

# Meshtastic CLI Reference

- [Meshtastic Python CLI Guide](#)
- [Complete Meshtastic CLI Command Reference](#)

# Meshtastic Python CLI Guide

The Meshtastic Python package provides both a command-line interface and a Python library for scripting. It is the primary tool for configuring nodes without the mobile app, backing up configurations, and automating tasks.

## Installation

```
pip install meshtastic
```

Requires Python 3.7+. On Linux you may need to add your user to the `dialout` group to access serial ports without sudo:

```
sudo usermod -aG dialout $USER
```

Serial access: connect via USB at **115200 baud**. The CLI auto-detects the port in most cases.

## Getting Device Information

```
# Full device info: firmware, hardware, channel config, user info
```

```
meshtastic --info
```

```
# List all nodes heard by this device
```

```
meshtastic --nodes
```

## Configuration Backup and Restore

```
# Export full configuration to YAML
```

```
meshtastic --export-config > config.yaml
```

```
# Restore configuration from YAML
```

```
meshtastic --import-config config.yaml
```

Back up your configuration before any firmware update. The export includes all channel PSKs, radio settings, and module configuration.

# Setting Node Identity

```
# Set the node's display name
meshtastic --set-owner "MyName"

# Set a short name (4 chars, shown on small displays)
meshtastic --set-owner-short "MN01"
```

# Radio Configuration

```
# Set LoRa region (required before use)
meshtastic --set lora.region US

# Set modem preset
meshtastic --set lora.modem_preset LONG_FAST

# Set TX power (dBm; check regional legal limits)
meshtastic --set lora.tx_power 30

# Set hop limit (default 3; max 7)
meshtastic --set lora.hop_limit 3
```

# Device Role

```
# Set device role
meshtastic --set device.role CLIENT_ROLE
meshtastic --set device.role CLIENT_MUTE_ROLE
meshtastic --set device.role CLIENT_HIDDEN_ROLE
meshtastic --set device.role CLIENT_BASE_ROLE
meshtastic --set device.role TRACKER_ROLE
meshtastic --set device.role LOST_AND_FOUND_ROLE
meshtastic --set device.role SENSOR_ROLE
meshtastic --set device.role TAK_ROLE
meshtastic --set device.role TAK_TRACKER_ROLE
meshtastic --set device.role ROUTER_ROLE
```

```
meshtastic --set device.role ROUTER_LATE_ROLE
meshtastic --set device.role REPEATER_ROLE
```

# Channel Management

```
# Add a new channel (assigns next available index)
meshtastic --ch-add MyPrivateChannel

# Enable MQTT uplink on channel 0
meshtastic --ch-index 0 --ch-set uplink_enabled true

# Enable MQTT downlink on channel 0
meshtastic --ch-index 0 --ch-set downlink_enabled true

# Set a custom PSK on channel 1
meshtastic --ch-index 1 --ch-set psk base64:YOUR_KEY_HERE
```

# Telemetry

```
# Set device metrics broadcast interval (seconds)
meshtastic --set telemetry.device_update_interval 3600

# Set environment sensor interval (seconds)
meshtastic --set telemetry.environment_update_interval 1800

# Enable environment telemetry
meshtastic --set telemetry.environment_measurement_enabled true
```

# Position

```
# Set fixed position (for nodes without GPS)
meshtastic --setlat 39.7392 --setlon -104.9903 --setalt 1609

# Set position broadcast interval (seconds; 0 = disable)
meshtastic --set position.position_broadcast_secs 3600
```

# Sending Messages

```
# Send a message to all nodes on the primary channel
meshtastic --sendtext "Hello mesh"

# Send on a specific channel (index 1)
meshtastic --ch-index 1 --sendtext "Private message"
```

# Serial Console (Advanced)

The Meshtastic serial interface also accepts raw JSON commands at 115200 baud. For most users the Python CLI is the right tool, but scripting directly over serial is possible for embedded integrations:

```
import serial, json
ser = serial.Serial('/dev/ttyUSB0', 115200, timeout=1)
# Send admin packets as protobuf; see Meshtastic Python library source for details
```

# Common CLI Patterns

```
# Full setup sequence for a new infrastructure repeater
meshtastic --set-owner "Tower-Node-01"
meshtastic --set lora.region US
meshtastic --set lora.modem_preset LONG_FAST
meshtastic --set device.role REPEATER_ROLE
meshtastic --set lora.hop_limit 3
meshtastic --export-config > tower-node-01-config.yaml
```

# Complete Meshtastic CLI Command Reference

The Meshtastic Python CLI provides the most comprehensive access to node configuration and data. This reference covers all major command categories.

## Installation

```
pip install meshtastic
# or for latest development version:
pip install meshtastic --pre
```

## Connection Options

```
# Auto-detect serial port (most common):
meshtastic --info

# Specify serial port (Windows):
meshtastic --port COM3 --info

# Specify serial port (Linux/Mac):
meshtastic --port /dev/ttyUSB0 --info

# Connect via TCP (WiFi-enabled nodes):
meshtastic --host 192.168.1.100 --info

# Connect via BLE:
meshtastic --ble "Meshtastic_ABCD" --info
```

## Information and Status

```
# Show device info, config, and channels:
```

```
meshtastic --info
```

```
# Show node database (all known nodes):
```

```
meshtastic --nodes
```

```
# Show channel list:
```

```
meshtastic --export-config
```

## Configuration Get/Set

```
# Get a specific config value:
```

```
meshtastic --get device.role
```

```
meshtastic --get lora.hop_limit
```

```
meshtastic --get power.is_power_saving
```

```
# Set a config value:
```

```
meshtastic --set device.role ROUTER
```

```
meshtastic --set lora.hop_limit 5
```

```
meshtastic --set lora.region US
```

```
# Set multiple values at once:
```

```
meshtastic --set device.role ROUTER --set lora.hop_limit 5 --set power.is_power_saving false
```

## Channel Management

```
# Show current channels:
```

```
meshtastic --info # includes channel list
```

```
# Set channel 0 (primary) name and key:
```

```
meshtastic --ch-index 0 --ch-set name "CommunityMesh"
```

```
meshtastic --ch-index 0 --ch-set psk "base64key=="
```

```
# Add a new channel:
```

```
meshtastic --ch-add
```

```
# Delete channel (specify index):
meshtastic --ch-del --ch-index 2

# Enable/disable MQTT uplink per channel:
meshtastic --ch-index 0 --ch-set uplink_enabled true
meshtastic --ch-index 0 --ch-set downlink_enabled true
```

## Identity

```
# Set long name:
meshtastic --set-owner "Your Name"

# Set short name (4 chars max):
meshtastic --set-owner-short "AB01"
```

## Position

```
# Set fixed GPS position (lat, lon, alt in meters):
meshtastic --setlat 45.5051 --setlon -122.6750 --setalt 50

# Remove fixed position (use live GPS):
meshtastic --remove-position
```

## Messaging

```
# Send a text message to primary channel:
meshtastic --sendtext "Hello, mesh!"

# Send to a specific channel index:
meshtastic --ch-index 1 --sendtext "Admin message"

# Listen for incoming messages and packets (Ctrl+C to stop):
meshtastic --listen

# Send GPS waypoint:
```

```
meshtastic --sendwaypoint --waypoint-name "Meeting Point" --waypoint-lat 45.5051 --waypoint-  
lon -122.6750
```

# Maintenance

```
# Export full config to file:
```

```
meshtastic --export-config > config.json
```

```
# Import config from file:
```

```
meshtastic --import-config config.json
```

```
# Factory reset (clears all config and node DB):
```

```
meshtastic --factory-reset
```

```
# Reboot device:
```

```
meshtastic --reboot
```