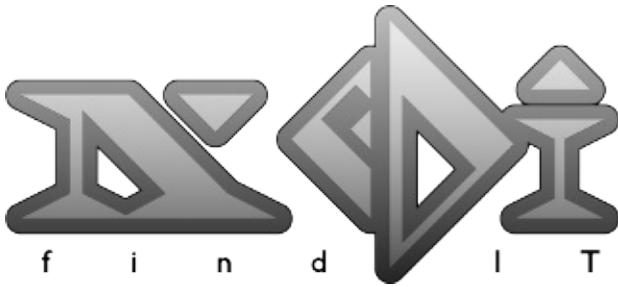


The Networkers Guide to

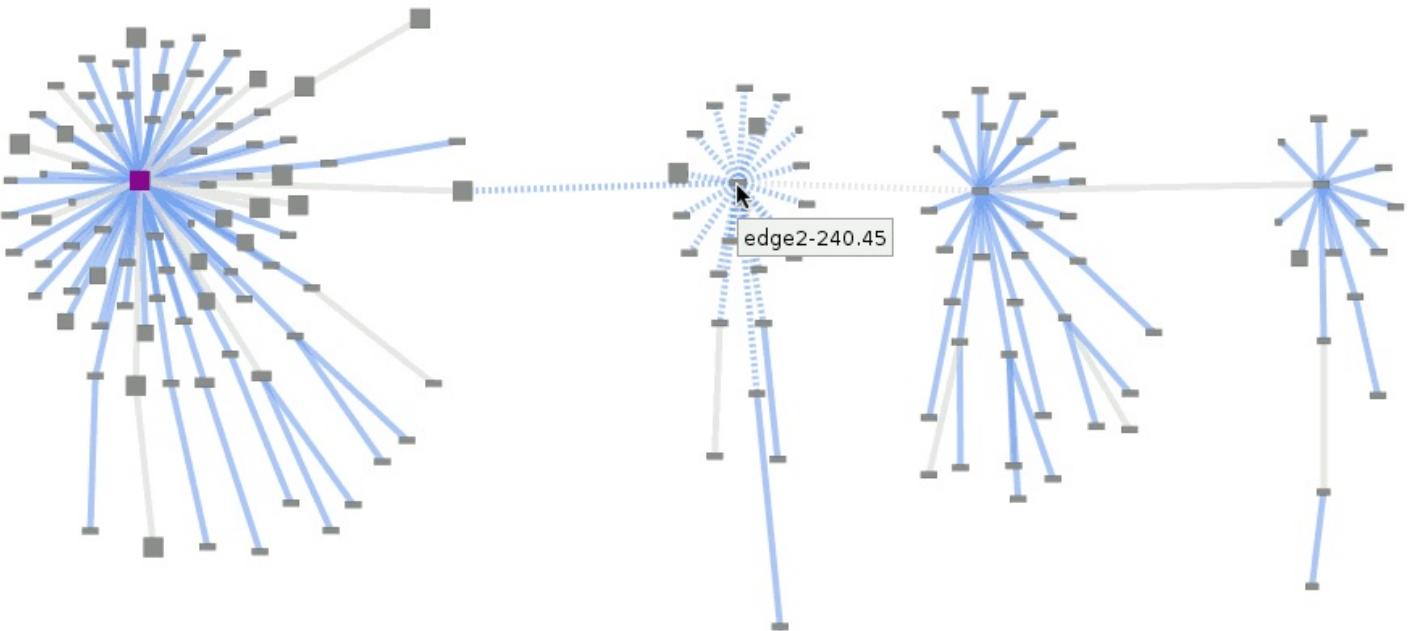


Introduction



NeDi discovers your network devices and tracks connected end-nodes. It contains many additional features for managing enterprise networks:

- Intelligent topology awareness
- MAC address mapping/tracking
- Traffic, error, discard and broadcast graphing with threshold based alerting
- Uptime, BGP peer and interface status monitoring
- Correlate syslog messages and traps with discovery events
- Network maps for documentation and monitoring dashboards
- Detect rouge access points and find missing devices
- Extensive reporting ranging from devices, modules, interfaces all the way to assets and nodes



NeDi's modular architecture allows for simple integration with other tools. For example Cacti graphs can be created, based on discovered information. Due to NeDi's versatility things like printer resources can be monitored as well...

Published on Sat Oct 14 13:45:05 2017

Installation Instructions

NeDi's website provides all necessary information for a successful installation.

The generic procedure with some links to external documentation:

<http://www.nedi.ch/installation>

OS Specific information:

<http://www.nedi.ch/installation/freebsd>

<http://www.nedi.ch/installation/os-x>

<http://www.nedi.ch/installation/suse-installation>

NeDi Appliance

There's a free OpenBSD based appliance called NeDiO14 available on the [Download](#) page. It will be succeeded by a Debian based OVA called NeDian17.

Partner Solutions

NeDi is integrated in commercially supported solutions as well. Have a look at the partners on NeDi's Download page to get more information.

General Overview

This chapter helps to get you acquainted with NeDi:

- Architecture: A quick overview of NeDi's components
- Functional Breakdown: A description of use cases
- Terminology: Definition of topics found in NeDi

The following chapters cover NeDi use cases:

- Network Management: The original intention
- Asset Discovery: Collect details on your nodes and devices

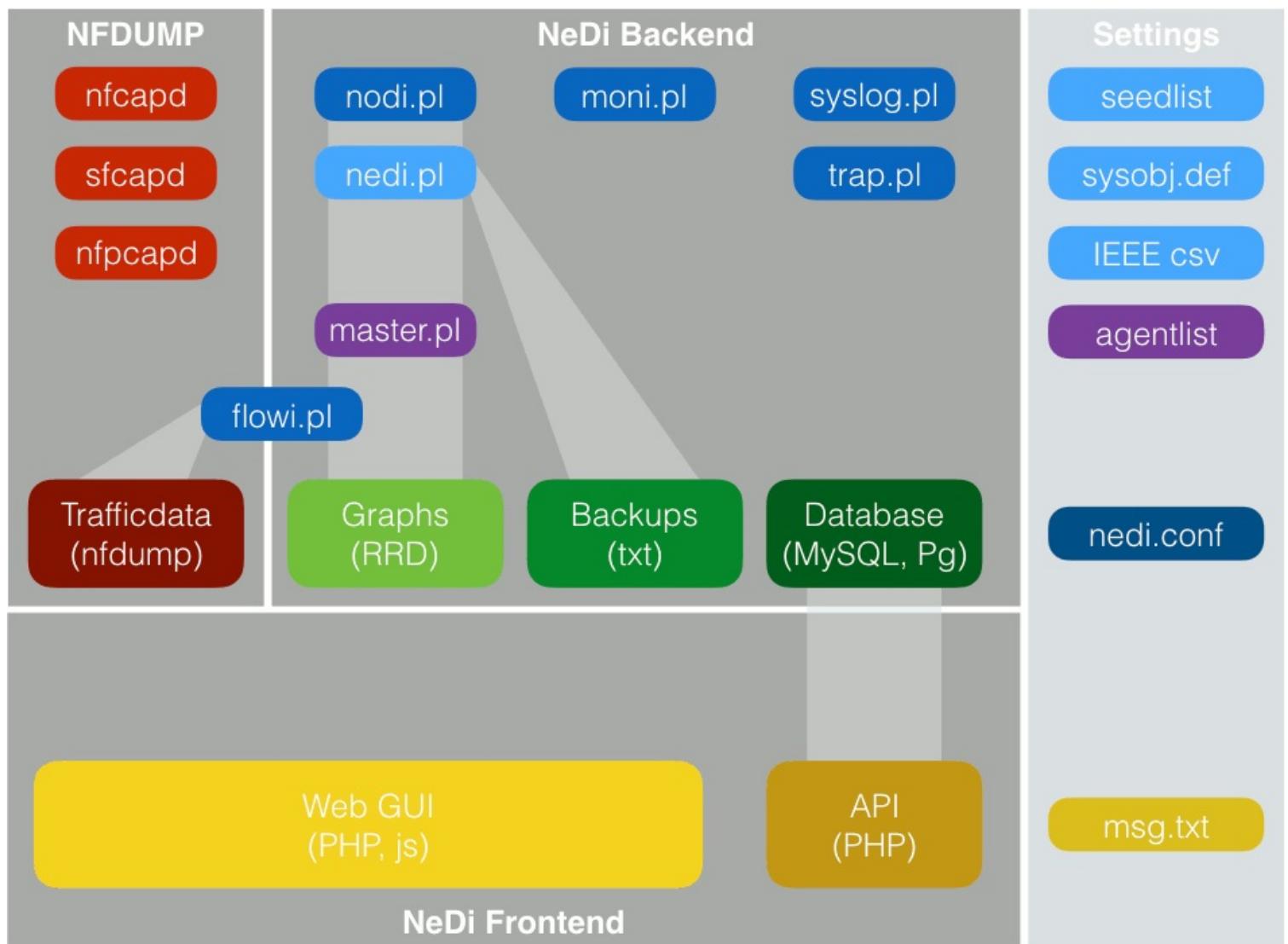
The frontend documentation is divided into the following sections:

- REST API
- GUI overview
- GUI modules

Architecture

NeDi's architecture can be divided into the following components:

- Network discovery (nedi.pl) in light blue above
- Monitoring (moni.pl, trap.pl and syslog.pl) in blue
- Master daemon and agentlist for centralizing distributed NeDi instances, in purple
- Node discovery for asset details (gathered by nodi.pl using WMI and SSH) in blue
- Modular web frontend written in PHP and some javascript in yellow
- Restful API frontend written in PHP in dark yellow
- Master settings file (nedi.conf) in dark blue
- Dependencies are indicated above as well (e.g. API only talks to the DB and flowi.pl uses Trafficdata to generate graphs)
- NFDUMP can optionally be integrated, in that the frontend can access and display netflow data



Functional Breakdown

Use this table in order to learn how the components can be used to achieve different goals. Detailed information is provided in the following chapters:

Goal	Component	Description
Discover network devices using SNMP and SSH/Telnet	nedi.pl	<ul style="list-style-type: none"> Run ned.pl via console, System-NeDi in the web GUI or use crontab to discover on fixed intervals This will also track MAC and IP addresses and collect the interface statistics
Monitor discovered network devices	moni.pl	<ul style="list-style-type: none"> Run moni.pl via console, System-Services in the web GUI, or have it autostart with init.d scripts Add desired devices (which usually have been discovered before) to monitoring Control the monitoring frequency (default is uptime check every 3 minutes)
Receive Syslog messages	syslog.pl	Run syslog.pl via console, System-Services in the web GUI, or have it autostart with init.d scripts
Receive SNMP traps	trap.pl	Configure trap.pl as traphandler for snmptrapd
Monitor remote NeDi hosts	master.pl	<ul style="list-style-type: none"> Add remote NeDi installations in agentlist Run master.pl via console, System-Services in the web GUI or have it autostart with initd scripts Configure how the remote agents provide their API connection (e.g. https and rootpath) Note: Don't run any other components on this host to avoid confusion
Discover assets	nodi.pl	<ul style="list-style-type: none"> Run nodi.pl via console, System-NoDi in the web GUI or use crontab to discover on fixed intervals It's recommended to use a different DB (and config file), if ned.pl is running here as well
Traffic Monitoring	nfdump, flowi.pl	<ul style="list-style-type: none"> Run nfcapd (for netflow), sfcapd (for sflow) or nfpcapd (to capture traffic on an interface) Specify path to netflow data in ned.conf Edit ned.conf to set nfdpath and the IP-ports you want to graph Run flowi.pl every 5 minutes to create the Protocol and Port graphs Make sure the frontend can execute nfdump (especially if nfdump is installed on another host and the data dir is mounted)

Terminology

Devices:

- SNMP capable network equipment, printer or server
- WMI capable Windows server or client
- SSH capable Unix (namely Linux and BSD) server or client

Modules:

- Linecards, powersupplies, fantrays or optical transceivers (usually with serial number) in network devices
- Members (usually classified as chassis) in a stack
- Virtual machines in hypervisors
- Supplies in printers
- CPU, Ram, HDD, display or installed software in WMI or SSH devices
- Go to [Modules](#) for more networking related information

Nodes:

- MAC address from a bridge-forward table on a switch (required)
- IP addresses of ARP tables on routers or layer 3 switches (optional)
- DNS names obtained by reverse lookup of IP addresses (optional)
- Go to [Network Population](#) for more networking related information

Links:

- Connection between devices stored in the links table
- Created using CDP, LLDP (ISDP under investigation)
- Calculated automatically with information derived from bridge-forward tables (MAC)
- Added statically using [Topology-Linked](#) (STAT)

Assets:

- Items with a serial number in the inventory table
- Added by NeDi's -Y option
- Added by hand using [Assets-Management](#)
- Imported via CSV file using [Assets-Management](#)

Policies:

- Rules defined in [System-Policies](#) to create alerts or actions
- They're evaluated at different points during the discovery
- Packets, Bytes or Flows rules are evaluated by flowi.pl

Age Shading

Date and time fields are usually displayed with a age based background color. This helps to quickly spot anomalies in a list for example:

- First seen dates start out to be bright green (affected by the "retire" setting in ned.conf) like a fresh fruit
- Last seen dates turn more red after time like a sunset
- The greater the difference between the two the more blue is added

Devices List

Name	Main Address	Devices Type	First Discover	Last Discover
 xerxes	10.10.10.13	 System x3250 M3 -[4251PAA]-	2.Apr 17 23:23	3.Apr 17 12:35
 new	10.10.11.2	 gen-ctrl	24.Mar 17 12:04	12.Apr 17 22:25
 ROMULUS	10.10.10.73	 System Product Name	13.Mar 17 15:00	13.Mar 17 15:35
 SEP001F6C7EDA33	10.10.10.147	 CP-7906G	5.Jan 17 16:25	5.Jan 17 16:30
 osiris	10.10.10.10	 H81M-HD3	24.Nov 16 13:46	12.Apr 17 22:00
 2520G8-P	10.10.10.4	 2520-8G-PoE	6.Nov 16 12:30	12.Apr 17 22:25
 linux-q8x4	10.10.10.164	 Standard PC (i440FX + PIIX, 1996)	2.Nov 16 22:06	2.Nov 16 22:06
 centos	10.10.10.163	 Standard PC (i440FX + PIIX, 1996)	31.Oct 16 12:21	31.Oct 16 15:22
 Summit200-48	10.10.10.248	 Summit200-48	13.Oct 16 11:47	13.Oct 16 13:40
 MyX450e-48p	10.10.10.47	 SummitX450e-48p	13.Oct 16 11:46	13.Oct 16 13:40

10 Devices, Sort: firstdis desc, Limit: 10

Colors quickly show new devices and those being offline for a while

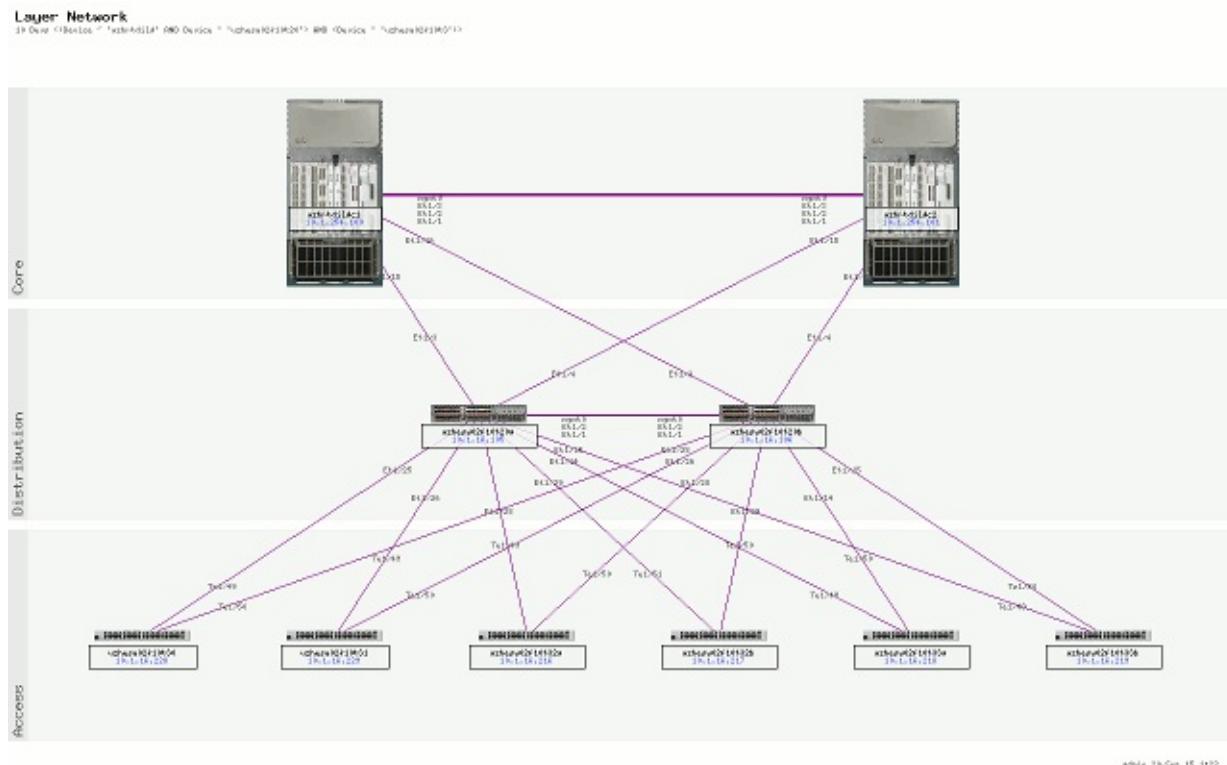
Network Management

Prerequisites

Before you begin, decide how you will use NeDi, and what you need to do to make it an effective tool in your network. The discovery nedis.pl needs SNMP read access in order to discover network devices. Privileged CLI access can be used to read the MAC address tables on Cisco devices, which is faster than vlan indexing via SNMP, and supports port security as well. The configurations are read via CLI and stored in the database or as text files. If you want to use NeDi to back up your configurations, you will need to enable privileged CLI access.

NeDi requires **unique device names**, since this is the primary key in the database. The domain part of the device name is usually discarded, because CDP is not consistent with domain names on all devices, and this can lead to problems when creating device links. Make sure you name your devices uniquely.

NeDi reaches its full potential when used with [CDP](#), FDP and/or [LLDP](#) capable devices in your network. The discovery and topology visualizations no longer depend on these protocols being enabled, however, so NeDi can be used effectively without them.



NeDi can draw maps for documentation and monitoring dashboards

Topology Awareness

If mapping your network with a clear and automated visual representation is important to you, you will want to enable the topology awareness features by preparing your devices to be placed in NeDi's visualizations and maps. NeDi is capable of visualizing your network down to rack level! In order to do this, a specific format for the SNMP location string is required on each device as follows (separators can be configured in ned.conf with locsep):

```
Region;City;Street;Floor ; [Room ; ] [Rack ; ] [RU ; ] [Height ]
```

- The building or street address may contain several sub-buildings separated with a second separator (e.g. _)
- The RU is counted upwards from the bottom of a rack
- The height is only necessary, if the device comes in different sizes (e.g. a VMware ESX server)

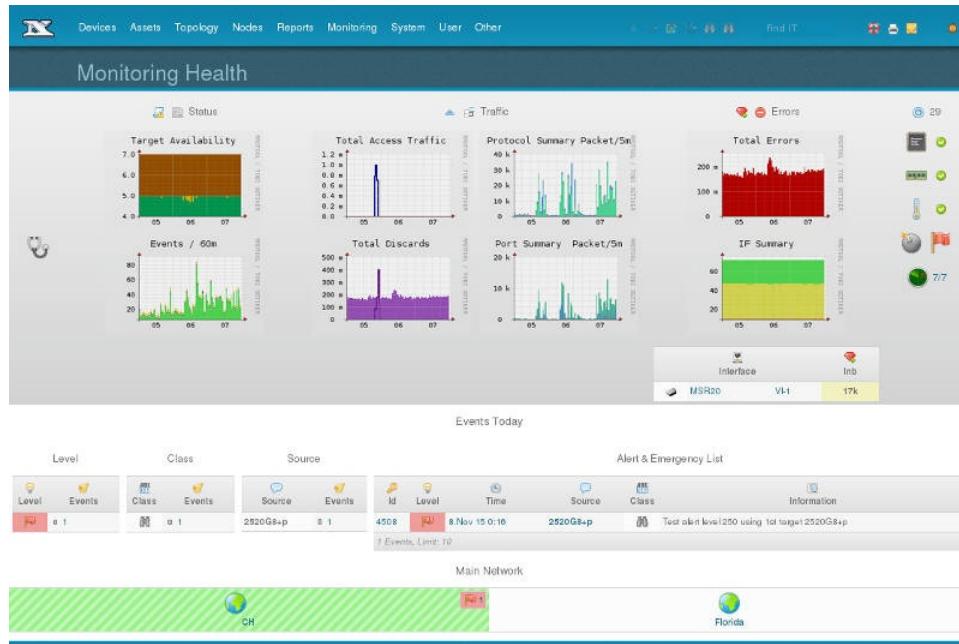
```
Switzerland;Zurich;Main Station;5;DC;Rack 17;7
```

- This example puts a device in position 7 of rack 17 in the DC room at the 5th floor

```
FL;Orlando;42 Pine St_A;54;Closet;Wallrack;1
```

- This device is located in a closet's wallrack in Building-A of 42 Pine st (there might be a building-B at the same address)

If network devices can be configured with this SNMP location scheme, NeDi can visualize your entire network topology. If it's not feasible to reconfigure all your devices, you can override locations for some of them in the seedlist, or map other information to the location scheme. You'll lose some of the dynamism of the mapping, but you can still leverage some topology features this way.



Topology aware overview in Monitoring-Health

Cities show their size based on devices:

Icon	Size	# of Devices
	small	1-2
	medium	3-9
	large	10-19
	extra large	20+

The same applies to Buildings where as important ones can be “painted” red using rebuild in ned.conf:

Icon	Size	# of Devices
	small	1-2
	medium	3-9
	large	10-19
	extra large (important)	20+

Configuration Backup

NeDi is capable of backing up switch, router and firewall configurations. Common brands and models are supported as well as some less known ones (backing up of some FW contexts needs more work). The backup is performed via CLI and corresponding "show conf" commands.

The backup can be performed in 2 ways:

1. DB only: -b
2. DB and keeping the last x versions as file: -Bx

In general NeDi only writes a new backup, if the config actually differs from the previous version. Some devices provide an SNMP OID that holds the timestamp of the last config change (Cisco and Comware are known). This makes the process more efficient as it won't require downloading the config to determine whether it has changed or not. A 2nd OID makes it possible to determine whether the running config has been written to the device's flash and alert, if not:

```
CFGC:Last change @5858408s uptime
EVNT:MOD=B/1 L=150 CL=cfgs TGT=3560CX MSG=Config changed
(@5858408s) 54.15 days after writing to flash (@1179413s)
```

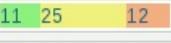
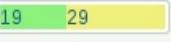
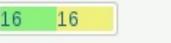
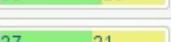
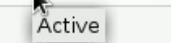
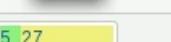
Once configs are backed up, they can be tested for compliance, searched, compared, be used as template for new deployments (e.g. via tftp) or be translated into new configs for completely different brands and models (starting with NeDi 1.7).

Id	Status	Class	Target	Devices	Port	Vlan	Action	Information	User	Time	Execute
1	⚡ Configuration	community public					⚠️		admin	7.Feb 17 15:07	📄 🚫 ✖️
3	⚡ Configuration	contact Remo					⚠️		admin	7.Feb 17 15:11	📄 🚫 ✖️
7	⚡ Port Configuration	switchport mode trunk	ABC 2960	---	---	---	⚠️	Tagged voice	admin	19.Jun 17 17:26	📄 🚫 ✖️
3 Values											

Configuration Compliance Policies

Device Modules

Most switches and routers contain linecards, removable fantrays and powersupplies and optical transceivers. NeDi is able to discover those modules to a good extent. They can be listed in [Devices-Modules](#) for review. However NeDi tries to present this information in a most useful manner. Stacks for example have become more popular in recent years. Management tools like NeDi should be aware of how they are physically built, but don't overwhelm the user with less relevant information. This gets even more complicated when whole network fabrics are being managed with a single IP address. To answer a simple question like how many switchports are available in a certain rack, becomes more challenging to answer. NeDi combines the modules with the interfaces to present such an answer:

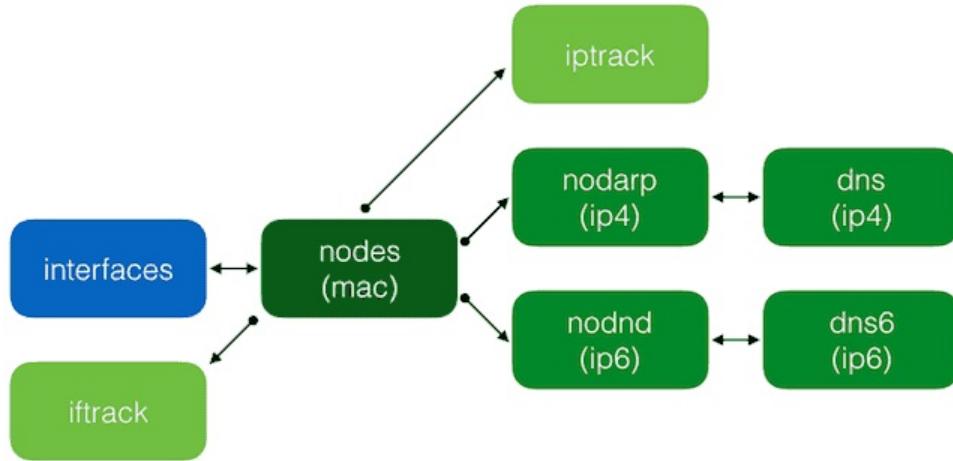
Fex-105 Nexus2248 Chassis	Fabric Extender Module: 48x1GE,	 11 25 12
Fex-105 Nexus2248 Chassis	Fabric Extender Module: 48x1GE,	 19 29
Fex-106 Nexus2200HP Chassis	Fabric Extender Module: 16x10GE	
Fex-106 Nexus2232 Chassis	Fabric Extender Module: 32x10GE	 16 16
Fex-106 Nexus2248 Chassis	Fabric Extender Module: 48x1GE,	 37 11
Fex-106 Nexus2248 Chassis	Fabric Extender Module: 48x1GE,	 30 18
Fex-106 Nexus2248 Chassis	Fabric Extender Module: 48x1GE,	 27 21
Fex-107 Nexus2200HP Chassis	Fabric Extender Module: 16x10GE	 Active
Fex-107 Nexus2232 Chassis	Fabric Extender Module: 32x10GE	 5 27

NeDi breaks complex fabrics down to report ports per extender

Network Population (Nodes)

NeDi treats all MAC addresses found in the bridge-forward tables of switches as nodes. They are combined with ARP information from layer 3 switches and routers. The IP addresses are resolved to provide actual hostnames, which will ideally complete the whole picture on the network.

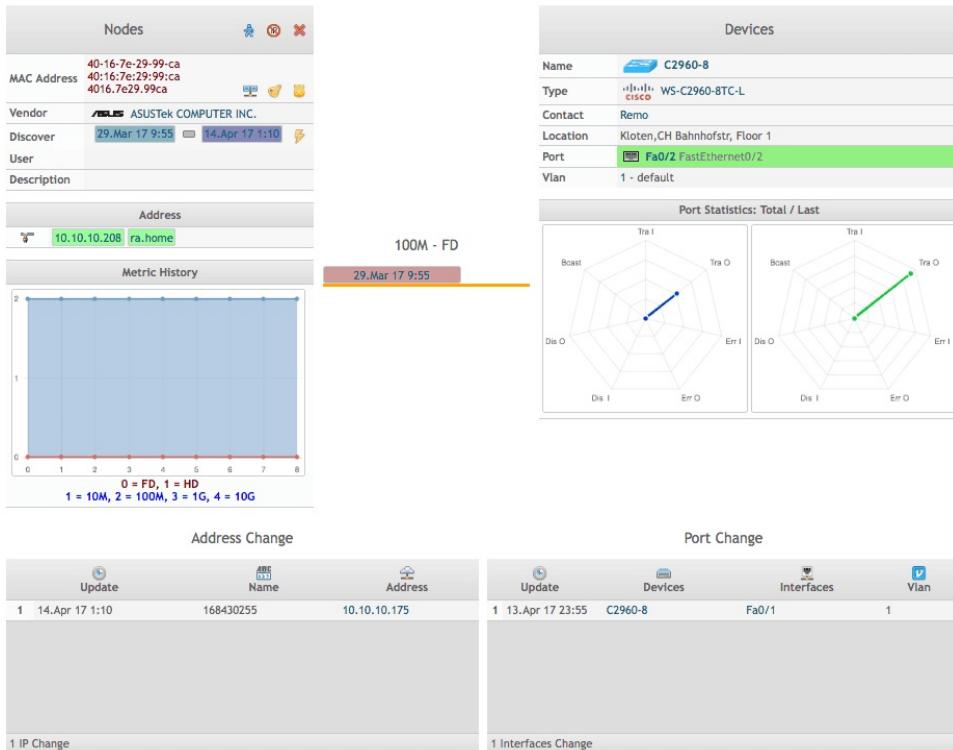
Over time the movement of the MAC addresses and changing of IP addresses is tracked in separate tables as well:



/>

Database relationship around nodes

Nodes can be listed in [Nodes-List](#) and then be closer examined in [Nodes-Status](#). All tables shown above are graphically represented in this view:



Node status is graphically organized

Edit ned.conf

The main configuration input for NeDi is the ned.conf file. The first task in configuring NeDi is editing this file. You can use [System-Files](#) in the web GUI to edit ned.conf, the seedlist and finally crontab to schedule recurring discoveries. Make sure you edit ned.conf before starting to discover your network. The configuration should be self explanatory with the comments in the file.

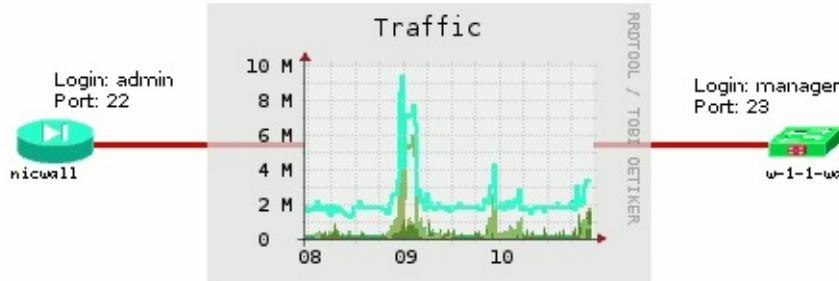
It's divided into the following sections:

1. **Device Access** defines credentials and methods for contacting devices
2. **Discovery** controls IP address space, ports used, and borders within which discovery should occur
3. **Backend** sets DB access, system settings and integration with other tools
4. **Messaging & Monitoring** takes care of polling and notification settings
5. **Nodes Related** controls how nodes should be read from devices, and how they should be treated afterwards
6. **GUI Settings** control menu items and appearance

User passwords can be entered encrypted with the usrsec keyword. The secret used to encrypt is in the function XORpass() within inc/libmisc.pm. Change it for more security (but don't forget to adapt after a NeDi upgrade or patch). This protects the passwords from prying eyes in ned.conf, but of course not, if the person has access to libmisc.pm.

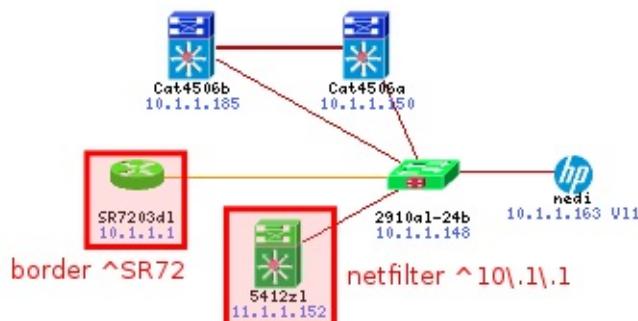
Many things can be fine-tuned at a later stage, but these parameters should be configured properly from the start:

- **rrdstep** sets the timestep of the graphs upon their creation (corresponds to the discovery interval)
- **cli-credentials** for CLI access to get MAC address tables on Cisco switches and/or configuration backup



If you discover your entire network with those settings wrong, you have to delete all graphs or reset the "CLI access information" on devices which failed due to incorrect credentials.

If you're using discovery protocols, you may have to specify a regular expression (regexp) to specify border devices where the discovery should stop, or limit the valid IP address range with netfilter regexps. Otherwise NeDi might wander off and send read community and CLI credentials to all destinations it can reach. Limiting the discovery may also be necessary if, for example, you do parallel discoveries (see table below).



Edit seedlist

Add single IPs or ranges like:

```
10.10.10.1
10.11.10.1-254
10.12.1,3,5,17.10  mycomm
10.13.1,3.10-15    newcomm - CH;Zug;Seeweg 1;U;Lab
```

- It is recommended to add "-P timeout" to ping seeds devices prior to discovery, if you use ranges.
- A community string can be added after the IP to override those in ned.conf (Note: SNMP v3 credentials still need to be defined in ned.conf, as they require more arguments that the seedlist does not support)
- Name, location and contact can be added as well to override information derived from devices (put a - to ignore, e.g. the name cannot be set for ranges)
- Use -u option to specify seedlist file (e.g. for parallel discoveries)

After the lines that add seeds, you can restrict ranges as well. Just put a ! at the beginning. This allows for more sophisticated scanning of network ranges. Here's another example:

```
# Adding desired ranges
10.11.10-13,15.1-254
10.11.22.11-200
# Excluding unnecessary IPs
!10.11.11,22.17
!10.11.11,22.17
```

If you don't want to edit seedlists you can add target(s) with the -a option followed by an IP or range.

Discover the Network

Once you have the prerequisites in place, and you have your ned.conf file (and optionally your seedlist) set to go, it is time to launch your discovery. The easiest way to do so is from the web GUI, and for instructions on doing this, read on. You can, however, launch the discovery script, ned.pl, from a command line, and control a given discovery run with command-line options. It has evolved to be a flexible tool, covering different needs. If you only want to know for example, where computers are connected to and don't care about linecards or interfaces, you can simply skip them. This speeds up the discovery and causes less traffic on the network. It can also make sense to update modules, device addresses and vlans only once at night (and maybe skip ARP and MAC address tables instead). That's where a proper crontab schedule lets you optimize regular discoveries (see below).

It's also important to get a feeling for discovering network components. Some devices (like stacked switches) can take their time to reveal their modules or even refuse if they're busy (as seen on Cisco Cat4k5).

Another aspect of the discovery is how all devices of the network should be found. The seedlist (described above) is one option and can be used in conjunction with discovery protocols, routing tables or OUI listings. Of course new devices can be added manually as well...

You can use the following examples to determine the best approach to discovering your network, and optimize your NeDi installation accordingly. When calculating discovery times, a good rule of thumb is to assume a 5s average discovery time per device:

Network Type	Discover Method
1 site, up to 100 devices of same vendor with LLDP or CDP enabled	<ul style="list-style-type: none">Set rrdstep to 900 in ned.confLeave seedlist empty or specify a core switchRun "ned.pl -p" every 15 minutes
1 main site and a couple of remote locations, up to 500 devices of several vendors	<ul style="list-style-type: none">Leave rrdstep at 3600 in ned.confPut an IP for every CDP or LLDP "island" in the seedlistRun "ned.pl -p" every hour
1 or 2 main sites and many remote locations, up to 2000 devices	<ul style="list-style-type: none">Leave rrdstep at 3600 in ned.confCreate 4 seedlists splitting up the sitesRun "ned.pl -pu seedlist" for every seedlist you've created in previous step with 5 min. offset every hour
Many sites with 5000 devices or more	<ul style="list-style-type: none">Set rrdstep to 14400 (4h) in ned.confCreate seedlists splitting up the discoveries, with that the longest ones take around 2hRun "ned.pl -pu seedlist" with 5 min. offset every 4hYou may also consider setting up NeDi agents in every major site and use a central NeDi master

Additional hints:

- If you don't want to add every device not supporting discovery protocols to the seedlist, you can discover them manually with ned.pl -a. Then you just have to make sure they're rediscovered by using -A dbquery in subsequent discoveries which will add them as seeds from the database.
- If you want to have less common devices added automatically, try the oui discovery method: Add a vendor to the ouidev regexp in ned.conf and use ned.pl -o. The MAC addresses of all arp entries are now resolved to their vendors and checked against this regexp. If it matches, the IP address is then used as new seed device. This

method is not recommended for vendors producing NIC chipsets or computers as NeDi would try SNMP access on all of those as well!

- Either use the GUI module [System-NeDi](#) or start it directly from the CLI. Make sure you're doing the latter as the same user as you run the crontab with or RRDs won't get updated correctly. You'll probably get the best results, with using the CLI and the -v options to closely follow the discovery.

These options define how neighbors should be added:

1. **-p** Use dynamic discovery protocols like CDP or LLDP
2. **-o** search arp entries for network equipment vendors matched by ouidev in ned.conf
3. **-r** use route table entries of L3 devices

A run without any options will result in a plain static discovery using the Seedlist or the default gateway, if you haven't added any seeds there yet.

Using -A lets you add seeds directly from DB. For example queue all snmp devices:

```
nedi.pl -Aall
```

Or queue all IOS devices:

```
nedi.pl -A"devos = 'IOS'"
```

Similarly -O can be used to queue ARP records matching certain MAC addresses or vendor strings:

```
nedi.pl -O"oui ~ 'Extreme'"
```

Edit crontab

After you set up nedipl to run the discovery the way you want it to, you will want to have it regularly check the network for new devices. NeDi will keep adding what it finds, and tell you when devices appear and disappear. As shown above, how often you run it is up to you, and should depend on the size of your network, how long discovery takes, and how important it is to you to find devices soon after they appear. Most installations like to have data up to date within a few hours, but for some once a day will suffice. Note that the frequency of discovery is mostly independent of the frequency of monitoring, and this section describes how you can set the frequency of discovery with the cron daemon.

Cron is a standard Unix daemon allowing execution of specific programs at given times. A file called crontab is used to schedule the tasks. Its format is fairly simple. Every line starts with the time fields (minute hour day month weekday) followed by the command to be executed. The output of the commands can be redirected to logfiles. These can be reviewed in the web GUI under [System-Files](#). The default path is /var/log/nedi. A % character needs to be preceded with a backslash.

```
# Crontab example running every 4h
0 */4 * * * /var/nedi/nedi.pl > /var/log/nedi-`date +%H`.run 2>&1
```

You can simply use [System-Files](#) to edit the crontab file. It'll be automatically applied for the user running the webserver upon writing. This means RRD files should belong to the same user or they can't be updated by the scheduled discovery. It's common practice to simply let this user own all files in the NeDi folder.

Asset Discovery

Life cycle management of IT infrastructure has become more and more important over the past years. NeDi can be optimized to cover many aspects of this process. It starts with collecting an inventory, and comparing it to vendor life-cycle information and maintenance contracts. The data can then be exported with NeDi's API for further processing in your environment.

Using NeDi

As mentioned before, the discovery has become very flexible and can be optimized for gathering assets only. In this scenario you're probably not interested in graphs, interfaces statistics, ARP or MAC-address tables.

On the other hand you want to add discovered devices and modules to the inventory table. The following command will achieve that:

```
nedi.pl -SAFGgadobewitjupv -Yam
```

If you use [System-Files](#) with "update-replace config" and select "ciscoeol.tgz", it'll essentially unpack a file called "ciscoeol.csv" in the nedid root folder. If nedid is called with -Y options, all device types and module models are compared against that file for EoL information, which will be added to the asset record.

- As of now only Cisco products are supported. Data from other vendors will be provided, should it become available
- As an alternative to EoL data, you can upload maintenance contract information in [Assets-Management](#)

Using NoDi

NoDi stands for node-discovery and moves one step further away from network infrastructure, towards the endnodes. This feature allows for completing the IT inventory or providing more insight in regards to security or monitoring tasks. As a side effect, Nodi monitors and graphs CPU, Memory, Temperature and Disk IO as well.

It uses SSH or WMI to retrieve information from Unix or Windows hosts. The latter relies on [wmic](#) provided by Openvas.

Edit nodi.conf to define the credentials (encrypted passwords are supported as well):

- The first usr or usrsec entry should be a domain admin as it's used for default WMI authentication
- All subsequent usr or usrsec entries are used for SSH
- A user can be forced with -u option
- The working user is stored in the DB and will automatically be used in successive discoveries

It's possible to store the node discovery information in a new database, to keep network mangament separated:

- Change dbname in nodi.conf to something like nedid_node
- Change arpwatch in nodi.conf to the nedid dbname (used with -O to read arp entries)
- Use nedid.pl -i -U nodi.conf to create it
- Use nodi.pl to discover the nodes
- Use System-Snapshot to switch between the databases

Troubleshooting

Testing

The **-t** option lets you test a particular discovery aspect. No data will be written upon completion.

For example, if you created a complex seedlist, you can test it with **-ts**. This should be combined with verbose or debugging output, to actually see something:

```
nedi.pl -vts
```

Debugging

If you encounter problems, make sure you understand what you're looking for. Any discovery related problems, such as dynamic discovery protocols, authentication or just properly identifying devices can be debugged with **-d** and **-D**:

- **-db** show basic debug information
- **-dd** show database queries
- **-ds** show system stats
- **-dc** log CLI access to input.log and output.log (open 2 more terminals and tail -f to them)
- **-dv** create *.db files to store internal variables after the discovery (for use with **-D**)
- **-D** will not discover your network, but rather use the previously generated *.db files on functions to be debugged in nedipl's "Debug Mode" section (intended for developers/me only)

Frontend Overview

REST API

Prior to NeDi 1.7 only POST calls with the following variables were supported:

- u = username (only users without a Device-Filter are accepted)
- p = password
- t = table (e.g. devices)
- q = query (e.g. device='charon')

A rewrite rule (e.g. for nginx) makes the requests more human readable:

```
location /api {  
    rewrite ^/api/(w*)$ /query.php?t=$1&q=$args last;  
}
```

As of NeDi 1.7 regular GET calls using "Basic Authentication" became available as well. This makes integration much easier as shown with the "RESTClient" addon for Firefox:

The screenshot shows the RESTClient interface in Firefox. The top bar includes File, Authentication, Headers, View, Favorite Requests, Setting, and the RESTClient logo. The main area has a title [-] Request. The Method is set to GET, and the URL is http://localhost/api/devices?device+regexp+'2520'. The Headers section contains an Authorization header with the value Basic YWRtaW46YWR... . The Body section is empty. Below is the [-] Response section with tabs for Response Headers, Response Body (Raw), Response Body (Highlight), and Response Body (Preview). The Response Body (Preview) tab is selected, displaying the following JSON output:

```
1. [  
2.     {  
3.         "sysname": "Linux",  
4.         "nodename": "nedidev",  
5.         "release": "3.16.0-4-amd64",  
6.         "version": "#1 SMP Debian 3.16.7-ckt20-1+deb8u1 (2015-12-14)",  
7.         "machine": "x86_64",  
8.         "nedi": "%VERSION%"  
9.     },  
10.    {  
11.        "device": "2520G8-P",  
12.        "devip": "168430084",
```

As you can see, some information about the NeDi host is returned in the first element.

Managing Assets

Nedi manages the life-cycle of your network infrastructure from purchasing until disposal. It allows you to include vendor's end of life information in order to identify unsupported hardware and maintenance contracts. The latter lets you find hardware not under maintenance or items you're paying for, that don't even exist in your network!

Assets are stored in the inventory table. They can be manually added with [Assets-Management](#) or automatically with the -Y switch in nedipl.

Possible life-cycle stages:

1. New: Adding devices and modules to inventory via barcode scanner (keeping track of spares)
2. Active: Items with serial numbers can automatically updated upon discovery (managing equipment in use)
3. Used: Item has been removed from network and put back in storage.
4. Replaced: Item has been replaced by another one (e.g. RMA)
5. Disposed: Item has been removed from network and trashed
6. Traded-in: Item has been removed from network and traded in for new ones

Cisco EoL information

A file called ciscoeol.tgz can be uploaded with [System-Files](#). It contains a list of all products with known EoL information. The fields are mapped as follows:

- "Migration Product ID" is added to comment
- "End of Routine Failure Analysis" Date is mapped to endsupport
- "End of Service Contract Renewal" is mapped to endwarranty
- "Last Date of Support" is mapped to endlife

The NeDi GUI

NeDi features a modular frontend, which can easily be customized. This is done by commenting out or including lines beginning with "module" in ned.conf. If a module is enabled in the file, the menu item corresponding to the module is included. The "Section" controls the top menu, and the "module" to the menu item. The Section-Module.php interprets these lines. The icon used is specified in the 3rd column. The group determines which users are allowed to see and use that particular module, so it can be customized for classes of users as well.



The NeDi web GUI modules have many common elements. Here's some useful information to keep in mind while using the GUI:

- Each module consists of a header row and a main input form. A larger version of the menu icon always shows up to the left and a click on it resets the module to its defaults. If you hover over it, the exact module name is revealed (shown in the footer as well)
- If "List optimize" is selected (next to 📁 in User-Profile), a history is shown next to the menu
- Use the "Find IT" search field in the header, to get quick results on any text, IP or MAC address
- The print icon 🖨️ in the header opens a printable view of the current module (usually without the main input form). On most lists you can click on the rows to highlight them
- You can save the state of most modules to a bookmark or use the notepad icon 📝 in the header to add a link to the admin message in User-Profile (look for "EDIT" on the bottom and change accordingly)
- Text links usually lead to applying a filter within the current form
- Numbers after a bar-image (e.g. # of device types) take you to the corresponding list module
- Used SQL queries can be shown by clicking on the debug icon 🐛 (only shown for admin). It executes the query in Other-Export for quick analysis
- Regular users see 🙋 and those having a view filter applied get 👀 instead. Hovering over it reveals the username and current server time

Lists

NeDi displays most of the data it finds in tabular displays, and these are controlled by "List modules". The presentation of data can be highly customized and exported to various formats. If you need to filter, show, and search through the data, you should learn how to master lists. Here's what the list controls do:



- By default some reports are shown on the bottom of most list modules. The Σ setting in User-Profile determines how many entries are shown
- Clicking on a text link takes you to the full-featured report
- Use the "Columns" select box to add or remove the columns you wish to see (hold down CTRL to select multiple columns)
- If "List optimize" is selected (next to in User-Profile), the columns are persistent for the entire session and a report is shown by default
- You can use the templates as quick list shortcuts
- In the filter section, you can define a criteria and select the combination operators AND/OR to add up to four conditions (first and second pairs may be grouped together with brackets)
- Alternatively you can compare 2 columns directly by using the other combination operators (e.g. "1 = 2" with columns "First Discover" and "Last Discover" selected to list devices only found once)
- The last map can be included via and a limit Σ can be chosen as well (default is 250)
- The triangles \downarrow in the header row allow for the list to be sorted accordingly. They're not available on special columns containing realtime data or graphs and other statistics
- You can export lists as XLS by clicking on the spreadsheet icon , if shown in the header

Monitoring

NeDi does monitoring as well as discovery. The program moni.pl is used to check the health and uptime of devices, and you can combine it with trap.pl for SNMP trap translation, syslog.pl for log messages, and ned.i.pl itself for the monitoring of discovery events. NeDi uses levels and triggers to categorize and alert you when monitoring finds something interesting. Discovered devices are not monitored by default. Any thresholds (CPU, Mem etc.) and notification triggers are applied from ned.i.conf. Syslog events only receive a level of 30 (Other), and thus can't generate alerts.

In order to monitor targets they need to be added to the monitoring table, since devices and nodes are dynamically overwritten by the network discovery (ned.i.pl) and you don't want to lose the list of monitored devices each time this happens. You can do this in [Devices-List](#) or [Nodes-List](#) by first filtering the devices you want to monitor with the list controls, then clicking the "Monitor" button. Alternatively you can add single targets in [Devices-Status](#) by clicking on the binoculars . Once added to monitoring, targets can be further configured in [Monitoring-Setup](#).

The monitoring daemon moni.pl first sends non-blocking uptime requests to all SNMP targets. Afterwards all other targets are tested sequentially (factoring in availability of their dependencies). For example, a dual homed web-server will only be checked if at least one of the connected switches returned an SNMP uptime.

- TCP ping is used by default for nodes and non-SNMP devices (this can be changed to ICMP in [Monitoring-Setup](#))
- Uptime (or SNMP-Engine time, if set in .def) is chosen for devices as it can detect intermittent reboots as well
- BGP peers can be monitored as well, if BGP4-MIB is supported on a device
- IF oper-status can be monitored as well (e.g. on router or server switches)
- The monitoring daemon should be started automatically. It also relies on ned.i.conf, where you can set the interval between polls, how many tests a device can fail before it is marked as down, and how alerts should be sent
- If you change the settings, they will be effective as of the next polling cycle. If you want to see results immediately, restart the daemon from [System-Services](#)
- If a target is reported to be down, an entry is created in the incidents table with the start time set to the time it's detected at. The end time will be added automatically, when the target is responding again. Incidents are acknowledged by classification in [Monitoring-Incidents](#)

Due to limitations of the SNMP perl module and non-blocking requests, latencies are not accurate unless you modify Net::SNMP's Message.pm:

Line 23:

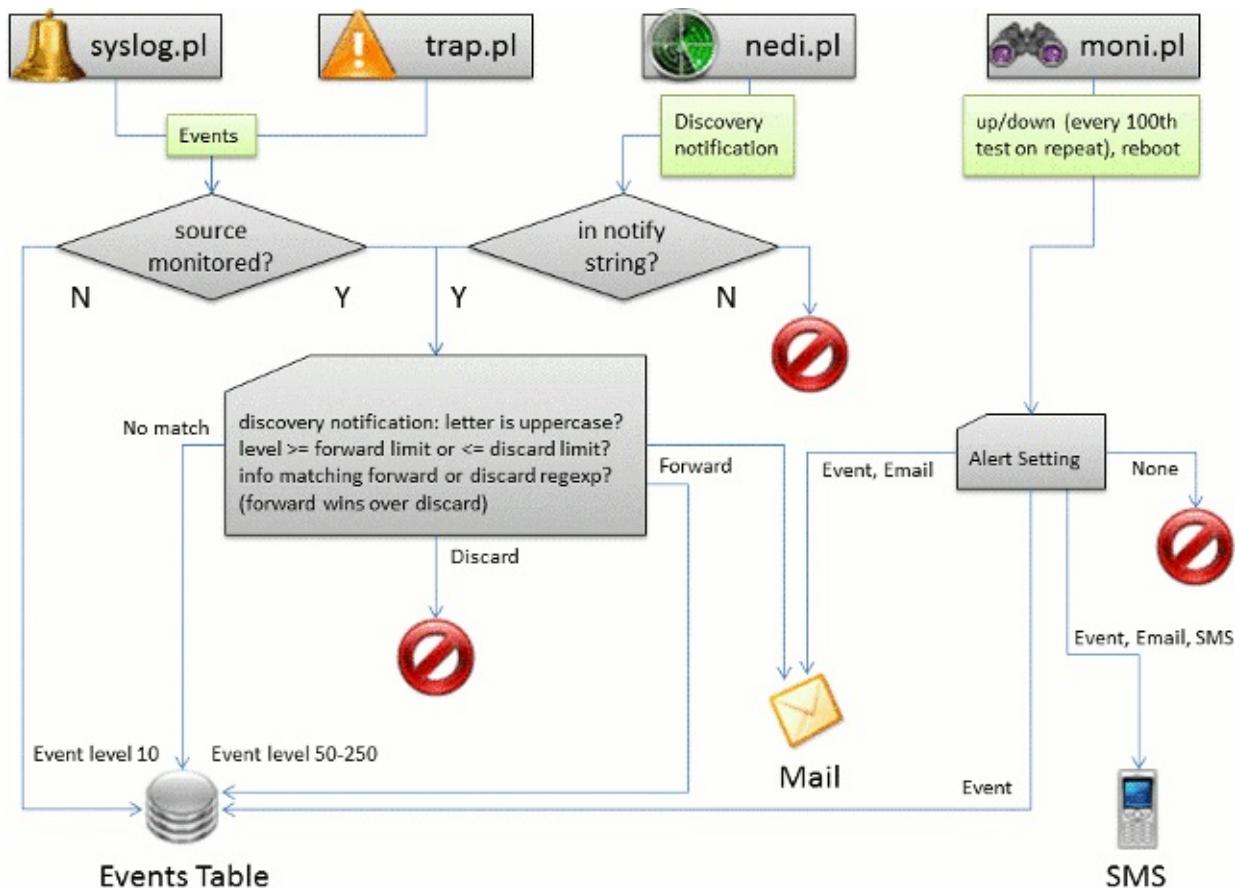
```
use Time::HiRes;
```

Line 691 or so, above debug output in send():

```
$this->{_transport}->{_send_time} = Time::HiRes::time;
```

Message Flows

The following diagram explains how events (originating from syslog, trap, discovery and monitoring) are processed.



Reporting

This section aggregates information and provides extensive views of the collected information. There are several reports tied to their respective database tables (i.e. devices, modules or nodes) and a "Combination Report", which focusses on a task (like asset management), which depends on several tables. Using the reports section involves the following:

- Setting a filter, if desired. Any filter you set is taken into account for links to other modules
- Selecting the reports you want (hold down CTRL for multiple selection)
- Use the templates as quick filter shortcuts (see the icons on the left of report select box)
- The last map can be included via and a display limit can be chosen as well (default is 10)
- Reports can be "optimized" with (table captions reveal, what's been optimized)
- "Alternative sort" via uses keys rather than values (table captions reveal what has been sorted on)

GUI Modules

This section describes the various modules and their functions. You can enable or disable these modules in the GUI settings section of the ned.conf file.

Assets-List

This module lists assets from the inventory table.

- You can make changes to multiple assets for the state and maintenance renewal by selecting desired values and clicking update.
- You can delete all filtered assets (ignoring limit setting) by clicking delete. This feature should be used with care!

Assets-Locations (Location List)

List locations created with the location editor [Assets-Loced](#). The following functions are available:

- The execute column reveals links to other modules or external maps
- If photos are uploaded to the appropriate topo folder, they can be accessed in the Files column
- A NeDi- or static-map can be added as well. If the coordinates are available, labeled markers are drawn

Assets-Loced (Location Editor)

This module can be used to place locations on previously uploaded backgrounds, using [System-Files](#). Alternatively, if you check the first box to the right of  you will enable NeDi's geocoding API, which automates the placement of locations. When enabled, location names are used to search for the correct coordinates. If you check the second box, the description is used instead. Create the locations and enter descriptions prior to enabling checking this option for best results.

Usage with Background Images

- The default lets you place your locations on a background image and can be leveraged with the "bgmap" map type in [Topology-Maps](#).
- At first you'll see the top level map, which is a world map by default (I'll change this as soon as NeDi manages networks on other celestial bodies).
- Select a region and click on the map to set the coordinates. You'll notice that values are being populated and the 'Add' button becomes 'Update', if the location already exists.
- If you want multiple layers for your maps, upload lower level maps to the correct location in the topo folder. For example, say your network sites are located in 2 regions (USA and Europe). Just name the map files background.jpg and upload them to topo/USA and topo/Europe. Now, when you select cities in those regions, the appropriate map should be shown, and you can place cities accordingly.
- This also works within cities (makes sense where you've got a big metropolitan network). Just upload background.jpg to topo/Europe/Zurich and as soon as you click on buildings in Zurich you can place them in that metropolitan map.
- The subfolders are created automatically when you drill down in Topology->Table with Openstreetmaps enabled.

Usage with Geocoding

- Select the location you wish to add
- If it doesn't exist or the coordinates are 0 (if it's been added to a background image previously), a geocoding lookup is performed and the coordinates are shown in blue
- If you use internal names for your locations, you can enter a "geocodable" name as comment and click add
- Activate description mode with the 2nd checkmark right of 
- A draggable marker is placed on the map, which can be adjusted to fit your needs (coordinates turn green). Enter a description and click the add button
- If this doesn't work for you, click on  to enter an address manually
- The coordinates should stay black, as they're read from the DB now

Assets-Management

This module allows you to add or edit one asset at a time.

- Asset summaries are shown by default. Click on the text to get a filtered list of matching assets, and click on the value to add items to the [Assets-List](#) module.
- Use a bar code scanner (send a "tab" upon successful reads) to scan type and serial number, or just enter them manually.
- Specify location, condition, source/provider and warranty. If the latter one is closer than a month away it'll be highlighted with the "warning" color, or with the "critical" color if already expired.
- Click on the  icon to open the panel browser.
- Refer to [Devices-Modules](#) for a list of possible classes.
- If you list by a property (e.g. location), the appropriate field on the top is populated as well for easier batch additions.
- You can edit the items listed by clicking on their serial numbers. The current list will stay. Note that the focus will move to the location field, as serial numbers cannot be edited. You can either update or delete an item now.
- If you click on a class icon, you get to the respective device or module if it has been discovered
- You can export a list as XLS, but the [Assets-List](#) module is more flexible in that respect.

You can upload a CSV file containing assets with their maintenance contract information as well. Specify the following in the form and select the file:

-  Select date format used in the CSV file
-  Field separator
-  Rows to skip from top

Currently the columns in the file to be imported need to be arranged like this:

Field	Example	Description
Class	License	Only Software or License is identified. Everything else (e.g. Chassis) can be determined upon discovery
SLA	7x24	Stored in 'Services Level'
type	2520-8G-PoE	The type as specified by vendor can be used to determine its EoL status
serial	123456ABC	The SN# is the primary key in the inventory table
count	-	Currently ignored (just add an empty column for now)
serial2	ITEM2345	Will be used, if the first SN# was not available for some reason
contact	Sherlock Holmes	Stored in 'Asset Contact'
address	221b Baker Street	Combined in asset location with place (to place;address)
place	London	Combined in asset location with address (to place;address)
description	anything useful	Stored in 'Maintenance Description'
renewal	Yes/No Ja/Nein	Determines whether maintenance contracts are renewed or not (Maintenance Status)
end of maintenance	05/26/2015	Current maintenance end date
end of sale	-	Currently ignored (just add an empty column for now)
end of support	05/26/2036	End of routine failure analysis
End of Life	05/26/2071	Last date of any support

Devices-Config

NeDi will back up your device configurations if it has privileged CLI access and you tell it to with -b, or -Bx. With the [Devices-Config](#) module you can review and compare backed up configurations and their changes.

- A config report and recent backup-related events are shown by default.
- There are two modes of operation which are list and compare.

List Configurations

For simple listing of configuration values, follow these steps:

1. Search for text by setting a filter
2. Limit number of displayed characters in the excerpts
3. Limit number of displayed devices
4. Click on an excerpt to view the whole configuration

Compare Configurations

You can use this module to quickly see differences between stored configurations.

1. Choose a reference device from the "List" selectbox.
2. Now either select the 2nd device from the left select box in "Comparison" or leave it at -Type- to compare against all configurations of the same type.
3. Select how the output should be displayed.

When viewing a configuration you've got the following options:

- Toggles line number display for easier change review.
- Suppresses the motd character with that configurations of Cisco devices can directly be copied and pasted.
- Use [System-Database](#) to display the config as plain-text or select a file version in the changes area to edit the actual file (available when you run nedipl -Bx).
- Clears configuration or changes.

Devices-Doctor (Device Doctor)

Presents device specific diagnostic reports and point out potential problems (alternatively you can select a config which will be displayed in context groups).

1. Generate a "show tech all" file on a HP ProCurve/Aruba or Cisco device and store it locally.
2. Browse for the tech file you wish to analyze.
3. Click Show to process it.

Note: This feature is still being refined for more accurate results.

- Red letters on a yellow background reveal potential problems (hover over it, to learn why).
- Adjust the broadcast/traffic ratio (default 10%) to identify problems on interfaces.
- Green lines mean that a checked condition looks ok.
- Dark red and Olive green letters represent interface status in the respective context.

Devices-Graph

This module allows you to dynamically generate stacked interface graphs and much more.

Please note that NeDi's graphing feature was implemented as an addition to the discovery with lowest possible resource and maintenance cost in mind.

It will not graph those 5 minute peaks (unless you run NeDi every 5 minutes in very small networks), but provides a longterm view of each and every interface. This translates to baselining and prediction of potential bottlenecks, instead of identifying erratic outbursts of any kind (You'd prefer using a tool like Cacti to monitor this instead).

- Select any top graphs if you wish to get the big picture on your network.
- Selecting a device will reveal its interfaces. You can choose several of them to be stacked dynamically (doesn't work for IF status!).
- Select several graph sources at once to correlate and investigate problems (e.g. CPU load, broadcasts on some interfaces of a device)
- System related graphs are CPU, Memory and Temperature and a custom graph for other values.
- Use double arrows to move start (top one), the whole graph (middle) or its end (bottom one) by weeks or single arrows for days. Click on a date icon to manually set a start or end time.
- If you can't live without degrees in Fahrenheit, adjust the setting in [User-Profile](#).
- CPU and memory corresponds to System load and battery capacity on UPS units'.

If you use Cacti on the same host, you can integrate it into NeDi:

- Configure the cacti options in ned.conf.
- Now you can add devices and interfaces to Cacti here in Devices-Graph.
- A cacti icon will be shown in [Devices-Status](#), if the device is available in Cacti. Clicking on it takes you there.

This is a premium module, only available with NeDi+. Find more details [here](#)

At this time only HP ProCurve Switches have been tested!

This module is part of NeDi's provisioning system. It allows for installing unconfigured switches upon discovery. The procedure is divided into the following steps:

1. Create install entries specifying device type and IP address to be matched. The desired name and IP settings need to be set as well, rest is optional
2. Create an install template with System-Files (see below)
3. Perform installation (with nedipl -T or checking "Install" in System-NeDi). If type and IP match an install entry with the state "New", the target IP is pinged
4. If no answer comes back the entry is used to create a device configuration from the install template. The state of the install entry is changed to "Active"
5. If the device is discovered with the new IP address the state of the install entry is changed to "Used"
6. Check verbose nedipl output, if status changes to "Broken"
7. By default an install entries summary report is shown

Install Template

An install template persists of a series of commands (1 command per line with optional confirmation and timeout separated by ;) to prepare the target device and a config template with placeholders, which are filled in from the install entry. If used, the password is taken from the appropriate user in nediconf, but usually is a fixed/encrypted string

```
Cli command1
Cli command2;y;600
Cli command3;y;0
===
sysname %NAME%
ip addr %IPADDR% %MASK%
ip default route %GATEWAY%
vlan %VLANID%
snmp location %LOCATION%
snmp contact %CONTACT%
username %LOGIN%
password %PASSWORD%
enable password %ENABLEPW%
```

Devices-Interfaces (Interface List)

List device interfaces, their population and graphs. It also allows to add selection to Node-Track or set individual thresholds.

- If the interface status is discovered, the type icon is imbued with the respective color (not realtime). It'll be "admin down" (or 0), if it's been skipped in every discovery. Interfaces of controlled access points are not polled and set to unknown (or 128).
- Set alert thresholds next to  and click Update to override the values in ned.conf (enter 0 to clear)
- Set traffic to 101% or broadcasts to 65000, if you want to ignore respective alerts on particular interfaces (101% due to potential rounding errors, larger values are ignored as of NeDi 1.8)
- Setting a MACflood threshold allows this interface to discover multiple CDP/LLDP neighbors (e.g. in a hub and spoke topology)
- The population takes you to the Nodes-List where you get detailed information on the connected nodes.
- The graph size corresponds to setting in User-Profile.
- By default a port type and status distribution report is shown

Devices-List

List devices, system graphs, population, free access ports and configuration status. Realtime Spanning-Tree information can be added for troubleshooting as well.

- Unselecting the device column hides the icons i.e. to create a simple text list.
- The serial number is checked against the inventory and reflects support and maintenance status. Click on it to add it or update an existing asset (e.g. to track decommissioned devices).
- The selected devices can be monitored by clicking the Monitor button (go to Monitor-Setup to configure them further).
- The selected devices and related information (e.g. modules and interfaces) can be deleted, by clicking the Delete button.
- By default a vendor and type distribution report without piecharts is shown.
- Device specific thresholds can be edited by clicking and Update to change it on visible devices
- If you set supply-alert, PoE-warning or ARPpoison-threshold to 0, the defaults from ned.conf are taken instead

Device Options are used internally to describe the device's capabilities. They can be used for filtering as well. A '-' indicates that a property is not available:

Position	Character	Description
1	A,-	ifAlias from IF-MIB
2	C,W,-	CPU utilization or Wattage on UPS devices
3	P,S,N,-	Power-Ethernet MIB support and how interfaces relate to it
4	I,-	Has interfaces or not
5	d,s,i,m,r	Name from DNS, sysname, IP, mapped, mapped with regex
6	c,m	Contact from syscontact, mapped
7	l,m	Location from syslocation, mapped
8	U,S	Uptime (overflow every 1.3 years, SNMP-engine-time)

Devices-Modules (Module List)

List modules and compare hard- and software revisions for example. You'll also see VMs, Serverblades (on HP Bladechassis) or even printsupplies as well (The supplylevels are held in the FW and HW fields).

- By default a model and description distribution report is shown

Module Classes

The following table lists possible classes for modules. They can be used for assets as well:

Icon	Class	Description
	1	Other
	2	Unknown
	3	Chassis
	4	Backplane
	5	Container
	6	Power Supply
	7	Fan
	8	Sensor
	9	Module
	10	Port
	11	Stack
	18	Keypad
	19	Camera
	20	Patchpanel
	21	Cover
	30	Printsupply
	40	Virtual Machine
	50	Controlled AP
	60	Server
	61	CPU
	62	Mem
	63	HDD
	64	Card
	69	Display
	80	OS
	81	Software
	82	License

Devices-Status (Device Status)

This is the centerpoint of individual device management. It's divided into 6 sections (General Info, Modules, Vlans, Links, Interfaces and Monitoring Stats), which correspond to the tables Devices, Modules, Vlans, Links and Interfaces. The monitoring part is shown on the bottom representing data from events, monitoring and incidents.

You can create devices like clouds via . On those devices you can add more interfaces with the same button in the Interface section. Those devices can be used to link WAN-routers to a cloud or monitor external services.

Due to performance reasons, only uptime, poe, interface (and VM) operational status and interface last-change is realtime (if device was seen in last discovery). Everything else is retrieved from the database.

Overview

- Hover over the icons for hints on what they do.
- You'll see print supply levels on supported printers or VMs on VMware ESXis. If ssh access is enabled in addition, the VMs can be turned on and off.
- takes you straight to DefGen, in case you want to edit the definition file.
- adds the device to monitoring and tests SNMP uptime by default. This icon turns into a clock in that case or another symbol, if you change the test method in Monitoring-Setup (click on icon to get there).
- The serial number is checked against the inventory and reflects support and maintenance status. Click on it to add it or update an existing asset (e.g. to track decommissioned devices).

Interfaces

- Active interfaces' names are blue and clicking on it pops up a realtime SVG graph window, which lets you observe the traffic in a 1 - 300 second interval.
- If the absolute counter is not 0, the background becomes blue, showing you there has been traffic on this interface.
- Recent status changes, high error count or PoE values will affect the background as well.
- If the last status change is more recent than last discovery, Vlan, Speed and Duplex become grey as they may have changed.
- The background of population turns blue if a node was ever discovered on this port, even if it's empty now. The last seen MAC will be revealed upon hovering over it.
- Filter interfaces by status (only works, if device is reachable)
- Filter interfaces by Vlan uses the PVID field from the interfaces table. If you check "Untagged & Tagged" the vlanport table is used instead
- "IF Information" lets you choose what interface related data is displayed
- By default population, addresses and counter values are shown (0 fields are left empty)
- Interface graph size corresponds to the setting in User-Profile

Chances are you won't get complaints if you unplug a port where:

1. No link (icon not green)
2. Last change is as much as switch uptime...or at least a long time ago!
3. No nodes shown when Population is checked and field itself is not blue.
4. No traffic is shown and respective fields are not blue.

If the switch has been rebooted lately you may want to click on in the summary section to review free ports in the Interface List.

Managing

SNMP write enabled:

- You can change location, contact and admin status of interfaces directly.
- If the device is using standard MIBs you may also edit IF-alias (enter a "-" to clear) or toggle PoE delivery (e.g. to reset a hanging AP or VoIP phone).

CLI access enabled:

- Click on  to save the running configuration to flash.
- Click on  to look at the device's log.
- The "CLI Send ->" selectbox allows for sending command files (files starting with 'cmd' in the cli folder) to a device. Refer to System-Files for creating command files.

Customizing

A great way of customizing or integrating NeDi with other tools are two includes, which can be edited in System-Files:

1. devtools.php is included once and will be shown next to the device icon
2. iftools.php is included with every interface and shown with the interface addresses.

Devices-Translator (Configuration Translator)

This module allows for automated migration from an old device to a new model.

1. Add rules on how configuration blocks should be translated to a new device
2. In [Devices-Status](#) or [Devices-Config](#) click to prepare actual translation
3. Select target group(s) to generate desired config.
4. Write config to tftp folder (can be edited before in System-Files)

Translation Rules

Field	Description
Source Type	Exact device type to filter appropriate rules for the existing device
Target Group	<ul style="list-style-type: none">• Should describe the new devices• A group usually contains many rules• One or more are selected when preparing the translation, to enable flexible translations on the fly
Context	For example "interface" or "vlan" to limit context specific matches <ul style="list-style-type: none">• Regular expression to match an existing config line like "/interface (\d)\$/" (if "interface" is used as context in other rules, they'll be added after this rule)• Append _&&_regexp to match context names, e.g. adding _&&_/Fa([1-9]1[0-9]2[0-2])\$/ matches only Fa1-9,Fa10-19 and Fa20-22• If you append _&&_USEPRI the priority of the rule is used (not the one from the context), useful to move a line from a context to the global config
Destination	A replacement string like "interface Fa0/\$1" where \$1,\$2,\$3 replace Source matches in ()
Priority	Can be A-Z to define where the resulting config should be placed in the output
User	NeDi user who updated rule (a timestamp is available for filtering rules as well)

- Click on to duplicate all rules for specific source type to a new source type and/or destination group (only shown with first rule of a source type)
- Click on to edit a rule
- Click on to copy a rule
- Click on to delete a rule
- Click Show to list all, or a value in the default report to list specific rules
- Click Delete to remove all visible rules (use with care)
- You may want to export the translations table as gzip in System-Database for backup

Special Methods

Here are some useful additions for settings that can't be extracted from the source config or have been mapped to other values by NeDi (e.g. location). If necessary the function ProTrans() at the bottom of Devices-Translator.php can be customized even further.

Use Case	Description
Device IP address from DB	The placeholder %DEVIP% in the destination is replaced by devip from the devices table
Device location from DB	The placeholder %LOCATION% in the destination is replaced by location from the devices table
Device contact from DB	The placeholder %CONTACT% in the destination is replaced by contact from the devices table

Device group from DB	The placeholder %DEVGROUP% in the destination is replaced by devgroup from the devices table
Get Vlans from DB	Use "VLANNAMES" as source and something like "VLAN %VLID% name %VLANNAME%" as destination to list vlans from vlans table (e.g. if the source config is unusable)
Get interface Vlans from DB	Use "VLPORT-TAG" (or "VLPORT-UNTAG") as source and specify an interface context. Enter something like "switchport allowed vlan add %VLID% tagged" as destination to list tagged vlans from DB on that interface
	This method is able to extract statements like "tagged 1-10" within a vlan context and map it to interface based configs (e.g. translate from HP ProCurve to Cisco IOS)
Get tagged vlans from a vlan context and apply to an interface context	<ul style="list-style-type: none"> • Use a match like "/ tagged (.*)/" as source and "VLCONTEXT-TAG-ADD" as destination to add tagged vlans to list • Use a match like "/ untag (.*)/" as source and "VLCONTEXT-TAG-DEL" as destination to remove • Then in the "interface" (or similar) context use "VLCONTEXT-TAG" as source and something like "switchport allowed vlan add %VLID% tagged"
Copy tagged Vlans from an interface to another	<ul style="list-style-type: none"> • Enter VLPORT-TAG-COPY (source interface) in source • Comma separated interface list in destination • Add VLCONTEXT-TAG as described above, if you haven't done so already <p>This only works with numeric interface names at the moment. The reason behind, it was specifically developed to understand Zyxel configs.</p>

Devices-Vlans (Vlan List)

Lists vlans and their respective node population on the devices for example.

- By default a empty vlan report with a limit of 1000 is shown

This is a very helpful, but also dangerous module! Always use with caution, because you could create a big mess rather quickly!

This module lets you send CLI commands to devices and review the output instantly:

- Use filter to select the desired devices.
- Enter some commands in the "Execute / Configuration" area
- Click "Show" to simulate the process
- Click "Execute" to send the commands
- Click "Configuration" to enter configuration mode before sending the commands and save the configuration afterwards.
- On IOS or devices device with similar interface names you can use the "Interface Configuration" section to configure an interface range.
- Only devices of the same operating system can be used at once.
- Make sure you adjust GUI authentication (towards the end of ned.conf) to fit your needs.
- If you're unsure about this whole thing, delete inc/devwrite.pl to completely disable sending commands via webinterface!

Monitoring-Events

Incoming monitoring, discovery or syslog events and snmp traps (if enabled) are presented here. Several filter options allow you to examine problems very efficiently. Use the arrow keys (beneath show) to page around in your selection.

- MAC and IP addresses provide direct links for further investigation (e.g.  )
- Events can be acknowledged by clicking its Id. The level background turns grey and their level is divided by 10
- Based on the filter you get new shortcut icons for further investigation

Conditions involving criterias (e.g. location or contact) from the devices table, cannot be used to delete events due to query restrictions!

Event Classes

- Classes reveal the cause and source of an event (like syslog or discovery)
- Classes are represented with an icon and a mouseover description

Event Levels

Image	Level	Name	Description
	< 30	-	Any acknowledged event is divided by 10 (image shows an acknowledged event with alarm level)
	30	Other	Unspecified level (e.g. from unknown syslog sources)
	50	Info	Informational and good news
	100	Notice	You might want to look at this, if time permits
	150	Warning	You probably should look at this...
	200	Alert	Definitely look at this!
	250	Critical	Serious condition, fix it now!

Monitoring-Health

If you do use NeDi's network monitoring features, this is the module to just leave open in a browser.

- It'll refresh every minute to alert you (with original www.PSI.ch siren sounds!), if something goes down
- If you drill down into locations, the messages and events will be filtered accordingly
- A mobile version without graphs and session management (no login required) can be accessed with mh.php (delete this file, if you don't want to allow this)

Top section:

- Displays overall network condition
- The size of the graphs can be set (or turned off altogether) in User-Profile
- Target availability, excessive traffic or errors on interfaces
- Exceeded CPU, memory and temperature thresholds of devices

Event section:

- Some statistic to the left and important events within the last 24h to the right are shown by default
- Adjust #top events in User-Profile (< 6 shows less statistics, < 3 no events at all)
- Any event can be acknowledged by clicking its Id (internally dividing its level by 10; acknowledged events receive a gray background)

Topology section:

- Works just like Topology-Map, providing operational status of locations in addition
- Failed nodes and non-SNMP in a location are shown with  , but don't affect the background
- A shaded background indicates that not all SNMP devices are monitored in a location
- Events with a level of 250 cause a red flag to appear on the respective location (acknowledging it removes flag from location)
- Adjust #columns in User-Profile fit your screen (setting it to 0 hides this section)

Monitoring-History

Analyze events over time to disclose abnormal behaviour in the past.

- Use the filter to narrow down the events
- Select start and end point and the granularity for your analysis
- Group the events by level, source or class
- The output format can be bars or interactive graphs

Monitoring-Incidents (Incident List)

An incident is created whenever a device did not respond for 'uptime-alert' times (see ned.conf). Here you can acknowledge and classify them for future analysis.

- Once you know what happened select an appropriate category and enter some info
- You can filter on a category or active incidents where target hasn't recovered yet

The easiest way to acknowledge a heap of new incidents:

1. Set filter to "new"
2. Enter a description, where applicable
3. Select class (event disappears as you filter on new ones)

Monitoring-Map

This is an alternative to Monitoring-Health, displaying dynamic network maps on various dashboards. Alternatively you can add locations (at least the buildings) in Assets-Loced and use their coordinates for interactive maps.

- Setting "No Graphs" in User-Profile hides the charts on top (other sizes affect their size and the previews in the editor)
- Click main title to hide the section until refresh
- Click  to hide the section for entire session
- Look at the PHP code for tweaking the default timeouts

Adding NeDi Maps

1. Create a png map in [Topology-Map](#)
2. Click "Monitor" when finished
3. Go to Monitoring-Map and click  to access the editor
 - There are 6 groups (A-F) which rotate through the assigned maps (change/refresh every 10s)
 - There are 6 groups (a-f) which display the assigned maps at once (refresh with reload of page, every 180s)
 - The priority determines the order of the maps within a group
 - Click on  or  to edit or copy a map via Topology-Map
 - If you set access to all, the map will be visible to other users, allowing for copying it into their own views

Adding Geo Maps

1. Click  to switch to the interactive Geo map
2. Each flag represents a region (click one to see its popup menu)
3. 'Filter Map' zooms into the selected region and displays its cities
4. If you didn't create region or city locations in Loced, it'll place the flag on one of its children
5. Click  on the bottom to show all buildings (with current filter)
6. Click  to display sites with broken targets only
7. Click  when finished
8. Enter editor with  to adjust the size (100% = Full HD)
9. If you want to show several maps put them in different groups (1-9)

Adding RRD Graphs

1. Select graphs and their size in [Devices-Graphs](#)
2. Select group where they should be added to
3. Click Show

Monitoring-Master

The master console is intended for use on a central NeDi host, where only the master.pl daemon is running. All other GUI modules except [Devices-List](#), [Devices-Status](#), Reports-Monitoring and Monitoring-Events should be disabled to avoid confusion. In addition a unique theme should be selected to further distinguish this host from regular NeDi installations.

Setup

1. Add remote NeDi installations to the agentlist and add the usernames and passwords to access them in ned.conf
2. Run master.pl from System-Services (only visible if Monitoring-Master is enabled in ned.conf)
3. Go to [Devices-List](#) and add detected agents to monitoring (NeDi agents are treated as devices)
4. Go to Monitoring-Setup and select http or https as test  , to tell master.pl how to access the agents
5. You can add a path like ned/ as test option  , if ned is not accessible in the root path
6. Go back to System-Services to restart master.pl or wait for a 'pause' intervall to get the agents polled

Operation

- Upon first access, master.pl reads the last event with level 200 (alert) or above and all unacknowledged incidents.
- On subsequent runs only new alert-events are read. Incidents are removed from the master console, if they're acknowledged on the agent.
- Monitoring-Master shows those events and incidents with quick links to the respective agents.

Monitoring-Setup

Configure how targets are monitored and how users are notified upon a failure. The concept of Monitoring-Setup is to use the filter in order to apply settings to a single or multiple targets. If you don't set a filter, all targets are updated at once.

Filter

- Use the templates (icons above filter) or click on the links of Target (to match a single target)
- Clicking on a test icon (e.g. ) executes a monitoring test on this target
- Clicking on Alert or Events Action (e.g. ) from the list applies it as filter

Monitor

- Define the Test  (Should be uptime for all switches and routers already)
- Setting it to "No" skips active polling. Can be used as maintenance mode or if you just want to set event-actions or discovery thresholds on a device
- Select icmp if TCP ping doesn't work on a target. Enter # of packets in  , if you want to send more than 1
- Test http/https: You can enter a string like "index.html" in  and a regexp matching a successful response in  . Only a SYN check (TCP ping on port 80) is performed, if you don't
- Test dns: you can send a hostname and a regexp matching the expected IP address
- Test ntp: you can send RFC2030 fields like "Stratum" and enter a match ^[1-5]\$ to detect if your ntp server lost sync
- Clicking "Update" applies the settings to the displayed targets
- Clicking "Delete" removes the displayed targets from monitoring
- Select email or SMS alerts, just have incidents create Monitoring-Events or nothing at all. If you select a repeat option, the alert is resent every 100th failed test
- The Latency textbox allows for changing the latency threshold for individual targets
- Click on  to simulate an outage of the first monitored target

Events/Threshold

You can forward events as emails based on their level or contained text:

- With Forward in the first box select a minimum event level
- With Forward in the first box enter a regexp as the Filter 
- Alternatively you can select Discard, a maximum event level and/or a regexp and matching events will not even be stored in the DB (Level limit can only be used to forward OR discard but not both)
- Setting a regexp for Maximum raises matching events to level 250 (Emergency) and shows those within the past 24h in Monitoring-Health (useful to identify failed power supplies or stack members)
- The notify settings from ned.conf can be overridden for each target in the "Discover Notice" field 
- To clear any fitler enter a "-" by itself

Reset

-  Sets dependency info, if available via links or device information (in case of node targets). After that, the dependencies can be adjusted on each target individually
-  Updates target IP address from devices or nodes (in case they've changed, there's a  icon in the target status)
-  Reset the availability counters (lost & ok) once a year if you need to know annual availability for example
- A yellow/shaded target status indicates that its not found as node or device anymore (and should probably be deleted)

Nodes-Create

Can be used to create VMs on an ESX hypervisor, if SSH access is enabled and credentials are set

- Select hypervisor and VM to be used as template
- Enter a target name
- Specify number of CPUs, memory and disk size
- Enter full path and filename, if you want to install from a ISO image
- Click show to review the VM config and Add to create it

CLI Tips

If powering on a VM doesn't provide any result:

```
vim-cmd vmsvc/message (vmid)
```

If message ask for an answer:

```
vim-cmd vmsvc/message (vmid) _vmx1 1
```

If a process gets stuck and you get "Another task is already in progress" error:

Determine id of process in question:

```
esxcli vm process list
```

Then kill it:

```
esxcli vm process kill --type=force --world-id=(id-from-above)
```

Shrink thin provisioned HDD image (zerofill unused space first)

```
vmkfstools -K hdd.vmdk
```

Nodes-List

List nodes, corresponding interfaces, their graphs and available services for example.

- The nodes table with MAC-interfaces mappings is the base for this module. Its combined with IP, IPv6 and DNS tables, which may result in many entries, if several IP addresses are found for a particular MAC address.
- If you list realtime services, make sure you don't match too many nodes as it will take a long time to scan the open ports.
- Clicking on the NIC vendor icon takes you to Nodes-Status where you get all node details at a glance.
- You can add the displayed nodes to monitoring (testing with a TCP ping by default).
- By default the "Node Summary" report is shown

Conditions involving criterias (e.g. location or contact) from the devices or interfaces (e.g. IF alias) table, cannot be used to delete nodes due to query restrictions!

Nodes-RogueAP (RogueAP List)

This is an approach to detect potentially rogue access points from the wired side. All nodes are compared against a list of MAC address samples from consumer access points.

- Check 'Population > 1' to only show matches where several nodes are found on a port with matching MAC sample

Nodes-Status (Node Status)

This is the [Devices-Status](#) counterpart for nodes. It displays the node relevant information on the left, device and interface on the right with the connection in between.

- You'd usually land here coming from other modules like Nodes-List. Alternatively you can enter/paste a MAC-address in any common format (grouped by - or . or : or plain HEX)
- If you need the MAC-address in a CLI window of a device, simply copy the appropriate format shown
- View syslog events containing this MAC address
- Create a MAC policy (e.g. mark this node as stolen)
- Allows administrators to delete the node

Clicking on the network icon of an IP address reveals a context menu:

- View syslog events coming from this IP
- Go to the Toolbox with this IP
- Send Wake on Lan packets
- Provision device using entry from Devices-Install
- Identifies host and available services
- Discover as an SNMP device

Nodes-Toolbox

Some node related functions to troubleshoot problems.

By default client customizations for better interoperability with NeDi are shown. If you're accessing it from a client in the field, this might be of interest as well:

-  Download kitty.exe to access devices using telnet or SSH.
-  Download iperf.exe to test network throughput (requires enabling the server in System-Services, or another iperf server somewhere else).

Nodes-Traffic

This is the main Netflow module. Knowledge about nfdump and the tcpdump filter syntax is helpful here.

The netflow data uses local unix timestamps, which are not adjusted to the client's timezone, if different!

- The first selectbox lets you select the columns to be aggregated by (defaults to proto, src/dst and src/dst port)
- The 2nd determines sorting
- The 3th lets you select the flow source(s)
- The textbox allows for using a filter (some templates above)
- IPs are checked against dns, arp, nodes, network and devices tables and set an icon accordingly
- The slider adjusts the start time (can be set with datepicker by doubleclick on time field)
- You can add a graph like pie chart, sankey or RRD (latter is not adjusting to displayed traffic)
- Enabling name lookup with  uses dns and whois (storing the result in the netinfo table, which can take a moment)
- Clicking on the sources and destinations cycles the filter (src/dst ip, ip, src/dst net, net) for quick changes
- Create an alert policy from an applied by filter by clicking the  icon (requires System-Policy)

Other-Calculator (IP Calculator)

Subnet calculator for sub- and supernetting

- Check "DB Comparison" to find used and unused address ranges
- A table of subnets can be exported to XLS for further processing

Other-Converter (Number Converter)

A very simple number converter, which can be helpful in finding the correct OIDs with Def-Editor:

- Paste OIDs or string containing HEX or decimal numbers and click Show
- The values are shown in decimal, HEX and ASCII

Other-Defed (Device Definition Editor)

Generate those infamous .def files with the help of this module, to make them as accurate and reliable as possible. Email me the resulting .def files by clicking on  , if they're 100% working and I'll include them in the distribution.

You'd usually click on a sysobjid column of an unknown device in [Devices-List](#) or  in [Devices-Status](#). This will add an IP address and SNMP community along with the sysobjid you wish to take care of.

In case a .def file exists already, it's values will be filled into the form.

The  button submits IP and community, reads the existing .def and marks the sysobjid to be used as source for an unknown device with no suitable source .defs within range.

In case a .def exists with it's last sysobjid digit within -+10 of the chosen one, it'll be added to a list of potential source .defs, which can be copied as template. (a previously marked .def appears as source with green background,if none were found).

Here's some useful information on Sysobjids: [Cisco](#)

It's also recommended to watch the [DefGen Tutorial!](#)

- Hover over the input fields, to get hints on what to fill in.
- Find the most official type (there's usually a sticker with a barcode somewhere).
- Select the icon according to the GUI docs on the NeDi Homepage.
- Contact me, if you need a new OS selection.
- Some vendors use vlan community indexing to get Bridge forwarding information on the switches.
- Some vendors use twice the bandwidth to indicate fullduplex. Just use 'doublespeed' as keyword for **IF Duplex**.
- Only populate the Alias- Duplex- and Vlan- Index fields, if they're not the same as the interface indexes.
- If MAU type (1.3.6.1.2.1.26.2.1.1.11) is used, no actual duplex values are required.
- Use modifiers to multiply/divide temperature and memory if necessary. The latter also accepts % if the value reflects percentage of available memory or -% in the case of used memory.
- Add an "N" to an OID, if of the last number can vary for CPU or temperature.
- Add 1-x to bootimage, if the info is spread across several OIDs (e.g. Zyxel, ESXi)
- Use a negative custom threshold to alert if result is less than threshold
- Once you start editing the text area, the input fields above will be locked to prevent accidental input.

Other-Flower (Flower Openflows)

Openflow is a standard, which allows for a controller to directly manage flowtables on switches. This forms the foundation of Software Defined Networking (SDN) and can be used to build firewalls, loadbalancers and a lot more that we can't even think of, yet.

This module makes it easy to create and remove static flows on such an Openflow controller (right now Floodlight is supported and tested).

- Set the name or IP address of your controller in the \$flc variable at the top of the php code or simply call it with Other-Flower.php?flc=CONTROLLER
- All switches managed by the controller show up with their flows in a list on the bottom part.
- Hovering over icons and input fields reveal their purpose.
- If Other-Flower is enabled in ned.conf, you'll see its icon in Nodes-List's MAC and IP address fields, which lets you quickly add new flows based on them
- To push a new flow, enter a name for it and a priority if desired.
- Define the filter to match packets based on ingress port, source/dest MAC or IP address or UDP/TCP ports. You'll need to add 0x800 as Ethertype and 6 as protocol, if you wish to match TCP packets for example.
- Now set an action to take, which can be a destination interface, vlan and even modifying MAC or IP address or port. If you don't specify an action, the matching packet will be dropped.
- Select the switches from the list below, where you want to install the flow on and click Add.

Other-Info

Simple wrapper for phpinfo();

Other-Invoice (Invoice Generator)

Here's a way to finance NeDi's development in form of an annual contribution based on the size of your network:

- Enter your address, a comment to inform purchasing what it's for and click update
- Deselect checkboxes, if you don't want to pay for the respective items
- select a currency and click on the "Print" icon the create an invoice
- The resulting invoice can be printed by clicking on top left icon

THANKS IN ADVANCE!

Other-Noodle (Noodle Search)

This is a simple search tool (Google-like NeDi Search) to find strings in the whole database. It's usually called by the "Find IT" box in the header

Reports-Combination (Combination Reports)

This module combines actual reports from the other reporting modules in order to provide enhanced views on specific aspects:

1. Asset lists all device relevant info and the distribution of modules within
2. Population shows how the nodes are distributed across your network
3. Monitoring summarizes events and incidents
4. Error lists duplicates that shouldn't be, IF errors/discards and link mismatches

Reports-Custom (Custom Report)

This module allows for creating customized reports. Some knowledge about how databases work, is helpful here.

- The Device table is used as base for every report
- Select another table, if you don't just want to look at devices
- Define a filter (up to 4 conditions)
- Select (multiple) columns to group the results by
- Select a chart type to be displayed on top
- Use location level in combination with location columns to group on cities for example
- Use the template icons for quick examples

Reports-Devices (Device Reports)

Reports focussing on devices, their connections and configurations.

Type Distribution	Distribution of device vendors and types
Class Distribution	Distribution of device classes and their services
SW Distribution	Distribution of operating systems and software versions
Duplicate Serial#	Duplicate serial numbers of devices and modules
Duplicate IP	Duplicate mgmt IP addresses of devices
Group Distribution	Device group and mode statistics (can be VTP related or AP groups in Wlan controllers)
Configuration	CLI devices missing config and configs without changes
Device PoE	Top PoE budgets and their usage (based on Power-Ethernet MIB)
Discover History	Discover history, where each column is limited individually (use filter to narrow down the timeframe)
Device Connection	Unlinked devices and undiscovered neighbors
Connection Errors	Link mismatches based on discovery protocol information

Reports-Interfaces (Interface Reports)

Interface reports provide information on the perimeter of your network, but also reveal internal problems or misconfiguration.

Summary	Shows Top interface types and respective status
Traffic, Errors, Discards and Broadcasts	lists the busiest and most problematic interfaces of your network. Check 'Alternative Sort' to take IF speed into account of the traffic stats and the actual traffic for the errors. 'Optimize' uses absolute errors rather than those seen within the last discovery period
Port Availability	Reveals which switches can be replaced by smaller ones or which are getting really full (based on recent ingress traffic). 'Optimize' restricts this statistic to bridges and ethernet interfaces
Port Disabled	Quickly find that interface you disabled a week ago, because some infected notebook tried to attack the rest of your network
PoE Statistics	Displays top power delivery per device and interface average, based on per interface PoE information (e.g. from discovery protocol or interface MIBs)
Vlan Distribution	Generates a vlan matrix, showing number of untagged ports with an icon (1,2 and 3 or more) and number of tagged ports with background color (shaded, if untagged ports are found)

Reports-Modules (Module Reports)

Need to know how many modules of a kind you've got? This report also helps, if you need to generate a HW inventory for support contracts based on serial numbers etc.

Distribution	Presents an overview of which modules are installed in which devices
Inventory	Generates a complete list of devices and their individual modules
Print supplies	Lists print supplies sorted by availability or location (to make filling them up easier for the guy who has to go to every printer)
Virtual Machines	List all hypervisors with allocated VMs, CPUs and memory

Reports-Monitoring (Monitoring Reports)

General monitoring statistics like availability, event sources and incidents and how they're acknowledged.

Availability Distribution	Statistics of targets and their locations
Latency Statistics	Last, average and maximum latency of targets (inaccurate at the moment, sorry)
Uptime Statistics	List devices with the highest service time
Events Distribution	Statistical breakdown of events, their levels and sources
Incident Group	Distribution and duration of categorized incidents
Incident Distribution	Distribution of incidents across targets and their locations
Incident Acknowledge	Acknowledge statistics and time per user
Incident History	Log in calendar form to "spot patterns" (optimize reveals detailed view, increase limit for more years)

Reports-Networks (Interface Reports)

Find how nodes are distributed across your IP ranges or how subnets are being used.

Network Distribution	Lists discovered networks and their usage. Click optimize to verify all interface IPs and prefixes on devices with each other
Network Population	Shows all subnets (< /16) and maps IPs of nodes (green) and devices (blue) or both (yellow) into the address space. Empty DNS entries show up red

Reports-Nodes (Node Reports)

Reports focused around the anything connected to your network.

Summary	Node statistics at a glance
Node Distribution	Distribution of nodes by port and device to detect unmanaged switches or hubs
Duplicate Nodes	Shows duplicate node names (e.g. having a Wlan and Ethernet connection) or MAC addresses
Node Address	Shows duplicate or multiple IP addresses
OS & Services	Show top node OS and type statistics if nodes are identified with NeDi's scan feature
Nomads	IP and IF changes multiplied yield NeDi's nomad factor, an indicator for those who seem to travel a lot
Discover History	This history can reveal major changes or problems in your network. Each column is limited individually (use filter to narrow down the timeframe)
Empty Vlans	Unpopulated vlans can be identified and eventually removed, if not needed on particular devices

System-Database

Backup SQL tables, perform DB maintenance, export configurations as text files or other tables as CSV files. By default the complete DB structure including number of records is shown:

- Quickly view (the first 1000) entries of a table by clicking on the  , if shown
- Optimize a table with  or repair with 
- Delete all records with 

Execute

- Select a query from the "--DB List-- selectbox. Entries begin with simple SELECT statements to display entire tables, but also contain maintenance tasks towards the bottom
- "Configuration Backup" simply adds a query to select all configs, but creates a downloadable gzip archive as well
- All other select statements list the respective table contents, which can be displayed as CSV (with destination set to "plain")
-  changes IP addresses and timestamps to a human readable format and adds a timestamp to the archive name, if destination is Gzip or Bzip2
- Bzip2 needs more resources, but generally creates smaller archives.
- Depending on the amount of data you're dealing with, the module requires more memory or time to finish processing!

SQL Dump

- Select (hold down CTRL for multiple) tables to be exported in SQL format
- The resulting file can be imported again via System-Files, if Destination was set to Gzip

System-Files

This module provides the following major features:

1. Edit/View system, device configuration and ned log files
2. Import SQL data or update NeDi files
3. Manage files in html/ log, map, topo and tftpboot
4. Manage CLI command files and install templates (see Devices-Install)
5. Delete outdated RRDs (older than retire in ned.conf) to free up disk space

Editor/Viewer

- Simply choose the file you want to edit and click save, when you're done.
- You can only edit files, which are writable by the webserver.
- A device configuration can be written to "tftpboot" and used for PXE provisioning
- When editing ned.conf or nodi.conf you can click  for the password encryption pop-up
- Click on  to create a new install templates or CLI command files

Import/Update

- Select "Import DB" and upload a .sql.gz (packed) file which will replace the DB data. You can restore dumps created with System-Export for example. Create and activate a snapshot to add data from another NeDi system
- Select "Update Image" and upload an archive with alternative user icons (usr/0-99.jpg) or device panels (panel/devtype.jpg)
- Upload a ned.tgz archive and choose whether you wish to backup your existing config (check for compatibility!) or not (e.g. for patches)

Manage Files

Upload files in the appropriate area. Files in html/log can be accessed directly by clicking on file name.

- Delete files by clicking on 
- Folders in the topo section can be selected to upload a background.jpg or other files to this location. They can be used by Topology-Map as "geo" map backgrounds or Topology-Table and Assets-Location for documentation
- Click on  to create a new file in "tftpboot" for PXE provisioning

System-NeDi

Execute nedipl from the GUI. The module can be used to perform the following tasks:

1. By default the help is displayed, which reveals options and the output legend
2. Definitions shows all available .def files, sysobjids are linked to Def-Editor
3. Discover will actually find devices
4. Services scans for certain open ports on given IP addresses and uses the answers for host identification
5. Init drops and recreate the whole database, but does not remove any config files or RRDs

Double click in the output area to have it turn yellow and scroll down automatically. Do it again to turn this feature off.

Discover

This is NeDi's core. You can use this module to determine the best way to discover your network. Once you've found the right options, copy the command above the output and put it in crontab via System-Files. There are several approaches to discover a network. First the right method to use the sources needs to be found:

1. Don't add any IPs to the seedlist and check "Protocol". This discovers the default gateway of the NeDi host and any neighbors via CDP or LLDP
2. If you have firewalls or other "hurdles" separating your networks (not supporting CDP or LLDP), you need to add a seed for each island
3. Use a static seedlist and don't use any discovery protocols
4. Select "Address" from the Seed-selectbox and enter a single IP or range like 1.2.3,6,8,10-15
5. Alternatively you can click  to select Devices with the 'all' option to discover all devices in the DB
6. You can also use a query to only discover a subset and use crontab to parallelize the discoveries this way
7. To find more "exotic" devices, add the vendor strings to ouidev in nediconf and check "OUI". Discover a router connected to those devices and they'll be queued
8. You can use route tables as layer3 discovery by checking "Routes"

The behavior can be controlled with the following options:

- Select a Configuration option to back up device configurations to DB and the config folder
- Click  to skip interface info,  to avoid graphs or  to ignore nodes (any combination is possible, to speed up the discovery)
- Select "Version" to force using an SNMP version (only tested upon first discovery and the first one working is stored in DB)
- Check "Read" to re-test SNMP read access (useful to rediscover an existing device in conjunction with -V)
- Check "Write" to re-test SNMP write community strings (only tested upon first discovery, can be turned off via snmpwrite in nediconf)
- Check FQDN to use complete device names. Otherwise everything after a '.' is truncated as fqdn's can cause wrong links
- NeDi relies on unique device names. Check DevIP to use their IP addresses instead
- Select a command file from CLI-Send selectbox to have it executed on each discovered device (see System-Files for creating them)

DNS Names

- Select Address from the Seed-selectbox and enter a single IP or range like 1.2.3,6,8,10-15
- Check verbose to follow the progress of the name resolution
- Click Execute to resolve all names in that IP range
- The Network Population report in Reports-Networks leverages this information to show unused DNS records for example

Services

- Select Address from the Seed-selectbox and enter a single IP or range like 1.2.3,6,8,10-15
- Alternatively you can select Nodes and enter a query like oui ~ 'intel'
- Select Ping (1-3s timeout) to make sure an address is in use (TCP echo is used and may not work on some hosts)
- Check verbose to follow the progress of the host identification
- The 'id' option uses ssh, sendmail, http, https and netbios for host identification
- If used from CLI, additional ports can be checked like -sid,3128,5900

System-NoDi

Execute nodi.pl from the GUI.

NoDi stands for node-discovery (refer to the NeDi Guide for more information).
Make sure you edit nodi.conf before using this module.

1. By default the help is displayed, which reveals options and the output legend
2. Enter an IP address/range or select Nodes and enter a SQL query
3. Select a user to avoid trying all available ones
4. Skip what you don't need
5. Click "Execute" to start discovery

Double click in the output area to have it turn yellow and scroll down automatically. Do it again to turn this feature off.

System-Policy

This is a premium module, only available with NeDi+. Find more details [here](#)

Make sure you understand how policy actions work! You can disable all network interfaces for example, if you don't know what you're doing!

Search for 'safety on!' in libmisc.pm and toggle commenting on the 2 '\$clistat' lines, if you're confident!

This module lets you define conditions on device configurations, neighbors or learned MAC addresses and take action upon hit or miss.

The class of a policy determines where in the discovery it's processed. This is important, if you want to take action on neighbor names and learned MAC addresses for example, as only the last matching policy with an action will be executed.

Order	Class	Operator	Description
1	Neighbor Name	~ or !~	After collecting all LLDP, CDP or FDP neighbors their names are processed
2	Neighbor Type	~ or !~	Right after the names, their types are processed
3	MAC Address	~ or !~	After collecting the bridge-forward entries (MAC address table) they're processed
4	Connection Before	~ or !~	When writing the interfaces to the DB, the previous connection information (linktype) is processed to detect changes in device interconnections
-	Configuration	~ or !~	Configurations are processed with -b or -Bx, but this policy does not depend on the others above
-	Port Configuration	~ or !~	Configuration of interface contexts (e.g. in conjunction with "Connection Type")
-	Device Monitor	any	Add new devices to monitoring. If you enter - or no in target, it'll be added in maintenance mode. CPU & Mem thresholds are taken from .def, alert action is applied to target and does not create alerts itself
-	Total # of MACs	> or <	This policy refers to total # of learned MAC addresses (including those on uplinks). It does not depend on the others above as it's evaluated after writing nodes of a device
-	Packets, Bytes and Flows	> or <	Those policies are used by flowi.pl (on nfdump files) allowing for alerts on excessive or missing traffic

Stolen Nodes

1. Click on  in Nodes-Status to create a MAC policy of that node
2. Adjust Alert setting or info text and click add
3. Everytime this MAC address is found, you'll be notified according to the alert setting

Configuration Compliance

1. Select "Configuration" from the class selectbox and enter regexp to match (e.g. 'snmp-server community public')
2. Alternatively you can change the operator to '!~' to get alerts on missing configuration statements
3. Narrow down the matches by specifying a regexp for device type, location or group for example

4. Adjust Alert setting and information text and click add

Port Configuration Compliance

1. Select "Port Configuration" from the class selectbox and enter regexp to match (e.g. 'switchport mode trunk')
2. Alternatively you can change the operator to '!~' to get alerts on missing configuration statements
3. Narrow down the matches by specifying a regexp for device type or connection-type = Phone for example
4. Adjust Alert setting and information text and click add

Device Monitor

1. Select "Device Monitor" from the class selectbox, enter "-" or "no" as target to set test to none or specify a test like "ping"
2. If you leave target blank it'll default to uptime for SNMP devices and icmp for non-SNMP ones
3. Narrow down the matches by specifying a regexp for device type, location or group for example
4. Adjust Alert setting for the monitored target (repeat options are not supported yet) and click add
5. Dependencies are not resolved automatically and should be configured in Monitoring-Setup

PoE Police

1. Add a Neighbor Policy with the "Skip Action" to allow Poe delivery to phones or controlled APs.
2. Add a MAC Policy to either match (~) on particular addresses or enter a '.' to match any
3. Narrow down the matches by specifying a regexp for device type, location or group for example
4. Optionally select an interface condition to only trigger if PoE was active in the previous discovery
5. Select 'PoE Disabled' Action and add a reset policy by selecting a timeframe after which PoE should be re-enabled
6. Upon the first discovery, when its timestamp is in the past, the reset policy is executed to restore PoE delivery
7. Adjust Alert setting and information text and click add

Link Alerts

1. Add a "Connection Before" Policy and enter "D\$" to match regular devices
2. Select the "Status Change" condition
3. Alternatively you can select a connection type to match the current status (e.g. if someone replace a device with a phone)
4. Adjust Alert setting and information text and click add

Traffic

1. In Nodes-Traffic choose columns to aggregate (group), sorting, source and a filter then click Show
2. The System-Policy icon appears, click it
3. Set operator and a threshold, then specify how you want to get notified
4. This policy creates events with class 'spr' (System-Policy-Traffic) using its id as source

General Topics

- A policy cannot be edited, but copied by clicking on and then added again
- A policy can be disabled by clicking on (and enabled respectively)
- A policy can be removed by clicking on
- The "Skip Action" withlists a port, thus avoids any other action to be executed
- You should add a reset action to recover disabled ports or re-enable PoE after a given time (they're added with status new and a timestamp set in the future, when the action takes place)
- The reset action is performed, when its timestamp is in the past
- If skippol or -S contains p or F no actions will take place, except those of reset policies
- If skippol or -S contains P policies are completely ignored
- Thoroughly test policies without actions before 'arming' them with one
- Actions are supported on IOS and ProCurve devices at the moment (changed config is not saved to flash)
- In case an error occurred while getting device neighbors, the skip action is applied to concerned interfaces (inhibiting erratic actions)

- The information text is used in events, emails and sms, but also serves as comment in the policy list (e.g. if no Alert is selected)
- Actions commands are written to pol_files in the cli folder and can be reviewed along with their logs in System-Files
- By default a policy summary report is shown

System-Services (NeDi Services)

View processes and resources of your NeDi host and start or stop certain services.

- The top section shows and controls NeDi related services.
- The lower section shows all running processes and some system stats.
- Click on  to stop or  start a service
- This only works, if the services don't need to open any privileged ports (< 1024). Of course you could run the webserver as root, but that can create security risks! Therefore NeDi's Syslog (syslog.pl) and snmptrapd run on high-ports and usually are redirected by an internal firewall.

Discovery

Depending on the size and topology of your network, it makes sense to run several discovery threads at the same time.

- Do this by dividing the network in a few sections using borders and different seedfiles and add crontab entries accordingly
- On the far right you see the discovery status (# of threads is revealed by hovering over )
- In case a discovery terminated unexpectedly, you can reset it by clicking on .

System-Snapshot

This module lets you take a snapshot of the current database. This may be very helpful for a network migration for example, as you can go back in time and examine your network prior any changes

In addition you can import a NeDi database from a completely different network for review, without affecting your "real" data.

Adding a Snapshot

- Enter a suffix to identify your snapshot. By default a timestamp is filled in.
- Provide DB admin user (usually root) and password.
- Click the "Add" button to copy the current database to the snapshot (might take a while).

Activating a Snapshot

- The database used in the current session is indicated by  .
- Click on  in the snapshot list to activate either the main database or a snapshot.
- The  logo on the top left is replaced by  to remind you, that you're working in a snapshot now. Hover over it to reveal which one.
- Alarm sounds and rrdgraphs are turned off as well to avoid any confusion until you select the main database again (usually 'nedi').
- You can manipulate data in a snapshot, but it won't have an effect on the current database, since the discovery keeps using the main database.
- This applies for importing a DB with System-Files as well, meaning you can actually import a completely different database

Deleting a Snapshot

- Click on  to delete a snapshot (only shown on inactive snapshots).
- After confirmation the snapshot will be deleted and its disk space freed up.

Topology-Linked (Link Editor)

Edit static links here, if the discovery protocols don't deliver satisfying results.

- Select a device, any existing links of this device are shown automatically.
- Select the desired interface (green indicates link-status is up)
- Do the same for the neighbour.
- Click 'Add' to create this and the reverse link.
- Both links need to be deleted separately, if they're no longer required.
- The right  deletes the link and shows the neighbor for easier deletion of the opposite link.
- Select the link type, if you just want to see what's in the DB.
- Select Isolated to identify links, without device in the DB.
- By default the "Connection Error" report is shown

Topology-Links (Link List)

List links of the devices.

- By default the "Device Connection" report is shown

Topology-Map

This module was intended for documentation purposes, even though it features interactive handling now. It can also be used to observe traffic, errors, broadcasts, discards, cpu usage or temperature of devices. Maps are written on a per user basis to html/log or used in Monitoring-Map. Upon accessing this module the last map will be displayed without interactive features.

- Graphs are only drawn in PNG and only for the 1st time the map is generated, because they'll be deleted afterwards. This may be a problem, if you wish to save the picture (screenshot always works, though).
- Alternatively SVG or even interactive D3js maps can be created. Drag a node to fix it on the canvas. Doubleclick to let it float again.
- "PNG"png" generates truecolor, "8bit" generates 256 color png images respectively. They can be included in the combination report or various lists.
- SVG is used for vector drawings, which can be imported by other applications. You probably want to use "shapes" instead of "icons" unless you copy them into the right place on the destination.
- Hover over the input fields and icons to get hints.
- If you enable dynamic-edit (far right walk-icon above "Execute"), the map will be redrawn upon any input and fields are disabled if they're of no use with the current settings. This works best if the browser supports HTML5 properly.
- To get a feel for this rather complex part of NeDi, click on the  icons in other modules to create maps in different contexts.
- A "bgmap" map finds the best suited background image automatically. E.g. the regional one, if you're only drawing the "Shire" region **and** you've uploaded a background.jpg to topo/Shire with System-Files for example.
- Assuming you've edited this region with Loced before, it'll now use the city coordinates, you've entered to put the city icons. If you draw at building level, they'll simply be arranged around the city coordinates in a ring.
- Click the Monitor button to add current map to Monitoring-Map

Internally maps are calculated using polar coordinates (except in "layer" mode), where each level (e.g. a city) forms a ring. Devices are arranged based on their neighbors. This does not always work out, but generally yields acceptable results after some tweaking. The following sections explain how this is done.

Filter

- Layer mode: The 4 fields correspond to core,distribution, access and access2 layers and select devices for each desired layer
- All others: Same as the filter section in the list modules

Main

- Title of Map
- Size (can be adjusted in URI) and output format of map
- For hierarchical maps use "bld" (draws buildings with floors) or "ring" (draws buildings as circles). This lets you draw region, city or building level maps leveraging NeDi's SNMP location scheme.
- The "bgmap" type relies uploaded backgrounds and information you've added with Loced
- Alternatively you can select "flat" which still gives you the ability draw maps without any location awareness but display non-SNMP devices or even nodes
-  adds an additional conditions to filter on SNMP devices only
-  defines the center of your map
- Rotate map at top, city or building level (shift layers on X-axis in "layer" mode)

Layout

-  defines how links are presented. Length/level determines how much shorter a link between buildings is going to be than a link between cities for example. The next field defines the offset from the link endpoint for interface

information (if displayed)

-  length sets the top-level link length (can be looked at zoom-level too). They're drawn "straight" as default, but sometimes you'd prefer an "arc"
-  Link Information can be bandwidth or even a RRD graph. It can be moved away from the center, if it gets in the way of other items
-  defines how map-nodes are represented. Positive numbers use the position in the topology, negative just number of neighbors to determine its distance from the center. In "layer" mode this only set y-amplitude for access layer alternating
-  Floor size sets the building size when actual devices are drawn in hierarchical maps. This value can be as small as 8 if "Tiny Shapes" is selected above to generate a [bird-eye view of your network](#)
- Columns lets you control how wide those buildings are represented

Show

- Select various details to show up on the map

Topology-Multicast

Simple tool to show PIM routing table on a Cisco® router or IGMP info on a ProCurve® switch.

Topology-Networks (Network List)

List IPv4 and IPv6 addresses by VRFs for example.

- If an IPv4 address is empty, the entry is an IPv6 address. Hover over the network icon to reveal it's class.
- The status of the corresponding interface or VRF/VPN is shown with the network icon (stays white if not available).
- You can search for networks using CIDR notation (1.2.3.4/24) or regexps (^1.2.3)
- Some prefixes show 0, if NeDi couldn't read them properly from the device.

Topology-Routes (Routes Toolbox)

This is the former Realtime Routes module, which now provides 3 modes of operation:

- List routes stored in the database (NeDi 1.8 feature)
- Display the routing table of a device, by selecting one with the right selectbox and clicking "Show"
- Trace a route by selecting source, destination and clicking "Route"

Topology-Spanningtree (Realtime Spanningtree)

Displays Spanningtree status of a layer2 device.

- Select switch from list.
- Select vlan, to display per vlan spanningtree information, if applicable.
- Additionally display traffic graphs (if RRD is enabled) to verify operation.
- The interface pointing to the root bridge is indicated with 
- The MAC address of the root bridge can be searched for by clicking on 
- In the IF status column you can see if a port is blocking or forwarding etc.

Topology-Table

If your devices are configured with SNMP location information according to [NeDi's scheme](#), you can drill down into your network in a tabular fashion here.

Those buttons on the top right help navigating and reveal more information:

- Click to get to the top, to region, to city, or to building level.
- displays number of devices per location
- adds node population per location
- adds free access ports per location
- Clicking on location names lists all its devices (the displayed width is set in User-Profile).
- You can "paint" important buildings red(ish) with rebuild in ned.conf.
- A street address can have several buildings, if bldsep is configured correctly. A digit showing the amount of sub-buildings is added, if there are more than one.

The next button cycles the display of your sites. The state is preserved within the session and is used in Monitoring-Health as well:

- Switches to small icons (good for displaying hundreds of sites)
- Shows NeDi maps (for a glance inside)
- Shows static maps which are cached in the "topo/" tree
- Adds weather information for cities, with that you know when it's down because of a thunderstorm.
- Reverts to the default icon display.

Building Level

Inside a building you get to see the devices on each floor and room. If you specified the rack and rack-unit, the room name becomes a link which takes you to the rackview.

- Clicking on a floor lists all matching devices.
- Toggles displaying non-SNMP devices.
- Shows device panels instead of icons.
- If photos or documents named Building-Floor-something (ignoring non-word characters) are found in topo/Region/City they're presented with an icon underneath the floor label. Clicking on them reveals the photo in a popup window or opens the file.

User-Chat

A very simple chat interface for NeDi users. You can also run `stati.pl` every week or so and it will add statistics to the chat, similar to a bot in an IRC channel.

- Hover over a user image, if you're unsure who it is
- The greener a message the more recent
- Your events are a little brighter than those of others

User-Management

Admins can add and manage users and their groups here. In addition a Device Filter can be applied to a non-admin user to restrict his access to the network.

- Assign groups by clicking on the Group icons.
- Select device filter, if required. Enter a - to clear, click on  to verify
- Delete an account by clicking 
- Use  to reset a lost password
- Other icons list devices, assets and events related to the user

User-Profile

This is your starting page, when signing in (except for December ;-). It also serves to display any administrative notifications and to edit your password and information.

- You'll only receive monitoring emails and SMS, if you enter your info accordingly  and are in the monitoring group
-  lets list tables remember the column settings and adds "breadcrumbs" to the header. If you are using an ssh and telnet plugin that recognizes plain IP addresses, you can turn off any IP links as well
-  # of events or report entries are shown in certain modules
-  # of columns to be shown in topology table views
-  label length in tables and maps
- Language and theme are not updated immediately and require a reload
-  lets you edit the Admin Message (if you're an admin)

User-Radius

This is a NeDi Enterprise module, only available through a certified partner

Managers can add Radius groups and users with this module (requires radius database settings in ned.conf).

- In the Vlan section of [Devices-Status](#) click on  to prefill the group fields
- Change to your needs and click "Add" to create a groupreply entry
- For MAC authentication, filter desired nodes in Nodes-List
- Select a group (or not) and click "Radius" to add visible nodes to the radius DB
- Add other users by entering name, password im User-Radius and select group (or not) and click "Add"
- Create a user list by using the filter and clicking "Show"
- By default the available groups and user-group mappings are shown

Table of Contents

Introduction	2
Installation Instructions	3
General Overview	4
Architecture	5
Functional Breakdown	6
Terminology	7
Network Management	9
Prerequisites	9
Topology Awareness	10
Configuration Backup	12
Device Modules	13
Network Population (Nodes)	14
Edit ned.conf	15
Edit seedlist	16
Discover the Network	17
Edit crontab	19
Asset Discovery	20
Troubleshooting	21
Frontend Overview	22
REST API	22
Managing Assets	23
The NeDi GUI	24
Lists	25
Monitoring	26
Reporting	28
GUI Modules	29
Assets-List	30
Assets-Locations (Location List)	31
Assets-Loced (Location Editor)	32
Assets-Management	33
Devices-Config	34
Devices-Doctor (Device Doctor)	35
Devices-Graph	36
Devices-Install	37
Devices-Interfaces (Interface List)	38
Devices-List	39
Devices-Modules (Module List)	40
Devices-Status (Device Status)	41
Devices-Translator (Configuration Translator)	43
Devices-Vlans (Vlan List)	45
Devices-Write	46
Monitoring-Events	47
Monitoring-Health	48
Monitoring-History	49

Monitoring-Incidents (Incident List)	50
Monitoring-Map	51
Monitoring-Master	52
Monitoring-Setup	53
Nodes-Create	54
Nodes-List	55
Nodes-RogueAP (RogueAP List)	56
Nodes-Status (Node Status)	57
Nodes-Toolbox	58
Nodes-Traffic	59
Other-Calculator (IP Calculator)	60
Other-Converter (Number Converter)	61
Other-Defed (Device Definition Editor)	62
Other-Flower (Flower Openflows)	63
Other-Info	64
Other-Invoice (Invoice Generator)	65
Other-Noodle (Noodle Search)	66
Reports-Combination (Combination Reports)	67
Reports-Custom (Custom Report)	68
Reports-Devices (Device Reports)	69
Reports-Interfaces (Interface Reports)	70
Reports-Modules (Module Reports)	71
Reports-Monitoring (Monitoring Reports)	72
Reports-Networks (Interface Reports)	73
Reports-Nodes (Node Reports)	74
System-Database	75
System-Files	76
System-NeDi	77
System-NoDi	79
System-Policy	80
System-Services (NeDi Services)	83
System-Snapshot	84
Topology-Linked (Link Editor)	85
Topology-Links (Link List)	86
Topology-Map	87
Topology-Multicast	89
Topology-Networks (Network List)	90
Topology-Routes (Routes Toolbox)	91
Topology-Spanningtree (Realtime Spanningtree)	92
Topology-Table	93
User-Chat	94
User-Management	95
User-Profile	96
User-Radius	97