

nmcli - NetworkManager

nmcli manages NetworkManager in Linux. Work with two core concepts: **Device** (physical hardware like eth0 or wlan0) and **Connection** (a saved configuration profile).

Alternatively you could use nmtui application with simple console UI (NetworkManager Textual User Interface).

Overview & Diagnostics

* View overall network status:

```
$ nmcli general status
```

* List physical network interfaces and their status:

```
$ nmcli device status
-----
DEVICE   TYPE        STATE                CONNECTION
enp2s0   ethernet    connected            Wired connection 1
lo       loopback    connected (externally) lo
virbr0   bridge      connected (externally) virbr0
```

* Show detailed interface info (IP, MAC, DNS, Gateway):

```
$ nmcli device show enp2s0
-----
GENERAL.DEVICE:                enp2s0
GENERAL.TYPE:                  ethernet
GENERAL.HWADDR:                50:E5:49:ED:A9:BF
GENERAL.MTU:                    1500
GENERAL.STATE:                  100 (connected)
GENERAL.CONNECTION:            Wired connection 1
GENERAL.CON-PATH:              /org/freedesktop/NetworkManager/ActiveConnection/2
WIRED-PROPERTIES.CARRIER:      on
IP4.ADDRESS[1]:                192.168.178.18/24
IP4.GATEWAY:                    192.168.178.1
IP4.ROUTE[1]:                  dst = 192.168.178.0/24, nh =
0.0.0.0, mt = 100
IP4.ROUTE[2]:                  dst = 0.0.0.0/0, nh = 192.168.178.1,
mt = 100
IP4.DNS[1]:                    192.168.178.1
IP4.DOMAIN[1]:                  lan
IP6.ADDRESS[1]:                2001:1c00:2e07:fa0a:c363:14e8:16bd:6436/64
```

```

IP6.ADDRESS[2]:          fd00::66:67:a:4b23:15d3:3424:d1ab/64
IP6.ADDRESS[3]:          fd00::66:67:a::e78/128
IP6.ADDRESS[4]:          2001:1c00:2e07:fa0a::e78/128
IP6.ADDRESS[5]:          2001:1c00:2e07:fa0a:52e5:49ff:feed:a9bf/64
IP6.ADDRESS[6]:          fd00::66:67:a:52e5:49ff:feed:a9bf/64
IP6.ADDRESS[7]:          fe80::52e5:49ff:feed:a9bf/64
IP6.GATEWAY:             fe80::ee08:6bff:fe84:2043
IP6.ROUTE[1]:            dst = fe80::/64, nh = ::, mt = 1024
IP6.ROUTE[2]:            dst = fd00::66:67:a::/64, nh = ::, mt
= 100
IP6.ROUTE[3]:            dst = 2001:1c00:2e07:fa0a::/64, nh =
::, mt = 100
IP6.ROUTE[4]:            dst = ::/0, nh =
fe80::ee08:6bff:fe84:2043, mt = 100
IP6.ROUTE[5]:            dst = fd00::66:67:a::e78/128, nh =
::, mt = 100
IP6.ROUTE[6]:            dst = 2001:1c00:2e07:fa0a::e78/128,
nh = ::, mt = 100
IP6.DNS[1]:              fd00::66:67:a::1

```

*** List all saved connection profiles:**

```
$ nmcli connection show
```

*** Show only active connections:**

```

$ nmcli connection show --active
-----
NAME                UUID                                TYPE      DEVICE
Wired connection 1  483fb914-06e5-4b66-b00f-e2b31025eaff  ethernet  enp2s0
lo                  ce29234b-7809-4b88-b743-79bf79ac55bf  loopback  lo
virbr0             4778a23b-a938-412c-8c64-e78ecb98571c  bridge    virbr0

```

Wi-Fi Management

*** Scan for nearby Wi-Fi networks:**

```
$ nmcli device wifi list
```

*** Connect to a Wi-Fi network:**

```
# nmcli device wifi connect "SSID_NAME" password "WIFI_PASSWORD"
```

*** Prompt securely for Wi-Fi password (prevents saving password in shell history):**

```
# nmcli --ask device wifi connect "SSID_NAME"
```

*** Toggle Wi-Fi on or off:**

```
$ nmcli radio wifi off  
$ nmcli radio wifi on
```

Connection Controls (Up / Down / Delete)

*** Activate a connection profile:**

```
# nmcli connection up "MyConnection"
```

*** Deactivate a connection profile:**

```
# nmcli connection down "MyConnection"
```

*** Disconnect a hardware device directly:**

```
# nmcli device disconnect eth0
```

*** Delete a saved connection profile:**

```
sudo nmcli connection delete "MyConnection"
```

Network Configuration (IP, Gateway, DNS)

*** Set a static IPv4 address and gateway:**

```
# nmcli connection modify "Wired connection 1" ipv4.addresses  
192.168.1.50/24 ipv4.gateway 192.168.1.1 ipv4.method manual  
# nmcli connection up "Wired connection 1"
```

*** Switch back to dynamic IP (DHCP):**

```
# nmcli connection modify "Wired connection 1" ipv4.method auto  
# nmcli connection up "Wired connection 1"
```

*** Set custom DNS servers:**

```
# nmcli connection modify "Wired connection 1" ipv4.dns "1.1.1.1 8.8.8.8"  
# nmcli connection up "Wired connection 1"
```

Adding New Profiles

* **Add a standard Ethernet connection profile (DHCP):**

```
# nmcli connection add type ethernet con-name "Office-Eth" ifname eth0
```

* **Add a static Ethernet connection profile directly:**

```
# nmcli connection add type ethernet con-name "Static-Eth" ifname eth0 ip4 10.0.0.10/24 gw4 10.0.0.1
```

NetworkManager Configuration

Locations

NetworkManager configuration files and connection profiles are stored across a few specific directories on your system.

Location	Description
/etc/NetworkManager/system-connections/	Saved Connection Profile. Contains all individual network profiles (Wi-Fi access points, static IP settings, VPNs, Ethernet configurations). Each file is stored in Keyfile format and contains sensitive credentials like Wi-Fi passwords.
/etc/NetworkManager/NetworkManager.conf /etc/NetworkManager/conf.d/	Global Configuration & Custom Drops. NetworkManager.conf is the primary system configuration file. Any custom overrides or plugin configurations belong in /etc/NetworkManager/conf.d/.
/run/NetworkManager/ /var/lib/NetworkManager/	DNS& Runtime Network State (Auto-generated). Contains internal DHCP leases, runtime connection states, and auto-generated DNS configurations. These do not need to be backed up.

Backup/Restore Configuration

When testing network changes such as bridges, VLANs, or bonds with `nmcli`, back up the NetworkManager connection profiles first. Also capture the current runtime state as a readable reference.

Back up NetworkManager

Because `/etc/NetworkManager/system-connections/` stores Wi-Fi passwords, you must use `sudo` to read and archive the files. Create a compressed tarball archive of all configurations.

```
# tar -czvf /tmp/nm-config-backup-$(date +%F).tar.gz /etc/NetworkManager/
```

After restoring reload NetworkManager to apply changes:

```
# nmcli connection reload
# systemctl restart NetworkManager
```

Save a readable snapshot of the current state

These files are not used for automatic restoration, but they are useful for comparing the working setup with the changed configuration:

```
mkdir -p "$HOME/network-state-before-change"

nmcli connection show \
> "$HOME/network-state-before-change/nm-connections.txt"

nmcli device show \
> "$HOME/network-state-before-change/nm-devices.txt"

ip address show \
> "$HOME/network-state-before-change/ip-address.txt"

ip route show table all \
> "$HOME/network-state-before-change/ip-routes.txt"

ip rule show \
> "$HOME/network-state-before-change/ip-rules.txt"
```

For more detail, export every NetworkManager profile in a human-readable form:

```
while IFS= read -r uuid; do
  nmcli --show-secrets connection show uuid "$uuid"
  printf '\n%s\n\n' '-----'
done < <(nmcli -g UUID connection show) \
> "$HOME/network-state-before-change/nm-connections-full.txt"

chmod 600 "$HOME/network-state-before-change/nm-connections-full.txt"
```

Because `--show-secrets` may reveal credentials, protect this file carefully. Omit `--show-secrets` if the secrets are not needed in the reference copy.

From:

<https://wiki.oscardegroot.nl/> - HomeWiki

Permanent link:

<https://wiki.oscardegroot.nl/doku.php?id=networking:networkmanager-nmcli>



Last update: **2026/09/14 16:50**