Linux ifconfig command

On some Unix-like operating systems, ifconfig is used to configure, or view the configuration of, a network interface.

This covers the GNU/Linux version of ifconfig.

Note

On modern Linux systems, the **ip** command has replaced **ifconfig**.

Description

ifconfig stands for "interface configuration." It is used to view and change the configuration of the network interfaces on your system.

Running the **ifconfig** command with no arguments, like this:

```
ifconfig
```

...displays information about all network interfaces currently in operation. The output resembles the following:

```
eth0      Link encap:Ethernet  HWaddr 09:00:12:90:e3:e5

          inet addr:192.168.1.29 Bcast:192.168.1.255  Mask:255.255.255.0

          inet6 addr: fe80::a00:27ff:fe70:e3f5/64 Scope:Link

          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
```

```
          RX packets:54071 errors:1 dropped:0 overruns:0 frame:0

          TX packets:48515 errors:0 dropped:0 overruns:0 carrier:0

          collisions:0 txqueuelen:1000

          RX bytes:22009423 (20.9 MiB)  TX bytes:25690847 (24.5 MiB)

          Interrupt:10 Base address:0xd020

lo        Link encap:Local Loopback

          inet addr:127.0.0.1  Mask:255.0.0.0

          inet6 addr: ::1/128 Scope:Host

          UP LOOPBACK RUNNING  MTU:16436  Metric:1

          RX packets:83 errors:0 dropped:0 overruns:0 frame:0

          TX packets:83 errors:0 dropped:0 overruns:0 carrier:0

          collisions:0 txqueuelen:0

          RX bytes:7766 (7.5 KiB)  TX bytes:7766 (7.5 KiB)

wlan0     Link encap:Ethernet  HWaddr 58:a2:c2:93:27:36

          inet addr:192.168.1.64  Bcast:192.168.2.255  Mask:255.255.255.0
```

```
inet6 addr: fe80::6aa3:c4ff:fe93:4746/64 Scope:Link

UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1

RX packets:436968 errors:0 dropped:0 overruns:0 frame:0

TX packets:364103 errors:0 dropped:0 overruns:0 carrier:0

collisions:0 txqueuelen:1000

RX bytes:115886055 (110.5 MiB)  TX bytes:83286188 (79.4 MiB)
```

Here, **eth0**, **lo** and **wlan0** are the names of the active network interfaces on the system.

- **eth0** is the first Ethernet interface. (Additional Ethernet interfaces would be named **eth1**, **eth2**, etc.) This type of interface is usually a NIC connected to the network by a category 5 cable.
- **lo** is the loopback interface. This is a special network interface that the system uses to communicate with itself.
- **wlan0** is the name of the first wireless network interface on the system. Additional wireless interfaces would be named **wlan1**, **wlan2**, etc.

These are the traditional naming conventions for network interfaces under Linux; other operating systems may have different names. For instance, under many BSD operating systems, Ethernet interfaces are named **em0**, **em1**, etc. Check your configuration, or consult your documentation, to determine the exact names of your interfaces.

Viewing the configuration of all interfaces

If you'd like to view the configuration of all network interfaces on the system (not just the ones that are currently active), you can specify the **-a** option, like this:

```
ifconfig -a
```

This produces output similar to running **ifconfig**, but if there are any inactive interfaces on the system, their configuration is also shown.

Viewing the configuration of a specific interface

To view the configuration of a specific interface, specify its name as an option. For instance,

```
ifconfig eth0
```

...displays the configuration of device **eth0** only.

Enabling and disabling an interface

When a network interface is active, it can send and receive data; when it is inactive, it is not able to transmit or receive. You can use **ifconfig** to change the status of a network interface from inactive to active, or vice versa.

To enable an inactive interface, provide **ifconfig** with the interface name followed by the keyword **up**.

Enabling or disabling a device requires superuser permissions, so you either have to be logged in as root, or prefix your command with **sudo** to run it with superuser privileges.

For instance, if network interface **eth1** is inactive, you can activate it with the command:

```
sudo ifconfig eth1 up
```

Similarly, you can disable an active network interface using the **down** keyword. For instance, to disable the wireless network interface **wlan0**, use the command:

```
sudo ifconfig wlan0 down
```

Configuring an interface

ifconfig can be used at the command line to configure (or re-configure) a network interface. This is often unnecessary since this configuration is often handled by a script when you boot the system. If you'd like to do so manually, you need superuser privileges, so we'll use **sudo** again when running these commands.

To assign a static IP address to an interface, specify the interface name and the IP address. For example, to assign the IP address **69.72.169.1** to the interface **wlan0**, use the command:

```
sudo ifconfig wlan0 69.72.169.1
```

To assign a network mask to an interface, use the keyword **netmask** and the netmask address. For instance, to configure the interface **eth1** to use a network mask of **255.255.255.0**, the command would be:

```
sudo ifconfig eth1 netmask 255.255.255.0
```

To assign a broadcast address to an interface, use the keyword **broadcast** and the broadcast address. For instance, to

configure the interface **wlan1** to use a broadcast address of **172.16.25.98**, the command would be:

```
sudo ifconfig wlan1 broadcast 172.16.25.98
```

These configurations can combined in a single command. For instance, to configure interface **eth0** to use the static IP address **192.168.2.5**, the network mask **255.255.255.0**, and the broadcast address **192.168.2.7**, the command would be:

```
sudo ifconfig eth0 192.168.2.5 netmask 255.255.255.0
broadcast 192.168.2.7
```

These are the most commonly-used configuration options for **ifconfig**. A complete list is provided below.

What about DHCP?

ifconfig can only assign a static IP address to a network interface. If you want to assign a dynamic IP address using DHCP, use the **dhclient** command.

Syntax

```
ifconfig [-v] [-a] [-s] [interface]

ifconfig [-v] interface [aftype] options | address ...
```

Technical description

ifconfig is used to configure the system's kernel-resident network interfaces. It is used at boot time to set up interfaces as necessary.

After that, it is usually only needed when debugging, or when system tuning is needed.

If no arguments are given, **ifconfig** displays the status of the system's active interfaces.

If a single *interface* argument is given, it displays the status of the given interface only.

If a single **-a** argument is given, it displays the status of all interfaces, even those that are "down" (inactive).

In all other cases, **ifconfig** configures an interface according to the options provided.

About address families

If the first argument after the interface name is recognized as the name of a supported address family, that address family is used for decoding and displaying all protocol addresses. Currently supported address families include, **inet** (TCP/IP; this is the default), **inet6** (IPv6), **ax25** (AMPR Packet Radio), **ddp** (Appletalk Phase 2), **ipx** (Novell IPX) and **netrom** (AMPR Packet Radio).

Options

-a	**Display information for all network interfaces, even if they are down.**
-s	Display a short list in a format identical to the command "**netstat** **-i**".
-v	Verbose mode; display additional information for certain error conditions.

interface	The name of the interface. This is usually a driver name followed by a unit number, for example "**eth0**" for the first Ethernet interface. If your kernel supports alias interfaces, you can specify them with **eth0:0** for the first alias of **eth0**. You can use them to assign a second address. To delete an alias interface, use **ifconfig eth0:0 down**. Note: for every scope (i.e., same net with address/netmask combination) all aliases are deleted, if you delete the first (primary).
up	This flag causes the interface to be activated. It is implicitly specified if an address is assigned To the interface.
down	This flag causes the driver for this interface to be shut down.
[-]arp	Enable (or disable, if the "**-**" prefix is specified) the use of the ARP protocol on this interface.
[-]promisc	Enable (or disable, if the "**-**" prefix is specified) the promiscuous mode of the interface. If promiscuous mode is enabled, all packets on the network will be received by the interface.
[-]allmulti	Enable or disable all-multicast mode. If multicast mode is enabled, all multicast packets on the network will be received by the interface.
metric *N*	This parameter sets the interface metric, which is used by the interface to make routing decisions. *N* must be an integer between **0** and **4294967295**. If you're not sure what a network metric is, or whether to change it, you can safely leave this setting alone.
mtu *N*	This parameter sets the MTU (maximum transfer unit) of an interface. This setting is used to limit the maximum packet size transferred by the interface. If you're not sure about it, you can safely leave this setting alone.
dstaddr *address*	Set the remote IP address for a point-to-point link (such as PPP). This keyword is now obsolete;use the **pointopoint** keyword instead.
netmask *address*	Set the IP network mask for this interface. This value defaults to the usual class A, B or C network mask (as derived from the interface IP address), but it can be set to any value.
add *address/prefixlen*	Add an IPv6 address to an interface.
del *address/prefixlen*	Remove an IPv6 address from an interface.
tunnel *aa.bb.cc.dd*	Create a new SIT (IPv6-in-IPv4) device, tunnelling to the given destination.
irq *address*	Set the interrupt line used by this device. Not all devices can dynamically change their IRQ setting.
io_addr *address*	Set the start address in I/O space for this device.
mem_start *address*	Set the start address for shared memory used by this device. Only a few devices need this.
media *type*	Set the physical port or medium type to be used by the device. Not all devices can change this setting, and those that can vary in what values they support. Typical values for *type* are **10base2** (thin Ethernet), **10baseT** (twisted-pair 10 Mbps Ethernet), AUI (external transceiver) , etc. The special medium type of **auto** tells the driver to auto-sense the media. Again,

	not all drivers can do this.
[-]broadcast [*address*]	If the address argument is given, this sets the protocol broadcast address for this interface. Otherwise, it sets (or clear, if the "**-**" prefix is used) the **IFF_BROADCAST** flag for the interface.
[-]pointopoint [*address*]	This keyword enables the **point-to-point** mode of an interface, meaning that it is a direct link between two machines with nobody else listening on it. If the address argument is also given, set the protocol address of the other side of the link, just like the obsolete dstaddr keyword does. Otherwise, set or clear the **IFF_POINTOPOINT** flag for the interface.
hw class *address*	Set the hardware address of this interface, if the device driver supports this operation. The keyword must be followed by the name of the hardware class and the printable ASCII equivalent of the hardware address. Hardware classes currently supported include **ether** (Ethernet), **ax25** (AMPR AX.25), **ARCnet** and **netrom** (AMPR NET/ROM).
multicast	Set the multicast flag on the interface. This should not normally be needed as the drivers set the flag correctly themselves.
address	The IP address to be assigned to this interface.
txqueuelen *length*	Set the length of the transmit queue of the device. It is useful to set this to small values for slower devices with a high latency (such as a connection over a modem, or over ISDN) to prevent fast bulk transfers from disturbing interactive traffic like telnet too much.

Examples

```
ifconfig
```

Running **ifconfig** with no options displays the configuration of all active interfaces.

```
ifconfig -a
```

Displays the configuration of all interfaces, both active and inactive.

```
ifconfig eth0
```

View the network settings on the interface **eth0**, which (under Linux) is the first Ethernet adapter installed in the system.

```
ifconfig eth1 up
```

Activate the network interface **eth1**.

```
ifconfig wlan0 down
```

Deactivate the network interface **wlan0**.

```
ifconfig wlan1 122.140.201.66
```

Configure the network interface **wlan1** to use the static IP address **122.140.201.66**.

```
ifconfig wlan0 netmask 255.255.255.0
```

Configure the network interface **wlan0** to use the network mask **255.255.255.0**.

```
ifconfig eth0 192.168.1.102 netmask 255.255.255.0
broadcast 192.168.1.255
```

Configure **eth0** to use the static IP address **192.168.1.102** using the network mask **255.255.255.0**, and the broadcast address **192.168.1.255**.