

## Objective 2    **Set up Network Interfaces with the ip Tool**

You normally configure a network card with YaST during or after installation. You can use the tool **ip** to change the network interface configuration quickly from the command line.

Changing the network interface configuration at the command line is especially useful for testing purposes; but if you want a configuration to be permanent, you must save it in a configuration file. These configuration files are generated automatically when you set up a network card with YaST.

You can use `ip` to perform the following tasks:

- [Display the Current Network Configuration](#)
- [Change the Current Network Configuration](#)



You can enter `/sbin/ip` as a normal user to display the current network setup only. To change the network setup, you have to be logged in as root.

---

As changes made with `ip` are lost with the next reboot, you also have to know how to:

- [Save Device Settings to a Configuration File](#)

### ***Display the Current Network Configuration***

With the `ip` tool, you can display the following information:

- [IP Address Setup](#)
- [Device Attributes](#)
- [Device Statistics](#)

## IP Address Setup

To display the IP address setup of all interfaces, enter **ip address show**. Depending on your network setup, you see information similar to the following:

```
da2:~ # ip address show
1: lo: <LOOPBACK,UP> mtu 16436 qdisc noqueue
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 brd 127.255.255.255 scope host lo
   inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,NOTRAILERS,UP> mtu 1500 qdisc
   pfifo_fast qlen 1000
   link/ether 00:30:05:4b:98:85 brd ff:ff:ff:ff:ff:ff
   inet 10.0.0.2/24 brd 10.0.0.255 scope global eth0
   inet6 fe80::230:5ff:fe4b:9885/64 scope link
       valid_lft forever preferred_lft forever
3: sit0: <NOARP> mtu 1480 qdisc noqueue
   link/sit 0.0.0.0 brd 0.0.0.0
```

The information is grouped by network interfaces. Every interface entry starts with a digit, called the *interface index*; the interface name is displayed after the interface index.

In the above example, there are 3 interfaces:

- **lo**. The loopback device, which is available on every Linux system, even when no network adapter is installed. (As stated above, “device” and “interface” are often used synonymously in the context of network configuration.) Using this virtual device, applications on the same machine can use the network to communicate with each other.

For example, you can use the IP address of the loopback device to access a locally installed web server by typing **http://127.0.0.1** in the address bar of your web browser.

- **eth0**. The first Ethernet adapter of the computer in this example. Ethernet devices are normally called eth0, eth1, eth2, and so on.

- **sit0**. A special virtual device which can be used to encapsulate IPv4 packets into IPv6 packets. It's not used in a normal IPv4 network.

You always have the entries for the loopback and sit devices. Depending on your hardware setup, you might have more Ethernet devices in the ip output.

Several lines of information are displayed for every network interface, such as eth0 in the preceding example:

```
2: eth0: <BROADCAST,MULTICAST,NOTRAILERS,UP> mtu 1500 qdisc
pfifo_fast qlen 1000
```

The most important information of the line in this example is the interface index (**2**) and the interface name (**eth0**).

The other information shows additional attributes set for this device, such as the hardware address of the Ethernet adapter (00:30:05:4b:98:85):

```
link/ether 00:30:05:4b:98:85 brd ff:ff:ff:ff:ff:ff
```

In the following line, the IPv4 setup of the device is displayed:

```
inet 10.0.0.2/24 brd 10.0.0.255 scope global eth0
```

The IP address (**10.0.0.2**) follows inet, and the broadcast address (**10.0.0.255**) follows brd. The length of the network mask is displayed after the IP address and separated from it by a /. The length is displayed in bits (**24**).

The following lines show the IPv6 configuration of the device:

```
inet6 fe80::230:5ff:fe4b:9885/64 scope link
valid_lft forever preferred_lft forever
```

The address shown here is automatically assigned, even though IPv6 is not used in the network that is connected with the device. The address is generated from the hardware address of the device.

Depending on the device type, the information can differ. However, the most important information (such as assigned IP addresses) is always shown.

### Device Attributes

If you are only interested in the device attributes and not in the IP address setup, you can enter **ip link show**:

```
da2:~ # ip link show
1: lo: <LOOPBACK,UP> mtu 16436 qdisc noqueue
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: eth0: <BROADCAST,MULTICAST,NOTRAILERS,UP> mtu 1500 qdisc
   pfifo_fast qlen 1000
   link/ether 00:30:05:4b:98:85 brd ff:ff:ff:ff:ff:ff
3: sit0: <NOARP> mtu 1480 qdisc noqueue
   link/sit 0.0.0.0 brd 0.0.0.0
```

The information is similar to what you have seen when entering **ip address show**, but the information about the address setup is missing. The device attributes are displayed in brackets right after the device name.

The following is a list of possible attributes and their meanings:

- **UP**. The device is turned on. It is ready to transmit packets to and receive packets from the network.
- **LOOPBACK**. The device is a loopback device.
- **BROADCAST**. The device can send packets to all hosts sharing the same network.
- **POINTOPOINT**. The device is only connected to one other device. All packets are sent to and received from the other device.

- **MULTICAST.** The device can send packets to a group of other systems at the same time.
- **PROMISC.** The device listens to all packets on the network, not only to those sent to the device's hardware address. This is usually used for network monitoring.

### Device Statistics

You can use the **-s** option with the `ip` command to display additional statistics information about the devices. The command looks like the following:

**ip -s link show eth0**

By giving the device name at the end of the command line, the output is limited to one specific device. This can also be used to display the address setup or the device attributes.

The following is an example of the information displayed for the device `eth0`:

```
da2:~ # ip -s link show eth0
2: eth0: <BROADCAST,MULTICAST,NOTRAILERS,UP> mtu 1500 qdisc
pfifo_fast qlen 1000
    link/ether 00:30:05:4b:98:85 brd ff:ff:ff:ff:ff:ff
    RX: bytes  packets  errors  dropped  overrun  mcast
      849172787 9304150  0      0        0        0
    TX: bytes  packets  errors  dropped  carrier  collsns
      875278145 1125639  0      0        0        0
```

Two additional sections with information are displayed for every device. Each of the sections has a headline with a description of the information displayed.

The section starting with `RX` displays information about received packets, and the section starting with `TX` displays information about sent packets.

The sections display the following information:

- **Bytes.** The total number of bytes received or transmitted by the device.
- **Packets.** The total number of packets received or transmitted by the device.
- **Errors.** The total number of receiver or transmitter errors.
- **Dropped.** The total number of packets dropped due to a lack of resources.
- **Overrun.** The total number of receiver overruns resulting in dropped packets.

As a rule, if a device is overrun, it means that there are serious problems in the Linux kernel or that your computer is too slow for the device.

- **Mcast.** The total number of received multicast packets. This option is supported by only a few devices.
- **Carrier.** The total number of link media failures, because of a lost carrier.
- **Collsns.** The total number of collision events on Ethernet-like media.
- **Compressed.** The total number of compressed packets.

## Change the Current Network Configuration

You can also use the `ip` tool to change the network configuration by performing the following tasks:

- [Assign an IP Address to a Device](#)
- [Delete the IP Address from a Device](#)
- [Change Device Attributes](#)

### Assign an IP Address to a Device

To assign an address to a device, use a command similar to the following:

```
da2:~ # ip address add 10.0.0.2/24 brd + dev eth0
```

In this example, the command assigns the IP address **10.0.0.2** to the device **eth0**. The network mask is **24** bits long, as determined by the **/24** after the IP address. The **brd +** option sets the broadcast address automatically as determined by the network mask.

You can enter **ip address show dev eth0** to verify the assigned IP address. The assigned IP address is displayed in the output of the command line.

You can assign more than one IP address to a device.

### Delete the IP Address from a Device

To delete the IP address from a device, use a command similar to the following:

```
da2:~ # ip address del 10.0.0.2 dev eth0
```

In this example, the command deletes the IP address **10.0.0.2** from the device **eth0**.

Use **ip address show eth0** to verify that the address was deleted.

### Change Device Attributes

You can also change device attributes with the **ip** tool. The following is the basic command to set device attributes:

#### **ip link set device attribute**

The possible attributes are described in “[Device Attributes](#)” on 4-6. The most important attributes are *up* and *down*. By setting these attributes, you can enable or disable a network device.

To enable a network device (such as eth0), enter the following command:

```
da2:~ # ip link set eth0 up
```

To disable a network device (such as eth0), enter the following command:

```
da2:~ # ip link set eth0 down
```

### **Save Device Settings to a Configuration File**

All device configuration changes you make with `ip` are lost when the system is rebooted. To restore the device configuration automatically when the system is started, the settings need to be saved in configuration files.

The configuration files for network devices are located in the **`/etc/sysconfig/network/`** directory.

If the network devices are set up with YaST, one configuration file is created for every device.

For Ethernet devices, the filenames consist of `ifcfg-eth-id-` and the hardware address of the device. For a device with the hardware address `00:30:05:4b:98:85`, the filename would be `ifcfg-eth-id-00:30:05:4b:98:85`.

We recommend that you set up a device with YaST first and make changes in the configuration file. Setting up a device from scratch is a complex task, because the hardware driver also needs to be configured manually.

If you have more than one network adapter in your system, it might be difficult to find the corresponding configuration file for a device.

You can use the **`ip link show`** command to display the hardware address for each Ethernet device. Because the hardware address is part of the file name, you can identify the right configuration file.

The content of the configuration files depends on the configuration of the device. To change the configuration file, you need to know how to do the following:

- [Configure a Device Statically](#)
- [Configure a Device Dynamically with DHCP](#)
- [Start and Stop Configured Interfaces](#)

### **Configure a Device Statically**

The content of a configuration file of a statically configured device is similar to the following:

```
BOOTPROTO='static'  
BROADCAST=' '  
ETHTOOL_OPTIONS=' '  
IPADDR='10.0.0.2'  
MTU=' '  
NAME='Digital DECchip 21142/43'  
NETMASK='255.255.255.0'  
NETWORK=' '  
REMOTE_IPADDR=' '  
STARTMODE='auto'  
UNIQUE='rBUF.+xOL8ZCSAQC'  
USERCONTROL='no'  
_nm_name='bus-pci-0000:00:0b.0'  
ETHTOOL_OPTIONS=' '
```

The configuration file includes several lines. Each line has an option and a value assigned to that option, as explained below:

- **BOOTPROTO='static'**

The **BOOTPROTO** option determines the way the device is configured. There are 2 possible values:

- **static.** The device is configured with a static IP address.
- **dhcp.** The device is configured automatically with a DHCP server.

- **REMOTE\_IPADDR=''**

You need to set the value for the **REMOTE\_IPADDR** option only if you are setting up a point-to-point connection.

---

- **STARTMODE='onboot'**

The STARTMODE option determines how the device is started. The option can include the following values:

- **auto.** The device is started at boot time or when initialized at runtime.
- **manual.** The device must be started manually with ifup.
- **ifplugd.** The interface is controlled by ifplugd. If you want to use interfaces mutually exclusive, also set IFPLUGD\_PRIORITY

- **UNIQUE='rBUF.+xOL8ZCSAQC'  
\_nm\_name='bus-pci-0000:00:0b.0'**

These 2 lines contain options added by YaST when the device is configured. They don't affect the network configuration itself.

- **BROADCAST=""  
IPADDR='10.0.0.2'  
NETMASK='255.255.255.0'  
NETWORK=""**

These 4 lines contain the options for the network address configuration. The options have the following meanings:

- **BROADCAST.** The broadcast address of the network. If empty, the broadcast address is derived from the IP address and the netmask, according to the configuration in **/etc/sysconfig/network/config.**
- **IPADDR.** The IP address of the device.
- **NETMASK.** The network mask.
- **NETWORK.** The address of the network itself.

- **MTU=""**

You can use the MTU option to specify a value for the MTU (Maximum Transmission Unit). If you don't specify a value, the default value is used. For an Ethernet device, the default value is 1500 bytes.

- **ETHCTOOL\_OPTIONS=**'

**ethctool** is used for querying settings of an Ethernet device and changing them, for instance setting the speed or half/full duplex mode. The manual page for ethctool lists the available options.

If you want ethctool to modify any settings, list the options here; if no options are listed, ethctool is not called.

The **/etc/sysconfig/network/ifcfg.template** file contains a template that you can use as a base for device configuration files. It also has comments explaining the various options.

### **Configure a Device Dynamically with DHCP**

If you want to configure a device by using a DHCP server, you set the BOOTPROTO option to **dhcp** as shown in the following:

**BOOTPROTO='dhcp'**

When the device is configured by DHCP, you don't need to set any options for the network address configuration in the file. If there are any settings, they are overwritten by the settings of the DHCP server.

### **Start and Stop Configured Interfaces**

To apply changes to a configuration file, you need to stop and restart the corresponding interface. You can do this with the ifdown and ifup command.

For example, entering **ifdown eth0** disables the device eth0. **ifup eth0** enables eth0 again.

When the device is restarted, the new configuration is read from the configuration file.



---

Configuring the interfaces with IP addresses, routes, etc. with the `ip` tool requires an existing device setup, including a correctly loaded kernel module. This is usually done at boot time by `/sbin/hwup`, using the configuration contained in files in the `/etc/sysconfig/hardware/` directory. Information is available in the manual page for `hwup`.

---



---

Under certain circumstances physical network devices can change the interface name, for instance the interface that used to be called `eth0` now becomes `eth1` and vice versa. Sometimes this happens from one boot to the next, even without any physical changes on the hardware. Information on how to achieve persistent interface names is contained in the `/usr/share/doc/packages/sysconfig/README.Persistent_Interface_Names` file.

---