

**NAME**

**nm-online** – ask NetworkManager whether the network is connected

**SYNOPSIS**

**nm-online** [*OPTIONS...*]

**DESCRIPTION**

**nm-online** is a utility to find out whether we are online. It is done by asking NetworkManager about its status. When run, **nm-online** waits until NetworkManager reports an active connection, or specified timeout expires. On exit, the returned status code should be checked (see the return codes below).

This tool is not very useful to call directly. It is however used by NetworkManager-wait-online.service with `--wait-for-startup` argument. This is used to delay the service and indirectly network-online.target, until networking is up. Don't order your own systemd services after NetworkManager-wait-online.service directly. Instead if necessary, order your services after network-online.target. Even better is to have your services react to network changes dynamically and don't order them with respect to network-online.target at all.

By default, connections have the `ipv4.may-fail` and `ipv6.may-fail` properties set to yes; this means that NetworkManager waits for one of the two address families to complete configuration before considering the connection activated. If you need a specific address family configured before network-online.target is reached, set the corresponding `may-fail` property to no.

**OPTIONS**

**-h** | **--help**

Print help information.

**-q** | **--quiet**

Don't print anything.

**-s** | **--wait-for-startup**

Wait for NetworkManager startup to complete, rather than waiting for network connectivity specifically. Startup is considered complete once NetworkManager has activated (or attempted to activate) every auto-activate connection which is available given the current network state. This corresponds to the moment when NetworkManager logs "startup complete". This mode is generally only useful at boot time. After startup has completed, **nm-online -s** will just return immediately, regardless of the current network state.

There are various ways to affect when startup complete is reached. For example, by setting a connection profile to `autoconnect`, such a profile possibly will activate during startup and thus delay startup complete being reached. Also, a profile is considered ready when it fully reached the logical connected state in NetworkManager. That means, properties like `ipv4.may-fail` and `ipv6.may-fail` affect whether a certain address family is required. Also, the connection property `connection.wait-device-timeout` affects whether to wait for the driver to detect a certain device. Generally, a failure of NetworkManager-wait-online.service indicates a configuration error, where NetworkManager won't be able to reach the desired connectivity state during startup. An example for that are bridge or bond master profiles, that get autoconnected but without activating any slaves. Such master devices hang in activating state indefinitely, and cause NetworkManager-wait-online.service to fail.

**-t** | **--timeout** *seconds*

Time to wait for a connection, in seconds. If the option is not provided, the environment variable `NM_ONLINE_TIMEOUT` is honored. The default timeout is 30 seconds.

**-x** | **--exit**

Exit immediately if NetworkManager is not running or connecting.

**EXIT STATUS**

**nm-online** exits with status 0 if it succeeds, a value greater than 0 is returned if an error occurs.

0

Success – already online or connection established within given timeout.

**1**

Offline or not online within given timeout.

**2**

Unknown or unspecified error.

**SEE ALSO**

**nmcli(1), NetworkManager(8).**