

# @serve.zone/nupst

a command-line tool that monitors SNMP-enabled UPS devices and initiates system shutdown when power outages are detected and battery levels are low.

- [readme.md for @serve.zone/nupst](#)
- [changelog.md for @serve.zone/nupst](#)

# readme.md for @serve.zone/nupst

## ⚡ NUPST — Network UPS Shutdown Tool

**Keep your systems safe when the power goes out.** NUPST is a lightweight, battle-tested CLI tool that monitors UPS devices via SNMP or NUT (UPSD) and orchestrates graceful shutdowns during power emergencies — including Proxmox VMs, LXC containers, and the host itself.

Distributed as **self-contained binaries** with zero runtime dependencies. No Node.js, no Python, no package managers. Just download and run.

## Issue Reporting and Security

For reporting bugs, issues, or security vulnerabilities, please visit [community.foss.global/](https://community.foss.global/). This is the central community hub for all issue reporting. Developers who sign and comply with our contribution agreement and go through identification can also get a [code.foss.global/](https://code.foss.global/) account to submit Pull Requests directly.

## ▣ Features

- **▣ Multi-UPS Support** — Monitor multiple UPS devices from a single daemon
- **▣ Dual Protocol Support** — SNMP (v1/v2c/v3) for network UPS + UPSD/NIS for USB-connected UPS via NUT
- **▣ Proxmox Integration** — Gracefully shut down QEMU VMs and LXC containers before host shutdown
- **▣ Group Management** — Organize UPS devices into groups with flexible operating modes

- **Redundant Mode** — Only trigger actions when ALL UPS devices in a group are critical
- **Non-Redundant Mode** — Trigger actions when ANY UPS device is critical
- **⚙️ Action System** — Define custom responses with flexible trigger conditions
  - Battery & runtime threshold triggers
  - Power status change triggers
  - Webhook notifications (POST/GET)
  - Custom shell scripts
  - Proxmox VM/LXC shutdown
  - Configurable shutdown delays
- **📦 Multiple UPS Brands** — CyberPower, APC, Eaton, TrippLite, Liebert/Vertiv, and custom OID configurations
- **📦 HTTP API** — Optional JSON status endpoint with token authentication
- **📦 Pause/Resume** — Temporarily suppress actions during maintenance windows
- **📦 Network Loss Detection** — Detects unreachable UPS devices and prevents false shutdowns
- **📦 Power Metrics** — Monitor output load, power (watts), voltage, and current
- **📦 Single Binary** — Zero runtime dependencies. Download, `chmod +x`, run.
- **📦 Cross-Platform** — Linux (x64, ARM64), macOS (Intel, Apple Silicon), Windows

## 📦 Quick Start

### One-Line Installation

```
curl -sSL https://code.foss.global/serve.zone/nupst/raw/branch/main/install.sh | sudo bash
```

### Initial Setup

```
# 1. Add your first UPS device (interactive wizard)
sudo nupst ups add

# 2. Test the connection
nupst ups test

# 3. Enable and start monitoring
sudo nupst service enable
sudo nupst service start
```

```
# 4. Check status
nupst service status
```

**That's it!** Your system is now protected. ☑

# ☑ Installation

## Automated Installer (Recommended)

```
curl -sSL https://code.foss.global/serve.zone/nupst/raw/branch/main/install.sh | sudo bash
```

### What it does:

1. Detects your platform (OS + architecture)
2. Downloads the latest pre-compiled binary
3. Installs to `/opt/nupst/nupst`
4. Creates symlink at `/usr/local/bin/nupst`
5. Preserves existing configuration

### Options:

```
# Install specific version
curl -sSL https://code.foss.global/serve.zone/nupst/raw/branch/main/install.sh | \
  sudo bash -s -- --version v5.0.0

# Custom installation directory
curl -sSL https://code.foss.global/serve.zone/nupst/raw/branch/main/install.sh | \
  sudo bash -s -- --install-dir /usr/local/nupst

# Show help
curl -sSL https://code.foss.global/serve.zone/nupst/raw/branch/main/install.sh | bash -s -- --
help
```

## Manual Installation

Download the binary for your platform from [releases](#):

Platform	Binary
----------	--------

Linux x64	nupst-linux-x64
Linux ARM64	nupst-linux-arm64
macOS Intel	nupst-macos-x64
macOS Apple Silicon	nupst-macos-arm64
Windows x64	nupst-windows-x64.exe

```
# Download (replace with your platform)
curl -sSL https://code.foss.global/serve.zone/nupst/releases/download/v5.0.0/nupst-linux-x64 -
o nupst
chmod +x nupst
sudo mv nupst /usr/local/bin/nupst
```

## Via npm

```
npm install -g @serve.zone/nupst
```

“ This downloads the appropriate pre-compiled binary for your platform during installation.

## Verify Installation

```
nupst --version
nupst help
```

## ☐ CLI Reference

### Command Structure

```
nupst <command> [subcommand] [options]
```

## Global Options

Flag	Description
<code>--version</code> , <code>-v</code>	Show version
<code>--help</code> , <code>-h</code>	Show help
<code>--debug</code> , <code>-d</code>	Enable debug mode (verbose SNMP/UPSD logging)

## Service Management

```
nupst service enable      # Install and enable systemd service
nupst service disable    # Stop and disable systemd service
nupst service start      # Start the service
nupst service stop       # Stop the service
nupst service restart    # Restart the service
nupst service status     # Show service and UPS status
nupst service logs       # Tail live service logs (Ctrl+C to exit)
```

## UPS Device Management

```
nupst ups add             # Add a new UPS device (interactive wizard)
nupst ups edit [id]      # Edit a UPS device
nupst ups remove <id>    # Remove a UPS device
nupst ups list           # List all UPS devices
nupst ups test           # Test all UPS connections
```

During `nupst ups add`, you'll choose a communication protocol:

- **SNMP** — For network-attached UPS with an SNMP agent (default)
- **UPSD/NIS** — For USB-connected UPS managed by a local NUT server

## Group Management

```
nupst group add          # Create a new UPS group
nupst group edit <id>    # Edit a group
nupst group remove <id>  # Remove a group
nupst group list         # List all groups
```

## Action Management

```
nupst action add <target-id>          # Add action to a UPS or group
nupst action remove <target-id> <idx> # Remove an action by index
nupst action list [target-id]         # List actions (optionally for a target)
```

## Pause/Resume

Temporarily suppress actions during maintenance (UPS polling continues):

```
nupst pause                          # Pause indefinitely
nupst pause --duration 30m           # Pause for 30 minutes (auto-resume)
nupst pause --duration 2h            # Pause for 2 hours
nupst pause --duration 1d            # Pause for 1 day (max: 24h)
nupst resume                          # Resume immediately
```

When paused:

- UPS polling continues (status is still visible)
- All actions are suppressed (no shutdowns, webhooks, scripts)
- The HTTP API response includes `"paused": true`
- Status display shows a `[PAUSED]` indicator

## Feature Management

```
nupst feature httpServer             # Configure HTTP JSON status API
```

## Other Commands

```
nupst config show                    # Display current configuration
nupst upgrade                         # Upgrade to latest version (requires root)
nupst uninstall                       # Completely remove NUPST (requires root)
```

## Configuration

NUPST stores configuration at `/etc/nupst/config.json`. The easiest way to configure is through the interactive CLI commands, but you can also edit the JSON directly.

# Example Configuration

```
{
  "version": "4.3",
  "checkInterval": 30000,
  "httpServer": {
    "enabled": true,
    "port": 8080,
    "path": "/ups-status",
    "authToken": "your-secret-token"
  },
  "upsDevices": [
    {
      "id": "ups-main",
      "name": "Main Server UPS",
      "protocol": "snmp",
      "snmp": {
        "host": "192.168.1.100",
        "port": 161,
        "community": "public",
        "version": 1,
        "timeout": 5000,
        "upsModel": "cyberpower",
        "runtimeUnit": "ticks"
      },
    },
    "actions": [
      {
        "type": "proxmox",
        "triggerMode": "onlyThresholds",
        "thresholds": { "battery": 30, "runtime": 15 },
        "proxmoxHost": "localhost",
        "proxmoxPort": 8006,
        "proxmoxTokenId": "root@pam!nupst",
        "proxmoxTokenSecret": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"
      },
      {
        "type": "shutdown",
        "triggerMode": "onlyThresholds",
        "thresholds": { "battery": 20, "runtime": 10 },
      }
    ]
  }
}
```

```
        "shutdownDelay": 10
    }
],
"groups": ["datacenter"]
},
{
    "id": "ups-usb",
    "name": "Local USB UPS",
    "protocol": "upsd",
    "upsd": {
        "host": "127.0.0.1",
        "port": 3493,
        "upsName": "ups",
        "timeout": 5000
    },
    "actions": [
        {
            "type": "shutdown",
            "triggerMode": "onlyThresholds",
            "thresholds": { "battery": 15, "runtime": 5 },
            "shutdownDelay": 5
        }
    ],
    "groups": []
}
],
"groups": [
    {
        "id": "datacenter",
        "name": "Data Center",
        "mode": "redundant",
        "description": "Redundant UPS setup",
        "actions": [
            {
                "type": "shutdown",
                "triggerMode": "onlyThresholds",
                "thresholds": { "battery": 10, "runtime": 5 },
                "shutdownDelay": 15
            }
        ]
    }
]
```

```

    ]
  }
]
}

```

# UPS Device Configuration

## Protocol Selection

Each UPS device has a `protocol` field:

Protocol	Use Case	Default Port
<code>snmp</code>	Network-attached UPS with SNMP agent	161
<code>upsd</code>	USB-connected UPS via local NUT server	3493

## SNMP Settings (`snmp` object)

Field	Description	Values / Default
<code>host</code>	IP address or hostname	e.g., <code>"192.168.1.100"</code>
<code>port</code>	SNMP port	Default: <code>161</code>
<code>version</code>	SNMP version	<code>1</code> , <code>2</code> , or <code>3</code>
<code>timeout</code>	Timeout in milliseconds	Default: <code>5000</code>
<code>upsModel</code>	UPS brand/model	<code>cyberpower</code> , <code>apc</code> , <code>eaton</code> , <code>tripplite</code> , <code>liebert</code> , <code>custom</code>
<code>runtimeUnit</code>	Battery runtime unit	<code>minutes</code> , <code>seconds</code> , or <code>ticks</code> (1/100s). Overrides auto-detection
<code>community</code>	Community string (v1/v2c)	Default: <code>"public"</code>

**SNMPv3 fields** (when `version: 3`):

Field	Description	Values
<code>securityLevel</code>	Security level	<code>noAuthNoPriv</code> , <code>authNoPriv</code> , <code>authPriv</code>
<code>username</code>	Authentication username	—
<code>authProtocol</code>	Auth protocol	<code>MD5</code> or <code>SHA</code>
<code>authKey</code>	Auth password	—
<code>privProtocol</code>	Encryption protocol	<code>DES</code> or <code>AES</code>

Field	Description	Values
<code>privKey</code>	Encryption password	—

## UPSD/NIS Settings (`upsd` object)

For USB-connected UPS via [NUT \(Network UPS Tools\)](#):

Field	Description	Default
<code>host</code>	NUT server address	<code>127.0.0.1</code>
<code>port</code>	NUT UPSD port	<code>3493</code>
<code>upsName</code>	NUT device name	<code>ups</code>
<code>timeout</code>	Connection timeout (ms)	<code>5000</code>
<code>username</code>	Optional auth username	—
<code>password</code>	Optional auth password	—

**NUT variables mapped:** `ups.status`, `battery.charge`, `battery.runtime`, `ups.load`, `ups.realpower`, `output.voltage`, `output.current`

## Action Configuration

Actions define automated responses to UPS conditions. They run **sequentially in array order**, so place Proxmox actions before shutdown actions.

### Action Types

Type	Description
<code>shutdown</code>	Graceful system shutdown with configurable delay
<code>webhook</code>	HTTP POST/GET notification to external services
<code>script</code>	Execute custom shell scripts from <code>/etc/nupst/</code>
<code>proxmox</code>	Shut down Proxmox QEMU VMs and LXC containers via REST API

### Common Fields

Field	Description	Values / Default
<code>type</code>	Action type	<code>shutdown</code> , <code>webhook</code> , <code>script</code> , <code>proxmox</code>
<code>thresholds</code>	Battery and runtime limits	<code>{ "battery": 0-100, "runtime": minutes }</code>

Field	Description	Values / Default
<code>triggerMode</code>	When to trigger	See Trigger Modes below

## Trigger Modes

Mode	Description
<code>onlyPowerChanges</code>	Only when power status changes (online ↔ onBattery)
<code>onlyThresholds</code>	Only when battery or runtime thresholds are violated
<code>powerChangesAndThresholds</code>	On power changes OR threshold violations (default)
<code>anyChange</code>	On every polling cycle

## Shutdown Action

```
{
  "type": "shutdown",
  "thresholds": { "battery": 20, "runtime": 10 },
  "triggerMode": "onlyThresholds",
  "shutdownDelay": 10
}
```

Field	Description	Default
<code>shutdownDelay</code>	Seconds to wait before shutdown	5

## Webhook Action

```
{
  "type": "webhook",
  "thresholds": { "battery": 30, "runtime": 15 },
  "triggerMode": "powerChangesAndThresholds",
  "webhookUrl": "https://hooks.slack.com/services/...",
  "webhookMethod": "POST",
  "webhookTimeout": 10000
}
```

Field	Description	Default
<code>webhookUrl</code>	URL to call	Required
<code>webhookMethod</code>	HTTP method	POST
<code>webhookTimeout</code>	Timeout in ms	10000

# Script Action

```
{
  "type": "script",
  "thresholds": { "battery": 25, "runtime": 10 },
  "triggerMode": "onlyThresholds",
  "scriptPath": "pre-shutdown.sh",
  "scriptTimeout": 60000
}
```

Field	Description	Default
<code>scriptPath</code>	Script filename in <code>/etc/nupst/</code>	Required
<code>scriptTimeout</code>	Execution timeout in ms	<code>60000</code>

# Proxmox Action

Gracefully shuts down QEMU VMs and LXC containers on a Proxmox node before the host is shut down.

```
{
  "type": "proxmox",
  "thresholds": { "battery": 30, "runtime": 15 },
  "triggerMode": "onlyThresholds",
  "proxmoxHost": "localhost",
  "proxmoxPort": 8006,
  "proxmoxTokenId": "root@pam!nupst",
  "proxmoxTokenSecret": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
  "proxmoxExcludeIds": [100, 101],
  "proxmoxStopTimeout": 120,
  "proxmoxForceStop": true,
  "proxmoxInsecure": true
}
```

Field	Description	Default
<code>proxmoxHost</code>	Proxmox API host	<code>localhost</code>
<code>proxmoxPort</code>	Proxmox API port	<code>8006</code>
<code>proxmoxNode</code>	Proxmox node name	Auto-detect via hostname
<code>proxmoxTokenId</code>	API token ID (e.g. <code>root@pam!nupst</code> )	Required
<code>proxmoxTokenSecret</code>	API token secret (UUID)	Required

Field	Description	Default
<code>proxmoxExcludeIds</code>	VM/CT IDs to skip	<code>[]</code>
<code>proxmoxStopTimeout</code>	Seconds to wait for graceful shutdown	<code>120</code>
<code>proxmoxForceStop</code>	Force-stop VMs/CTs that don't shut down	<code>true</code>
<code>proxmoxInsecure</code>	Skip TLS verification (self-signed certs)	<code>true</code>

### Setting up the API token on Proxmox:

```
# Create token with full privileges (no privilege separation)
pveum user token add root@pam nupst --privsep=0
```

“ **⚠ Important:** Place the Proxmox action **before** the shutdown action in the actions array so VMs are stopped before the host shuts down.

## Group Configuration

Groups coordinate actions across multiple UPS devices:

Field	Description	Values
<code>id</code>	Unique group identifier	—
<code>name</code>	Human-readable name	—
<code>mode</code>	Group operating mode	<code>redundant</code> , <code>nonRedundant</code>
<code>description</code>	Optional description	—
<code>actions</code>	Array of action configurations	—

### Group Modes:

- **redundant** — Actions trigger only when ALL UPS devices in the group are critical. Use for setups with backup power units.
- **nonRedundant** — Actions trigger when ANY UPS device is critical. Use when all UPS units must be operational.

## HTTP Server Configuration

```
# Interactive setup
sudo nupst feature httpServer
```

```
{
  "httpServer": {
    "enabled": true,
    "port": 8080,
    "path": "/ups-status",
    "authToken": "your-secret-token"
  }
}
```

### Query the API:

```
# Bearer token
curl -H "Authorization: Bearer your-secret-token" http://localhost:8080/ups-status

# Query parameter
curl "http://localhost:8080/ups-status?token=your-secret-token"
```

### Response format:

```
{
  "upsDevices": [
    {
      "id": "ups-main",
      "name": "Main Server UPS",
      "powerStatus": "online",
      "batteryCapacity": 100,
      "batteryRuntime": 45,
      "outputLoad": 23,
      "outputPower": 115,
      "outputVoltage": 230.5,
      "outputCurrent": 0.5,
      "consecutiveFailures": 0,
      "unreachableSince": 0,
      "lastStatusChange": 1729685123456,
      "lastCheckTime": 1729685153456
    }
  ],
}
```

```
"paused": false
}
```

When monitoring is paused:

```
{
  "upsDevices": [...],
  "paused": true,
  "pauseState": {
    "pausedAt": 1729685123456,
    "pausedBy": "cli",
    "resumeAt": 1729686923456
  }
}
```

## ☐ Network Loss Detection

NUPST tracks communication failures per UPS device:

- After **3 consecutive failures**, the UPS status transitions to `unreachable`
- **Shutdown actions will NOT fire** on `unreachable` — this prevents false shutdowns from network glitches
- Webhook and script actions still fire, allowing you to send alerts
- When connectivity is restored, NUPST logs a recovery event with downtime duration
- The failure counter is capped at 100 to prevent overflow

**Power status values:** `online` | `onBattery` | `unknown` | `unreachable`

## ☐ Monitoring

### Status Display

```
$ nupst service status
```

```
UPS Devices (2):
```

- ```
✓ Main Server UPS (online - 100%, 3840min)
  Host: 192.168.1.100:161 (SNMP)
```

Groups: Data Center

Action: proxmox (onlyThresholds: battery<30%, runtime<15min)

Action: shutdown (onlyThresholds: battery<20%, runtime<10min, delay=10s)

✓ Local USB UPS (online - 95%, 2400min)

Host: 127.0.0.1:3493 (UPSD)

Action: shutdown (onlyThresholds: battery<15%, runtime<5min, delay=5s)

Groups (1):

□ Data Center (redundant)

UPS Devices (1): Main Server UPS

Action: shutdown (onlyThresholds: battery<10%, runtime<5min, delay=15s)

## Live Logs

nupst service logs

[2026-02-20 10:30:15] □ NUPST daemon started

[2026-02-20 10:30:15] ✓ Connected to Main Server UPS (192.168.1.100)

[2026-02-20 10:30:45] □ Status check: All systems normal

[2026-02-20 10:31:15] △ Main Server UPS on battery (85%, 45min remaining)

[2026-02-20 10:35:00] △ UPS Unreachable: Backup UPS (3 consecutive failures)

[2026-02-20 10:37:30] ✓ UPS Recovered: Backup UPS (downtime: 2m 30s)

## □ Security

### Architecture

- **Single Binary** — Self-contained executable with zero runtime dependencies
- **Minimal Attack Surface** — Compiled Deno binary with only essential functionality
- **Reduced Supply Chain Risk** — Pre-compiled binaries with SHA256 checksums
- **No Telemetry** — No data sent to external servers

### SNMP Security

Full SNMPv3 support with authentication and encryption:

| Security Level | Description                                                  |
|----------------|--------------------------------------------------------------|
| noAuthNoPriv   | No authentication, no encryption                             |
| authNoPriv     | MD5/SHA authentication without encryption                    |
| authPriv       | Authentication + DES/AES encryption <input type="checkbox"/> |

## Network Security

- Connects only to UPS devices and optionally Proxmox on local network
- HTTP API disabled by default; token-required when enabled
- No external internet connections

## Verifying Downloads

```
curl -sSL https://code.foss.global/serve.zone/nupst/releases/download/v5.0.0/nupst-linux-x64 -o nupst
curl -sSL https://code.foss.global/serve.zone/nupst/releases/download/v5.0.0/SHA256SUMS.txt -o SHA256SUMS.txt
sha256sum -c SHA256SUMS.txt --ignore-missing
```

## Supported UPS Models

### SNMP-based

| Brand          | Config Value | Notes                                |
|----------------|--------------|--------------------------------------|
| CyberPower     | cyberpower   | Full support including power metrics |
| APC            | apc          | Smart-UPS, Back-UPS series           |
| Eaton          | eaton        | Eaton/Powerware UPS                  |
| TrippLite      | tripplite    | SmartPro and similar                 |
| Liebert/Vertiv | liebert      | GXT, PSI series                      |
| Custom         | custom       | Provide your own OID mappings        |

**Custom OIDs example:**

```
{
  "upsModel": "custom",
  "runtimeUnit": "seconds",
  "customOIDs": {
    "POWER_STATUS": "1.3.6.1.4.1.1234.1.1.0",
    "BATTERY_CAPACITY": "1.3.6.1.4.1.1234.1.2.0",
    "BATTERY_RUNTIME": "1.3.6.1.4.1.1234.1.3.0"
  }
}
```

“ **Tip:** If your UPS (e.g., HPE, Huawei) reports runtime in seconds instead of minutes, set `"runtimeUnit": "seconds"`. For CyberPower-style TimeTicks (1/100 second), use `"ticks"`. When omitted, NUPST auto-detects based on `upsModel`.

## UPSD/NIS-based

Any UPS supported by [NUT \(Network UPS Tools\)](#) — this covers **hundreds of models** from virtually every manufacturer, including USB-connected devices. Check the [NUT hardware compatibility list](#).

## Updating

### Built-in Update

```
sudo nupst upgrade
```

### Re-run Installer

```
curl -sSL https://code.foss.global/serve.zone/nupst/raw/branch/main/install.sh | sudo bash
```

The installer preserves your configuration and restarts the service if it was running.

## Uninstallation

```
# Interactive uninstall
sudo nupst uninstall

# Or manual removal
sudo nupst service disable
sudo rm /usr/local/bin/nupst
sudo rm -rf /opt/nupst
sudo rm -rf /etc/nupst
sudo rm /etc/systemd/system/nupst.service
sudo systemctl daemon-reload
```

## ☐ Troubleshooting

### Binary Won't Execute

```
chmod +x /opt/nupst/nupst
uname -m # x86_64 = x64, aarch64 = arm64
```

### Service Won't Start

```
sudo systemctl status nupst
sudo journalctl -u nupst -n 50
nupst config show
nupst ups test --debug
```

### Can't Connect to UPS

```
# SNMP
nupst ups test --debug
ping <ups-ip>
nc -zv <ups-ip> 161

# UPSD/NUT
nc -zv 127.0.0.1 3493
```

```
upsc ups@localhost # if NUT CLI is installed
```

# Proxmox VMs Not Shutting Down

```
# Verify API token works
curl -k -H "Authorization: PVEAPIToken=root@pam!nupst=YOUR-SECRET" \
  https://localhost:8006/api2/json/nodes/$(hostname)/qemu

# Check token permissions
pveum user token list root@pam
```

## Actions Not Triggering

```
nupst action list
nupst service logs
# Check if monitoring is paused
nupst service status
```

## 📁 System Changes

### File System

| Path                                           | Description                    |
|------------------------------------------------|--------------------------------|
| <code>/opt/nupst/nupst</code>                  | Pre-compiled binary            |
| <code>/usr/local/bin/nupst</code>              | Symlink to binary              |
| <code>/etc/nupst/config.json</code>            | Configuration file             |
| <code>/etc/nupst/pause</code>                  | Pause state file (when paused) |
| <code>/etc/systemd/system/nupst.service</code> | Systemd service unit           |

### Services

- Creates `nupst.service` systemd unit (when enabled)
- Runs with root permissions (required for system shutdown)

# Network

- Outbound SNMP to UPS devices (port 161)
- Outbound TCP to NUT servers (port 3493)
- Outbound HTTPS to Proxmox API (port 8006, if configured)
- Optional inbound HTTP server (disabled by default)
- No external internet connections

## ☐ Migration from v3.x

```
curl -sSL https://code.foss.global/serve.zone/nupst/raw/branch/main/install.sh | sudo bash
```

The installer auto-detects v3.x installations, migrates the configuration, and swaps the binary. Your settings are preserved.

| Aspect                      | v3.x                            | v4.x+                                      |
|-----------------------------|---------------------------------|--------------------------------------------|
| <b>Runtime</b>              | Node.js + npm                   | Deno (self-contained)                      |
| <b>Distribution</b>         | Git repo + npm install          | Pre-compiled binaries                      |
| <b>Runtime Dependencies</b> | node_modules                    | Zero                                       |
| <b>Size</b>                 | ~150MB                          | ~80MB                                      |
| <b>Commands</b>             | Flat ( <code>nupst add</code> ) | Subcommands ( <code>nupst ups add</code> ) |

## ☐ Development

**Requirements:** [Deno](#) v1.x or later

```
git clone https://code.foss.global/serve.zone/nupst.git
cd nupst

# Run directly
deno run --allow-all mod.ts help

# Type check
deno task check
```

```
# Lint
deno task lint

# Run tests
deno test --allow-all test/

# Compile for current platform
deno compile --allow-all --output nupst mod.ts

# Compile for all platforms
deno task compile
```

## Project Structure

```
nupst/
├─ mod.ts           # Entry point
├─ deno.json        # Deno configuration
├─ ts/
│  ├─ cli.ts        # CLI command routing
│  ├─ nupst.ts      # Main coordinator class
│  ├─ daemon.ts     # Background monitoring daemon
│  ├─ systemd.ts    # Systemd service management
│  ├─ http-server.ts # Optional HTTP JSON API
│  ├─ constants.ts  # Centralized constants
│  ├─ snmp/         # SNMP protocol implementation
│  ├─ upsd/         # UPSD/NIS protocol implementation (NUT)
│  ├─ protocol/     # Protocol abstraction layer
│  ├─ actions/      # Action system (shutdown, webhook, script, proxmox)
│  ├─ migrations/   # Config version migrations
│  ├─ helpers/      # Utility functions
│  ├─ interfaces/   # Shared TypeScript interfaces
│  └─ cli/          # CLI command handlers
├─ scripts/         # Build and install scripts
└─ test/           # Test files
```

## License and Legal Information

This repository contains open-source code licensed under the MIT License. A copy of the license can be found in the [license](#) file.

**Please note:** The MIT License does not grant permission to use the trade names, trademarks, service marks, or product names of the project, except as required for reasonable and customary use in describing the origin of the work and reproducing the content of the NOTICE file.

## Trademarks

This project is owned and maintained by Task Venture Capital GmbH. The names and logos associated with Task Venture Capital GmbH and any related products or services are trademarks of Task Venture Capital GmbH or third parties, and are not included within the scope of the MIT license granted herein.

Use of these trademarks must comply with Task Venture Capital GmbH's Trademark Guidelines or the guidelines of the respective third-party owners, and any usage must be approved in writing. Third-party trademarks used herein are the property of their respective owners and used only in a descriptive manner, e.g. for an implementation of an API or similar.

## Company Information

Task Venture Capital GmbH Registered at District Court Bremen HRB 35230 HB, Germany

For any legal inquiries or further information, please contact us via email at [hello@task.vc](mailto:hello@task.vc).

By using this repository, you acknowledge that you have read this section, agree to comply with its terms, and understand that the licensing of the code does not imply endorsement by Task Venture Capital GmbH of any derivative works.

# changelog.md for @serve.zone/nupst

## 2026-03-30 - 5.4.1 - fix(deps)

bump tsdeno and net-snmp patch dependencies

- update @git.zone/tsdeno from ^1.2.0 to ^1.3.1
- update net-snmp import from 3.26.0 to 3.26.1 in the SNMP manager

## 2026-03-30 - 5.4.0 - feat(snmp)

add configurable SNMP runtime units with v4.3 migration support

- Adds explicit `runtimeUnit` support for SNMP devices with `minutes`, `seconds`, and `ticks` options.
- Updates runtime processing to prefer configured units over UPS model heuristics.
- Introduces a v4.2 to v4.3 migration that populates `runtimeUnit` for existing SNMP device configs based on `upsModel`.
- Extends the CLI setup and device summary output to configure and display the selected runtime unit.
- Updates default config version to 4.3 and documents the new SNMP runtime unit setting in the README.

## 2026-03-18 - 5.3.3 - fix(deps)

add @git.zone/tsdeno as a development dependency

- Adds @git.zone/tsdeno@^1.2.0 to devDependencies in package.json.

## 2026-03-18 - 5.3.2 - fix(build)

replace manual release compilation workflows with tsdeno-based build configuration

- removes obsolete CI and npm publish workflows
- switches the Deno compile task to use tsdeno
- adds reusable multi-platform compile targets in npmextra.json
- updates the release workflow to install Node.js and pnpm before building binaries
- deletes the custom compile-all.sh script in favor of centralized build tooling

## 2026-03-15 - 5.3.1 - fix(cli)

rename the update command references to upgrade across the CLI and documentation

- Updates command parsing and help output to use `upgrade` instead of `update`.
- Revises user-facing upgrade prompts in daemon, systemd, and runtime status messages.
- Aligns README and command migration documentation with the renamed command.

## 2026-02-20 - 5.3.0 - feat(daemon)

Add UPSD (NUT) protocol support, Proxmox VM shutdown action, pause/resume monitoring, and network-loss/unreachable handling; bump config version to 4.2

- Add UPSD client (ts/upsd) and ProtocolResolver (ts/protocol) to support protocol-agnostic UPS queries (snmp or upsd).
- Introduce new TProtocol and IUpsdConfig types, wire up Nupst to initialize & expose UPSD client, and route status requests through ProtocolResolver.
- Add 'unreachable' TPowerStatus plus consecutiveFailures and unreachableSince tracking; mark UPS as unreachable after NETWORK.CONSECUTIVE\_FAILURE\_THRESHOLD failures and suppress shutdown actions while unreachable.
- Implement pause/resume feature: PAUSE.FILE\_PATH state file, CLI commands (pause/resume), daemon pause-state polling, auto-resume, and include pause state in HTTP API responses.
- Add ProxmoxAction (ts/actions/proxmox-action.ts) with Proxmox API interaction, configuration options (token, node, timeout, force, insecure) and CLI prompts to configure proxmox actions.
- CLI and UI updates: protocol selection when adding UPS, protocol/host shown in lists, action details column supports proxmox, and status displays include protocol and unreachable state.
- Add migration MigrationV4\_1ToV4\_2 to set protocol:'snmp' for existing devices and bump config.version to '4.2'.
- Add new constants (NETWORK, UPSD, PAUSE, PROXMOX), update package.json scripts (test/build/lint/format), and wire protocol support across daemon, systemd, http-server,

and various handlers.

## 2026-01-29 - 5.2.4 - fix()

no changes

- No files changed in the provided git diff; no commit or version bump required.

## 2026-01-29 - 5.2.3 - fix(core)

fix lint/type issues and small refactors

- Add missing node:process imports in bin and scripts to ensure process is available
- Remove unused imports and unused type imports (e.g. writeFileSync, IActionConfig) to reduce noise
- Make some methods synchronous (service update, webhook call) to match actual usage
- Tighten SNMP typings and linting: added deno-lint-ignore comments, renamed unused params with leading underscore, and use `as const` for securityLevel fallbacks
- Improve error handling variable naming in systemd (use error instead of \_error)
- Annotate ANSI regex with deno-lint-ignore no-control-regex and remove unused color/symbol imports across CLI/daemon/logger

## 2026-01-29 - 5.2.2 - fix(core)

tidy formatting and minor fixes across CLI, SNMP, HTTP server, migrations and packaging

- Normalize import ordering and improve logger/string formatting across many CLI handlers, daemon, systemd, actions and tests
- Apply formatting tidies: trailing commas, newline fixes, and more consistent multiline strings
- Allow BaseMigration methods to return either sync or async results (shouldRun/migrate signatures updated)
- Improve SNMP manager and HTTP server logging/error messages and tighten some typings (raw SNMP types, server error typing)
- Small robustness and messaging improvements in npm installer and wrapper (platform/arch mapping, error outputs)
- Update tests and documentation layout/formatting for readability

# 2026-01-29 - 5.2.1 - fix(cli(ups-handler), systemd)

add type guards and null checks for UPS configs; improve SNMP handling and prompts; guard version display

- Introduce a type guard ('id' in config && 'name' in config) to distinguish IUpsConfig from legacy INupstConfig and route fields (snmp, checkInterval, name, id) accordingly.
- displayTestConfig now handles missing SNMP by logging 'Not configured' and returning, computes checkInterval/upsName/upsId correctly, and uses groups only for true UPS configs.
- testConnection now safely derives snmpConfig for both config types, throws if SNMP is missing, and caps test timeout to 10s for probes.
- Clear auth/priv credentials by setting undefined (instead of empty strings) when disabling security levels to avoid invalid/empty string values.
- Expanded customOIDs to include OUTPUT\_LOAD, OUTPUT\_POWER, OUTPUT\_VOLTAGE, OUTPUT\_CURRENT with defaults; trim prompt input and document RFC 1628 fallbacks.
- systemd.displayVersionInfo: guard against missing nupst (silent return) and avoid errors when printing version info; use ignored catch variables for clarity.

# 2026-01-29 - 5.2.0 - feat(core)

Centralize timeouts/constants, add CLI prompt helpers, and introduce webhook/script actions with safety and SNMP refactors

- Add ts/constants.ts to centralize timing, SNMP, webhook, script, shutdown and UI constants and replace magic numbers across the codebase
- Introduce helpers/prompt.ts with createPrompt() and withPrompt() and refactor CLI handlers to use these helpers (cleaner prompt lifecycle)
- Add webhook action support: ts/actions/webhook-action.ts, IWebhookPayload type, and export from ts/actions/index.ts
- Enhance ShutdownAction safety checks (only trigger onBattery, stricter transition rules) and use constants/UI widths for displays
- Refactor SNMP manager to use logger instead of console, pull SNMP defaults from constants, improved debug output, and add INupstAccessor interface to break circular dependency (Nupst now implements the interface)
- Update many CLI and core types (stronger typing for configs/actions), expand tests and update README and npmextra.json to document new features

# 2025-11-09 - 5.1.11 - fix(readme)

Update README installation instructions to recommend automated installer script and clarify npm installation

- Replace the previous 'Via npm (NEW! - Recommended)' section with a clear 'Automated Installer Script (Recommended)' section and example curl installer.
- Move npm installation instructions into an 'Alternative: Via npm' subsection and clarify that the npm package downloads the appropriate pre-compiled binary for the platform during installation.
- Remove the 'NEW!' badge and streamline notes about binary downloads and installation methods.

# 2025-10-23 - 5.1.10 - fix(config)

Synchronize deno.json version with package.json, tidy formatting, and add local tooling settings

- Bumped deno.json version to 5.1.9 to match package.json/commitinfo
- Reformatted deno.json arrays (lint, fmt, compilerOptions) for readability
- Added .claude/settings.local.json for local development/tooling permissions (no runtime behaviour changes)

# 2025-10-23 - 5.1.9 - fix(dev)

Add local assistant permissions/settings file (.claude/settings.local.json)

- Added .claude/settings.local.json containing local assistant permission configuration used for development tasks (deno check, deno lint/format, npm/pack, running packaged binaries, etc.)
- This is a development/local configuration file and does not change runtime behavior or product code paths
- Patch version bump recommended

# 2025-10-23 - 5.1.2 - fix(scripts)

Add build script to package.json and include local dev tool settings

- Add a 'build' script to package.json (no-op placeholder) to provide an explicit build step
- Minor scripts section formatting tidy in package.json
- Add a hidden local settings file for development tooling permissions to the repository (local-only configuration)

## 2025-10-23 - 5.1.1 - fix(tooling)

Add .claude/settings.local.json with local automation permissions

- Add .claude/settings.local.json to specify allowed permissions for local automated tasks
- Grants permissions for various developer/CI actions (deno check/lint/fmt, npm/npm pack, selective Bash commands, WebFetch to docs.deno.com and code.foss.global, and file/read/replace helpers)
- This is a developer/local tooling config only and does not change runtime code or package behavior

## 2025-10-22 - 5.1.0 - feat(packaging)

Add npm packaging and installer: wrapper, postinstall downloader, publish workflow, and packaging files

- Add package.json (v5.0.5) and npm packaging metadata to publish @serve.zone/nupst
- Include a small Node.js wrapper (bin/nupst-wrapper.js) to execute platform-specific precompiled binaries
- Add postinstall script (scripts/install-binary.js) that downloads the correct binary for the current platform and sets executable permissions
- Add GitHub Actions workflow (.github/workflows/npm-publish.yml) to build binaries, pack and publish to npm, and create releases
- Add .npmignore to keep source, tests and dev files out of npm package; keep only runtime installer and wrapper
- Move example action script into docs (docs/example-action.sh) and remove the top-level example-action.sh
- Include generated npm package artifact (serve.zone-nupst-5.0.5.tgz) and npmextra.json

# 2025-10-18 - 4.0.0 - BREAKING CHANGE(core): Complete migration to Deno runtime

**MAJOR RELEASE: NUPST v4.0 is a complete rewrite powered by Deno**

This release fundamentally changes NUPST's architecture from Node.js-based to Deno-based, distributed as pre-compiled binaries. This is a **breaking change** in terms of installation and distribution, but configuration files from v3.x are **fully compatible**.

## Breaking Changes

### Installation & Distribution:

- **Removed:** Node.js runtime dependency - NUPST no longer requires Node.js
- **Removed:** npm package distribution (no longer published to npmjs.org)
- **Removed:** `bin/nupst` wrapper script
- **Removed:** `setup.sh` dependency installation
- **Removed:** All Node.js-related files (package.json, tsconfig.json, pnpm-lock.yaml, npmextra.json)
- **Changed:** Installation now downloads pre-compiled binaries instead of cloning repository
- **Changed:** Binary-based distribution (~340MB self-contained executables)

### CLI Structure (Backward Compatible):

- **Changed:** Commands now use subcommand structure (e.g., `nupst service enable` instead of `nupst enable`)
- **Maintained:** Old command format still works with deprecation warnings for smooth migration
- **Added:** Aliases for common commands (`nupst ls`, `nupst rm`)

## New Features

### Distribution & Installation:

- Pre-compiled binaries for 5 platforms:
  - Linux x86\_64
  - Linux ARM64
  - macOS x86\_64 (Intel)

- macOS ARM64 (Apple Silicon)
- Windows x86\_64
- Automated binary releases via Gitea Actions
- SHA256 checksum verification for all releases
- Installation from Gitea releases via updated `install.sh`
- Zero dependencies - completely self-contained binaries
- Cross-platform compilation from single codebase

### CI/CD Automation:

- Gitea Actions workflows for continuous integration
- Automated release workflow triggered by git tags
- Automatic binary compilation for all platforms on release
- Type checking and linting in CI pipeline
- Build verification on every push

### CLI Improvements:

- New hierarchical command structure with subcommands
- `nupst service` - Service management (enable, disable, start, stop, restart, status, logs)
- `nupst ups` - UPS device management (add, edit, remove, list, test)
- `nupst group` - Group management (add, edit, remove, list)
- `nupst config show` - Display configuration
- `nupst --version` - Show version information
- Better help messages organized by category
- Backward compatibility maintained with deprecation warnings

### Technical Improvements:

- Deno runtime for modern TypeScript/JavaScript execution
- Native TypeScript support without compilation step
- Faster startup and execution compared to Node.js
- Smaller memory footprint
- Built-in permissions system
- No build step required for development

# Migration Guide

### For Users:

1. Stop existing v3.x service: `sudo nupst disable`
2. Install v4.0 using new installer: `curl -sSL https://code.foss.global/serve.zone/nupst/raw/branch/main/install.sh | sudo bash -s -- -y`
3. Your configuration at `/etc/nupst/config.json` is preserved and fully compatible
4. Enable service with new CLI: `sudo nupst service enable && sudo nupst service start`

5. Update systemd commands to use new syntax (old syntax still works with warnings)

### Configuration Compatibility:

- All configuration files from v3.x work without modification
- No changes to `/etc/nupst/config.json` format
- All SNMP settings, thresholds, and group configurations preserved

### Command Mapping:

```
# Old (v3.x)           → New (v4.0)
nupst enable           → nupst service enable
nupst disable          → nupst service disable
nupst start            → nupst service start
nupst stop             → nupst service stop
nupst status           → nupst service status
nupst logs             → nupst service logs
nupst add              → nupst ups add
nupst edit [id]        → nupst ups edit [id]
nupst delete <id>     → nupst ups remove <id>
nupst list             → nupst ups list
nupst test             → nupst ups test
nupst group add        → nupst group add (unchanged)
nupst group edit <id> → nupst group edit <id> (unchanged)
nupst group delete <id> → nupst group remove <id>
nupst group list       → nupst group list (unchanged)
nupst config           → nupst config show
```

## Technical Details

### Commit History:

- `df6a44d`: Complete migration with Gitea Actions workflows and `install.sh` updates
- `9efcc4b`: CLI reorganization with subcommand structure
- `5903ae7`: Cross-platform compilation scripts
- `a649c59`: Deno migration with `npm:` and `node:` specifiers
- `5f4f3ec`: Initial migration to Deno

### Files Changed:

- Removed: 11 files (`package.json`, `tsconfig.json`, `pnpm-lock.yaml`, `npmextra.json`, `bin/nupst`, `setup.sh`)

- Added: 3 Gitea Actions workflows (ci.yml, release.yml, README.md)
- Modified: 14 TypeScript files for Deno compatibility
- Updated: install.sh, .gitignore, readme.md
- Net reduction: -10,242 lines (93% reduction in repository size)

### Dependencies:

- Runtime: Deno v1.x (bundled in binary, no installation required)
- SNMP: npm:net-snmp@3.20.0 (bundled in binary via npm: specifier)
- Node.js built-ins: Accessed via node: specifier (node:fs, node:child\_process, etc.)

## Benefits

### For Users:

- **Faster Installation:** Download single binary instead of cloning repo + installing Node.js + npm dependencies
- **Zero Dependencies:** No Node.js or npm required on target system
- **Smaller Footprint:** Single binary vs repo + Node.js + node\_modules
- **Easier Updates:** Download new binary instead of git pull + npm install
- **Better Security:** No npm supply chain risks, binary checksums provided
- **Platform Support:** Official binaries for all major platforms

### For Developers:

- **Modern Tooling:** Native TypeScript support without build configuration
- **Faster Development:** No compilation step during development
- **CI/CD Automation:** Automated releases and testing
- **Cleaner Codebase:** 93% reduction in configuration files
- **Cross-Platform:** Compile for all platforms from any platform

## Known Issues

- Windows ARM64 not supported (Deno limitation)
- Binary sizes are larger (~340MB) due to bundled runtime (trade-off for zero dependencies)
- First-time execution may be slower on some systems (binary extraction)

## Acknowledgments

This release represents a complete modernization of NUPST's infrastructure while maintaining full backward compatibility for user configurations. Special thanks to the Deno team for creating an excellent runtime that made this migration possible.

---

# 2025-03-28 - 3.1.2 - fix(cli/ups-handler)

Improve UPS device listing table formatting for better column alignment

- Adjusted header spacing for the Host column and overall table alignment in the UPS handler output.

# 2025-03-28 - 3.1.1 - fix(cli)

Improve table header formatting in group and UPS listings

- Adjusted column padding in group listing for proper alignment
- Fixed UPS table header spacing for consistent CLI output

# 2025-03-28 - 3.1.0 - feat(cli)

Refactor CLI commands to use dedicated handlers for UPS, group, and service management

- Extracted UPS-related CLI logic into a new UpsHandler
- Introduced GroupHandler to manage UPS groups commands
- Added ServiceHandler for systemd service operations
- Updated CLI routing in cli.ts to delegate commands to the new handlers
- Exposed getters for the new handlers in the Nupst class

# 2025-03-28 - 3.0.1 - fix(cli)

Simplify UPS ID generation by removing the redundant promptForUniqueUpsId function in the CLI module and replacing it with the shortId helper.

- Deleted the unused promptForUniqueUpsId method from ts/cli.ts.
- Updated UPS configuration to generate a unique ID directly using helpers.shortId().
- Improved code clarity by removing unnecessary interactive prompts for UPS IDs.

# 2025-03-28 - 3.0.0 - BREAKING CHANGE(core)

Add multi-UPS support and group management; update CLI, configuration and documentation to support multiple UPS devices with group modes

- Implemented multi-UPS configuration with an array of UPS devices and groups in the configuration file
- Added group management commands (group add, edit, delete, list) with redundant and non-redundant modes
- Revamped CLI command parsing for UPS management (add, edit, delete, list, setup) and group subcommands
- Updated readme and documentation to reflect new configuration structure and features
- Enhanced logging and status display for multiple UPS devices

# 2025-03-26 - 2.6.17 - fix(logger)

Preserve logbox width after logBoxEnd so that subsequent logBoxLine calls continue using the set width.

- Removed the reset of currentBoxWidth in logBoxEnd to allow persistent width across logbox calls.
- Ensures that logBoxLine uses the previously set width when no new width is provided.

# 2025-03-26 - 2.6.16 - fix(cli)

Improve CLI logging consistency by replacing direct console output with unified logger calls.

- Replaced console.log and console.error with logger.log and logger.error in CLI commands
- Standardized debug, error, and status messages using logger's logbox utilities
- Enhanced consistency of log output throughout the ts/cli.ts file

# 2025-03-26 - 2.6.15 - fix(logger)

Replace direct console logging with unified logger interface for consistent formatting

- Substitute console.log, console.error, and related calls with logger methods in cli, daemon, systemd, nupst, and index modules
- Integrate logBox formatting for structured output and consistent log presentation
- Update test expectations in test.logger.ts to check for standardized error messages
- Refactor logging calls throughout the codebase for improved clarity and maintainability

## 2025-03-26 - 2.6.14 - fix(systemd)

Shorten closing log divider in systemd service installation output for consistent formatting.

- Replaced the overly long footer with a shorter one in ts/systemd.ts.
- This change improves log readability without affecting functionality.

## 2025-03-26 - 2.6.13 - fix(cli)

Fix CLI update output box formatting

- Adjusted the closing box line in the update process log messages for consistent visual formatting

## 2025-03-26 - 2.6.12 - fix(systemd)

Adjust logging border in systemd service installation output

- Updated the closing border line for consistent output formatting in ts/systemd.ts

## 2025-03-26 - 2.6.11 - fix(cli, systemd)

Adjust log formatting for consistent output in CLI and systemd commands

- Fixed spacing issues in service installation and status log messages in the systemd module.
- Revised output formatting in the CLI to improve message clarity.

# 2025-03-26 - 2.6.10 - fix(daemon)

Adjust console log box formatting for consistent output in daemon status messages

- Updated closing box borders to align properly in configuration error, periodic updates, and UPS status logs
- Improved visual consistency in log messages

# 2025-03-26 - 2.6.9 - fix(cli)

Improve console output formatting for status banners and logging messages

- Standardize banner messages in daemon status updates
- Refine version information banner in nupst logging
- Update UPS connection and status banners in systemd

# 2025-03-26 - 2.6.8 - fix(cli)

Improve CLI formatting, refine debug option filtering, and remove unused dgram import in SNMP manager

- Standardize whitespace and formatting in ts/cli.ts for consistency
- Refine argument filtering for debug mode and prompt messages
- Remove unused 'dgram' import from ts/snmp/manager.ts

# 2025-03-26 - 2.6.7 - fix(setup.sh)

Clarify net-snmp dependency installation message in setup.sh

- Updated echo statement to indicate installation of net-snmp along with 2 subdependencies
- Improves clarity on dependency installation during setup

# 2025-03-26 - 2.6.6 - fix(setup.sh)

Improve setup script to detect and execute npm-cli.js directly using the Node.js binary

- Replace use of the npm binary with direct execution of npm-cli.js
- Add fallback logic to locate npm-cli.js when not found at the expected path
- Simplify cleanup by removing unnecessary PATH modifications

## 2025-03-26 - 2.6.5 - fix(daemon, setup)

Improve shutdown command detection and fallback logic; update setup script to use absolute Node/npm paths

- Use execFileAsync to execute shutdown commands reliably
- Add multiple fallback alternatives for shutdown and emergency shutdown handling
- Update setup.sh to log the Node and NPM versions using absolute paths without modifying PATH

## 2025-03-26 - 2.6.4 - fix(setup)

Improve installation process in setup script by cleaning up package files and ensuring a minimal net-snmp dependency installation.

- Remove existing package-lock.json along with node\_modules to prevent stale artifacts.
- Back up the original package.json before modifying it.
- Create a minimal package.json with only the net-snmp dependency based on the backed-up version.
- Use a clean install to guarantee that only net-snmp is installed.
- Restore the original package.json if the installation fails.

## 2025-03-26 - 2.6.3 - fix(setup)

Update setup script to install only net-snmp dependency and create a minimal package-lock.json for better dependency control.

- Removed full production dependency install in favor of installing only net-snmp@3.20.0
- Added verification step to confirm net-snmp installation
- Generate a minimal package-lock.json if one does not exist

# 2025-03-26 - 2.6.2 - fix(setup/readme)

Improve force update instructions and dependency installation process in setup.sh and readme.md

- Clarify force update commands with explicit paths in readme.md
- Remove existing node\_modules before installing dependencies in setup.sh
- Switch from 'npm ci --only=production' to 'npm install --omit=dev' with updated error instructions

# 2025-03-26 - 2.6.1 - fix(setup)

Update setup.sh to temporarily add vendor Node.js binary to PATH for dependency installation, log Node and npm versions, and restore the original PATH afterwards.

- Temporarily prepend vendor Node.js binary directory to PATH to ensure proper npm execution.
- Log Node.js and npm versions for debugging purposes.
- Restore the original PATH after installing dependencies.

# 2025-03-26 - 2.6.0 - feat(setup)

Add --force update flag to setup script and update installation instructions

- Implemented --force option in setup.sh to force-update Node.js binary and dependencies
- Updated readme.md to document the --force flag and revised update steps
- Modified ts/cli.ts update command to pass the --force flag to setup.sh

# 2025-03-26 - 2.5.2 - fix(installer)

Improve Node.js binary detection, dependency management, and SNMPv3 fallback logic

- Enhanced bin/nupst to detect OS and architecture (Linux and Darwin) and fall back to system Node.js for unsupported platforms
- Moved net-snmp from devDependencies to dependencies in package.json

- Updated setup.sh to install production dependencies and handle installation errors gracefully
- Refined SNMPv3 user configuration and fallback mechanism in ts/snmp/manager.ts
- Revised README to clarify minimal runtime dependencies and secure SNMP features

## 2025-03-25 - 2.5.1 - fix(snmp)

Fix Eaton UPS support by updating power status OID and adjusting battery runtime conversion.

- Updated Eaton UPS power status OID to '1.3.6.1.4.1.534.1.4.4.0' to correctly detect online/battery status.
- Added conversion for Eaton UPS battery runtime from seconds to minutes in SNMP manager.

## 2025-03-25 - 2.5.0 - feat(cli)

Automatically restart running NUPST service after configuration changes in interactive setup

- Added restartServiceIfRunning() to check and restart the service if it's active.
- Invoked the restart function post-setup to apply configuration changes immediately.

## 2025-03-25 - 2.4.8 - fix(installer)

Improve Git dependency handling and repository cloning in install.sh

- Add explicit check for git installation and prompt the user interactively if git is missing.
- Auto-install git when '-y' flag is provided in non-interactive mode.
- Ensure proper cloning of the repository when running the installer outside the repo.

## 2025-03-25 - 2.4.7 - fix(readme)

Update installation instructions to combine download and execution into a single command for clarity

- Method 1 now uses a unified one-line command to download and run the install script

# 2025-03-25 - 2.4.6 - fix(installer)

Improve installation instructions for interactive and non-interactive setups

- Changed install.sh to require explicit download of the install script and updated error messages for non-interactive modes
- Updated readme.md to include three distinct installation methods with clear command examples

# 2025-03-25 - 2.4.5 - fix(install)

Improve interactive terminal detection and update installation instructions

- Enhanced install.sh to better detect non-interactive environments and provide clearer guidance for both interactive and non-interactive installations
- Updated README.md quick install instructions to recommend process substitution and clarify auto-yes usage

# 2025-03-25 - 2.4.4 - fix(install)

Improve interactive mode detection and non-interactive installation handling in install.sh

- Detect and warn when running without a controlling terminal
- Attempt to use /dev/tty for user input when possible
- Update prompts and error messages for auto-installation of dependencies
- Clarify installation instructions in readme for interactive and non-interactive modes

# 2025-03-25 - 2.4.3 - fix(readme)

Update Quick Install command syntax in readme for auto-yes installation

- Changed installation command to use: `curl -sSL https://code.foss.global/serve.zone/nupst/raw/branch/main/install.sh | sudo bash -c "bash -s -- -y"`

# 2025-03-25 - 2.4.2 - fix(daemon)

Refactor shutdown initiation logic in daemon by moving the initiateShutdown and monitorDuringShutdown methods from the SNMP manager to the daemon, and update calls accordingly

- Moved initiateShutdown and monitorDuringShutdown to the daemon class for improved cohesion
- Updated references in the daemon to call its own shutdown method instead of the SNMP manager
- Removed redundant initiateShutdown method from the SNMP manager

## 2025-03-25 - 2.4.1 - fix(docs)

Update readme with detailed legal and trademark guidance

- Clarified legal section by adding trademark and company information
- Ensured users understand that licensing terms do not imply endorsement by the company

## 2025-03-25 - 2.4.0 - feat(installer)

Add auto-yes flag to installer and update installation documentation

- Enhance install.sh to parse -y/--yes and -h/--help options, automating git installation when auto-yes is provided
- Improve user prompts for dependency installation and provide clearer instructions
- Update readme.md to document new installer options and enhanced file system and service changes details

## 2025-03-25 - 2.3.0 - feat(installer/cli)

Add OS detection and git auto-installation support to install.sh and improve service setup prompt in CLI

- Implemented helper functions in install.sh to detect OS type and automatically install git if missing
- Prompt user for git installation if not present before cloning the repository

- Enhanced CLI service setup flow to offer starting the NUPST service immediately after installation

## 2025-03-25 - 2.2.0 - feat(cli)

Add 'config' command to display current configuration and update CLI help

- Introduce new 'config' command to show SNMP settings, thresholds, and configuration file location
- Update help text to include details for 'nupst config' command

## 2025-03-25 - 2.1.0 - feat(cli)

Add uninstall command to CLI and update shutdown delay for graceful VM shutdown

- Implement uninstall command in ts/cli.ts that locates and executes uninstall.sh with user prompts
- Update uninstall.sh to support environment variables for configuration and repository removal
- Increase shutdown delay in ts/snmp/manager.ts from 1 minute to 5 minutes to allow VMs more time to shut down

## 2025-03-25 - 2.0.1 - fix(cli/systemd)

Fix status command to pass debug flag and improve systemd status logging output

- ts/cli.ts: Now extracts debug options from process arguments and passes debug mode to getStatus.
- ts/systemd.ts: Updated getStatus to accept a debugMode parameter, enabling detailed SNMP debug logging, explicitly reloading configuration, and printing connection details.

# 2025-03-25 - 2.0.0 - BREAKING CHANGE(snmp)

refactor: update SNMP type definitions and interface names for consistency

- Renamed SnmpConfig to ISnmpConfig, OIDSet to IOidSet, UpsStatus to IUpsStatus, and UpsModel to TUpsModel in ts/snmp/types.ts.
- Updated internal references in ts/daemon.ts, ts/snmp/index.ts, ts/snmp/manager.ts, ts/snmp/oid-sets.ts, ts/snmp/packet-creator.ts, and ts/snmp/packet-parser.ts to use the new interface names.

# 2025-03-25 - 1.10.1 - fix(systemd/readme)

Improve README documentation and fix UPS status retrieval in systemd service

- Updated README features and installation instructions to clarify SNMP version support, UPS models, and configuration
- Modified default SNMP host to '192.168.1.100' and added 'upsModel' property in configuration examples
- Enhanced instructions for real-time log viewing and update process in README
- Fixed systemd.ts to use a test configuration with an appropriate timeout when fetching UPS status

# 2025-03-25 - 1.10.0 - feat(core)

Add update checking and version logging across startup components

- In daemon.ts, log version info on startup and check for updates in the background using npm registry response
- In nupst.ts, implement getVersion, checkForUpdates, getUpdateStatus, and compareVersions functions with update notifications
- Establish bidirectional reference between Nupst and NupstSnmp to support version logging
- Update systemd service status output to include version information

# 2025-03-25 - 1.9.0 - feat(cli)

Add update command to CLI to update NUPST from repository and refresh the systemd service

- Integrate 'update' subcommand in CLI command parser
- Update documentation and help output to include new command
- Implement update process that fetches changes from git, runs install.sh/setup.sh, and refreshes systemd service if installed

# 2025-03-25 - 1.8.2 - fix(cli)

Refactor logs command to use child\_process spawn for real-time log tailing

- Replaced execSync call with spawn to properly follow logs
- Forward SIGINT to the spawned process for graceful termination
- Await the child process exit to ensure clean shutdown of the CLI log command

# 2025-03-25 - 1.8.1 - fix(systemd)

Update ExecStart in systemd service template to use /opt/nupst/bin/nupst for daemon startup

- Changed ExecStart from '/usr/bin/nupst daemon-start' to '/opt/nupst/bin/nupst daemon-start' in the systemd service file
- Ensures the service uses the correct binary installation path

# 2025-03-25 - 1.8.0 - feat(core)

Enhance SNMP module and interactive CLI setup for UPS shutdown

- Refactored SNMP packet parsing and encoding utilities for clearer error handling and debugging
- Improved systemd service management with detailed logging during installation and control
- Enhanced interactive CLI setup process for configuring SNMP settings and UPS models
- Expanded test coverage with simulated SNMP responses for various response types

# 2025-03-25 - 1.7.6 - fix(core)

Refactor SNMP, systemd, and CLI modules to improve error handling, logging, and code clarity

- Removed unused dependency 'net-snmp' from package.json
- Extracted helper functions for SNMP packet creation and parsing (using SnmpEncoder, SnmpPacketCreator and SnmpPacketParser)
- Improved debug logging and added detailed documentation comments across SNMP, systemd, CLI and daemon modules
- Refactored systemd service management to extract status display and service disabling logic
- Updated test suite to use proper modular methods from the new SNMP utilities

# 2025-03-25 - 1.7.5 - fix(cli)

Enable SNMP debug mode in CLI commands and update debug flag handling in daemon-start and test; bump version to 1.7.4

- Call enableDebug() on SNMP client earlier in command parsing
- Pass debug flag to 'daemon-start' and 'test' commands for consistent debug output
- Update package version from 1.7.3 to 1.7.4

# 2025-03-25 - 1.7.3 - fix(SNMP)

Refine SNMP packet creation and response parsing for more reliable UPS status monitoring

- Improve error handling and fallback logic when parsing SNMP responses
- Optimize TimeTicks conversion for CyberPower UPS devices
- Enhance test coverage for various UPS scenarios

# 2025-03-25 - 1.7.2 - fix(core)

Refactor internal SNMP response parsing and enhance daemon logging for improved error reporting and clarity.

- Improved fallback and error handling in SNMP response parsing
- Enhanced logging messages in daemon and systemd service management

- Minor refactoring for better maintainability without functional changes

## 2025-03-25 - 1.7.1 - fix(snmp-cli)

Improve SNMP response parsing and CLI UPS connection timeout handling

- Expand parsing loop in SNMP responses to capture Gauge32 and Timeticks values
- Add detailed debug logging for both SNMP v1/v2 and v3 responses
- Configure CLI test commands to use a shortened timeout for UPS connection tests

## 2025-03-25 - 1.7.0 - feat(SNMP/UPS)

Add UPS model selection and custom OIDs support to handle different UPS brands

- Introduce distinct OID sets for CyberPower, APC, Eaton, TrippLite, Liebert, and a custom option
- Update interactive setup to prompt for UPS model selection and custom OID entry when needed
- Refactor SNMP status retrieval to dynamically select the appropriate OIDs based on the configured UPS model
- Extend default configuration with an upsModel property for consistent behavior

## 2025-03-25 - 1.6.0 - feat(cli,snmp)

Enhance debug logging and add debug mode support in CLI and SNMP modules

- Enable debug flags (--debug, -d) in CLI to trigger detailed SNMP logging
- Pass debug mode options to daemonStart and test commands for improved diagnostics
- Add enableDebug method to the SNMP module for on-demand debug logging
- Improve timeout and discovery logging details for streamlined troubleshooting

## 2025-03-25 - 1.5.0 - feat(cli)

Enhance CLI output: display SNMPv3 auth/priv details and support timeout customization during setup

- Display authentication and privacy protocol details when SNMP version is 3
- Show timeout value in the configuration test output
- Clear out unused authentication or privacy settings based on selected security level
- Allow users to customize SNMP timeout during interactive setup

## 2025-03-25 - 1.4.1 - fix(version)

Bump patch version for consistency with commit info

## 2025-03-25 - 1.4.0 - feat(snmp)

Implement native SNMPv3 support with simulated encryption and enhanced authentication handling.

- Add fully native SNMPv3 GET request implementation replacing the snmpwalk fallback
- Simulate encryption for authPriv using AES and DES protocols
- Enhance SNMP response parser to properly handle SNMPv3 responses
- Introduce detailed security parameter management for SNMPv3

## 2025-03-25 - 1.3.1 - fix(cli)

Remove redundant SNMP tools checks in CLI and Systemd modules

- Eliminate unnecessary snmpwalk dependency checks in the test command and interactive setup flow.
- Adjust systemd configuration file check to avoid external dependency verification.

## 2025-03-25 - 1.3.0 - feat(cli)

add test command to verify UPS SNMP configuration and connectivity

- Introduce a new 'test' command in the CLI to check the SNMP configuration and UPS connection.
- Validate installation of SNMP tools and configuration file before testing.

- Output UPS status details and compare against defined shutdown thresholds.

## 2025-03-25 - 1.2.6 - fix(cli)

Refactor interactive setup to use dynamic import for readline and ensure proper cleanup

- Replaced synchronous require() with async import for ESM compatibility
- Wrapped interactive setup in try/finally to guarantee readline interface closure
- Enhanced error logging by outputting error.message

## 2025-03-25 - 1.2.5 - fix(error-handling)

Improve error handling in CLI, daemon, and systemd lifecycle management with enhanced logging for configuration issues

- Wrap daemon and service start commands in try-catch blocks to properly handle and log errors
- Throw explicit errors when configuration file is missing instead of silently defaulting
- Enhance log messages for service installation, startup, and status retrieval for clearer debugging

## 2025-03-25 - 1.2.4 - fix(cli/daemon)

Improve logging and user feedback in interactive setup and UPS monitoring

- Refactor configuration summary output in the interactive setup for clearer display
- Enhance logging in the daemon to show detailed SNMP settings and UPS status changes
- Improve error messages and user guidance during configuration and monitoring

## 2025-03-24 - 1.2.3 - fix(nupst)

No changes

## 2025-03-24 - 1.2.2 - fix(bin/nupst)

Improve symlink resolution in launcher script to correctly determine project root based on execution path.

- Replace directory determination with readlink for accurate symlink resolution
- Set project root to '/opt/nupst' when script is run via symlink from /usr/local/bin
- Add debugging comments to assist with path resolution

## 2025-03-24 - 1.2.1 - fix(bin)

Simplify Node.js binary detection in installation script

- Directly set Node binary path to vendor/node-linux-x64/bin/node
- Remove redundant architecture-specific detection logic
- Fallback to system Node if vendor binary is not found

## 2025-03-24 - 1.2.0 - feat(installer)

Improve Node.js binary detection and dynamic LTS version retrieval in setup scripts

- Enhanced bin/nupst to search multiple possible locations for the Node.js binary and fallback to system node if necessary
- Updated setup.sh to fetch the latest LTS Node.js version from nodejs.org and use a fallback version when the request fails

## 2025-03-24 - 1.1.2 - fix(setup.sh)

Improve error handling in setup.sh: exit immediately when the downloaded npm package lacks the dist\_ts directory, removing the fallback build-from-source mechanism.

- Removed BUILD\_FROM\_SOURCE logic that attempted to build from source on missing dist\_ts directory
- Updated error messages to clearly indicate failure in downloading a valid package
- Ensured installation halts if essential files are missing

# 2025-03-24 - 1.1.1 - fix(package.json)

Remove unused prepublishOnly script and update files field in package.json

- Removed prepublishOnly build trigger
- Updated files list to accurately include intended directories and files

# 2025-03-24 - 1.1.0 - feat(installer- setup)

Enhance installer and setup scripts for improved global installation and artifact management

- Detect piped installation in install.sh, clone repository automatically, and clean up previous installations
- Update readme.md with correct repository URL and clearer installation instructions
- Improve setup.sh to remove existing dist\_ts, download build artifacts from the npm registry, and simplify dependency installation

# 2025-03-24 - 1.0.1 - fix(version)

Bump version to 1.0.1

- Updated commitinfo data to reflect the new patch version.
- Synchronized version information between commitinfo file and package metadata.

# 2025-03-24 - 1.0.1 - fix(build)

Update build script to use 'tsbuild tsfolders --allowimplicitany' and adjust distribution paths in .gitignore

- Replaced 'tsc' with 'tsbuild tsfolders --allowimplicitany' in package.json
- Updated .gitignore to reflect new compiled distribution folder pattern
- Updated changelog to document build improvements and regenerated type definitions

# 2025-03-24 - 1.0.1 - fix(build)

Update build script to use 'tsbuild tsfolders --allowimplicitany' and regenerate distribution type definitions for CLI, daemon, index, nupst, snmp, and systemd modules

- Replaced 'tsc' command with tsbuild in package.json
- Updated .gitignore to reflect new compiled distribution folder pattern
- Added new dist\_ts files including .d.ts type definitions and compiled JavaScript for multiple modules

# 2025-03-24 - 1.0.1 - fix(build)

Update build script to use 'tsbuild tsfolders --allowimplicitany' and regenerate distribution type definitions for CLI, daemon, nupst, snmp, and systemd modules.

- Replaced the 'tsc' command with 'tsbuild tsfolders --allowimplicitany' in package.json.
- Added new dist\_ts files including type definitions (d.ts) and compiled JavaScript for CLI, daemon, index, nupst, snmp, and systemd.
- Improved the generated CLI declarations and overall distribution build.

# 2025-03-23 - 1.0.0 - initial setup

This range covers the early commits that mainly established the repository structure.

- Initial repository commit with basic project initialization.