SIDEBAR Operating Systems

Windows 11 Memory Troubleshooting Cheat Sheet

Sidebar

Operating Systems · Windows 11 Memory Troubleshooting Cheat Sheet

What the memory words mean, which numbers to read, and how to tell a leak from a cache from a busy PC. Counter names were all run on this PC (Windows 11 Pro, build 26300, 32 GB installed / 29.8 GB usable) as a normal user. Interpretations come from Microsoft's documentation and the course Windows 11 Memory, Performance & Crash Forensics; the tools RAMMap and PoolMon were not run here.

WORDS THAT GET CONFUSED

`Working set`	Physical RAM a process is using now (shared pages included)
`Private working set`	RAM only that process uses. Task Manager's *Memory* column on the Processes tab shows this
`Commit (private bytes)`	Memory a process has *promised*: RAM plus page file. Task Manager's *Commit size* column
`Commit charge / limit`	System total promised / the maximum (RAM + page file). Here 29.5 of 77.8 GB
`In use`	RAM holding data programs need right now
`Standby (cached)`	Old data kept in RAM because it might be needed again; Windows gives it back on demand. Counts as Available
`Modified`	Changed pages waiting to be written to disk
`Free`	Truly empty RAM. Normally small on a healthy PC
`Available`	Free + Standby: what can be handed to a program right away

COUNTERS THAT MATTER

`\Memory\Available MBytes`	RAM that can be used immediately (13,584 here)
`\Memory\Committed Bytes`	Commit charge
`\Memory\Commit Limit`	RAM + page file
`\Memory\% Committed Bytes In Use`	Charge as a percentage of the limit (35 here)
`\Paging File(_Total)\% Usage`	How much of the page file is in use; *% Usage Peak* is the high-water mark
`\Memory\Pages Input/sec`	Pages read from disk to satisfy page faults. Sustained high values with low Available memory mean a starved PC
`\Memory\Page Faults/sec`	Includes harmless *soft* faults: large numbers alone mean little
`\Memory\Pool Nonpaged Bytes`	Kernel memory that must stay in RAM; steady growth points at a driver
`\Memory\Pool Paged Bytes`	Kernel memory that can be paged out
`\Memory\Standby Cache Normal Priority Bytes`	The bulk of the Standby cache (13.6 GB here)
`\Process(_Total)\Handle Count`	Open handles system-wide (192,314 here); watch a single process for runaway growth

QUICK LOOKS (POWERSHELL, NO ADMIN)

`Get-Counter '\Memory\Available MBytes'`	Add `-Continuous` to sample every second; Ctrl+C to stop
`Get-CimInstance Win32_OperatingSystem`	`TotalVisibleMemorySize`, `FreePhysicalMemory`. Careful: *Free* here is not the Free list; use the counters above
`Get-CimInstance Win32_PageFileUsage`	Page file name, size (`AllocatedBaseSize`) and `PeakUsage` in MB
`Get-CimInstance Win32_PhysicalMemory`	The modules: size, rated `Speed`, `ConfiguredClockSpeed`, part number
`Get-Process -Name 'Memory Compression'`	The compressed-memory store (around 1.4 GB here); its size counts in *In use*

Memory Compression's on/off setting needs `Get-MMAgent`, which was access-denied without administrator rights here.

TOP CONSUMERS

```
Get-Process | Sort-Object WorkingSet64 -Descending | Select-Object -First 10 Name, Id, @{n='MB';e={[int]($_.WorkingSet64/1MB)}}
```

Biggest by working set

```
Get-Process | Sort-Object PrivateMemorySize64 -Descending | Select-Object -First 10 Name, Id, @{n='CommitMB';e={[int]($_.PrivateMemorySize64/1MB)}}
```

Biggest by commit (private bytes)

```
Get-Process | Sort-Object Handles -Descending | Select-Object -First 10 Name, Id, Handles
```

Most handles

```
Get-Process firefox | Measure-Object WorkingSet64 -Sum
```

Total for one program with many processes (browsers: 27 here). Working sets overlap, so the sum overstates real use; compare PrivateMemorySize64

LEAK, CACHE OR BUSY?

`Leak`	A process's commit or a kernel pool grows steadily and never gives anything back, even when the program is idle. It needs hours of samples, not minutes
`Cache`	Growth that stops by itself, or shrinks when other programs ask for memory (the Standby list does this by design)
`Sawtooth`	Grows, drops, grows: ordinary garbage collection or a program cleaning up. Not a leak
`Busy`	Available low but Pages Input/sec near zero and page file use low: Windows is using spare RAM as cache. Fine
`Starved`	Available low, Pages Input/sec high, page file filling, everything slow: too little RAM for the workload
`Handle leak`	One process's Handles climbs while its memory stays flat
`Pool leak`	Pool Nonpaged/Paged Bytes climbs while no process's memory does: a driver. PoolMon (Windows Driver Kit) finds the tag

PAGE FILE AND COMMIT

`Why have one`	It backs commit (so the limit is higher than RAM) and holds crash dumps if the dump type needs it
`System-managed`	Windows picks the size. Here 48 GB (1.5 x RAM) with a peak use of 153 MB: far larger than needed, and the reason is unexplained
`RAM + page file`	= Commit Limit (checked here to the byte: 83,522,240,512)
`Changing it`	Settings > System > About > Advanced system settings > Performance > Advanced > Virtual memory. Note the old value first; needs admin and a restart
`Never turn it off`	Programs can then fail when commit runs out, even with RAM free

OTHER TOOLS

`Task Manager > Performance > Memory`	In use, Available, Committed, Cached, Paged/Non-paged pool, Speed, Slots used
`resmon`	Resource Monitor > Memory tab: the Hard Faults/sec column and the physical memory bar (Hardware Reserved, In Use, Modified, Standby, Free)
`RAMMap (Sysinternals)`	Shows where RAM goes by type, and can empty the Standby list. Needs administrator rights. Not run here
`PoolMon (WDK)`	Kernel pool by tag: for finding a leaking driver. Not installed here
`mdsched.exe`	Windows Memory Diagnostic: restarts the PC and tests the RAM. A pass is weak evidence; a fail is strong. Never run here

FREE IS NOT THE NUMBER TO WATCH

A PC showing 13.7 GB *free* in one place and 2 GB in another is not broken: some tools report Free + Standby (Available) as free. Read Available MBytes and Pages Input/sec, not the word *free*.

ONE SAMPLE PROVES NOTHING

A cache looks like a leak for the first few minutes. Sample for hours and compare like with like (same program, same activity) before deciding.