WinDbg Cheatsheet 2026 (Userland)

Navigation

t (F11)	Step Into
p (F10)	Step Over
g (F5)	Run
gu	Execute until the current function is complete
gn	Execute passing exceptions to debugged process
restart	Stop and restart execution

Breakpoints

bl	Lists breakpoints
bp [addr]	Set breakpoint
bc #	Clear breakpoint (takes wildcards)
bd #	Disable breakpoint (takes wildcards)
be #	Enable breakpoint (takes wildcards)
ba [a] [s] [addr]	Hardware breakpoint where [a] is access (rwx) and [s] is size.

Registers

r	Display all registers and their values
r [register]	Displays information about a single register
r [register]=[value]	Changes value of a register

Inspecting Memory

db [addr]	displays data as Bytes + ASCII (at address [addr])
da [addr]	displays data ASCII string until NULL is found
du [addr]	displays data UNICODE string until NULL is found

Patching Memory

eb [addr] [value]	patches [addr] with single Byte
ew [addr] [value]	patches [addr] with word (16 bits)
ed [addr] [value]	patches [addr] with double word (32 bits)
eza [addr] [value]	patches [addr] with ASCII string
ezu [addr] [value]	patches [addr] with UNICODE string

Searching Memory

s -[type] [range] [pattern]	search to a specific type of pattern in a memory range.
s -a 0 100 ”string”	search ASCII string from address 0 until 100.

Vieweing Memory Maps

!address	Displays all memory maps and properties.
!address [addr]	Checks if [addr] is part of a valid memory map.
!address -summary	Displays general information about memory usage.
!address [filters]	Shows only memory maps with specific properties (/f:Type=MEM_PRIVATE and /f:Protect=PAGE_EXECUTE_READWRITE).
!vprot [addr]	Displays information about protection.

Inspecting Modules

lm	Lists modules
lm o	Lists only loaded modules
lm a [addr]	Lists the module that contains the address [addr]

Disassembler

u [addr] #	Disassembly a number of instructions from a memory address

Symbols

ld *	Downloads, caches and loads symbols of all loaded modules from configured source.
ld [module]	Downloads, caches and loads symbols of a specific loaded module.
x module!symbol	Display the symbols that match the specified pattern, can contain wildcard.

Data Types

dt name addr	Specify the address of the struct (e.g. “dt ntdll!_TEB @$teb”).
dt -r name	Recursively dump the subtype fields.
dt name field	Specify the specific field to display.
dds [range]	Display DWORD (4 byte) values and symbols.

Stack

k	Display basic call stack.
kp	Display call stack with full parameters.
kb	Display call stack with three first parameters.
!stack	Summary of the current threadś stack usage.

Pseudo-Registers

$peb	Address of the current process’ Process Execution Block (e.g. “*dt _PEB @$peb*”).
$ted	Address of the current thread’s Thread Execution Block.
$exentry	The address of the executable’s entry point.

Help

?	Help on Debugee commands.
.help	Help on Debugger commands.

MISC

.writemem [file] [addr] L[size]	Dumps L Bytes from address addr to file.
.effmach [arch]	Switches architecture used by the engine (interesting to analyse WoW).
!exchain	Display the current exception handler chain.
!teb	Displays the Thread Environment Block.
!gle	Shows the GetLastError code for the current thread.
!error [code]	Decodes a specific hex error code (like $0x80070005$) into its Windows error name.
.printf [fmt] [prams]	meta-command implementing C-like printf function.
.reload	Forces the debugger to discard its current symbol information and reload it.
.cls	Clears the command output screen.
.sympath [path]	Sets where WinDbg looks for symbols (“*.pdb*” files).
.dvalloc	Allocates memory using VirtualAllocEx.
f [range] [pattern]	Fills a specified memory range with a repeating pattern (*e.g. “f 0012ff40 L20 0”*).
m [range] [addr]	Moves memory from one address to another. (*e.g. “m 0012ff40 L20 0012ff80”*).

UI Shortcuts

ENTER	Repeats last command.
ALT + 1	Command window.
ALT + 7	Disassembly window.
Alt + Shift + T	Opens the Threads window.