

.NET in WinDbg cheatsheet

by Sebastian Solnica
www.lowleveldesign.org

WinDbg extensions for .NET

- SOS (comes with .NET), help: `!sos.help`
- SOSEX (<http://www.stevetechspot.com/default.aspx>), help: `!sosexhelp`
- netext (<http://netext.codeplex.com/>), help: `.browse !whelp`
- Check also my debug recipes:
<https://github.com/lowleveldesign/debug-recipes>

Windbg start

`windbg app.exe`
 (start a new process)
`windbg -pn app.exe`
 (find process by name and attach to it)
`windbg -p <process-pid>`
 (find process by pid and attach to it)

General commands

`.cls` (clear the screen)
`.formats 0x41` (display a number in various formats)
`? 20+10` (evaluate an expression)
`.hh <command>` (help)

Execution & stack native commands

`t` (step into), `p` (step over), `g` (go)
`u .` (disassembly), `uf .` (disassembly whole function)
`k` (stack), `kb` (stack + 3 args), `kd` (raw stack)
`bp` (place breakpoint)

Native memory commands

`!address -summary`
`d{a|b|c|w|d|q|p}` (display memory)
`e{a|b|w|d|q|p}` (edit memory)

Modules and symbols commands

`!mv[m]` (list modules)
`!d` (load module symbols), `.reload /all` (reload all symbols)
`.sympath[+]` (set symbols path), `.symfix` (use MS symbol servers)

Managed apps in WinDbg

`sxe !d:clrjit` (wait for clr to load)
`.loadby sos clr` (local machine)
`.load netext; !wver` (loads SOS automatically)
`.load sosex`

`.chain` (list extensions)

Metadata: appdomains and assemblies

`!DumpDomain` and `!lwdomain` (appdomains info)
`!DumpAssembly <asmAddr>` (dump assembly info)
`!DumpModule -mt <moduleAddr>` (dump module info)

Metadata: types

!Name2EE <Module> <TypeName> or
!mx <TypeName*> (find type address)

!DumpClass <EEClass> (dump EEClass info) or
!DumpMT -md <MT> or **!mdt** (dump type or object info)

Metadata: methods

!DumpMD <MD> (method description info)
!DumpIL <MD> (show IL)
!U <code address> or **!mu <code address>**
 or **!muf <MD|code address>** (show assembly and IL code)

Object instances

!DumpObject <address> (!do) or
!wdo <address> or
!mdt <address> (dump object info)
!DumpVC <MT> <address> (dump value object info)
!wselect (select properties of an object or items in an array)

Managed stack

!ClrStack -a or **!mk -a** or **!wclrstack** (print call stack)
!wstack or **!DumpStackObjects (!dso)** or
!mdso (dump object references on stack)
!ClrStack -gc (object references on stack with call stacks)
~*e!mk -a (print call stacks for all the threads)

Exceptions

!sxe clr (enable notifications for managed exceptions)
!PrintException (!pe) or **!wpe** (dump the exception info for the current thread)
!Threads or **!wthreads** (list threads with exception info)
!wdae (dump all exceptions on the heap)

Threads and locks

!Threads or
!wthreads (list threads with exception info)
!ThreadPool (thread pool info)

Locks

!SyncBlk (list Sync Blocks acquired on objects)
!mwaits (list all waiting threads), **!dlk** (list deadlocks between SyncBlocks or RWlocks)
!mlocks (list all managed locks), **!rwlock** (list all RWlocks)
!DumpHeap -thinlock (show thin locks acquired on objects from the heap)

Breakpoints

!bpmd -md <MD> or **!mbm <method>** (set breakpoint on a method)
!mbi, !mbe, !mbd, !mbc (managed breakpoints control)
!mbp <source-file> <line> (set breakpoint at the source code)

General heap information

!EEHeap [-loader] [-gc] (show details about GC and loader heaps)
!EEVersion (show the mode in which the GC runs)

GC heap (SOS)

!DumpHeap [-stat] [-mt] [-type] (list object on the heap)
!TraverseHeap <file> (dump heap graph for CLR Profiler)
!ObjSize <address> (total size of an object)
!GCWhere <address> (GC gen in which object exists)

GC heap (SOSEX)

!bhi [file] (build the heap index)
!lhi [file] (load the heap index)
!dumpgen <genNum> [-type <typeName> [-stat]] (dump objects in a given generation)
!gcgen <address> (GC gen in which object exists)

GC heap (netext)

!windex -save <file> (build the heap index)
!windex -load <file> (load the heap index)
!windex [-type <type*>] -implement <type*> [-mt <MT>] (load the heap index)
!wfrom (very powerful heap querying, check !whelp)

Object GC roots

!GCRoot [-all] <address> or
!mroot [-all] <address> (list object roots)
!refs [-target|-source] <address> (display all references from and to the object)

GC Handles

!GCHandles (show statistics of used GC handles)
!GCHandleLeaks (track down GC handle leaks)
!gch [HandleType] ... [-stat] (show information about GC handles)
!wgchandle (list GC handles)

Finalization queue

!FinalizeQueue [-detail] (display objects on the Finalization queue)
!finq [GenNum] [-stat] (display objects on the Finalization queue)
!frq [-stat] (display objects on the Freachable queue)

Misc commands (netext)

!wdict (display dictionary objects)
!whash (display HashTable objects)
!whttp (list HttpContext objects)
!wconfig (show all .config file lines in memory)
!wservice (list WCF service objects)

Misc commands (netext)

!wkeyvalue (display pair key/value for NameObjectCollection type objects)
!wcookie (display HTTP cookies)
!wruntime (display HTTP Runtime Info including Active Requests)
!wtoken (display WIF tokens and cookies)