TELNET :

TELNET (**TEL**ecommunication **NET**work) is a network protocol used on the Internet or local area network (LAN) connections. It was developed in 1969 beginning with RFC 15 and standardized as IETF STD 8, one of the first Internet standards.

It is a network protocol used on the Internet or local area networks to provide a bidirectional interactive communications facility. Typically, telnet provides access to a command-line interface on a remote host via a virtual terminal connection which consists of an 8-bit byte oriented data connection over the Transmission Control Protocol (TCP). User data is interspersed in-band with TELNET control information. The user's computer, which initiates the connection, is referred to as the local computer.

The computer being connected to, which accepts the connection, is referred to as the remote computer. The remote computer can be physically located in the next room, the next town or in another country.

The network terminal protocol (TELNET) allows a user to log in on any other computer on the network. We can start a remote session by specifying a computer to connect to. From that time until we finish the session, anything we type is sent to the other computer.

The Telnet program runs on the computer and connects your PC to a server on the network. We can then enter commands through the Telnet program and they will be executed as if we were entering them directly on the server console. This enables we to control the server and communicate with other servers on the network. To start a Telnet session, we must log in to a server by entering a valid username and password. Telnet is a common way to remotely control Web servers.

The term telnet also refers to software which implements the client part of the protocol. TELNET clients have been available on most Unix systems for many years and are available virtually for all platforms. Most network equipment and OSs with a TCP/IP stack support some kind of TELNET service server for their remote configuration including ones based on Windows NT. TELNET is a client server protocol, based on a reliable connection oriented transport. Typically this protocol used to establish a connection to TCP port 23, where a getty-equivalent program (telnetd) is listening, although TELNET predates.

IPCONFIG

Ipconfig is a computer network tool for windows (ifconfig is equivalent for unix) that displays all current tcp/ip network configuration values and refreshes dynamic host configuration protocol

dhcp and domain name system dns setting. Used without parameters, **ipconfig** displays the IP address, subnet mask, and default gateway for all adapters.

Syntax

ipconfig [**/all**] [**/renew** [*Adapter*]] [**/release** [*Adapter*]] [**/flushdns**] [**/displaydns**] [**/registerdns**] [**/showclassid** *Adapter*] [**/setclassid** *Adapter* [*Class*

Parameters

/all : Displays the full TCP/IP configuration for all adapters. Without this parameter, **ipconfig** displays only the IP address, subnet mask, and default gateway values for each adapter. Adapters can represent physical interfaces, such as installed network adapters, or logical interfaces, such as dial-up connections.

/renew [*Adapter* **] :** Renews DHCP configuration for all adapters (if an adapter is not specified) or for a specific adapter if the *Adapter* parameter is included. This parameter is available only on computers with adapters that are configured to obtain an IP address automatically. To specify an adapter name, type the adapter name that appears when you use **ipconfig** without parameters.

/release [*Adapter* **] :** Sends a DHCPRELEASE message to the DHCP server to release the current DHCP configuration and discard the IP address configuration for either all adapters (if an adapter is not specified) or for a specific adapter if the *Adapter* parameter is included. This parameter disables TCP/IP for adapters configured to obtain an IP address automatically. To specify an adapter name, type the adapter name that appears when you use **ipconfig** without parameters.

/flushdns : Flushes and resets the contents of the DNS client resolver cache. During DNS troubleshooting, you can use this procedure to discard negative cache entries from the cache, as well as any other entries that have been added dynamically.

/displaydns : Displays the contents of the DNS client resolver cache, which includes both entries preloaded from the local Hosts file and any recently obtained resource records for name queries resolved by the computer. The DNS Client service uses this information to resolve frequently queried names quickly, before querying its configured DNS servers.

/registerdns : Initiates manual dynamic registration for the DNS names and IP addresses that are configured at a computer. You can use this parameter to troubleshoot a failed DNS name registration or resolve a dynamic update problem between a client and the DNS server without rebooting the client computer. The DNS settings in the advanced properties of the TCP/IP protocol determine which names are registered in DNS.

/showclassid *Adapter* **:** Displays the DHCP class ID for a specified adapter. To see the DHCP class ID for all adapters, use the asterisk (*) wildcard character in place of *Adapter*. This parameter is available only on computers with adapters that are configured to obtain an IP address automatically.

/setclassid *Adapter* **[** *ClassID* **] :** Configures the DHCP class ID for a specified adapter. To set the DHCP class ID for all adapters, use the asterisk (*) wildcard character in place of *Adapter*. This parameter is available only on computers with adapters that are configured to obtain an IP address automatically. If a DHCP class ID is not specified, the current class ID is removed.

/? : Displays help at the command prompt.

IfCONFIG

Ifconfig is used to configure the 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 currently 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. Otherwise, it configures an interface.

Viewing The Configuration Of All Interfaces

    ifconfig -a

Viewing The Configuration Of A Specific Interface

    ifconfig eth0

Enabling And Disabling An Interface

    sudo ifconfig eth1 up

    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 typically handled by a script when you boot the system. If you'd like to do so manually, you will 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 eth0 192.168.2.5 netmask 255.255.255.0 broadcast 192.168.2.7

Options

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.

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 the use of the ARP protocol on this interface.

[-]promisc

Enable or disable the **promiscuous** mode of the interface. If selected, all packets on the network will be received by the interface.

[-]allmulti

Enable or disable **all-multicast** mode. If selected, all multicast packets on the network will be received by the interface.

metric N

This parameter sets the interface metric. It is not available under GNU/Linux.

mtu N

This parameter sets the Maximum Transfer Unit (MTU) of an interface.

dstaddr addr

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 addr

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 addr/prefixlen

Add an IPv6 address to an interface.

del addr/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 addr

Set the interrupt line used by this device. Not all devices can dynamically change their IRQ setting.

io_addr addr

Set the start address in I/O space for this device.

mem_start addr

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 10Mbps Ethernet), **AUI** (external transceiver) and so on. The special medium type of **auto** can be used to tell the driver to auto-sense the media. Again, not all drivers can do this.

[-]broadcast [addr]

If the address argument is given, set the protocol broadcast address for this interface. Otherwise, set (or clear) the**IFF_BROADCAST** flag for the interface.

[-]pointopoint [addr]

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 (modem links, ISDN) to prevent fast bulk transfers from disturbing interactive traffic like telnet too much.

The netstat Command

The netstat command displays the protocol statistics and current TCP/IP connections on the local system. Used without any switches, the netstatcommand shows the active connections for all outbound TCP/IP connections. In addition, several switches are available that change the type of information netstat displays. Table 5 shows the various switches available for the netstat utility.

Table 5 netstat Switches

Switch	Description
-a	Displays the current connections and listening ports
-e	Displays Ethernet statistics
-n	Lists addresses and port numbers in numerical form
-p	Shows connections for the specified protocol
-r	Shows the routing table
-s	Lists per-protocol statistics
interval	Specifies the length of time to wait before redisplaying statistics

The netstat utility is used to show the port activity for both TCP and UDP connections, showing the inbound and outbound connections. When used without switches, the netstat utility has four information headings.
Proto Lists the protocol being used, either UDP or TCP.
Local address Specifies the local address and port being used.
Foreign address Identifies the destination address and the port being used.
State Specifies whether the connection is established.
In its default usage, the netstat command shows outbound connections that have been established by TCP. The following shows a sample output from a netstat command without using any switches:

```
C:\>netstat
Active Connections
  Proto  Local Address          Foreign Address        State
  TCP    laptop:2848            MEDIASERVICES1:1755    ESTABLISHED
  TCP    laptop:1833            www.test.com:80  ESTABLISHED
  TCP    laptop:2858            194.70.58.241:80       ESTABLISHED
  TCP    laptop:2860            194.70.58.241:80       ESTABLISHED
  TCP    laptop:2354            www.test.com:80  ESTABLISHED
  TCP    laptop:2361            www.test.com:80  ESTABLISHED
  TCP    laptop:1114            www.test.com:80  ESTABLISHED
  TCP    laptop:1959            www.test.com:80  ESTABLISHED
  TCP    laptop:1960            www.test.com:80  ESTABLISHED
  TCP    laptop:1963            www.test.com:80  ESTABLISHED
  TCP    laptop:2870            localhost:8431         TIME_WAIT
  TCP    laptop:8431            localhost:2862         TIME_WAIT
  TCP    laptop:8431            localhost:2863         TIME_WAIT
  TCP    laptop:8431            localhost:2867         TIME_WAIT
  TCP    laptop:8431            localhost:2872         TIME_WAIT
```

Like any other command-line utility, they are often used with switches. The following sections provide a brief explanation of the switches and a sample output from each.