

---

# Debugger Command Reference

## Command Groups

1. WinDBG Command Line Options
2. General Commands
3. Symbol File Commands
4. Symbol Usage Commands
5. Executable Image Commands
6. Extension DLL Commands
7. Memory Display Commands
8. Memory Commands
9. Breakpoint Commands
10. Execution Control Commands
11. Stack Commands
12. Memory Edit Commands
13. Source Code Commands
14. Script Keywords
15. Disassembler Commands

## WinDBG Command Line Options

Option	Description
-Q	Suppresses prompt for saving workspace
-y	Path to symbols, symbol server
-srcpath	Source files directory path
-b	Debugger breaks into target system upon connection and immediately after reboot
-g	Ignores initial breakpoint in target application
-z	Path to crash dump used for post-mortem debugging
-snul	Restricts symbol search to loaded modules only
-p	Specifies the decimal process ID to be debugged. This is used to debug a process that is already running
-k	Specified options for kernel debugging
-QSY	Automatically reload source files (suppresses reload prompt)
-c	Specifies the initial debugger command to run at start-up

## General Commands

Command	Description
.cls	Clears the debugger's command screen
vertarget	Displays OS version, system uptime and debug time
vercommand	Displays command line passed to the debugger
.logopen	Writes the subsequent contents of the debugger command window to the specified logfile overwriting its contents
.logclose	Closes the existing logfile (if any)
.logappend	Writes the subsequent contents of the debugger command window to the specified logfile <b>without</b> overwriting its contents
.logfile	Determines whether a log file exists and displays the file's status
.pcmd	Causes a command to be issued when the debugger prompt is displayed
.write_cmd_hist	Writes Command history to file (in reverse chronological order)
.wtitle	Sets the debuggers window title
.shell	Executes a shell program, redirects the debuggers output to the program and displays the output of the program in the debugger

## Symbol File Commands

Command	Description
.sympath	Sets / Queries the symbol search path
!lmi	Displays symbol information about the module
.symfix	Sets the symbol path to point to the Microsoft public symbol server
!sym noisy quiet	Enables/Disables verbose symbol loading
.reload	Forces debugger to reload the symbols for all modules
!chksym	Checks .PDB for a module and validates it
.symopt	Enables/disables symbol options
lm	List Modules and their symbol status
ld	Loads symbols for the specified module and updates all module information. Same as .reload -f and .symopt- 0x4

## Symbol Usage Commands

Command	Description
x	Displays the address of the symbol in the specified module
ln	List symbols near the given address
dps	Displays contents of the specified address range and maps the contents to symbols
dv	Displays local variables and functions parameters for current stack frame. Requires private symbols.
dt	Displays structure type information
.typeopt	Controls Display of variables in Watch and Locals windows
.symopt	Controls symbol options
.frame .f+ .f-	Changes or displays the local context (scope) to be used to interpret local variables
k	Display call stack of the current thread

## Executable Image Commands

Command	Description
!dh	Displays the Portable Executable (PE) header information of a module
!mv	Display the start and end load address of modules
!chkimg	Check for corruptions in the copy of an executable loaded in memory against the binary image from the symbol server or executable file path
.imgscan	Scans user VAS for PE image headers and displays them
!imgreloc	Displays information about modules inside a process that have been relocated from their preferred base addresses
!dlls	Display information about DLLs loaded into a process

## Extension DLL Commands

Command	Description
.chain	Prints information on all currently loaded extension DLLs
.extpath	Sets the extension DLL search path
.load	Display words values(16 bit)
.loadby	Loads a new extension DLL into the debugger
.unload	Unloads an extension DLL from the debugger
.unloadall	Unloads all extension DLLs from the debugger
.setdll	Changes the default extension DLL used by the debugger
.extmatch	Show exported functions of the given extension DLL

## Memory Display Commands

Command	Description
db	Display byte values and ASCII characters (8 bit)
dW	Display words and wide-char characters (16 bit)
dw	Display words values (16 bit)
dd	Display double word values (32 bit)
dq	Display quad words values (64 bit)
dp	Display pointer values (32 bit on x86 and 64 bit on x64)
dps	Display pointer values and resolve them to symbols
dpa	Display pointer values and the ASCII string they point to
dpu	Display pointer values and the Unicode string they point to
da	Display a NULL terminated ASCII string
du	Display a NULL terminated wide-char string
ds	Display an ANSI_STRING structure
dS	Display an UNICODE_STRING structure
dc	Display double word values and ASCII characters

## Memory Commands

Command	Description
s	Searches memory for the specified pattern
c	Compares contents of memory blocks
f	Fills memory with a pattern
m	Moves memory
poi	Returns pointer-sized data from the specified address
wo, dwo, qwo	Returns a word, dword or qword respectively from the specified address
.writemem	Writes contents of a memory range to a file
.readmem	Reads the contents of a file into a memory range
.holdmem	Hold and Compare Memory (used if memory contents change)

## Debugger Breakpoint Commands

Command	Description
ba	Sets a data breakpoint ( breakpoint on access )
bc	Clears a previously set breakpoint
be	Enables a previously set breakpoint
bd	Disables a previously set breakpoint
bl	Lists all breakpoints (including ones that are disabled)
bp	Sets an execution breakpoint on an address
bu	Sets an execution breakpoint on a symbol
bm	Sets multiple execution breakpoints based on wildcard
.bpcmds	Displays the commands that were used to set the breakpoints

## Execution Control Commands

Command	Description
g	Continues instruction execution
gu	Step out of the current function
gc	Resumes execution from a conditional breakpoint
p t	Step (skip over function calls)   Trace (trace into function calls)
pa ta	Steps   Traces through instructions till the specified address
pb tb	Steps   Traces through instructions till a branch instruction
pc tc	Steps   Traces through instructions till the next call
pt tt	Steps   Traces through instructions till the next ret
pct tct	Steps   Traces through instructions till the next ret or call
wt	Executes instructions and displays statistics
.prompt_allow	Controls the information that is displayed along with the debugger prompt while single stepping or tracing through code
.suspend_ui	Suspends the refresh of WinDbg debugging information windows while single stepping through code
.step_filter	Creates a list of functions that are skipped when tracing through source code

## Stack Commands

Command	Description
k, kb, kp, kP, kv	Displays stack back trace
dv, x -v	Displays local variables (private symbols required)
.frame	Displays / Sets the function context for local variables
.kframes	Sets the debugger's stack frame display depth (default 255)
~*kvn	Displays call stacks for all threads in a process
!uniqstack	Displays thread stacks in a process that are not duplicates
!findstack	Displays all stacks that contain the specified symbol inside a process
kv = Address	Displays the stack using Address as the frame pointer
!teb	Displays the commit limits of the current thread's user mode stack
dps @esp	Displays the raw contents of the call stack

## Memory Edit Commands

Command	Description
eb	Edit byte (8 bit) values
ew	Edit word (16 bit) values
ed	Edit double word (32 bit) values
eq	Edit quad word (64 bit) values
ep	Display pointer (32 bit on x86 and 64 bit on x64) values
ea	Edit ASCII string (not NULL-terminated)
eza	Edit ASCII string (NULL-terminated)
eu	Edit Unicode string (not NULL-terminated)
ezu	Edit Unicode string (NULL-terminated)

## Source Code Commands

Command	Description
.srcpath	Sets / Queries the source search path
.lines	Toggle Source Line Support
.srcfix/.lsrcfix	Automatically set the source path to indicate that a source server will be used
.srcnoisy	Controls the verbosity of source file loading
.open	Searches the source path for a source file and opens this file.
l (lower case L)	Set the source line options that control source display and program stepping options
ls/lsa	Display a series of lines from the current source file and advance the current source line number
lsc	Displays the current source file name and line number.
lse	Opens an editor (WinDBG or external) for the current source file
lsf	Load or unload a source file
lsp	Controls the number of source lines displayed while single stepping through code or by the ls/lsa command

## Script Keywords

Command	Description
.for	Implement loops with initialization, condition and step parameters
.break	Break out of a .for/.while/.do loop
.continue	Used to skip the remaining part of loop for the current iteration.
.if	Used to execute a block of statements conditionally
.block	Implements a block to enclose statements like {} in C.
.foreach	Parses and iterates over the output of debugger command, string or file contents
.echo	Displays a string
.printf	Formats and displays variables and strings like printf() in C.
.else	Used with .if to enclose statements to be executed if the condition evaluates to a false
.elseif	Used to replace a .else followed by a .if
.catch	Prevents a program from terminating if an error occurs.
.leave	Exits from a .catch block.
.while	Implement loops with a condition

## Disassembler Commands

Command	Description
u	Disassembles a few instructions starting at a specific address using linear scan
ub	Disassembles backwards starting at a specific address
uf	Disassembles an entire function, given any address within the function
uf /c	Displays only the call instructions in a routine instead of the full disassembly
.asm	Specifies assembly display options
#	Searches for specific disassembly patterns