

**NAME**

nm-settings-keyfile – Description of *keyfile* settings plugin

**DESCRIPTION**

NetworkManager is based on the concept of connection profiles that contain network configuration (see **nm-settings(5)** for details). The profiles can be stored in various formats. NetworkManager uses plugins for reading and writing the data. The plugins can be configured in **NetworkManager.conf(5)**.

The *keyfile* plugin is the generic plugin that supports all the connection types and capabilities that NetworkManager has. The files are in a .ini-style format and located in /etc/NetworkManager/system-connections/, /usr/lib/NetworkManager/system-connections/ and /run/NetworkManager/system-connections/. This plugin is always enabled and will automatically be used to store any connections that are not supported by any other active plugin. For security, it will ignore files that are readable or writable by any user other than 'root' since private keys and passphrases may be stored in plaintext inside the file.

**FILE FORMAT**

The *keyfile* config format is a simple .ini-style format. It consists of sections (groups) of key-value pairs. Each section corresponds to a setting name as described in the settings specification (**nm-settings(5)**). Each configuration key/value pair in the section is one of the properties listed in the settings specification. The majority of properties of the specification is written in the same format into the *keyfile* too. However some values are inconvenient for people to use. These are stored in the files in more readable ways. These properties are described below. An example could be IP addresses that are not written as integer arrays, but more reasonably as "1.2.3.4/12 1.2.3.254". More information of the generic key file format can be found at [GLib key file format](#)<sup>[1]</sup> (Lines beginning with a '#' are comments, lists are separated by character ; etc.).

Users can create or modify the *keyfile* connection files manually, even if that is not the recommended way of managing the profiles. However, if they choose to do that, they must inform NetworkManager about their changes (for example via *nmcli con (re)load*).

**Examples of keyfile configuration.****A sample configuration for an ethernet network:**

```
[connection]
id=Main eth0
uuid=27afa607-ee36-43f0-b8c3-9d245cdc4bb3
type=802-3-ethernet
autoconnect=true

[ipv4]
method=auto

[802-3-ethernet]
mac-address=00:23:5a:47:1f:71
```

**A sample configuration for WPA-EAP (PEAP with MSCHAPv2) and always-ask secret:**

```
[connection]
id=CompanyWIFI
uuid=cdac6154-a33b-4b15-9904-666772cfa5ee
type=wifi
autoconnect=false

[wifi]
ssid=CorpWLAN
mode=infrastructure
```

```
security=802-11-wireless-security
```

```
[wifi-security]
```

```
key-mgmt=wpa-eap
```

```
[ipv4]
```

```
method=auto
```

```
[ipv6]
```

```
method=auto
```

```
[802-1x]
```

```
eap=peap;
```

```
identity=joe
```

```
ca-cert=/home/joe/.cert/corp.crt
```

```
phase1-peapver=1
```

```
phase2-auth=mschapv2
```

```
password-flags=2
```

#### **A sample configuration for openvpn:**

```
[connection]
```

```
id=RedHat-openvpn
```

```
uuid=7f9b3356-b210-4c0e-8123-bd116c9c280f
```

```
type=vpn
```

```
timestamp=1385401165
```

```
[vpn]
```

```
service-type=org.freedesktop.NetworkManager.openvpn
```

```
connection-type=password
```

```
password-flags=3
```

```
remote=ovpn.my-company.com
```

```
cipher=AES-256-CBC
```

```
reneg-seconds=0
```

```
port=443
```

```
username=joe
```

```
ca=/etc/openvpn/ISCA.pem
```

```
tls-remote=ovpn.my-company.com
```

```
[ipv6]
```

```
method=auto
```

```
[ipv4]
```

```
method=auto
```

```
ignore-auto-dns=true
```

```
never-default=true
```

#### **A sample configuration for a bridge and a bridge port:**

```
[connection]
```

```
id=MainBridge
```

```
uuid=171ae855-a0ab-42b6-bd0c-60f5812eea9d
```

```
[connection]
```

```
id=br-port-1
```

```
uuid=d6e8ae98-71f8-4b3d-9d2d-2e26048fe794
```

```

interface-name=MainBridge      interface-name=em1
type=bridge                    type=ethernet
                                master=MainBridge
[bridge]                       slave-type=bridge
interface-name=MainBridge

```

#### A sample configuration for a VLAN:

```

[connection]
id=VLAN for building 4A
uuid=8ce1c9e0-ce7a-4d2c-aa28-077dda09dd7e
interface-name=VLAN-4A
type=vlan

[vlan]
interface-name=VLAN-4A
parent=eth0
id=4

```

## DETAILS

*keyfile* plugin variables for the majority of NetworkManager properties have one-to-one mapping. It means a NetworkManager property is stored in the keyfile as a variable of the same name and in the same format. There are several exceptions to this rule, mainly for making keyfile syntax easier for humans. The exceptions handled specially by *keyfile* plugin are listed below. Refer to **nm-settings(5)** for all available settings and properties and their description.

**Name aliases.** Some of the NetworkManager setting names are somewhat hard to type or remember. Therefore *keyfile* introduces aliases that can be used instead of the names.

<i>setting name</i>	<i>keyfile alias</i>
802-3-ethernet	= ethernet
802-11-wireless	= wifi
802-11-wireless-security	= wifi-security

**Table 1. bridge setting (section)**

Property	Keyfile Variable	Format	Description
mac-address	mac-address	usual hex-digits-and-colons notation	MAC address in traditional hex-digits-and-colons notation, or semicolon separated list of 6 decimal bytes (obsolete)  <b>Example:</b> mac-address=00:22:68:12:79:A2 mac-address=0;34;104;18;121;162;

**Table 2. infiniband setting (section)**

Property	Keyfile Variable	Format	Description
mac-address	mac-address	usual hex-digits-and-colons notation	MAC address in traditional hex-digits-and-colons notation, or or semicolon separated list of 20 decimal bytes (obsolete)  <b>Example:</b> mac-address= 80:00:00:6d:fe:80:00:00:00:00:00:00:00:02:55:00:70:3

**Table 3. ipv4 setting (section)**

Property	Keyfile Variable	Format	Description
dns	dns	list of DNS IP addresses	List of DNS servers.  <b>Example:</b> dns=1.2.3.4;8.8.8.8;8.8.4.4;
addresses	address1, address2, ...	address/plen	List of static IP addresses.  <b>Example:</b> address1=192.168.100.100/24 address2=10.1.1.5/24
gateway	gateway	string	Gateway IP addresses as a string.  <b>Example:</b> gateway=192.168.100.1
routes	route1, route2, ...	route/plen[,gateway,metric]	List of IP routes.  <b>Example:</b> route1=8.8.8.0/24,10.1.1.1,77 route2=7.7.0.0/16

**Table 4. ipv6 setting (section)**

Property	Keyfile Variable	Format	Description
dns	dns	list of DNS IP addresses	List of DNS servers.  <b>Example:</b> dns=2001:4860:4860::8888;2001:4860:4860::884
addresses	address1, address2, ...	address/plen	List of static IP addresses.  <b>Example:</b> address1=abbe::cafe/96 address2=2001::1234
gateway	gateway	string	Gateway IP addresses as a string.  <b>Example:</b> gateway=abbe::1
routes	route1, route2, ...	route/plen[,gateway,metric]	List of IP routes.  <b>Example:</b> route1=2001:4860:4860::/64,2620:52:0:2219:222:

**Table 5. serial setting (section)**

Property	Keyfile Variable	Format	Description
parity	parity	'e', 'o', or 'n'	The connection parity; even, odd, or none. Note that older versions of NetworkManager stored this as an integer: 69 ('E') for even, 111 ('o') for odd, or 110 ('n') for none.  <b>Example:</b> parity=n

**Table 6. vpn setting (section)**

Property	Keyfile Variable	Format	Description
data	separate variables named after keys of the dictionary		The keys of the data dictionary are used as variable names directly under [vpn] section.  <b>Example:</b> remote=ovpn.corp.com cipher=AES-256-CBC username=joe
secrets	separate variables named after keys of the dictionary		The keys of the secrets dictionary are used as variable names directly under [vpn-secrets] section.  <b>Example:</b> password=Popocatepetl

**Table 7. wifi-p2p setting (section)**

Property	Keyfile Variable	Format	Description
peer	peer	usual hex-digits-and-colons notation	MAC address in traditional hex-digits-and-colons notation (e.g. 00:22:68:12:79:A2), or semicolon separated list of 6 bytes (obsolete) (e.g. 0;34;104;18;121;162).

**Table 8. 802-3-ethernet setting (section)**

Property	Keyfile Variable	Format	Description
mac-address	mac-address	usual hex-digits-and-colons notation	MAC address in traditional hex-digits-and-colons notation (e.g. 00:22:68:12:79:A2), or semicolon separated list of 6 bytes (obsolete) (e.g. 0;34;104;18;121;162)
cloned-mac-address	cloned-mac-address	usual hex-digits-and-colons notation	Cloned MAC address in traditional hex-digits-and-colons notation (e.g. 00:22:68:12:79:B2), or semicolon separated list of 6 bytes (obsolete) (e.g. 0;34;104;18;121;178).
mac-address-blacklist	mac-address-blacklist	list of MACs (separated with semicolons)	MAC address blacklist.  <b>Example:</b> mac-address-blacklist= 00:22:68:12:79:A6;00:22:68:12:79:78

**Table 9. 802-11-wireless setting (section)**

Property	Keyfile Variable	Format	Description
ssid	ssid	string (or decimal-byte list – obsolete)	SSID of Wi-Fi network.  <b>Example:</b> ssid=Quick Net
mac-address	mac-address	usual hex-digits-and-colons notation	MAC address in traditional hex-digits-and-colons notation (e.g. 00:22:68:12:79:A2), or semicolon separated list of 6 bytes (obsolete) (e.g. 0;34;104;18;121;162).
cloned-mac-address	cloned-mac-address	usual hex-digits-and-colons notation	Cloned MAC address in traditional hex-digits-and-colons notation (e.g. 00:22:68:12:79:B2), or semicolon separated list of 6 bytes (obsolete) (e.g. 0;34;104;18;121;178).
mac-address-blacklist	mac-address-blacklist	list of MACs (separated with semicolons)	MAC address blacklist.  <b>Example:</b> mac-address-blacklist=00:22:68:12:79:A6;00:22:68:12:79:78

**Table 10. wpan setting (section)**

Property	Keyfile Variable	Format	Description
mac-address	mac-address	usual hex-digits-and-colons notation	MAC address in hex-digits-and-colons notation (e.g. 76:d8:9b:87:66:60:84:ee).

**Secret flags**

Each secret property in a NetworkManager setting has an associated *flags* property that describes how to handle that secret. In the *keyfile* plugin, the value of *–flags* variable is a decimal number (0 – 7) defined as a sum of the following values:

- 0 – (NM owned) – the system is responsible for providing and storing this secret.
- 1 – (agent-owned) – a user-session secret agent is responsible for providing and storing this secret; when it is required, agents will be asked to provide it.
- 2 – (not-saved) – this secret should not be saved but should be requested from the user each time it is required.
- 4 – (not-required) – in some situations it cannot be automatically determined that a secret is required or not. This flag hints that the secret is not required and should not be requested from the user.

**FILES**

/etc/NetworkManager/system-connections/\*

**SEE ALSO**

**nm-settings(5)**, **nm-settings-ifcfg-rh(5)**, **NetworkManager(8)**, **NetworkManager.conf(5)**, **nmcli(1)**,  
**nmcli-examples(7)**

**NOTES**

1. GLib key file format  
<https://developer.gnome.org/glib/stable/glib-Key-value-file-parser.html#glib-Key-value-file-parser.description>