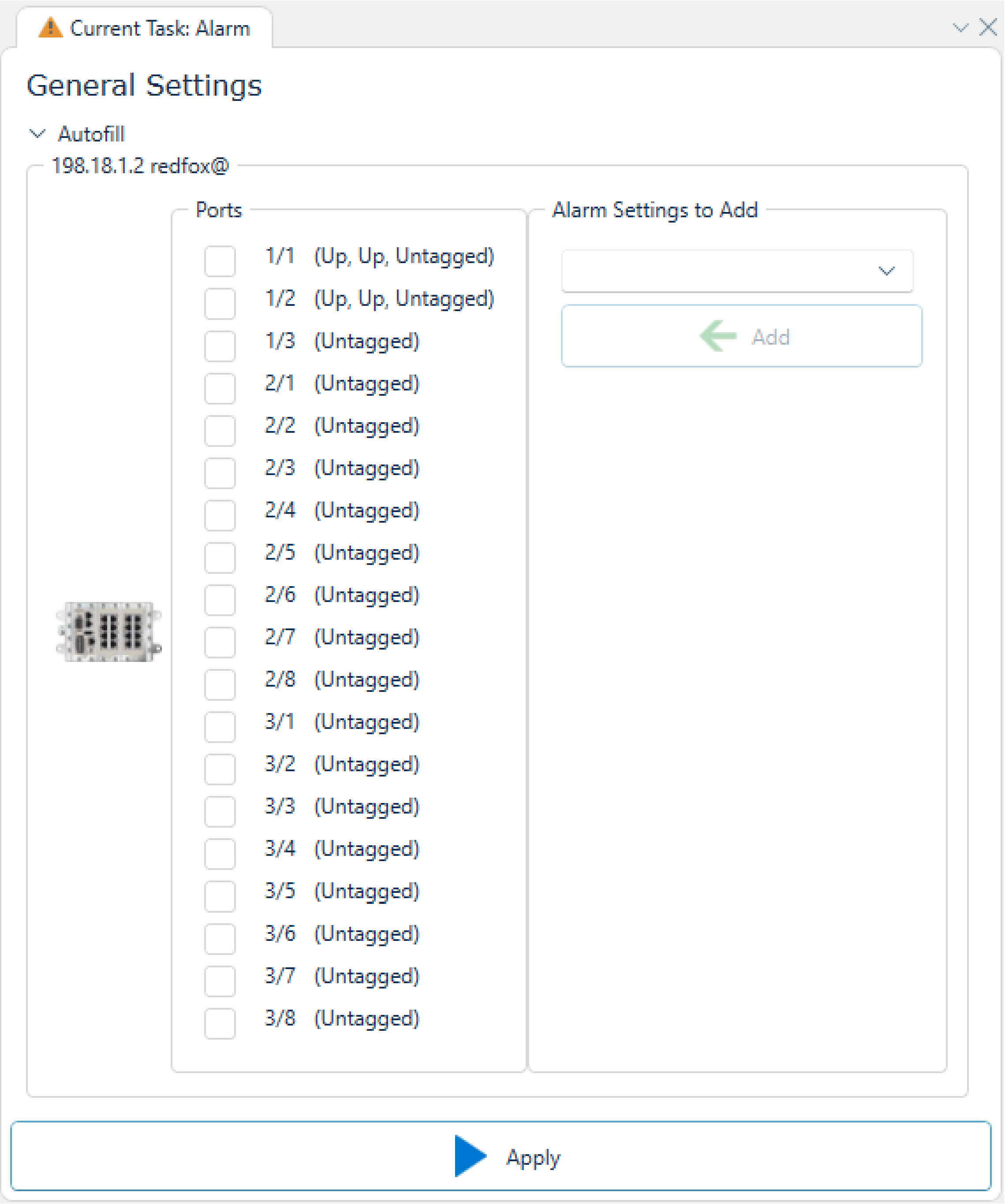
Note

Only WeOS 4.23 or above is supported.

4.3.3.12. Alarm

Alarm is a task that allows for configuration of link up/down alarms on device ports.

Interface components



Current Task: Alarm

General Settings

Autofill

198.18.1.2 redfox@

Ports

☐ 1/1 (Up, Up, Untagged)
☐ 1/2 (Up, Up, Untagged)
☐ 1/3 (Untagged)
☐ 2/1 (Untagged)
☐ 2/2 (Untagged)
☐ 2/3 (Untagged)
☐ 2/4 (Untagged)
☐ 2/5 (Untagged)
☐ 2/6 (Untagged)
☐ 2/7 (Untagged)
☐ 2/8 (Untagged)
☐ 3/1 (Untagged)
☐ 3/2 (Untagged)
☐ 3/3 (Untagged)
☐ 3/4 (Untagged)
☐ 3/5 (Untagged)
☐ 3/6 (Untagged)
☐ 3/7 (Untagged)
☐ 3/8 (Untagged)

Alarm Settings to Add

Depicted above is the user interface of this task, where we can see a list of devices, in this case a singular one, each containing a list of ports and a template selector called “Alarm Settings to Add”. Above the device list exists an autofill section that is handy for rapidly configuring many devices.

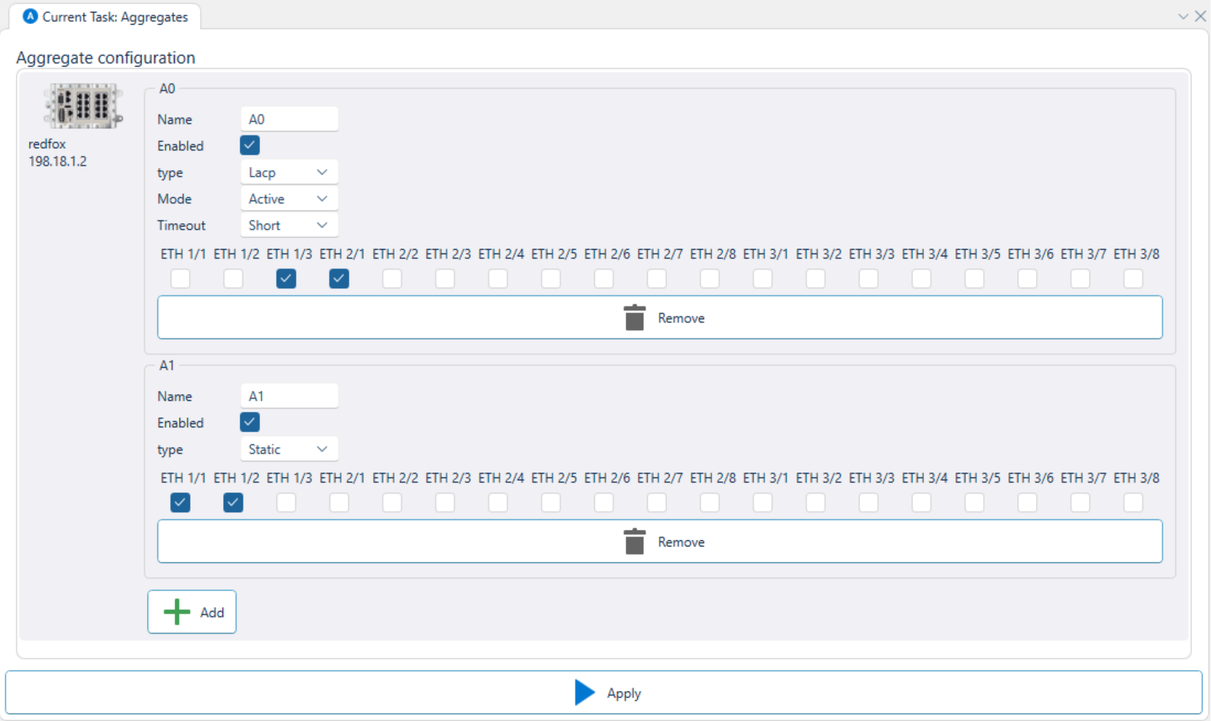
Each port in each devices port list indicates it’s current link/enabled status, as well as if it’s considered tagged or untagged on any VLAN. Each port can either be configured manually with a port alarm, or a preset selection amongst the following:

Preset	Description
Enable on all tagged ports	Set alarm as enabled on all ports that are tagged in at least one VLAN.
Enable on all untagged ports	Set alarm as enabled on all ports that are untagged in at least one VLAN.
Enable on all FRNT ports	Set alarm as enabled on all ports that are configured to be part of at least one FRNT ring.
Enabled on all MRP ports	Set alarm as enabled on all ports that are configured to be part of at least one MRP ring.
Enabled on all RSTP ports (Non Admin Edge)	Set alarm as enabled on all ports that are configured for RSTP meshing, except those configured as an admin edge.
Enabled on All ports currently having link status up	All ports that are currently up will have alarms enabled, except if that port connects directly to the configuring PC.
Disable on all Ports	No port will have alarm enabled

4.3.3.13. Aggregates

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

Interface components



Current Task: Aggregates

Aggregate configuration

redfox
198.18.1.2

A0

Name A0

Enabled ☑

type Lacp

Mode Active

Timeout Short

ETH 1/1	ETH 1/2	ETH 1/3	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
□	□	☑	☑	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□

A1

Name A1

Enabled ☑

type Static

ETH 1/1	ETH 1/2	ETH 1/3	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
☑	☑	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□

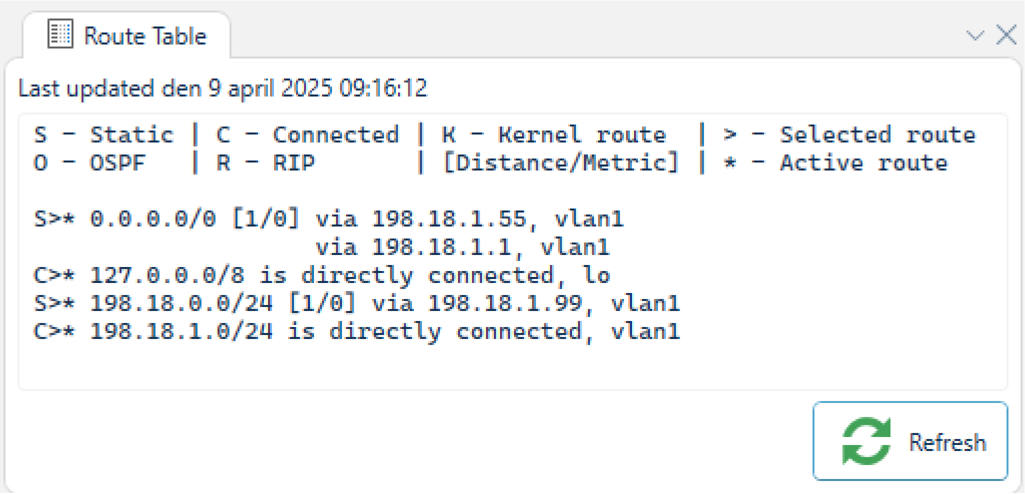
As depicted above, the aggregate configuration interface consists of a list of devices, in this example a singular one, with a list of zero-or-more configured aggregates within. Each aggregate can be configured with a name, whether or not it's enabled, which type of aggregate it is and the ports that 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 desird can be removed with the "Remove" button, and new aggregates can be created with the "Add" button beneath the list of each device.

4.3.4. Per-Device Views

4.3.4.1. Route Table



Last updated den 9 april 2025 09:16:12

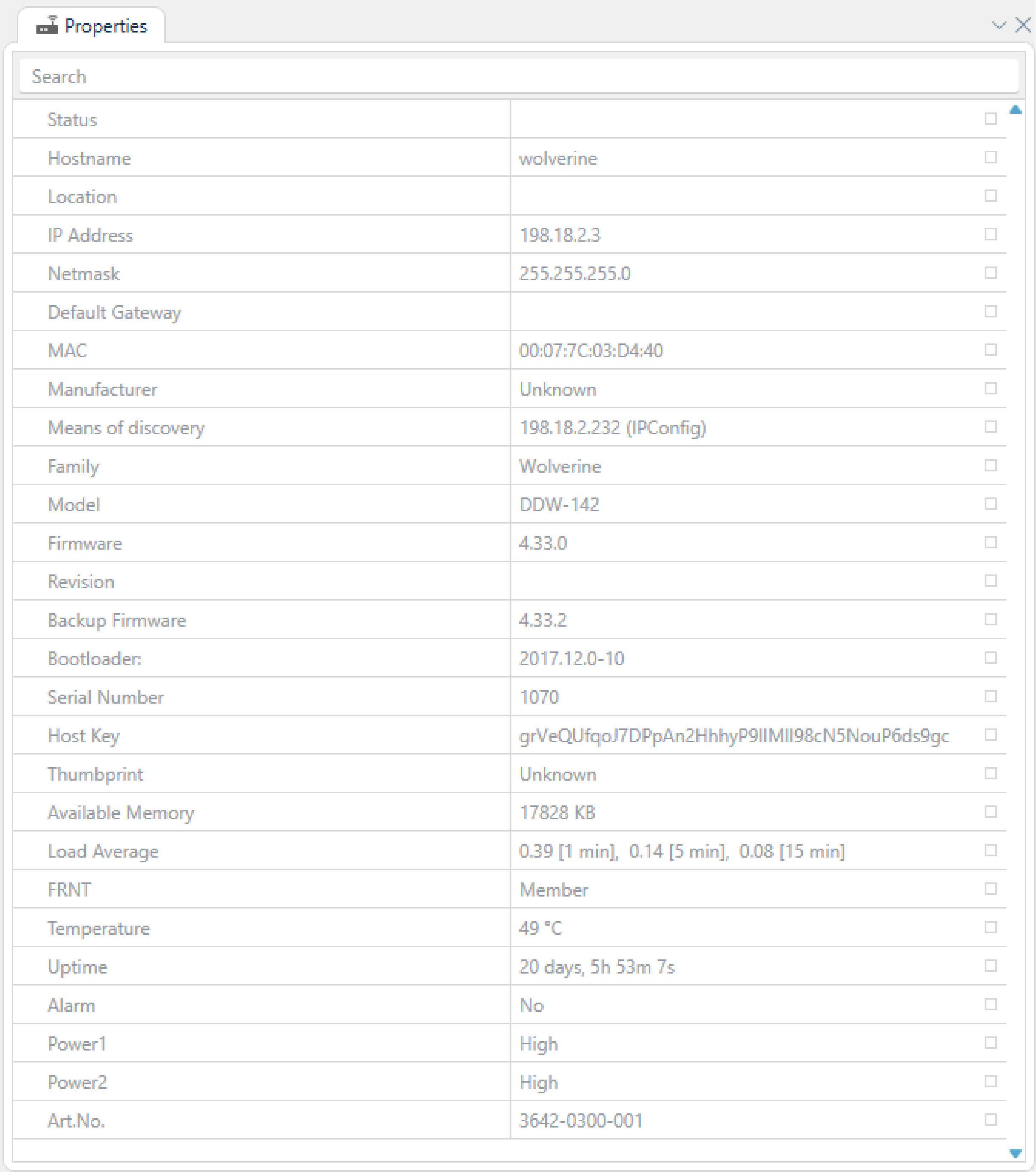
```
S - Static | C - Connected | K - Kernel route  | > - Selected route
O - OSPF   | R - RIP       | [Distance/Metric] | * - Active route

S>* 0.0.0.0/0 [1/0] via 198.18.1.55, vlan1
                     via 198.18.1.1, vlan1
C>* 127.0.0.0/8 is directly connected, lo
S>* 198.18.0.0/24 [1/0] via 198.18.1.99, vlan1
C>* 198.18.1.0/24 is directly connected, vlan1
```

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.

4.3.4.2. Properties



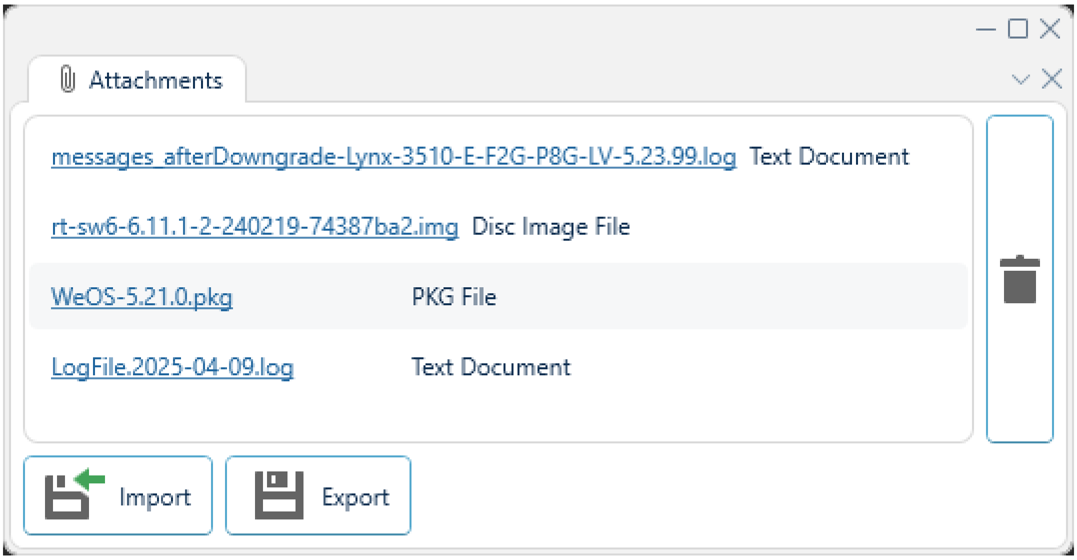
Status		□
Hostname	wolverine	□
Location		□
IP Address	198.18.2.3	□
Netmask	255.255.255.0	□
Default Gateway		□
MAC	00:07:7C:03:D4:40	□
Manufacturer	Unknown	□
Means of discovery	198.18.2.232 (IPConfig)	□
Family	Wolverine	□
Model	DDW-142	□
Firmware	4.33.0	□
Revision		□
Backup Firmware	4.33.2	□
Bootloader:	2017.12.0-10	□
Serial Number	1070	□
Host Key	grVeQUfqoJ7DPpAn2HhhyP9lIMlI98cN5NouP6ds9gc	□
Thumbprint	Unknown	□
Available Memory	17828 KB	□
Load Average	0.39 [1 min], 0.14 [5 min], 0.08 [15 min]	□
FRNT	Member	□
Temperature	49 °C	□
Uptime	20 days, 5h 53m 7s	□
Alarm	No	□
Power1	High	□
Power2	High	□
Art.No.	3642-0300-001	□

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

Property	Description
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 protocol, i.e `Ping` , or an address plus protocol, i.e as in the picture `169.254.145.82 [IPConfig]` which, in the latter 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

4.3.4.3. Attachments



messages_afterDowngrade-Lynx-3510-E-F2G-P8G-LV-5.23.99.log	Text Document
rt-sw6-6.11.1-2-240219-74387ba2.img	Disc Image File
WeOS-5.21.0.pkg	PKG File
LogFile.2025-04-09.log	Text Document

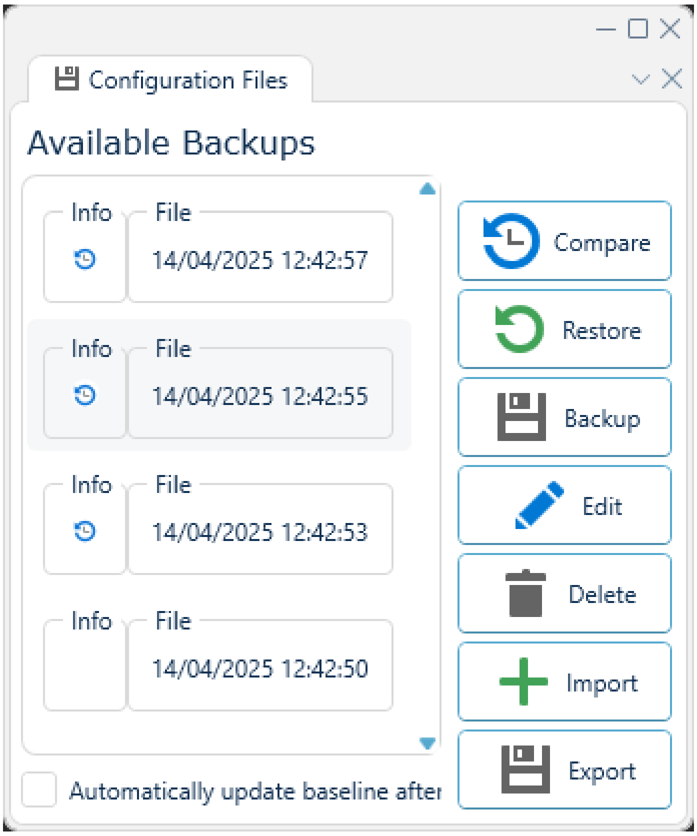
This panel displays the file attachments associated with the currently last selected device as a list of links and format explanations.

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.

4.3.4.4. Configuration Files

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



Available Backups

Info	File
⟲	14/04/2025 12:42:57
⟲	14/04/2025 12:42:55
⟲	14/04/2025 12:42:53
	14/04/2025 12:42:50

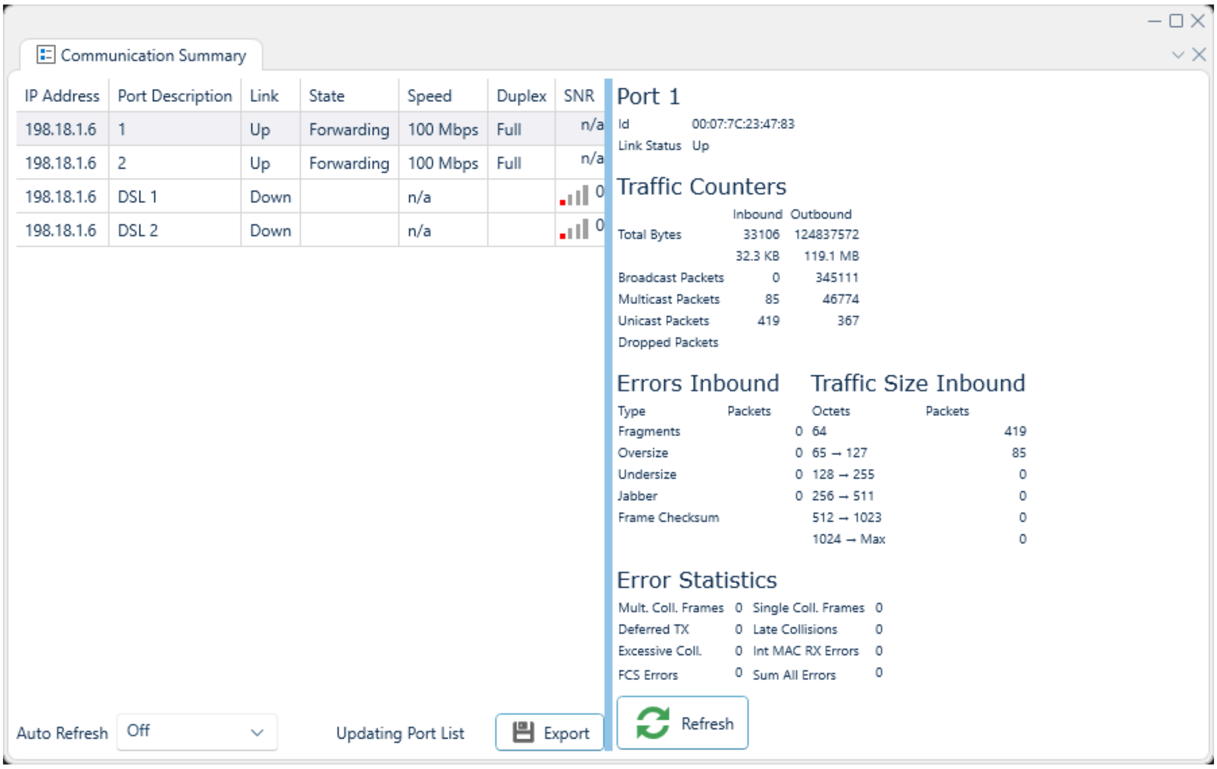
□ Automatically update baseline after

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 “i” 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.

4.3.4.5. Communication Summary



IP Address	Port Description	Link	State	Speed	Duplex	SNR
198.18.1.6	1	Up	Forwarding	100 Mbps	Full	n/a
198.18.1.6	2	Up	Forwarding	100 Mbps	Full	n/a
198.18.1.6	DSL 1	Down		n/a		0
198.18.1.6	DSL 2	Down		n/a		0

Auto Refresh: Off

Updating Port List

Port 1

Id 00:07:7C:23:47:83

Link Status Up

Traffic Counters

	Inbound	Outbound
Total Bytes	33106	124837572
	32.3 KB	119.1 MB
Broadcast Packets	0	345111
Multicast Packets	85	46774
Unicast Packets	419	367
Dropped Packets		

Errors Inbound

Type	Packets
Fragments	0
Oversize	0
Undersize	0
Jabber	0
Frame Checksum	

Traffic Size Inbound

Octets	Packets
64	419
65 → 127	85
128 → 255	0
256 → 511	0
512 → 1023	0
1024 → Max	0

Error Statistics

Mult. Coll. Frames	0	Single Coll. Frames	0
Deferred TX	0	Late Collisions	0
Excessive Coll.	0	Int MAC RX Errors	0
FCS Errors	0	Sum All Errors	0

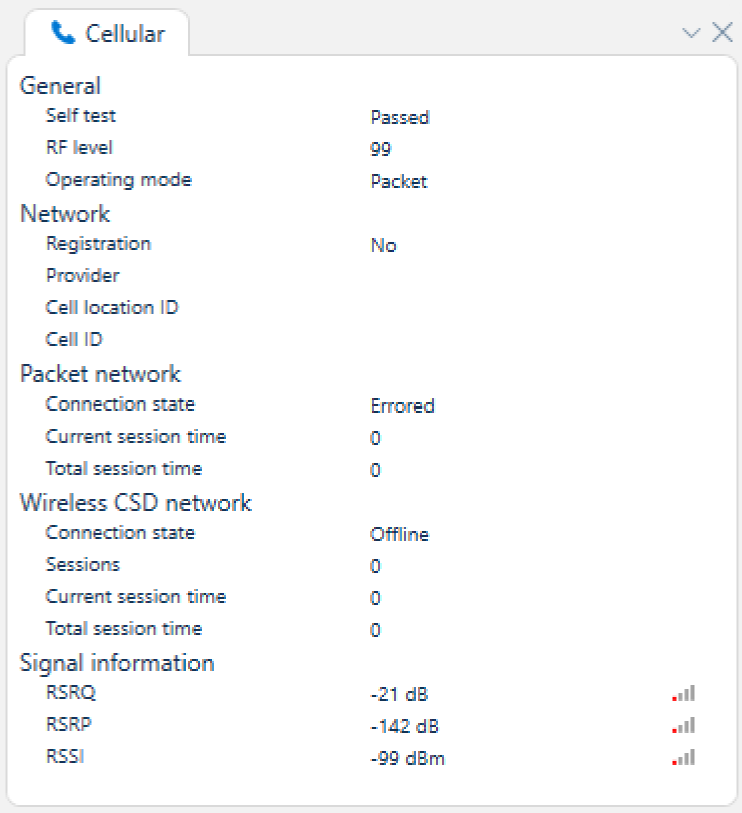
This panel lists a summary of communication information for ports on selected device.

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, it’s link status and various transmission history properties and statistics.

4.3.4.6. Cellular



General	
Self test	Passed
RF level	99
Operating mode	Packet
Network	
Registration	No
Provider	
Cell location ID	
Cell ID	
Packet network	
Connection state	Errored
Current session time	0
Total session time	0
Wireless CSD network	
Connection state	Offline
Sessions	0
Current session time	0
Total session time	0
Signal information	
RSRQ	-21 dB
RSRP	-142 dB
RSSI	-99 dBm

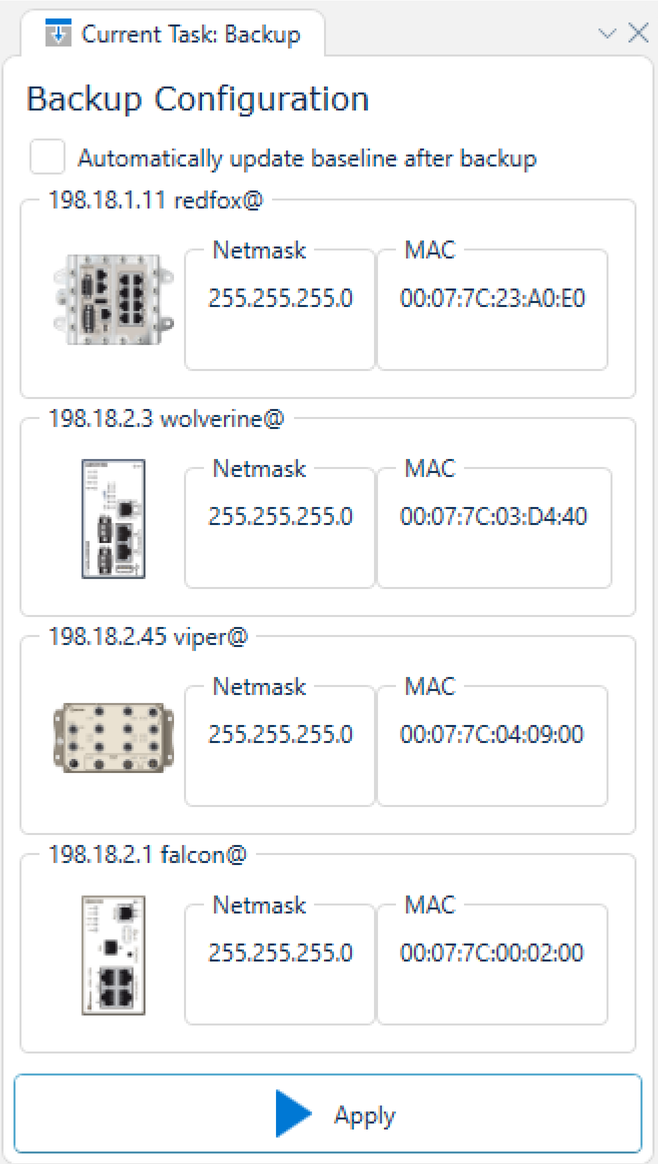
This panel displays information regarding the cellular status of the device, if applicable. The information held within primarily concerns xRD and Ibex devices, but may be applicable to more device types in the future. Depicted above is an example of a disconnected MRD-405, where it can be observed that,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

4.3.5. Maintenance

4.3.5.1. Backup

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



Current Task: Backup

Backup Configuration

□ Automatically update baseline after backup

198.18.1.11 redfox@
Netmask 255.255.255.0
MAC 00:07:7C:23:A0:E0

198.18.2.3 wolverine@
Netmask 255.255.255.0
MAC 00:07:7C:03:D4:40

198.18.2.45 viper@
Netmask 255.255.255.0
MAC 00:07:7C:04:09:00

198.18.2.1 falcon@
Netmask 255.255.255.0
MAC 00:07:7C:00:02:00

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

When selecting Automatically update baseline after backup down in the left-hand corner of the interface, then the following backup will be used as new baseline for the devices.

4.3.5.2. Firmware Upgrade

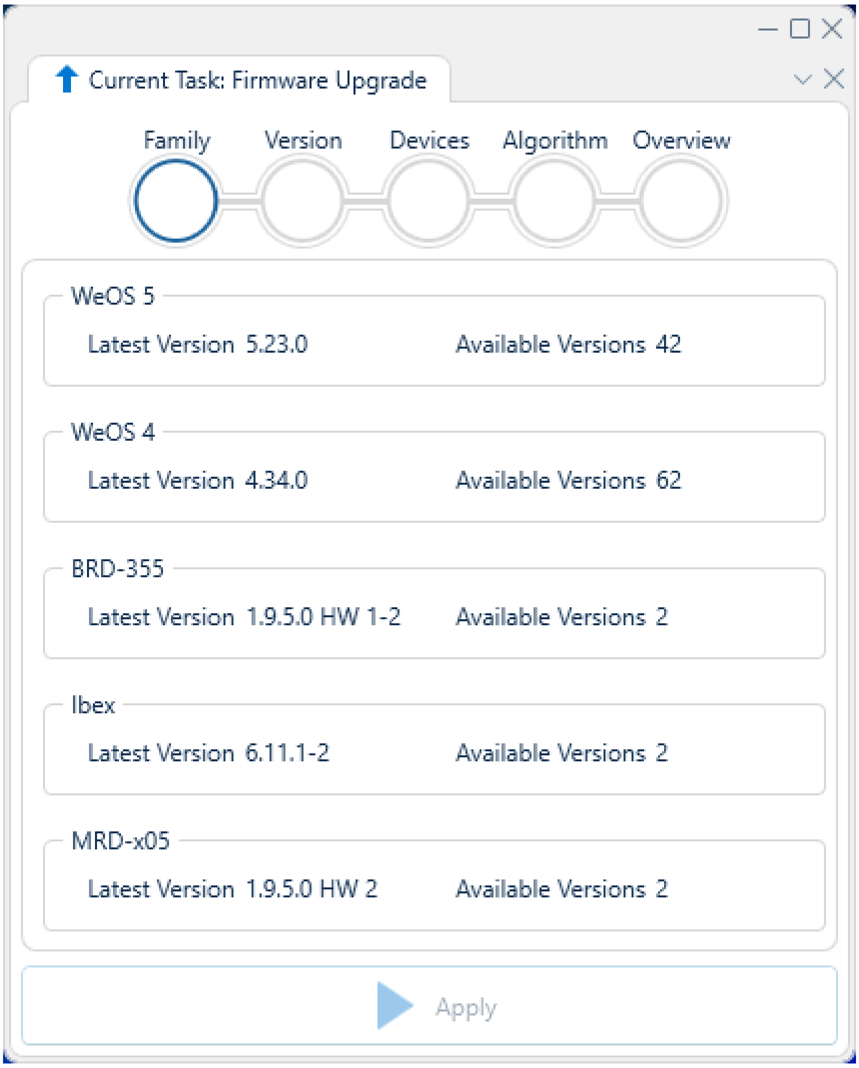
Firmware upgrade is a staged task that allows for upgrading or downgrading the running, backup and bootloader firmware on supported devices.

> *i* Info
>
> To be able to use this feature, you must be connected to the internet, or have downloaded desire 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



Current Task: Firmware Upgrade

WeOS 5
Latest Version 5.23.0 Available Versions 42

WeOS 4
Latest Version 4.34.0 Available Versions 62

BRD-355
Latest Version 1.9.5.0 HW 1-2 Available Versions 2

Ibex
Latest Version 6.11.1-2 Available Versions 2

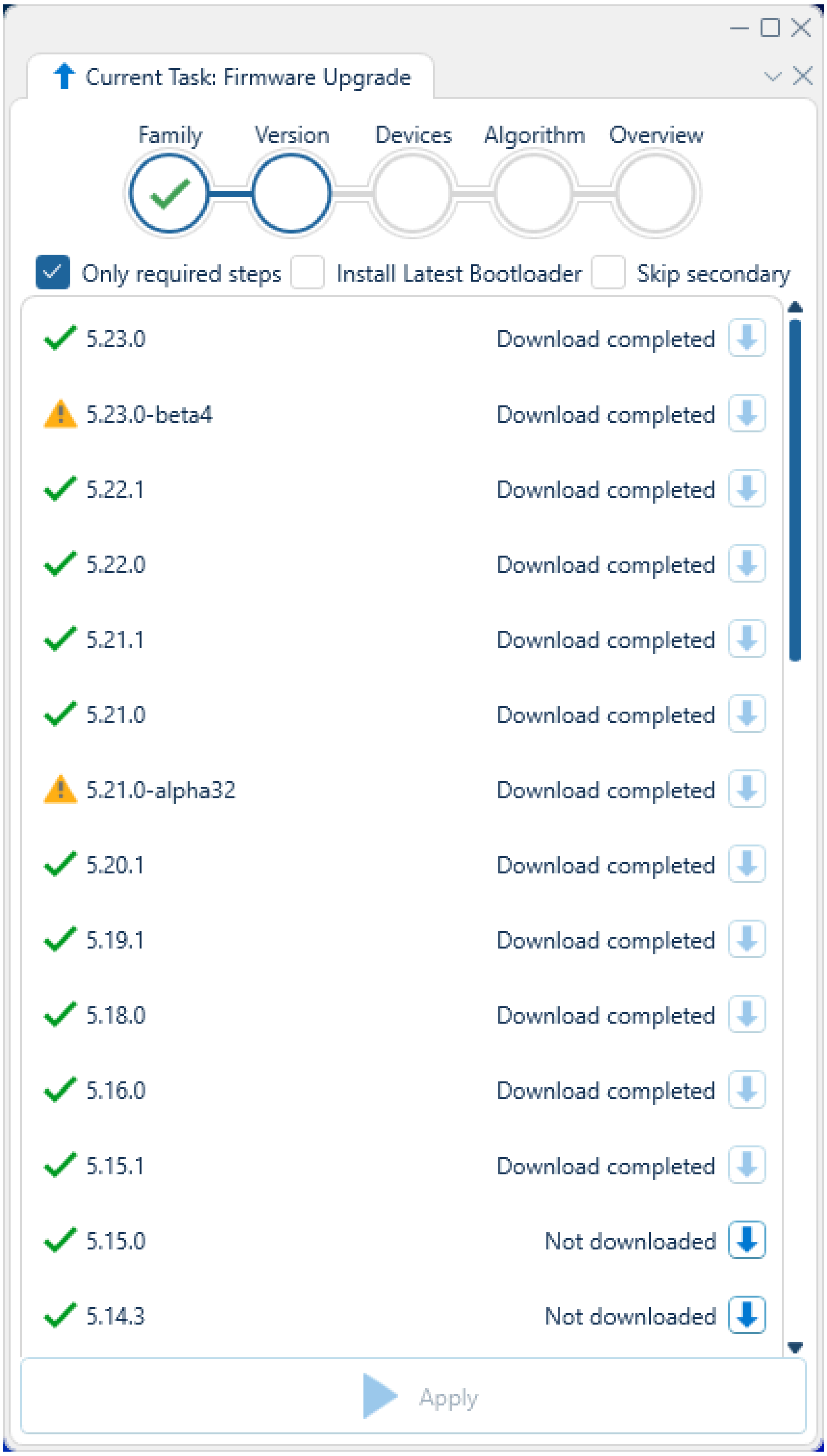
MRD-x05
Latest Version 1.9.5.0 HW 2 Available Versions 2

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 proceeed to the next stage.

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

Version



☑ Only required steps □ Install Latest Bootloader □ Skip secondary

✓	5.23.0	Download completed
⚠	5.23.0-beta4	Download completed
✓	5.22.1	Download completed
✓	5.22.0	Download completed
✓	5.21.1	Download completed
✓	5.21.0	Download completed
⚠	5.21.0-alpha32	Download completed
✓	5.20.1	Download completed
✓	5.19.1	Download completed
✓	5.18.0	Download completed
✓	5.16.0	Download completed
✓	5.15.1	Download completed
✓	5.15.0	Not downloaded
✓	5.14.3	Not downloaded

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.

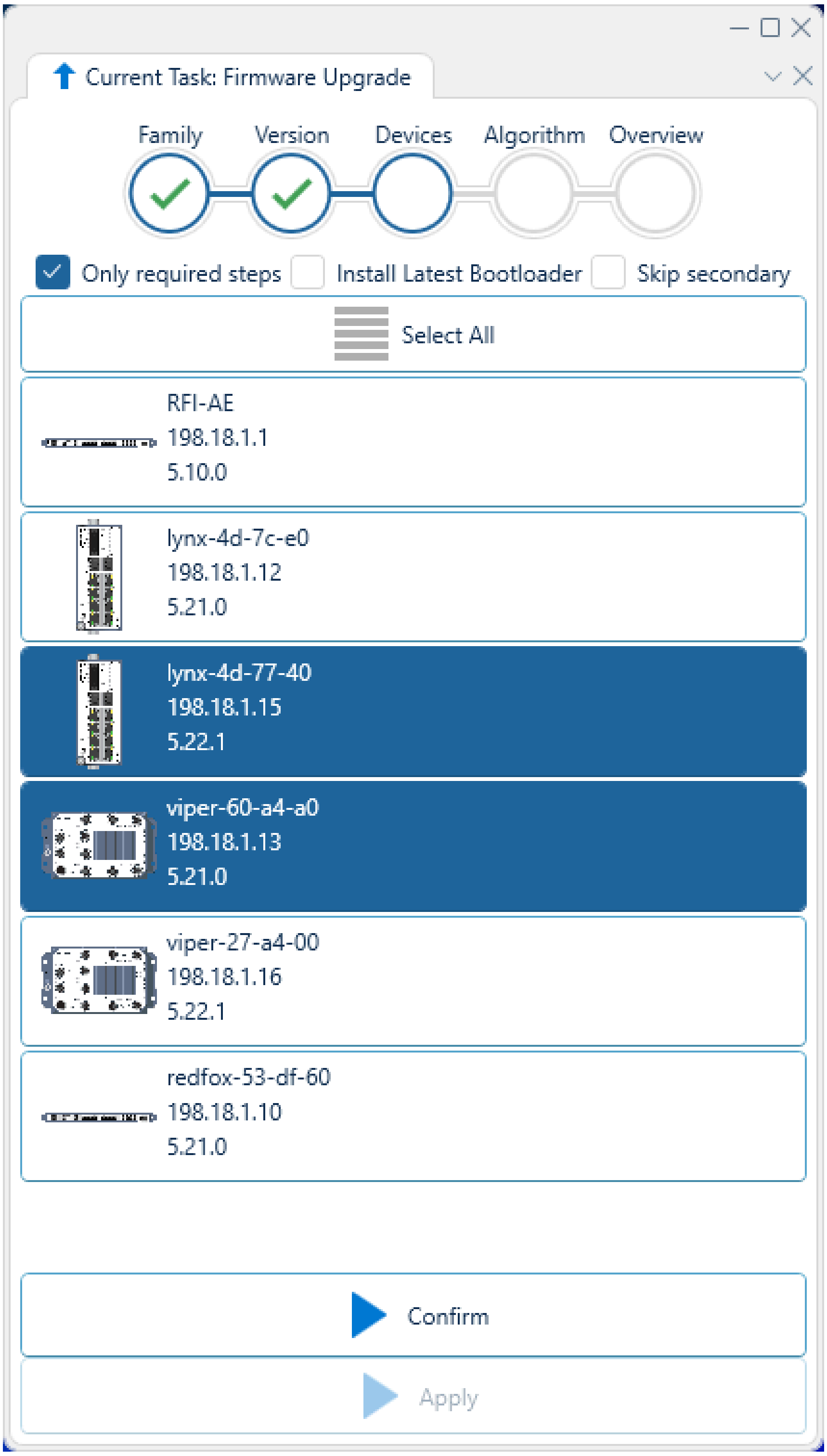
WeConfig makes no guarantees about the functionality or safety of Unvalidated firmware.

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 proceeed to the next stage.

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

Devices



☑ Only required steps □ Install Latest Bootloader □ Skip secondary

RFI-AE
198.18.1.1
5.10.0

lynx-4d-7c-e0
198.18.1.12
5.21.0

lynx-4d-77-40
198.18.1.15
5.22.1

viper-60-a4-a0
198.18.1.13
5.21.0

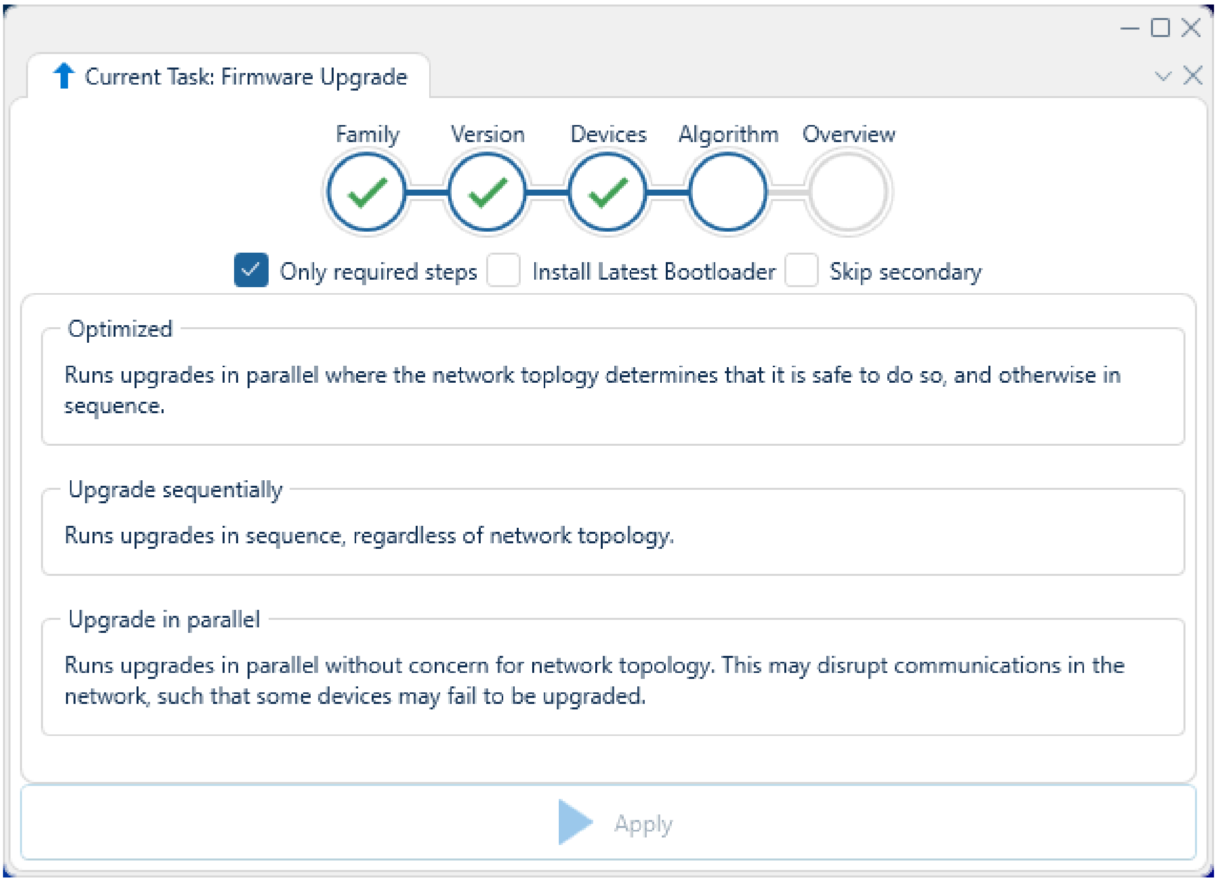
viper-27-a4-00
198.18.1.16
5.22.1

redfox-53-df-60
198.18.1.10
5.21.0

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, such as a detected device downgrade in the example above.

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

Algorithm


Current Task: Firmware Upgrade

☑ Only required steps □ Install Latest Bootloader □ Skip secondary

Optimized

Runs upgrades in parallel where the network toplogy determines that it is safe to do so, and otherwise in sequence.

Upgrade sequentially

Runs upgrades in sequence, regardless of network topology.

Upgrade in parallel

Runs upgrades in parallel without concern for network topology. This may disrupt communications in the network, such that some devices may fail to be upgraded.

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

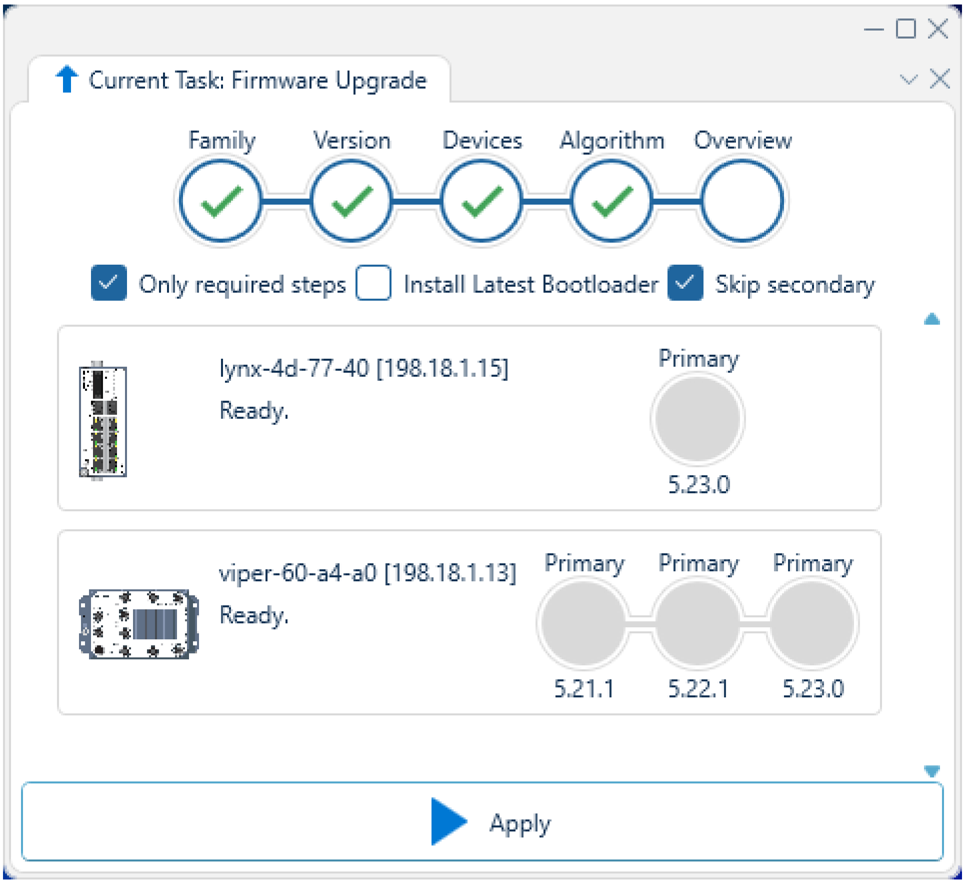
Option	Description
Optimized	Parallelizes firmware upgrade when possible according to network order .
Upgrade sequentially	Do not parallelize firmware upgrade, run them in displayed top-down order.
Upgrade in parallel	Attempt to do all firmware upgrades in parallel

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



☑ Only required steps □ Install Latest Bootloader ☑ Skip secondary

lynx-4d-77-40 [198.18.1.15]
Ready.
Primary
5.23.0

viper-60-a4-a0 [198.18.1.13]
Ready.
Primary Primary Primary
5.21.1 5.22.1 5.23.0

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 `5.23.0`, WeConfig needs to upgrade it through the required versions of `5.21.1` and `5.22.1` before finally upgrading to `5.23.0`.

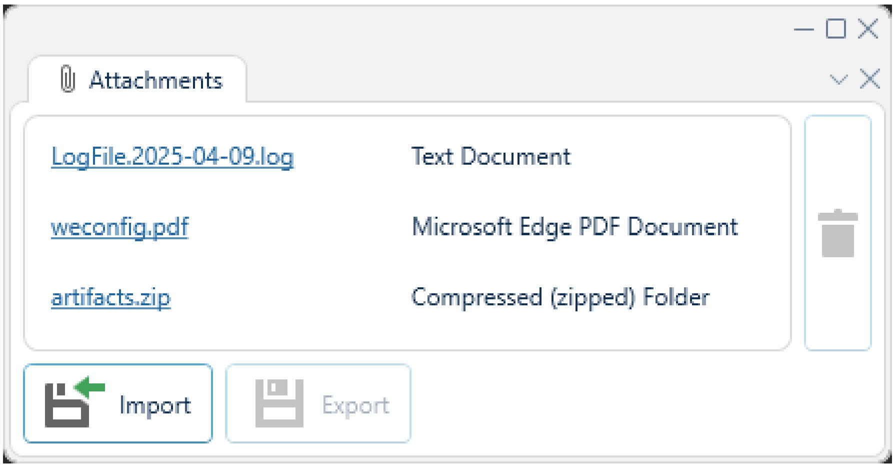
Additionally, any remaining potential issues are also displayed on the devices here.

Global options

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

Option	Description
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 nessecary
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 nessecary
Skip Secondary	When checked, WeConfig will not upgrade the backup firmware on any of the devices, otherwise, it will

4.3.5.3. Attachments



LogFile.2025-04-09.log	Text Document
weconfig.pdf	Microsoft Edge PDF Document
artifacts.zip	Compressed (zipped) Folder

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.

4.3.5.4. Device Access Settings

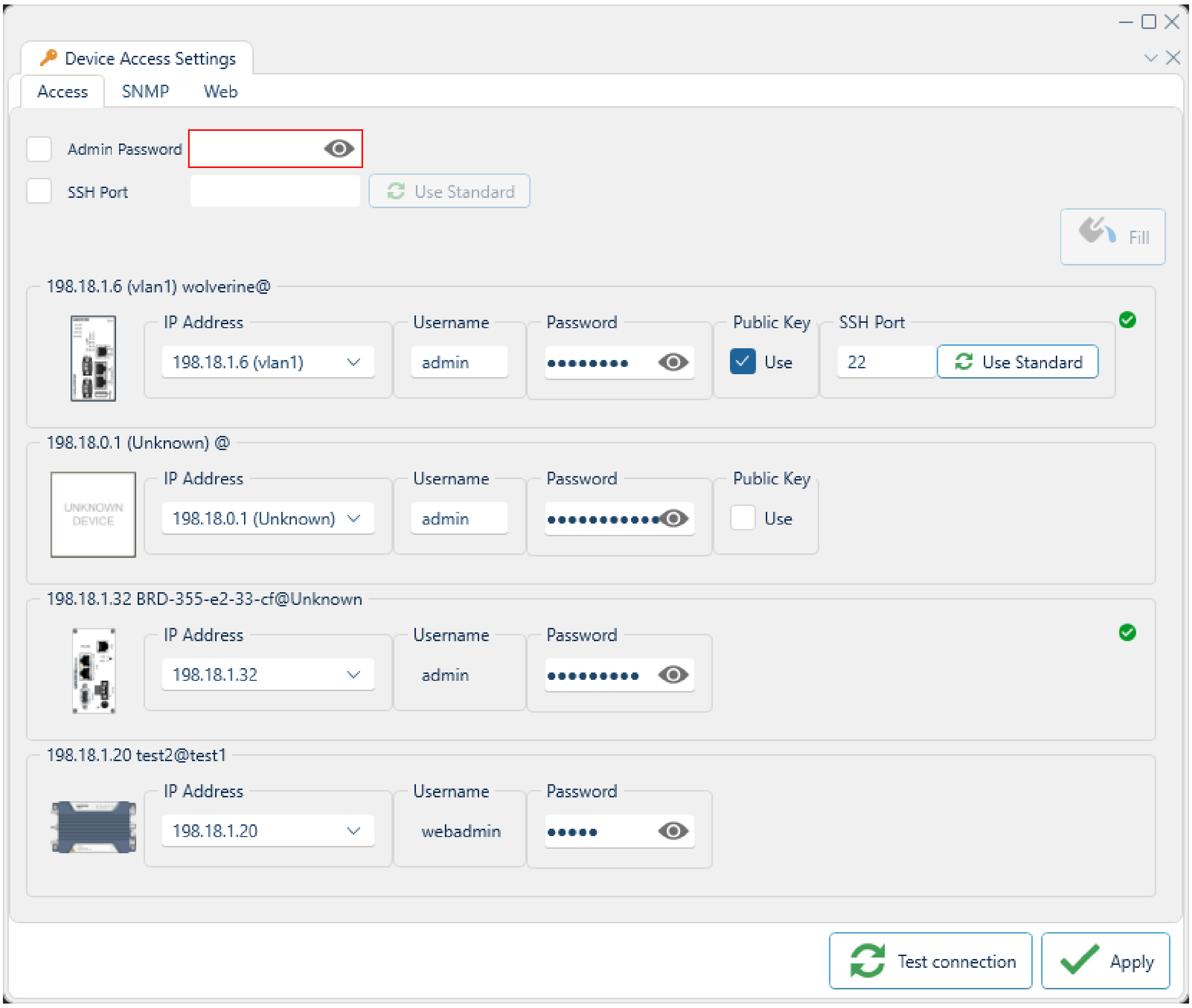
Device access settings is a panel dedicated to controling how WeConfig authenticates and communicates with the devices in the network.

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



Device Access Settings

Access | SNMP | Web

□ Admin Password

□ SSH Port | Use Standard

Fill

198.18.1.6 (vlan1) wolverine@

IP Address	Username	Password	Public Key	SSH Port
198.18.1.6 (vlan1)	admin	••••••••	☑ Use	22 Use Standard

✓

198.18.0.1 (Unknown) @

UNKNOWN DEVICE

IP Address	Username	Password	Public Key
198.18.0.1 (Unknown)	admin	•••••••••••	□ Use

198.18.1.32 BRD-355-e2-33-cf@Unknown

IP Address	Username	Password
198.18.1.32	admin	•••••••••

✓

198.18.1.20 test2@test1

IP Address	Username	Password
198.18.1.20	webadmin	•••••

Test connection | Apply

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:

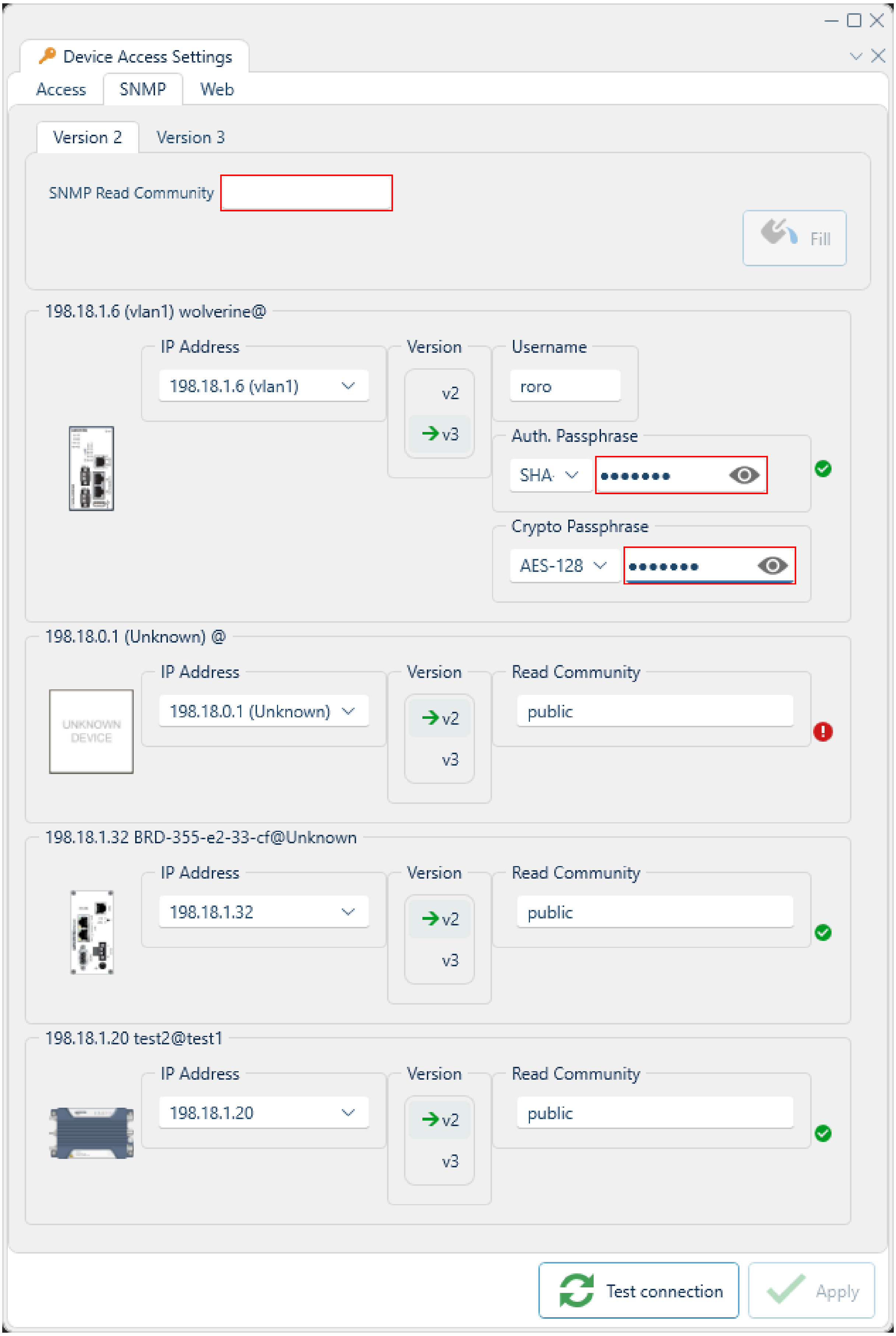
- IP Address: The management IP address used to reach the device, it is a dropdown list containing all known network interfaces / addresses that WeConfig has associated with the device.
- Username: The account username that WeConfig will try and authenticate with over SSH / Web.
- Password: The account 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` and `MRD-455` 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.

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

SNMP Tab



Device Access Settings

Access | SNMP | Web

Version 2 | Version 3

SNMP Read Community

198.18.1.6 (vlan1) wolverine@

IP Address: 198.18.1.6 (vlan1)

Version: v2, → v3

Username: roro

Auth. Passphrase: SHA ●●●●●●●

Crypto Passphrase: AES-128 ●●●●●●●

198.18.0.1 (Unknown) @

IP Address: 198.18.0.1 (Unknown)

Version: → v2, v3

Read Community: public

198.18.1.32 BRD-355-e2-33-cf@Unknown

IP Address: 198.18.1.32

Version: → v2, v3

Read Community: public

198.18.1.20 test2@test1

IP Address: 198.18.1.20

Version: → v2, v3

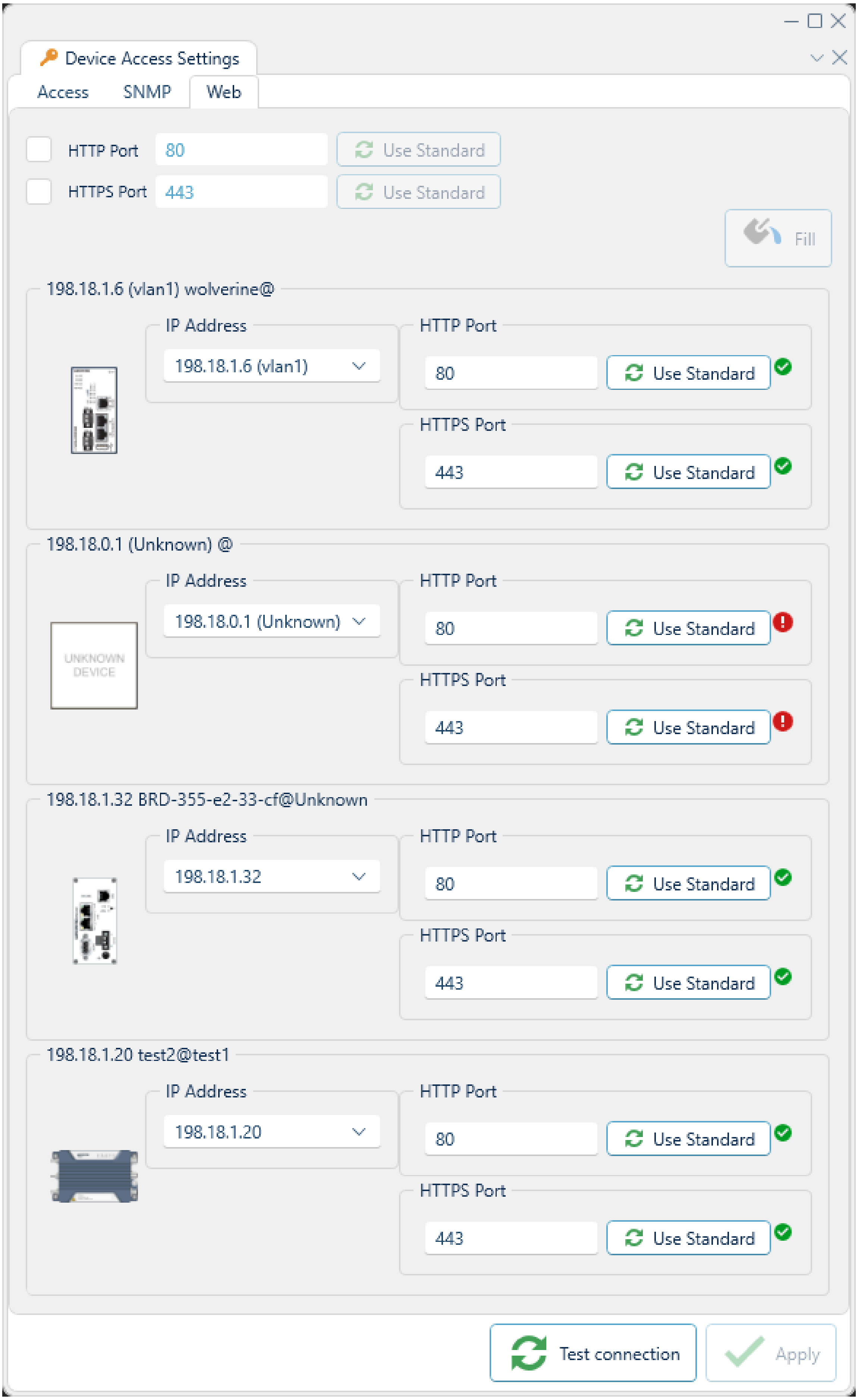
Read Community: public

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. For SNMP v2, only the read community needs to be set here. For SNMP v3, as can be seen, you must set a username, and may optionally set an authentication digest and password, as well as a privacy crypto and password.

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 contentst of the SNMP tab may be automatically updated when configuring SNMP



□ HTTP Port 80

□ HTTPS Port 443

198.18.1.6 (vlan1) wolverine@

IP Address: 198.18.1.6 (vlan1)

HTTP Port: 80 ✓

HTTPS Port: 443 ✓

198.18.0.1 (Unknown) @

IP Address: 198.18.0.1 (Unknown)

HTTP Port: 80 !

HTTPS Port: 443 !

198.18.1.32 BRD-355-e2-33-cf@Unknown

IP Address: 198.18.1.32

HTTP Port: 80 ✓

HTTPS Port: 443 ✓

198.18.1.20 test2@test1

IP Address: 198.18.1.20

HTTP Port: 80 ✓

HTTPS Port: 443 ✓

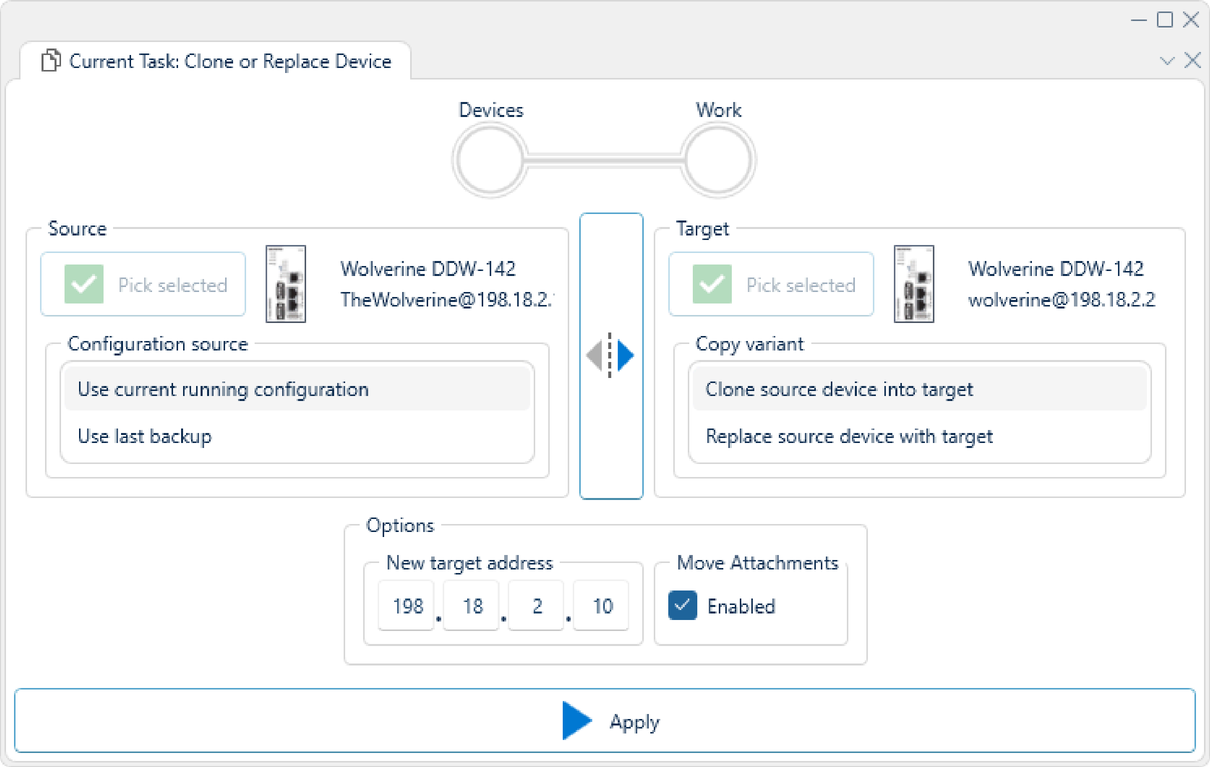
The third and final tab in this panel, labeled Web, corresponds to device specific settings for overriding the default HTTP and HTTPs ports. This interface is fairly straightforwards, and does not differ between device firmware families.

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

4.3.5.5. Clone or Replace Device

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



Wolverine DDW-142
TheWolverine@198.18.2.

Use current running configuration
Use last backup

Wolverine DDW-142
wolverine@198.18.2.2

Clone source device into target
Replace source device with target

New target address: 198 . 18 . 2 . 10

☑ Enabled

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.

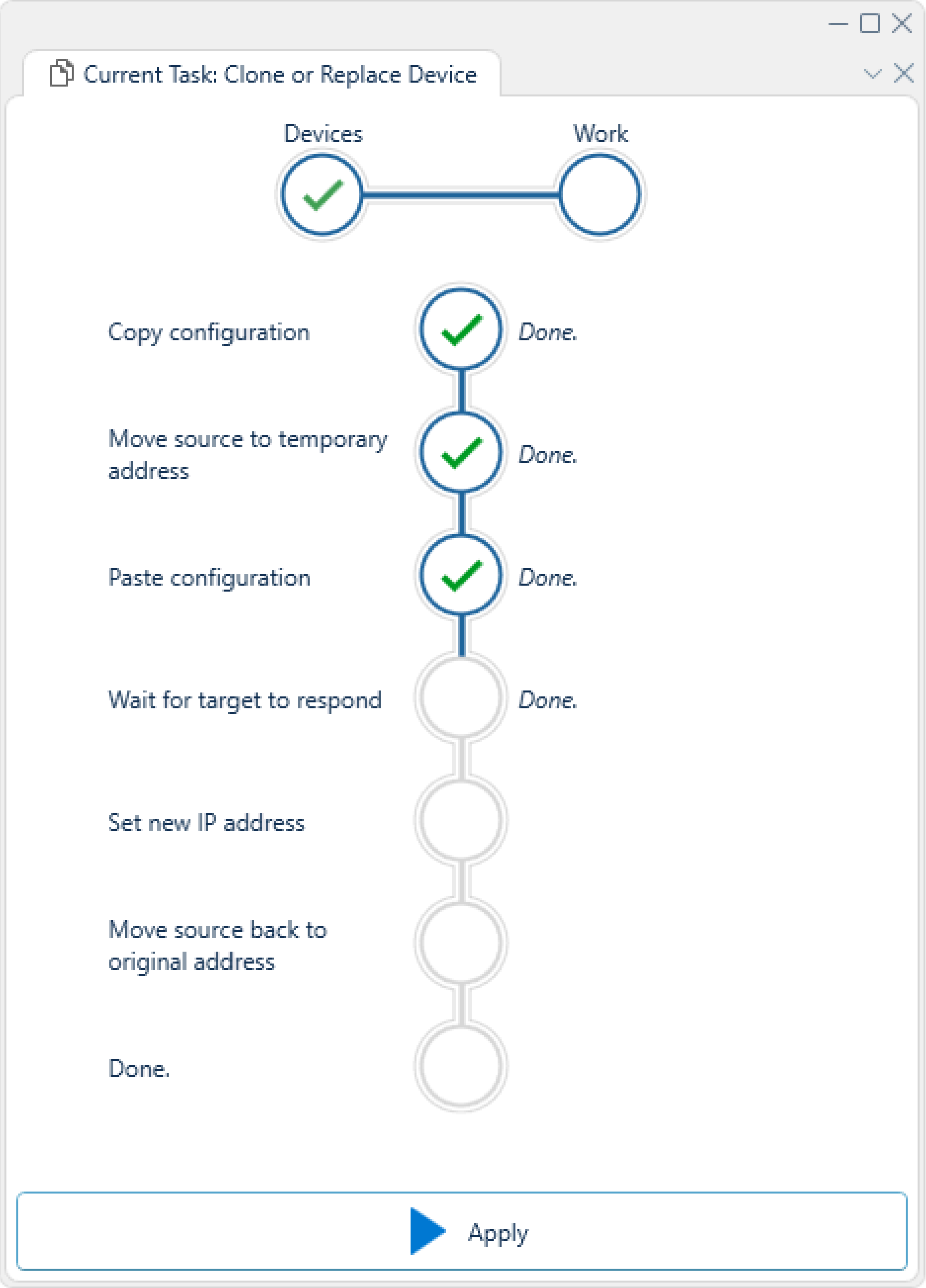
If you have selected the devices in the wrong order, and wish to swap the source and target of the operation, the button between the two selections allows you to do so trivially.

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 recieve a new IP address, such as

Option	Description
	for older versions of WeOS 4, this option will appear and prompt the user to select the adapter that the device is connected to.
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:



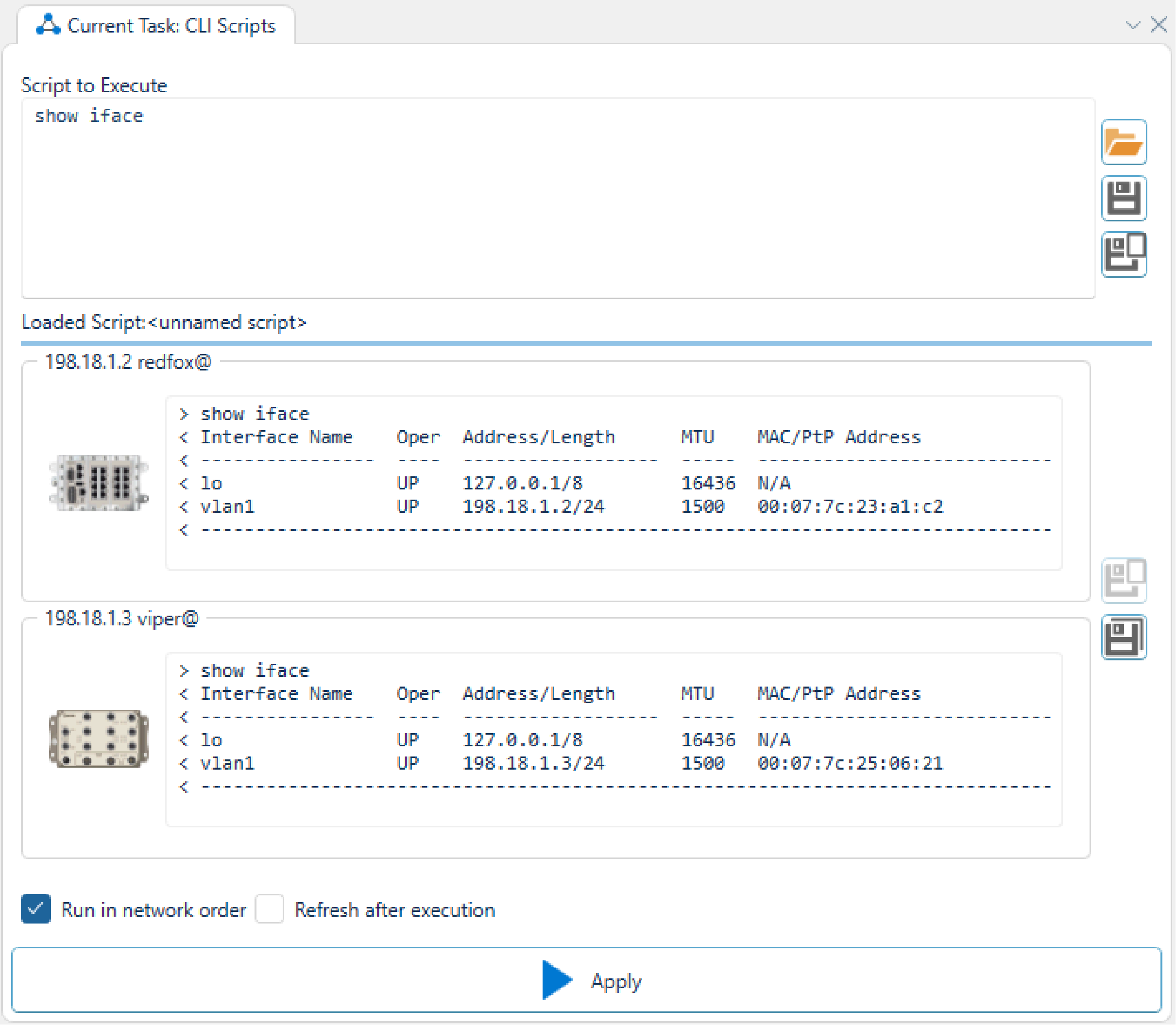
Copy configuration	Done.
Move source to temporary address	Done.
Paste configuration	Done.
Wait for target to respond	Done.
Set new IP address	
Move source back to original address	
Done.	

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.

4.3.5.6. CLI Scripts

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

Interface Components



```
show iface
```

Loaded Script:<unnamed script>

198.18.1.2 redfox@

```
> show iface
< Interface Name    Oper  Address/Length      MTU    MAC/PtP Address
< ----------------  ----  ------------------  -----  --------------------------
< lo                UP    127.0.0.1/8         16436  N/A
< vlan1             UP    198.18.1.2/24       1500   00:07:7c:23:a1:c2
< ------------------------------------------------------------------------------
```

198.18.1.3 viper@

```
> show iface
< Interface Name    Oper  Address/Length      MTU    MAC/PtP Address
< ----------------  ----  ------------------  -----  --------------------------
< lo                UP    127.0.0.1/8         16436  N/A
< vlan1             UP    198.18.1.3/24       1500   00:07:7c:25:06:21
< ------------------------------------------------------------------------------
```

☑ Run in network order □ Refresh after execution

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 set of two save buttons, these will save the input/output of the scripts per device to the project / a user specified location respectively.

Run in network order

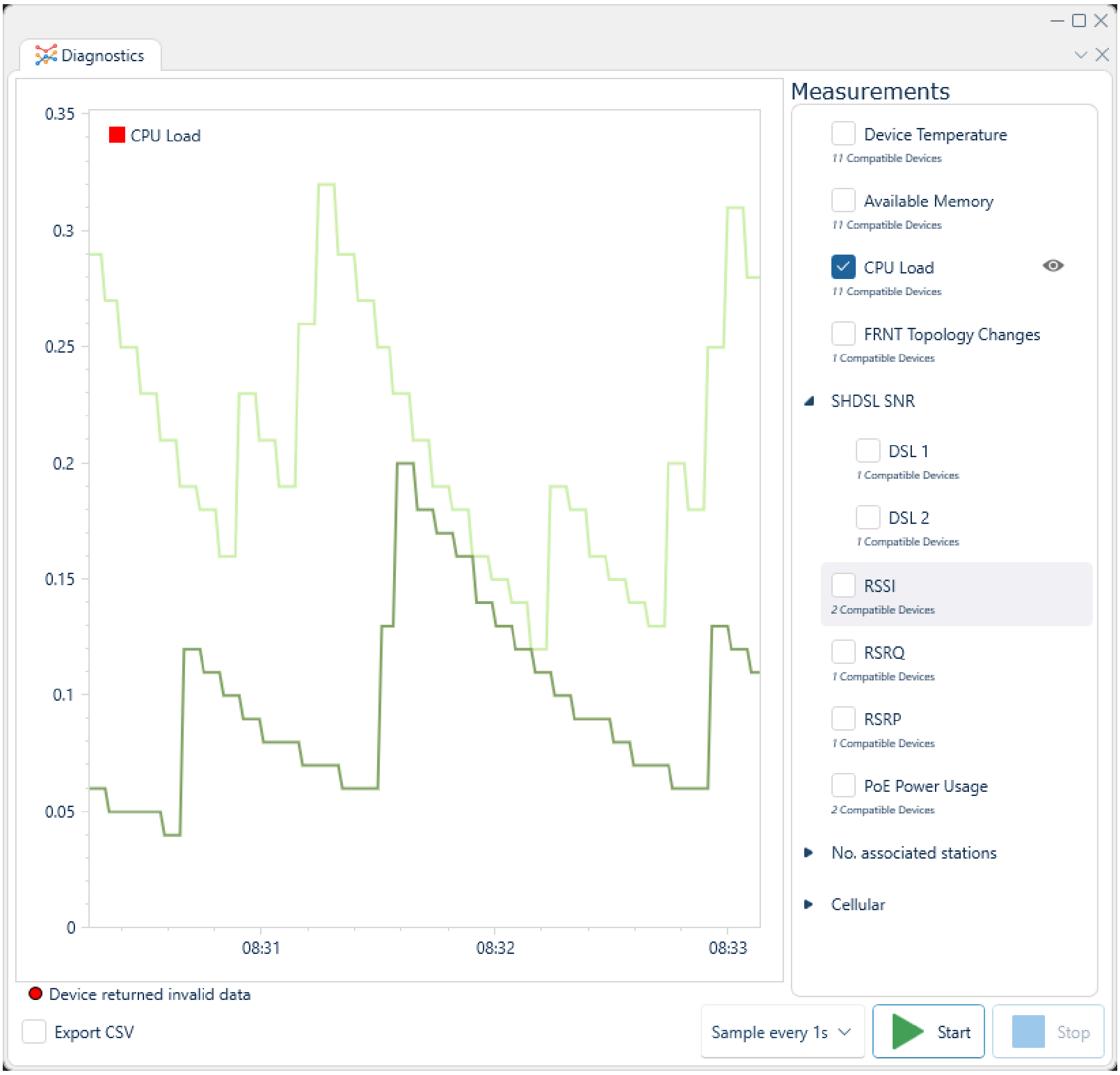
This checkbox, when checked, will cause the script to run in network topology order, rather than in complete parallel.

Refresh after execution

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

4.3.6. Diagnostics

4.3.6.1. Diagnostics



Measurements

- □ Device Temperature — 11 Compatible Devices
- ☑ CPU Load — 11 Compatible Devices
- □ Available Memory — 11 Compatible Devices
- □ FRNT Topology Changes — 1 Compatible Devices
- SHDSL SNR
  - □ DSL 1 — 1 Compatible Devices
  - □ DSL 2 — 1 Compatible Devices
- □ RSSI — 2 Compatible Devices
- □ RSRQ — 1 Compatible Devices
- □ RSRP — 1 Compatible Devices
- □ PoE Power Usage — 2 Compatible Devices
- No. associated stations
- Cellular

□ Export CSV

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 “eye” icon 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.

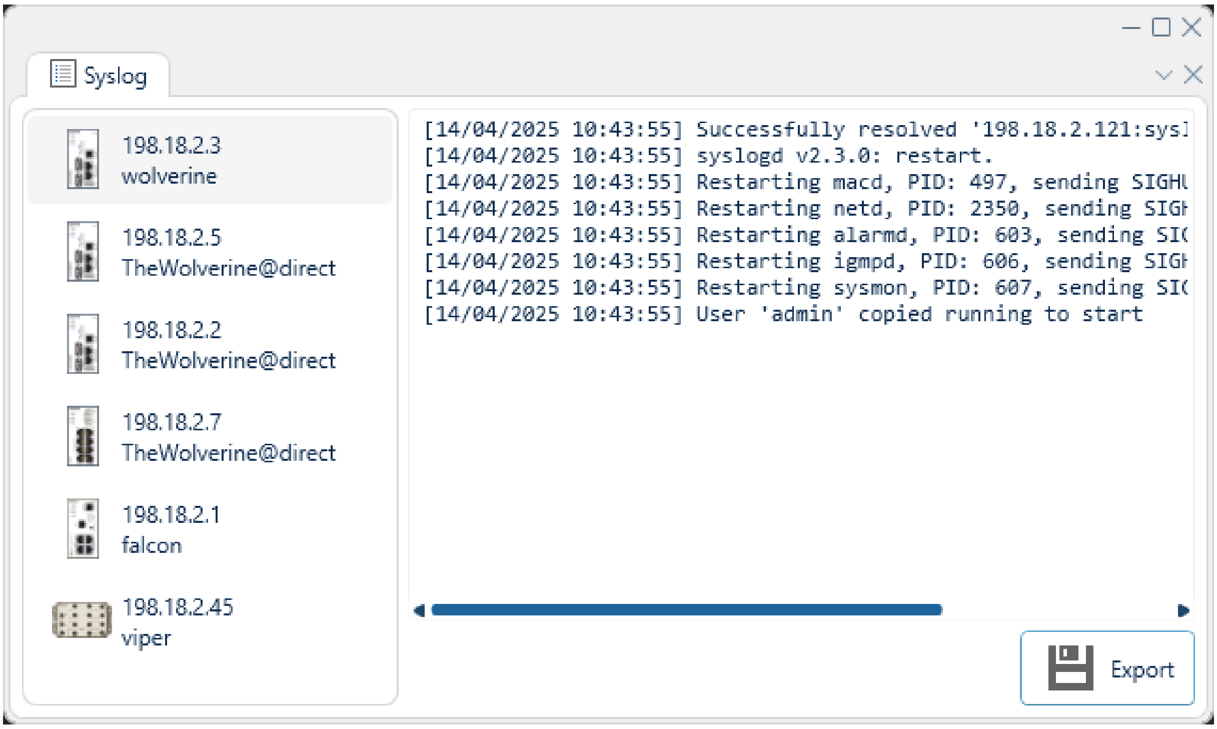
> Note
>
> When a monitor session is restarted, the graph is cleared. Data saved to CSV will not be lost. A new monitor session will add data to the CSV file, not replace it.

> ⚠ 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.

4.3.6.2. Syslog

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

Interface components



```
[14/04/2025 10:43:55] Successfully resolved '198.18.2.121:sysl
[14/04/2025 10:43:55] syslogd v2.3.0: restart.
[14/04/2025 10:43:55] Restarting macd, PID: 497, sending SIGHU
[14/04/2025 10:43:55] Restarting netd, PID: 2350, sending SIGH
[14/04/2025 10:43:55] Restarting alarmd, PID: 603, sending SIG
[14/04/2025 10:43:55] Restarting igmpd, PID: 606, sending SIGH
[14/04/2025 10:43:55] Restarting sysmon, PID: 607, sending SIG
[14/04/2025 10:43:55] User 'admin' copied running to start
```

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.

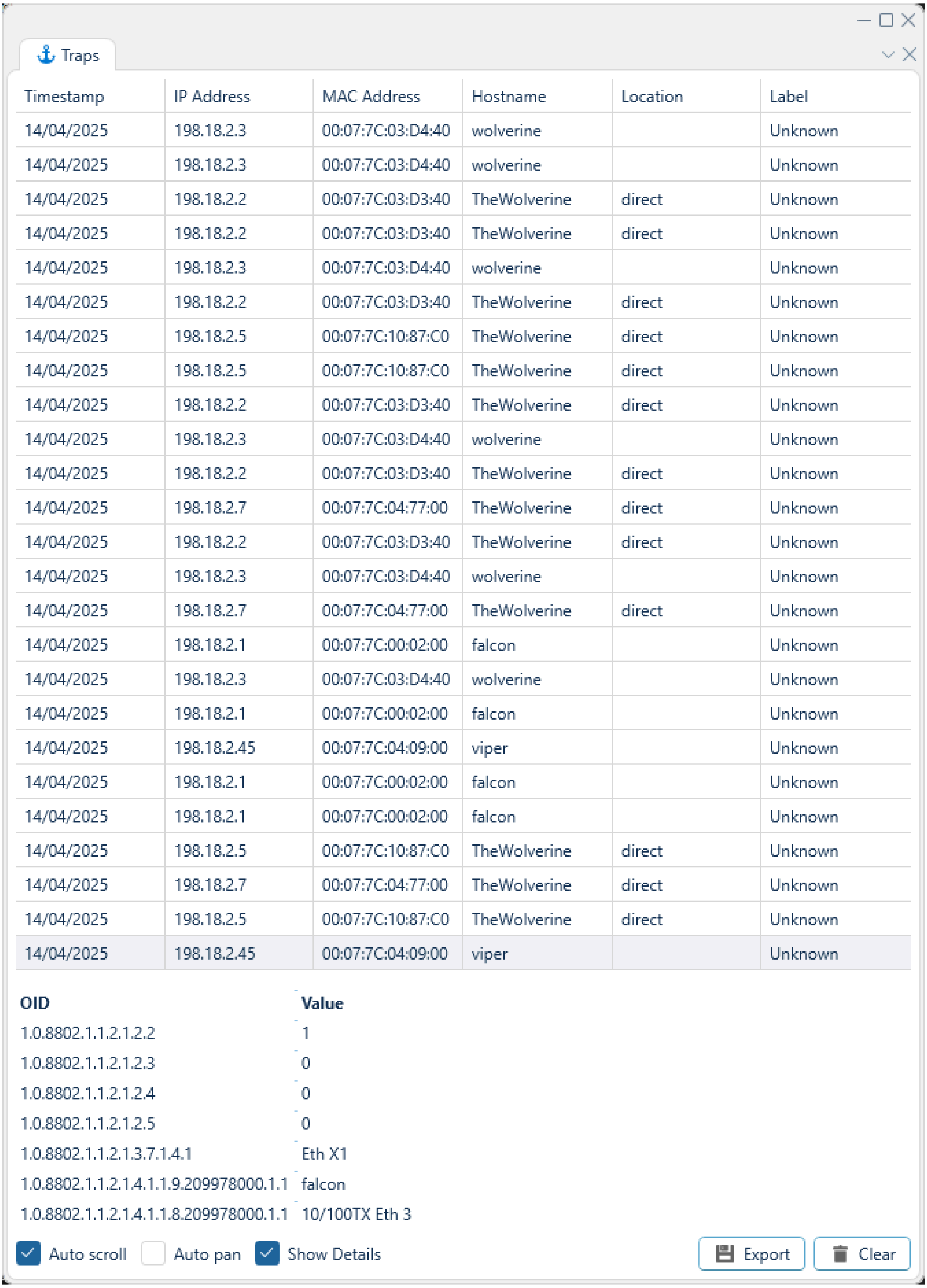
> *i* Info
>
> In order to receive Syslog messages, the running machine's firewall must be configured to allow said syslog messages, which arrive on Port 514, see usage for more information.

4.3.6.3. Traps

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



Timestamp	IP Address	MAC Address	Hostname	Location	Label
14/04/2025	198.18.2.3	00:07:7C:03:D4:40	wolverine		Unknown
14/04/2025	198.18.2.3	00:07:7C:03:D4:40	wolverine		Unknown
14/04/2025	198.18.2.2	00:07:7C:03:D3:40	TheWolverine	direct	Unknown
14/04/2025	198.18.2.2	00:07:7C:03:D3:40	TheWolverine	direct	Unknown
14/04/2025	198.18.2.3	00:07:7C:03:D4:40	wolverine		Unknown
14/04/2025	198.18.2.2	00:07:7C:03:D3:40	TheWolverine	direct	Unknown
14/04/2025	198.18.2.5	00:07:7C:10:87:C0	TheWolverine	direct	Unknown
14/04/2025	198.18.2.5	00:07:7C:10:87:C0	TheWolverine	direct	Unknown
14/04/2025	198.18.2.2	00:07:7C:03:D3:40	TheWolverine	direct	Unknown
14/04/2025	198.18.2.3	00:07:7C:03:D4:40	wolverine		Unknown
14/04/2025	198.18.2.2	00:07:7C:03:D3:40	TheWolverine	direct	Unknown
14/04/2025	198.18.2.7	00:07:7C:04:77:00	TheWolverine	direct	Unknown
14/04/2025	198.18.2.2	00:07:7C:03:D3:40	TheWolverine	direct	Unknown
14/04/2025	198.18.2.3	00:07:7C:03:D4:40	wolverine		Unknown
14/04/2025	198.18.2.7	00:07:7C:04:77:00	TheWolverine	direct	Unknown
14/04/2025	198.18.2.1	00:07:7C:00:02:00	falcon		Unknown
14/04/2025	198.18.2.3	00:07:7C:03:D4:40	wolverine		Unknown
14/04/2025	198.18.2.1	00:07:7C:00:02:00	falcon		Unknown
14/04/2025	198.18.2.45	00:07:7C:04:09:00	viper		Unknown
14/04/2025	198.18.2.1	00:07:7C:00:02:00	falcon		Unknown
14/04/2025	198.18.2.1	00:07:7C:00:02:00	falcon		Unknown
14/04/2025	198.18.2.5	00:07:7C:10:87:C0	TheWolverine	direct	Unknown
14/04/2025	198.18.2.7	00:07:7C:04:77:00	TheWolverine	direct	Unknown
14/04/2025	198.18.2.5	00:07:7C:10:87:C0	TheWolverine	direct	Unknown
14/04/2025	198.18.2.45	00:07:7C:04:09:00	viper		Unknown

OID	Value
1.0.8802.1.1.2.1.2.2	1
1.0.8802.1.1.2.1.2.3	0
1.0.8802.1.1.2.1.2.4	0
1.0.8802.1.1.2.1.2.5	0
1.0.8802.1.1.2.1.3.7.1.4.1	Eth X1
1.0.8802.1.1.2.1.4.1.1.9.209978000.1.1	falcon
1.0.8802.1.1.2.1.4.1.1.8.209978000.1.1	10/100TX Eth 3

☑ Auto scroll ☐ Auto pan ☑ Show Details

As depicted in the example above, arriving SNMP traps will be listed in this panel, and can be sorted or filtered within. 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.

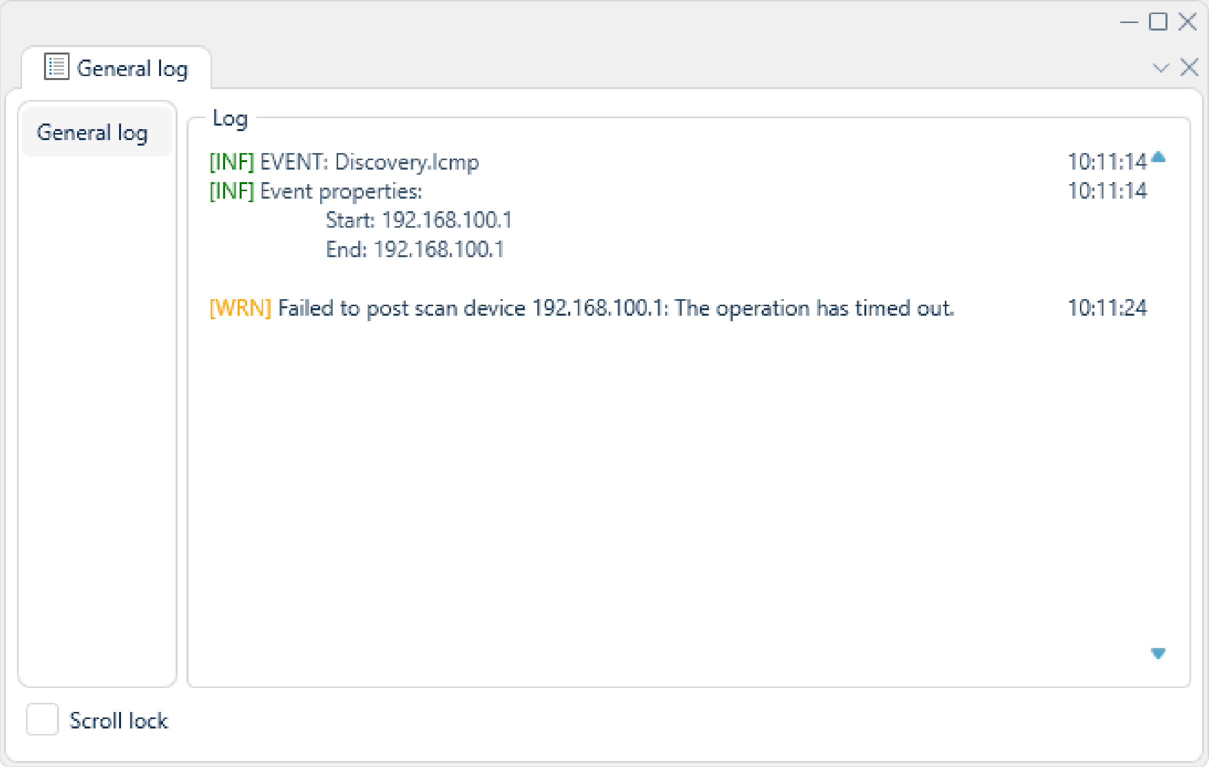
Click the Export button to export the list to a CSV file.

Click the Clear button to clear the list.

> *i* Info
>
> 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.

4.3.7. Support

4.3.7.1. General Log



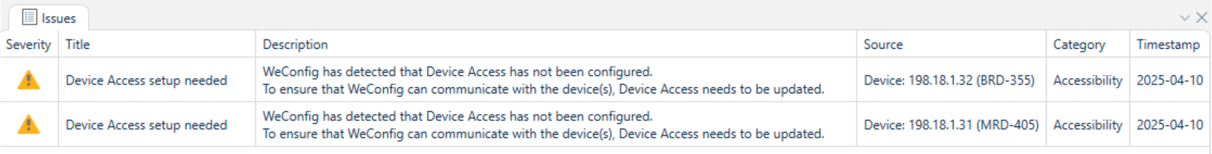
```
[INF] EVENT: Discovery.Icmp                                                    10:11:14
[INF] Event properties:                                                        10:11:14
            Start: 192.168.100.1
            End: 192.168.100.1

[WRN] Failed to post scan device 192.168.100.1: The operation has timed out.   10:11:24
```

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.

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

4.3.7.2. Issues



Issues

Severity	Title	Description	Source	Category	Timestamp
⚠	Device Access setup needed	WeConfig has detected that Device Access has not been configured. To ensure that WeConfig can communicate with the device(s), Device Access needs to be updated.	Device: 198.18.1.32 (BRD-355)	Accessibility	2025-04-10
⚠	Device Access setup needed	WeConfig has detected that Device Access has not been configured. To ensure that WeConfig can communicate with the device(s), Device Access needs to be updated.	Device: 198.18.1.31 (MRD-405)	Accessibility	2025-04-10

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.

i 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.

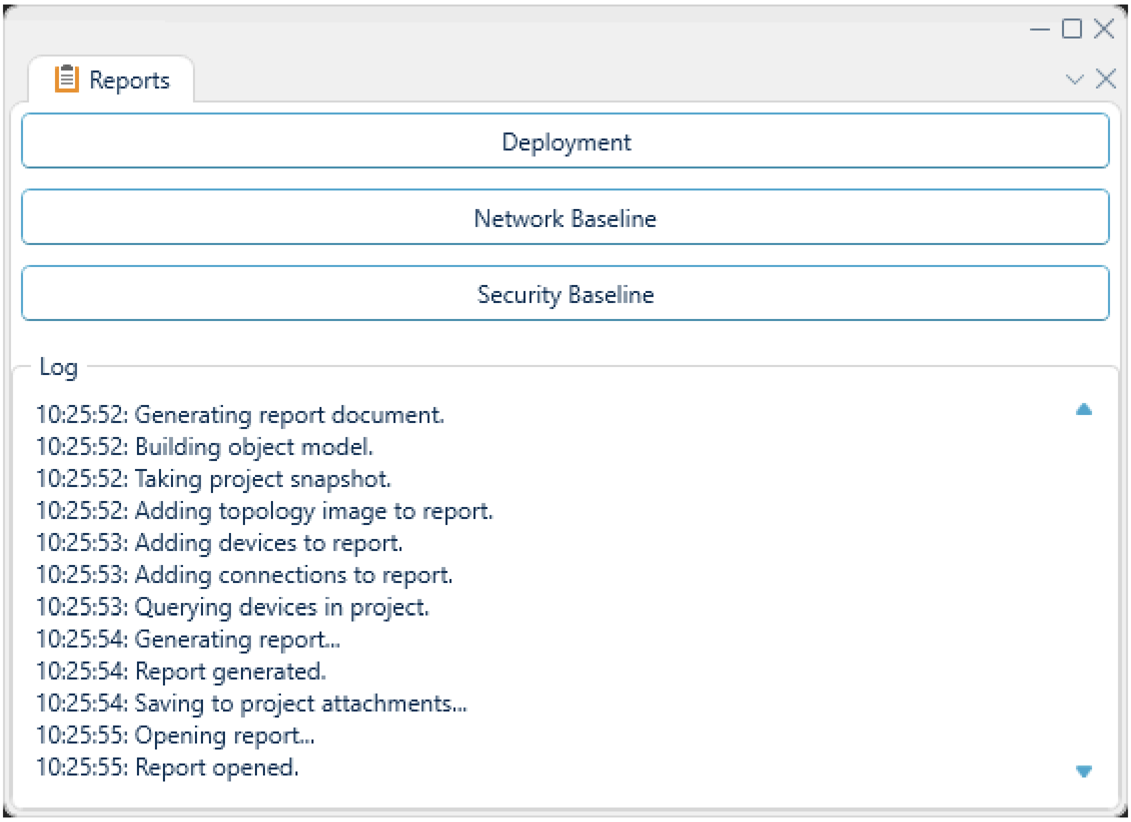
✕ Danger

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

4.3.7.3. Reports

This panel allows for the generation of reports about the current network.

Interface components

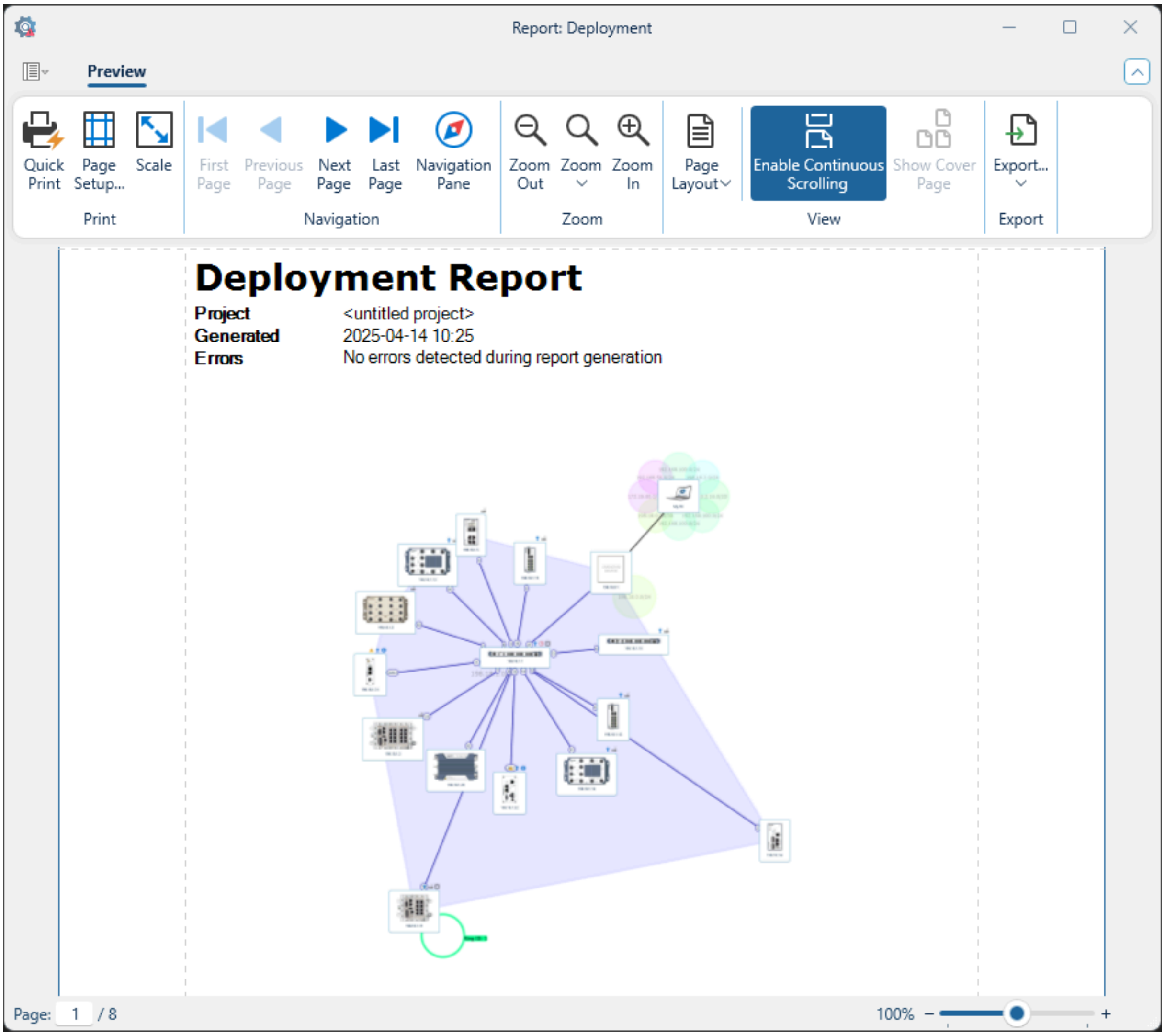


```
10:25:52: Generating report document.
10:25:52: Building object model.
10:25:52: Taking project snapshot.
10:25:52: Adding topology image to report.
10:25:53: Adding devices to report.
10:25:53: Adding connections to report.
10:25:53: Querying devices in project.
10:25:54: Generating report...
10:25:54: Report generated.
10:25:54: Saving to project attachments...
10:25:55: Opening report...
10:25:55: Report opened.
```

As depicted above, this panel today contains three different reports, as follows:

Report	Contents
Deployment	Devices and their connections
Security Baseline	Potential vulnerabilities and security issues
Network Baseline	Characteristics and settings tied to system function / performance

In order to generate the report of choice, click the corresponding button in the interface depicted above. During report generation, a log will be populated by the current progress. Once report generation has finished, a preview of the report will be opened, as with an example depicted below:



Deployment Report

Project	<untitled project>
Generated	2025-04-14 10:25
Errors	No errors detected during report generation

In this preview, you may edit margins, scale, and export the report into various formats or print it. The initial report will also be added to the project attachments.

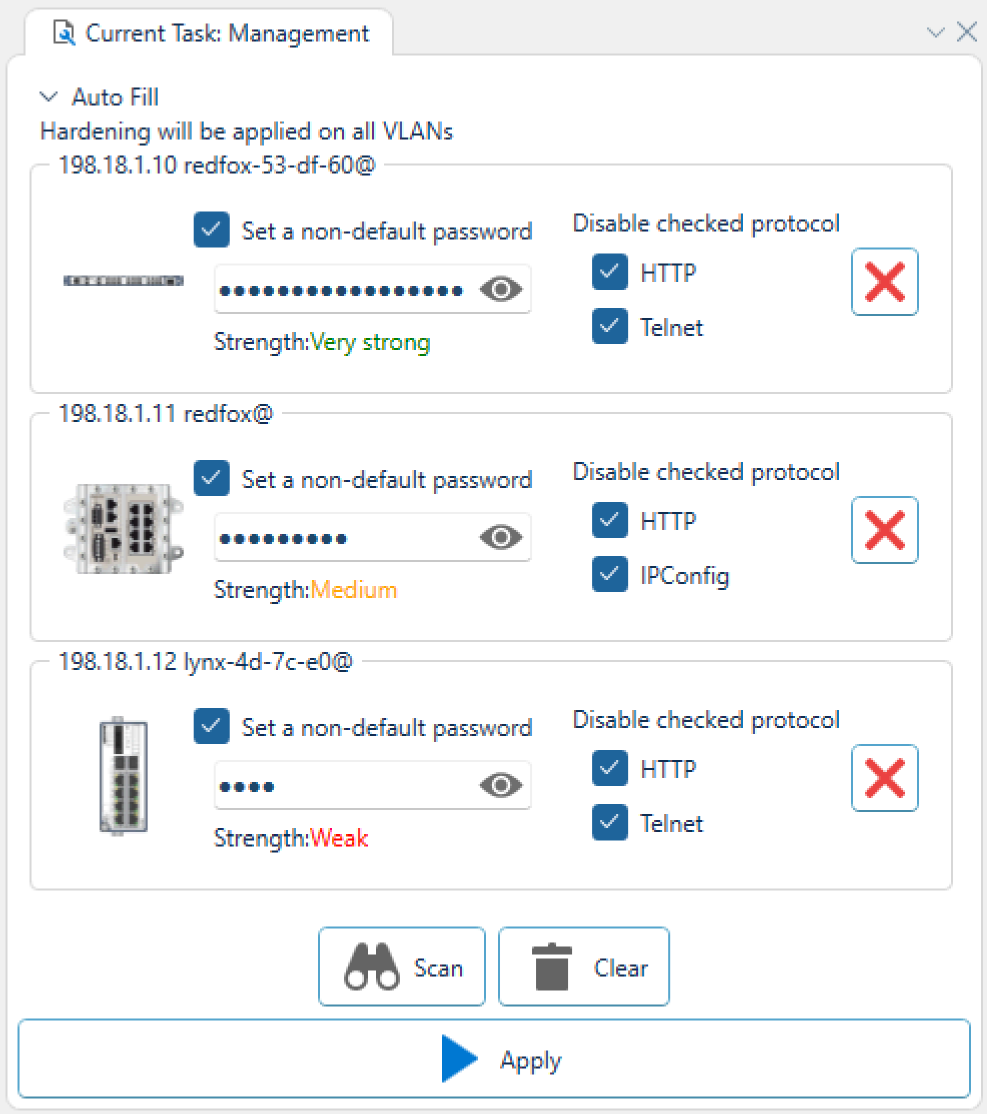
> ⚠ Warning
>
> In order to successfully generate any report, WeConfig must have a network connection to the devices involved.

4.3.8. Security

4.3.8.1. Management Hardening

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

Interface components



Current Task: Management

Auto Fill

Hardening will be applied on all VLANs

198.18.1.10 redfox-53-df-60@

☑ Set a non-default password

Strength:Very strong

Disable checked protocol

☑ HTTP

☑ Telnet

198.18.1.11 redfox@

☑ Set a non-default password

Strength:Medium

Disable checked protocol

☑ HTTP

☑ IPConfig

198.18.1.12 lynx-4d-7c-e0@

☑ Set a non-default password

Strength:Weak

Disable checked protocol

☑ HTTP

☑ Telnet

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.

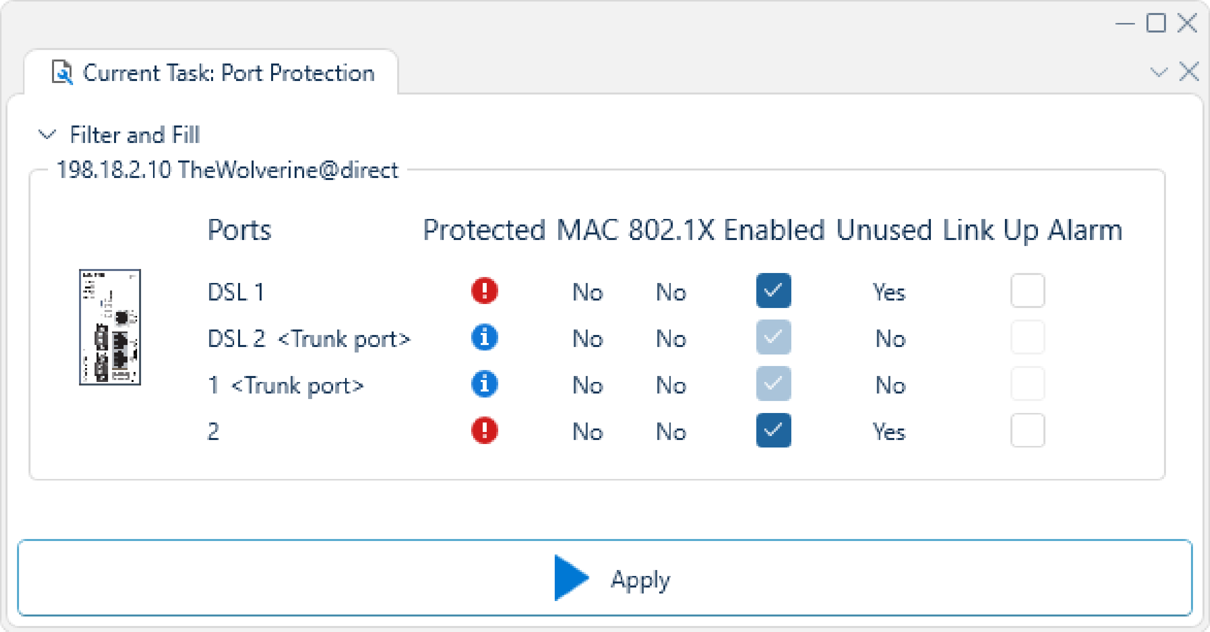
WeConfig will by default suggest removing all issues. If the default admin password has been used on any device, it will not be possible to apply the fixes until the password has been changed.

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.

4.3.8.2. Port Protection

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



198.18.2.10 TheWolverine@direct

Ports	Protected	MAC	802.1X	Enabled	Unused	Link Up Alarm
DSL 1	!	No	No	☑	Yes	□
DSL 2 <Trunk port>	ℹ	No	No	☑	No	□
1 <Trunk port>	ℹ	No	No	☑	No	□
2	!	No	No	☑	Yes	□

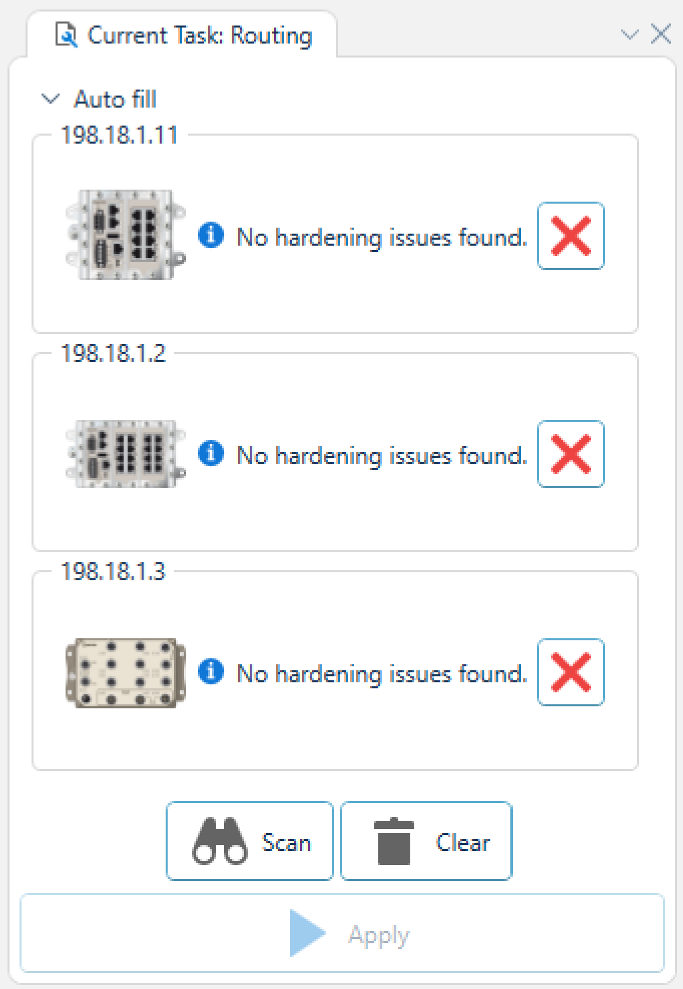
As can be seen in the interface depicted above, this panel will present a list of selected devices, where each device contains a matrix of ports with 7 columns, as follows:

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
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

4.3.8.3. Routing Hardening

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

Interface components



198.18.1.11

No hardening issues found.

198.18.1.2

No hardening issues found.

198.18.1.3

No hardening issues found.

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.

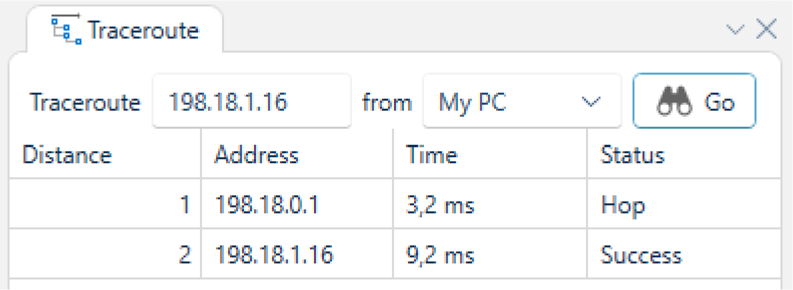
When Scan is clicked, each device in the project will be interrogated to see whether there are router configurations that do not use MD5-HMAC signatures.

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.

4.3.9. Tools

4.3.9.1. Traceroute

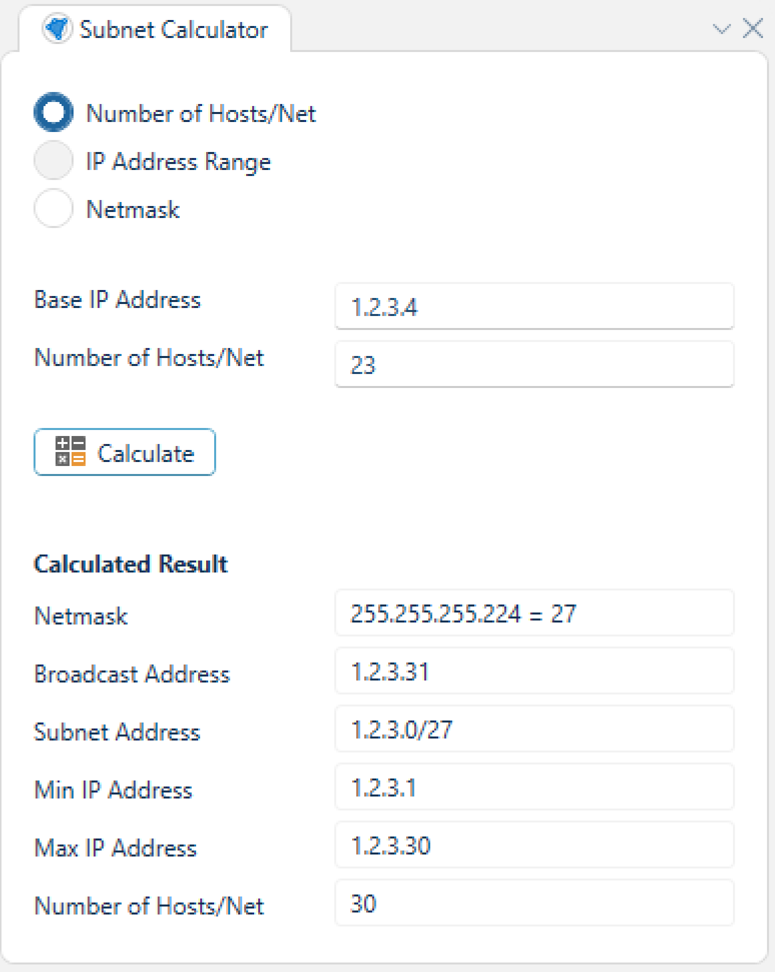


Traceroute 198.18.1.16 from My PC

Distance	Address	Time	Status
1	198.18.0.1	3,2 ms	Hop
2	198.18.1.16	9,2 ms	Success

This panel allows the user to either trace the route between devices. In order to run a traceroute, fill in the target IP address 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.

4.3.9.2. Subnet Calculator



● Number of Hosts/Net
○ IP Address Range
○ Netmask

Base IP Address	1.2.3.4
Number of Hosts/Net	23

Calculated Result

Netmask	255.255.255.224 = 27
Broadcast Address	1.2.3.31
Subnet Address	1.2.3.0/27
Min IP Address	1.2.3.1
Max IP Address	1.2.3.30
Number of Hosts/Net	30

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

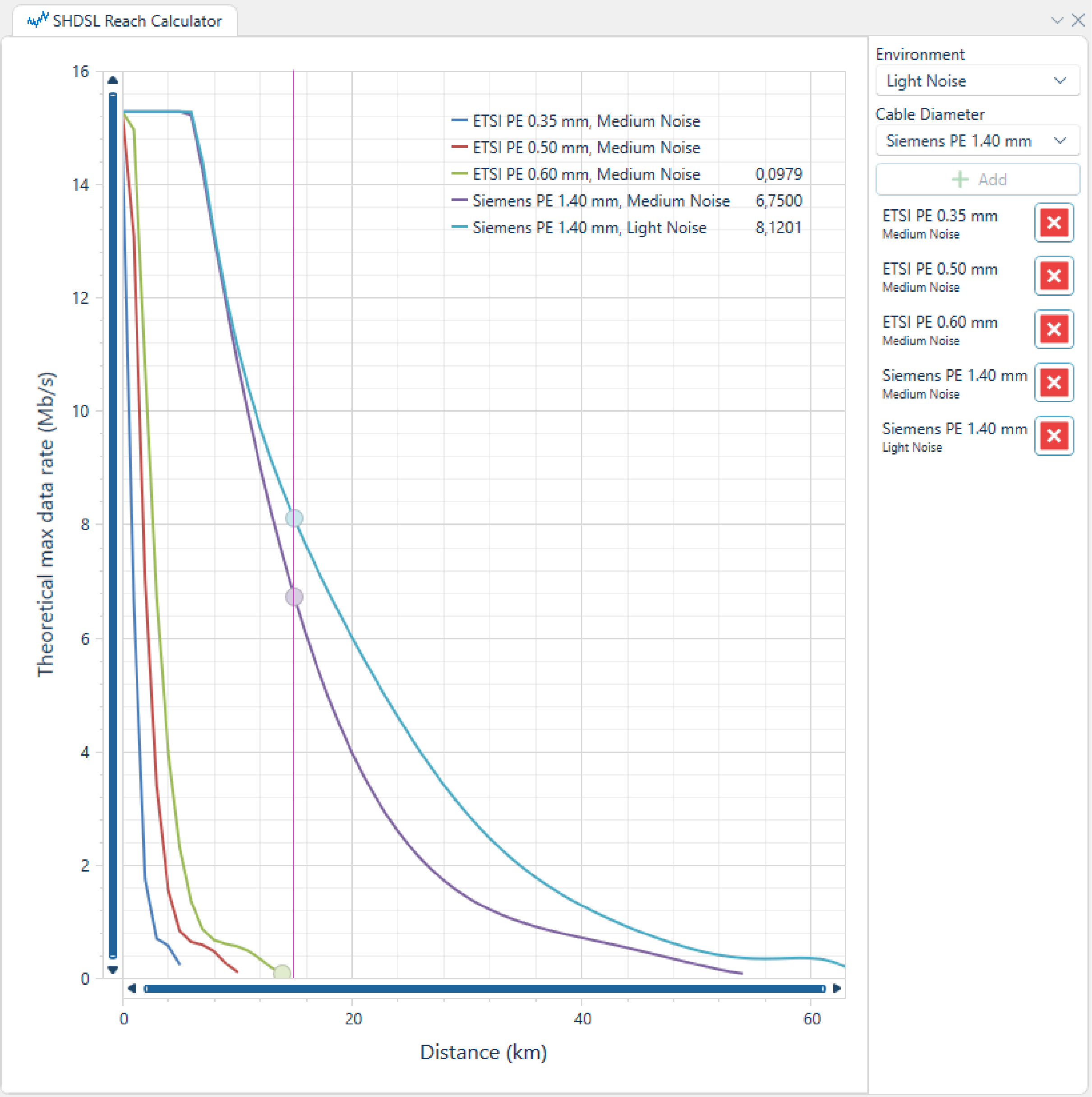
- IP address and Number of wanted hosts/net
- IP range
- IP address and a given netmask.

In order to utilzie this panel, select the calculation variant you desire to use from the radiobutton list at the top of the panel, fill in the nessecary 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)

4.3.9.3. SHDSL Reach Calculator


SHDSL Reach Calculator
16
14
12
10
8
6
4
2
0
Theoretical max data rate (Mb/s)
0
20
40
60
Distance (km)
ETSI PE 0.35 mm, Medium Noise
ETSI PE 0.50 mm, Medium Noise
ETSI PE 0.60 mm, Medium Noise 0,0979
Siemens PE 1.40 mm, Medium Noise 6,7500
Siemens PE 1.40 mm, Light Noise 8,1201
+ Add

Environment
Light Noise

Cable Diameter
Siemens PE 1.40 mm

ETSI PE 0.35 mm
Medium Noise ☒

ETSI PE 0.50 mm
Medium Noise ☒

ETSI PE 0.60 mm
Medium Noise ☒

Siemens PE 1.40 mm
Medium Noise ☒

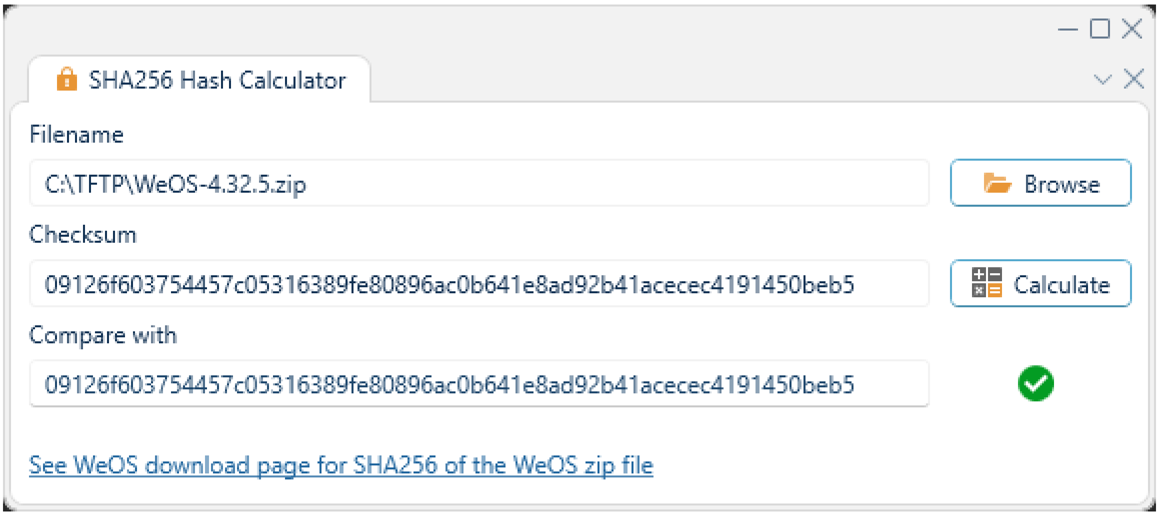
Siemens PE 1.40 mm
Light Noise ☒

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 ‘X’ button below the dropdown menus and add button.

4.3.9.4. SHA256 Hash Calculator



Filename: C:\TFTP\WeOS-4.32.5.zip

Checksum: 09126f603754457c05316389fe80896ac0b641e8ad92b41acecec4191450beb5

Compare with: 09126f603754457c05316389fe80896ac0b641e8ad92b41acecec4191450beb5 ✓

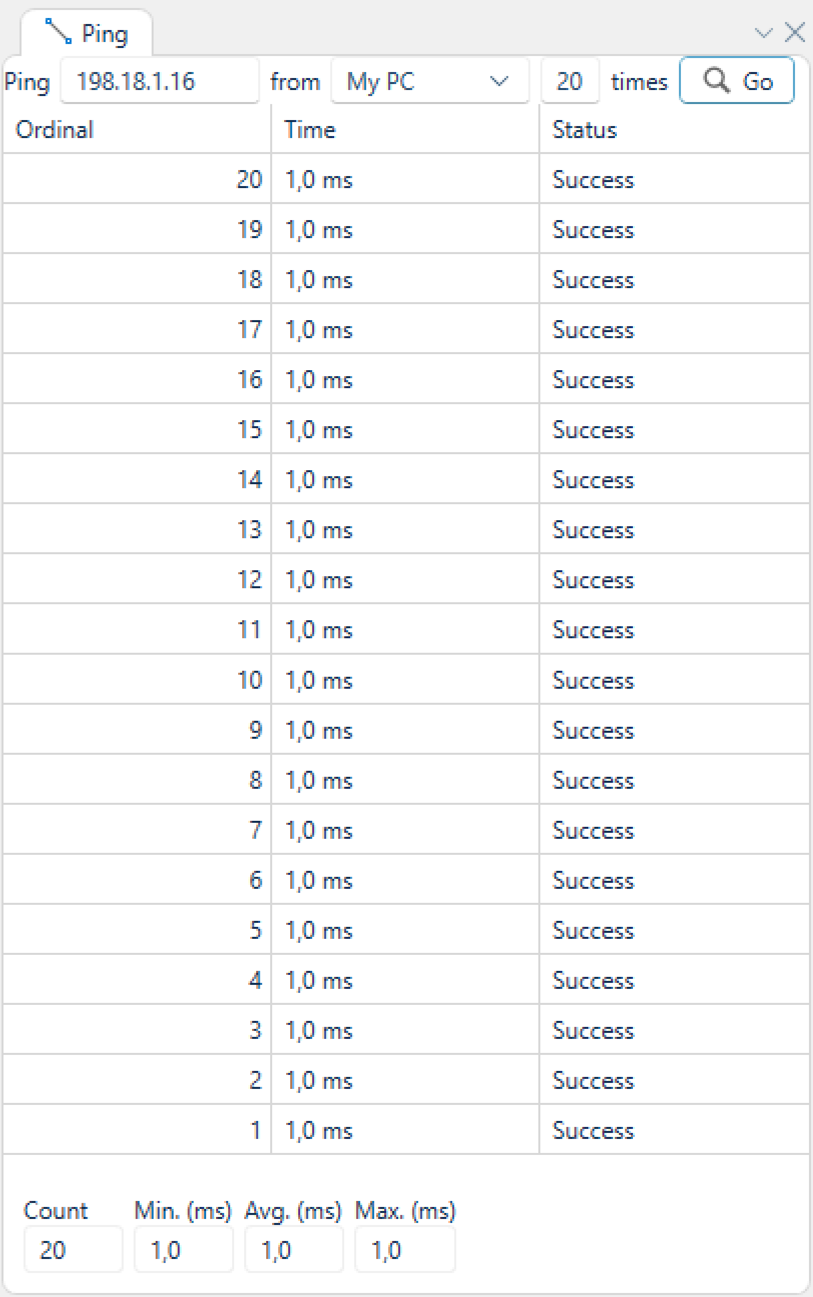
See WeOS download page for SHA256 of the WeOS zip file

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-4.32.5.zip` yields a hash starting with `09126f....`

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.

4.3.9.5. Ping



Ordinal	Time	Status
20	1,0 ms	Success
19	1,0 ms	Success
18	1,0 ms	Success
17	1,0 ms	Success
16	1,0 ms	Success
15	1,0 ms	Success
14	1,0 ms	Success
13	1,0 ms	Success
12	1,0 ms	Success
11	1,0 ms	Success
10	1,0 ms	Success
9	1,0 ms	Success
8	1,0 ms	Success
7	1,0 ms	Success
6	1,0 ms	Success
5	1,0 ms	Success
4	1,0 ms	Success
3	1,0 ms	Success
2	1,0 ms	Success
1	1,0 ms	Success

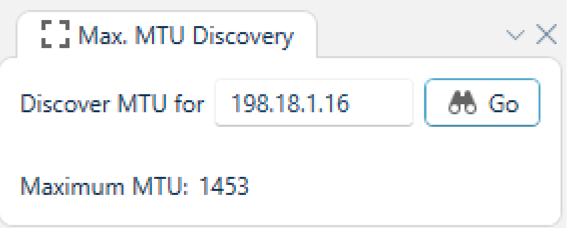
Count	Min. (ms)	Avg. (ms)	Max. (ms)
20	1,0	1,0	1,0

The Ping panel allows the user to ping a device from another device. In order to execute a ping, fill in the target address 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.

4.3.9.6. Maximum MTU Discovery



Discover MTU for 198.18.1.16

Maximum MTU: 1453

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 into the indicated box and press “Go”

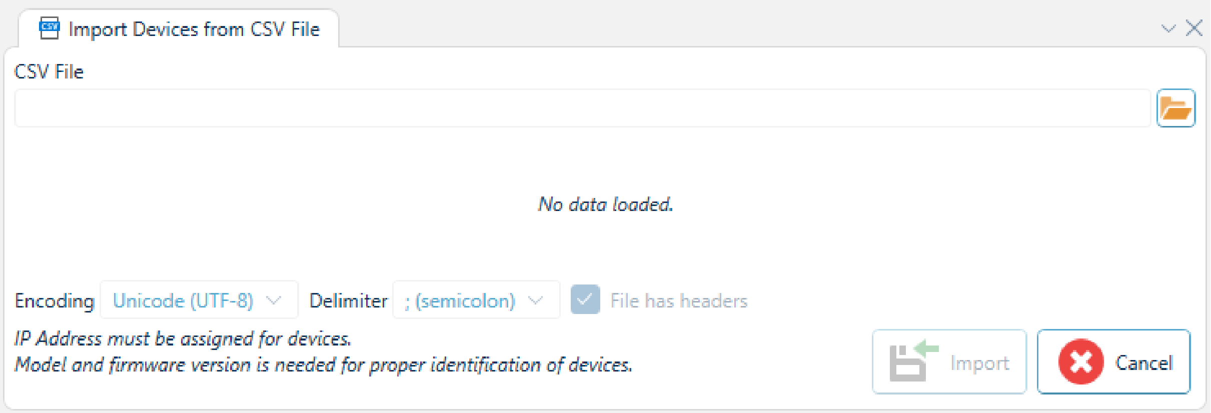
The maximum MTU size will then be displayed below, as seen in the example above.

4.3.10. Project Template

4.3.10.1. Import Devices from CSV File

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.



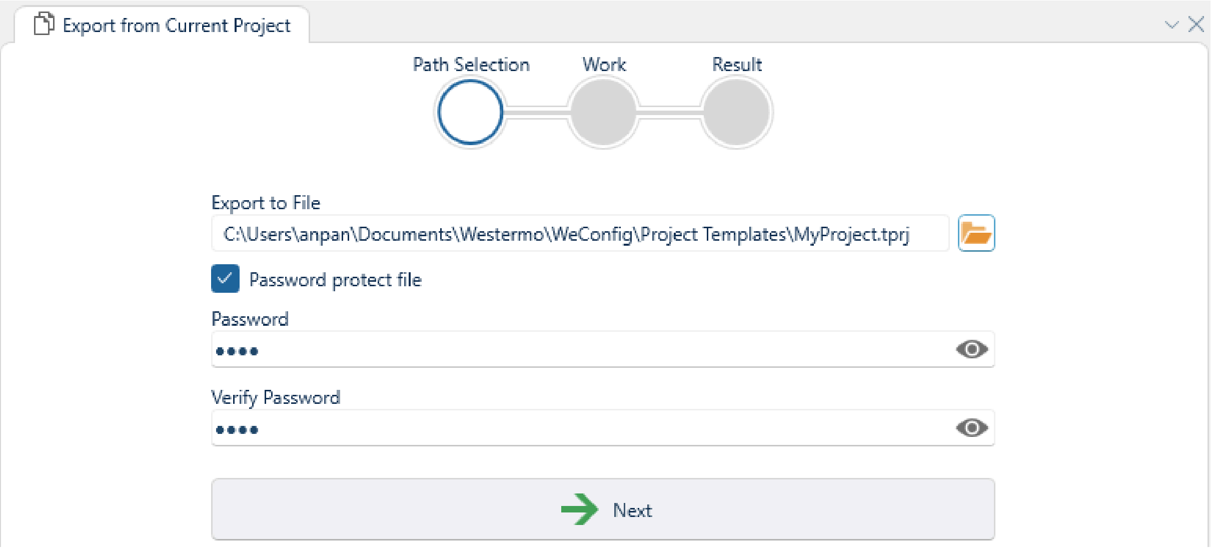
Encoding Unicode (UTF-8) Delimiter ; (semicolon) ☑ File has headers

IP Address must be assigned for devices.
Model and firmware version is needed for proper identification of devices.

4.3.10.2. Export from Current Project

Export from Current Project allows for exporting the current project to a Gold File.

Path Selection



Export to File: C:\Users\anpan\Documents\Westermo\WeConfig\Project Templates\MyProject.tprj

☑ Password protect file

Password: ••••

Verify Password: ••••

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.

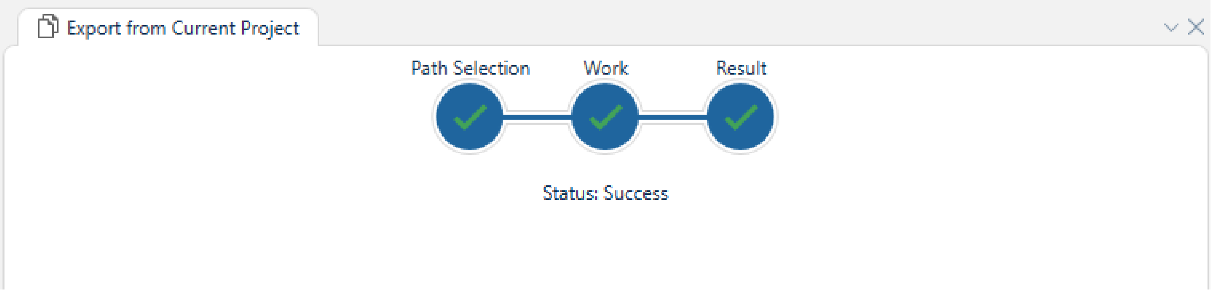
Note

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



Status: Success

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.

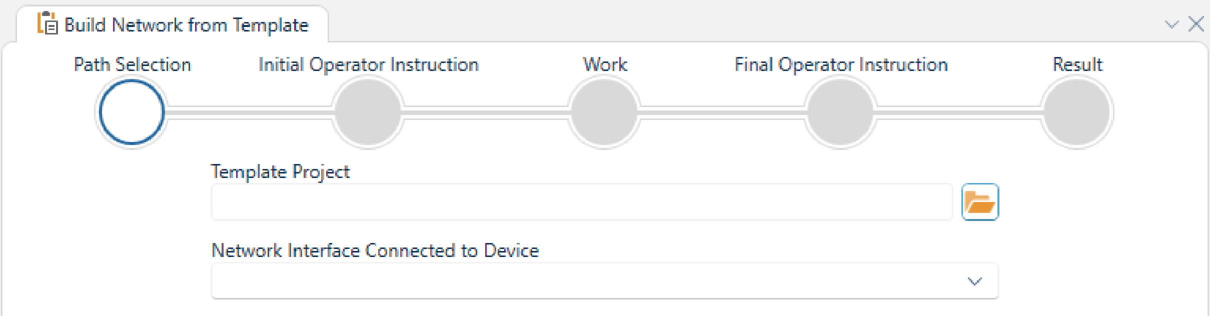
> Note
>
> The export process may take some time, depending on the size of the project and the number of devices in the topology.

4.3.10.3. Build Network from Template

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 at least equally many unused IP Addresses as there are devices in the network. This is because the import process will provide temporary IP Addresses to the devices in the network, and then change them to the correct IP Address after the import is complete.
>
> If you have 10 devices, there need to be at least 10 unused IP Addresses (for a total of at least 20).

Interface Components

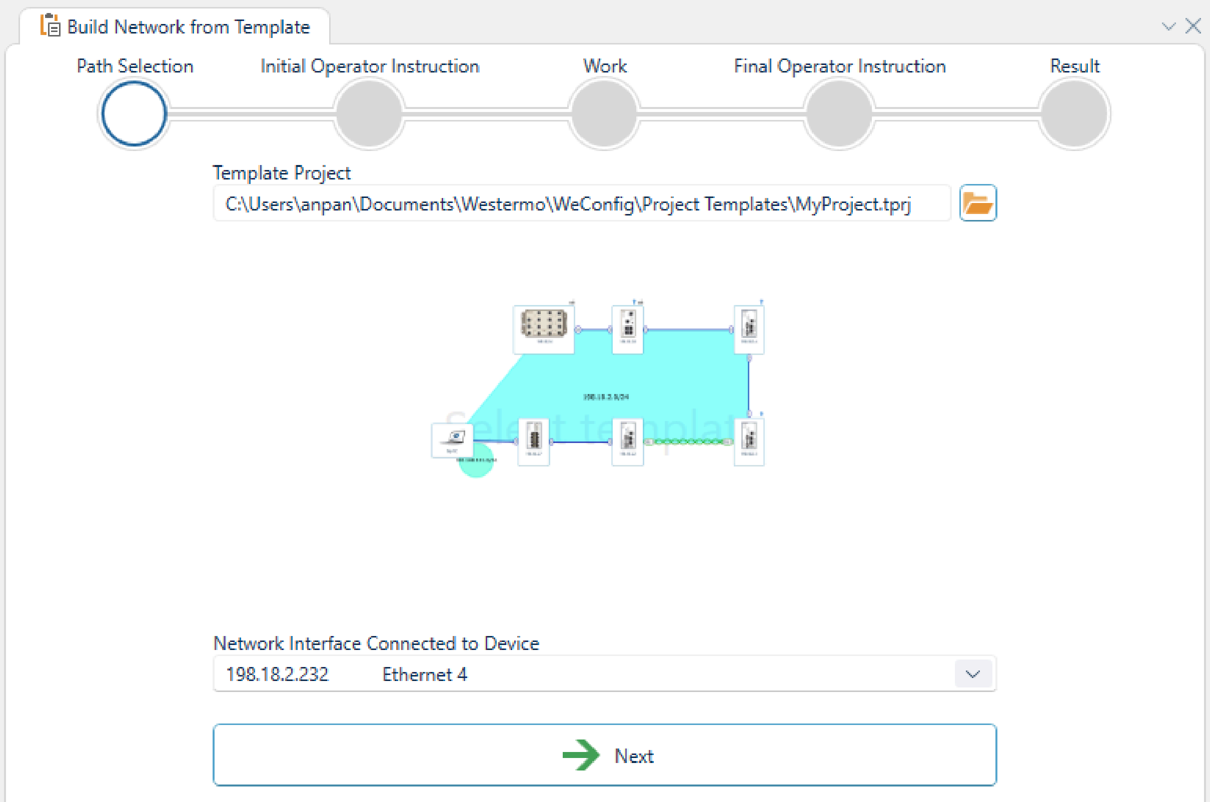

Build Network from Template
Path Selection
Initial Operator Instruction
Work
Final Operator Instruction
Result

Template Project

Network Interface Connected to Device

The build-network-process is wizard-based. It will guide and inform what to do to complete the operation.

Path Selection


Build Network from Template
Path Selection
Initial Operator Instruction
Work
Final Operator Instruction
Result
198.18.2.0/24
Select template
Next

Template Project: C:\Users\anpan\Documents\Westermo\WeConfig\Project Templates\MyProject.tprj

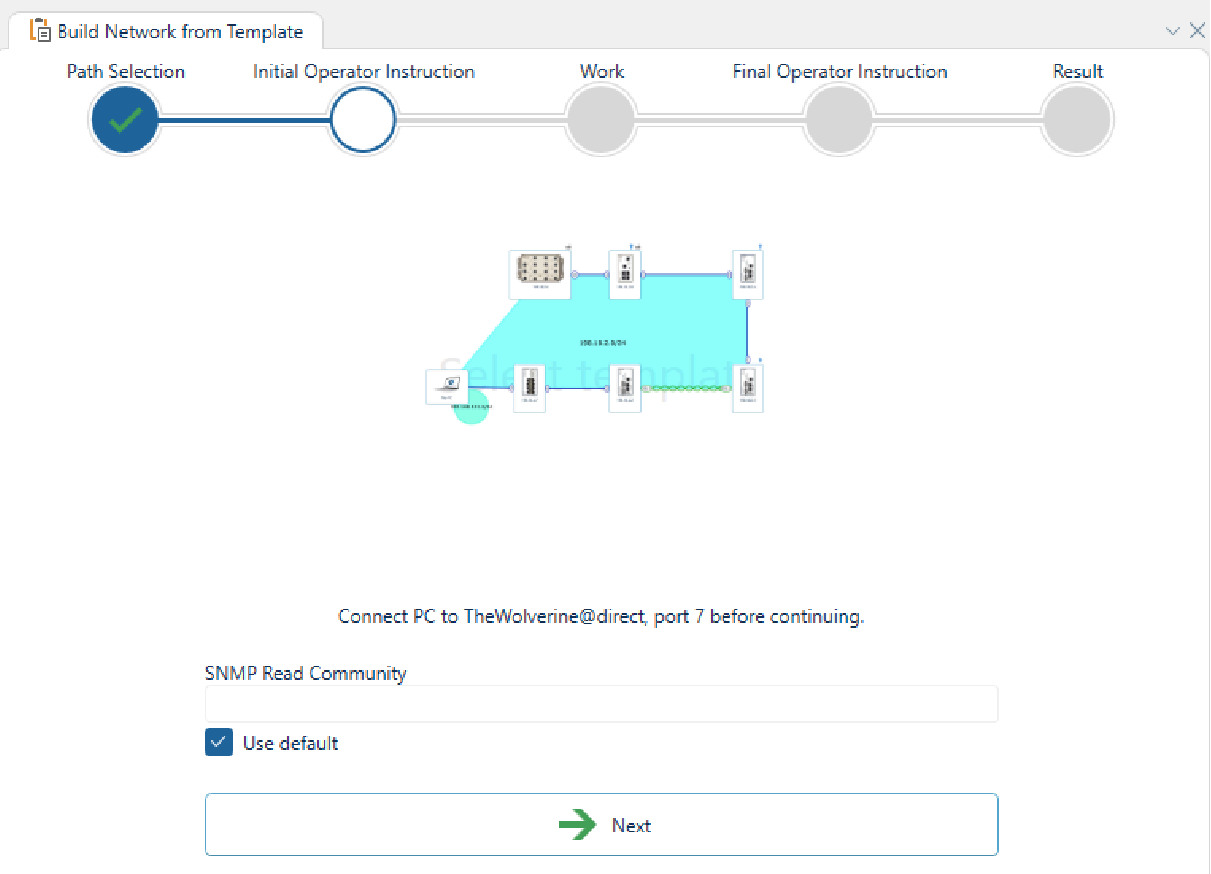
Network Interface Connected to Device: 198.18.2.232 Ethernet 4

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


Build Network from Template
Path Selection
Initial Operator Instruction
Work
Final Operator Instruction
Result
Select template
Next

Connect PC to TheWolverine@direct, port 7 before continuing.

SNMP Read Community

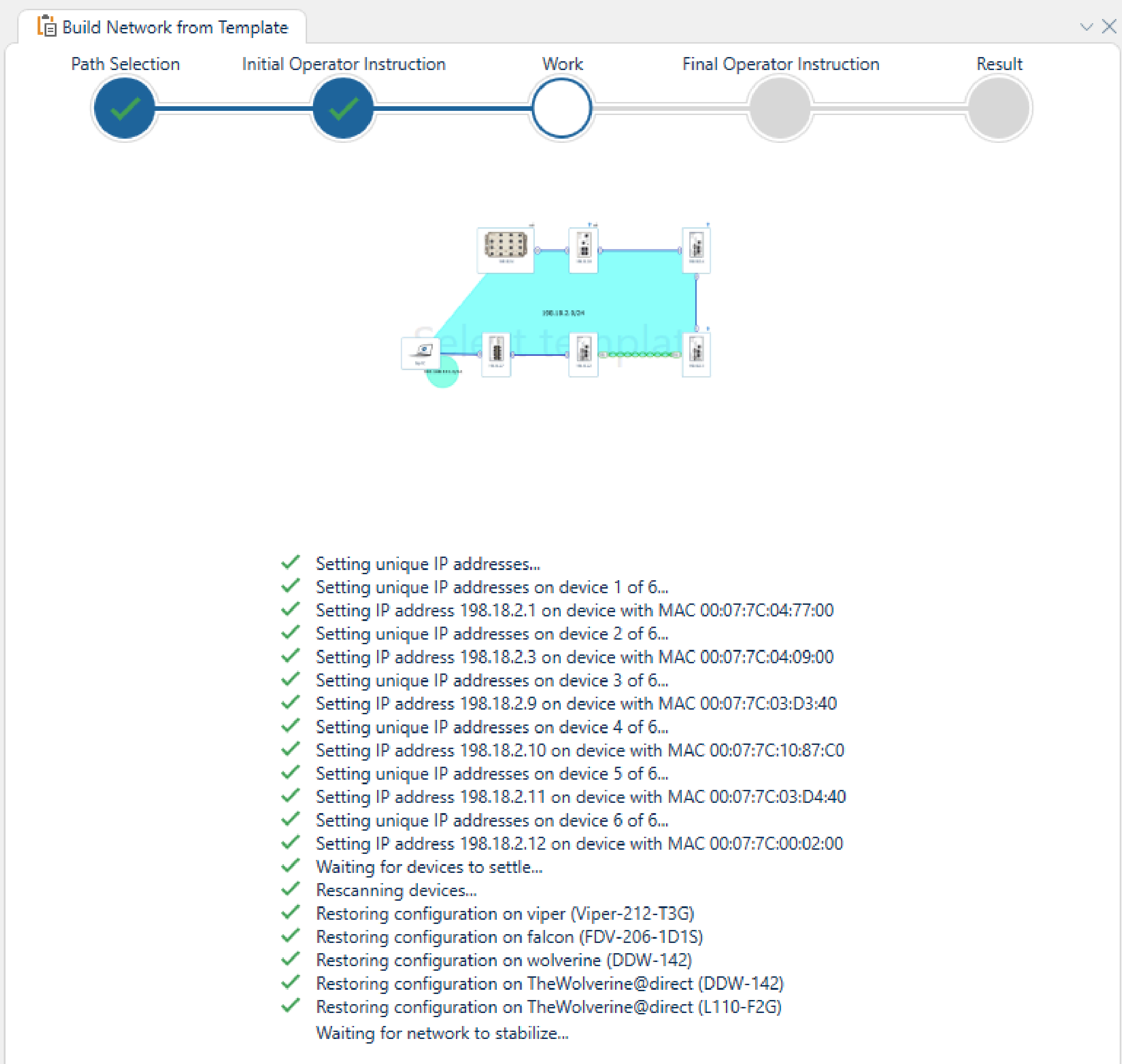
☑ Use default

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


Build Network from Template
Path Selection
Initial Operator Instruction
Work
Final Operator Instruction
Result

```
✓ Setting unique IP addresses...
✓ Setting unique IP addresses on device 1 of 6...
✓ Setting IP address 198.18.2.1 on device with MAC 00:07:7C:04:77:00
✓ Setting unique IP addresses on device 2 of 6...
✓ Setting IP address 198.18.2.3 on device with MAC 00:07:7C:04:09:00
✓ Setting unique IP addresses on device 3 of 6...
✓ Setting IP address 198.18.2.9 on device with MAC 00:07:7C:03:D3:40
✓ Setting unique IP addresses on device 4 of 6...
✓ Setting IP address 198.18.2.10 on device with MAC 00:07:7C:10:87:C0
✓ Setting unique IP addresses on device 5 of 6...
✓ Setting IP address 198.18.2.11 on device with MAC 00:07:7C:03:D4:40
✓ Setting unique IP addresses on device 6 of 6...
✓ Setting IP address 198.18.2.12 on device with MAC 00:07:7C:00:02:00
✓ Waiting for devices to settle...
✓ Rescanning devices...
✓ Restoring configuration on viper (Viper-212-T3G)
✓ Restoring configuration on falcon (FDV-206-1D1S)
✓ Restoring configuration on wolverine (DDW-142)
✓ Restoring configuration on TheWolverine@direct (DDW-142)
✓ Restoring configuration on TheWolverine@direct (L110-F2G)
  Waiting for network to stabilize...
```

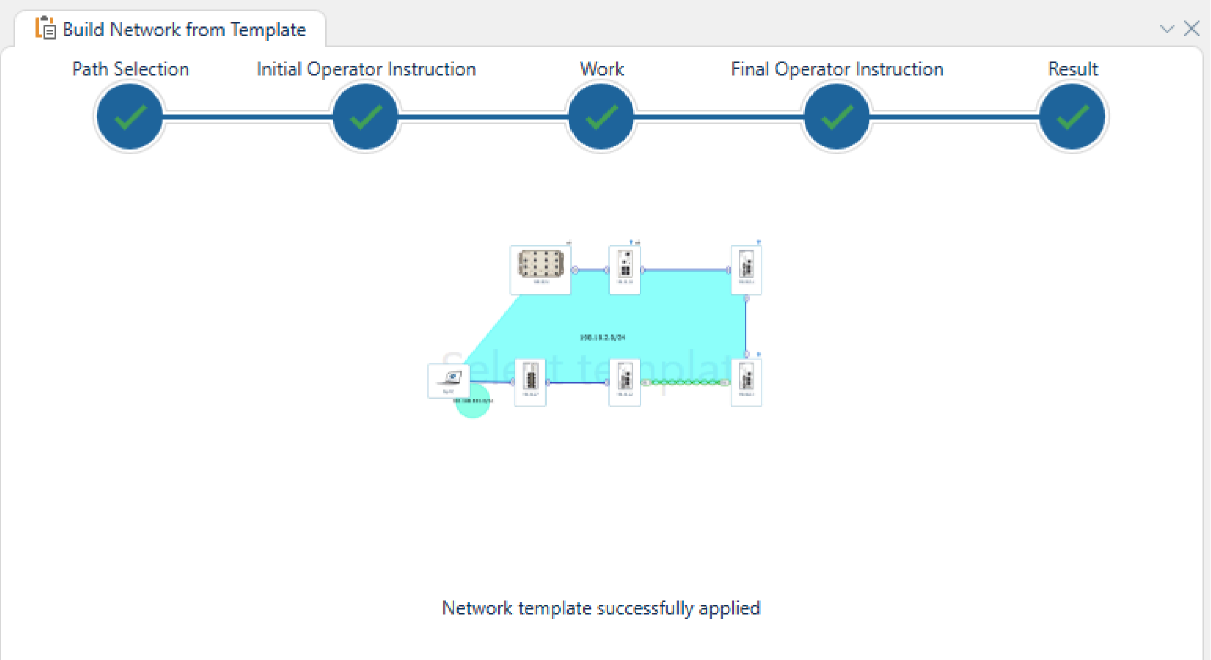
The work step will show the progress of the import process.

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


Build Network from Template
Path Selection
Initial Operator Instruction
Work
Final Operator Instruction
Result

Network template successfully applied

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.

4.3.11. 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.

4.4. Backstage Menu

The backstage menu contains a variety of settings and options for WeConfig:

Project 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
Project Password		Sets the password to encrypt the project file with, if not set, the project file will be unecrypted

Application Commands

Command	Hotkey	Explanation
Create portable installation		Generate a portable installation of WeConfig
About		Navigates to the about menu.
Release Notes		Opens the latest release notes
User's Guide		Opens this user guide
Support		Creates a support file to help developers identify the source of encountered issues
Settings		Navigates to the application settings menu.
Reset Layout		Restores the document panel to a default, empty state.
Exit	Alt-F4	Exits WeConfig

4.4.1. About

This menu contains additional information about WeConfig, it is divided into four sections, the about page, the licensing information, the EULA and the synchronized data status.

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.

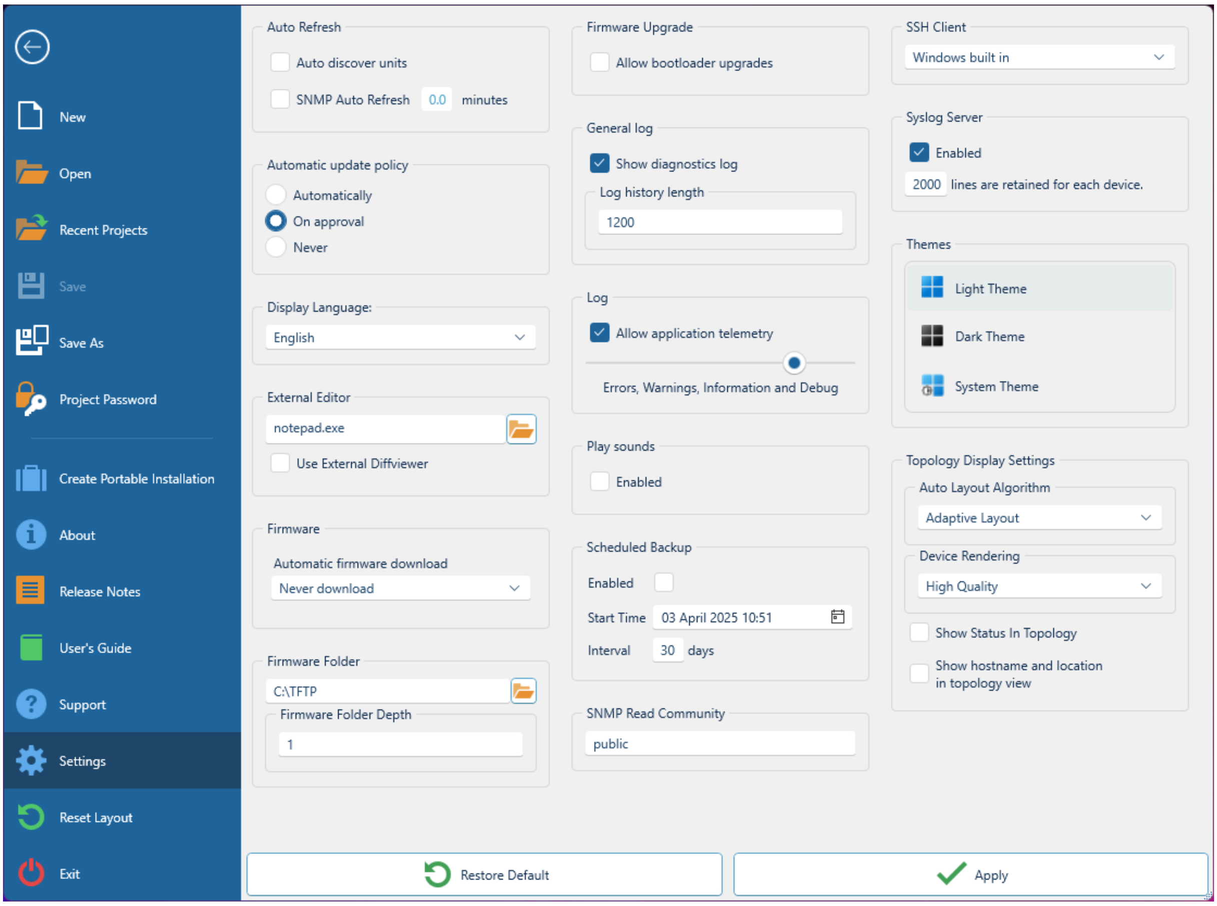
EULA

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.

4.4.2. Application Settings



Auto Refresh
☐ Auto discover units
☐ SNMP Auto Refresh 0.0 minutes

Automatic update policy
○ Automatically
● On approval
○ Never

Display Language: English

External Editor: notepad.exe
☐ Use External Diffviewer

Firmware
Automatic firmware download: Never download

Firmware Folder: C:\TFTP
Firmware Folder Depth: 1

Firmware Upgrade
☐ Allow bootloader upgrades

General log
☑ Show diagnostics log
Log history length: 1200

Log
☑ Allow application telemetry
Errors, Warnings, Information and Debug

Play sounds
☐ Enabled

Scheduled Backup
Enabled ☐
Start Time 03 April 2025 10:51
Interval 30 days

SNMP Read Community: public

SSH Client: Windows built in

Syslog Server
☑ Enabled
2000 lines are retained for each device.

Themes
Light Theme
Dark Theme
System Theme

Topology Display Settings
Auto Layout Algorithm: Adaptive Layout
Device Rendering: High Quality
☐ Show Status In Topology
☐ Show hostname and location in topology view

This menu 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. This menu also contains theme selection to the right hand side of the interface, where the following themes are available:

Theme	Explanation
Light theme	A light mode theme with brighter colors
Dark Theme	A dark mode theme with darker colors
System Theme	A theme that becomes either the light or dark theme depending on the operating system preferences

The other settings are grouped as follows, with any group that has a pending change being highlighted in the interface:

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 periodically run interface based device discovery on connected network interfaces.

Option	Description
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.

Display Language
This setting allows the user to change the user interface language. Supported languages:

- English (default)
- German
- French
- Chinese

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. By default, `notepad.exe` is used.
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 Windows user. The second is the Firmware folder depth

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 log files on the harddrive (can be found in `%USERPROFILE%\AppData\Local\Westermo\WeConfig\Logs`) and as application telemetry, if enabled via the checkbox in this group. The amount of information being logged there 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.

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
This freeform textbox contains the fallback SNMP read community that will be used towards devices where no specific SNMP settings have been configured in device access.

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 Windows. 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 recieve 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.

Device Rendering

This setting instructs WeConfig how much CPU usage it may spend on rendering device images in the topology view. The following options are available

Option	Explanation
High Quality (default)	Render device images in the best possible quality
Linear	Use linear pixel interpolation, slightly reduced quality / improved performance
Nearest-neighbor	Use nearest neighbor pixel interpolation, further reduced quality / improved performance

Show Status In Topology

When enabled, the topology view will display current operation status in the device icons in the topology view.

Show hostname and location in topology view

When enabled, the topology view will display the host name and location, when they exist, on each device node in the topology view.

4.4.3. Portable Installations

Located under the backstage 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 airgapped 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.

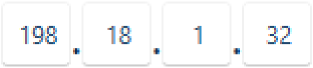
Limitations

Certain functionalities of WeConfig are limited in portable mode due to it's airgapped 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.

4.5. Components

4.5.1. IPv4 Address box



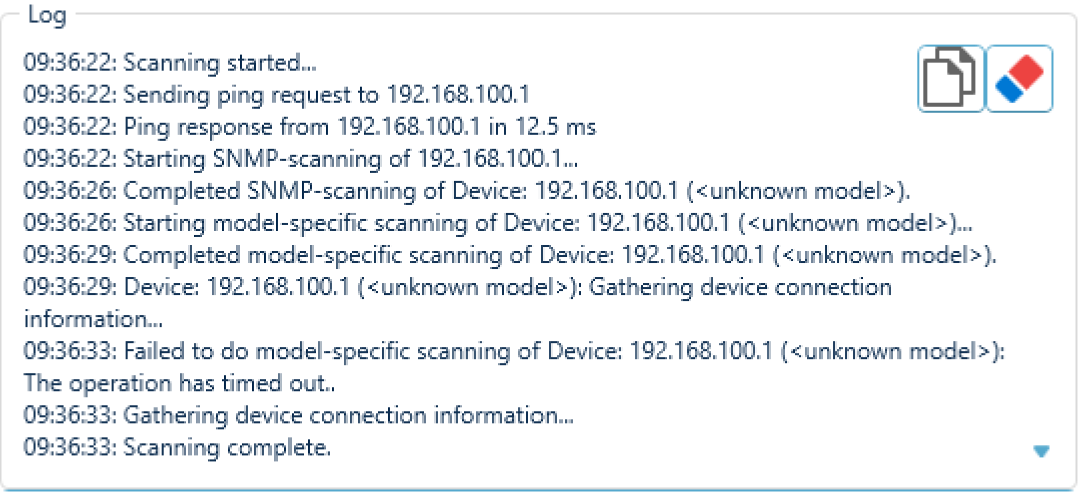
198 . 18 . 1 . 32

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 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.

4.5.2. Log box



```
09:36:22: Scanning started...
09:36:22: Sending ping request to 192.168.100.1
09:36:22: Ping response from 192.168.100.1 in 12.5 ms
09:36:22: Starting SNMP-scanning of 192.168.100.1...
09:36:26: Completed SNMP-scanning of Device: 192.168.100.1 (<unknown model>).
09:36:26: Starting model-specific scanning of Device: 192.168.100.1 (<unknown model>)...
09:36:29: Completed model-specific scanning of Device: 192.168.100.1 (<unknown model>).
09:36:29: Device: 192.168.100.1 (<unknown model>): Gathering device connection
information...
09:36:33: Failed to do model-specific scanning of Device: 192.168.100.1 (<unknown model>):
The operation has timed out..
09:36:33: Gathering device connection information...
09:36:33: Scanning complete.
```

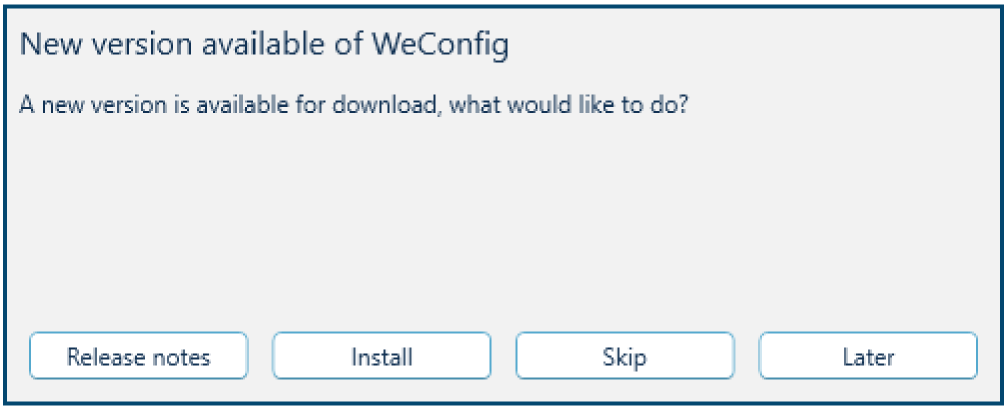
The log box is a component used in WeConfig to display logs and messages. It is designed to show real-time updates and historical logs, making it easier for users to monitor network operations that WeConfig undertakes.

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.

4.6. Notifications

WeConfig has the ability to generate toast notifications, which may appear as depicted below:



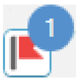
New version available of WeConfig

A new version is available for download, what would like to do?

Toasts may be dismissed by clicking the X button at it's top-left corner, but they will still be available in the notification list described below.

The toast also display any actionable link in the notification message, allowing for direct action directly in the popup.

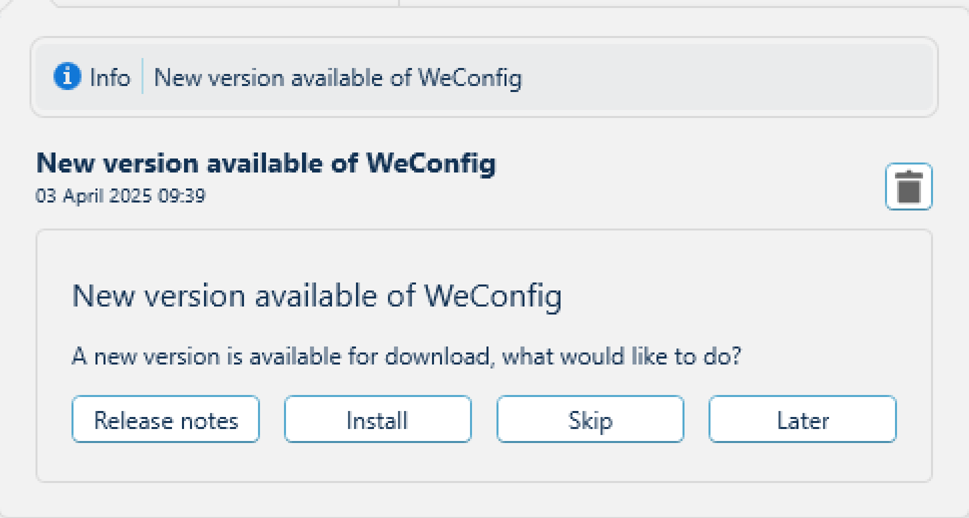
Additionally, notifications are collected under the top-level notification button, located at the top-left of the main user interface:



Where the numbered badge attached to the button indicates the current number of notifications contained within, and the flag is colored according to the highest severity of the notifications therein. Three types of notification severities are shown:

- Information / Blue
- Warning / Yellow
- Error / Red

Clicking this button will open up a flyout containing a list of active notifications, each notification can then be expanded to show more details:



New version available of WeConfig

03 April 2025 09:39

New version available of WeConfig

A new version is available for download, what would like to do?

5. Algorithms

5.1. Recursive Discovery

Recursive discovery is an algorithm for discovery used by WeConfig that is fundementally relatively simple. It's general workings are descriped 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 ICMP Ping scan with recursive discovery enabled run on `192.168.1.1` 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

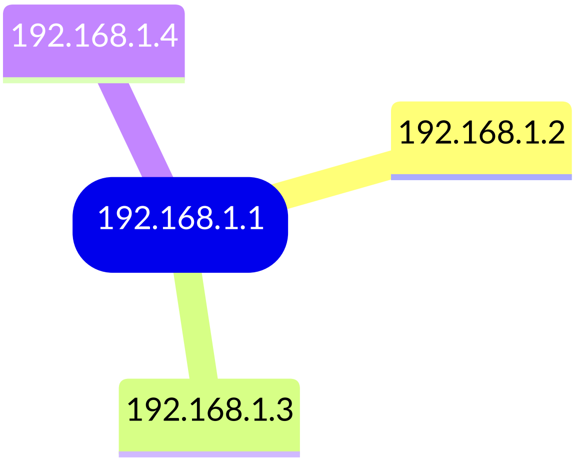
WeConfig will then inspect its LLDP table (which, for this example, will yield the address `192.168.1.2` as connected on the port `Eth 2`), its ARP table (which will produce `192.168.1.2` and `192.168.1.3`) and its route table (which will indicate it has routes going via the gateway `192.168.1.4`).

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



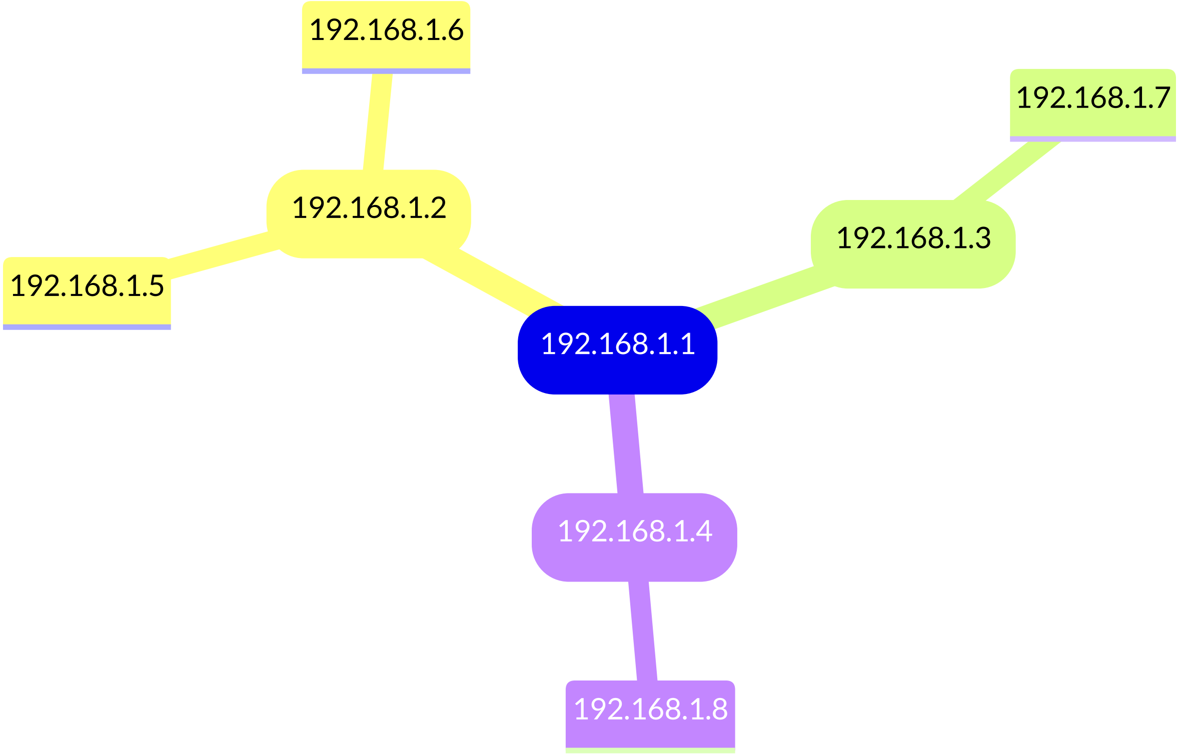
Where the rounded boxes indicate devices whose tables have been inspected.

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`)

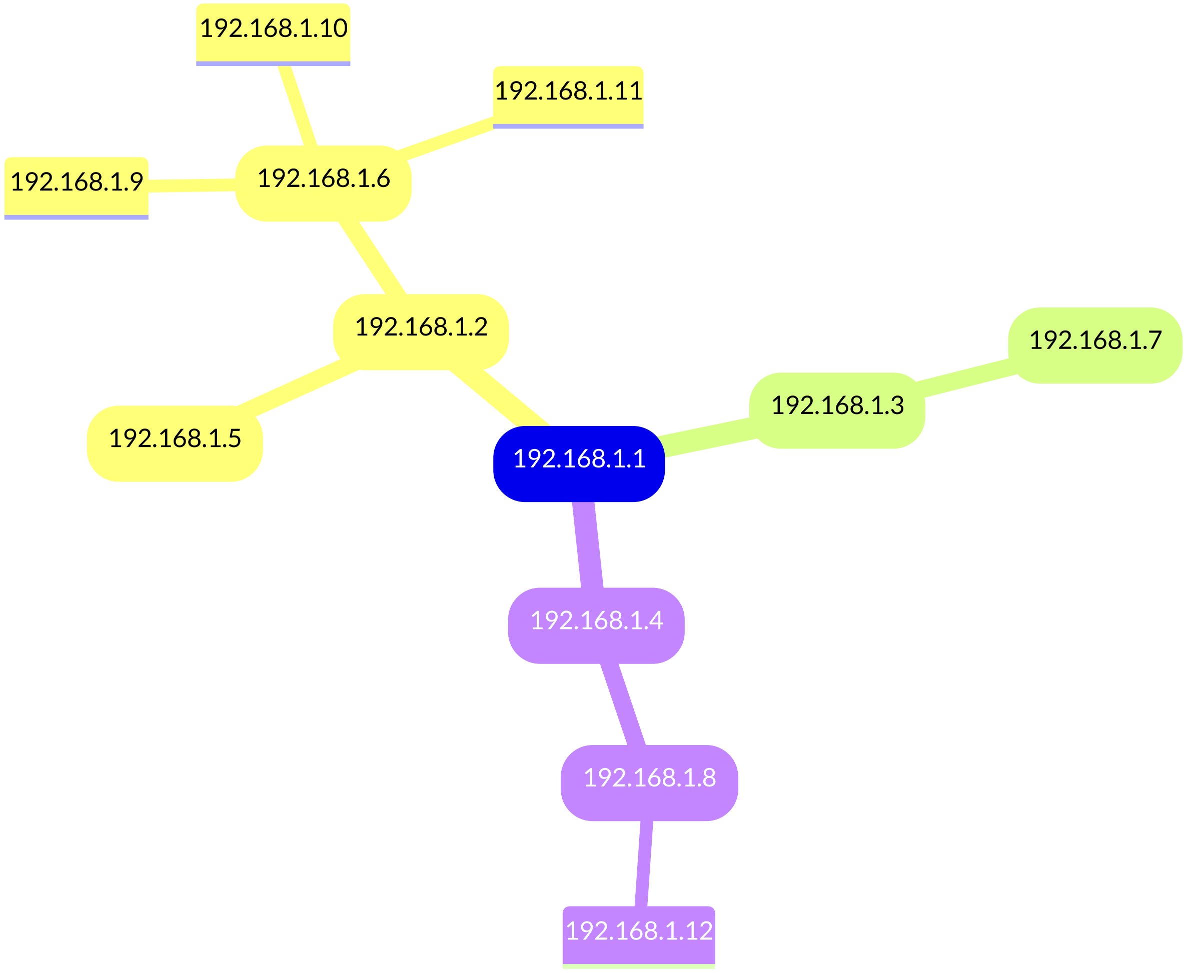
Since the three first addresses have already been found elsewhere, they will be ignored, the new ones will be scanned, and their tables checked, resulting in the network:

Step 3

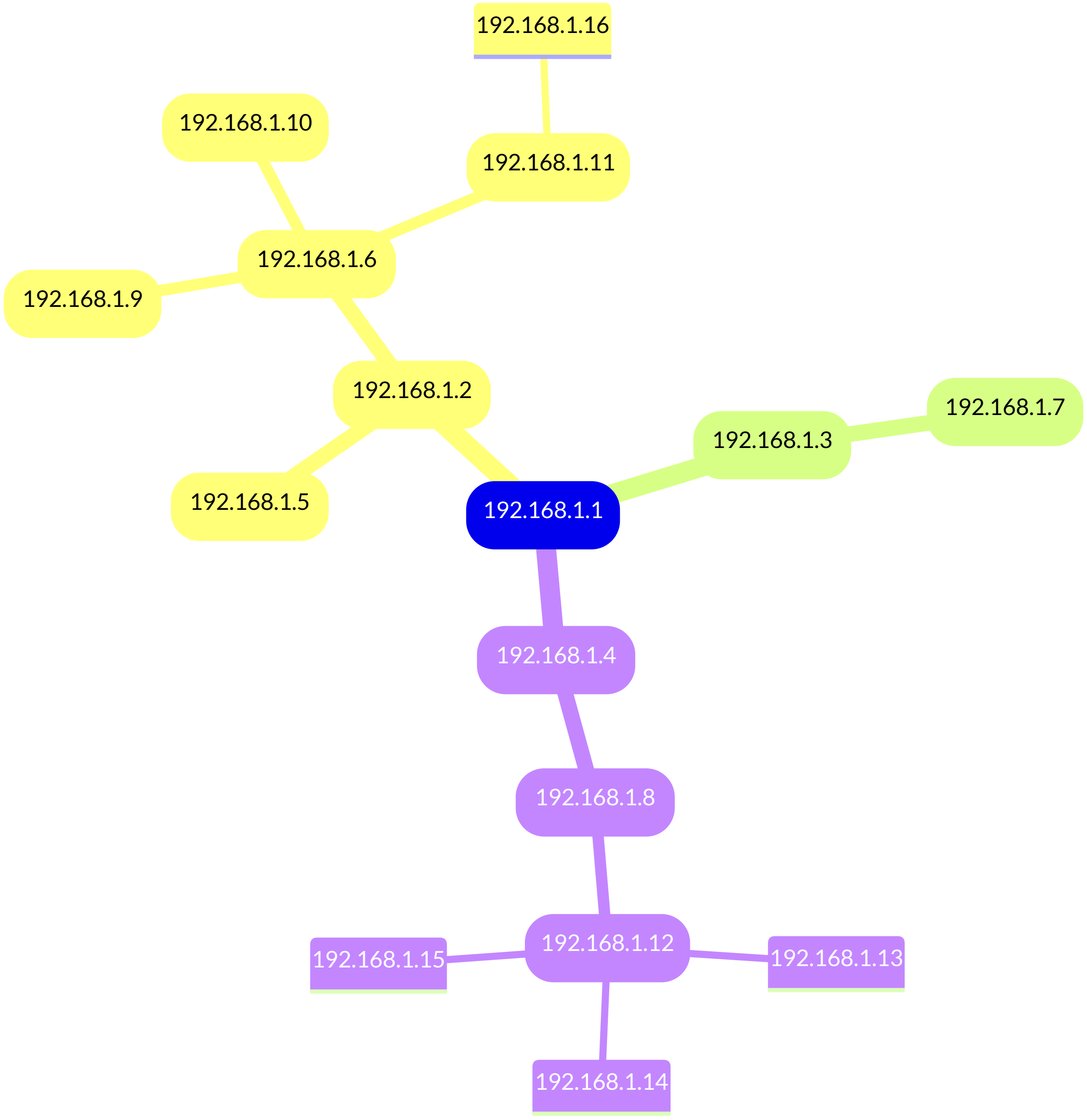


This procedure will then repeat until no new devices are found, with the operation hypothetically looking something like this.

Step 4


192.168.1.10
192.168.1.11
192.168.1.9
192.168.1.6
192.168.1.2
192.168.1.7
192.168.1.3
192.168.1.5
192.168.1.1
192.168.1.4
192.168.1.8
192.168.1.12

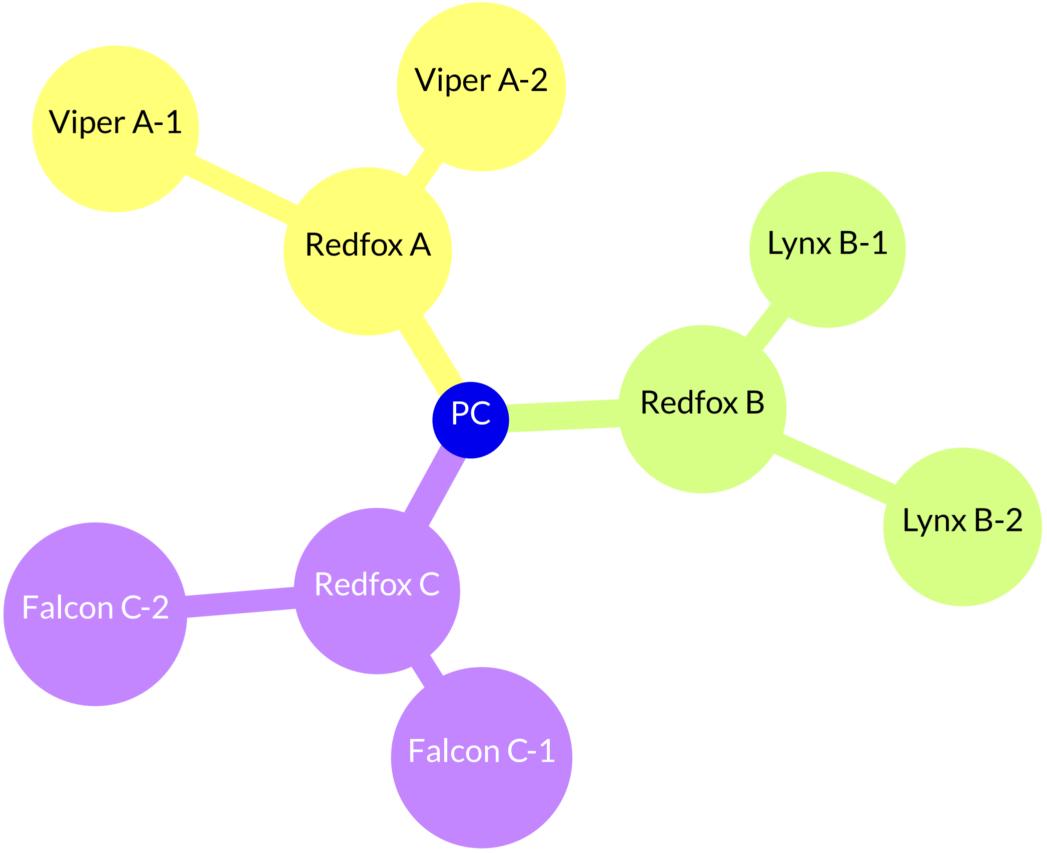
Step 5



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 being the set:

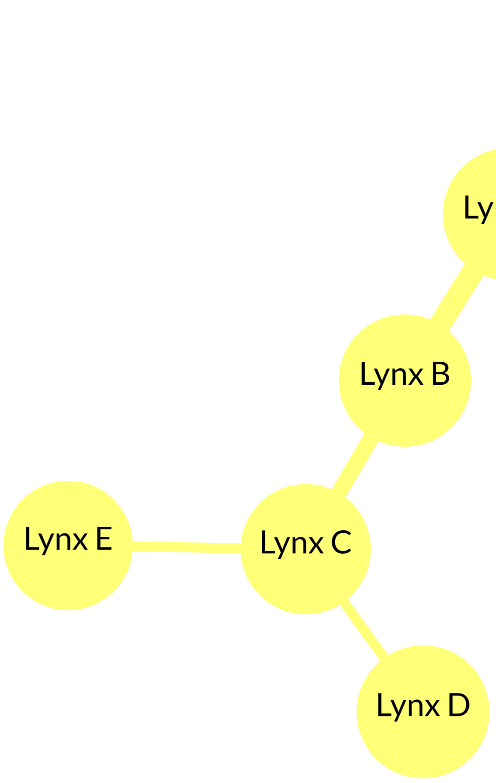
- Viper A-1
- Viper A-2
- Lynx B-1
- Lynx B-2
- Falcon C-1
- Falcon C-2

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 irregardless never a performance loss over just executing the commands one-by-one in outside-in order.

6. Changelog

6.1. WeConfig 2.0 Changelog

Updated Graphical User Interface

Dark Mode Support
- Introduced support for dark mode.

Dockable Elements
- Enabled docking for all panels and elements.

Enhanced Selection Mechanics
- Reworked the way that WeConfig handles selection, “Add” and “Clear” buttons have been removed in favor of directly matching the selection in Physical Network / Devices.

New Panels

Discovery
- Interface-Based Device Discovery: Relocated from the top bar to its own panel.
- ICMP Ping Discovery: Relocated from the top bar to its own panel.

Configuration
- Firewall: Introduced a new firewall configuration panel.
- Routing/OSPF: Licensed feature. Provides detailed OSPF configuration for devices.
- Routing/Static: Licensed feature. Allows configuration of specific static routes.
- Aggregates: Enables LACP configuration.

Per-Device Views
- Route Table: Relocated from the Selected Device section.
- Properties: Relocated from the Selected Device section.
- Attachments: Relocated from the Selected Device section.
- Configuration Files: Relocated from the Selected Device section.
- Cellular: Relocated from the Selected Device section.

Maintenance
- Clone or Replace Device: Replaces the Paste and Replace function.

Support
- Issues: Collects and displays application and device issues.

Reworked Panels

Network Visualization
- Physical Network: Includes Layer 3 visualization, updated connection colors, ribbon controls, new indicators, and new context menu options.
- Devices: Removed Backup, Bootloader, MAC, and other columns.

Configuration
- Accounts: Reworked to represent the configuration of user accounts on the device, not just admin password configuration.

- VLAN: Merged tabs into a matrix view.
- System/SNMP: Now a Staged Task.
- System/Logging: Now a Staged Task.

Maintenance

- Firmware Upgrade: Now a Staged Task.
- Device Access Settings: Allows editing of the username and PKI usage. Test connection now also tests SSH, SNMP, and HTTP(S) reachability.
- CLI Scripts: Renamed from “CLI”.

Diagnostics

- Diagnostics: Uses a new charting framework. Measurements are split per unit. Select measurements, rather than device/measurements.

Support

- Help: Now links to the user guide.
- Release Notes: Now links to the relevant changelog.
- Reports: Now a licensed feature. Is now a panel and includes a progress log.

Tools

- SHDSL Reach Calculator: Uses a new charting framework.

Project Template

- Export from Current Project: Now a Staged Task.
- Build Network from Template: Now a Staged Task.

Changelog

- Accounts (previously passwords) now have a quick-button for generating and copying a public/private key pair to the clipboard.
- “On approval” is now the default update option.
- Added “Never” as an update option.
- Added Dark mode and system theme.
- Added Discovery Neighbours as a discovery option to both ICMP and interface-based discovery.
- Added Expand/Collapse functionality to physical network.
- Added Opacity Controls to physical network.
- Added a periodic ping check that tries to reach devices in the project.
- Added a search box to physical network.
- Added actual toast for notifications.
- Added an icon to physical network that indicates that a firmware upgrade is available for the device.
- Added backstage menu..——————————————————
- Added context menu options for RIP, OSPF, and static routes.
- Added favorites list.
- Added firewall configuration panel.
- Added full docking layout handling.
- Added icons to all panels.
- Added observable ingress/egress rates to Configuration/Ports/Ethernet.
- Added read support for HSR-PRP and redbox.

- Added spinners to clickable links.
- Added static route configuration.
- Added licenses for unlocking/locking features, with an option for temporary trial licenses built into the software.
- Added the ability for WeConfig to proxy SSH connections via WeOS devices.
- Added the ability to download missing firmware directly in Firmware Upgrade.
- Added the ability to edit manually edited connections.
- Added the ability to mirror device locations across X/Y axis in physical network.
- Added the ability to remove FRNT rings.
- Basic setup now allows empty host names.
- Connection colors have been updated to match with other Westermo resources.
- Deleted Configuration Baselines panel.
- Deleted Configuration Manager panel.
- Deleted logical network.
- Device access: For devices supporting PKI authentication, PKI authentication can now be enabled or disabled.
- Device properties now contain hardware revision when applicable.
- Devices, sorting by IP address now sorts numerically rather than alphabetically.
- Diagnostics now only need to select measurements, not measurements per device.
- Discovery: When detecting an unknown device through LLDP, WeConfig will now add its known port information to the device.
- Dropped support for project files of versions lower than 3.0.
- Fixed a bug involving reading SFP diagnostics.
- Fixed a bug where 192.168.2.200 would remain configured on WeOS 4 devices after basic setup was complete.
- Fixed a bug where local firmware files occasionally would not show up in Firmware Upgrade.
- Fixed an issue with “fill” zooming where it hid the top icons on devices.
- Fixed an issue with the RedFox family occasionally being considered as incompatible with WeOS 4.33.2.
- Fixed an issue with the devices list where the Refresh context menu option was occasionally not present.
- Fixed issues with FRNT configuration.
- FRNT Configuration: Focal point is now indicated.
- FRNT Configuration: Selecting a device now selects it in the topology/devices.
- Improved performance of LLDP frame receiver.
- Improved the performance of Project file Loading and Saving.
- Improved the performance of many WeOS5 associated operations.
- Made navigation menu collapsible.
- Merged communication summary and communication details into one view.
- Merged firmware download settings and firmware download status.
- Moved Discovery into separate panels.
- Moved Project Gold file into separate panels.
- Moved Reports into a panel.
- Moved unreachable device notifications to issues.
- Moved to Velopack as installer.
- New startup screen.
- Notification badge color now reacts to the intensity of notifications.

- Prevented PC from going to sleep during a firmware upgrade sequence.
- Prevented downgrade of WeOS bootloader below barebox 2017.12.0-6.
- Rearranged VLAN editing interface.
- Reworked Firmware Upgrade into a Staged Task.
- Reworked Logging configuration into a Staged Task.
- Reworked SNMP configuration into a Staged Task.
- Rings are now rendered in physical network and can be selected / dragged.
- Reset layout button added to backstage.
- Split "Selected device" into multiple panels.
- Split "selected device" into multiple different panels.
- Subnets are now rendered in physical network and can be selected / dragged.
- Text in communication summary/details is now copyable.
- The "CLI" panel has been renamed to "CLI Scripts".
- Unified list views across many panels to have a similar look and feel.
- Updated DSL connection visualization.
- Updated General log to improve performance and appearance.
- Updated project serialization to version 3.1.
- Updated target runtime to .NET 9.
- Updated how unknown devices are rendered in the topology when their MAC is a known OUI.
- WeConfig now remembers the last selected interface if it is disconnected and then reconnected.
- WeConfig will now prompt the user for trust when attempting to send non-factory-defaulted credentials to a device with an unknown/unrecognized host key.