NAME

nm-settings-dbus − Description of settings and properties of NetworkManager connection profiles on the D−Bus API

DESCRIPTION

NetworkManager is based on a concept of connection profiles, sometimes referred to as connections only. These connection profiles contain a network configuration. When NetworkManager activates a connection profile on a network device the configuration will be applied and an active network connection will be established. Users are free to create as many connection profiles as they see fit. Thus they are flexible in having various network configurations for different networking needs. The connection profiles are handled by NetworkManager via *settings service* and are exported on D−Bus (*/org/freedesktop/NetworkManager/Settings/<num>* objects). The conceptual objects can be described as follows:

Connection (profile)
: A specific, encapsulated, independent group of settings describing all the configuration required to connect to a specific network. It is referred to by a unique identifier called the UUID. A connection is tied to a one specific device type, but not necessarily a specific hardware device. It is composed of one or more *Settings* objects.

Setting
: A group of related key/value pairs describing a specific piece of a *Connection (profile)*. Settings keys and allowed values are described in the tables below. Keys are also referred to as properties. Developers can find the setting objects and their properties in the libnm−core sources. Look for the ***_class_init** functions near the bottom of each setting source file.

The settings and properties shown in tables below list all available connection configuration options. However, note that not all settings are applicable to all connection types. NetworkManager provides a command−line tool *nmcli* that allows direct configuration of the settings and properties according to a connection profile type. *nmcli* connection editor has also a built−in *describe* command that can display description of particular settings and properties of this page.

connection setting
: General Connection Profile Settings.

6lowpan setting

6LoWPAN Settings.

Key Name	Value Type	Default Value	Value Description
parent	string		If given, specifies the parent interface name or parent connection UUID from which this 6LowPAN interface should be created.

802−1x setting

IEEE 802.1x Authentication Settings.

adsl setting

ADSL Settings.

Key Name	Value Type	Default Value	Value Description
encapsulation	string		Encapsulation of ADSL connection. Can be "vcmux" or "llc".
password	string		Password used to authenticate with the ADSL service.
password−flags	NMSettingSecretFlags (uint32)		Flags indicating how to handle the "password" property. (see the section called “Secret flag types:” for flag values)
protocol	string		ADSL connection protocol. Can be "pppoa", "pppoe" or "ipoatm".
username	string		Username used to authenticate with the ADSL service.
vci	uint32	0	VCI of ADSL connection
vpi	uint32	0	VPI of ADSL connection

bluetooth setting

Bluetooth Settings.

Key Name	Value Type	Default Value	Value Description
bdaddr	byte array		The Bluetooth address of the device.
type	string		Either "dun" for Dial−Up Networking connections or "panu" for Personal Area Networking connections to devices supporting the NAP profile.

bond setting

Bonding Settings.

Key Name	Value Type	Default Value	Value Description
interface−name	string		Deprecated in favor of connection.interface−name, but can be used for backward−compatibility with older daemons, to set the bond's interface name.
options	dict of string to string	{'mode': 'balance−rr'}	Dictionary of key/value pairs of bonding options. Both keys and values must be strings. Option names must contain only alphanumeric characters (ie, [a−zA−Z0−9]).

bridge setting

Bridging Settings.

bridge−port setting

Bridge Port Settings.

Key Name	Value Type	Default Value	Value Description
hairpin−mode	boolean	FALSE	Enables or disables "hairpin mode" for the port, which allows frames to be sent back out through the port the frame was received on.
path−cost	uint32	100	The Spanning Tree Protocol (STP) port cost for destinations via this port.
priority	uint32	32	The Spanning Tree Protocol (STP) priority of this bridge port.
vlans	array of vardict		Array of bridge VLAN objects. In addition to the VLANs specified here, the port will also have the default−pvid VLAN configured on the bridge by the bridge.vlan−default−pvid property. In nmcli the VLAN list can be specified with the following syntax: $vid [pvid] [untagged] [, $vid [pvid] [untagged]]... where $vid is either a single id between 1 and 4094 or a range, represented as a couple of ids separated by a dash.

cdma setting

CDMA−based Mobile Broadband Settings.

Key Name	Value Type	Default Value	Value Description
mtu	uint32	0	If non−zero, only transmit packets of the specified size or smaller, breaking larger packets up into multiple frames.
number	string		The number to dial to establish the connection to the CDMA−based mobile broadband network, if any. If not specified, the default number (#777) is used when required.
password	string		The password used to authenticate with the network, if required. Many providers do not require a password, or accept any password. But if a password is required, it is specified here.
password−flags	NMSettingSecretFlags (uint32)		Flags indicating how to handle the "password" property. (see the section called “Secret flag types:” for flag values)
username	string		The username used to authenticate with the network, if required. Many providers do not require a username, or accept any username. But if a username is required, it is specified here.

dcb setting

Data Center Bridging Settings.

dummy setting
Dummy Link Settings.

ethtool setting
Ethtool Ethernet Settings.

generic setting
Generic Link Settings.

gsm setting
GSM−based Mobile Broadband Settings.

infiniband setting

Infiniband Settings.

Key Name	Value Type	Default Value	Value Description
mac−address	byte array		If specified, this connection will only apply to the IPoIB device whose permanent MAC address matches. This property does not change the MAC address of the device (i.e. MAC spoofing).
mtu	uint32	0	If non−zero, only transmit packets of the specified size or smaller, breaking larger packets up into multiple frames.
p−key	int32	−1	The InfiniBand P_Key to use for this device. A value of −1 means to use the default P_Key (aka "the P_Key at index 0"). Otherwise, it is a 16−bit unsigned integer, whose high bit is set if it is a "full membership" P_Key.
parent	string		The interface name of the parent device of this device. Normally NULL, but if the "p_key" property is set, then you must specify the base device by setting either this property or "mac−address".
transport−mode	string		The IP−over−InfiniBand transport mode. Either "datagram" or "connected".

ipv4 setting

IPv4 Settings.

ipv6 setting
IPv6 Settings.

ip−tunnel setting

IP Tunneling Settings.

Key Name	Value Type	Default Value	Value Description
encapsulation−limit	uint32	0	How many additional levels of encapsulation are permitted to be prepended to packets. This property applies only to IPv6 tunnels.
flags	uint32	0	Tunnel flags. Currently, the following values are supported: NM_IP_TUNNEL_FLAG_IP6_IGN_ENCAP_LIMIT (0x1), NM_IP_TUNNEL_FLAG_IP6_USE_ORIG_TCLASS (0x2), NM_IP_TUNNEL_FLAG_IP6_USE_ORIG_FLOWLABEL (0x4), NM_IP_TUNNEL_FLAG_IP6_MIP6_DEV (0x8), NM_IP_TUNNEL_FLAG_IP6_RCV_DSCP_COPY (0x10), NM_IP_TUNNEL_FLAG_IP6_USE_ORIG_FWMARK (0x20). They are valid only for IPv6 tunnels.
flow−label	uint32	0	The flow label to assign to tunnel packets. This property applies only to IPv6 tunnels.
input−key	string		The key used for tunnel input packets; the property is valid only for certain tunnel modes (GRE, IP6GRE). If empty, no key is used.
local	string		The local endpoint of the tunnel; the value can be empty, otherwise it must contain an IPv4 or IPv6 address.
mode	uint32	0	The tunneling mode, for example NM_IP_TUNNEL_MODE_IPIP (1) or NM_IP_TUNNEL_MODE_GRE (2).
mtu	uint32	0	If non−zero, only transmit packets of the specified size or smaller, breaking larger packets up into multiple fragments.
output−key	string		The key used for tunnel output packets; the property is valid only for certain tunnel modes (GRE, IP6GRE). If empty, no key is used.
parent	string		If given, specifies the parent interface name or parent connection UUID the new device will be bound to so that tunneled packets will only be routed via that interface.
path−mtu−discovery	boolean	TRUE	Whether to enable Path MTU Discovery on this tunnel.
remote	string		The remote endpoint of the tunnel; the value must contain an IPv4 or IPv6 address.
tos	uint32	0	The type of service (IPv4) or traffic class (IPv6) field to be set on tunneled packets.
ttl	uint32	0	The TTL to assign to tunneled packets. 0 is a special value meaning that packets inherit the TTL value.

macsec setting

MACSec Settings.

Key Name	Value Type	Default Value	Value Description
encrypt	boolean	TRUE	Whether the transmitted traffic must be encrypted.
mka−cak	string		The pre−shared CAK (Connectivity Association Key) for MACsec Key Agreement.
mka−cak−flags	NMSettingSecretFlags (uint32)		Flags indicating how to handle the "mka−cak" property. (see the section called “Secret flag types:” for flag values)
mka−ckn	string		The pre−shared CKN (Connectivity−association Key Name) for MACsec Key Agreement.
mode	int32	0	Specifies how the CAK (Connectivity Association Key) for MKA (MACsec Key Agreement) is obtained.
parent	string		If given, specifies the parent interface name or parent connection UUID from which this MACSEC interface should be created. If this property is not specified, the connection must contain an "802−3−ethernet" setting with a "mac−address" property.
port	int32	1	The port component of the SCI (Secure Channel Identifier), between 1 and 65534.
send−sci	boolean	TRUE	Specifies whether the SCI (Secure Channel Identifier) is included in every packet.
validation	int32	2	Specifies the validation mode for incoming frames.

macvlan setting

MAC VLAN Settings.

Key Name	Value Type	Default Value	Value Description
mode	uint32	0	The macvlan mode, which specifies the communication mechanism between multiple macvlans on the same lower device.
parent	string		If given, specifies the parent interface name or parent connection UUID from which this MAC−VLAN interface should be created. If this property is not specified, the connection must contain an "802−3−ethernet" setting with a "mac−address" property.
promiscuous	boolean	TRUE	Whether the interface should be put in promiscuous mode.
tap	boolean	FALSE	Whether the interface should be a MACVTAP.

match setting

Match settings.

802−11−olpc−mesh setting

OLPC Wireless Mesh Settings.

Key Name	Value Type	Default Value	Value Description
channel	uint32	0	Channel on which the mesh network to join is located.
dhcp−anycast−address	byte array		Anycast DHCP MAC address used when requesting an IP address via DHCP. The specific anycast address used determines which DHCP server class answers the request. This is currently only implemented by dhclient DHCP plugin.
ssid	byte array		SSID of the mesh network to join.

ovs−bridge setting

OvsBridge Link Settings.

Key Name	Value Type	Default Value	Value Description
datapath−type	string		The data path type. One of "system", "netdev" or empty.
fail−mode	string		The bridge failure mode. One of "secure", "standalone" or empty.
mcast−snooping−enable	boolean	FALSE	Enable or disable multicast snooping.
rstp−enable	boolean	FALSE	Enable or disable RSTP.
stp−enable	boolean	FALSE	Enable or disable STP.

ovs−dpdk setting

OvsDpdk Link Settings.

Key Name	Value Type	Default Value	Value Description
devargs	string		Open vSwitch DPDK device arguments.
n−rxq	uint32	0	Open vSwitch DPDK number of rx queues. Defaults to zero which means to leave the parameter in OVS unspecified and effectively configures one queue.

ovs−interface setting

Open vSwitch Interface Settings.

Key Name	Value Type	Default Value	Value Description
type	string		The interface type. Either "internal", "system", "patch", "dpdk", or empty.

ovs−patch setting

OvsPatch Link Settings.

Key Name	Value Type	Default Value	Value Description
peer	string		Specifies the name of the interface for the other side of the patch. The patch on the other side must also set this interface as peer.

ovs−port setting

OvsPort Link Settings.

Key Name	Value Type	Default Value	Value Description
bond−downdelay	uint32	0	The time port must be inactive in order to be considered down.
bond−mode	string		Bonding mode. One of "active−backup", "balance−slb", or "balance−tcp".
bond−updelay	uint32	0	The time port must be active before it starts forwarding traffic.
lacp	string		LACP mode. One of "active", "off", or "passive".
tag	uint32	0	The VLAN tag in the range 0−4095.
vlan−mode	string		The VLAN mode. One of "access", "native−tagged", "native−untagged", "trunk" or unset.

ppp setting

Point−to−Point Protocol Settings.

pppoe setting

PPP−over−Ethernet Settings.

Key Name	Value Type	Default Value	Value Description
parent	string		If given, specifies the parent interface name on which this PPPoE connection should be created. If this property is not specified, the connection is activated on the interface specified in "interface−name" of NMSettingConnection.
password	string		Password used to authenticate with the PPPoE service.
password−flags	NMSettingSecretFlags (uint32)		Flags indicating how to handle the "password" property. (see the section called “Secret flag types:” for flag values)
service	string		If specified, instruct PPPoE to only initiate sessions with access concentrators that provide the specified service. For most providers, this should be left blank. It is only required if there are multiple access concentrators or a specific service is known to be required.
username	string		Username used to authenticate with the PPPoE service.

proxy setting

WWW Proxy Settings.

Key Name	Value Type	Default Value	Value Description
browser−only	boolean	FALSE	Whether the proxy configuration is for browser only.
method	int32	0	Method for proxy configuration, Default is NM_SETTING_PROXY_METHOD_NONE (0)
pac−script	string		PAC script for the connection.
pac−url	string		PAC URL for obtaining PAC file.

serial setting

Serial Link Settings.

Key Name	Value Type	Default Value	Value Description
baud	uint32	57600	Speed to use for communication over the serial port. Note that this value usually has no effect for mobile broadband modems as they generally ignore speed settings and use the highest available speed.
bits	uint32	8	Byte−width of the serial communication. The 8 in "8n1" for example.
parity	byte		The connection parity: 69 (ASCII 'E') for even parity, 111 (ASCII 'o') for odd, 110 (ASCII 'n') for none.
send−delay	uint64	0	Time to delay between each byte sent to the modem, in microseconds.
stopbits	uint32	1	Number of stop bits for communication on the serial port. Either 1 or 2. The 1 in "8n1" for example.

sriov setting

SR−IOV settings.

tc setting

Linux Traffic Control Settings.

Key Name	Value Type	Default Value	Value Description
qdiscs	array of vardict		Array of TC queueing disciplines. When the "tc" setting is present, qdiscs from this property are applied upon activation. If the property is empty, all qdiscs are removed and the device will only have the default qdisc assigned by kernel according to the "net.core.default_qdisc" sysctl. If the "tc" setting is not present, NetworkManager doesn't touch the qdiscs present on the interface.
tfilters	array of vardict		Array of TC traffic filters. When the "tc" setting is present, filters from this property are applied upon activation. If the property is empty, NetworkManager removes all the filters. If the "tc" setting is not present, NetworkManager doesn't touch the filters present on the interface.

team setting

Teaming Settings.

team−port setting

Team Port Settings.

Key Name	Value Type	Default Value	Value Description
config	string		The JSON configuration for the team port. The property should contain raw JSON configuration data suitable for teamd, because the value is passed directly to teamd. If not specified, the default configuration is used. See man teamd.conf for the format details.
lacp−key	int32	−1	Corresponds to the teamd ports.PORTIFNAME.lacp_key.
lacp−prio	int32	−1	Corresponds to the teamd ports.PORTIFNAME.lacp_prio.
link−watchers	array of vardict		Link watchers configuration for the connection: each link watcher is defined by a dictionary, whose keys depend upon the selected link watcher. Available link watchers are 'ethtool', 'nsna_ping' and 'arp_ping' and it is specified in the dictionary with the key 'name'. Available keys are: ethtool: 'delay−up', 'delay−down', 'init−wait'; nsna_ping: 'init−wait', 'interval', 'missed−max', 'target−host'; arp_ping: all the ones in nsna_ping and 'source−host', 'validate−active', 'validate−inactive', 'send−always'. See teamd.conf man for more details.
prio	int32	0	Corresponds to the teamd ports.PORTIFNAME.prio.
queue−id	int32	−1	Corresponds to the teamd ports.PORTIFNAME.queue_id. When set to −1 means the parameter is skipped from the json config.
sticky	boolean	FALSE	Corresponds to the teamd ports.PORTIFNAME.sticky.

tun setting

Tunnel Settings.

Key Name	Value Type	Default Value	Value Description
group	string		The group ID which will own the device. If set to NULL everyone will be able to use the device.
mode	uint32	1	The operating mode of the virtual device. Allowed values are NM_SETTING_TUN_MODE_TUN (1) to create a layer 3 device and NM_SETTING_TUN_MODE_TAP (2) to create an Ethernet−like layer 2 one.
multi−queue	boolean	FALSE	If the property is set to TRUE, the interface will support multiple file descriptors (queues) to parallelize packet sending or receiving. Otherwise, the interface will only support a single queue.
owner	string		The user ID which will own the device. If set to NULL everyone will be able to use the device.
pi	boolean	FALSE	If TRUE the interface will prepend a 4 byte header describing the physical interface to the packets.
vnet−hdr	boolean	FALSE	If TRUE the IFF_VNET_HDR the tunnel packets will include a virtio network header.

user setting

General User Profile Settings.

Key Name	Value Type	Default Value	Value Description
data	dict of string to string	{}	A dictionary of key/value pairs with user data. This data is ignored by NetworkManager and can be used at the users discretion. The keys only support a strict ascii format, but the values can be arbitrary UTF8 strings up to a certain length.

vlan setting
VLAN Settings.

vpn setting

VPN Settings.

Key Name	Value Type	Default Value	Value Description
data	dict of string to string	{}	Dictionary of key/value pairs of VPN plugin specific data. Both keys and values must be strings.
persistent	boolean	FALSE	If the VPN service supports persistence, and this property is TRUE, the VPN will attempt to stay connected across link changes and outages, until explicitly disconnected.
secrets	dict of string to string	{}	Dictionary of key/value pairs of VPN plugin specific secrets like passwords or private keys. Both keys and values must be strings.
service−type	string		D−Bus service name of the VPN plugin that this setting uses to connect to its network. i.e. org.freedesktop.NetworkManager.vpnc for the vpnc plugin.
timeout	uint32	0	Timeout for the VPN service to establish the connection. Some services may take quite a long time to connect. Value of 0 means a default timeout, which is 60 seconds (unless overridden by vpn.timeout in configuration file). Values greater than zero mean timeout in seconds.
user−name	string		If the VPN connection requires a user name for authentication, that name should be provided here. If the connection is available to more than one user, and the VPN requires each user to supply a different name, then leave this property empty. If this property is empty, NetworkManager will automatically supply the username of the user which requested the VPN connection.

vrf setting
VRF settings.

Key Name	Value Type	Default Value	Value Description
table	uint32	0	The routing table for this VRF.

vxlan setting
VXLAN Settings.

wifi−p2p setting

Wi−Fi P2P Settings.

Key Name	Value Type	Default Value	Value Description
peer	string		The P2P device that should be connected to. Currently, this is the only way to create or join a group.
wfd−ies	byte array		The Wi−Fi Display (WFD) Information Elements (IEs) to set. Wi−Fi Display requires a protocol specific information element to be set in certain Wi−Fi frames. These can be specified here for the purpose of establishing a connection. This setting is only useful when implementing a Wi−Fi Display client.
wps−method	uint32	0	Flags indicating which mode of WPS is to be used. There's little point in changing the default setting as NetworkManager will automatically determine the best method to use.

wimax setting

WiMax Settings.

Key Name	Value Type	Default Value	Value Description
mac−address	byte array		If specified, this connection will only apply to the WiMAX device whose MAC address matches. This property does not change the MAC address of the device (known as MAC spoofing). Deprecated: 1
network−name	string		Network Service Provider (NSP) name of the WiMAX network this connection should use. Deprecated: 1

802−3−ethernet setting
Wired Ethernet Settings.

wireguard setting
WireGuard Settings.

802−11−wireless setting
Wi−Fi Settings.

802−11−wireless−security setting
Wi−Fi Security Settings.

wpan setting

IEEE 802.15.4 (WPAN) MAC Settings.

Key Name	Value Type	Default Value	Value Description
channel	int32	−1	IEEE 802.15.4 channel. A positive integer or −1, meaning "do not set, use whatever the device is already set to".
mac−address	string		If specified, this connection will only apply to the IEEE 802.15.4 (WPAN) MAC layer device whose permanent MAC address matches.
page	int32	−1	IEEE 802.15.4 channel page. A positive integer or −1, meaning "do not set, use whatever the device is already set to".
pan−id	uint32	65535	IEEE 802.15.4 Personal Area Network (PAN) identifier.
short−address	uint32	65535	Short IEEE 802.15.4 address to be used within a restricted environment.

bond−port setting

Bond Port Settings.

Key Name	Value Type	Default Value	Value Description
queue−id	uint32	0	The queue ID of this bond port. The maximum value of queue ID is the number of TX queues currently active in device.

hostname setting

Hostname settings.

ovs−external−ids setting

OVS External IDs Settings.

Key Name	Value Type	Default Value	Value Description
data	dict of string to string	{}	A dictionary of key/value pairs with exernal−ids for OVS.

veth setting

Veth Settings.

Key Name	Value Type	Default Value	Value Description
peer	string		This property specifies the peer interface name of the veth. This property is mandatory.

Secret flag types:

Each password or secret property in a setting has an associated *flags* property that describes how to handle that secret. The *flags* property is a bitfield that contains zero or more of the following values logically OR−ed together.

- 0x0 (none) − the system is responsible for providing and storing this secret. This may be required so that secrets are already available before the user logs in. It also commonly means that the secret will be stored in plain text on disk, accessible to root only. For example via the keyfile settings plugin as described in the "PLUGINS" section in **NetworkManager.conf**(5).
- 0x1 (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.
- 0x2 (not−saved) − this secret should not be saved but should be requested from the user each time it is required. This flag should be used for One−Time−Pad secrets, PIN codes from hardware tokens, or if the user simply does not want to save the secret.
- 0x4 (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 or distro plugin−specific location

SEE ALSO

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