■ HardwareInfoProvider v2.2

C++ / Qt library for retrieving detailed system information (CPU, GPU, RAM, Disk, OS).
Supports **Windows** and **Linux**.

■■ Supported Platforms

Component	Windows	Linux
CPU (name, cores, frequency)	■ WMI / Win32_Processor	■ `/proc/cpuinfo`
GPU (name)	■ DXGI with deduplication	■ `lspci` / `/sys/class/drm`
GPU VRAM (total, free, used)	■ DXGI 1.4 (Windows 10+)	■ `nvidia-smi` / sysfs
RAM	■ GlobalMemoryStatusEx	■ `/proc/meminfo`
Disk (type, size, filesystem)	■ WMI / MSFT_PhysicalDisk	■ `lsblk`
OS (name, version, kernel, architecture)	■ WMI / Win32_OperatingSystem	■ `/etc/os-release`

■ Public API

Method	Returns
`QString getCPUInfo()`	CPU name, core count, frequency
`QString getGPUInfo()`	Full list of all GPUs
`QString getGPUName()`	Name of the first GPU
`quint64 getGPUMemoryMB()`	Total VRAM size
`quint64 getGPUFreeMemoryMB()`	Free VRAM size
`quint64 getGPUUsedMemoryMB()`	Used VRAM size
`double getGPUMemoryUsagePercent()`	VRAM usage percentage
`QList getDisks()`	List of disks with type (SSD/HDD/External/Removable)
`QString getOSInfo()`	OS name, version, and architecture
`QString getRAMInfo()`	Total, free, and used RAM

■ Platform-Specific Details

■ *Windows*

- **GPU:**
Uses **DXGI (DirectX Graphics Infrastructure)** through `EnumAdapters()`,
with **deduplication by VendorId, DeviceId, and Name**.
This eliminates duplicate entries when using multiple GPUs (e.g., iGPU + dGPU).

- **GPU Memory:**
Retrieved using **IDXGIAdapter3::QueryVideoMemoryInfo()** (Windows 10+).
Supports the following fields:
- `DedicatedVideoMemory`
- `Budget`
- `CurrentUsage`

- **Disk Info:**
Hybrid detection using **MSFT_PhysicalDisk** and **Win32_DiskDrive**.
Supports type detection:
- `SSD`, `HDD`, `External`, `Removable`

■ ***Linux***

- **GPU:**
Uses `lspci -v` for analyzing `VGA controller` devices and `/sys/class/drm` (reads `vendor` and `device` files).
Detects VRAM for NVIDIA and AMD via `nvidia-smi` and sysfs.

- **Disk Info:**
Uses `lsblk -d -o NAME,ROTA,TRAN,TYPE,MODEL`.
Includes caching via `QMap` to minimize external process calls.

■ Example Output (Windows)

```
=== System Information ===
Platform: "Windows"

OS: "Windows 11 Version 24H2"
Kernel: "10.0.26100"
Arch: "x86_64"

CPU: "AMD Ryzen 7 5700X3D 8-Core Processor"
CPU Cores: 16
CPU Frequency: 3.2 GHz

GPU:
GPU 1: "NVIDIA GeForce RTX 3080"
Total VRAM: 10240 MB ("10.00 GB")
Used VRAM: 3584 MB ("3.50 GB")
Free VRAM: 6656 MB ("6.50 GB")
Usage: 35.0%

GPU 2: "Intel(R) UHD Graphics 770"
Total VRAM: 512 MB ("0.50 GB")
Used VRAM: 128 MB ("0.12 GB")
Free VRAM: 384 MB ("0.38 GB")
Usage: 25.0%

Disk Info:
• "C:/" (NTFS)
Type: "SSD"
Size: "500 GB"
Free: "320 GB"
Usage: "36%"
```

■ Implementation Notes

- **GPU deduplication fixed:**
DXGI no longer lists duplicate adapters (filtered by `VendorId + DeviceId + Name`).

- **Linux `lsblk` results cached:**
Disk type (SSD/HDD/External) is cached in memory for faster subsequent calls.

- **Cross-platform consistency:**
`getGPUInfo()` always returns a single `QString`—concatenated GPU list on Windows, single entry on Linux.

■ Integration Example (Qt)

```
// mainwindow.cpp
#include "HardwareInfoProvider.h"
#include

MainWindow::MainWindow(QWidget *parent)
: QMainWindow(parent)
{
HardwareInfoProvider hw;

qDebug() << "=== System Info ===";
qDebug() << "OS:" << hw.getOSInfo();
qDebug() << "CPU:" << hw.getCPUName();
qDebug() << "Cores:" << hw.getCpuCoreCount();
qDebug() << "Frequency:" << hw.getCpuFrequencyGHz() << "GHz";
qDebug() << "RAM total:" << hw.getTotalRAMGB() << "GB";

#ifdef _WIN32
qDebug() << "GPU info:" << hw.getGPUInfo();
qDebug() << "VRAM total:" << hw.getGPUMemoryMB() << "MB";
qDebug() << "VRAM used:" << hw.getGPUUsedMemoryMB() << "MB";
qDebug() << "VRAM free:" << hw.getGPUFreeMemoryMB() << "MB";
#else
qDebug() << "GPU:" << hw.getGPUName();
#endif

QList disks = hw.getDisks();
for (const auto &disk; : disks) {
qDebug().noquote() << QString("Disk %1 (%2) - %3")
.arg(disk.mountPoint)
.arg(disk.fileSystem)
.arg(disk.type);
}
}
```

■ ***Build Notes***

- On **Windows**, make sure the project links to `dxgi.lib` and `wbemuuid.lib`
(required for WMI and DXGI).
- On **Linux**, ensure `lsblk` and `lspci` utilities are installed (they are included by default in most distros).
- In your **Qt .pro file**, include:

```
QT += core gui
```

```
CONFIG += c++17
```

■■ Version History

- **v2.2** — Added Qt integration example (mainwindow.cpp)
- **v2.1** — DXGI GPU deduplication + Linux disk caching
- **v2.0** — Initial WMI + `/proc` implementation
- **v1.0** — Basic version without VRAM or disk support