5 заповедей PDB

Сергей Щегрикович

dotmailer

shchegrikovich@gmail.com
@shchegrikovich

Сколько времени занимает отладка?

35% - 50%

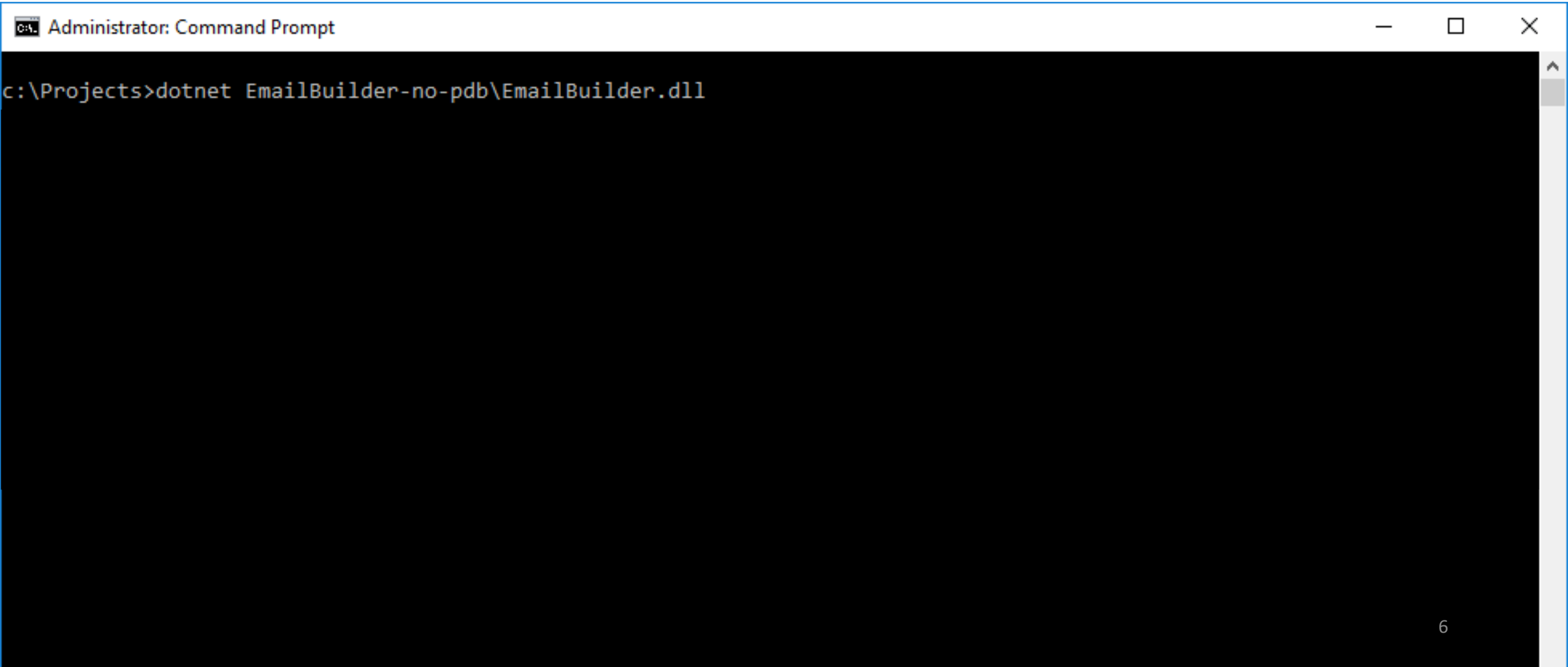
Такой разный дебаггинг

- PrintF
- Логирование в файлы
- Через прерывание – int 3
- В дебагере с PDB

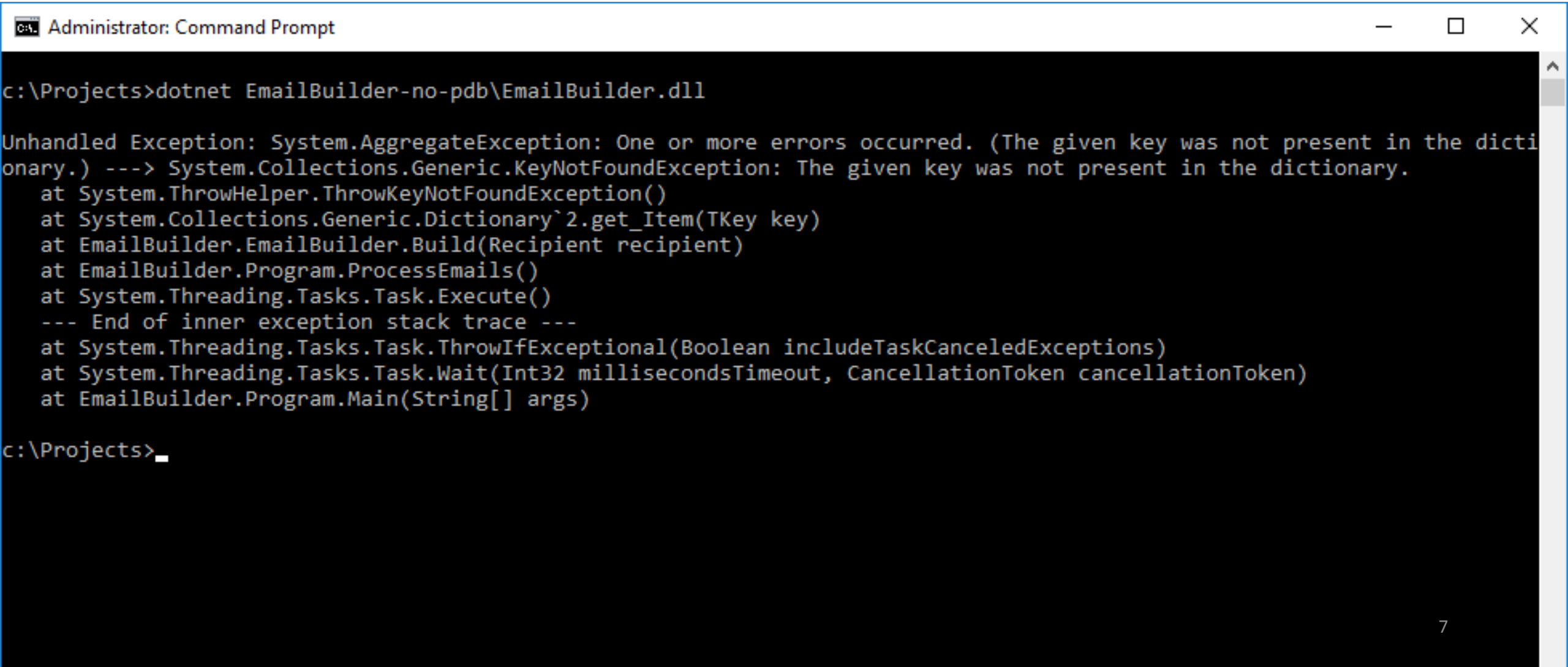
PDB – это …

- Program Database, представлен в 1993 году
- Контейнер отладочной информации для Windows
- Отображение кода приложения в машинные инструкции

PDB отсутствуют на продакшене



PDB отсутствуют на продакшене



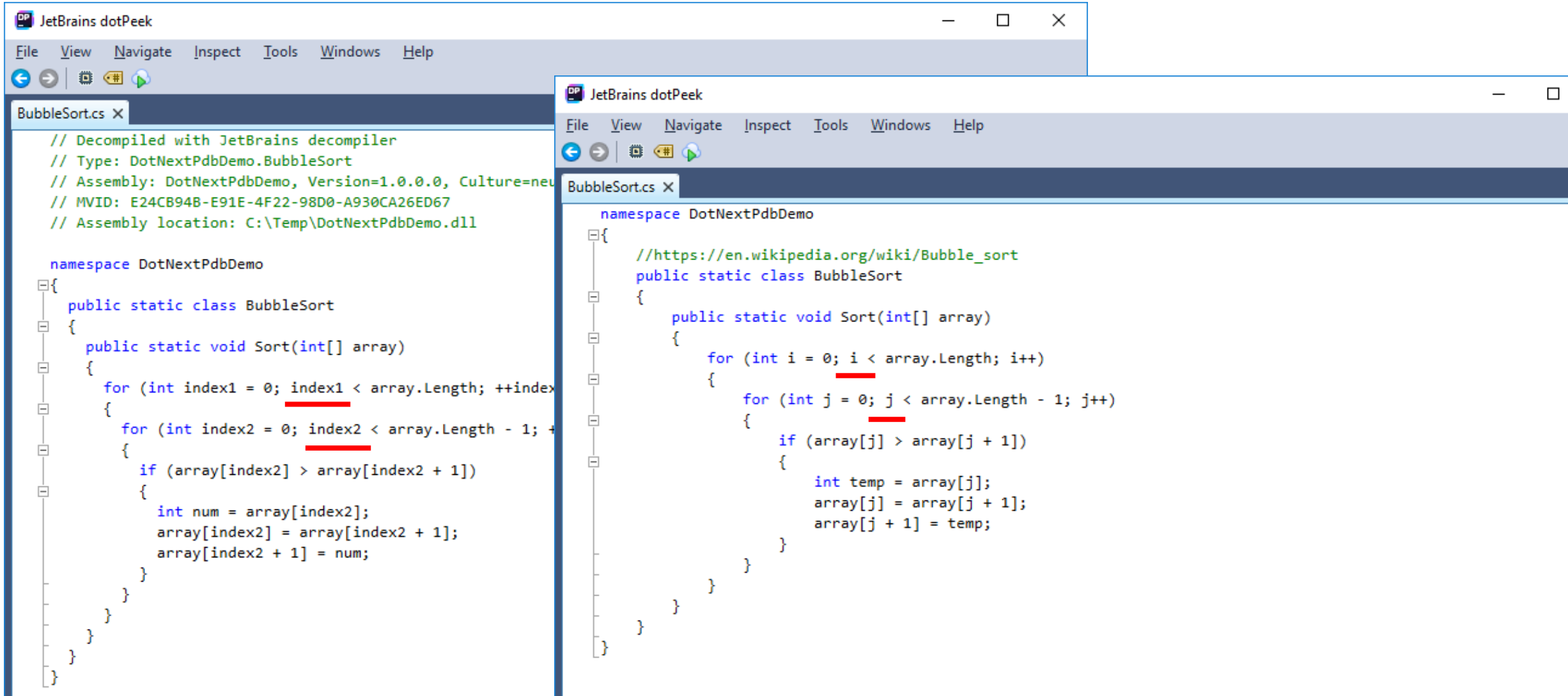
PDB отсутствуют на продакшене

Administrator: Command Prompt

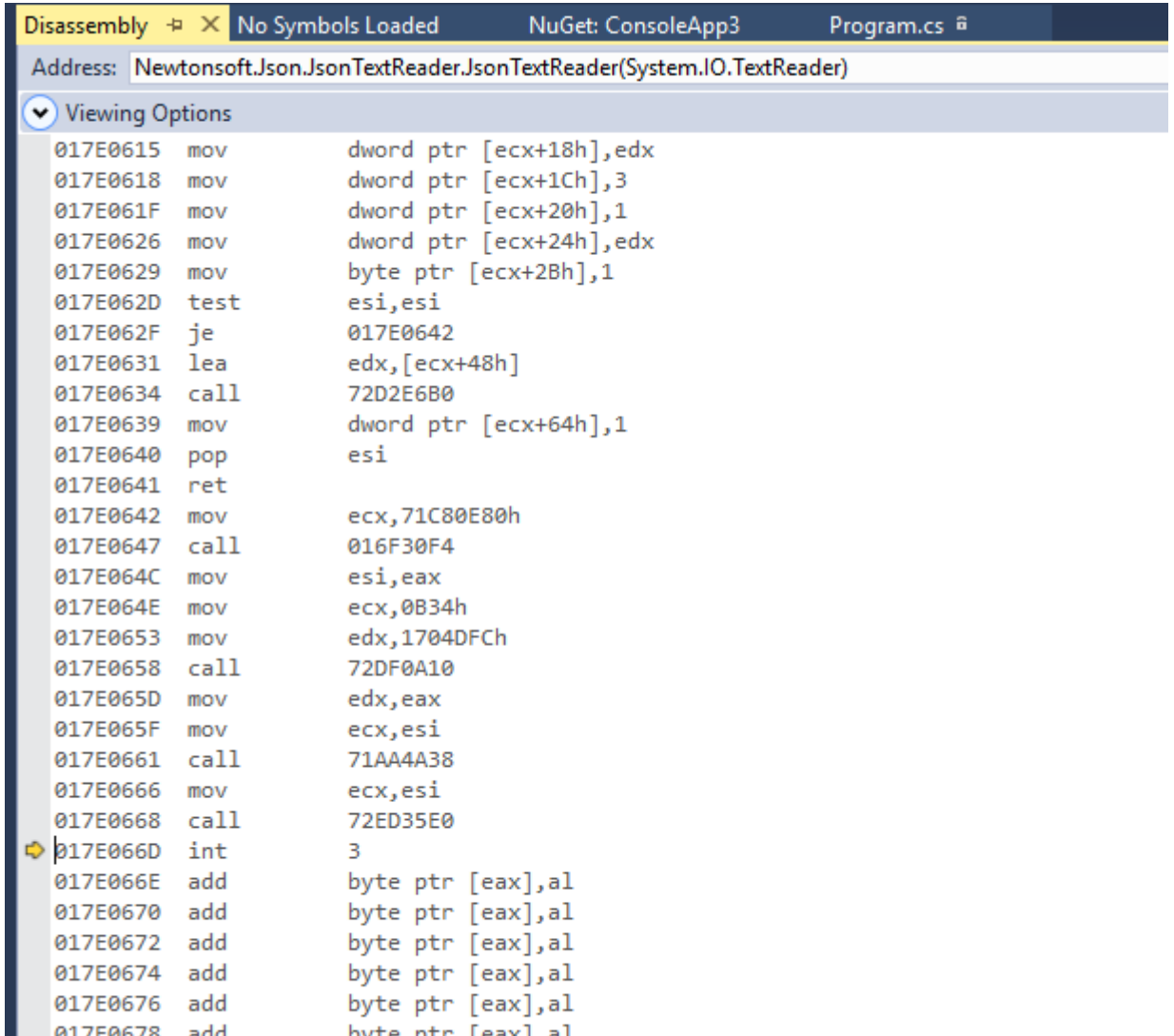
```
c:\Projects>dotnet EmailBuilder-no-pdb\EmailBuilder.dll

Unhandled Exception: System.AggregateException: One or more errors occurred. (The given key was not present in the dicti
onary.) ---> System.Collections.Generic.KeyNotFoundException: The given key was not present in the dictionary.
   at System.ThrowHelper.ThrowKeyNotFoundException()
   at System.Collections.Generic.Dictionary`2.get_Item(TKey key)
   at EmailBuilder.EmailBuilder.Build(Recipient recipient)
```

```
48

        1 reference
49      internal String Build(Recipient recipient)
50      {
51          var user = _users[recipient.UserId];
52          var email = _emails[$"{recipient.UserId}-{recipient.EmailTypeId}"];
53          var result = email.Body.Replace("{email}", user.Email);
54          return result;
55      }
56  }
57
```

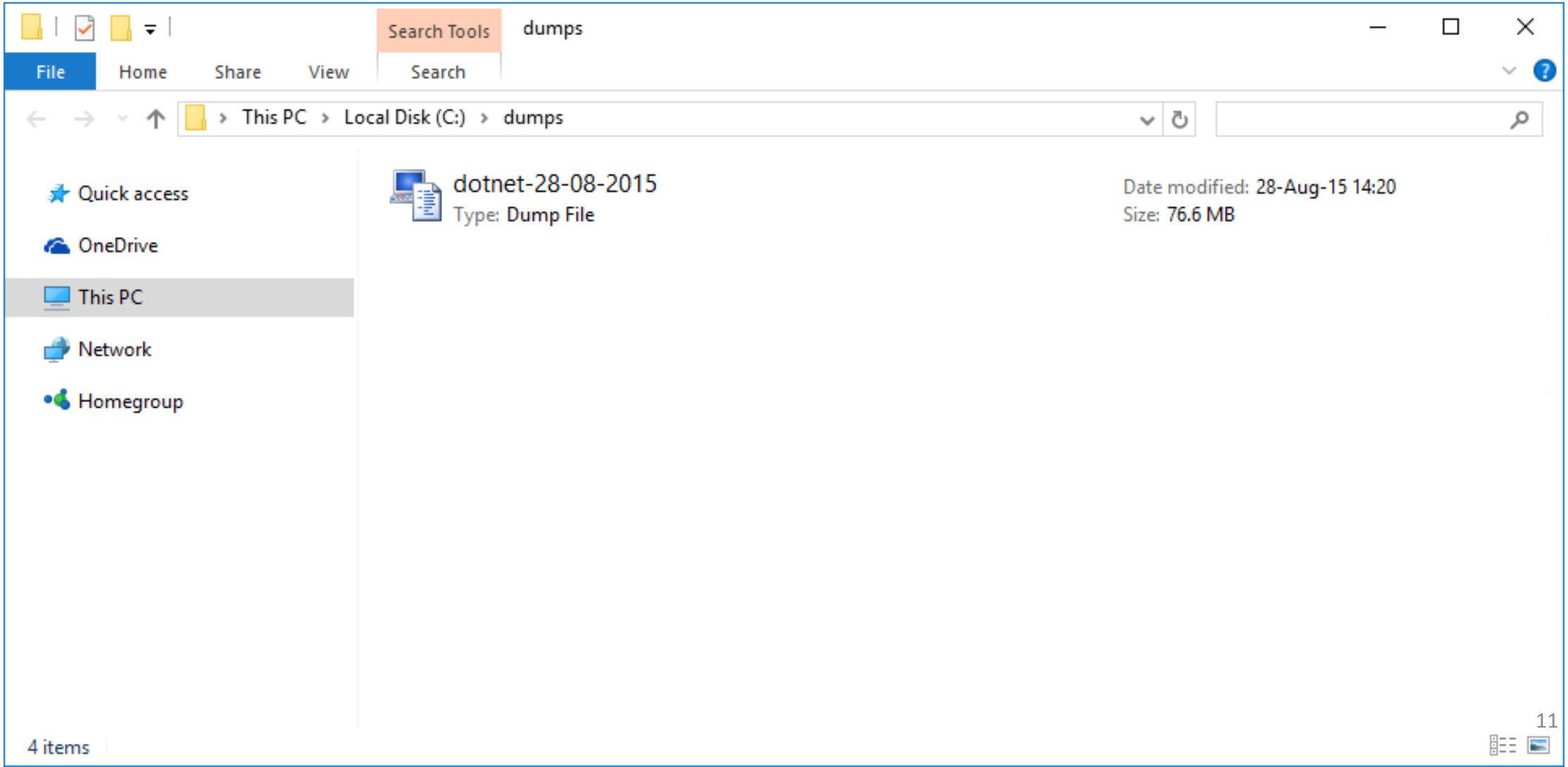
PDB не хранят информацию о исходниках



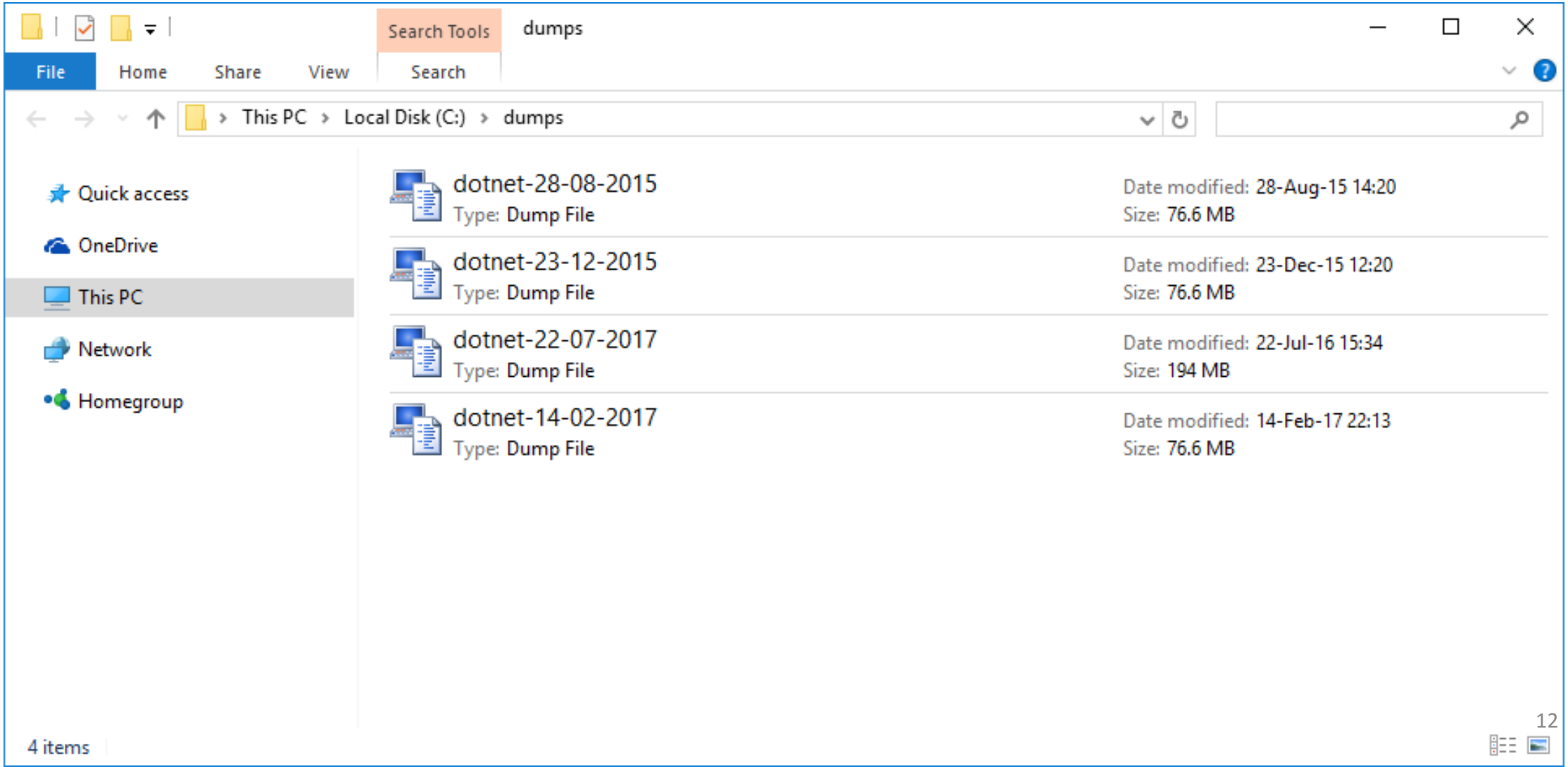
PDB отсутствуют для сторонних сборок



PDB удаляются после релиза



PDB удаляются после релиза



А как же dotPeek?

- Декомпилятор
- Трудно автоматизировать работу команды

Содержание

- Где взять PDB?
- Что такое PDB?
- Долгосрочное хранение – Symbol server
- Отладка без исходников – Source indexing
- Будущее отладочных символов – PortablePDB формат

5 заповедей PDB

#1 Используй _NT_SYMBOL_PATH

#2 Знай свои PDB

#3 Индексируй исходники

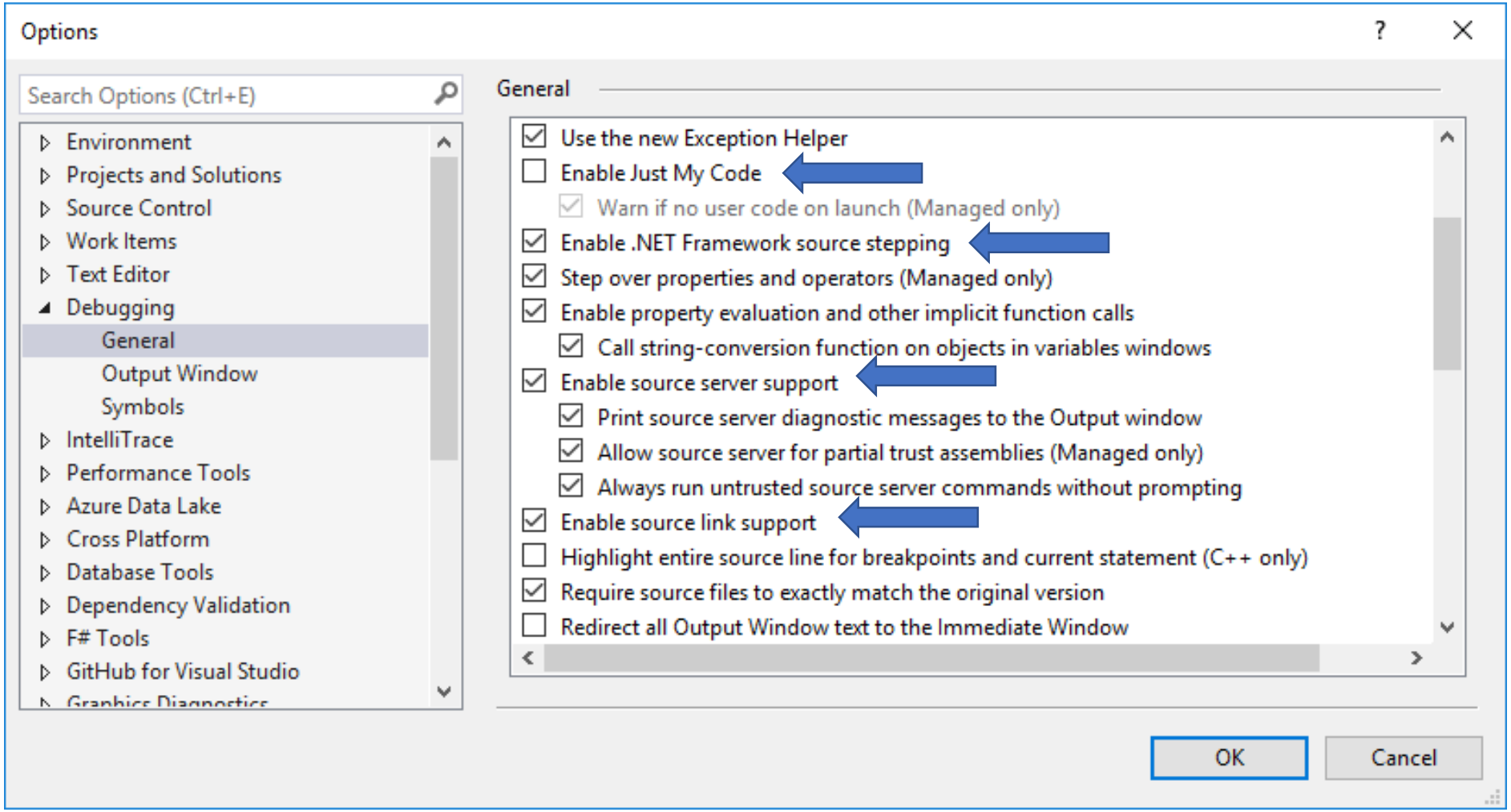
#4 Держи PDB рядом

#5 Делись

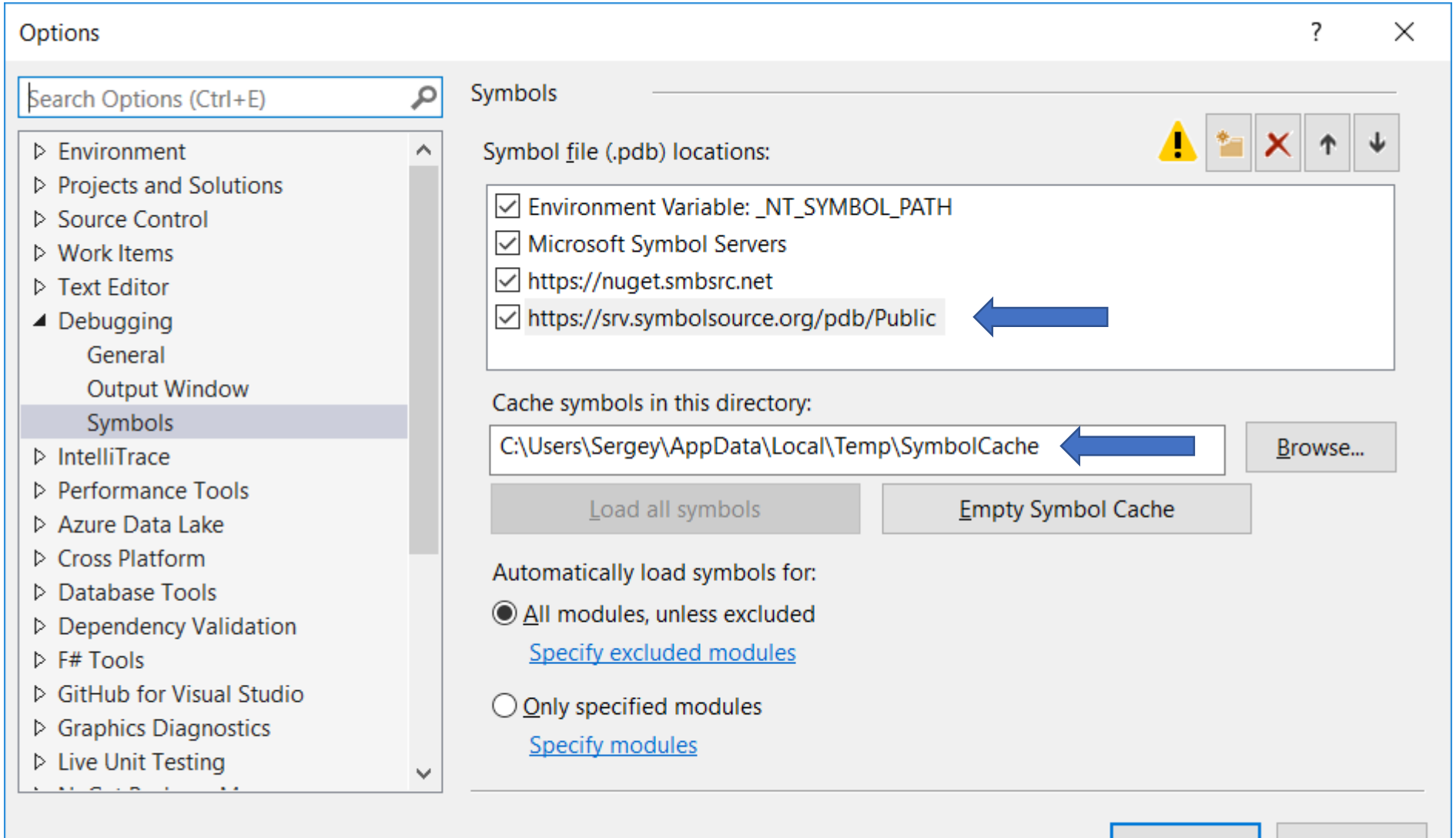
Symbol Server

- https://msdl.microsoft.com/download/symbols
- https://srv.symbolsource.org/pdb/Public
- https://nuget.smbsrc.net

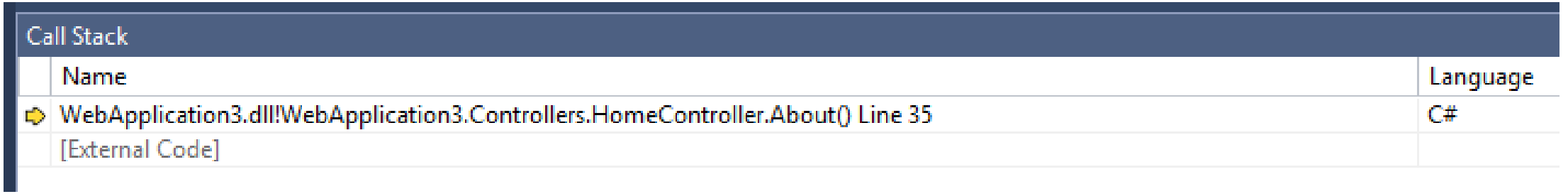
VS – Tools -> Options -> Debugging -> General



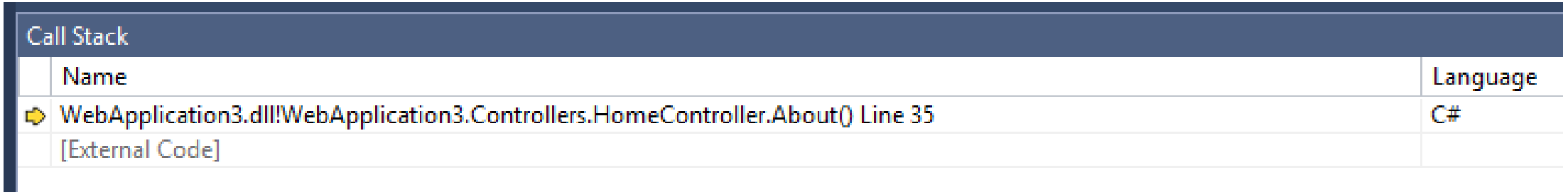
VS – Tools -> Options -> Debugging -> Symbols

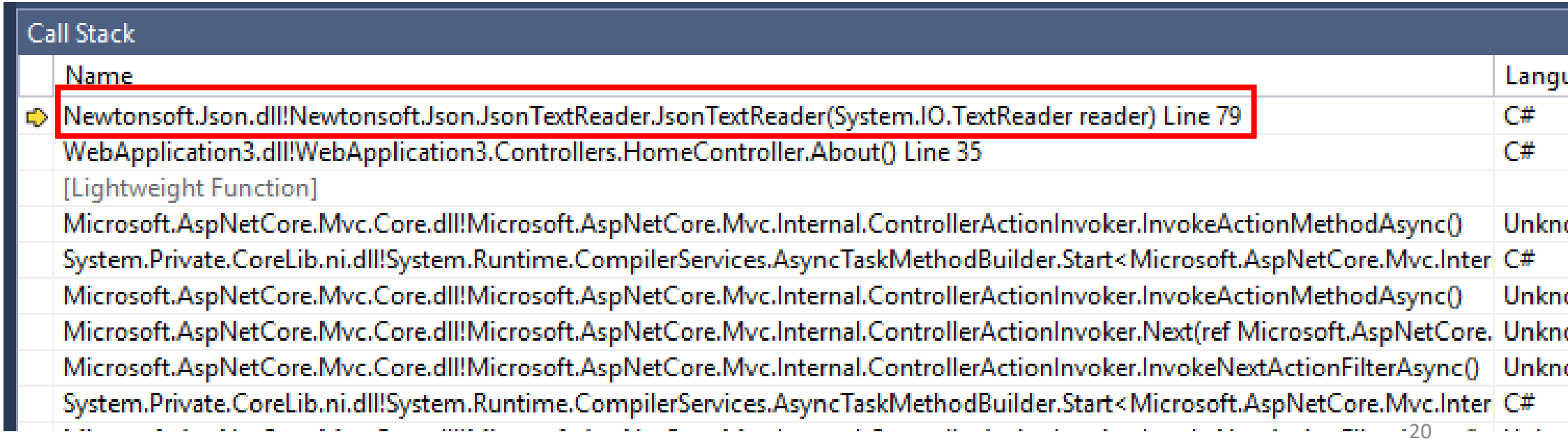


Отладка сторонних сборок



Отладка сторонних сборок





Кто еще использует Symbol Server?

- Отладчики – WinDBG
- Профилировщики – PerfView
- Системные утилиты – SysInternals Tools

#1 Используй _NT_SYMBOL_PATH

SET _NT_SYMBOL_PATH

=C:\Symbols

=C:\Symbols ;\\net-path\

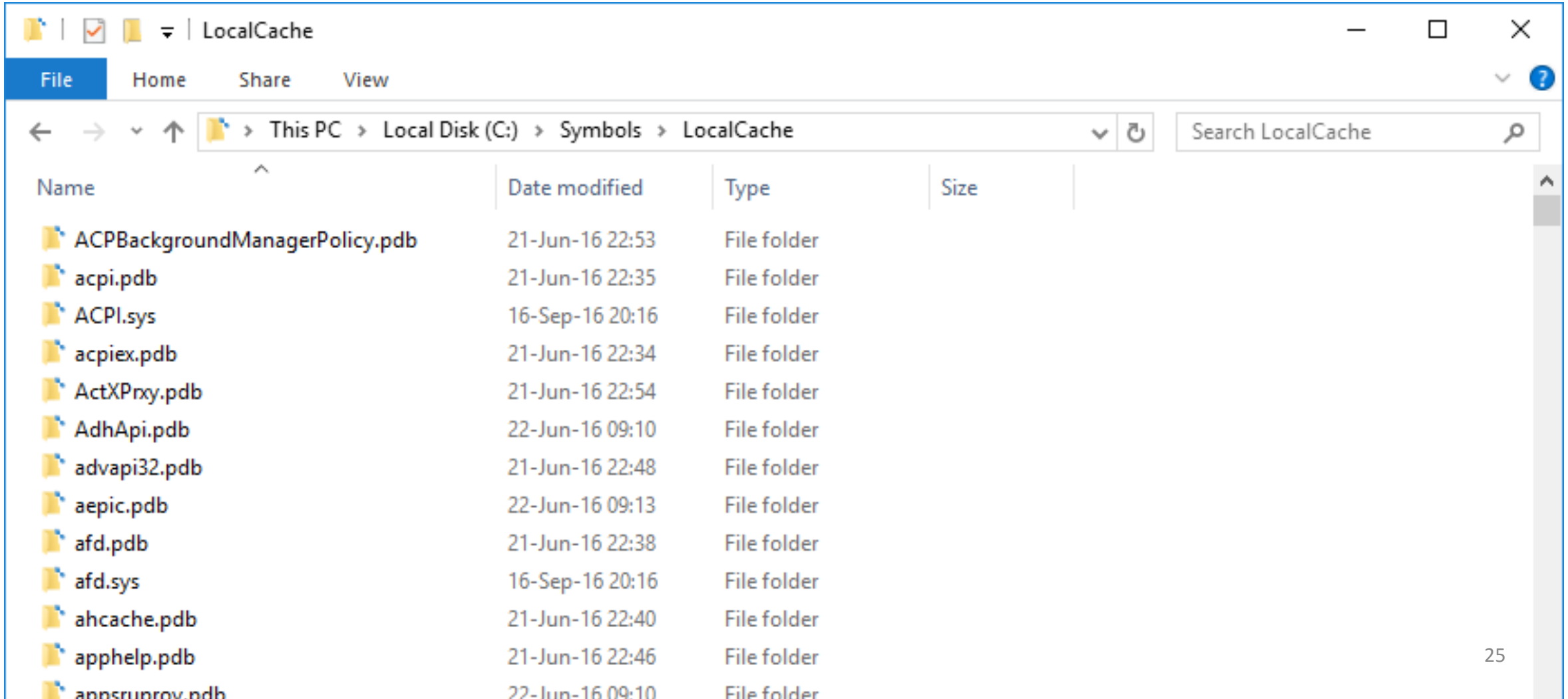
=SRV*C:\Cache*https://nuget.smbsrc.net

=cache*C:\Symbols;\\server\share

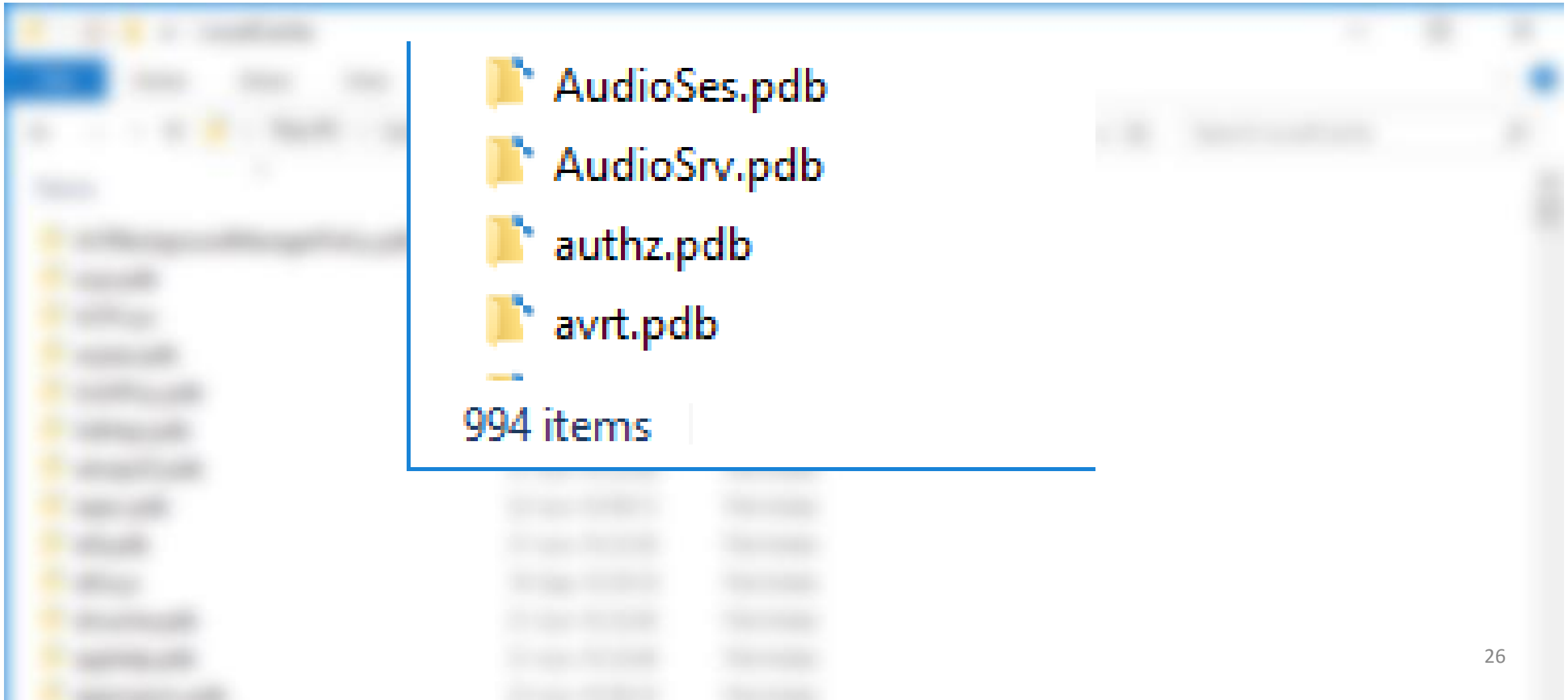
SET _NT_SYMBOL_PATH

=C:\Symbols\Local;
cache*C:\Symbols\LocalCache;
srv*https://nuget.smbsrc.net;
srv*https://srv.symbolsource.org/pdb/Public;
srv*https://msdl.microsoft.com/download/symbols;

Локальный кеш

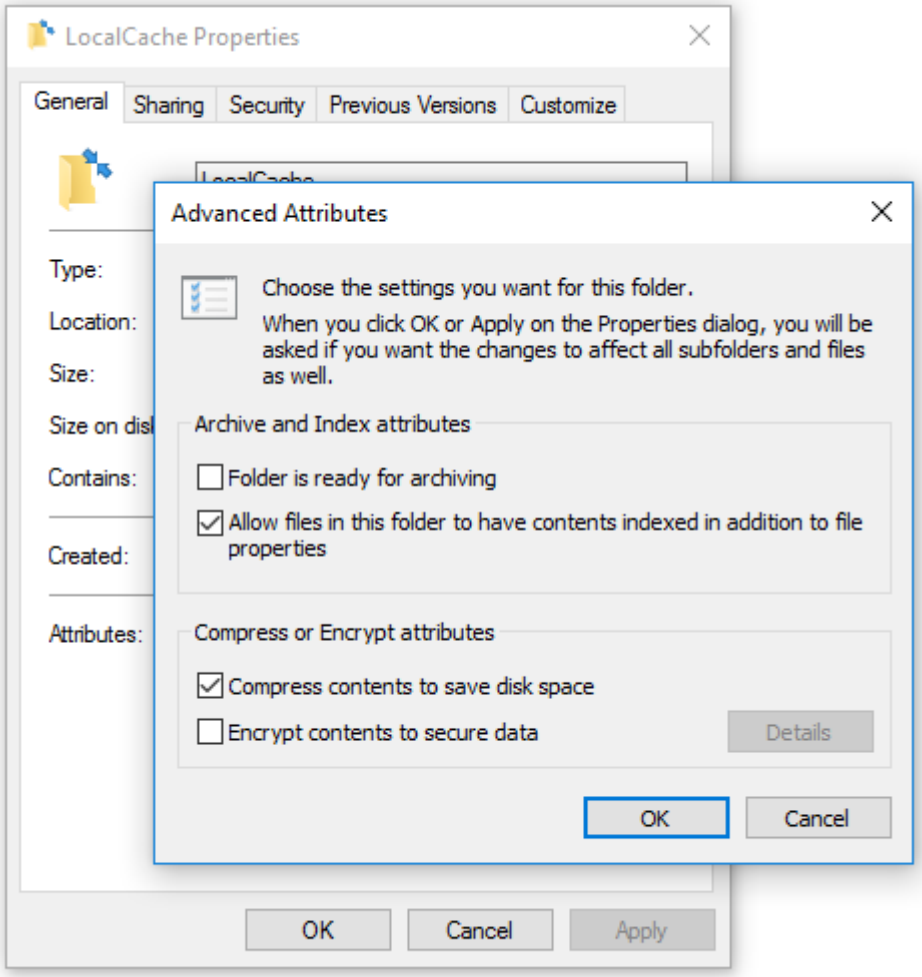


Локальный кеш



Оптимизация локального кеша

- Архивировать



Оптимизация локального кеша

- Архивировать
- agestore c:\symbols -date=01-07-2008

  agestore -lat=on

Много символов не бывает

- https://msdl.microsoft.com/download/symbols
- https://srv.symbolsource.org/pdb/Public
- https://nuget.smbsrc.net
- https://referencesource.microsoft.com/symbols
- https://dotnet.myget.org/F/dotnet-core/symbols
- https://download.amd.com/dir/bin
- https://symbols.mozilla.org
- https://chromium-browser-symsrv.commondatastorage.googleapis.com
- https://symbols.autodesk.com/symbols
- \\network-share\symbols

Много символов не бывает … или бывает?

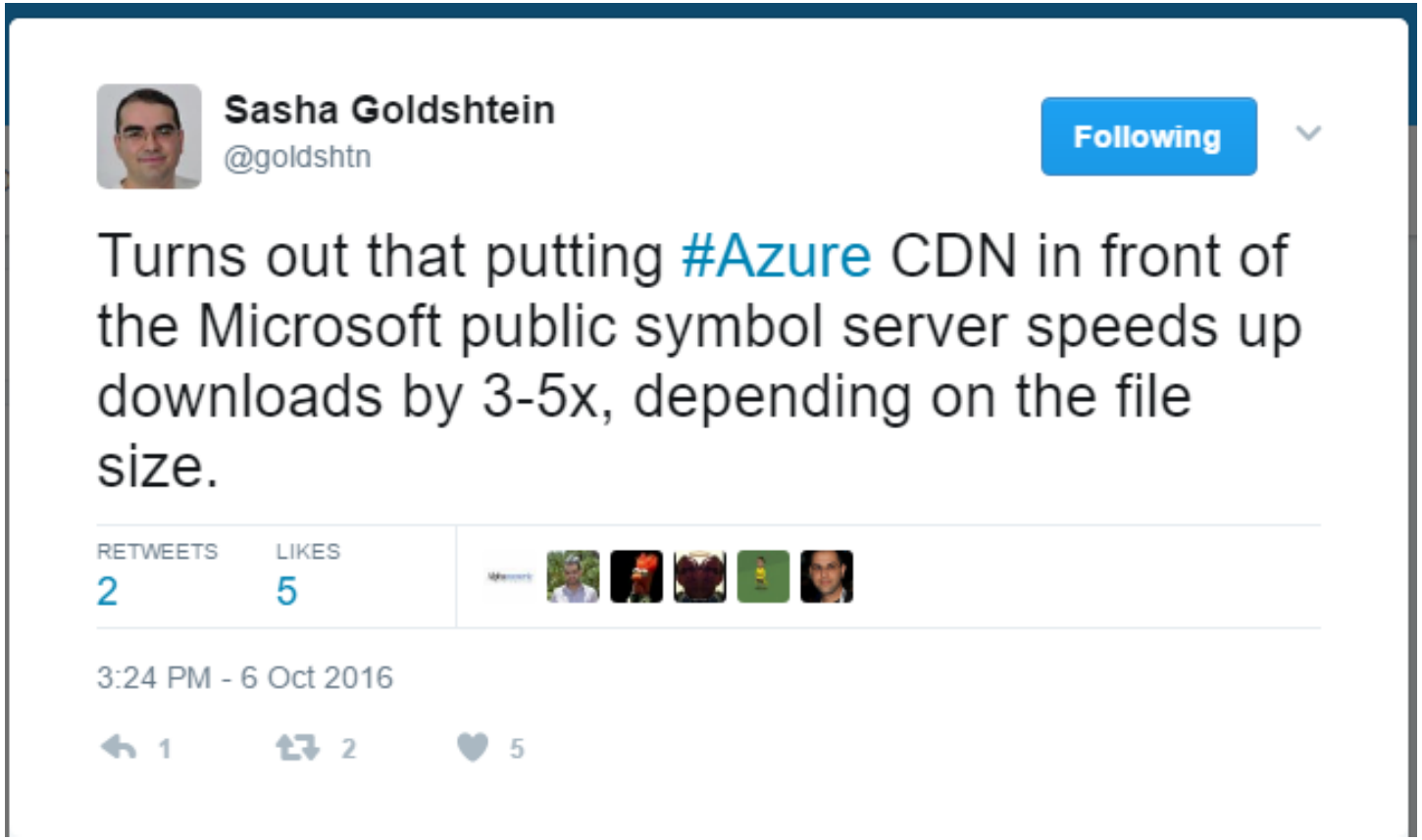
- https://msdl.microsoft.com/download/symbols
- https://srv.symbolsource.org/pdb/Public
- https://nuget.smbsrc.net
- https://referencesource.microsoft.com/symbols
- https://dotnet.myget.org/F/dotnet-core/symbols
- https://download.amd.com/dir/bin
- https://symbols.mozilla.org
- https://chromium-browser-symsrv.commondatastorage.googleapis.com
- https://symbols.autodesk.com/symbols
- \\network-share\symbols

Symbol Server Hell

- Все пути проверяются по очереди
- 404 ответы серверов не кешируются
- Тратим время

Symbol Server Hell решение

- Public CDN

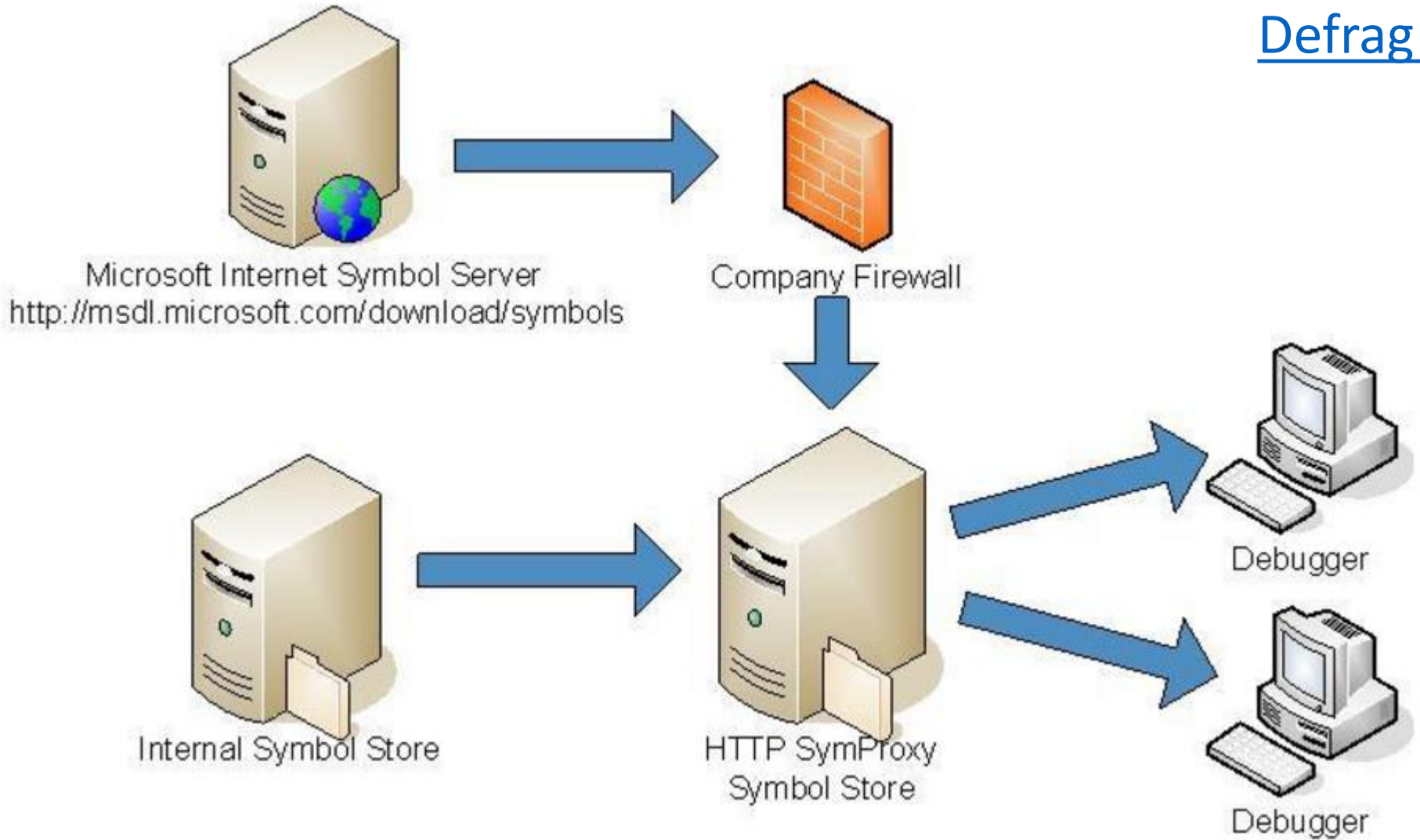


Symbol Server Hell решение

- Public CDN
- SymProxy

SymProxy от Andrew Richards

[Defrag Tools #169]



SymProxy

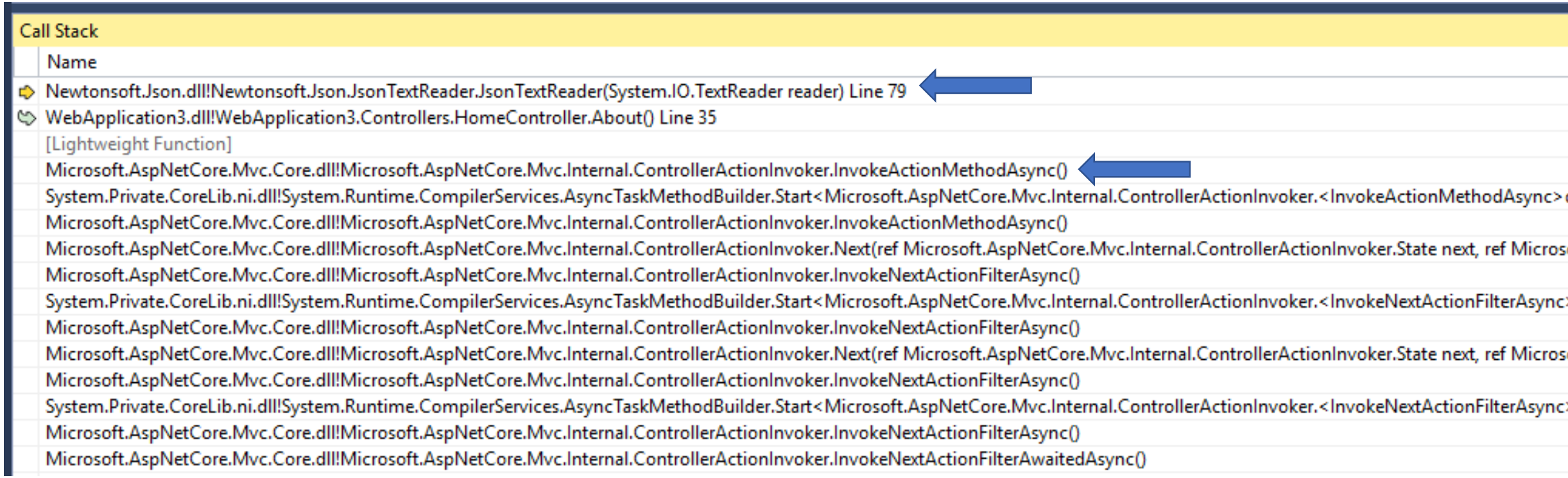
- Скрывает детали реализации
- Кеширует 404 ответы
- Много разных настроек

Вывод – #1 Используй _NT_SYMBOL_PATH

- Унифицированный доступ к Symbol Server
- Добавь в _NT_SYMBOL_PATH

  https://msdl.microsoft.com/download/symbols
  https://srv.symbolsource.org/pdb/Public
  https://nuget.smbsrc.net
- {Debugging Tools For Windows}\symproxy

Все PDB разные



#2 Знай свои PDB

Public и Private PDB

	Public	Private
Локальные переменные	Нет	Да
Информацию о типах	Частично	Полностью
Номера строк	Нет	Полностью

Public и Private PDB

	Public	Private
Локальные переменные	Нет	Да
Информацию о типах	Частично	Полностью
Номера строк	Нет	Полностью
Размер coreclr.pdb	3.7 MB	53 MB

Код без PDB

```
00 KERNELBASE!RaiseException+0x68
01 coreclr+0x8788d
02 coreclr+0x8800f
03 0x00007ffa`dbbc04ed
04 coreclr!MetaDataGetDispenser+0x23bb3
05 coreclr+0x8e256
06 coreclr!MetaDataGetDispenser+0x1363f
07 coreclr!coreclr_execute_assembly+0x10215
08 coreclr+0x53e0
09 coreclr!coreclr_execute_assembly+0xde
```

Код с Public PDB

```
00 KERNELBASE!RaiseException+0x68
01 coreclr!RaiseTheExceptionInternalOnly+0x245
02 coreclr!IL_Throw+0x10f
03 0x00007ffa`dbbd04ed
04 coreclr!CallDescrWorkerInternal+0x83
05 coreclr!MethodDescCallSite::CallTargetWorker+0x14e
06 coreclr!RunMain+0x17b
07 coreclr!Assembly::ExecuteMainMethod+0xb5
08 coreclr!CorHost2::ExecuteAssembly+0x170
09 coreclr!coreclr_execute_assembly+0xde
```

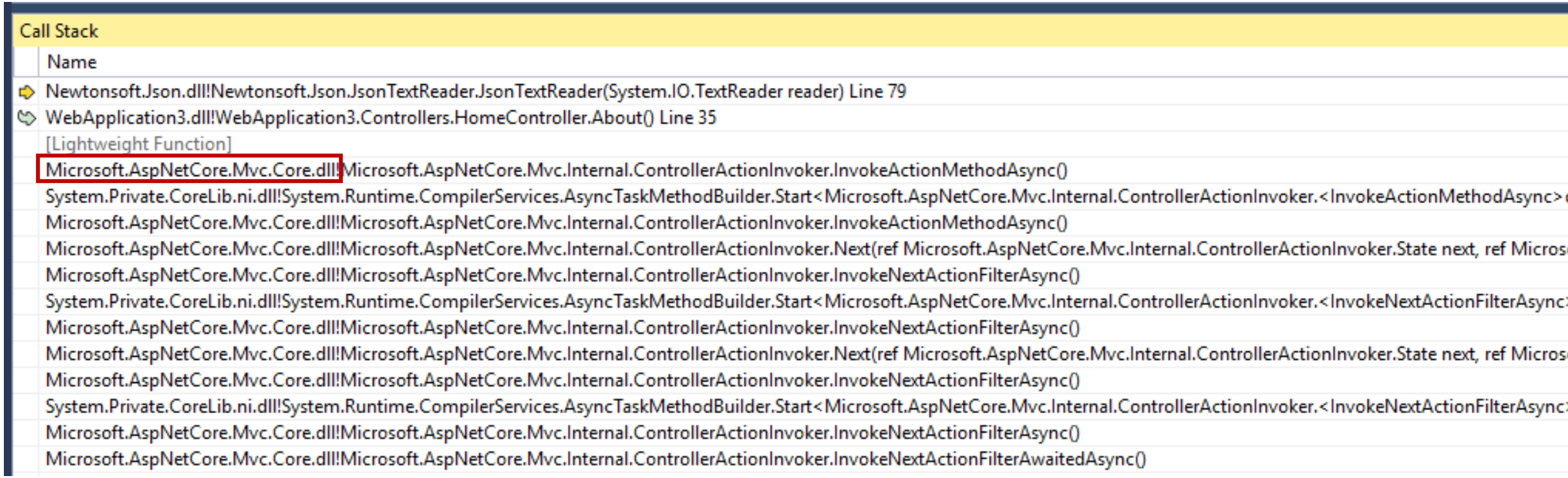
Код с Private PDB

```
00 KERNELBASE!RaiseException+0x68
01 coreclr!RaiseTheExceptionInternalOnly+0x245 [e:\src\vm\excep
02 coreclr!IL_Throw+0x10f [e:\src\vm\jithelpers.cpp @ 5449]
03 0x00007ffa`dbba04ed
04 coreclr!CallDescrWorkerInternal+0x83 [E:\src\vm\amd64\CallD
05 coreclr!MethodDescCallSite::CallTargetWorker+0x14e [e:\src\v
06 (Inline Function) coreclr!MethodDescCallSite::Call+0x3f [e:\
07 coreclr!RunMain+0x17b [e:\src\vm\assembly.cpp @ 2639]
08 coreclr!Assembly::ExecuteMainMethod+0xb5 [e:\src\vm\assembly
09 coreclr!CorHost2::ExecuteAssembly+0x170 [e:\src\vm\corhost.c
0a coreclr!coreclr_execute_assembly+0xde [e:\src\dlls\mscoree\u
```

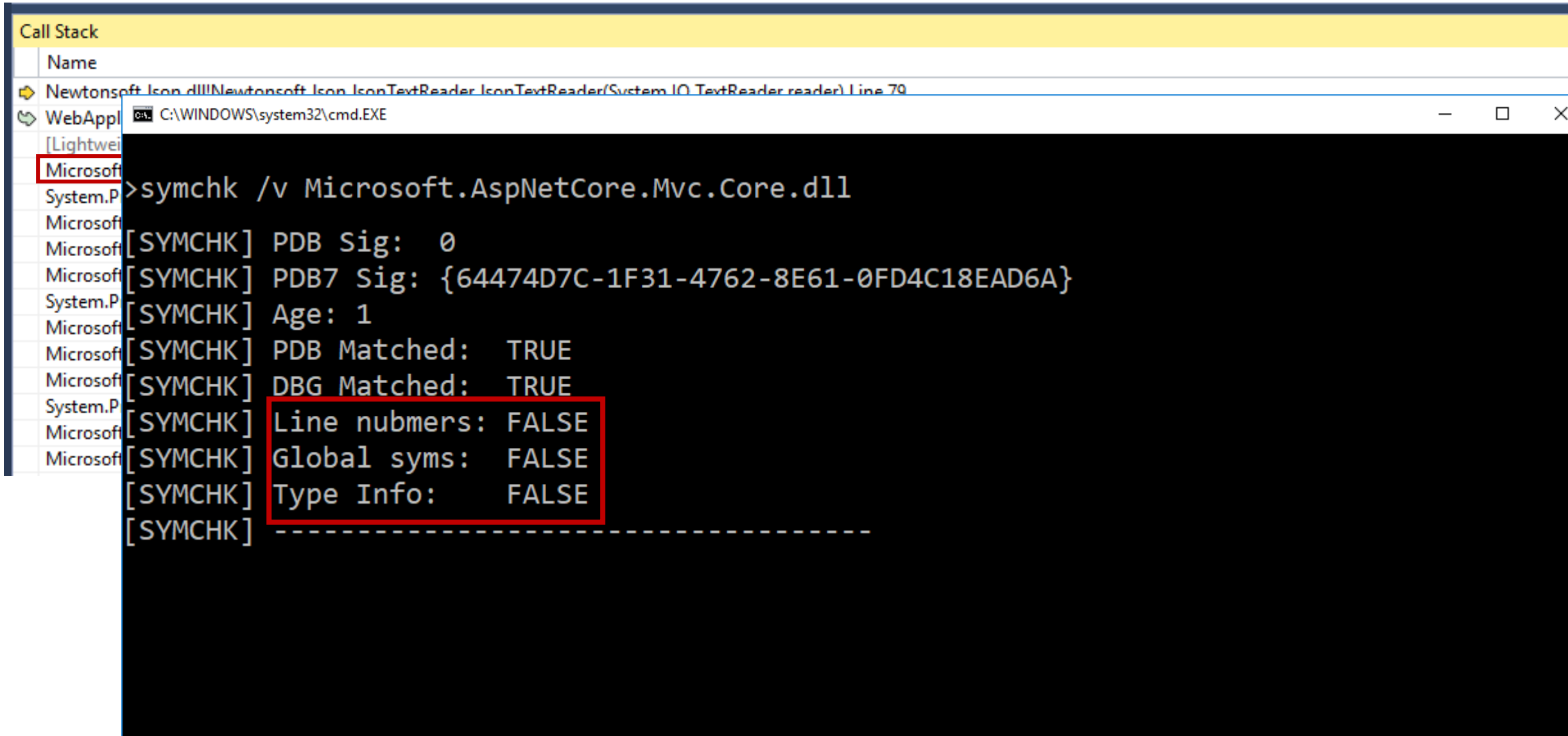
А как в .Net?

- Информация о типах хранится в мета-данных
- PDB нужны только для локальных переменных и строк

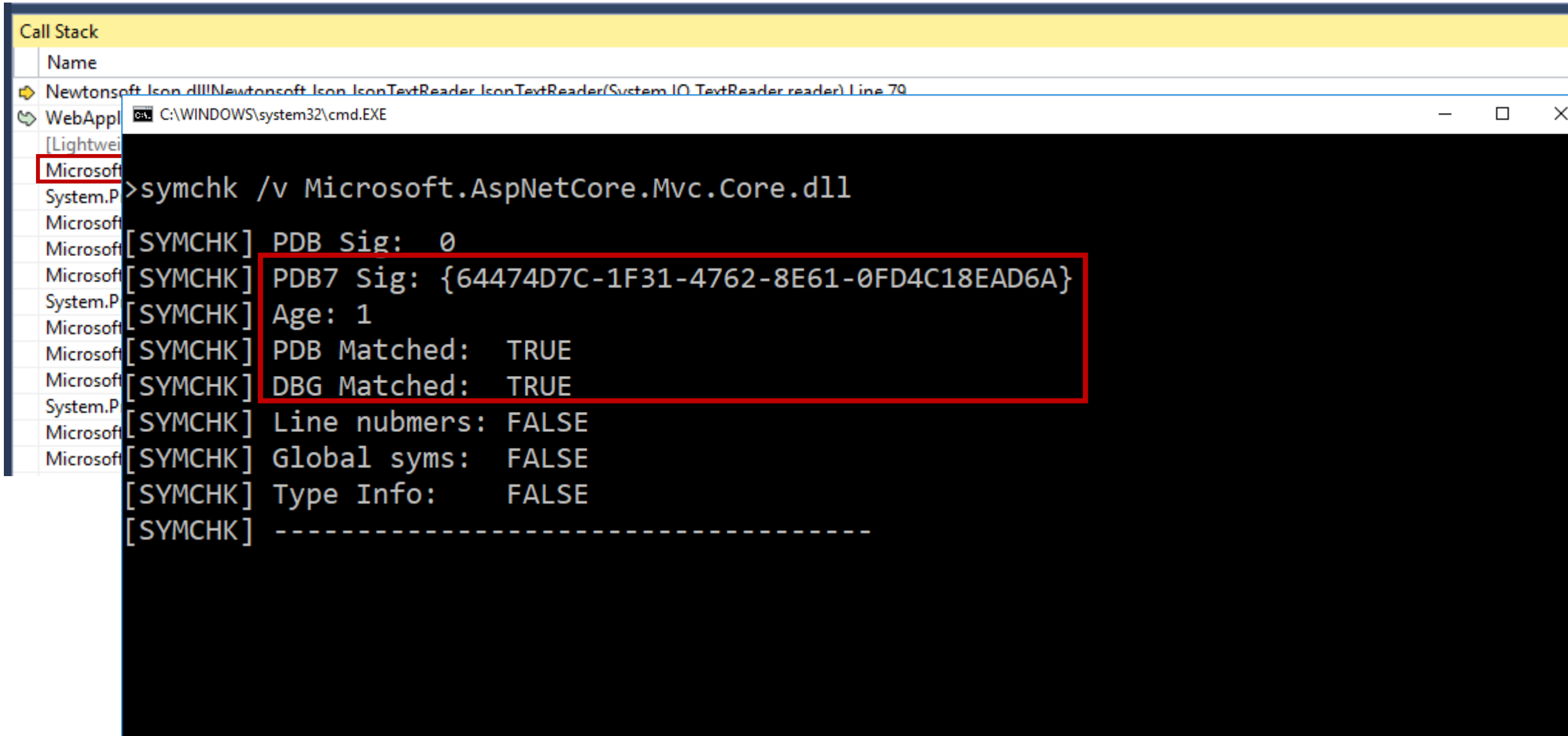
symchk.exe


Call Stack
Name
Newtonsoft.Json.dll!Newtonsoft.Json.JsonTextReader.JsonTextReader(System.IO.TextReader reader) Line 79
WebApplication3.dll!WebApplication3.Controllers.HomeController.About() Line 35
[Lightweight Function]
Microsoft.AspNetCore.Mvc.Core.dll!Microsoft.AspNetCore.Mvc.Internal.ControllerActionInvoker.InvokeActionMethodAsync()
System.Private.CoreLib.ni.dll!System.Runtime.CompilerServices.AsyncTaskMethodBuilder.Start<Microsoft.AspNetCore.Mvc.Internal.ControllerActionInvoker.<InvokeActionMethodAsync>
Microsoft.AspNetCore.Mvc.Core.dll!Microsoft.AspNetCore.Mvc.Internal.ControllerActionInvoker.InvokeActionMethodAsync()
Microsoft.AspNetCore.Mvc.Core.dll!Microsoft.AspNetCore.Mvc.Internal.ControllerActionInvoker.Next(ref Microsoft.AspNetCore.Mvc.Internal.ControllerActionInvoker.State next, ref Micros
Microsoft.AspNetCore.Mvc.Core.dll!Microsoft.AspNetCore.Mvc.Internal.ControllerActionInvoker.InvokeNextActionFilterAsync()
System.Private.CoreLib.ni.dll!System.Runtime.CompilerServices.AsyncTaskMethodBuilder.Start<Microsoft.AspNetCore.Mvc.Internal.ControllerActionInvoker.<InvokeNextActionFilterAsync
Microsoft.AspNetCore.Mvc.Core.dll!Microsoft.AspNetCore.Mvc.Internal.ControllerActionInvoker.InvokeNextActionFilterAsync()
Microsoft.AspNetCore.Mvc.Core.dll!Microsoft.AspNetCore.Mvc.Internal.ControllerActionInvoker.Next(ref Microsoft.AspNetCore.Mvc.Internal.ControllerActionInvoker.State next, ref Micros
Microsoft.AspNetCore.Mvc.Core.dll!Microsoft.AspNetCore.Mvc.Internal.ControllerActionInvoker.InvokeNextActionFilterAsync()
System.Private.CoreLib.ni.dll!System.Runtime.CompilerServices.AsyncTaskMethodBuilder.Start<Microsoft.AspNetCore.Mvc.Internal.ControllerActionInvoker.<InvokeNextActionFilterAsync
Microsoft.AspNetCore.Mvc.Core.dll!Microsoft.AspNetCore.Mvc.Internal.ControllerActionInvoker.InvokeNextActionFilterAsync()
Microsoft.AspNetCore.Mvc.Core.dll!Microsoft.AspNetCore.Mvc.Internal.ControllerActionInvoker.InvokeNextActionFilterAwaitedAsync()

symchk.exe


Call Stack
Name
Newtonsoft.Json.dll!Newtonsoft.Json.JsonTextReader.JsonTextReader(System.IO.TextReader reader) Line 79
WebAppl
[Lightwei
Microsoft
System.P
Microsoft
Microsoft
Microsoft
System.P
Microsoft
Microsoft
Microsoft
System.P
Microsoft
Microsoft
C:\WINDOWS\system32\cmd.EXE
>symchk /v Microsoft.AspNetCore.Mvc.Core.dll
[SYMCHK] PDB Sig: 0
[SYMCHK] PDB7 Sig: {64474D7C-1F31-4762-8E61-0FD4C18EAD6A}
[SYMCHK] Age: 1
[SYMCHK] PDB Matched: TRUE
[SYMCHK] DBG Matched: TRUE
[SYMCHK] Line nubmers: FALSE
[SYMCHK] Global syms: FALSE
[SYMCHK] Type Info: FALSE
[SYMCHK] ----------

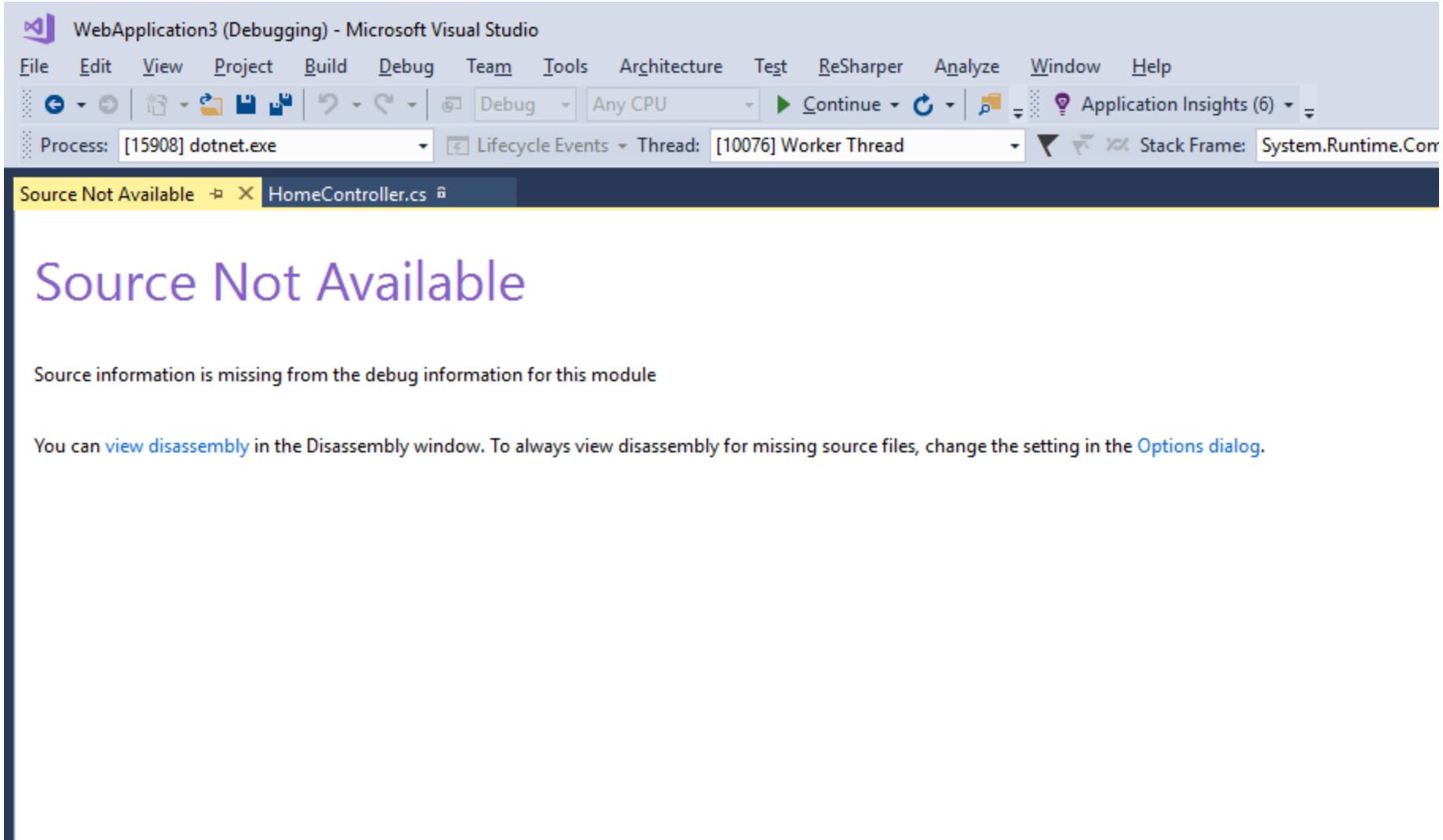
symchk.exe


Call Stack
Name
Newtonsoft.Json.dll!Newtonsoft.Json.JsonTextReader.JsonTextReader(System.IO.TextReader reader) Line 79
WebAppl
[Lightwei
Microsoft
System.P
Microsoft
Microsoft
Microsoft
System.P
Microsoft
Microsoft
Microsoft
System.P
Microsoft
Microsoft
C:\WINDOWS\system32\cmd.EXE
>symchk /v Microsoft.AspNetCore.Mvc.Core.dll
[SYMCHK] PDB Sig: 0
[SYMCHK] PDB7 Sig: {64474D7C-1F31-4762-8E61-0FD4C18EAD6A}
[SYMCHK] Age: 1
[SYMCHK] PDB Matched: TRUE
[SYMCHK] DBG Matched: TRUE
[SYMCHK] Line nubmers: FALSE
[SYMCHK] Global syms: FALSE
[SYMCHK] Type Info: FALSE
[SYMCHK] ----------

Вывод – #2 Знай свои PDB

- Позволяет быстро понять перспективы отладки
- Export, Public или Private
- symchk /v coreclr.dll /s coreclr.pdb

Поиск исходников с PDB



SRCSRV поток

```
SRCSRV: ini ------------------------------------------------
VERSION=2
SRCSRV: variables ------------------------------------------
SRCSRVTRG=%var2%
SRCSRVCMD=download-sources.exe %var3%
SRCSRV: source files ---------------------------------------
d:\_work\src\inc\daccess.h*src/inc/daccess.h*15
SRCSRV: end ------------------------------------------------
```

SRCSRV поток

SRCSRV: ini --
VERSION=2
SRCSRV: variables ------------------------------------
SRCSRVTRG=%var2%
SRCSRVCMD=download-sources.exe %var3%
SRCSRV: source files ---------------------------------
d:_work\src\inc\daccess.h*src/inc/daccess.h*15
SRCSRV: end --

SRCSRV поток

SRCSRV: ini --
VERSION=2
SRCSRV: variables ------------------------------------
SRCSRVTRG=%var2%
SRCSRVCMD=download-sources.exe %var3%
SRCSRV: source files ---------------------------------
d:_work\src\inc\daccess.h ***src/inc/daccess.h** ***15**
SRCSRV: end --

SRCSRV для HTTP

SRCSRV: variables --
GIT_COMMIT_HASH=1735a1d453677717e68803da6a85284d15dca891
GIT_REPO_URI=https://raw.githubusercontent.com/dotnet/coreclr
HTTP_ALIAS=%GIT_REPO_URI%/%GIT_COMMIT_HASH%
SRCSRVTRG=%HTTP_ALIAS%/%var2%
SRCSRVCMD=
SRCSRVVERCTRL=http
SRCSRV: source files ---------------------------------------
e:\src\toolbox\sos\strike\disasm.cpp*src\ToolBox\SOS\Strike\disasm.cpp

SRCSRV для HTTP

SRCSRV: variables ------------------------------------
GIT_COMMIT_HASH=1735a1d453677717e68803da6a85284d15dca891
GIT_REPO_URI=https://raw.githubusercontent.com/dotnet/coreclr
HTTP_ALIAS=%GIT_REPO_URI%/%GIT_COMMIT_HASH%
SRCSRVTRG=%HTTP_ALIAS%/%var2%
SRCSRVCMD=
SRCSRVVERCTRL=http
SRCSRV: source files ---------------------------------
e:\src\toolbox\sos\strike\disasm.cpp*src\ToolBox\SOS\Strike\disasm.cpp

SRCSRV функции

- %fnvar%(%var1%)
- %fnbksl%(%var2%)
- %fnfile%(%var3%)

#3 Индексируй исходники

- Быстрый доступ к исходному коду
- SRCSRVCMD – вызывает любую команду
- pdbstr.exe -r –p:App.pdb -s:srcsrv

  pdbstr.exe -w -p:App.pdb -s:srcsrv -i:srcsrv.txt
- srctool.exe indexed.pdb
- Автоматизация github.com/Haemoglobin/GitHub-Source-Indexer

Где хранить PDB после релиза?

- https://www.myget.org
- TFS
- TeamCity
- Сделать самому

Как сделать свой symbol server

Symstore.exe add /f lib.pdb /s //network/share /t {product-name}

Demo

- Добавим SRCSRV в PDB
- Создадим локальный symbol server
- Покажу как PerfView загружает исходники

Вывод – #4 Держи PDB рядом

- Уменьшает время начала отладки
- Symstore.exe add /f lib.pdb /s //network/share /t {product-name}
- Храним индексированные PDB

Как хранить PDB для NuGet пакетов?

- https://nuget.smbsrc.net

  Для хранения PDB

  Для хранения исходников

DotNextPdbDemo.nuspec

```
1  <?xml version="1.0"?>
2  <package xmlns="http://schemas.microsoft.com/packaging/2013/05/nuspec.xsd">
3    <metadata>
4      <id>DotNextPdbDemo-symbols</id>
5      <version>0.9.2</version>
6      <authors>Sergey Shchegrikovich</authors>
7      <projectUrl>http://github.com/shchahrykovich/</projectUrl>
8      <description>Demo project</description>
9    </metadata>
10
11     <files>
12       <file src=".\bin\Release\DotNextPdbDemo.*" target="lib\net45" />
13       <file src=".\**\*.cs" target="src" />
14     </files>
15 </package>
16
```

DotNextPdbDemo.nuspec

```
<?xml version="1.0"?>
<package xmlns="http://schemas.microsoft.com/packaging/2013/05/nuspec.xsd">
  <metadata>
    <id>DotNextPdbDemo-symbols</id>
    <version>0.9.2</version>
    <authors>Sergey Shchegrikovich</authors>
    <projectUrl>http://github.com/shchahrykovich/</projectUrl>
    <description>Demo project</description>
  </metadata>

  <files>
    <file src=".\bin\Release\DotNextPdbDemo.*" target="lib\net45" />
    <file src=".\**\*.cs" target="src" />
  </files>
</package>

```

DotNextPdbDemo.nuspec

```
1  <?xml version="1.0"?>
2  <package xmlns="http://schemas.microsoft.com/packaging/2013/05/nuspec.xsd">
3    <metadata>
4      <id>DotNextPdbDemo-symbols</id>
5      <version>0.9.2</version>
6      <authors>Sergey Shchegrikovich</authors>
7      <projectUrl>http://github.com/shchahrykovich/</projectUrl>
8      <description>Demo project</description>
9    </metadata>
10
11     <files>
12       <file src=".\bin\Release\DotNextPdbDemo.*" target="lib\net45" />
13       <file src=".\**\*.cs" target="src" />
14     </files>
15 </package>
16
```

Залить на сервера NuGet

```
nuget pack -Version "1.9" -Symbols demo.nuspec

nuget push -Source
"https://www.nuget.org/api/v2/package"
"demo.1.9.nupkg"

nuget push -Source "https://nuget.smbsrc.net"
"demo.1.9.symbols.nupkg"
```

#5 Делись

- nuget push -Source "https://nuget.smbsrc.net/" "Lib.symbols.nupkg"
- Zip архивы с PDB
- github.com/ctaggart/SourceLink
- github.com/shchahrykovich/AwesomePdb

PDB и .Net Core

- 25-летний плохо документированный формат
- PDB разработали без учета .NET
- Нет поддержки на Linux

Portable PDB

- Разработан для .Net
- Метадата, ECMA-335 Partition II
- Может быть частью сборки
- Встроенное сжатие
- Поддержка на Linux

Изменения в компиляторе

- Поддержка Portable PDB

  /debug:{full|pdbonly|portable|embedded}

- Deterministic builds

  /deterministic Produce a deterministic assembly
  (including module version GUID and timestamp)

- Работа с исходниками

  /embed Embed all source files in the PDB.
  /embed:<file list> Embed specfic files the PDB
  /sourcelink:<file> Source link info to embed into Portable PDB.

Source Link

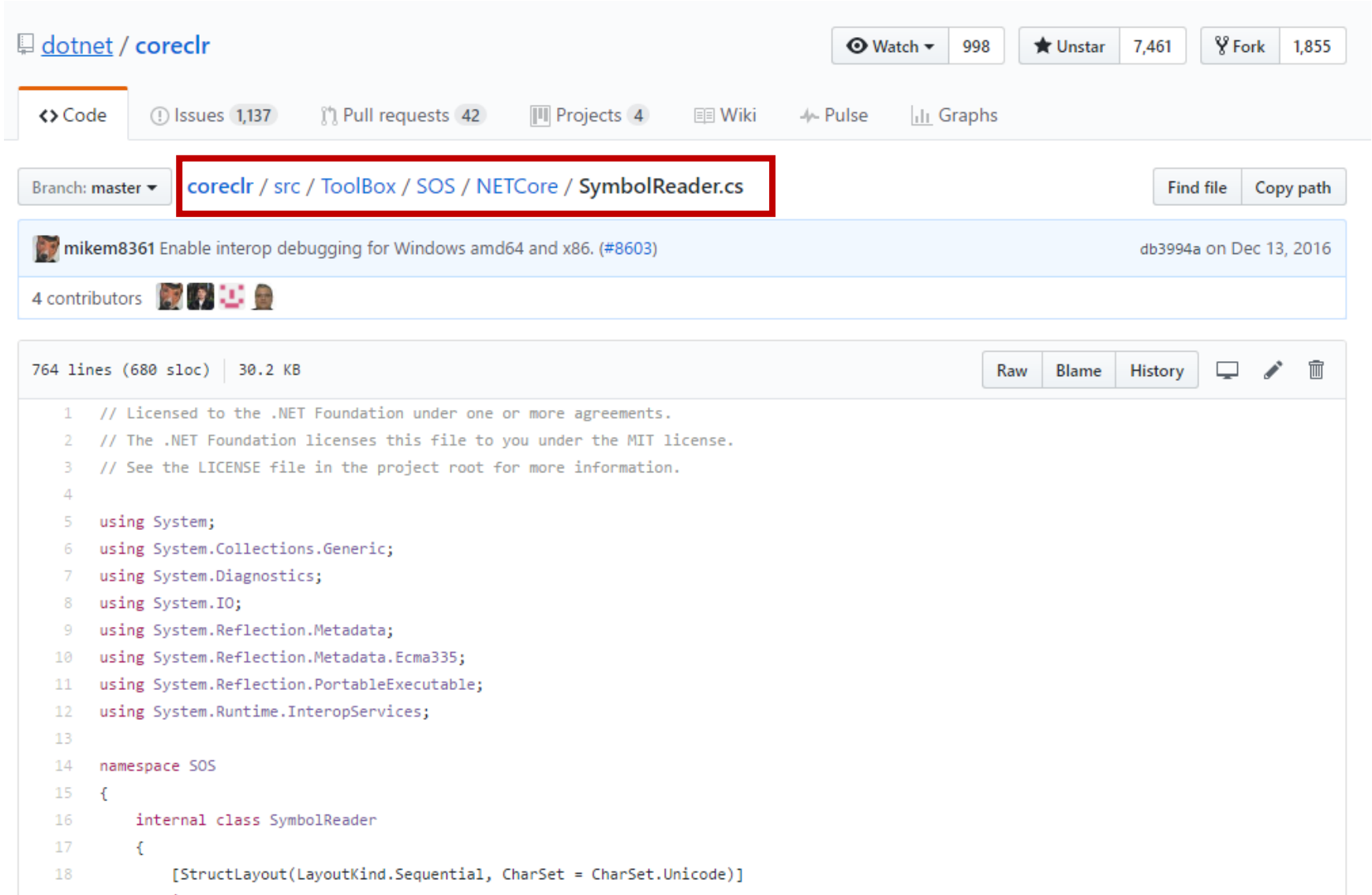
```
{
  "documents": {
     “c:/build/*" : "https://raw.githubusercontent.com/*"
  }
}
```

https://github.com/dotnet/core/blob/master/Documentation/diagnostics/source_link.md

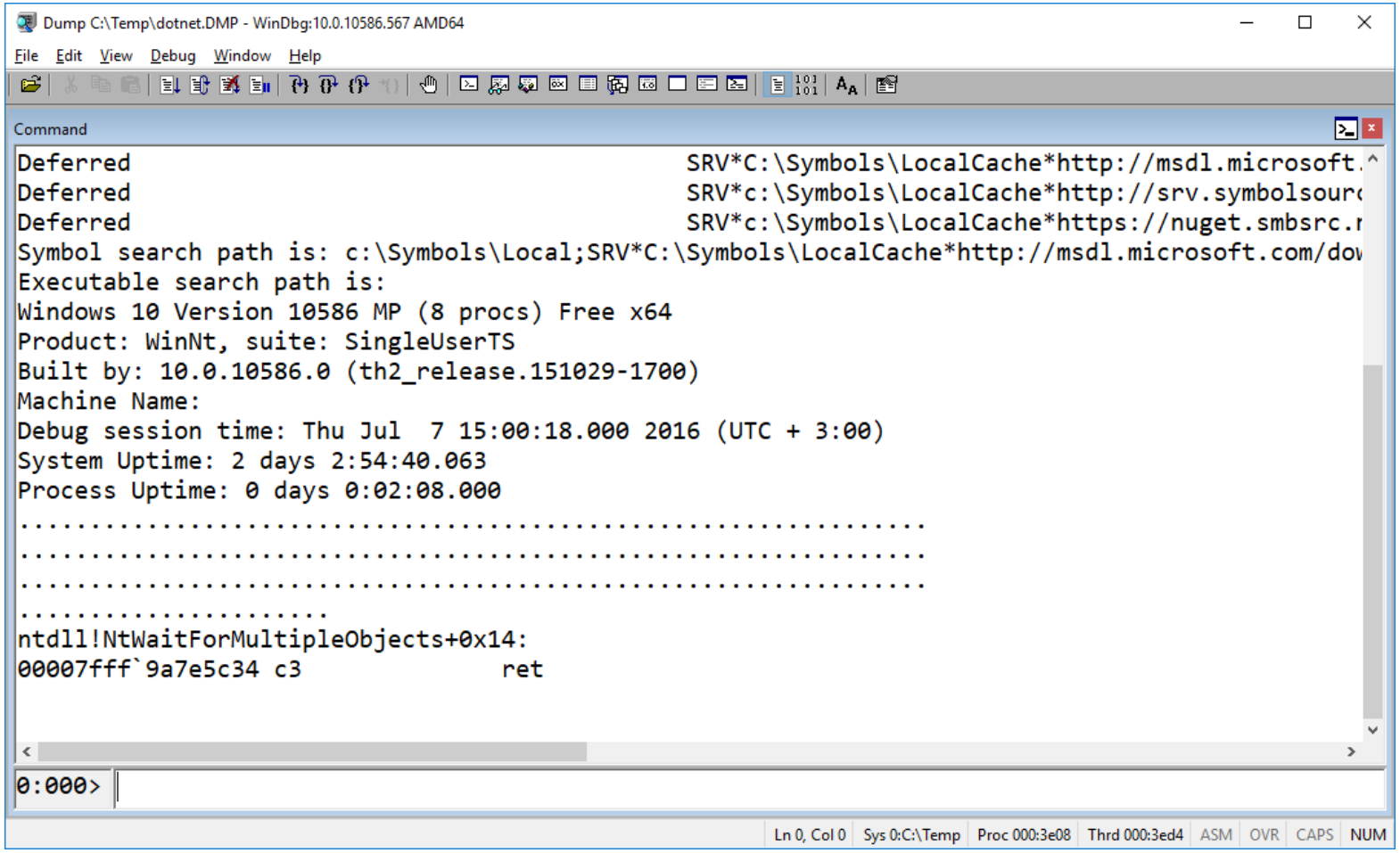
Demo

- PortablePDB настроки в Visual Studio
- Как добавить SourceLink
- .Net API для PortablePDB

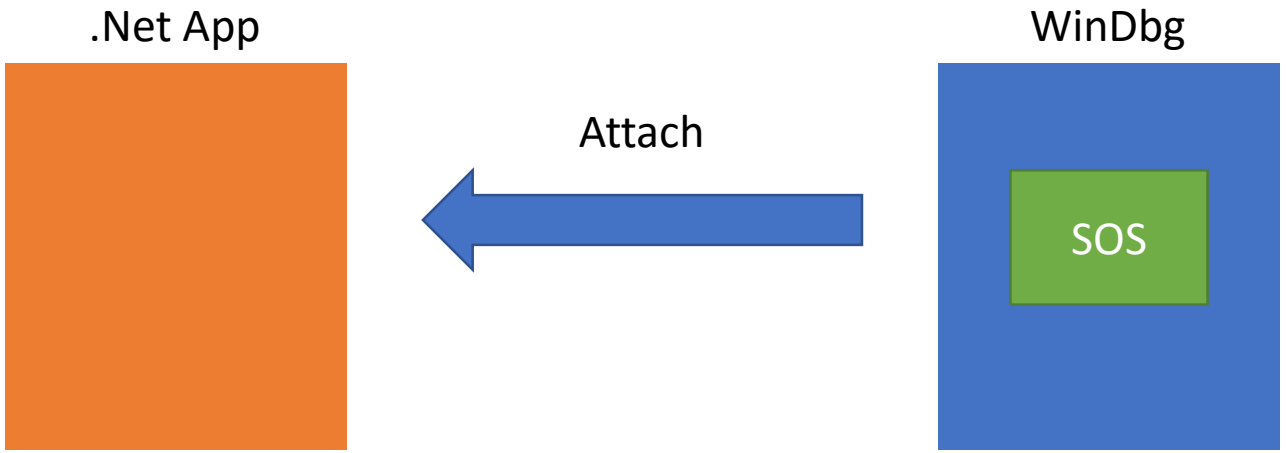
Изменения в отладчике



WinDbg


Dump C:\Temp\dotnet.DMP - WinDbg:10.0.10586.567 AMD64
File Edit View Debug Window Help
Command
Deferred SRV*C:\Symbols\LocalCache*http://msdl.microsoft.
Deferred SRV*c:\Symbols\LocalCache*http://srv.symbolsourc
Deferred SRV*c:\Symbols\LocalCache*https://nuget.smbsrc.r
Symbol search path is: c:\Symbols\Local;SRV*C:\Symbols\LocalCache*http://msdl.microsoft.com/dow
Executable search path is:
Windows 10 Version 10586 MP (8 procs) Free x64
Product: WinNt, suite: SingleUserTS
Built by: 10.0.10586.0 (th2_release.151029-1700)
Machine Name:
Debug session time: Thu Jul 7 15:00:18.000 2016 (UTC + 3:00)
System Uptime: 2 days 2:54:40.063
Process Uptime: 0 days 0:02:08.000
.....
.....
.....
.....
ntdll!NtWaitForMultipleObjects+0x14:
00007fff`9a7e5c34 c3 ret
0:000>
Ln 0, Col 0 Sys 0:C:\Temp Proc 000:3e08 Thrd 000:3ed4 ASM OVR CAPS NUM

Эксперимент



Эксперимент

Эксперимент

Настройка окружения

- Где найти отладочную информацию?

  Качаем Private Symbols с github.com/dotnet/core/

- Где найти исходники?

  Добавляем поток SRCSRV на github.com/dotnet/coreclr/

- Как не повторяться?

  Сохраняем индексированные PDB на Symbol Server

Pid 6328 - WinDbg:10.0.10586.567 AMD64

File Edit View Debug Window Help

https://raw.githubusercontent.com/dotnet/coreclr/1735a1d453677717e68803da6a85284d15dca891/src\ToolBox\SOS\Strike\strike.cpp

```
    {
        ExtOut("Unrecognized argument: %s\n", expression.data);
    }

    return Status;
}

#endif // FEATURE_PAL

DECLARE_API(ClrStack)
{
    INIT_API();

    BOOL bAll = FALSE;
    BOOL bParams = FALSE;
    BOOL bLocals = FALSE;
    BOOL bSuppressLines = FALSE;
    BOOL bICorDebug = FALSE;
    BOOL bGC = FALSE;
    BOOL dml = FALSE;
    BOOL bFull = FALSE;
    BOOL bDisplayRegVals = FALSE;
    DWORD frameToDumpVariablesFor = -1;
    StringHolder cvariableName;
    ArrayHolder<WCHAR> wvariableName = new NOTHROW WCHAR[mdNameLen];
    if (wvariableName == NULL)
    {
        ReportOOM();
        return E_OUTOFMEMORY;
    }

    memset(wvariableName, 0, sizeof(wvariableName));

    size_t nArg = 0;
    CMDOption option[] =
    {   // name, vptr, type, hasValue
        {"-a", &bAll, COBOOL, FALSE},
```

Command

```
0:013> dv
                client = <value unavailable>
                  args = 0x000000bc`3d8fd6c8 ""
                   arg = struct CMDValue [2]
               bParams = 0n0
                 bFull = 0n0
             dmlHolder = class EnableDMLHolder
                   dml = 0n1850522640
                option = struct CMDOption [9]
            bICorDebug = 0n656
     __extensionCleanUp = class __ExtensionCleanUp
                  bAll = 0n0
                  nArg = <value unavailable>
frameToDumpVariablesFor = 0x29cf
               bLocals = 0n0
         cvariableName = struct StringHolder
                 spISD = class ToRelease<ISOSDacInterface>
       bDisplayRegVals = 0n0
                 spIDP = class ToRelease<IXCLRDataProcess>
                Status = 0x00000000
                   bGC = 0n0
        bSuppressLines = 0n0
         wvariableName = class ArrayHolder<unsigned short>
             dmlHolder = <value unavailable>
    firstParamIsNumber = <value unavailable>
                     i = <value unavailable>
          g_ExtSystem = <value unavailable>
```

0:013>

78

Ln 12633, Col 1 | Sys 0:<Local> | Proc 000:18b8 | Thrd 013:21b8 | ASM | OVR | CAPS | NUM

Pid 6328 - WinDbg:10.0.10586.567 AMD64

File Edit View Debug Window Help

https://raw.githubusercontent.com/dotnet/coreclr/1735a1d453677717e68803da6a85284d15dca891/src/ToolBox/SOS/Strike/strike.cpp

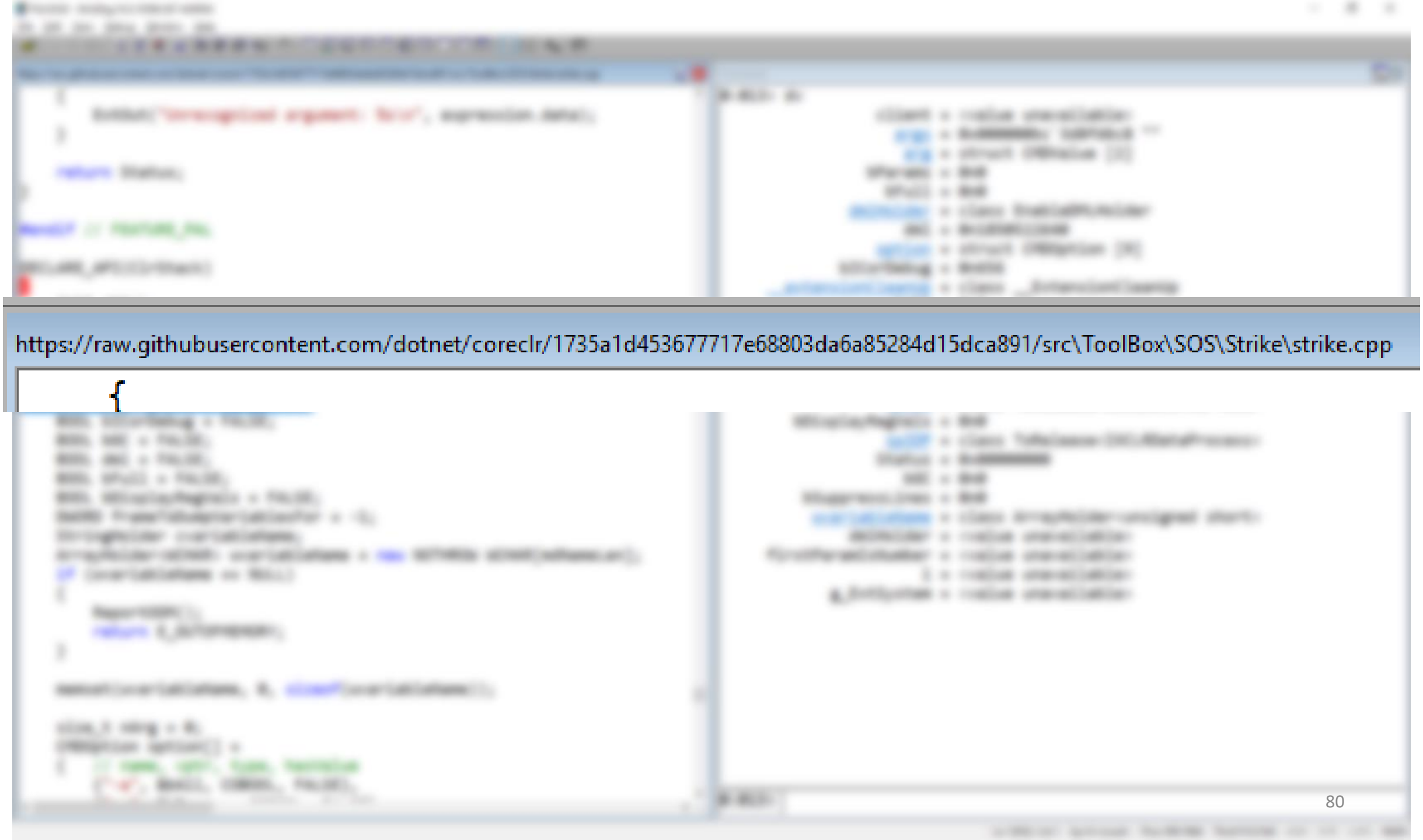
```
    {
        ExtOut("Unrecognized argument: %s\n", expression.data);
    }

    return Status;
}

#endif // FEATURE_PAL

DECLARE_API(ClrStack)
{
    INIT_API();

    BOOL bAll = FALSE;
    BOOL bParams = FALSE;
    BOOL bLocals = FALSE;
    BOOL bSuppressLines = FALSE;
    BOOL bICorDebug = FALSE;
    BOOL bGC = FALSE;
    BOOL dml = FALSE;
    BOOL bFull = FALSE;
    BOOL bDisplayRegVals = FALSE;
    DWORD frameToDumpVariablesFor = -1;
    StringHolder cvariableName;
    ArrayHolder<WCHAR> wvariableName = new NOTHROW WCHAR[mdNameLen];
    if (wvariableName == NULL)
    {
        ReportOOM();
        return E_OUTOFMEMORY;
    }

    memset(wvariableName, 0, sizeof(wvariableName));

    size_t nArg = 0;
    CMDOption option[] =
    {   // name, vptr, type, hasValue
        {"-a", &bAll, COBOOL, FALSE},
```

Command

```
0:013> dv
                client = <value unavailable>
                  args = 0x000000bc`3d8fd6c8 ""
                   arg = struct CMDValue [2]
               bParams = 0n0
                 bFull = 0n0
             dmlHolder = class EnableDMLHolder
                   dml = 0n1850522640
                option = struct CMDOption [9]
            bICorDebug = 0n656
      __extensionCleanUp = class __ExtensionCleanUp
                  bAll = 0n0
                  nArg = <value unavailable>
frameToDumpVariablesFor = 0x29cf
               bLocals = 0n0
         cvariableName = struct StringHolder
                 spISD = class ToRelease<ISOSDacInterface>
        bDisplayRegVals = 0n0
                 spIDP = class ToRelease<IXCLRDataProcess>
                Status = 0x00000000
                  bGC = 0n0
        bSuppressLines = 0n0
         wvariableName = class ArrayHolder<unsigned short>
             dmlHolder = <value unavailable>
    firstParamIsNumber = <value unavailable>
                     i = <value unavailable>
          g_ExtSystem = <value unavailable>
```

0:013>

Ln 12633, Col 1 | Sys 0:<Local> | Proc 000:18b8 | Thrd 013:21b8 | ASM | OVR | CAPS | NUM


https://raw.githubusercontent.com/dotnet/coreclr/1735a1d453677717e68803da6a85284d15dca891/src\ToolBox\SOS\Strike\strike.cpp
{

Изменения в отладчике – Эксперимент

```
 # Call Site
00 SOS_NETCore+0x6200
01 0x00007ffb`bfa4e6f2
02 coreclr!UMThunkStub+0x6a
03 sos!SymbolReader::GetLineByILOffset+0xf2
04 sos!GetLineByOffset+0x1de
05 sos!MethodNameFromIP+0x4cf
06 sos!ClrStackImpl::PrintThread+0x611
07 sos!ClrStackImpl::PrintCurrentThread+0xe0
08 sos!ClrStack+0x91b
09 dbgeng!ExtensionInfo::CallA+0x2a5
0a dbgeng!ExtensionInfo::Call+0x16c
0b dbgeng!ExtensionInfo::CallAny+0x78
0c dbgeng!ParseBangCmd+0x4d1
0d dbgeng!ProcessCommands+0xf48
0e dbgeng!ProcessCommandsAndCatch+0xfc
0f dbgeng!Execute+0x2d4
10 dbgeng!DebugClient::ExecuteWide+0x83
11 windbg!ProcessCommand+0x306
12 windbg!ProcessEngineCommands+0x16c
13 windbg!EngineLoop+0x5ed
14 KERNEL32!BaseThreadInitThunk+0x14
15 ntdll!RtlUserThreadStart+0x21
```

Результат эксперимента

- Загружается CoreCLR и SOS.NETCore
- Создаются делегаты (UMThunk)

  LoadSymbolsForModule, GetLineByILOffset, GetLocalVariableName
- Код для работы с Portable PDB написан на C#

Заключение

#1 Используй _NT_SYMBOL_PATH

#2 Знай свои PDB

#3 Индексируй исходники

#4 Держи PDB рядом

#5 Делись

Q&A

#1 Используй _NT_SYMBOL_PATH

#2 Знай свои PDB

#3 Индексируй исходники

#4 Держи PDB рядом

#5 Делись

dotmailer

Сергей Щегрикович
shchegrikovich@gmail.com
@shchegrikovich