WinDbg CheatSheet

Help Commands

`?`	Help on debugee commands
`.help`	Help on debugger commands
`.hh command`	Open WinDbg's help
`!extension.help`	Get help on the extension

Sessions

`.tlist`	List running processes
`.attach pid`	Attach to a process
`.detach`	Detach from a process
`restart`	Stop and restart execution

Execution Control

`t (F11)`	Step into (trace)
`p (F10)`	Step over
`p count`	Step over count instructions
`gu (Shift + F11)`	Step return
`g (F5)`	Continue (go)
`pa address`	Run to address
`wt`	Watch and Trace
`wt -l depth`	Watch and Trace up to depth levels
`wt -oR`	Watch and Trace and show return values
`(Ctrl + break)`	Break

Breakpoints

`bl`	List breakpoints
`bp address`	Set a breakpoint

`bp address script`	Set a breakpoint and run script upon hitting
	
	Examples:
	`bp kernel32!CreateFileW`
	`bp 0x07004020 ".echo function called;g"`

`bc #`	Clear a breakpoint
`bc *`	Clear all breakpoints
`bd #`	Disable a breakpoint
`bd *`	Disable all breakpoints
`ba [rwe] [size] address`	Set a breakpoint on memory access, size can be 1, 2, or 4
`bm symbol`	Break on symbol access
`sxe cpr`	Break on process creation
`sxe epr`	Break on process exit
`sxe ct`	Break on thread creation
`sxe et`	Break on thread exit
`sxe ld`	Break on module load
`sxe ud`	Break on module unload

Registers

`r`	Display registers
`r reg`	Display single register
`r reg=value`	Set register value
`r reg:[iu][bwdqf]`	Display register value as type (see memory)

Loaded Modules

`lm f`	Lists all loaded modules and file paths
`!lmi module`	Gives detailed info about a module
`!dh module`	Displays headers for a module
`.imgscan`	Scan memory for modules

Symbols

`.sympath path`	Set symbols path
`ld module`	Load symbols for a module
`ld *`	Load symbols for all modules
`!sym`	Show the status of symbol loading
`.reload [/f]`	Reload symbols
`ln address`	Find nearest symbol to address

Memory

`d[aubwdqfD] [/c #] address`	Display memory
	
	a = ascii
	u = Unicode
	b = byte
	w = word
	d= dword
	q = qword
	f = float
	D = double
	
	Examples:
	`dd 0x07004020`
	`da 0x04000100`

`d[sS] [/s width] address`	Display String
`dl address # size`	Display linked list

`e[aubwdqfD] address value`	Edit memory
	
	Examples:
	`ed 0x04001000 0x0badf00d`
	`eb 0x01000468 0x90`
	`ea module!Msg "hello!"`

`e[za|zu] address string`	Edit string (w/ NULL)

`c range address`	Compare two memory ranges
	
	Note: a range can be
	`startAddr size`
	`startAddr endAddr`

`m range address`	Move memory
`f range pattern`	Fill memory with a pattern
`s range pattern`	Search memory for pattern
`!address address`	Display info on memory usage
`!vprot address`	Show memory protection information
`!mapped_file address`	Lists the name of that file that is mapped into that address (if any)
`!heap`	Show info about heap memory usage
`.writemem file range`	Dump memory range to file

Crash Analysis

`!analyze -v`	Display detailed info about crash
`!error #`	Display error message for error
`!cppexr`	Display info about C++ exception
`!gle`	Displays the last error
`.lastevent`	Displays the most recent exception
`vertarget`	Show target computer information
`.effmach`	Displays processor mode

Callstack

`k n`	Display the callstack
`kd`	Display raw stack
`.frame #`	Jump back to frame number
`!findstack symbol`	Locates all stacks that reference the symbol

Kernel Debugging Commands

`!process 0 0`	Lists all processes running
`!process proc`	Displays details about a process
`.process proc`	Sets the current process context
`.thread thread`	Sets the current thread context
`!object \driver`	Lists all objects in the driver
`!drvobj driverObj`	Displays details about a DRIVER_OBJECT
`!devobj deviceObj`	Displays details about a DEVICE_OBJECT
`!devstack deviceObj`	Displays the device stack for the device object
`dds poi(nt!KeServiceDescriptorTable) L120`	Displays the System Service Dispatch Table (SSDT)

BLUE COAT®