SIDEBAR Operating Systems

Windows 11 Event Viewer Cheat Sheet

Sidebar

Operating Systems · Windows 11 Event Viewer Cheat Sheet

Event Viewer's logs and filters, the event IDs that matter when a PC crashes or misbehaves, and the PowerShell and wevtutil equivalents. Event IDs and counts quoted as here were read from the System and Application logs of the PC this was written on (Windows 11 Pro, build 26300) as a normal user; the Security log is not readable without administrator rights. The Event Viewer window itself was not clicked through.

OPENING AND LAYOUT

`eventvwr.msc`	Win+R, or right-click Start > Event Viewer
`Windows Logs`	Application, Security, Setup, System, Forwarded Events
`Applications and Services Logs`	Per-component logs; `Microsoft > Windows > …` holds hundreds (e.g. Kernel-Power, WHEA)
`Custom Views > Administrative Events`	Critical, Error and Warning from all logs in one list: a good first look
`Middle pane`	The event list; the bottom pane has the General text and a Details tab (Friendly or XML view)
`Right pane (Actions)`	Filter Current Log, Create Custom View, Save All Events As, Attach Task, Refresh, Clear Log

LEVELS

`Critical`	Level 1: serious failure, e.g. Kernel-Power 41
`Error`	Level 2
`Warning`	Level 3
`Information`	Level 4
`Verbose`	Level 5: diagnostic detail
`Audit success / failure`	Keywords used in the Security log

FILTER CURRENT LOG

`Logged`	A time range: last hour, last 24 hours, 7 days, 30 days or custom
`Event level`	Tick Critical, Error, Warning to cut the noise
`By source`	Choose the provider, e.g. `Microsoft-Windows-Kernel-Power`
`Event IDs`	Type `41,6008` to include; `-7036` (with a minus) to exclude; `1-100,-17` for a range minus one
`Task category / Keywords`	Narrow further; usually unnecessary
`Save Filter to Custom View`	Keep a useful filter for next time
`Details tab > XML View`	Shows the raw event, including the named fields (e.g. `BugcheckCode`)

EVENT IDS FOR CRASHES AND RESTARTS (SYSTEM LOG)

`Kernel-Power 41`	Critical: restart without a clean shutdown (4 here). `BugcheckCode` 0 = no stop code
`EventLog 6008`	*The previous system shutdown at … was unexpected* (3 here)
`EventLog 6005 / 6006`	The Event Log service started / stopped: roughly boot and shutdown (41 / 38 here)
`Kernel-General 12`	The operating system started (41 here)
`User32 1074`	A process or user initiated a restart/shutdown, with the reason (39 here)
`Kernel-Power 109`	Shutdown transition started: normal (37 here)
`WER-SystemErrorReporting 1001`	After a blue screen: the code, four parameters and dump path (2 here)
`WHEA-Logger 17 / 18 / 19`	Hardware error reports. Per third-party guides, 18 is a fatal machine-check, 19 a corrected one. None here

Event 1001 here has the Provider name *Microsoft-Windows-WER-SystemErrorReporting*, not *BugCheck*: filter on the first.

EVENT IDS FOR DISKS, SERVICES AND UPDATES

`disk 7` *The device … has a bad block* (1,402 here)

`disk 11` *The driver detected a controller error on …* (128)

`disk 51` *An error was detected on device … during a paging operation* (326)

`disk 153` *The IO operation … was retried* (1,693)

`Ntfs 55` File system structure is corrupt: run chkdsk (none here)

`Service Control Manager 7034 / 7031` A service terminated unexpectedly / and will be restarted

`Service Control Manager 7045` A new service was installed: useful when something new appears

`WindowsUpdateClient 19 / 20` Update installed successfully / installation failure (435 / 44 here)

`Application Error 1000` A program crashed: faulting application and module

`Application Hang 1002` A program stopped responding

GET-WINEVENT (POWERSHELL)

```
Get-WinEvent -FilterHashtable @{LogName='System'; ProviderName='Microsoft-Windows-Kernel-Power'; Id=41}
```

All unexpected-restart events, newest first

```
Get-WinEvent -FilterHashtable @{LogName='System'; Level=1,2; StartTime=(Get-Date).AddHours(-24)}
```

Critical and Error events in the last day

```
Get-WinEvent -LogName System -MaxEvents 20
```

The 20 newest events

```
Get-WinEvent -LogName System -FilterXPath "*[System[(EventID=6008)]]"
```

XPath filter (what Event Viewer builds from your filter)

```
Get-WinEvent -ListLog * | Where-Object RecordCount -gt 0
```

Which logs hold anything

```
([xml]$e.ToXml()).Event.EventData.Data
```

Read an event's named fields (`$e` is one event from the above)

```
$e.Message
```

The rendered text of one event

Use `-ProviderName` and `-Id` inside the hashtable, together with `LogName`; a wrong provider name gives *The parameter is incorrect.*

WEVTUTIL (COMMAND PROMPT)

```
wevtutil qe System /q:"*[System[(EventID=41)]]" /c:5 /rd:true /f:text
```

Query: the 5 newest event-41s as text

```
wevtutil el
```

List every log name

```
wevtutil gl System
```

Show a log's settings (size, retention, path)

```
wevtutil epl System C:\evidence\System.evtx
```

Export a log (may need admin)

```
wevtutil cl System
```

Clear the log. Destroys evidence: export first, and do not do it while investigating

LOG SIZE AND SAFETY

`Right-click a log > Properties`	Maximum log size and what to do when full
`System / Application here`	20 MB each; the System log was full and reached back only to 21 July
`Enlarge a log`	Raise Maximum log size (e.g. to 100 MB) so crashes stay available for longer; needs admin. Undo: set the old size back
`Save All Events As…`	Saves an `.evtx` file you can reopen in Event Viewer on any PC
`Open Saved Log…`	View an exported log without touching the live one
`Clear Log…`	Offers to save first. Don't clear logs while a problem is open

COUNTS HERE ARE A SNAPSHOT

The numbers in brackets are from this PC on 24 October 2026 and will differ on yours. The IDs and sources are what to keep.

A LOG THAT WRAPPED HAS LOST ITS PAST

The System log here holds only 20 MB and reaches back about three months. If a problem is rare, export the log when it happens, or the evidence may be overwritten before you look.