NAME

NetworkManager-dispatcher − Dispatch user scripts for NetworkManager

SYNOPSIS

NetworkManager [OPTIONS...]

DESCRIPTION

NetworkManager−dispatcher service is a D−Bus activated service that runs user provided scripts upon certain changes in NetworkManager.

NetworkManager−dispatcher will execute scripts in the /{etc,usr/lib}/NetworkManager/dispatcher.d directory or subdirectories in alphabetical order in response to network events. Each script should be a regular executable file owned by root. Furthermore, it must not be writable by group or other, and not setuid.

Each script receives two arguments, the first being the interface name of the device an operation just happened on, and second the action. For device actions, the interface is the name of the kernel interface suitable for IP configuration. Thus it is either VPN_IP_IFACE, DEVICE_IP_IFACE, or DEVICE_IFACE, as applicable. For the *hostname* action the device name is always "none" and for *connectivity−change* it is empty.

The actions are:

pre−up
: The interface is connected to the network but is not yet fully activated. Scripts acting on this event must be placed or symlinked into the /etc/NetworkManager/dispatcher.d/pre−up.d directory, and NetworkManager will wait for script execution to complete before indicating to applications that the interface is fully activated.

up
: The interface has been activated.

pre−down
: The interface will be deactivated but has not yet been disconnected from the network. Scripts acting on this event must be placed or symlinked into the /etc/NetworkManager/dispatcher.d/pre−down.d directory, and NetworkManager will wait for script execution to complete before disconnecting the interface from its network. Note that this event is not emitted for forced disconnections, like when carrier is lost or a wireless signal fades. It is only emitted when there is an opportunity to cleanly handle a network disconnection event.

down
: The interface has been deactivated.

vpn−pre−up
: The VPN is connected to the network but is not yet fully activated. Scripts acting on this event must be placed or symlinked into the /etc/NetworkManager/dispatcher.d/pre−up.d directory, and NetworkManager will wait for script execution to complete before indicating to applications that the VPN is fully activated.

vpn−up
: A VPN connection has been activated.

vpn−pre−down
: The VPN will be deactivated but has not yet been disconnected from the network. Scripts acting on this event must be placed or symlinked into the /etc/NetworkManager/dispatcher.d/pre−down.d directory, and NetworkManager will wait for script execution to complete before disconnecting the VPN from its network. Note that this event is not emitted for forced disconnections, like when the VPN terminates unexpectedly or general connectivity is lost. It is only emitted when there is an opportunity to cleanly handle a VPN disconnection event.

vpn−down
: A VPN connection has been deactivated.

hostname
The system hostname has been updated. Use gethostname(2) to retrieve it. The interface name (first argument) is empty and no environment variable is set for this action.

dhcp4−change
The DHCPv4 lease has changed (renewed, rebound, etc).

dhcp6−change
The DHCPv6 lease has changed (renewed, rebound, etc).

connectivity−change
The network connectivity state has changed (no connectivity, went online, etc).

The environment contains more information about the interface and the connection. The following variables are available for the use in the dispatcher scripts:

NM_DISPATCHER_ACTION
The dispatcher action like "up" or "dhcp4−change", identical to the first command line argument. Since NetworkManager 1.12.0.

CONNECTION_UUID
The UUID of the connection profile.

CONNECTION_ID
The name (ID) of the connection profile.

CONNECTION_DBUS_PATH
The NetworkManager D−Bus path of the connection.

CONNECTION_FILENAME
The backing file name of the connection profile (if any).

CONNECTION_EXTERNAL
If "1", this indicates that the connection describes a network configuration created outside of NetworkManager.

DEVICE_IFACE
The interface name of the control interface of the device. Depending on the device type, this differs from *DEVICE_IP_IFACE*. For example for ADSL devices, this could be 'atm0' or for WWAN devices it might be 'ttyUSB0'.

DEVICE_IP_IFACE
The IP interface name of the device. This is the network interface on which IP addresses and routes will be configured.

IP4_ADDRESS_N
The IPv4 address in the format "address/prefix gateway", where N is a number from 0 to (# IPv4 addresses − 1). gateway item in this variable is deprecated, use IP4_GATEWAY instead.

IP4_NUM_ADDRESSES
The variable contains the number of IPv4 addresses the script may expect.

IP4_GATEWAY
The gateway IPv4 address in traditional numbers−and−dots notation.

IP4_ROUTE_N
The IPv4 route in the format "address/prefix next−hop metric", where N is a number from 0 to (# IPv4 routes − 1).

IP4_NUM_ROUTES
The variable contains the number of IPv4 routes the script may expect.

IP4_NAMESERVERS
The variable contains a space−separated list of the DNS servers.

IP4_DOMAINS

The variable contains a space−separated list of the search domains.

DHCP4_<dhcp−option−name>
: If the connection used DHCP for address configuration, the received DHCP configuration is passed in the environment using standard DHCP option names, prefixed with "DHCP4_", like "DHCP4_HOST_NAME=foobar".

IP6_<name> and DHCP6_<name>
: The same variables as for IPv4 are available for IPv6, but the prefixes are IP6_ and DHCP6_ instead.

CONNECTIVITY_STATE
: The network connectivity state, which can take the values defined by the NMConnectivityState type, from the org.freedesktop.NetworkManager D−Bus API: unknown, none, portal, limited or full. Note: this variable will only be set for connectivity−change actions.

In case of VPN, VPN_IP_IFACE is set, and IP4_*, IP6_* variables with VPN prefix are exported too, like VPN_IP4_ADDRESS_0, VPN_IP4_NUM_ADDRESSES.

Dispatcher scripts are run one at a time, but asynchronously from the main NetworkManager process, and will be killed if they run for too long. If your script might take arbitrarily long to complete, you should spawn a child process and have the parent return immediately. Scripts that are symbolic links pointing inside the /etc/NetworkManager/dispatcher.d/no−wait.d/ directory are run immediately, without waiting for the termination of previous scripts, and in parallel. Also beware that once a script is queued, it will always be run, even if a later event renders it obsolete. (Eg, if an interface goes up, and then back down again quickly, it is possible that one or more "up" scripts will be run after the interface has gone down.)

BUGS

Please report any bugs you find in NetworkManager at the **NetworkManager issue tracker**[1].

SEE ALSO

NetworkManager home page[2], **NetworkManager**(8),

NOTES

1. NetworkManager issue tracker
   https://gitlab.freedesktop.org/NetworkManager/NetworkManager/-/issues

2. NetworkManager home page
   https://networkmanager.dev