

5 заповедей РДВ

Сергей Щегрикович



shchegrikovich@gmail.com
[@shchegrikovich](https://www.linkedin.com/in/shchegrikovich)

Сколько времени занимает
отладка?

35% - 50%

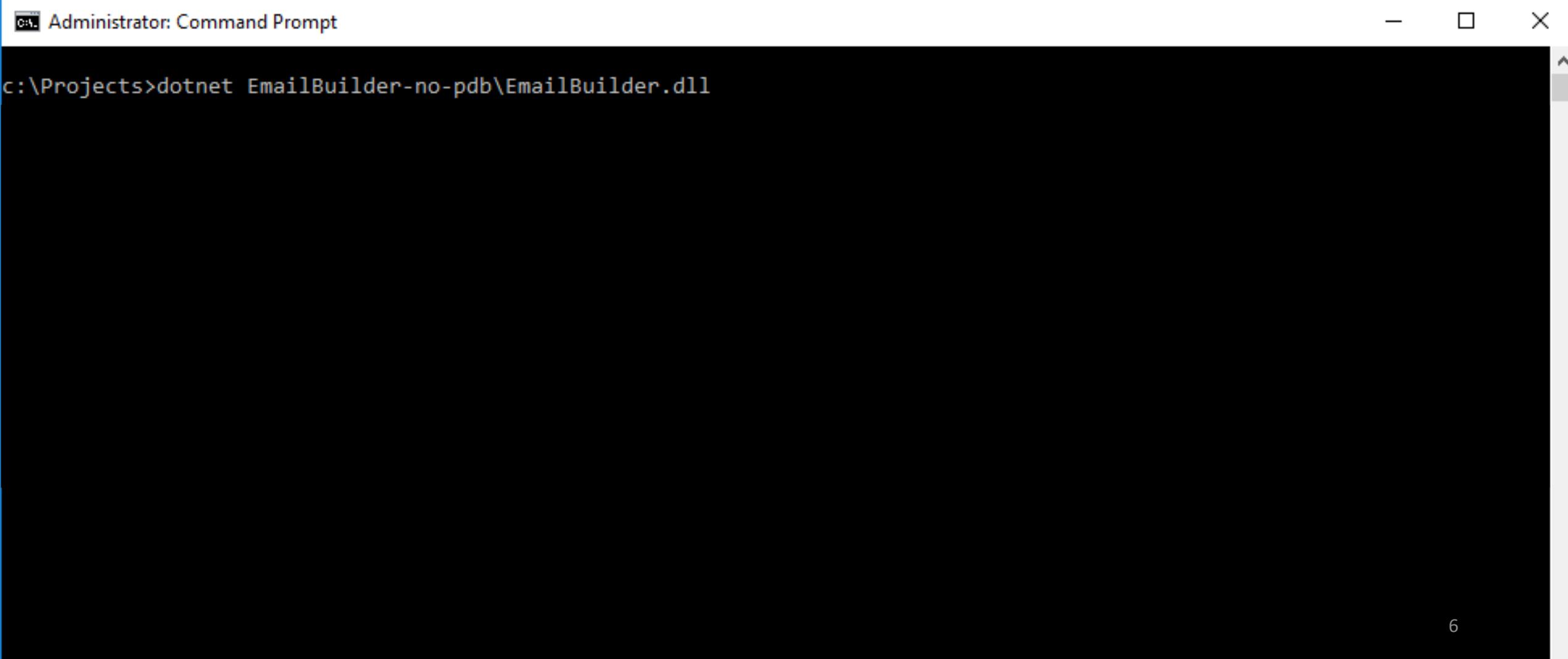
Такой разный дебагинг

- `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 продакшне



Administrator: Command Prompt

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

PDB отсутствуют на продакшне

```
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 dictionary.) ---> 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)
   at EmailBuilder.Program.ProcessEmails()
   at System.Threading.Tasks.Task.Execute()
--- End of inner exception stack trace ---
   at System.Threading.Tasks.Task.ThrowIfExceptional(Boolean includeTaskCanceledExceptions)
   at System.Threading.Tasks.Task.Wait(Int32 millisecondsTimeout, CancellationToken cancellationToken)
   at EmailBuilder.Program.Main(String[] args)

c:\Projects>
```

PDB отсутствуют на продакшне

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

```

PDB не хранят информацию о исходниках

The image shows two side-by-side windows of the JetBrains dotPeek decompiler. Both windows have the title 'BubbleSort.cs' and the file path 'BubbleSort.cs' in the tab bar.

The left window displays the decompiled code with several lines highlighted in red, indicating differences between the decompiled version and the original source. The right window shows the original source code for comparison.

Left Window (Decompiled code):

```
// Decompiled with JetBrains decompiler
// Type: DotNextPdbDemo.BubbleSort
// Assembly: DotNextPdbDemo, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null
// MVID: E24CB94B-E91E-4F22-98D0-A930CA26ED67
// Assembly location: C:\Temp\DotNextPdbDemo.dll

namespace DotNextPdbDemo
{
    public static class BubbleSort
    {
        public static void Sort(int[] array)
        {
            for (int index1 = 0; index1 < array.Length; ++index1)
            {
                for (int index2 = 0; index2 < array.Length - 1; +)
                {
                    if (array[index2] > array[index2 + 1])
                    {
                        int num = array[index2];
                        array[index2] = array[index2 + 1];
                        array[index2 + 1] = num;
                    }
                }
            }
        }
    }
}
```

Right Window (Original source):

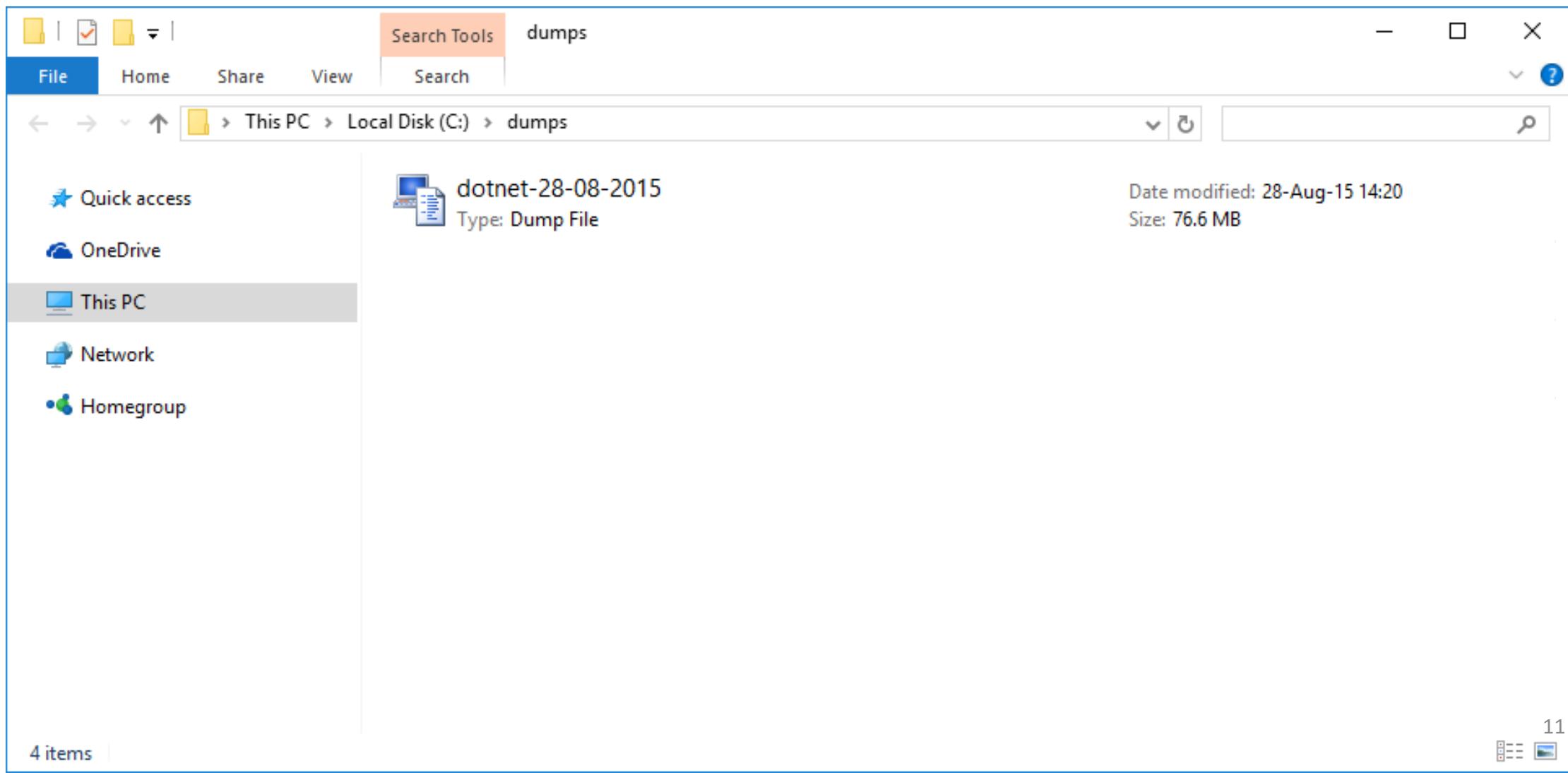
```
namespace DotNextPdbDemo
{
    //https://en.wikipedia.org/wiki/Bubble_sort
    public static class BubbleSort
    {
        public static void Sort(int[] array)
        {
            for (int i = 0; i < array.Length; i++)
            {
                for (int j = 0; j < array.Length - 1; j++)
                {
                    if (array[j] > array[j + 1])
                    {
                        int temp = array[j];
                        array[j] = array[j + 1];
                        array[j + 1] = temp;
                    }
                }
            }
        }
    }
}
```

PDB отсутствуют для сторонних сборок

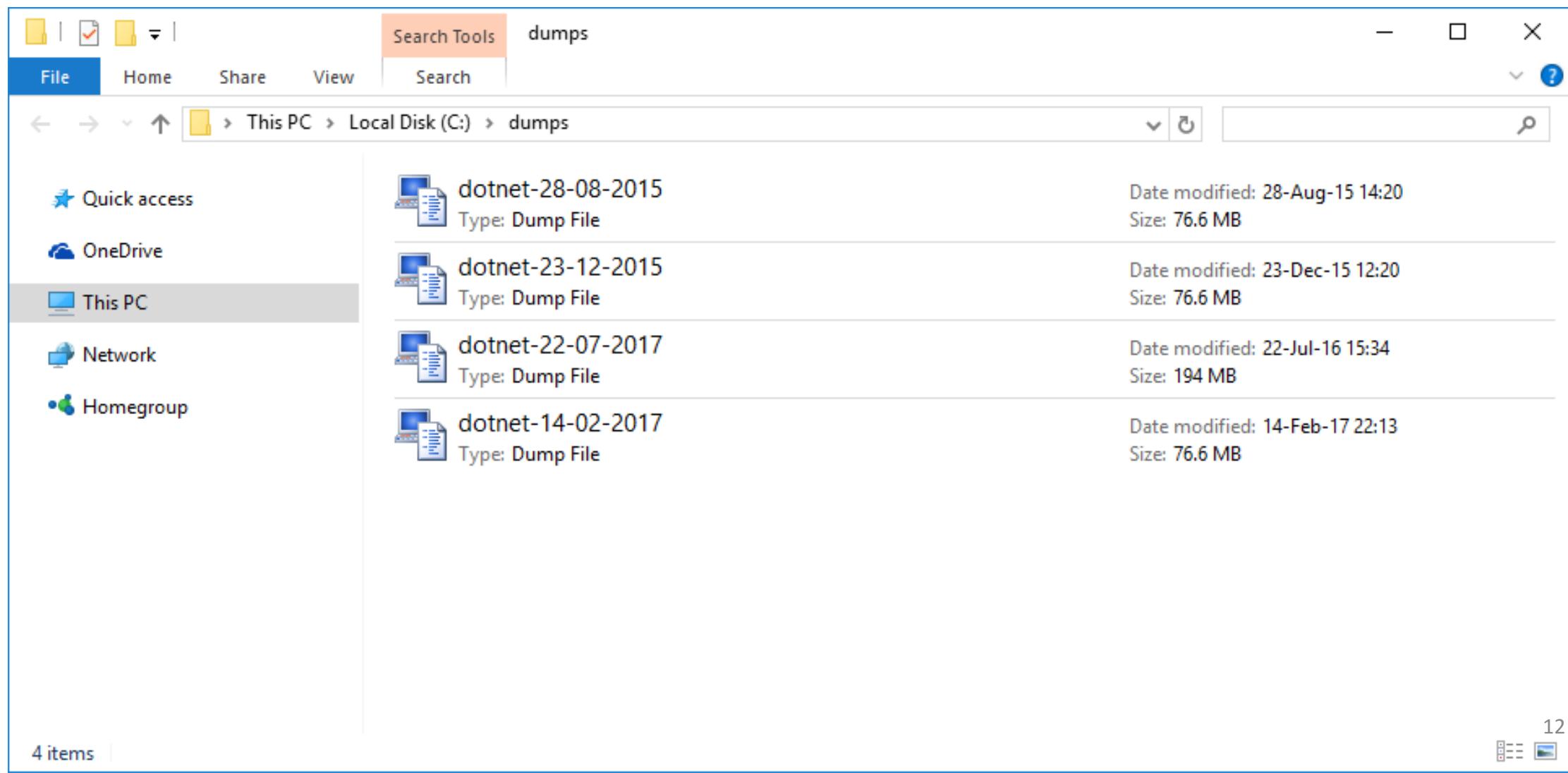
```
Disassembly No Symbols Loaded NuGet: ConsoleApp3 Program.cs
Address: Newtonsoft.Json.JsonPropertyReader(System.IO.TextReader)

017E0615  mov      dword ptr [ecx+18h],edx
017E0618  mov      dword ptr [ecx+1Ch],3
017E061F  mov      dword ptr [ecx+20h],1
017E0626  mov      dword ptr [ecx+24h],edx
017E0629  mov      byte ptr [ecx+2Bh],1
017E062D  test    esi,esi
017E062F  je      017E0642
017E0631  lea     edx,[ecx+48h]
017E0634  call    72D2E6B0
017E0639  mov      dword ptr [ecx+64h],1
017E0640  pop     esi
017E0641  ret
017E0642  mov      ecx,71C80E80h
017E0647  call    016F30F4
017E064C  mov      esi,eax
017E064E  mov      ecx,0B34h
017E0653  mov      edx,1704DFCh
017E0658  call    72DF0A10
017E065D  mov      edx,eax
017E065F  mov      ecx,esi
017E0661  call    71AA4A38
017E0666  mov      ecx,esi
017E0668  call    72ED35E0
017E066D int     3
017E066E  add     byte ptr [eax],al
017E0670  add     byte ptr [eax],al
017E0672  add     byte ptr [eax],al
017E0674  add     byte ptr [eax],al
017E0676  add     byte ptr [eax],al
017E0678  add     byte ptr [eax],al
```

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

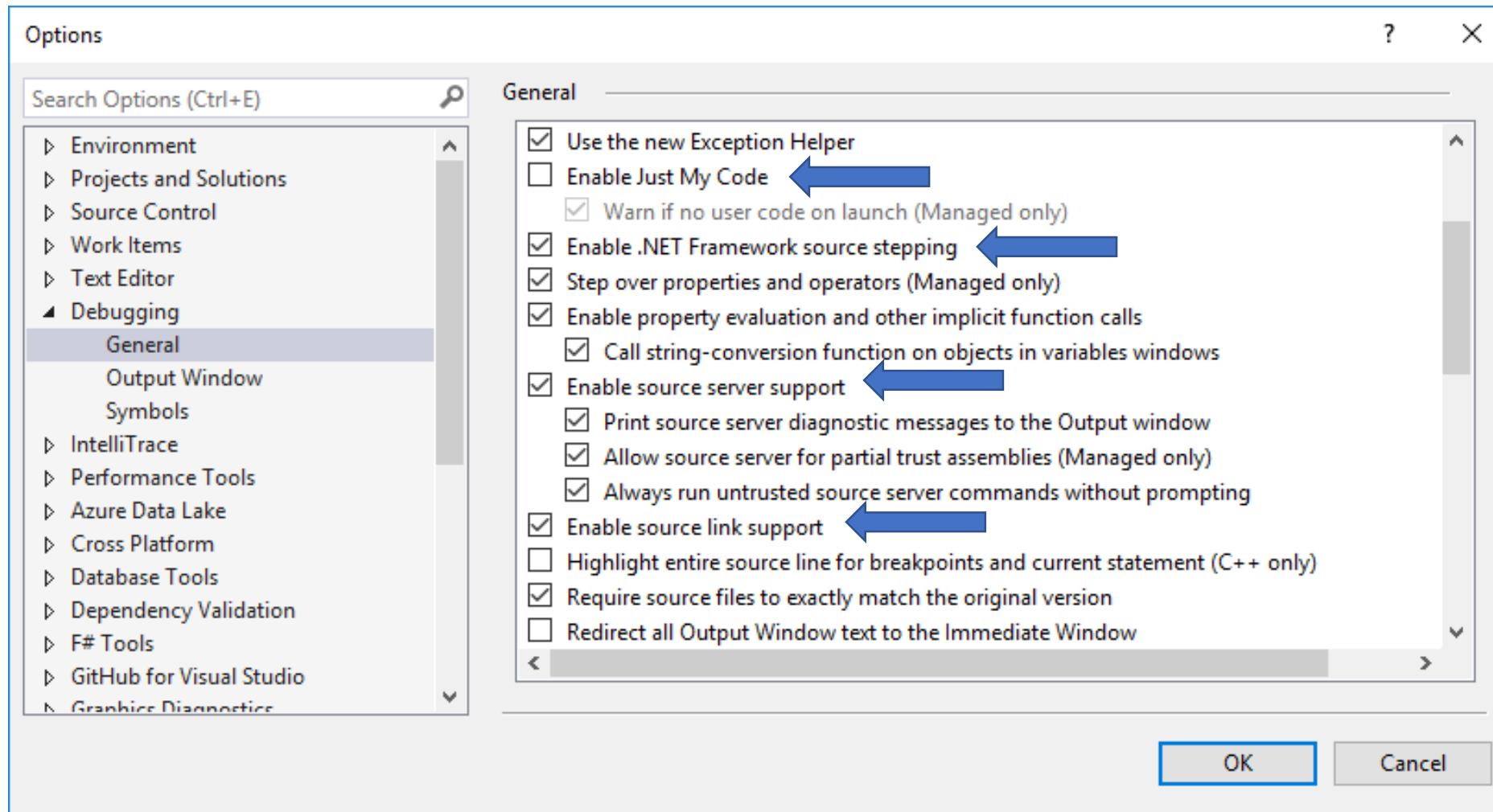
#4 Держи PDB рядом

#5 Делись

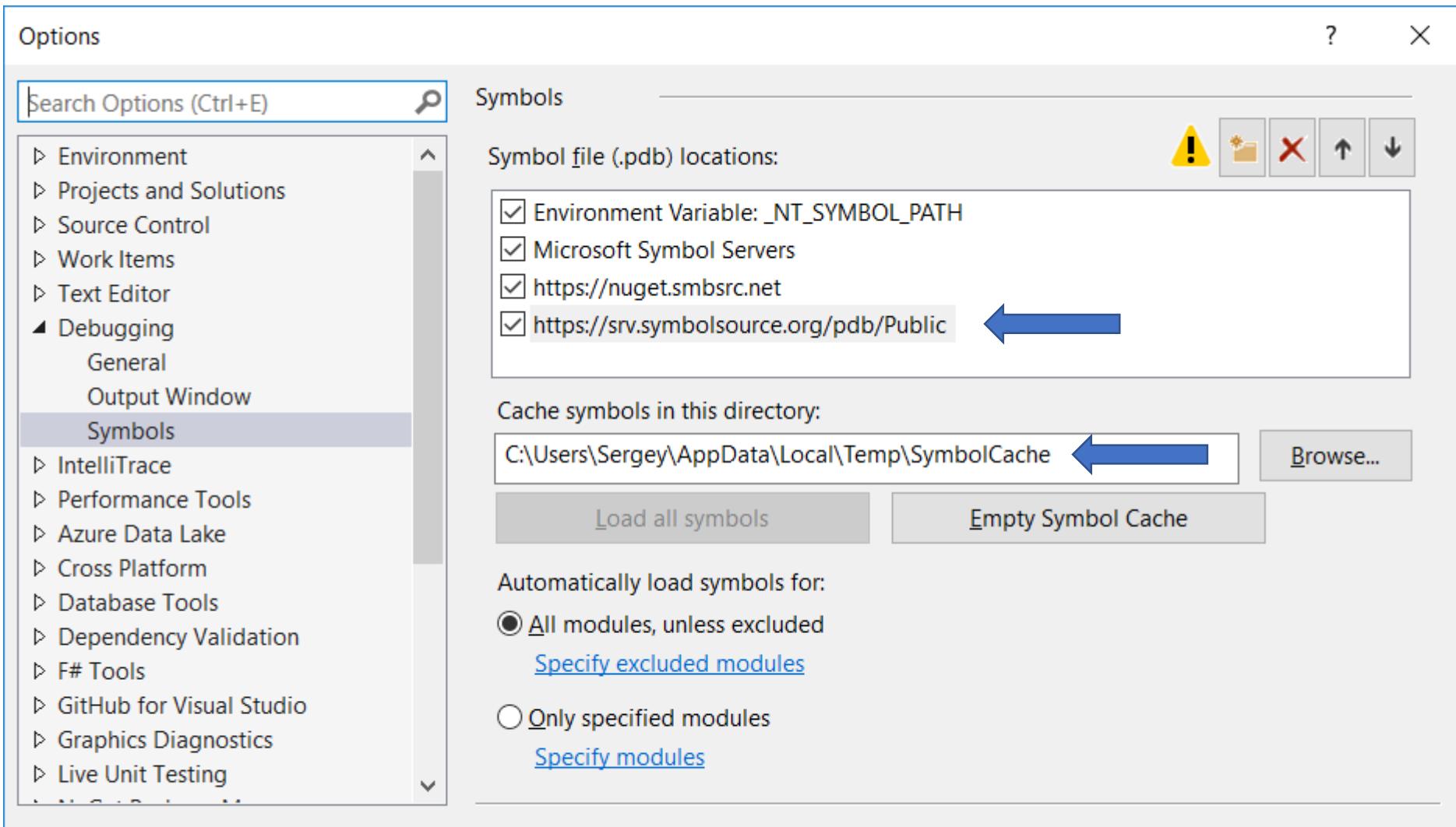
Symbol Server

- <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

Call Stack	
Name	Language
WebApplication3.dll!WebApplication3.Controllers.HomeController.About() Line 35	C#
[External Code]	

Отладка сторонних сборок

Call Stack	
Name	Language
WebApplication3.dll!WebApplication3.Controllers.HomeController.About() Line 35	C#
[External Code]	

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

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;
```

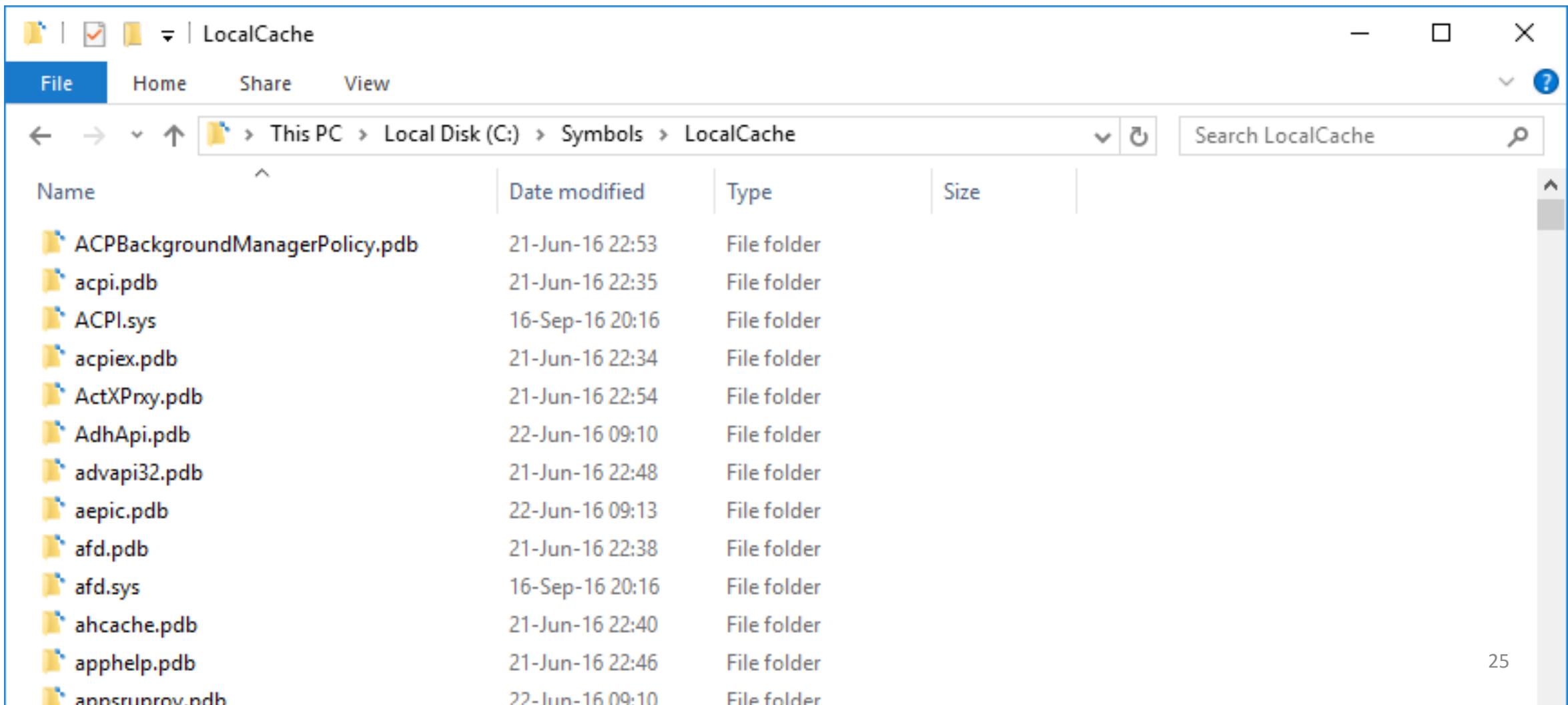
```
cache*C:\Symbols\LocalCache;
```

```
srv*https://nuget.smbsrc.net;
```

```
srv*https://srv.symbolsource.org/pdb/Public;
```

```
srv*https://msdl.microsoft.com/download/symbols;
```

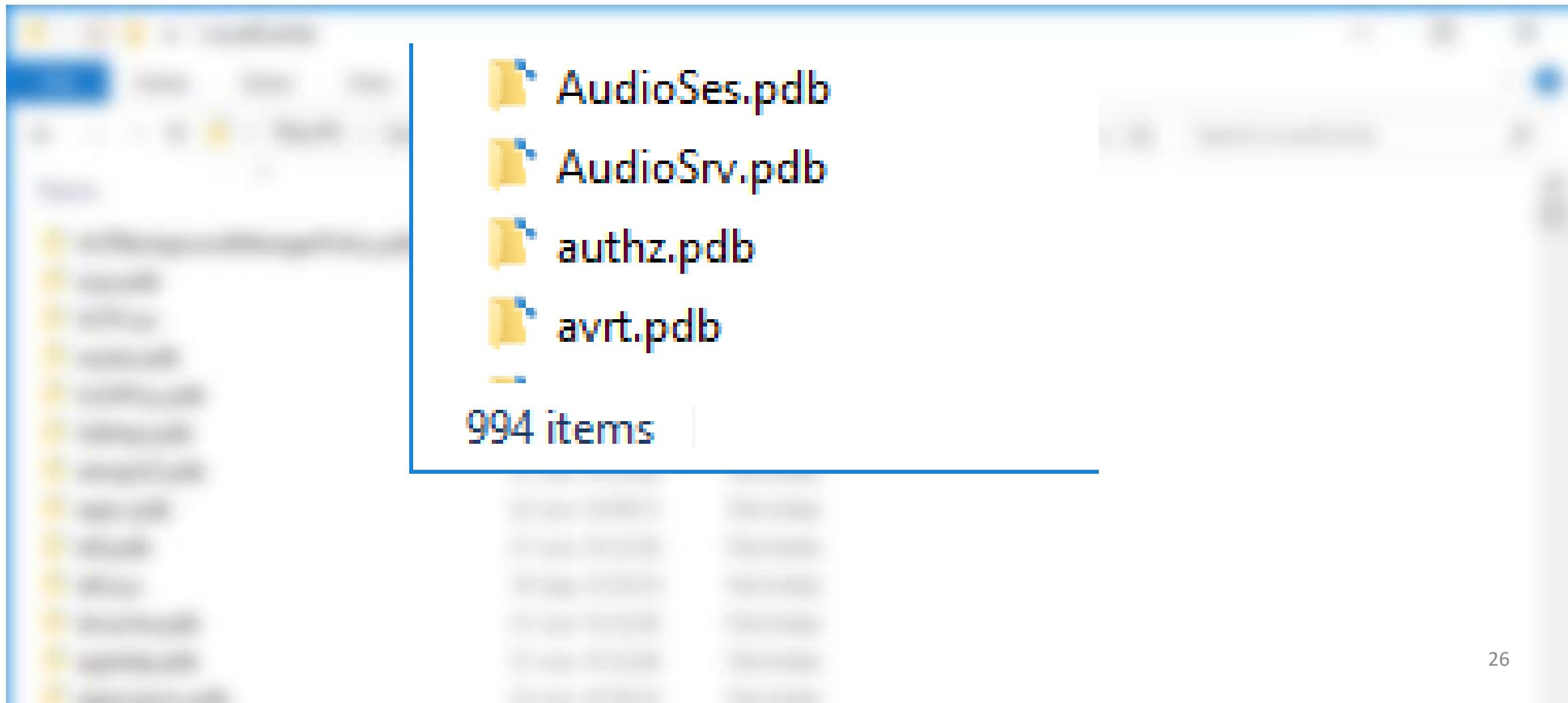
Локальный кеш



The screenshot shows a Windows File Explorer window with the title bar "LocalCache". The menu bar includes "File" (selected), "Home", "Share", and "View". The address bar shows the path: "This PC > Local Disk (C:) > Symbols > LocalCache". The search bar contains "Search LocalCache". The main area displays a list of files and folders in the "LocalCache" folder, sorted by name. The columns are "Name", "Date modified", "Type", and "Size". All items listed are file folders.

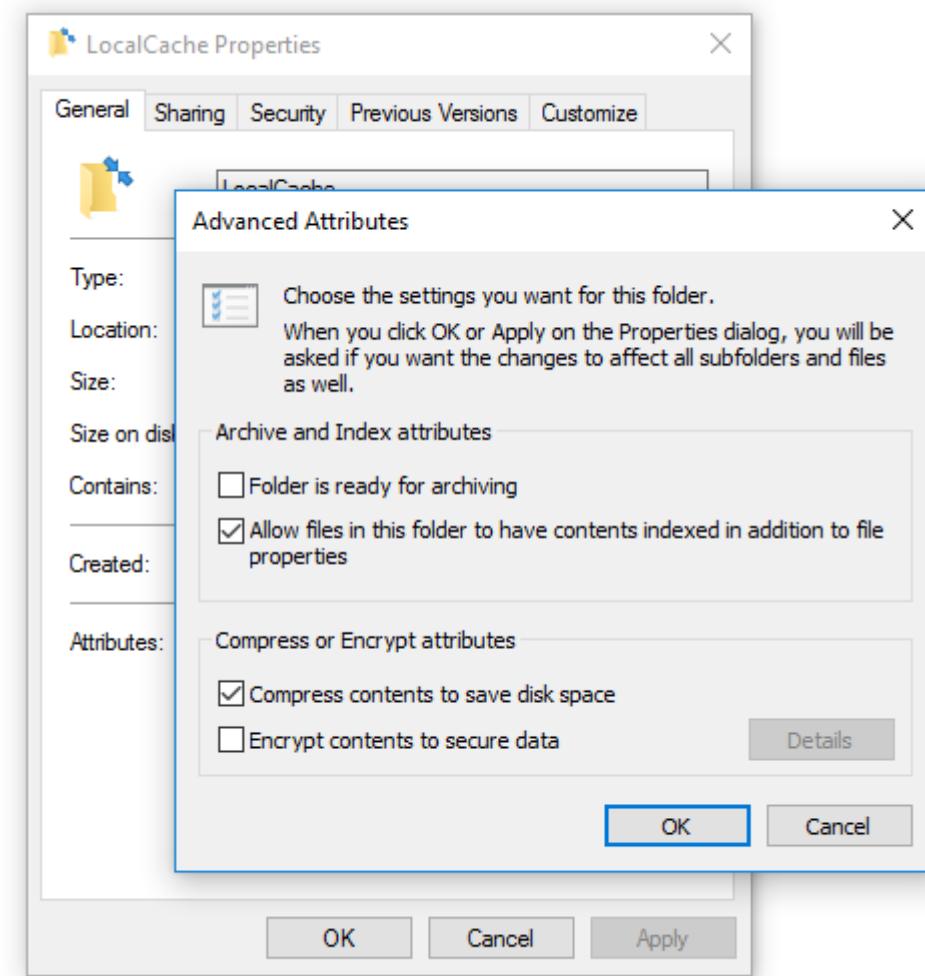
Name	Date modified	Type	Size
ACPBackgroundManagerPolicy.pdb	21-Jun-16 22:53	File folder	
acpi.pdb	21-Jun-16 22:35	File folder	
ACPI.sys	16-Sep-16 20:16	File folder	
acpiex.pdb	21-Jun-16 22:34	File folder	
ActXPrxy.pdb	21-Jun-16 22:54	File folder	
AdhApi.pdb	22-Jun-16 09:10	File folder	
advapi32.pdb	21-Jun-16 22:48	File folder	
aepic.pdb	22-Jun-16 09:13	File folder	
afd.pdb	21-Jun-16 22:38	File folder	
afd.sys	16-Sep-16 20:16	File folder	
ahcache.pdb	21-Jun-16 22:40	File folder	
apphelp.pdb	21-Jun-16 22:46	File folder	
appsrunrov.pdb	22-Jun-16 09:10	File folder	

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

Sasha Goldshtain (@goldshtn) posted on October 6, 2016, at 3:24 PM. The tweet reads: "Turns out that putting #Azure CDN in front of the Microsoft public symbol server speeds up downloads by 3-5x, depending on the file size." The post has 2 retweets and 5 likes. The tweet card includes a "Following" button and a dropdown menu.

Turns out that putting #Azure CDN in front of the Microsoft public symbol server speeds up downloads by 3-5x, depending on the file size.

RETWEETS 2 LIKES 5

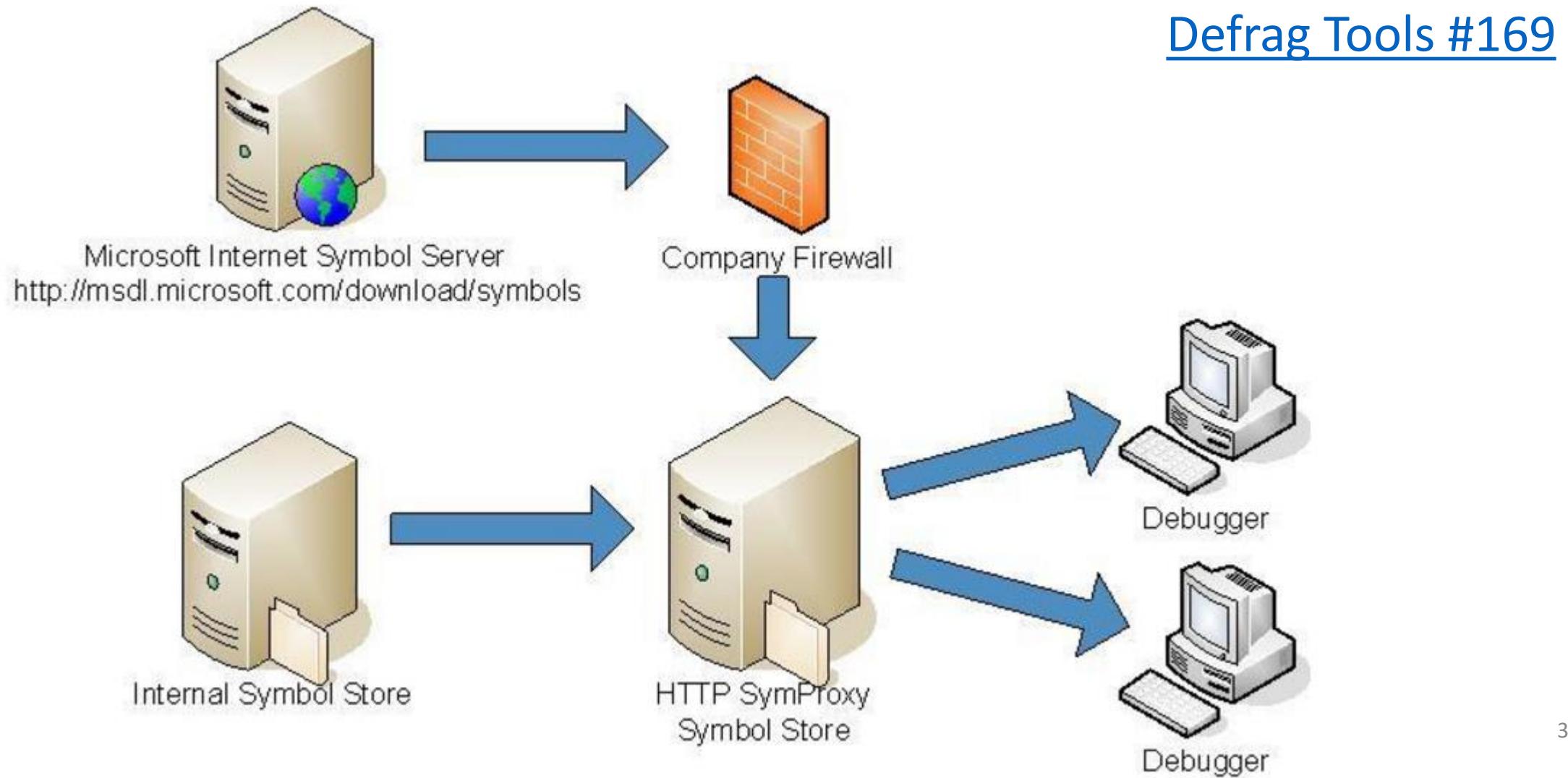
3:24 PM - 6 Oct 2016

1 2 5

Symbol Server Hell решение

- Public CDN
- SymProxy

SymProxy OT Andrew Richards



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

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>c>(Microsoft.AspNetCore.Mvc.Internal.ControllerActionInvoker)

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 Microsoft.AspNetCore.Mvc.Internal.ControllerActionInvoker)

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>c>(Microsoft.AspNetCore.Mvc.Internal.ControllerActionInvoker)

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 Microsoft.AspNetCore.Mvc.Internal.ControllerActionInvoker)

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>c>(Microsoft.AspNetCore.Mvc.Internal.ControllerActionInvoker)

Microsoft.AspNetCore.Mvc.Core.dll!Microsoft.AspNetCore.Mvc.Internal.ControllerActionInvoker.InvokeNextActionFilterAsync()

Microsoft.AspNetCore.Mvc.Core.dll!Microsoft.AspNetCore.Mvc.Internal.ControllerActionInvoker.InvokeNextActionFilterAwaitedAsync()

#2 Знай свои РДВ

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\...
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...
0a coreclr!coreclr_execute_assembly+0xde [e:\src\dlls\mscoree\...
```

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>c>(Microsoft.AspNetCore.Mvc.Core.dll!Microsoft.AspNetCore.Mvc.Internal.ControllerActionInvoker)

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 Microsoft.AspNetCore.Mvc.Core.dll!Microsoft.AspNetCore.Mvc.Internal.ControllerActionInvoker)

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>c>(Microsoft.AspNetCore.Mvc.Core.dll!Microsoft.AspNetCore.Mvc.Internal.ControllerActionInvoker)

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 Microsoft.AspNetCore.Mvc.Core.dll!Microsoft.AspNetCore.Mvc.Internal.ControllerActionInvoker)

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>c>(Microsoft.AspNetCore.Mvc.Core.dll!Microsoft.AspNetCore.Mvc.Internal.ControllerActionInvoker)

Microsoft.AspNetCore.Mvc.Core.dll!Microsoft.AspNetCore.Mvc.Internal.ControllerActionInvoker.InvokeNextActionFilterAsync()

Microsoft.AspNetCore.Mvc.Core.dll!Microsoft.AspNetCore.Mvc.Internal.ControllerActionInvoker.InvokeNextActionFilterAwaitedAsync()

symchk.exe

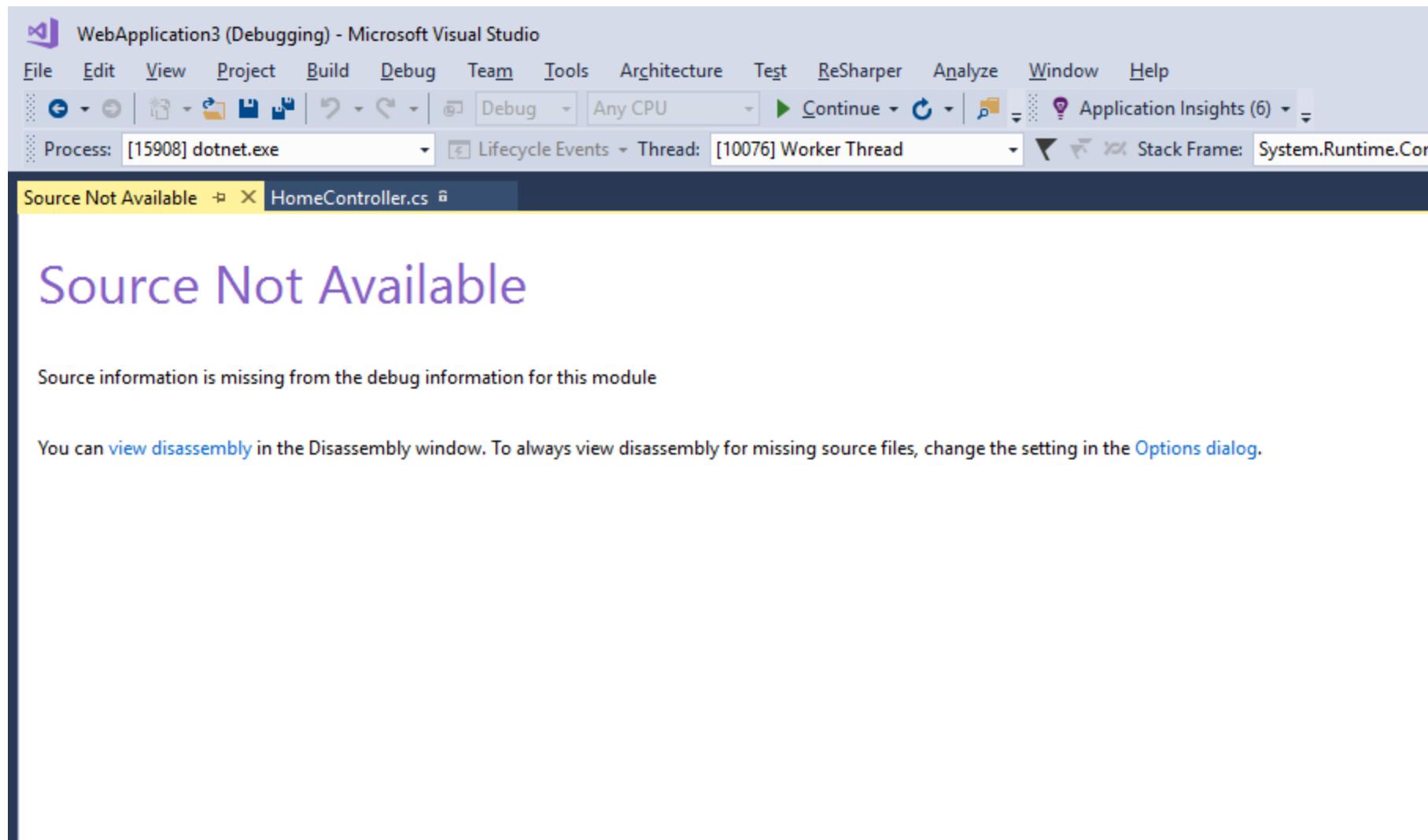
```
>symchk /v Microsoft.AspNetCore.Mvc.Core.dll
[SYMCHK] PDB Sig: 0
[SYMCHK] PDB7 Sig: {64474D7C-1F31-4762-8E61-0FD4C18EAD6A}
[SYMCHK] Age: 1
[SYMCHK] PDB Matched: TRUE
[SYMCHK] DBG Matched: TRUE
[SYMCHK] Line numbers: FALSE
[SYMCHK] Global syms: FALSE
[SYMCHK] Type Info: FALSE
[SYMCHK] -----
```

symchk.exe

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

```
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 
```

```
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
```

```
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 specific 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 настройки в Visual Studio
- Как добавить SourceLink
- .Net API для PortablePDB

Изменения в отладчике

The screenshot shows a GitHub repository page for 'dotnet / coreclr'. The 'Code' tab is selected. A red box highlights the path 'coreclr / src / ToolBox / SOS / NETCore / SymbolReader.cs'. The commit details show a pull request by 'mikem8361' to enable interop debugging for Windows amd64 and x86, with a timestamp of 'db3994a on Dec 13, 2016'. Below the commit, it says '4 contributors'. The code listing shows the beginning of the 'SymbolReader' class definition.

dotnet / coreclr

Watch 998 Unstar 7,461 Fork 1,855

Code Issues 1,137 Pull requests 42 Projects 4 Wiki Pulse Graphs

Branch: master coreclr / src / ToolBox / SOS / NETCore / SymbolReader.cs Find file Copy path

mikem8361 Enable interop debugging for Windows amd64 and x86. (#8603) db3994a on Dec 13, 2016

4 contributors

764 lines (680 sloc) 30.2 KB Raw Blame History

```
1 // Licensed to the .NET Foundation under one or more agreements.
2 // The .NET Foundation licenses this file to you under the MIT license.
3 // See the LICENSE file in the project root for more information.
4
5 using System;
6 using System.Collections.Generic;
7 using System.Diagnostics;
8 using System.IO;
9 using System.Reflection.Metadata;
10 using System.Reflection.Metadata.Ecma335;
11 using System.Reflection.PortableExecutable;
12 using System.Runtime.InteropServices;
13
14 namespace SOS
15 {
16     internal class SymbolReader
17     {
18         [StructLayout(LayoutKind.Sequential, CharSet = CharSet.Unicode)]
```

WinDbg

Dump C:\Temp\dotnet.DMP - WinDbg:10.0.10586.567 AMD64

File Edit View Debug Window Help

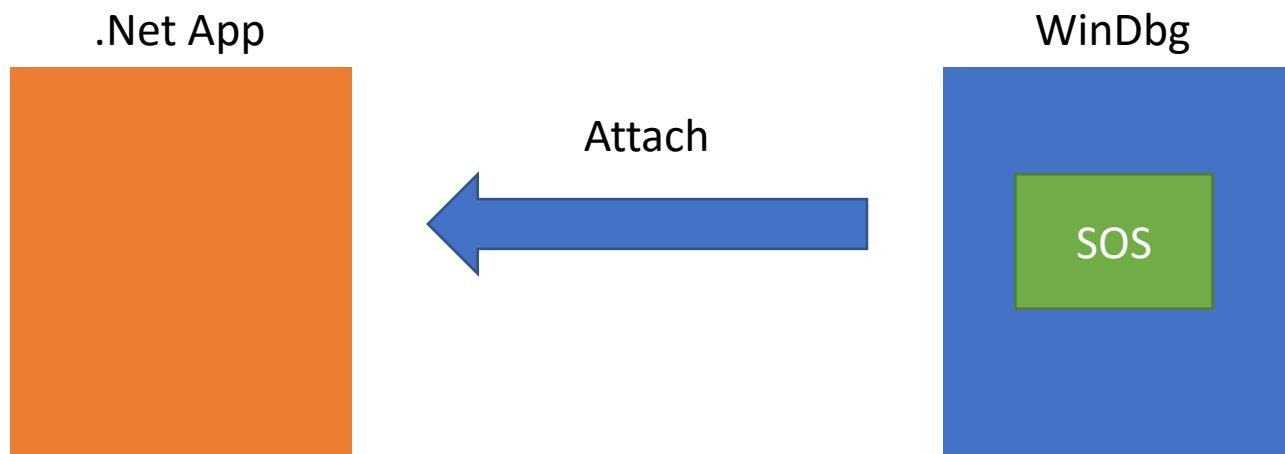
Command

```
Deferred SRV*C:\Symbols\LocalCache*http://msdl.microsoft.com/symbols
Deferred SRV*c:\Symbols\LocalCache*http://srv.symbolsource.org/symbols
Deferred SRV*c:\Symbols\LocalCache*https://nuget.smbsrc.r.appspot.com/symbols
Symbol search path is: c:\Symbols\Local;SRV*C:\Symbols\LocalCache*http://msdl.microsoft.com/symbols
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

```

{
    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},
        {"-c", &bParams, COBOOL, FALSE},
        {"-l", &bLocals, COBOOL, FALSE},
        {"-f", &bFull, COBOOL, FALSE},
        {"-d", &bDisplayRegVals, COBOOL, FALSE},
        {"-g", &bGC, COBOOL, FALSE},
        {"-m", &dml, COBOOL, FALSE},
        {"-s", &bSuppressLines, COBOOL, FALSE},
        {"-r", &nArg, COINT32, 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

```

{
    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},
        {"-n", &nArg, COINT32, 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>

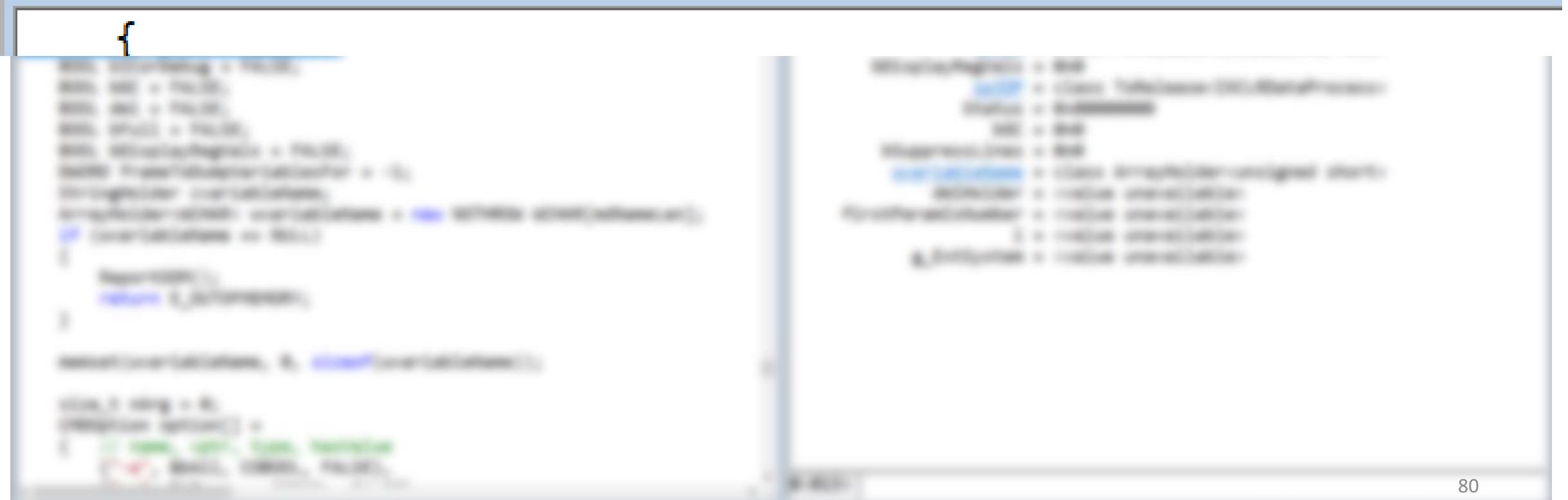
```

0:013>

79



<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!IClrStackImpl::PrintThread+0x611
07 sos!IClrStackImpl::PrintCurrentThread+0xe0
08 sos!IClrStack+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



Сергей Щегрикович
shchegrikovich@gmail.com
[@shchegrikovich](https://twitter.com/shchegrikovich)