Advanced .NET debugging

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

Version: 1.0beta2

Starting debugging process

In .NET2.0

TODO

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. With **-list** you may view a list of breakpoints. And with **-clear** and **-clearall** remove them.

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.

```
0:000> !CLRStack -a
OS Thread Id: 0x176c (0)
Child SP IP       Call Site
0046ec24 002700e0 Protoss2ComClient.Program.CallMethod(Protoss2ComClient.Nexus) [C:\Us
    PARAMETERS:
        n (<CLR reg>) = 0x027cbe0c

0046ec28 002700bd Protoss2ComClient.Program.Main(System.String[]) [C:\Users\solnica\Do
    PARAMETERS:
        args (0x0046ec30) = 0x027cbde4
    LOCALS:
        0x0046ec2c = 0x027cbe0c

0046ee6c 66f721db [GCFrame: 0046ee6c]
```

!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.

```
0:000> !dso
OS Thread Id: 0x176c (0)
ESP/REG  Object   Name
ecx      027cbe0c Protoss2ComClient.NexusClass
0046EC28 027cbe0c Protoss2ComClient.NexusClass
0046EC2C 027cbe0c Protoss2ComClient.NexusClass
0046EC30 027cbde4 System.Object[]    (System.String[])
0046ECE4 027cbde4 System.Object[]    (System.String[])
0046EE90 027cbde4 System.Object[]    (System.String[])
0046EEC4 027cbde4 System.Object[]    (System.String[])
```

!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.

```
0:000> !DumpHeap -stat
total 0 objects
Statistics:
              MT    Count    TotalSize Class Name
000007ff00025018        1           24 TestAttrAttribute
000007fee946e420        1           24 System.RuntimeType+TypeCacheQueue
000007fee9466c00        1           24 System.Reflection.__Filters
000007fee945bd08        1           24 System.Collections.Generic.ObjectEqualityComparer`1[[Syst
000007fee9453540        1           24 System.IntPtr
000007fee9453378        1           24 System.Security.HostSecurityManager
000007fee9452838        1           24 System.Collections.Generic.ObjectEqualityComparer`1[[Syst
000007fee9451220        1           24 System.Security.Permissions.UIPermission
000007fee94510d0        1           24 System.Security.Permissions.ReflectionPermission
000007fee9450f28        1           24 System.Security.Permissions.FileDialogPermission
000007fee9450d38        1           24 System.Security.PolicyManager
000007fee9450b68        1           24 System.Security.SecurityDocument
000007fee944d7e0        1           24 System.Nullable`1[[System.Boolean, mscorlib]]
000007fee944c650        1           24 System.Collections.Generic.GenericEqualityComparer`1[[Sys
000007fee9e16b90        1           32 System.IO.TextReader+SyncTextReader
000007fee9481328        1           32 Microsoft.Win32.SafeHandles.SafeFileMappingHandle
```

!DumpRuntimeTypes

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

```
0:000> !DumpRuntimeTypes
          Address           Domain               MT Type Name
------------------------------------------------------------------------------
0000000002c826d8 00000000005a1680 000007fee9446268 System.ValueType
0000000002c82708 00000000005a1680 000007fee94462d8 System.Enum
0000000002c82738 00000000005a1680 000007fee9445ab8 System.Object
0000000002c82768 00000000005a1680 000007fee9446960 System.String
0000000002c82798 00000000005a1680 000007fee94471a0 System.Delegate
0000000002c827c8 00000000005a1680 000007fee9446c08 System.TypedReference
0000000002c827f8 00000000005a1680 000007fee9446960 System.String
0000000002c82828 00000000005a1680 000007fee944c1c8 System.Byte
0000000002c82858 00000000005a1680 000007fee944c3b0 System.IEquatable`1[[System.String, mscorlib]]
0000000002c82888 00000000005a1680 000007fee944c480 System.Collections.Generic.GenericEqualityCompar
0000000002c829c8 00000000005a1680 000007fee944e2a8 System.Security.Policy.ApplicationTrust
0000000002c89380 00000000005a1680 000007fee94513e8 System.Runtime.Remoting.RemotingServices
0000000002c90b68 00000000005a1680 000007fee94482e8 System.Type
0000000002c90b98 00000000005a1680 000007fee94525f8 System.IEquatable`1[[System.Type, mscorlib]]
0000000002c90bc8 00000000005a1680 000007fee9452658 System.Reflection.MemberInfo
0000000002c90c80 00000000005a1680 000007fee9452988 System.Runtime.Hosting.ActivationArguments
0000000002c90cb0 00000000005a1680 000007fee9452a18 System.Security.Policy.ApplicationDirectory
0000000002c90ce0 00000000005a1680 000007fee9452a88 System.Security.Policy.GacInstalled
0000000002c90d10 00000000005a1680 000007fee9452b38 System.Security.Policy.Hash
0000000002c90d40 00000000005a1680 000007fee9452be0 System.Security.Policy.Publisher
0000000002c90d70 00000000005a1680 000007fee9452c80 System.Security.Policy.Site
0000000002c90da0 00000000005a1680 000007fee944e368 System.Security.Policy.StrongName
0000000002c90dd0 00000000005a1680 000007fee9452d20 System.Security.Policy.Url
0000000002c90e00 00000000005a1680 000007fee9452dd0 System.Security.Policy.Zone
0000000002c913e0 00000000005a1680 000007ff00024118 TestCmd
```

!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.

```
0:000> !EEHeap -gc
Number of GC Heaps: 1
generation 0 starts at 0x0000000002c81030
generation 1 starts at 0x0000000002c81018
generation 2 starts at 0x0000000002c81000
ephemeral segment allocation context: none
         segment             begin         allocated  size
0000000002c80000  0000000002c81000  0000000002c95fe8  0x14fe8(85992)
Large object heap starts at 0x0000000012c81000
         segment             begin         allocated  size
0000000012c80000  0000000012c81000  0000000012c8b040  0xa040(41024)
Total Size:              Size: 0x1f028 (127016) bytes.
------------------------------
GC Heap Size:    Size: 0x1f028 (127016) bytes.
```

!HeapStat

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

```
0:000> !heapstat
Heap             Gen0         Gen1         Gen2          LOH
Heap0           85944           24           24        41024

Free space:                                                  Percentage
Heap0              24            0            0         7232SOH:  0% LOH: 17%
```

!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.

```
0:000> !GCHandles
GC Handle Statistics:
Strong Handles:       10
Pinned Handles:       5
Async Pinned Handles: 0
Ref Count Handles:    0
Weak Long Handles:    0
Weak Short Handles:   0
Other Handles:        0
Statistics:
              MT    Count    TotalSize Class Name
000007fee9445ab8        1           24 System.Object
000007fee944aef0        1           48 System.SharedStatics
000007fee944e018        1           64 System.Security.PermissionSet
000007fee9447430        1           88 System.Threading.Thread
000007fee9447090        1          160 System.ExecutionEngineException
000007fee9447008        1          160 System.StackOverflowException
000007fee9446f80        1          160 System.OutOfMemoryException
000007fee9446d28        1          160 System.Exception
000007fee9449740        1          216 System.AppDomain
000007fee9447118        2          320 System.Threading.ThreadAbortException
000007fee944ae68        4        33792 System.Object[]
Total 15 objects
```

!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.

```
0:000> !DumpHeap -type System.Exception
         Address               MT     Size
0000000002c81048 000007fee9446d28      160
total 0 objects
Statistics:
              MT    Count    TotalSize Class Name
000007fee9446d28        1          160 System.Exception
Total 1 objects
0:000> !GCWhere 0000000002c81048
Address            Gen   Heap   segment            begin              allocated          size
0000000002c81048   0       0    0000000002c80000   0000000002c81000   0000000002c95fe8   0xa0(160)
0:000> !GCRoot 0000000002c81048
Note: Roots found on stacks may be false positives. Run "!help gcroot" for
more info.
Scan Thread 0 OSTHread 1228
Scan Thread 2 OSTHread 11c4
DOMAIN(00000000005A1680):HANDLE(Strong):5213e0:Root:  0000000002c81048(System.Exception)
```

!GCWhere

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

```
0:000> !DumpHeap -type System.Exception
         Address               MT     Size
0000000002c81048 000007fee9446d28      160
total 0 objects
Statistics:
              MT    Count    TotalSize Class Name
000007fee9446d28        1          160 System.Exception
Total 1 objects
0:000> !GCWhere 0000000002c81048
Address           Gen  Heap  segment           begin             allocated         size
0000000002c81048  0    0     0000000002c80000  0000000002c81000  0000000002c95fe8  0xa0(160)
```

!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.

```
0:000> !Name2EE Protoss2ComClient.exe Protoss2ComClient.Program.CallMethod
Module:      00152e9c
Assembly:    Protoss2ComClient.exe
Token:       06000005
MethodDesc:  0015382c
Name:        Protoss2ComClient.Program.CallMethod(Protoss2ComClient.Nexus)
JITTED Code Address: 002200e0
```

!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).

```
0:000> u
00230089 8955f8          mov     dword ptr [ebp-8],edx
0023008c 90              nop
0023008d b990391400      mov     ecx,143990h
00230092 e8032b335b      call    clr!JIT_NewCrossContext (5b562b9a)
00230097 8945f4          mov     dword ptr [ebp-0Ch],eax
0023009a 8b4df4          mov     ecx,dword ptr [ebp-0Ch]
0023009d e8debff1ff      call    0014c080
002300a2 8b45f4          mov     eax,dword ptr [ebp-0Ch]
```

```
0:000> !U eip
Normal JIT generated code
Protoss2ComClient.Program.Main(System.String[])
Begin 00230070, size 53

C:\Users\solnica\Documents\Visual Studio 2010\Projects\Protoss2Com\Protoss2ComClient\Program.cs @ 30:
>>> 00230070 55              push    ebp
00230071 8bec            mov     ebp,esp
00230073 83ec0c          sub     esp,0Ch
00230076 894dfc          mov     dword ptr [ebp-4],ecx
00230079 833d3c31140000  cmp     dword ptr ds:[14313Ch],0
00230080 7405            je      00230087
00230082 e8c85a5c5b      call    clr!JIT_DbgIsJustMyCode (5b7f5b4f)
00230087 33d2            xor     edx,edx
00230089 8955f8          mov     dword ptr [ebp-8],edx
0023008c 90              nop

C:\Users\solnica\Documents\Visual Studio 2010\Projects\Protoss2Com\Protoss2ComClient\Program.cs @ 31:
0023008d b990391400      mov     ecx,143990h (MT: Protoss2ComClient.NexusClass)
00230092 e8032b335b      call    clr!JIT_NewCrossContext (5b562b9a)
00230097 8945f4          mov     dword ptr [ebp-0Ch],eax
0023009a 8b4df4          mov     ecx,dword ptr [ebp-0Ch]
0023009d e8debff1ff      call    0014c080 (Protoss2ComClient.NexusClass..ctor(), mdToken: 06000003)
002300a2 8b45f4          mov     eax,dword ptr [ebp-0Ch]
002300a5 8945f8          mov     dword ptr [ebp-8],eax

C:\Users\solnica\Documents\Visual Studio 2010\Projects\Protoss2Com\Protoss2ComClient\Program.cs @ 33:
002300a8 8b0d30207303    mov     ecx,dword ptr ds:[3732030h] ("test")
002300ae e8f96f1350      call    mscorlib_ni+0x2570ac (503670ac) (System.Console.WriteLine(System.Strir
002300b3 90              nop

C:\Users\solnica\Documents\Visual Studio 2010\Projects\Protoss2Com\Protoss2ComClient\Program.cs @ 35:
002300b4 8b4df8          mov     ecx,dword ptr [ebp-8]
002300b7 ff1534381400    call    dword ptr ds:[143834h] (Protoss2ComClient.Program.CallMethod(Protoss2(
002300bd 90              nop

C:\Users\solnica\Documents\Visual Studio 2010\Projects\Protoss2Com\Protoss2ComClient\Program.cs @ 36:
002300be 90              nop
002300bf 8be5            mov     esp,ebp
002300c1 5d              pop     ebp
002300c2 c3              ret
```

!DumpIL

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

```
0:000> !dumpmt -md 00143840
EEClass:      00141484
Module:       00142e9c
Name:         Protoss2ComClient.Program
mdToken:      02000005
File:         C:\Users\solnica\Documents\Visual Studio 2010\Projects\Protoss2Com\Protoss2Co
BaseSize:        0xc
ComponentSize:   0x0
Slots in VTable: 7
Number of IFaces in IFaceMap: 0
--------------------------------------
MethodDesc Table
   Entry MethodDesc      JIT Name
5033a7e0   50114934   PreJIT System.Object.ToString()
5033e2e0   5011493c   PreJIT System.Object.Equals(System.Object)
5033e1f0   5011495c   PreJIT System.Object.GetHashCode()
503c1600   50114970   PreJIT System.Object.Finalize()
0014c019   00143838     NONE Protoss2ComClient.Program..ctor()
00230070   00143820      JIT Protoss2ComClient.Program.Main(System.String[])
0014c015   0014382c     NONE Protoss2ComClient.Program.CallMethod(Protoss2ComClient.Nexus)
0:000> !DumpIL 00143820
ilAddr = 01322050
IL_0000: nop
IL_0001: newobj Protoss2ComClient.NexusClass::.ctor
IL_0006: stloc.0
IL_0007: ldstr "test"
IL_000c: call System.Console::WriteLine
IL_0011: nop
IL_0012: ldloc.0
IL_0013: call Protoss2ComClient.Program::CallMethod
IL_0018: nop
IL_0019: ret
```

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.

```
--------------------------------------
Domain 1:            004e41f8
LowFrequencyHeap:    004e4574
HighFrequencyHeap:   004e45c0
StubHeap:            004e460c
Stage:               OPEN
SecurityDescriptor:  004e5970
Name:                Protoss2ComClient.exe
Assembly:            005266d8 [C:\Windows\Microsoft.Net\assembly\GAC_32\mscorlib\
ClassLoader:         00526518
SecurityDescriptor:  00526640
  Module Name
63a71000             C:\Windows\Microsoft.Net\assembly\GAC_32\mscorlib\v4.0_4.0.0

Assembly:            00539cf8 [C:\Users\solnica\Documents\Visual Studio 2010\Proj
ClassLoader:         00539d98
SecurityDescriptor:  00535d98
  Module Name
000d2e9c             C:\Users\solnica\Documents\Visual Studio 2010\Projects\Proto
```

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.

```
Domain 1:           005e41f8
LowFrequencyHeap:   005e4574
HighFrequencyHeap:  005e45c0
StubHeap:           005e460c
Stage:              OPEN
SecurityDescriptor: 005e5970
Name:               Protoss2ComClient.exe
Assembly:           0062bd18 [C:\Windows\Microsoft.Net\assembly\GAC_32\mscorlib\v4
ClassLoader:        0062bdb8
SecurityDescriptor: 00629d50
  Module Name
50111000            C:\Windows\Microsoft.Net\assembly\GAC_32\mscorlib\v4.0_4.0.0.0

Assembly:           00636900 [C:\Users\solnica\Documents\Visual Studio 2010\Projec
ClassLoader:        006369a0
SecurityDescriptor: 00635d48
  Module Name
00152e9c            C:\Users\solnica\Documents\Visual Studio 2010\Projects\Protoss

0:000> !DumpModule -mt 00152e9c
Name:       C:\Users\solnica\Documents\Visual Studio 2010\Projects\Protoss2Com\Pro
Attributes: PEFile
Assembly:   00636900
LoaderHeap:               00000000
TypeDefToMethodTableMap: 001500c4
TypeRefToMethodTableMap: 001500dc
MethodDefToDescMap:       00150158
FieldDefToDescMap:        00150174
MemberRefToDescMap:       00150178
FileReferencesMap:        001501e0
AssemblyReferencesMap:    001501e4
MetaData start address:   00a82088 (2312 bytes)

Types defined in this module

      MT  TypeDef Name
------------------------------------------------------------------------------
001538bc 0x02000002 Protoss2ComClient.INexus
001538f8 0x02000003 Protoss2ComClient.Nexus
00153990 0x02000004 Protoss2ComClient.NexusClass
00153840 0x02000005 Protoss2ComClient.Program

Types referenced in this module

      MT    TypeRef Name
------------------------------------------------------------------------------
5042f5e8 0x01000001 System.Object
504375f4 0x0100001e System.Console
```

!DumpAssembly

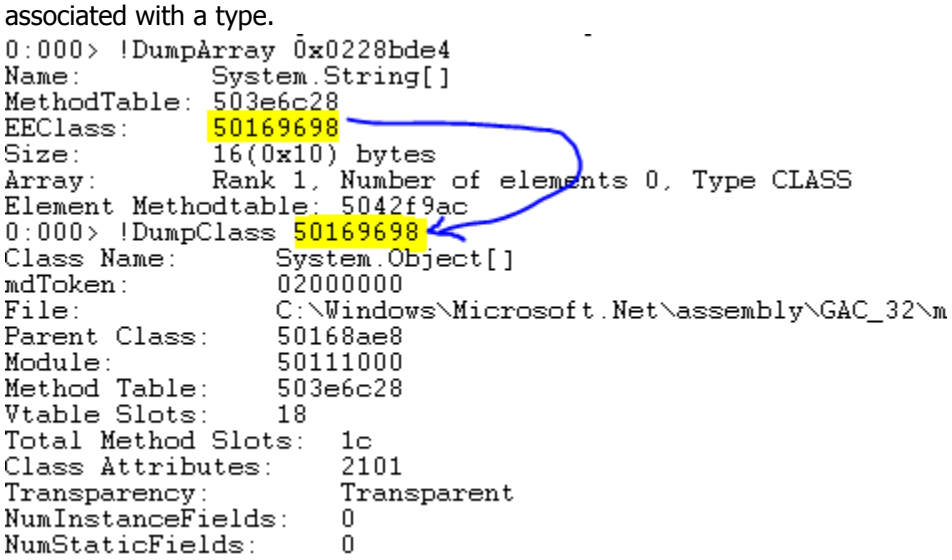
`!DumpAssembly <assembly address>` (**SOS/PSSCOR2**) - dumps information about assembly including all its modules. You may get the assembly address from DumpDomain command.

```
0:000> !DumpAssembly 00539cf8
Parent Domain:      004e41f8
Name:               C:\Users\solnica\Documents\Visual Studio 2010\Projects\Protoss2Com\Protoss2ComClient\bin\Debug\
ClassLoader:        00539d98
  Module Name
000d2e9c            C:\Users\solnica\Documents\Visual Studio 2010\Projects\Protoss2Com\Protoss2ComClient\bin\Debug\
```

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.

```
C:\Users\solnica\Documents\Visual Studio 2010\Projects\Protoss2Com\Protoss2ComClient\Program.cs @ 31:
0023008d b990391400      mov     ecx,143990h (MT: Protoss2ComClient.NexusClass)
00230092 e8032b335b      call    clr!JIT_NewCrossContext (5b562b9a)
00230097 8945f4          mov     dword ptr [ebp-0Ch],eax
0023009a 8b4df4          mov     ecx,dword ptr [ebp-0Ch]
0023009d e8debff1ff      call    0014c080 (Protoss2ComClient.NexusClass..ctor(), mdToken: 06000003)
002300a2 8b45f4          mov     eax,dword ptr [ebp-0Ch]
002300a5 8945f8          mov     dword ptr [ebp-8],eax

C:\Users\solnica\Documents\Visual Studio 2010\Projects\Protoss2Com\Protoss2ComClient\Program.cs @ 33:
002300a8 8b0d30207303    mov     ecx,dword ptr ds:[3732030h] ("test")
002300ae e8f96f1350      call    mscorlib_ni+0x2570ac (503670ac) (System.Console.WriteLine(System.String), mdToken:
002300b3 90              nop

C:\Users\solnica\Documents\Visual Studio 2010\Projects\Protoss2Com\Protoss2ComClient\Program.cs @ 35:
002300b4 8b4df8          mov     ecx,dword ptr [ebp-8]
002300b7 ff1534381400    call    dword ptr ds:[143834h] (Protoss2ComClient.Program.CallMethod(Protoss2ComClient.Nex
002300bd 90              nop

C:\Users\solnica\Documents\Visual Studio 2010\Projects\Protoss2Com\Protoss2ComClient\Program.cs @ 36:
002300be 90              nop
002300bf 8be5            mov     esp,ebp
002300c1 5d              pop     ebp
002300c2 c3              ret
0:000> !IP2MD 0023009d
MethodDesc:   00143820
Method Name:  Protoss2ComClient.Program.Main(System.String[])
Class:        00141484
MethodTable:  00143840
mdToken:      06000004
Module:       00142e9c
IsJitted:     yes
CodeAddr:     00230070
Transparency: Critical
Source file:  C:\Users\solnica\Documents\Visual Studio 2010\Projects\Protoss2Com\Protoss2ComClient\Program.cs @ 31
```

!DumpMD

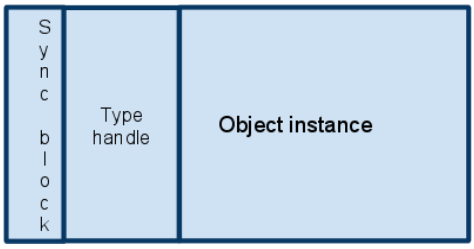
`!DumpMD <MethodDesc address>` (**SOS/PSSCOR2**) - dumps method description. m_CodeOrIL if method is jited points to the assembly code that was generated for this method.

```
0:000> !DumpMT -md 00153990
EEClass:       001514f0
Module:        00152e9c
Name:          Protoss2ComClient.NexusClass
mdToken:       02000004
File:          C:\Users\solnica\Documents\Visual Studio 2010\Projects\Protoss2Com\Protoss2ComClient\bi
BaseSize:         0x10
ComponentSize:    0x0
Slots in VTable: 9
Number of IFaces in IFaceMap: 2
--------------------------------------
MethodDesc Table
   Entry MethodDesc      JIT Name
5033a7e0   50114934   PreJIT System.Object.ToString()
5033e2e0   5011493c   PreJIT System.Object.Equals(System.Object)
5033e1f0   5011495c   PreJIT System.Object.GetHashCode()
503c1600   50114970   PreJIT System.Object.Finalize()
503d5aa0   50197f58   PreJIT System.MarshalByRefObject.GetLifetimeService()
50337e94   50197f60   PreJIT System.MarshalByRefObject.InitializeLifetimeService()
5037bf0c   50197f68   PreJIT System.MarshalByRefObject.CreateObjRef(System.Type)
0015c070   00153948     NONE Protoss2ComClient.NexusClass.Test()
0015c080   0015396c     NONE Protoss2ComClient.NexusClass..ctor()
0:000> !DumpMD 5011493c
Method Name:  System.Object.Equals(System.Object)
Class:        50113ef8
MethodTable:  5042f5e8
mdToken:      06000003
Module:       50111000
IsJitted:     yes
CodeAddr:     5033e2e0
Transparency: Transparent
0:000> !U 5033e2e0
preJIT generated code
System.Object.Equals(System.Object)
Begin 5033e2e0, size 6
*** WARNING: Unable to verify checksum for C:\Windows\assembly\NativeImages_v4.0.30319_32\mscorlib\4f

f:\dd\ndp\clr\src\BCL\System\Object.cs @ 67:
>>> 5033e2e0 e8b388faff      call    mscorlib_ni+0x1d6b98 (502e6b98) (System.Runtime.CompilerServices
5033e2e5 c3              ret
```

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.

```
    PARAMETERS:
        args (0x0046ec30) = 0x027cbde4
    LOCALS:
        0x0046ec2c = 0x027cbe0c

0046ee6c 66f721db [GCFrame: 0046ee6c]
0:000> dd 0x027cbde4
027cbde4  63d46c28 00000000 63d8f9ac 80000000
027cbdf4  63d8f9ac 00000004 00650074 00740073
027cbe04  00000000 08000001 000d3990 00000000
027cbe14  00000000 00000000 63d97490 00000000
027cbe24  00000000 00000000 00000000 00000000
027cbe34  00000000 63d8f5e8 00000000 40010000
027cbe44  63d96034 00000007 00000004 00000100
027cbe54  00000000 63d96f40 00000000 00000000
0:000> dd 0x027cbde4 - 4
027cbde0  00000000 63d46c28 00000000 63d8f9ac
027cbdf0  80000000 63d8f9ac 00000004 00650074
027cbe00  00740073 00000000 08000001 000d3990
027cbe10  00000000 00000000 00000000 63d97490
027cbe20  00000000 00000000 00000000 00000000
027cbe30  00000000 00000000 63d8f5e8 00000000
027cbe40  40010000 63d96034 00000007 00000004
027cbe50  00000100 00000000 63d96f40 00000000
```

!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.

```
0:000> !CLRStack -a
OS Thread Id: 0x1ad4 (0)
Child SP IP       Call Site
003cf254 75d8b727 [HelperMethodFrame: 003cf254] System.S
003cf2d0 00220194 DomainBoundILStubClass.IL_STUB_CLRtoCO
    PARAMETERS:
        this = <no data>

003cf2d4 002200ff [InlinedCallFrame: 003cf2d4] Protoss2C
003cf35c 002200ff Protoss2ComClient.Program.CallMethod(P
    PARAMETERS:
        n (0x003cf35c) = 0x0228be0c

003cf368 002200bd Protoss2ComClient.Program.Main(System.
    PARAMETERS:
        args (0x003cf370) = 0x0228bde4
    LOCALS:
        0x003cf36c = 0x0228be0c

003cf5a4 5b5421db [GCFrame: 003cf5a4]
0:000> !DumpArray 0x0228bde4
Name:        System.String[]
MethodTable: 503e6c28
EEClass:     50169698
Size:        16(0x10) bytes
Array:       Rank 1, Number of elements 0, Type CLASS
Element Methodtable: 5042f9ac
```

!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

```
0:000> !CLRStack -p
OS Thread Id: 0x176c (0)
Child SP IP       Call Site
0046ec24 002700e0 Protoss2ComClient.Program.CallMethod(Protoss2ComClient.Nexus) [C:\Use
    PARAMETERS:
        n (<CLR reg>) = 0x027cbe0c

0046ec28 002700bd Protoss2ComClient.Program.Main(System.String[]) [C:\Users\solnica\Doc
    PARAMETERS:
        args (0x0046ec30) = 0x027cbde4

0046ee6c 66f721db [GCFrame: 0046ee6c]
0:000> !do 0x027cbe0c
Name:        Protoss2ComClient.NexusClass
MethodTable: 000d3990
EEClass:     000d14f0
Size:        16(0x10) bytes
File:        C:\Users\solnica\Documents\Visual Studio 2010\Projects\Protoss2Com\Protoss2
Fields:
      MT    Field   Offset                 Type VT     Attr    Value Name
63d8f5e8  40001cf        4        System.Object  0 instance 00000000 __identity
63d9537c  40002cc        8 ...ections.Hashtable  0 instance 00000000 m_ObjectToDataMap
```

!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.

```
0:000> !SyncBlk 1
Index SyncBlock MonitorHeld Recursion Owning Thread Info  SyncBlock Owner
    1 005523dc            0         0 00000000     none    027cbe0c Protoss2ComClient
-----------------------------
Total           1
CCW             0
RCW             1
ComClassFactory 0
Free            0
```

ASP.NET

!DumpHttpContext (!ASPXPages)

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

```
0:022> !DumpHttpContext
Going to dump the HttpContexts found in the heap.
Loading the heap objects into our cache.
HttpContext    Timeout  Completed     Running  ThreadId ReturnCode   Verb RequestPath+QueryString
0x00000000ffa82ce8    0                    no       336 Sec     XXX       200    /
0x00000000ffb38c70    19200 Sec        no       333 Sec     XXX       200    GET /default.aspx
0x00000000ffb3ac60    19200 Sec        no        36 Sec     XXX       200    GET /default.aspx
0x000000013fac1230    19200 Sec        no       336 Sec     XXX       200    GET /default.aspx
Total 4 HttpContext objects
```

!DumpHttpRuntime

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

```
0:022> !DumpHttpRuntime
Going to dump the HttpRuntimes found in the heap.
Loading the heap objects into our cache.
HttpRuntime 0x000000013f9cb600:
_shutdownInProgress: 0
_requestQueue:

0x000000013fac1d88
_appDomainAppPath: C:\websites\webtest\
_appDomainAppId: /LM/W3SVC/2/ROOT
_fcm:

Name: System.Web.FileChangesMonitor
MethodTable: 000007feead2c4d8
EEClass: 000007feea9f9118
Size: 104(0x68) bytes
GC Generation: 0
 (C:\Windows\assembly\GAC_64\System.Web\2.0.0.0__b03f5f7f11d50a3a\System.Web.dll)
Fields:
              MT            Field           Offset                 Type VT             Attr            Value Name
000007feead2e280  4000dcf       58 ...ReadWriteSpinLock  1 instance 000000013f9ccbc0 _lockDispose
000007feebc06c50  4000dd0       50       System.Boolean  1 instance                0 _disposed
000007feebc0f3d8  4000dd1        8 ...ections.Hashtable  0 instance 000000013f9ccd48 _aliases
000007feebc0f3d8  4000dd2       10 ...ections.Hashtable  0 instance 000000013f9ccda8 _dirs
000007feead2e300  4000dd3       18 ....DirectoryMonitor  0 instance 000000013f9cefa8 _dirMonSubdirs
000007feebc0f3d8  4000dd4       20 ...ections.Hashtable  0 instance 000000013f9ccf20 _subDirDirMons
000007feebc0e968  4000dd5       28 ...ections.ArrayList  0 instance 000000013f9cf3a0 _dirMonSpecialDirs
000007feead2e1a0  4000dd6       30 ...hangeEventHandler  0 instance 000000013f9ce788 _callbackRenameOrCriticaldirChange
000007feebc0ecf0  4000dd7       48          System.Int32  1 instance                0 _activeCallbackCount
000007feead2e300  4000dd8       38 ....DirectoryMonitor  0 instance 0000000000000000 _dirMonAppPathInternal
000007feebc07a80  4000dd9       40        System.String  0 instance 0000000000000000 _appPathInternal
000007feebc0ecf0  4000dda       4c          System.Int32  1 instance                0 _FCNMode
000007feebbf5870  4000dce       48      System.Object[]  0   shared           static s_dirsToMonitor
                                >> Domain:Value  0000000000fa9630:NotInit  0000000003362560:000000013f9ce8d0 <<

 cacheInternal:
```

!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***