



## ***Rocky Enterprise Linux 9.2 Manual Pages on command 'ifdown.8'***

**C:\>man ifdown.8**

ifup(8) ifup(8)

### NAME

ifup - bring a network interface up  
ifdown - take a network interface down  
ifquery - parse interface configuration

### SYNOPSIS

```
ifup [-nv] [--no-act] [--verbose] [-i FILE|--interfaces=FILE] [--state-dir=DIR]
[--allow CLASS] -a|IFACE...

ifup -h|--help

ifup -V|--version

ifdown [-nv] [--no-act] [--verbose] [-i FILE|--interfaces=FILE] [--state-dir=DIR]
[--allow CLASS] -a|IFACE...

ifquery [-nv] [--verbose] [-i FILE|--interfaces=FILE] [--state-dir=DIR] [--allow
CLASS] IFACE...

ifquery -l|--list [-nv] [--verbose] [-i FILE|--interfaces=FILE] [--state-dir=DIR]
[--allow CLASS] [-a|IFACE...]

ifquery --state [--state-dir=DIR] [--allow CLASS] [-a|IFACE...]
```

### DESCRIPTION

The ifup and ifdown commands may be used to configure (or, respectively, deconfigure) network interfaces based on interface definitions in the file /etc/network/interfaces. ifquery command may be used to parse interfaces configuration.

### OPTIONS

A summary of options is included below.

**-a, --all**

If given to `ifup`, affect all interfaces marked `auto`. Interfaces are brought up in the order in which they are defined in `/etc/network/interfaces`. Combined with `--allow`, acts on all interfaces of a specified class instead. If given to `ifdown`, affect all defined interfaces. Interfaces are brought down in the order in which they are currently listed in the state file. Only interfaces defined in `/etc/network/interfaces` will be brought down.

**--force**

Force configuration or deconfiguration of the interface.

**--ignore-errors**

If any of the commands of scripts fails, continue.

**-h, --help**

Show summary of options.

**--allow=CLASS**

Only allow interfaces listed in an `allow-CLASS` line in `/etc/network/interfaces` to be acted upon.

**-i FILE, --interfaces=FILE**

Read interface definitions from `FILE` instead of from `/etc/network/interfaces`.

**--state-dir=DIR**

Keep interface state in `DIR` instead of in `/run/network`.

**-X PATTERN, --exclude=PATTERN**

Exclude interfaces from the list of interfaces to operate on by the `PATTERN`. `PATTERN` uses a usual shell glob syntax. If shell wildcards are not used, it must match the exact interface name. This option may be specified multiple times resulting in more than one pattern being excluded.

**-o OPTION=VALUE**

Set `OPTION` to `VALUE` as though it were in `/etc/network/interfaces`.

**-n, --no-act**

Don't configure any interfaces or run any "up" or "down" commands.

**--no-mappings**

Don't run any mappings. See `interfaces(5)` for more information about the

mapping feature.

`--no-scripts`

Don't run any scripts under `/etc/network/if-*.d/`

`--no-loopback`

Disable special handling of the loopback interface. By default, the loopback interface (`lo` on Linux) is predefined internally as an auto interface, so it's brought up on `ifup -a` automatically. In the case the loopback device is redefined by user, the interface is configured just once anyway. If, however, another interface is also defined as loopback, it's configured as usual. Specifying this option disables this behaviour, so the loopback interface won't be configured automatically.

`-V, --version`

Show copyright and version information.

`-v, --verbose`

Show commands as they are executed.

`-l, --list`

For `ifquery`, list all the interfaces which match the specified class. If no class specified, prints all the interfaces listed as auto.

`--state`

For `ifquery`, dump the state of the interfaces. When no interfaces specified, lists all interfaces brought up together with logical interfaces assigned to them and exits with a status code indicating success. If one or more interfaces specified, display state of these interfaces only; successful code is returned if all of interfaces given as arguments are up. Otherwise, 0 is returned.

## EXAMPLES

`ifup -a`

Bring up all the interfaces defined with auto in `/etc/network/interfaces`

`ifup eth0`

Bring up interface `eth0`

`ifup eth0=home`

Bring up interface `eth0` as logical interface `home`

`ifdown -a`

Bring down all interfaces that are currently up.

`ifquery -l`

Print names of all interfaces specified with the `auto` keyword.

`ifquery -l --allow=hotplug`

Print names of all interfaces specified with the `allow-hotplug` keyword.

`ifquery eth0`

Display the interface options as specified in the `ifupdown` configuration.

Each key-value pair is printed out on individual line using ":" as separator.

## NOTES

`ifup`, `ifdown`, and `ifquery` are actually the same program called by different names.

The program does not configure network interfaces directly; it runs low level utilities such as `ip` to do its dirty work.

When invoked, `ifdown` checks if `ifup` is still running. In that case, `SIGTERM` is sent to `ifup`.

During interface deconfiguration, `ifdown` ignores errors the same way as `if --ignore-errors` was specified.

## FILES

`/etc/network/interfaces`

definitions of network interfaces See `interfaces(5)` for more information.

`/run/network/ifstate`

current state of network interfaces

## CONCURRENCY

`ifupdown` uses per-interface locking to ensure that concurrent `ifup` and `ifdown` calls to the same interface are run in serial. However, calls to different interfaces will be able to run in parallel.

## EXIT STATUS

For `ifup` and `ifdown`, the exit status will be 0 if the given interface(s) have all been (de)configured successfully, 1 if there was any error. The result of these commands is idempotent; running `ifup` on an interface that is already up will result in an exit status of 0, and similarly running `ifdown` on an interface that is not up will also result in an exit status of 0.

`ifquery` will normally return with exit status 0 if an interface with a matching

iface stanza, 1 if there is no matching stanza. `ifquery --state` will also return with exit status 1 if the given interface was known but was not up.

## KNOWN BUGS/LIMITATIONS

The program keeps records of whether network interfaces are up or down. Under exceptional circumstances these records can become inconsistent with the real states of the interfaces. For example, an interface that was brought up using `ifup` and later deconfigured using `ifconfig` will still be recorded as up. To fix this you can use the `--force` option to force `ifup` or `ifdown` to run configuration or deconfiguration commands despite what it considers the current state of the interface to be.

The file `/run/network/ifstate` must be writable for `ifup` or `ifdown` to work properly. If that location is not writable (for example, because the root filesystem is mounted read-only for system recovery) then `/run/network/ifstate` should be made a symbolic link to a writable location. If that is not possible then you can use the `--force` option to run configuration or deconfiguration commands without updating the file.

Note that the program does not run automatically: `ifup` alone does not bring up interfaces that appear as a result of hardware being installed and `ifdown` alone does not bring down interfaces that disappear as a result of hardware being removed. To automate the configuration of network interfaces you need to install other packages such as `udev(7)` or `ifplugd(8)`.

## AUTHORS

The `ifupdown` suite was created by Anthony Towns <[aj@azure.humbug.org.au](mailto:aj@azure.humbug.org.au)>, and is currently maintained by Guus Sliepen <[guus@debian.org](mailto:guus@debian.org)>.

Many others have helped develop `ifupdown` over time, see `/usr/share/doc/ifupdown/changelog.Debian.gz` for a full history.

## SEE ALSO

`interfaces(5)`, `ip(8)`, `ifconfig(8)`.

IFUPDOWN

11 Jan 2017

`ifup(8)`