**Debugging .NET with WinDbg**

Author: Sebastian Solnica (<http://lowleveldesign.wordpress.com>)

Version: 1.0beta2

[Starting debugging process](https://docs.google.com/document/d/1yMQ8NAQZEBtsfVp7AsFLSA_MkIKlYNuSowG72_nU0ek/edit?pli=1#heading=h.eeebb757dddd)

[In .NET2.0](https://docs.google.com/document/d/1yMQ8NAQZEBtsfVp7AsFLSA_MkIKlYNuSowG72_nU0ek/edit?pli=1#heading=h.b0a1bd9690ee)

[In .NET4.0](https://docs.google.com/document/d/1yMQ8NAQZEBtsfVp7AsFLSA_MkIKlYNuSowG72_nU0ek/edit?pli=1#heading=h.ec2b6a8a6367)

[Controlling debugging process](https://docs.google.com/document/d/1yMQ8NAQZEBtsfVp7AsFLSA_MkIKlYNuSowG72_nU0ek/edit?pli=1#heading=h.956119af011c)

[Setting breakpoint](https://docs.google.com/document/d/1yMQ8NAQZEBtsfVp7AsFLSA_MkIKlYNuSowG72_nU0ek/edit?pli=1#heading=h.087282f3698a)

[!mbm](https://docs.google.com/document/d/1yMQ8NAQZEBtsfVp7AsFLSA_MkIKlYNuSowG72_nU0ek/edit?pli=1#heading=h.dc4516598e9c)

[!mbp](https://docs.google.com/document/d/1yMQ8NAQZEBtsfVp7AsFLSA_MkIKlYNuSowG72_nU0ek/edit?pli=1#heading=h.229af95f846f)

[!bpmd](https://docs.google.com/document/d/1yMQ8NAQZEBtsfVp7AsFLSA_MkIKlYNuSowG72_nU0ek/edit?pli=1#heading=h.b45898ce5c73)

[Examining stack](https://docs.google.com/document/d/1yMQ8NAQZEBtsfVp7AsFLSA_MkIKlYNuSowG72_nU0ek/edit?pli=1#heading=h.bc947125f80d)

[!CLRStack](https://docs.google.com/document/d/1yMQ8NAQZEBtsfVp7AsFLSA_MkIKlYNuSowG72_nU0ek/edit?pli=1#heading=h.61c4a3948408)

[!DumpStack](https://docs.google.com/document/d/1yMQ8NAQZEBtsfVp7AsFLSA_MkIKlYNuSowG72_nU0ek/edit?pli=1#heading=h.1793733b966f)

[!DumpStackObjects](https://docs.google.com/document/d/1yMQ8NAQZEBtsfVp7AsFLSA_MkIKlYNuSowG72_nU0ek/edit?pli=1#heading=h.95fb67167d7d)

[!EEStack](https://docs.google.com/document/d/1yMQ8NAQZEBtsfVp7AsFLSA_MkIKlYNuSowG72_nU0ek/edit?pli=1#heading=h.5a5cc319b6d5)

[Examining heap & Garbage Collector status](https://docs.google.com/document/d/1yMQ8NAQZEBtsfVp7AsFLSA_MkIKlYNuSowG72_nU0ek/edit?pli=1#heading=h.89f41fd582e3)

[!DumpHeap](https://docs.google.com/document/d/1yMQ8NAQZEBtsfVp7AsFLSA_MkIKlYNuSowG72_nU0ek/edit?pli=1#heading=h.1e93fde5d255)

[!DumpRuntimeTypes](https://docs.google.com/document/d/1yMQ8NAQZEBtsfVp7AsFLSA_MkIKlYNuSowG72_nU0ek/edit?pli=1#heading=h.9652acd16fa1)

[!EEHeap](https://docs.google.com/document/d/1yMQ8NAQZEBtsfVp7AsFLSA_MkIKlYNuSowG72_nU0ek/edit?pli=1#heading=h.fa3ccdcd6e34)

[!HeapStat](https://docs.google.com/document/d/1yMQ8NAQZEBtsfVp7AsFLSA_MkIKlYNuSowG72_nU0ek/edit?pli=1#heading=h.8b1ea37dcecd)

[!TraverseHeap](https://docs.google.com/document/d/1yMQ8NAQZEBtsfVp7AsFLSA_MkIKlYNuSowG72_nU0ek/edit?pli=1#heading=h.ed913496606c)

[!VerifyHeap](https://docs.google.com/document/d/1yMQ8NAQZEBtsfVp7AsFLSA_MkIKlYNuSowG72_nU0ek/edit?pli=1#heading=h.01fbc6972fa9)

[!GCHandles](https://docs.google.com/document/d/1yMQ8NAQZEBtsfVp7AsFLSA_MkIKlYNuSowG72_nU0ek/edit?pli=1#heading=h.1aa6d9eba6d7)

[!GCHandleLeaks](https://docs.google.com/document/d/1yMQ8NAQZEBtsfVp7AsFLSA_MkIKlYNuSowG72_nU0ek/edit?pli=1#heading=h.854f10662f28)

[!GCInfo](https://docs.google.com/document/d/1yMQ8NAQZEBtsfVp7AsFLSA_MkIKlYNuSowG72_nU0ek/edit?pli=1#heading=h.46d7bdd31a86)

[!GCRoot](https://docs.google.com/document/d/1yMQ8NAQZEBtsfVp7AsFLSA_MkIKlYNuSowG72_nU0ek/edit?pli=1#heading=h.1a32f7c10905)

[!GCWhere](https://docs.google.com/document/d/1yMQ8NAQZEBtsfVp7AsFLSA_MkIKlYNuSowG72_nU0ek/edit?pli=1#heading=h.7ebd0df1109b)

[!FindRoots](https://docs.google.com/document/d/1yMQ8NAQZEBtsfVp7AsFLSA_MkIKlYNuSowG72_nU0ek/edit?pli=1#heading=h.b25258d736d4)

[Examining code](https://docs.google.com/document/d/1yMQ8NAQZEBtsfVp7AsFLSA_MkIKlYNuSowG72_nU0ek/edit?pli=1#heading=h.91e7777c5411)

[!Name2EE](https://docs.google.com/document/d/1yMQ8NAQZEBtsfVp7AsFLSA_MkIKlYNuSowG72_nU0ek/edit?pli=1#heading=h.6ae0291fc908)

[!U](https://docs.google.com/document/d/1yMQ8NAQZEBtsfVp7AsFLSA_MkIKlYNuSowG72_nU0ek/edit?pli=1#heading=h.de988a86e820)

[!DumpIL](https://docs.google.com/document/d/1yMQ8NAQZEBtsfVp7AsFLSA_MkIKlYNuSowG72_nU0ek/edit?pli=1#heading=h.9b59473b77ad)

[Examining CLR structures](https://docs.google.com/document/d/1yMQ8NAQZEBtsfVp7AsFLSA_MkIKlYNuSowG72_nU0ek/edit?pli=1#heading=h.fb571ac30fdd)

[AppDomains](https://docs.google.com/document/d/1yMQ8NAQZEBtsfVp7AsFLSA_MkIKlYNuSowG72_nU0ek/edit?pli=1#heading=h.13571aacca40)

[!DumpDomain](https://docs.google.com/document/d/1yMQ8NAQZEBtsfVp7AsFLSA_MkIKlYNuSowG72_nU0ek/edit?pli=1#heading=h.ca8429c9db81)

[Assemblies/Modules](https://docs.google.com/document/d/1yMQ8NAQZEBtsfVp7AsFLSA_MkIKlYNuSowG72_nU0ek/edit?pli=1#heading=h.64a9b8fb707a)

[!DumpModule](https://docs.google.com/document/d/1yMQ8NAQZEBtsfVp7AsFLSA_MkIKlYNuSowG72_nU0ek/edit?pli=1#heading=h.da4c49acc2c7)

[!DumpAssembly](https://docs.google.com/document/d/1yMQ8NAQZEBtsfVp7AsFLSA_MkIKlYNuSowG72_nU0ek/edit?pli=1#heading=h.5b9ffc0546cb)

[Classes/Types](https://docs.google.com/document/d/1yMQ8NAQZEBtsfVp7AsFLSA_MkIKlYNuSowG72_nU0ek/edit?pli=1#heading=h.25e970060636)

[!DumpClass](https://docs.google.com/document/d/1yMQ8NAQZEBtsfVp7AsFLSA_MkIKlYNuSowG72_nU0ek/edit?pli=1#heading=h.f0e9037f8147)

[!DumpMT](https://docs.google.com/document/d/1yMQ8NAQZEBtsfVp7AsFLSA_MkIKlYNuSowG72_nU0ek/edit?pli=1#heading=h.e6c921be72d5)

[Methods](https://docs.google.com/document/d/1yMQ8NAQZEBtsfVp7AsFLSA_MkIKlYNuSowG72_nU0ek/edit?pli=1#heading=h.76540d2292fc)

[!IP2MD](https://docs.google.com/document/d/1yMQ8NAQZEBtsfVp7AsFLSA_MkIKlYNuSowG72_nU0ek/edit?pli=1#heading=h.2c6ad8d82927)

[!DumpMD](https://docs.google.com/document/d/1yMQ8NAQZEBtsfVp7AsFLSA_MkIKlYNuSowG72_nU0ek/edit?pli=1#heading=h.38de537255ab)

[Objects](https://docs.google.com/document/d/1yMQ8NAQZEBtsfVp7AsFLSA_MkIKlYNuSowG72_nU0ek/edit?pli=1#heading=h.d39338b1eadc)

[!DumpArray](https://docs.google.com/document/d/1yMQ8NAQZEBtsfVp7AsFLSA_MkIKlYNuSowG72_nU0ek/edit?pli=1#heading=h.39125ed63313)

[!DumpObj](https://docs.google.com/document/d/1yMQ8NAQZEBtsfVp7AsFLSA_MkIKlYNuSowG72_nU0ek/edit?pli=1#heading=h.e9b0f0dc5561)

[!DumpVC](https://docs.google.com/document/d/1yMQ8NAQZEBtsfVp7AsFLSA_MkIKlYNuSowG72_nU0ek/edit?pli=1#heading=h.ce081f1be336)

[!ObjSize](https://docs.google.com/document/d/1yMQ8NAQZEBtsfVp7AsFLSA_MkIKlYNuSowG72_nU0ek/edit?pli=1#heading=h.ea4b2e465f31)

[Threads](https://docs.google.com/document/d/1yMQ8NAQZEBtsfVp7AsFLSA_MkIKlYNuSowG72_nU0ek/edit?pli=1#heading=h.8dc284476cc9)

[!SyncBlk](https://docs.google.com/document/d/1yMQ8NAQZEBtsfVp7AsFLSA_MkIKlYNuSowG72_nU0ek/edit?pli=1#heading=h.a35703065463)

[ASP.NET](https://docs.google.com/document/d/1yMQ8NAQZEBtsfVp7AsFLSA_MkIKlYNuSowG72_nU0ek/edit?pli=1#heading=h.9b0045951b12)

[!DumpHttpContext (!ASPXPages)](https://docs.google.com/document/d/1yMQ8NAQZEBtsfVp7AsFLSA_MkIKlYNuSowG72_nU0ek/edit?pli=1#heading=h.6314238ff815)

[!DumpHttpRuntime](https://docs.google.com/document/d/1yMQ8NAQZEBtsfVp7AsFLSA_MkIKlYNuSowG72_nU0ek/edit?pli=1#heading=h.4cc4666ab46a)

[!DumpBuckets](https://docs.google.com/document/d/1yMQ8NAQZEBtsfVp7AsFLSA_MkIKlYNuSowG72_nU0ek/edit?pli=1#heading=h.02cd5c1ab27f)

[!DumpASPNETCache](https://docs.google.com/document/d/1yMQ8NAQZEBtsfVp7AsFLSA_MkIKlYNuSowG72_nU0ek/edit?pli=1#heading=h.b150662a6d13)

[Exceptions](https://docs.google.com/document/d/1yMQ8NAQZEBtsfVp7AsFLSA_MkIKlYNuSowG72_nU0ek/edit?pli=1#heading=h.085afabc05d9)

[!CheckCurrentException](https://docs.google.com/document/d/1yMQ8NAQZEBtsfVp7AsFLSA_MkIKlYNuSowG72_nU0ek/edit?pli=1#heading=h.e4caf3656089)

[!CheckCurrentException](https://docs.google.com/document/d/1yMQ8NAQZEBtsfVp7AsFLSA_MkIKlYNuSowG72_nU0ek/edit?pli=1#heading=h.8cdab8e38342)

[!EHInfo](https://docs.google.com/document/d/1yMQ8NAQZEBtsfVp7AsFLSA_MkIKlYNuSowG72_nU0ek/edit?pli=1#heading=h.c18f5fca5eec)

[!PrintException](https://docs.google.com/document/d/1yMQ8NAQZEBtsfVp7AsFLSA_MkIKlYNuSowG72_nU0ek/edit?pli=1#heading=h.0ef85ff68fcb)

[!DumpAllExceptions](https://docs.google.com/document/d/1yMQ8NAQZEBtsfVp7AsFLSA_MkIKlYNuSowG72_nU0ek/edit?pli=1#heading=h.03591daef3a4)

[!StopOnException](https://docs.google.com/document/d/1yMQ8NAQZEBtsfVp7AsFLSA_MkIKlYNuSowG72_nU0ek/edit?pli=1#heading=h.53f66f1c2b27)

[](https://docs.google.com/document/d/1yMQ8NAQZEBtsfVp7AsFLSA_MkIKlYNuSowG72_nU0ek/edit?pli=1#b.60)

**Starting debugging process**

**In .NET2.0**

You need to wait till clrjit module is loaded:

sxe ld:mscorjit

Then you may load necessary extensions:

.loadby sos mscorwks  
.load sosex

**In .NET4.0**

You need to wait till clrjit module is loaded:

sxe ld:clrjit

Then you may load necessary extensions:

.loadby sos clr

.load sosex

**Controlling debugging process**

**Setting breakpoint**

**!mbm**

!mbm <Type/MethodFilter> [ILOffset] [Options] (**SOSEX**) - sets a managed breakpoint on methods matching the specified filter.

**!mbp**

!mbp <SourceFile> <nLineNum> [ColNum] [Options] (**SOSEX**) - sets a managed breakpoint at the specified source code location.

**!bpmd**

!BPMD [-nofuturemodule] [<module name> <method name>] [-md <MethodDesc>] -list-clear <pending breakpoint number> -clearall (**SOS/PSSCOR2**) - creates a breakpoint at the specified method in the specified module. And with **-clear** and **-clearall** remove them.

If you are using names for module and method !bpmd will scan all the loaded domains. If it finds that a given type was loaded more than once (in two separate domain for instance) it will print error:

0:000> !bpmd System.dll System.Diagnostics.EventLog.SourceExists  
There are multiple modules with the same name...

In that case just look for a module address (using !DumpDomain for instance or !Name2EE) and retype the above command but using module address instead of its name:

0:000> !Name2EE \* System.Diagnostics.EventLog.SourceExists  
Module: 000007fbffc31000 (mscorlib.dll)

Module: 000007fc0d741000 (System.Configuration.dll)

--------------------------------------  
Module: 000007fbff201000 (System.dll)

Token: 0x00000000060038fd

MethodDesc: 000007fbff308070

Name: System.Diagnostics.EventLog.SourceExists(System.String)

JITTED Code Address: 000007fbff996ec0

-----------------------

Token: 0x00000000060038fe

MethodDesc: 000007fbff308090

Name: System.Diagnostics.EventLog.SourceExists(System.String, System.String)

JITTED Code Address: 000007fbff996d70  
…  
Module: 000007fba9e243e0 (Castle.Core.DLL)  
--------------------------------------  
Module: **000007fba9e29368** (System.dll)

Token: 0x00000000060038fd

MethodDesc: <not loaded yet>

Name: System.Diagnostics.EventLog.SourceExists

Not JITTED yet.

-----------------------

Token: 0x00000000060038fe

MethodDesc: <not loaded yet>

Name: System.Diagnostics.EventLog.SourceExists

Not JITTED yet.

And finally:

0:000> !bpmd 000007fba9e29368  System.Diagnostics.EventLog.SourceExists

Adding pending breakpoints...

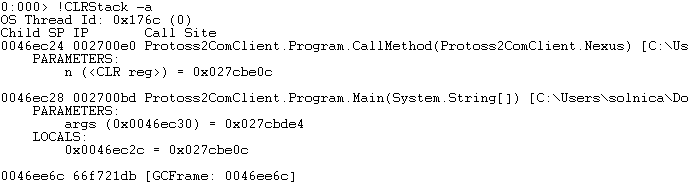
You can also use wildcards:

0:000> !bpmd mscorlib.dll System.AppDomain.CreateDomain

Found 6 methods...

**Examining stack**

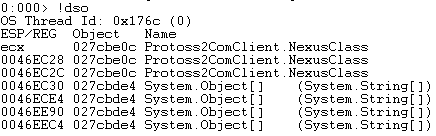
**!CLRStack**

!CLRStack [-a] [-l] [-p] [-n] (**SOS/PSSCOR2**) - provides stack trace of managed code only. The **-p** shows arguments to the managed function. The **-l** shows local function variables. **-a** is -p and -l combined. **-n** disables display of source file names and lines.

**!DumpStack**

!DumpStack [-EE] [-n] [top stack [bottom stack]] (**SOS/PSSCOR2**) - displays a stack trace (managed + native). The -EE parameter causes DumpStack to show only managed stack frames (top and bottom limits the frames displayed). -n option disables display of source line and file names. To check the top and bottom frames you may use !teb debugger extension command.

**!DumpStackObjects**

!DumpStackObjects [-verify] [top stack [bottom stack]] (**SOS/PSSCOR2**) (alias: *dso*) - displays all managed objects found within the bounds of the current stack.

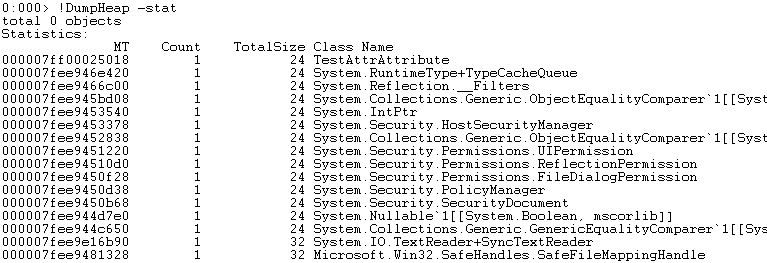
**!EEStack**

!EEStack [-short] [-EE] (**SOS/PSSCOR2**) - runs DumpStack on all threads in the process. The -EE argument is passed to the DumpStack command. When -short is given only threads that have taken the lock, threads stalled to allow garbage collection and threads that are currently in managed code are displayed.

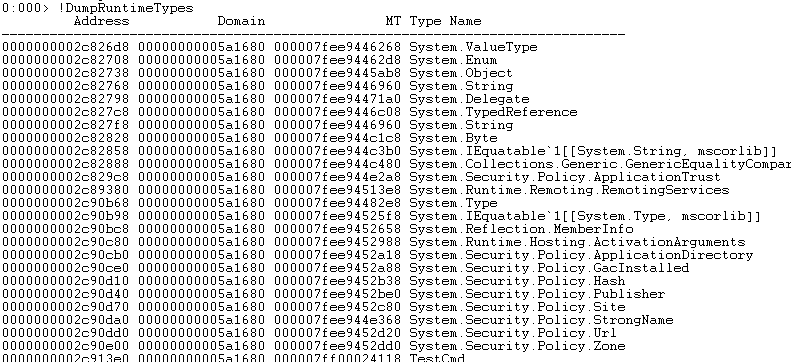
**Examining heap & Garbage Collector status**

**!DumpHeap**

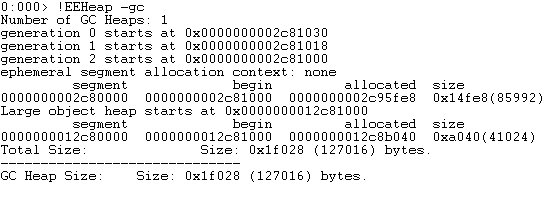
!DumpHeap [-stat] [-strings] [-short] [-min <size>] [-max <size>] [-thinlock] [-startAtLowerBound] [-mt <MethodTable address>] [-type <partial type name>][start [end]] (**SOS/PSSCOR2**) - displays information about objects that are allocated on the heap. **-type** and **-mt** switches enable you to search the managed heap for a given type name or method table address.

**-stat** displays only statistical type summary and **-strings** only strings statistical summary. **-short** limits output to just addresses of objects which might be useful when redirecting the output to other debugger command.

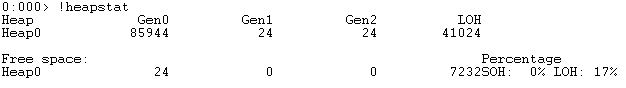
**!DumpRuntimeTypes**

!DumpRuntimeTypes (**SOS/PSSCOR2**) - displays runtime type objects in the GC heap and lists type names and method tables associated with them.

**!EEHeap**

!EEHeap [-gc] [-loader] (**SOS/PSSCOR2**) - display information about process memory consumed by internal CLR structures. **-gc** limits the output to Garbage Collector data and **-loader** to loader data structures.

**!HeapStat**

!HeapStat [-inclUnrooted | -iu] (**SOS/PSSCOR2**) - displays the generation sizes and the total of the free space for each heap.

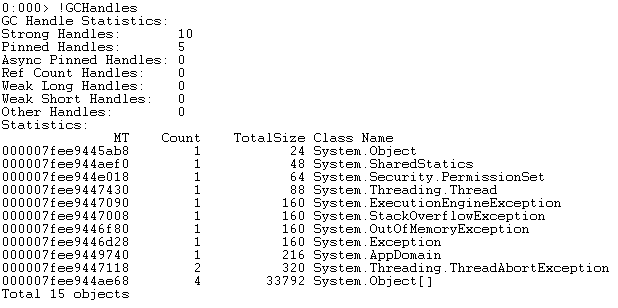
**!TraverseHeap**

!TraverseHeap [-xml] <filename> (**SOS/PSSCOR2**) - traverses the heap and saves its data to the file understandable by CLR Profiler. With **-xml** option the output will be saved as a xml file.

**!VerifyHeap**

!VerifyHeap (**SOS/PSSCOR2**) - checks the GC heap for signs of corruptions and displays all errors found.

**!GCHandles**

!GCHandles [-perdomain] (**SOS/PSSCOR2**) - displays statistics about GC handles in the process.

**!GCHandleLeaks**

!GCHandleLeaks [-perdomain] (**SOS/PSSCOR2**) - looks for any references to strong and pinned GC handles in the process and displays the results.

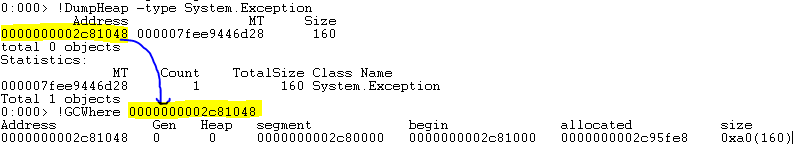
**!GCInfo**

!GCInfo <MethodDesc address><Code address> (**SOS/PSSCOR2**) - Displays data that indicates when registers or stack locations contain managed objects. If a garbage collection occurs, the collector must know the locations of references to objects so it can update them with new object pointer values.

**!GCRoot**

!GCRoot [-nostacks] <Object address> (**SOS/PSSCOR2**) - displays information about references to an object at the specified address.

**!GCWhere**

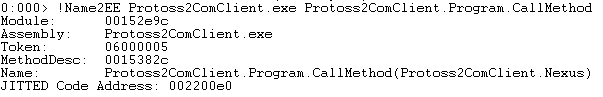
!GCWhere <object address>  (**SOS/PSSCOR2**) - displays the size and location in the GC heap for the given argument.

**!FindRoots**

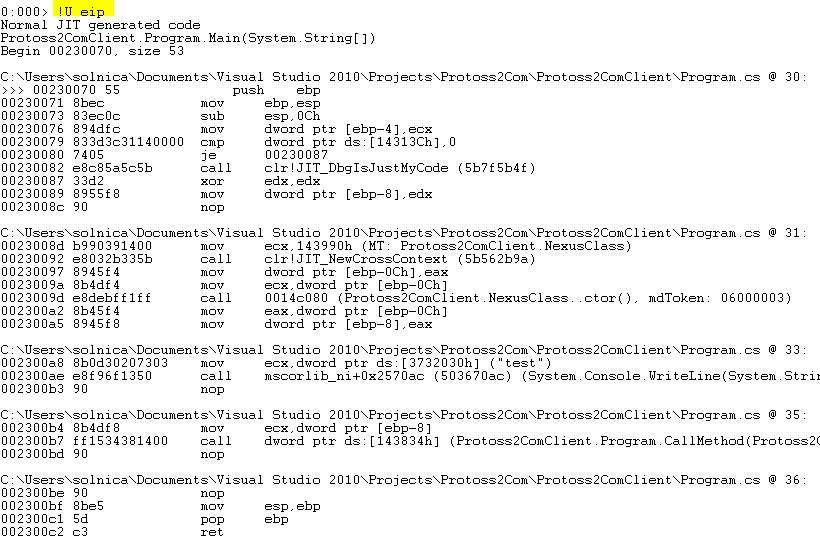
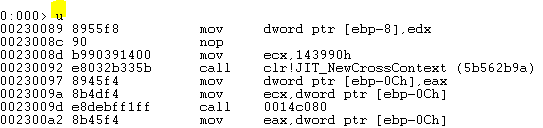
!FindRoots -gen <N> | -gen any |<object address> (**SOS/PSSCOR2**) - causes the debugger to break into the debuggee on the next collection of the specified generation.

**Examining code**

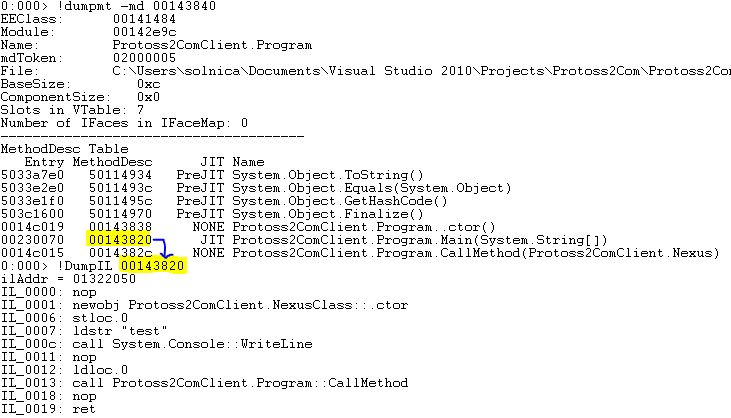
**!Name2EE**

!Name2EE <module name> <type or method name> (**SOS/PSSCOR2**) - displays method table and EEClass for the specified type or method in the given module.The module must be loaded. You may pass \* as the module parameter and all modules will be searched for a given method/type.

**!U**

!U [-gcinfo] [-ehinfo] [-n] <MethodDesc address> | <Code address> (**SOS/PSSCOR2**) - displays an annotated disassembly of a managed method. Whole method code is shown. -n disables source code information. !U has advantage over u command because it can decipher the method called by CLR (as in the example below).

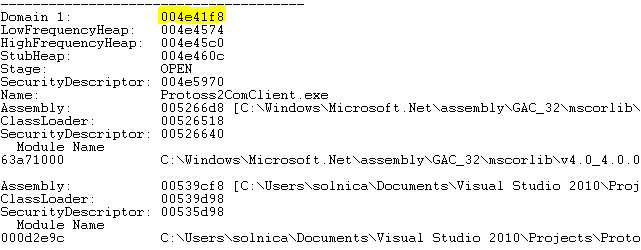
**!DumpIL**

!DumpIL <Managed DynamicMethod object> | <DynamicMethodDesc pointer> | <MethodDesc pointer>  (**SOS/PSSCOR2**) - displays the MSIL that is associated with a managed method.

**Examining CLR structures**

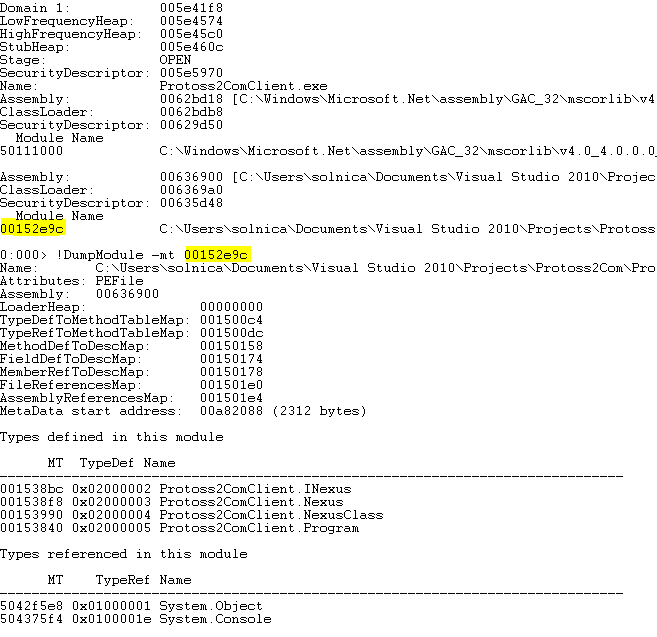
**AppDomains**

**!DumpDomain**

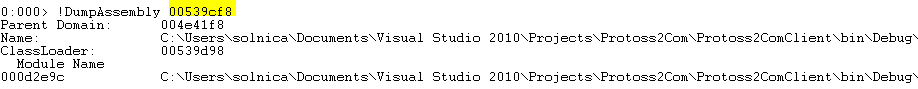
!DumpDomain [<domain address>] (**SOS/PSSCOR2**) - enumerates all appdomains loaded in the process or show information about only an appdomain at the specified address. It lists all the assemblies loaded into the appdomain. To get appdomain address you may also use the !Threads command.

**Assemblies/Modules**

**!DumpModule**

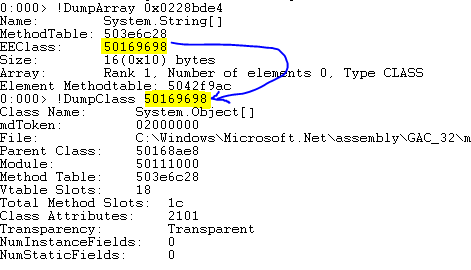
!DumpModule [-mt] <Module address> (**SOS/PSSCOR2**) - dumps information about module. The **-mt** option displays types exported by the module and types that are referenced by it.

**!DumpAssembly**

!DumpAssembly <assembly address> (**SOS/PSSCOR2**) - dumps information about assembly including all its modules. You may get the assembly address from DumpDomain command.

**Classes/Types**

**!DumpClass**

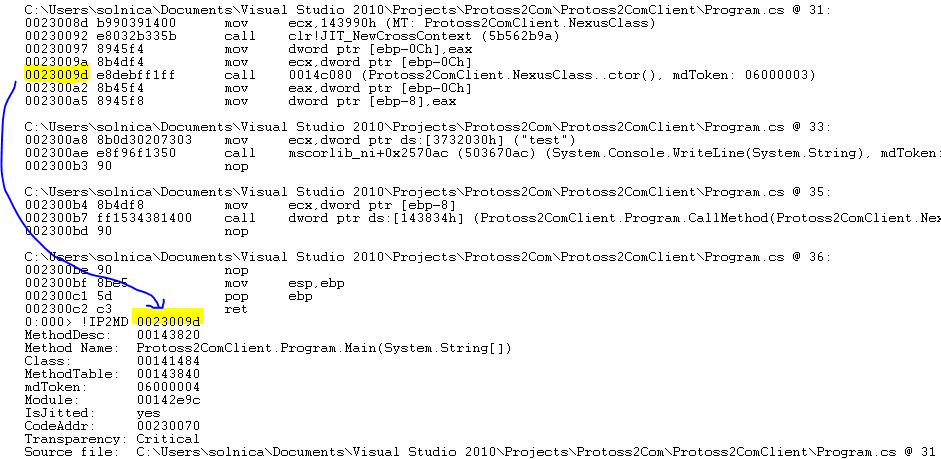
!DumpClass <EEClass address> (**SOS/PSSCOR2**) - displays information about EEClass structure associated with a type.

**!DumpMT**

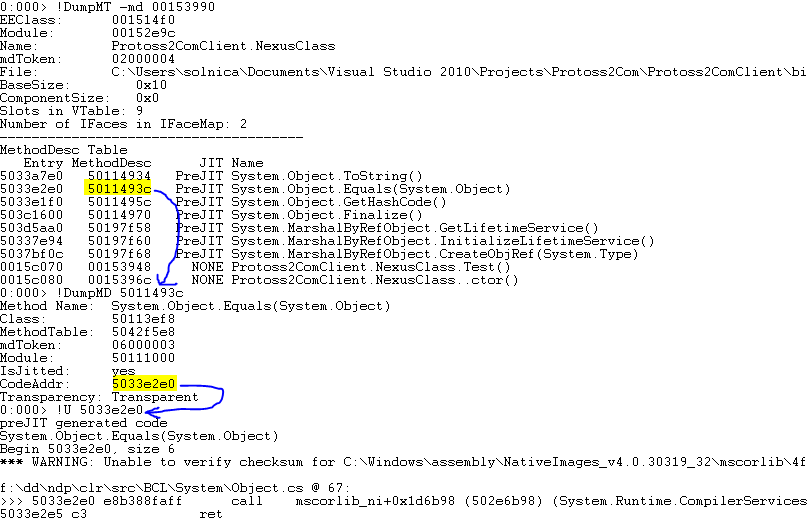
!DumpMT [-MD] <MethodTable address> (**SOS/PSSCOR2**) - displays information about a method table at the given address. With **-md** specified it will also list all methods for the given type.

**Methods**

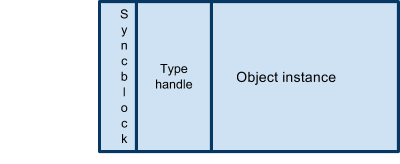
**!IP2MD**

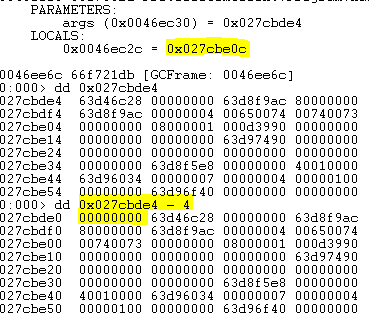
!IP2MD <Code address> (**SOS/PSSCOR2**) - displays a method description for a given code address.

**!DumpMD**

!DumpMD <MethodDesc address> (**SOS/PSSCOR2**) - dumps method description. m\_CodeOrIL if method is jited points to the assembly code that was generated for this method.

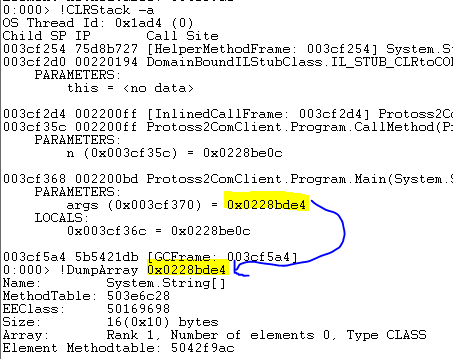
**Objects**

Each managed object located on the heap contains following pieces of information:

The object pointer always points to the type handle so to examine the sync block you need to dump memory -4 bytes back, eg.

**!DumpArray**

DumpArray [-start <startIndex>] [-length <length>] [-details] [-nofields] <array object address> (**SOS/PSSCOR2**) - shows elements of the array. -start specifies at which index to start, -length how many elements should be displayed and -details call DumpObj or DumpVC for every element.

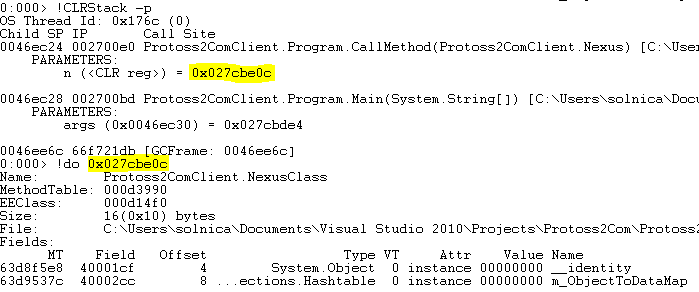


**!DumpObj**

!DumpObj [-nofields] <object address> (**SOS/PSSCOR2**) (alias: *do*) - dumps information about a reference type. The address might be taken from, for example !ClrStack -a. In addition to the type information we receive the fields offsets and description.

The fields table contains following columns:

* **MT** - Method Table
* **Field** - address of the field on the heap. You may use it to dump fields of the object being examined (works only for reference types, for value types use !dumpvc)
* **Offset** - offset of the field at the object
* **Type** - type name
* **VT** - information whether object is a value type
* **Attr** - attributes of the field
* **Name** - name of the field if available



**!DumpVC**

!DumpVC <MethodTable address> <Address> (**SOS/PSSCOR2**) - displays information about fields of value type at the specified address. You may get address of the field by calling DumpObj command.

**!ObjSize**

!ObjSize [<Object address>] | [-aggregate] [-stat] (**SOS/PSSCOR2**) - FIXME

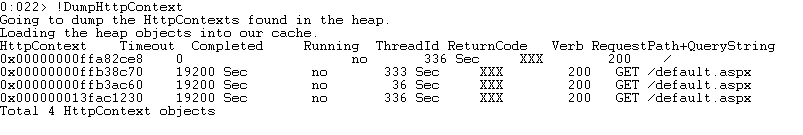
**Threads**

**!SyncBlk**

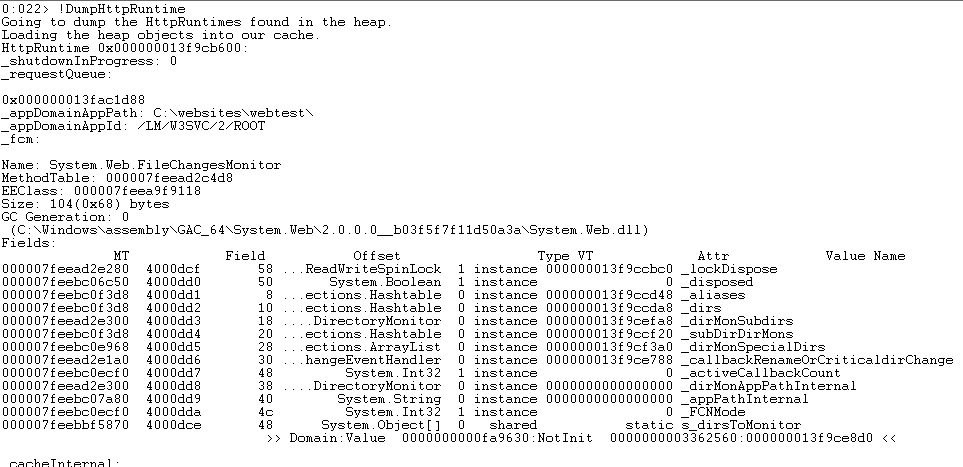
!SyncBlk [-all | <syncblk number>] (**SOS/PSSCOR2**) - displays the specified sync block structure or all SyncBlk structures corresponding to objects that are owned by a thread.

**ASP.NET**

**!DumpHttpContext (!ASPXPages)**

!DumpHttpContext (**PSSCOR2**) - dumps the HttpContexts in the heap. It shows the status of the request and the return code, etc.

**!DumpHttpRuntime**

!DumpHttpRuntime [-r] (**PSSCOR2**) - displays HttpRuntime objects and prints out some of their most common properties.

**!DumpBuckets**

!DumpBuckets (**PSSCOR2**) - dumps entire request table buckets.

**!DumpASPNETCache**

!DumpASPNETCache [-short] [-stat] [-s] (**PSSCOR2**) (alias: *dac*) - displays objects in the ASP.NET cache.

**Exceptions**

**!CheckCurrentException**

!CheckCurrentException <exception type> <pseudo register number>   (**PSSCOR2**) - checks if the current exception is the one specified and  stores 1 or 0 in the pseudo register supplied (1 = $t1 register).

**!CheckCurrentException**

!CurrentExceptionName (**PSSCOR2**) - prints out the name of the managed exception on the current stack.

**!EHInfo**

!EHInfo [<MethodDesc address>] [<Code address>] (**SOS**) - displays the exception handling blocks in a specified method.

**!PrintException**

!PrintException [-nested] [-lines] [<Exception object address>] (**SOS**) - display fields of any object derived from System.Exception. If no address is specified it displays last exception thrown on the current thread. With **-nested** it displays information about nested exception and with **-lines** source code information if available.

**!DumpAllExceptions**

!DumpAllExceptions [-v] (**PSSCOR2**) (alias: *dae*) - goes through the entire managed heap and finds any objects that inherit from System.Exception. Without **-v** will print only one exception per type with its count.

**!StopOnException**

!StopOnException [-derived] [-create | -create2] <Exception> <Pseudo-register number> (**SOS/PSSCOR2**) - forces debugger to stop when an exception is thrown. Exception is the name of .net exception time. With the **-derived** option you may stop on all exception thrown by CLR, ex. ***!StopOnException -derived System.Exception***