**Saisir les configurations machine :**

**Pour saisir la configuration matérielle (hardware) d’un PC, tu peux procéder selon plusieurs méthodes, en fonction du système d’exploitation. Voici une synthèse claire pour Linux, Windows, et avec un outil tiers multiplateforme :**

**1. Sous Linux (Ubuntu, Debian, etc.)**

**a) Commandes de base :**

**# Informations globales**

**lshw -short # Affiche un résumé de la configuration matérielle**

**sudo lshw > config.txt # Détail complet, exporté dans un fichier texte**

**# Informations sur le processeur**

**lscpu**

**# Mémoire vive**

**free -h**

**cat /proc/meminfo**

**# Disques**

**lsblk**

**df -h**

**sudo fdisk -l**

**# Carte graphique**

**lspci | grep VGA**

**glxinfo | grep "OpenGL renderer" # si mesa-utils est installé**

**# Carte réseau**

**lspci | grep Ethernet**

**ip a**

**# Carte mère et BIOS**

**sudo dmidecode -t baseboard**

**sudo dmidecode -t bios**

**Pour une saisie complète dans un rapport, tu peux rediriger chaque commande vers un fichier .txt.**

**2. Sous Windows**

**a) Interface graphique (rapide) :**

* **Touche Windows + Pause → Informations système de base.**
* **msinfo32 → Ouvre l'outil d'informations système détaillées.**
* **dxdiag → Pour les composants audio/vidéo/graphique.**

**b) En ligne de commande (PowerShell ou CMD) :**

**# Informations système complètes**

**systeminfo > config.txt**

**# Liste du matériel par catégorie**

**Get-ComputerInfo | Out-File info\_pc.txt**

**Get-WmiObject -Class Win32\_Processor**

**Get-WmiObject -Class Win32\_PhysicalMemory**

**Get-WmiObject -Class Win32\_DiskDrive**

**Get-WmiObject -Class Win32\_NetworkAdapterConfiguration | Where-Object {$\_.IPEnabled -eq "True"}**

**3. Outils tiers multiplateformes (optionnels)**

| **Outil** | **OS** | **Description** |
| --- | --- | --- |
| **CPU-Z** | **Windows** | **Affiche CPU, RAM, carte mère, etc.** |
| **Speccy** | **Windows** | **Vue d'ensemble claire du hardware** |
| **Hardinfo** | **Linux** | **GUI pour hardware + benchmarks** |
| **Neofetch** | **Linux/macOS** | **Affiche infos système stylisées** |
| **inxi** | **Linux** | **Résumé complet en ligne de commande** |

**Exemple inxi :**

**sudo apt install inxi**

**inxi -Fxz > config\_pc.txt**

**Structure de sortie type (pour saisie manuelle) :**

**Nom de la machine : [PC-DEMO-01]**

**Système : Ubuntu 22.04 / Windows 11 Pro**

**1. Processeur : Intel Core i7-9700K @ 3.60GHz**

**2. RAM : 16 Go DDR4**

**3. Disque principal : SSD 512 Go NVMe (Samsung)**

**4. Disque secondaire : HDD 1 To (Western Digital)**

**5. Carte graphique : NVIDIA GeForce GTX 1660**

**6. Carte mère : ASUS ROG STRIX Z390-F GAMING**

**7. Carte réseau : Intel I219-V Gigabit**

**8. BIOS : American Megatrends v3001, 2021**

Voici un **processus rapide, reproductible et exportable** pour **collecter les informations matérielles sur 4 PC Windows 10 / 11**.

**✅ Objectif :**

* Obtenir **un fichier .txt par PC** contenant les infos essentielles (CPU, RAM, disque, carte mère, BIOS, réseau, GPU).
* Compatible **Windows 10 et 11**.
* Ne nécessite **aucune installation**.
* Exécutable en ligne de commande (CMD ou PowerShell).

**🛠️ Script PowerShell à lancer sur chaque PC**

Ouvre **PowerShell en administrateur**, puis exécute ce script :

# Crée un fichier avec le nom du PC et la date

$pc = $env:COMPUTERNAME

$date = Get-Date -Format "yyyyMMdd\_HHmm"

$output = "$env:USERPROFILE\Desktop\config\_$pc`\_$date.txt"

# Début du fichier

"=== CONFIGURATION HARDWARE POUR $pc ===" | Out-File -FilePath $output

"Date : $(Get-Date)" | Out-File -Append $output

"" | Out-File -Append $output

# CPU

"== CPU ==" | Out-File -Append $output

Get-CimInstance Win32\_Processor | Format-List Name, NumberOfCores, NumberOfLogicalProcessors, MaxClockSpeed | Out-File -Append $output

# RAM

"`n== RAM ==" | Out-File -Append $output

Get-CimInstance Win32\_PhysicalMemory | Select-Object Capacity, Manufacturer, PartNumber, Speed | Format-Table | Out-String | Out-File -Append $output

Get-CimInstance Win32\_ComputerSystem | Select-Object TotalPhysicalMemory | Format-Table | Out-String | Out-File -Append $output

# Disques

"`n== Disques ==" | Out-File -Append $output

Get-CimInstance Win32\_DiskDrive | Select-Object Model, InterfaceType, MediaType, Size | Format-Table | Out-String | Out-File -Append $output

# Carte mère

"`n== Carte mère ==" | Out-File -Append $output

Get-CimInstance Win32\_BaseBoard | Format-List Product, Manufacturer, SerialNumber | Out-File -Append $output

# BIOS

"`n== BIOS ==" | Out-File -Append $output

Get-CimInstance Win32\_BIOS | Format-List Manufacturer, SMBIOSBIOSVersion, ReleaseDate | Out-File -Append $output

# Carte graphique

"`n== GPU ==" | Out-File -Append $output

Get-CimInstance Win32\_VideoController | Format-List Name, DriverVersion, AdapterRAM | Out-File -Append $output

# Réseau

"`n== Réseau (IP uniquement) ==" | Out-File -Append $output

Get-NetIPAddress | Where-Object {$\_.AddressFamily -eq "IPv4" -and $\_.InterfaceAlias -notlike "Loopback\*"} | Format-Table InterfaceAlias, IPAddress | Out-String | Out-File -Append $output

# Fin

"`n=== FIN DU RAPPORT ===" | Out-File -Append $output

# Affiche le chemin du fichier généré

Write-Host "Rapport généré : $output"

**📂 Résultat**

* Un fichier sera généré sur le **bureau de chaque PC**, par exemple :
* config-PC01\_20250709\_1430.txt
* Tu peux ensuite copier ces 4 fichiers sur une clé USB ou les envoyer.