Cheatography

NetworkManager Cheat Sheet

by misterrabinhalder via cheatography.com/35799/cs/19264/

Installation

RedHat

```
yum install NetworkManager
yum install NetworkManager-tui
```

Archlinux

```
pacman -S networkmanager
```

Debian

```
apt-get install network-manager
```

NetworkManager Initialization

Systemd

```
systemctl start NetworkManager
systemctl enable NetworkManager
systemctl status NetworkManager
```

General Commands

nmcli general status	Show overall status of NetworkManager
nmcli general hostname [hostname]	Get and change system hostname
nmcli general permissions	Show the permissions
nmcli general logging [level level] [domains domains...]	Get and change NetworkManager logging level and domains

Activity Monitor

nmcli monitor	Observe NetworkManager activity

Networking Control Commands

nmcli networking [on | off]	Enable or disable networking control by NetworkManager
nmcli networking connectivity [check]	connectivity [check]

Radio Transmission Control Commands

nmcli radio wifi [on | off]	Show or set status of Wi-Fi
nmcli radio wifi wwan [on | off]	Show or set status of WWAN
nmcli radio wifi all [on | off]	Show or set all previously mentioned radio switches at the same time

Secret Agent

nmcli agent {secret | polkit | all}	Run nmcli as a NetworkManager secret agent, or polkit agent

Examples

Listing available Wi-Fi APs

```
nmcli device wifi list
```

Showing general information and properties for a Wi-Fi interface

```
nmcli -p -f general,wifi-properties device show wlan0
```

Listing NetworkManager polkit permissions

```
nmcli general permissions
```

Listing NetworkManager log level and domains

```
nmcli general logging
```

Changing NetworkManager logging

```
nmcli g log level DEBUG domains CORE,ETHER,IP
nmcli g log level INFO domains DEFAULT
```

Activating a VPN connection profile requiring interactive password input

```
nmcli --ask con up my-vpn-con
```

Adding a bonding master and two slave connection profiles

```
nmcli con add type bond ifname mybond0 mode active-backup
nmcli con add type ethernet ifname eth1 master mybond0
nmcli con add type ethernet ifname eth2 master mybond0
```

Examples (cont)

Adding a team master and two slave connection profiles

```
nmcli con add type team con-name Team1 ifname Team1 config team1--master-json.conf
nmcli con add type ethernet con-name Team1-slave1 ifname em1 master Team1
nmcli con add type ethernet con-name Team1-slave2 ifname em2 master Team1
nmcli con add type ethernet con-name Team1-slave2 ifname em2 master Team1
nmcli con up Team1-slave1
nmcli con up Team1-slave2
```

Adding a bridge and two slave profiles

```
nmcli con add type bridge con-name TowerBridge ifname TowerBridge
nmcli con add type ethernet con-name br-slave-1 ifname ens3 master TowerBridge
nmcli con add type ethernet con-name br-slave-2 ifname ens4 master TowerBridge
nmcli con modify TowerBridge bridge.stp no
```

Adding an ethernet connection profile with manual IP configuration

```
nmcli con add con-name my-con-em1 ifname em1 type ethernet ip4 192.168.100.100/24 gw4 192.168.100.1 ip4 1.2.3.4 ip6 abbe::cafe
nmcli con mod my-con-em1 ipv4.dns "8.8.8.8 8.8.4.4"
nmcli con mod my-con-em1 +ipv4.dns 1.2.3.4
nmcli con mod my-con-em1 ipv6.dns "2001:4860:4860::8888 2001:4860:4860::8844"
nmcli -p con show my-con-em1
```

Convenient field values retrieval for scripting

```
nmcli -g ip4.address connection show my-con-eth0
nmcli -g ip4.address,ip4.dns connection show my-con-eth0
nmcli -g ip4 connection show my-con-eth0
```

Adding an Ethernet connection and configuring SR-IOV VFs

```
nmcli con add type ethernet con-name EthernetPF ifname em1
nmcli con modify EthernetPF sriov.total-vfs 3 sriov.autoprobe-drivers false
nmcli con modify EthernetPF sriov.vfs '0 mac=00:11:22:33:44:55 vlans=10, 1 trust=true spoof-check=false'
nmcli con modify EthernetPF +sriov.vfs '2 max-tx-rate=20'
```

Escaping colon characters in tabular mode

```
nmcli -t -f general -e yes -m tab dev show eth0
```

Adding an ethernet connection profile in interactive editor

```
nmcli connection edit type ethernet
print
goto ethernet
goto ipv4.addresses
set ipv4.gateway 192.168.1.1
verify
print
set ipv4.dns 8.8.8.8 8.8.4.4
print
verify
save
quit
```

Running NetworkManager

--version | -V	Print NetworkManager software version
--help | -h	Print NetworkManager options
--no-daemon | -n	Do not daemonize
--debug | -d	Print output to STDOUT
--pid-file | -p	Specify location of PID file
--state-file	Specify file for storing state
--config	Specifiy configuration file
--log-level	Set NetworkManager logging
--log--domain	List operations to log
--print-config	Print Networkmanger configuration

Connection Management Commands

nmcli connection show [--active] [--order [+-]category:...]	List all profiles
nmcli connection show [--active] [id | uuid | path | apath] ID...	Show details for specified connections
nmcli connection up [id | uuid | path] ID [ifname ifname] [ap BSSID] [passwd-file file]	Activate a connection
nmcli connection down [id | uuid | path | apath] ID...	Deactivate a connection
nmcli connection modify [--temporary] [id | uuid | path] ID {option value | [+|-]setting.property value} ...	Add, modify or remove properties
nmcli connection add [save {yes | no}] {option value | [+|-]setting.property value} ...	Create a new connection

Connection Management Commands (cont)

nmcli connection edit {[id | uuid | path] ID | [type type] [con-name name]}	Edit an existing connection or add a new one, using an interactive editor
nmcli connection clone [--temporary] [id | uuid | path] ID new_name	Clone a connection
nmcli connection delete [id | uuid | path] ID...	Delete a configured connection
nmcli connection monitor [id | uuid | path] ID...	Monitor connection profile activity
nmcli connection reload	Reload all connection files from disk
nmcli connection load filename...	Load/reload one or more connection files from disk
nmcli connection import [--temporary] type type file file	Import an external/foreign configuration
nmcli connection export [id | uuid | path] ID [file]	Export a connection

Configuring NetworkManager

plugins	Lists plugin separated by ','
auth-polkit	Whether to use PolicyKit for authorization
dhcp	Sets up DHCP client
dns	Set DNS processing mode e.g. dnsmasq, systemd-resolved, unbound, none.
level	Set log level e.g. OFF,ERR,WARN,INFO,DEBUG,TRACE
domains	Set log domain e.g. NONE,ALL,DEFAULT,DHCP,IP.
backend	Set logging backend e.g. syslog, journal

Device Management Commands

nmcli device status	Print status of devices
nmcli device show [ifname]	Show detailed information about devices
nmcli device set [ifname] ifname [autoconnect {yes |no}] [managed {yes | no}]	Set device properties
nmcli device connect ifname	Connect the device
nmcli device reapply ifname	Attempt to update device

Device Management Commands (cont)

nmcli device modify ifname {option value | [+|-]setting.property value} ...	Modify the settings
nmcli device disconnect ifname...	Disconnect a device
nmcli device delete ifname...	Delete a device
nmclie device monitor [ifname...]	Monitor device activity
nmcli device wifi [list [--rescan | auto | no | yes] [ifname ifname] [bssid BSSID]]	List available Wi-Fi access points
nmcli device wifi connect (B)SSID [password password] [wep-key-type {key | phrase}] [ifname ifname] [bssid BSSID] [name name] [private {yes | no}] [hidden {yes | no}]	Connect to a Wi-Fi network

Device Management Commands (cont)

nmcli device wifi hotspot [ifname ifname] [con-name name] [ssid SSID] [band {a | bg}] [channel channel] [password password]	Create a Wi-Fi hotspot
nmcli device wifi rescan [ifname ifname] [ssid SSID...]	Re-scan for available access points
nmcli device lldp [list [ifname ifname]]	Display information about neighboring devices

Dispatcher Scripts

Script should be owned by root

> chown root:root /etc/NetworkManager/dispatcher.d/10-script.sh

Must not be writable by group or other

> chmod 755 /etc/NetworkManager/dispatcher.d/10-script.sh

Each script receives two arguments

> The first argument is the interface name
> The second argument is the network action e.g. up, down, etc.