

Table Of Contents

Table Of Contents 1

Defining physical interfaces such as eth0 2

Setup interface to dhcp 2

Examples: How to set up interfaces 2

[Home](#) > [Faq](#) > [Debian / Ubuntu](#)

/etc/network/interfaces Ubuntu Linux networking example

Posted by [Vivek Gite](#) <vivek@nixcraft.com>



Q. Can you explain how to setup network parameters such as IP address, subnet, dhcp etc using /etc/network/interfaces file?

A. /etc/network/interfaces file contains network interface configuration information for the both Ubuntu and Debian Linux. This is where you configure how your system is connected to the network.

[1]

Defining physical interfaces such as eth0

Lines beginning with the word "auto" are used to identify the physical interfaces to be brought up when ifup is run with the -a option. (This option is used by the system boot scripts.) Physical interface names should follow the word "auto" on the same line. There can be multiple "auto" stanzas. ifup brings the named interfaces up in the order listed. For example following example setup eth0 (first network interface card) with 192.168.1.5 IP address and gateway (router) to 192.168.1.254:

```
iface eth0 inet static
address 192.168.1.5
netmask 255.255.255.0
gateway 192.168.1.254
```

Setup interface to dhcp

To setup eth0 to dhcp, enter:

```
auto eth0
iface eth0 inet dhcp
```

Examples: How to set up interfaces

Please read our previous

[How to: Ubuntu Linux convert DHCP network configuration to static IP configuration](#) for more information.

Following is file located at </usr/share/doc/ifupdown/examples/network-interfaces>, use this file as reference (don't forget [interfaces](#) man pages for more help):

```
#####
# /etc/network/interfaces -- configuration file for ifup(8), ifdown(8)
#
# A "#" character in the very first column makes the rest of the line
# be ignored. Blank lines are ignored. Lines may be indented freely.
# A "\" character at the very end of the line indicates the next line
# should be treated as a continuation of the current one.
#
# The "pre-up", "up", "down" and "post-down" options are valid for all
# interfaces, and may be specified multiple times. All other options
# may only be specified once.
#
# See the interfaces(5) manpage for information on what options are
# available.
#####

# We always want the loopback interface.
#
# auto lo
# iface lo inet loopback
```

```
# An example ethernet card setup: (broadcast and gateway are optional)
#
# auto eth0
# iface eth0 inet static
#     address 192.168.0.42
#     network 192.168.0.0
#     netmask 255.255.255.0
#     broadcast 192.168.0.255
#     gateway 192.168.0.1

# A more complicated ethernet setup, with a less common netmask, and a downright
# weird broadcast address: (the "up" lines are executed verbatim when the
# interface is brought up, the "down" lines when it's brought down)
#
# auto eth0
# iface eth0 inet static
#     address 192.168.1.42
#     network 192.168.1.0
#     netmask 255.255.255.128
#     broadcast 192.168.1.0
#     up route add -net 192.168.1.128 netmask 255.255.255.128 gw 192.168.1.2
#     up route add default gw 192.168.1.200
#     down route del default gw 192.168.1.200
#     down route del -net 192.168.1.128 netmask 255.255.255.128 gw 192.168.1.2

# A more complicated ethernet setup with a single ethernet card with
# two interfaces.
# Note: This happens to work since ifconfig handles it that way, not because
# ifup/down handles the ':' any differently.
# Warning: There is a known bug if you do this, since the state will not
# be properly defined if you try to 'ifdown eth0' when both interfaces
# are up. The ifconfig program will not remove eth0 but it will be
# removed from the interfaces state so you will see it up until you execute:
# 'ifdown eth0:1 ; ifup eth0; ifdown eth0'
# BTW, this is "bug" #193679 (it's not really a bug, it's more of a
# limitation)
#
# auto eth0 eth0:1
# iface eth0 inet static
#     address 192.168.0.100
#     network 192.168.0.0
#     netmask 255.255.255.0
#     broadcast 192.168.0.255
#     gateway 192.168.0.1
# iface eth0:1 inet static
#     address 192.168.0.200
#     network 192.168.0.0
#     netmask 255.255.255.0

# "pre-up" and "post-down" commands are also available. In addition, the
# exit status of these commands are checked, and if any fail, configuration
# (or deconfiguration) is aborted. So:
#
# auto eth0
# iface eth0 inet dhcp
#     pre-up [ -f /etc/network/local-network-ok ]
#
# will allow you to only have eth0 brought up when the file
# /etc/network/local-network-ok exists.

# Two ethernet interfaces, one connected to a trusted LAN, the other to
# the untrusted Internet. If their MAC addresses get swapped (because an
# updated kernel uses a different order when probing for network cards,
# say), then they don't get brought up at all.
#
# auto eth0 eth1
# iface eth0 inet static
#     address 192.168.42.1
#     netmask 255.255.255.0
```

```

# pre-up /path/to/check-mac-address.sh eth0 11:22:33:44:55:66
# pre-up /usr/local/sbin/enable-masq
# iface eth1 inet dhcp
# pre-up /path/to/check-mac-address.sh eth1 AA:BB:CC:DD:EE:FF
# pre-up /usr/local/sbin/firewall

# Two ethernet interfaces, one connected to a trusted LAN, the other to
# the untrusted Internet, identified by MAC address rather than interface
# name:
#
# auto eth0 eth1
# mapping eth0 eth1
# script /path/to/get-mac-address.sh
# map 11:22:33:44:55:66 lan
# map AA:BB:CC:DD:EE:FF internet
# iface lan inet static
# address 192.168.42.1
# netmask 255.255.255.0
# pre-up /usr/local/sbin/enable-masq $IFACE
# iface internet inet dhcp
# pre-up /usr/local/sbin/firewall $IFACE

# A PCMCIA interface for a laptop that is used in different locations:
# (note the lack of an "auto" line for any of these)
#
# mapping eth0
# script /path/to/pcmcia-compat.sh
# map home,*,*,* home
# map work,*,*,00:11:22:33:44:55 work-wireless
# map work,*,*,01:12:23:34:45:50 work-static
#
# iface home inet dhcp
# iface work-wireless bootp
# iface work-static static
# address 10.15.43.23
# netmask 255.255.255.0
# gateway 10.15.43.1
#
# Note, this won't work unless you specifically change the file
# /etc/pcmcia/network to look more like:
#
# if [ -r ./shared ] ; then . ./shared ; else . /etc/pcmcia/shared ; fi
# get_info $DEVICE
# case "$ACTION" in
#     'start')
#         /sbin/ifup $DEVICE
#         ;;
#     'stop')
#         /sbin/ifdown $DEVICE
#         ;;
# esac
# exit 0

# An alternate way of doing the same thing: (in this case identifying
# where the laptop is is done by configuring the interface as various
# options, and seeing if a computer that is known to be on each particular
# network will respond to pings. The various numbers here need to be chosen
# with a great deal of care.)
#
# mapping eth0
# script /path/to/ping-places.sh
# map 192.168.42.254/24 192.168.42.1 home
# map 10.15.43.254/24 10.15.43.1 work-wireless
# map 10.15.43.23/24 10.15.43.1 work-static
#
# iface home inet dhcp
# iface work-wireless bootp
# iface work-static static
# address 10.15.43.23
# netmask 255.255.255.0

```

```
# gateway 10.15.43.1
#
# Note that the ping-places script requires the iproute package installed,
# and the same changes to /etc/pcmcia/network are required for this as for
# the previous example.

# Set up an interface to read all the traffic on the network. This
# configuration can be useful to setup Network Intrusion Detection
# sensors in 'stealth'-type configuration. This prevents the NIDS
# system to be a direct target in a hostile network since they have
# no IP address on the network. Notice, however, that there have been
# known bugs over time in sensors part of NIDS (for example see
# DSA-297 related to Snort) and remote buffer overflows might even be
# triggered by network packet processing.
#
# auto eth0
# iface eth0 inet manual
# up ifconfig $IFACE 0.0.0.0 up
# up ip link set $IFACE promisc on
# down ip link set $IFACE promisc off
# down ifconfig $IFACE down

# Set up an interface which will not be allocated an IP address by
# ifupdown but will be configured through external programs. This
# can be useful to setup interfaces configured through other programs,
# like, for example, PPPOE scripts.
#
# auto eth0
# iface eth0 inet manual
# up ifconfig $IFACE 0.0.0.0 up
# up /usr/local/bin/myconfigscript
# down ifconfig $IFACE down
```

4000+ howtos and counting! Want to read more Linux / UNIX howtos, tips and tricks? Subscribe to our [daily email](#) newsletter or [weekly newsletter](#) to make sure you don't miss a single tip/tricks. Alternatively, subscribe via [RSS/XML](#) feed.

Article printed from Frequently Asked Questions About Linux / UNIX: <http://www.cyberciti.biz/faq/>

URL to article: <http://www.cyberciti.biz/faq/setting-up-an-network-interfaces-file/>

URLs in this post:

[1] Image: <http://www.cyberciti.biz/faq/faq/category/ubuntu-linux/>