

CMD (Command Prompt)

- **Full form:** Command Prompt
- **Introduced by:** Microsoft in the 1980s (MS-DOS based)
- **Purpose:** Used for simple system tasks and file operations.
- **Language:** Uses simple command-line syntax (not a full programming language).
- **Examples:**
  - `dir` → List files
  - `cd` → Change directory
  - `copy` → Copy files
  - `del` → Delete files

Limitations:

- Can’t automate complex tasks.
- No object handling, only text-based output.

PowerShell

- **Introduced by:** Microsoft in 2006
- **Purpose:** Used for **advanced automation and system administration**.
- **Language:** Full **scripting language** built on **.NET Framework**.
- **Examples:**
  - `Get-ChildItem` → List files (like `dir`)
  - `Set-Location` → Change directory (like `cd`)
  - `Get-Process` → Show running processes
  - `Stop-Process -Name notepad` → Stop a process

Advantages:

- Can automate Windows tasks and servers.
- Supports **loops, conditions, functions, and variables**.
- Handles **objects**, not just text.
- Used for **DevOps and cloud automation**.

Difference Table

Feature	CMD	PowerShell
Type	Command-line interpreter	Automation & scripting tool
Output	Text	Objects
Complexity	Simple	Advanced
Use case	Basic system tasks	Automation, scripting, server management
Example command	<code>dir</code>	<code>Get-ChildItem</code>
Based on	MS-DOS	.NET Framework

Basic CMD (Command Prompt) Commands

Command	Purpose
dir	Shows all files and folders in the current directory.
cd foldername	Changes the directory (open a folder).
cd ..	Moves one step back to the previous folder.
cls	Clears the screen.
mkdir foldername	Creates a new folder.
rmdir foldername	Deletes a folder (empty only).
del filename	Deletes a file.
copy file1 file2	Copies one file to another name/location.
move file1 foldername	Moves a file to a different folder.
exit	Closes the Command Prompt.

## Command Notes For CMD

### ATTRIB Command

Used to **change file or folder attributes** like *Hidden (H)*, *Read-only (R)*, and *System (S)*.

Command	Description
ATTRIB +H +R +S	Makes the file <b>Hidden, Read-only, and System protected</b> . <i>(Used for hiding important files)</i>
ATTRIB -H -R -S	Removes the Hidden, Read-only, and System attributes. <i>(Used to unhide and make file editable again)</i>

### ASSOC.EXE

Shows or changes **file extension associations** (which program opens a file type).

Example:

ASSOC .txt → shows which program opens .txt files (e.g., Notepad).

### SYSTEMINFO

Displays **complete system configuration details**, including OS version, RAM, BIOS, network info, and updates.

Example use: systeminfo

### MSCONFIG

- Opens the **System Configuration Utility**, used to:
  - Manage startup programs
  - Change boot options
  - Control system services

Example use: msconfig

# GETMAC

Displays the **MAC address** (physical address) of your network adapter.  
Example use: `getmac`

# CHKDSK

- Checks the **file system and disk errors** on your hard drive.
  - Can fix issues if used with parameters (e.g., `/f` for fix).
- Example use: `chkdsk`

# VOL D:

- Displays the **volume label** and **serial number** of the D: drive.
- Example use: `vol D:`

# WINVER

- Opens a small window showing **Windows version** and **build number**.
- Example use: `winver`

# MSINFO32

Displays the **System Information Tool** showing details like hardware resources, components, and software environment.  
Example use: `msinfo32`

# DISKPART

Used for **managing hard drives and partitions** via command line.

Example steps:

```
diskpart
list disk          ← Shows all disks
select disk 0      ← Selects the first disk
detail disk        ← Shows full details of the selected disk
```

# Summary Table

Command	Purpose
ATTRIB	Change file attributes
CIPHER	Encrypt / Decrypt files
ASSOC.EXE	Check file extension associations
SYSTEMINFO	Show full system details
MSCONFIG	Configure startup and services
GETMAC	Show MAC address

Command	Purpose
CHKDSK	Check and repair disk
VOL D:	Show volume info
WINVER	Show Windows version
MSINFO32	Detailed system info
DISKPART	Manage disks and partitions

## Windows Command (Task & WMIC Commands)

### TASKLIST

**Purpose:** Displays a list of all running tasks (programs and background processes).

**Use:**

```
tasklist
```

Shows details like:

- Image name (process name)
- Process ID (PID)
- Memory usage

*Note: Used to monitor running applications and services.*

### TASKKILL

**Purpose:** Terminates or stops a running process using its name or PID.

**Use:**

```
taskkill /im notepad.exe /f
```

→ /im = image name, /f = force close

Example:

```
taskkill /pid 1234 /f → Kills process with ID 1234.
```

### NET START

**Purpose:** Displays or starts Windows **services**.

**Use:**

```
net start
```

→ Lists all running services.

Or

```
net start <service-name>
```

→ Starts a specific service.

## NET STOP

**Purpose:** Stops a running Windows service using its **name or PID**.

**Use:**

```
net stop <service-name>
```

Example: `net stop spooler` → Stops the printer spooler service.

## 5. DRIVERQUERY

**Purpose:** Shows a list of all **installed device drivers** in the system.

**Use:**

```
driverquery
```

Optional: `driverquery /v` → Shows detailed driver info (version, date, etc.)

## WMIC (Windows Management Instrumentation Command-line)

`wmic` allows users to **get system information** through command line.

### **wmic cpu**

**Purpose:** Shows details of the CPU (processor).

Example Output: Name, number of cores, clock speed, etc.

### **wmic product get name, version**

Lists all **installed software** with their **version numbers**.

### **wmic csproduct get version**

Displays **computer product version** (useful for identifying device model or build).

### **wmic computersystem get name, systemtype**

Shows the **computer name** and **system architecture** (e.g., x64-based PC).

### **wmic computersystem get totalphysicalmemory**

Displays **total installed RAM** (in bytes).

### **wmic csproduct get identifyingnumber**

Shows **system serial number or ID** (useful for warranty or inventory).

### **wmic partition get name, size, type**

Lists all **disk partitions** with their **name, size, and type**.

### **wmic computersystem get manufacturer**

Displays **manufacturer name** (e.g., Dell, HP, Lenovo).

### **wmic computersystem get model**

Shows the **system model number** (e.g., HP Laptop 15-bs145tu).

### **wmic nic get macaddress, description**

Displays all **network interface cards (NICs)** with their **MAC addresses** and **descriptions**.

### **wmic process where name="app.exe" call terminate**

Terminates a specific running process by name using WMIC.

Example:

```
wmic process where name="chrome.exe" call terminate
```

→ Closes Google Chrome completely.

## **Summary Table**

<b>Command</b>	<b>Description</b>
TASKLIST	Show list of running tasks/processes
TASKKILL	Kill a process by name or PID
NET START	List or start Windows services
NET STOP	Stop running services
DRIVERQUERY	List installed device drivers
WMIC CPU	Show processor information
WMIC PRODUCT GET NAME, VERSION	Show installed software and versions
WMIC CSPRODUCT GET VERSION	Show computer product version
WMIC COMPUTERSYSTEM GET NAME, SYSTEMTYPE	Show PC name and architecture
WMIC COMPUTERSYSTEM GET TOTALPHYSICALMEMORY	Show total RAM
WMIC CSPRODUCT GET IDENTIFYINGNUMBER	Show system serial number
WMIC PARTITION GET NAME, SIZE, TYPE	Show partition details
WMIC COMPUTERSYSTEM GET MANUFACTURER	Show system manufacturer
WMIC COMPUTERSYSTEM GET MODEL	Show model number
WMIC NIC GET MACADDRESS, DESCRIPTION	Show MAC address and NIC details
WMIC PROCESS WHERE NAME="APP.EXE" CALL TERMINATE	Kill specific process

## **User Control Commands**

## 1. net user

**What it does:** Lists all user accounts on the local machine.

**Example:**

```
net user
```

## 2. net user [Username]

**What it does:** Shows detailed information about a specific user account (status, local groups, profile path, account active, expires, etc.).

**Example:**

```
net user neeraj
```

## 3. net user [Username] /add [Password]

**What it does:** Creates a new local user. You can supply the password on the command line (less secure) or use \* to be prompted.

```
net user <username> [<password> | *] /add [additional switches]
```

**Examples:**

- Create user with inline password:  
`net user neeraj P@ssw0rd /add`
- Create user and prompt for password (recommended so password not visible in history):  
`net user neeraj * /add`

**Common useful switches used with /add:**

- `/expires:{date|never}` — expire date for the account (see Date Format notes below).
- `/active:{yes|no}` — enable or disable the account after creation.
- `/fullname:"Full Name"` — set full name.
- `/comment:"text"` — add comment for the account.
- `/passwordchg:{yes|no}` — allow user to change password.

## 4. net user [Username] /del

**What it does:** Deletes the local user account (removes the account, not necessarily the profile folder).

**Example:**

```
net user neeraj /del
```

**Warning:** Deleting an account may not remove the user profile folder under `C:\Users\`. Remove profile manually if needed.

## 5. net user Administrator /active:yes

**What it does:** Enables the built-in local Administrator account. (Use only when needed.)

**Example:**

```
net user Administrator /active:yes
```

**Note:** On many Windows versions the built-in Administrator is disabled by default for security.

## 6. net user Administrator /active:no

**What it does:** Disables the built-in Administrator account.

**Example:**

```
net user Administrator /active:no
```

**Use case:** Re-disable Administrator after emergency tasks are done.

## 7. net user [Username] /add /expires:00/00/0000

**What it does / Fix:** `net user` expects a valid date format for `/expires:`. Using `00/00/0000` causes an error. Use a valid date format (see below) **or** `never`.

**Correct examples:**

```
net user neeraj * /add /expires:10/22/2025
net user neeraj * /add /expires:Oct,22,2025
net user neeraj * /add /expires:never
```

**Date format warning:** The accepted format depends on system Regional settings. To be safe use:

- MM/DD/YYYY (e.g. 10/22/2025) **or**
- Mon, DD, YYYY (e.g. Oct, 22, 2025) — month name (short or full) is locale-agnostic and safer.

**No spaces** around commas or slashes.

## 8. net user [Username] \*

**What it does:** Prompts you to enter a password for the user (secure — not shown on screen). Can be used when creating or changing password.

**Example (create user and prompt):**

```
net user neeraj * /add
```

**Example (change password for existing user):**

```
net user neeraj *
```

# File Managing Simple Commands

## 1. TASKLIST > FILE.txt

### Purpose:

To save the list of all running tasks (programs and background processes) into a text file.

```
tasklist > tasks.txt
```

### Explanation:

- `tasklist` displays all running processes.
- The `>` symbol redirects the output into a file (`tasks.txt`).
- This means the list of all running programs will be saved in `tasks.txt`.

## 2. ECHO TYPE\_TEXT > FILE.txt

### Purpose:

To create a text file and write a single line of text in it.

### Syntax:

```
echo [text] > [filename]
```

### Example:

```
echo Hello World > note.txt
```

### Explanation:

- `echo` displays a message.
- The `>` symbol sends (writes) that message into a file.
- If the file doesn't exist, it is created automatically.
- The above example creates a file named `note.txt` containing the text “**Hello World**”.

## 3. COPY CON FILE.txt

### Purpose:

To create a new text file and type multiple lines of text directly in CMD.

### Syntax:

```
copy con [filename]
```

### Example:

```
copy con info.txt
```

Then type:

```
This is line one.  
This is line two.
```

Then press **Ctrl + Z** and hit **Enter** to save.

**Explanation:**

- `copy con` means copy from console (keyboard input).
- You can type multiple lines.
- **Ctrl + Z** is used to save and exit.
- The file `info.txt` will be created with the typed content.

#### 4. TYPE FILE.txt

**Purpose:**

To display the contents of a text file.

**Example:**

```
type info.txt
```

**Explanation:**

- The `type` command shows all the text written in the specified file.
- It is used to quickly check file content without opening Notepad.

#### 5. TYPE NUL > FILE.psd or FILE.html

**Purpose:**

To create a **blank (empty)** file of any type (like `.txt`, `.psd`, `.html`, etc.).

**Syntax:**

```
type nul > [filename]
```

**Examples:**

```
type nul > blank.txt
type nul > index.html
type nul > design.psd
```

**Explanation:**

- `nul` means nothing (empty input).
- The command creates an empty file with the given name and extension.
- It is useful for quickly creating blank files for testing or web projects.

#### Summary Table

Command	Purpose	Example	Description
<code>tasklist &gt; file.txt</code>	Save list of running tasks	<code>tasklist &gt; tasks.txt</code>	Stores process list into a text file

Command	Purpose	Example	Description
echo text > file.txt	Create file with text	echo Hello > note.txt	Writes a single line into file
copy con file.txt	Create file manually	copy con info.txt	Type lines and save with Ctrl+Z
type file.txt	Show file contents	type info.txt	Displays text inside a file
type nul > file.html	Create blank file	type nul > index.html	Creates empty HTML or any file