Windows 11 Crash Triage Cheat Sheet

Sidebar

Operating Systems · Windows 11 Crash Triage Cheat Sheet

The first ten minutes after a blue screen or an unexpected restart: what to record, which events say what happened, how to decode the stop code, where the dumps are and what to save before anything overwrites it. Event IDs, field names and the dump settings were read from the PC this was written on (Windows 11 Pro, build 26300) as a normal user; the WinDbg commands and the bug check reference link are from Microsoft's documentation and were not run here.

FIRST TWO MINUTES: RECORD, DON'T FIX

`Time`	Write down when it happened, and what you were doing. Event times and your memory are the two halves of the timeline
`Stop code text`	Windows 11 shows a name such as *KERNEL_MODE_HEAP_CORRUPTION* and a QR code; a phone photo of the screen is enough
`Don't clean up`	Disk Cleanup and Storage Sense can delete *System error memory dump files*: the evidence. Leave them until the dumps are copied
`Don't uninstall yet`	One change at a time, recorded (see the Drivers & Updates sheet). Changing three things first makes the cause unknowable
`Keep a log`	Date, stop code, what changed recently, what you tried. A plain text file is fine

DID IT CRASH OR JUST RESTART?

`Kernel-Power 41`	*The system has rebooted without cleanly shutting down first.* Written at the next start. The fields tell you which kind
`BugcheckCode ≠ 0`	A blue screen. The code is stored in decimal: `314` is `0x13A`
`BugcheckCode = 0`	Windows had no chance to say why: power loss, a hard hang, a forced power-off or a heat cut-out. No dump will exist
`PowerButtonTimestamp`	A time here (and `LongPowerButtonPressDetected = true`) means the power button was held. On this PC both no-code restarts show *0 / false*
`EventLog 6008`	*The previous system shutdown at … was unexpected.* The same story from the Event Log service, with the last-alive time
`User32 1074`	A planned restart or shutdown, and which process asked for it. Not a crash
`Kernel-Power 109`	*The kernel power manager has initiated a shutdown transition.* Normal: it accompanies ordinary restarts
`WER-SystemErrorReporting 1001`	Written after a blue screen: *The bugcheck was: 0x0000013a (…four parameters…). A dump was saved in: C:\WINDOWS\Minidump\…*

Both Kernel-Power 41 and event 1001 are the first thing to read: together they give the code, the four parameters and the dump's file name.

DECODE THE STOP CODE

`'{0:X}' -f 314`	PowerShell: decimal from event 41 to hex (`13A`)
`[Convert]::ToString(314,16)`	The same, another way
`learn.microsoft.com`	Search *bug check code reference* and the code (e.g. *0x13A*): the page names the four parameters and what each means
`Four parameters`	Every bug check has four. For 0x13A: P1 type of heap corruption, P2 the heap, P3 where it was detected, P4 reserved
`Hex vs decimal`	Event 41 shows decimal; documentation, WinDbg and event 1001 show hex

WHERE THE EVIDENCE LIVES

`C:\Windows\Minidump\*.dmp`	Small dumps, one per crash (about 3.6 MB each on this PC). Listing works as a normal user; *reading them needs administrator rights*
`C:\Windows\MEMORY.DMP`	The full/kernel/automatic dump, if the dump type is set to one (not present on this PC)
`CrashDumpEnabled`	`HKLM\SYSTEM\CurrentControlSet\Control\CrashControl`: 0 none, 1 complete, 2 kernel, 3 small, 7 automatic (here: 3). Read-only to look at
`sysdm.cpl`	System Properties > Advanced > Startup and Recovery: the dump type, auto-restart and the dump path (changing it needs admin and a note of the old value)
`AutoReboot`	Set to 1 here: Windows restarts by itself after a blue screen, so you may miss the screen. Untick *Automatically restart* to read it
`perfmon /rel`	Reliability Monitor: a stability index (1 to 10) and a day-by-day list of crashes, failed updates and app failures

OTHER LOGS THAT HELP

`Application Error 1000`	An application crashed: the faulting program and module (e.g. `KAPS.exe`, 19 times here)
`Application Hang 1002`	A program stopped responding
`Windows Error Reporting 1001`	Application log: a report was created. Very noisy (4,755 here), so filter by time
`Service Control Manager 7034`	*The service terminated unexpectedly*; 7031 adds that it will be restarted
`disk 7 / 11 / 51 / 153`	Bad block / controller error / paging error / I/O retried. Storage trouble, often USB drives
`WHEA-Logger`	Hardware errors the CPU or memory reported. None on this PC. Reading rule: *none* is weak evidence, not an all-clear

SAVE IT BEFORE IT GOES

`Copy the dumps`	Administrator PowerShell: `Copy-Item C:\Windows\Minidump\*.dmp C:\crash-evidence\`. Open copies, never the originals
`wevtutil epl System C:\crash-evidence\System.evtx`	Export the System log (probably needs admin; not tested here). The System log holds only 20 MB and here reaches back to July
`systeminfo > C:\crash-evidence\si.txt`	OS build, hotfixes, RAM, BIOS version: a one-file snapshot
`Screenshot Reliability Monitor`	It keeps about a year but is easy to lose with a reinstall
`Hash the copies`	`Get-FileHash` on each, in a text file: proves later they are unchanged

THEN READ THE DUMP (NOT NEEDED IN THE FIRST TEN MINUTES)

`windbg -z <dump>`	Open a copy in WinDbg with the symbol path set (`srv*C:\Symbols*https://msdl.microsoft.com/download/symbols`)
`.bugcheck`	Shows the code and parameters. Check they match the event log: if not, wrong dump
`!analyze -v`	Microsoft's first command: bug check, *Probably caused by* and the stack. A suspect, not a verdict
`lm`	List the loaded modules (drivers) with dates
`Two dumps`	The same module in both is much stronger than one dump

Small dumps show only the thread that stopped: damage done earlier by something else may not be visible.

TWO CRASHES WITH THE SAME CODE

Same stop code twice is a lead, not a diagnosis. Line the crashes up against what changed (Windows Update history, new drivers, new software) before blaming a part. The Drivers & Updates sheet has the commands.

THE 17-MINUTE GAP

Event 41 is written at the next start, and *last alive* (event 6008) can be minutes before it. Use the earlier time when you look for what happened.