Sharing common assemblies between processes and domains

Article: https://habrahabr.ru/post/341950/

Source code: https://github.com/sflusov/IISSharingAssemblies

The presentation target

- Optimization path
- Tools for analyzing

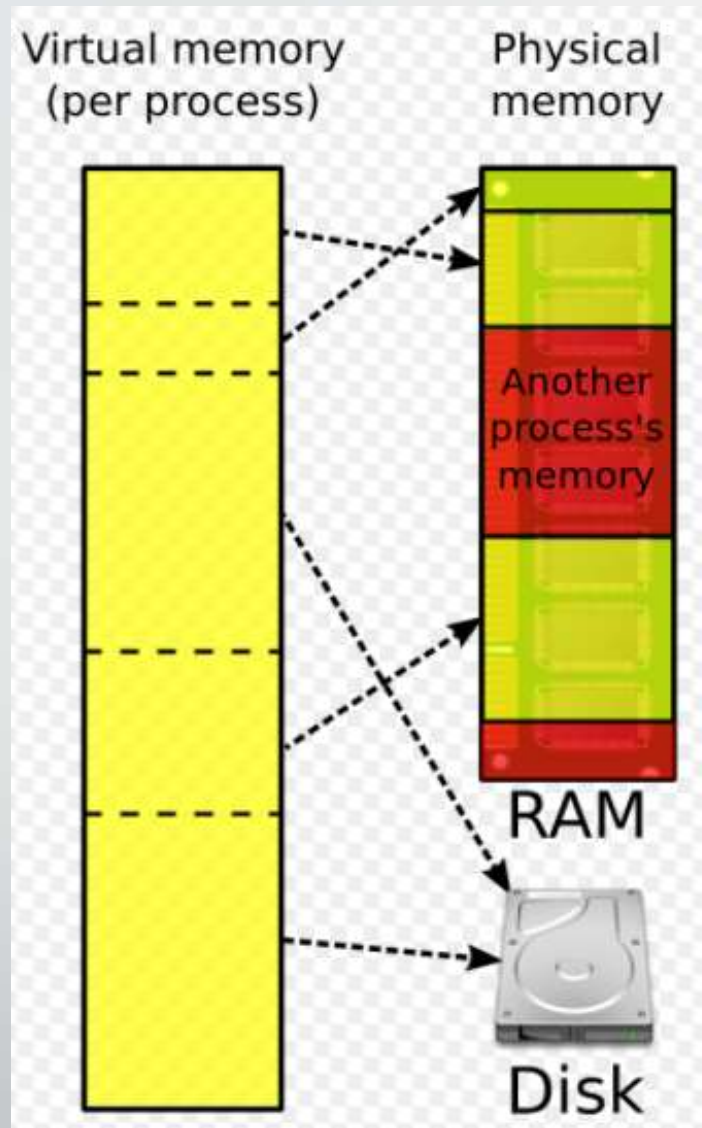# Used tools

- Sysinternals Process Explorer v16.21
- Sysinternals VMMap v3.21

Agenda

- Memory structure and types

- Sharing assemblies between application domains

- Sharing assemblies between processes

- Real example

What do you really mean by memory?

Virtual memory:
- 32-bit - 2GB per process
- 64-bit – 128TB per process

See also:
- http://blogs.microsoft.co.il/sasha/2016/01/05/windows-process-memory-usage-demystified/

4

What do you really mean by memory?

Working Set = WS Private + WS Shareable

What do you really mean by memory?
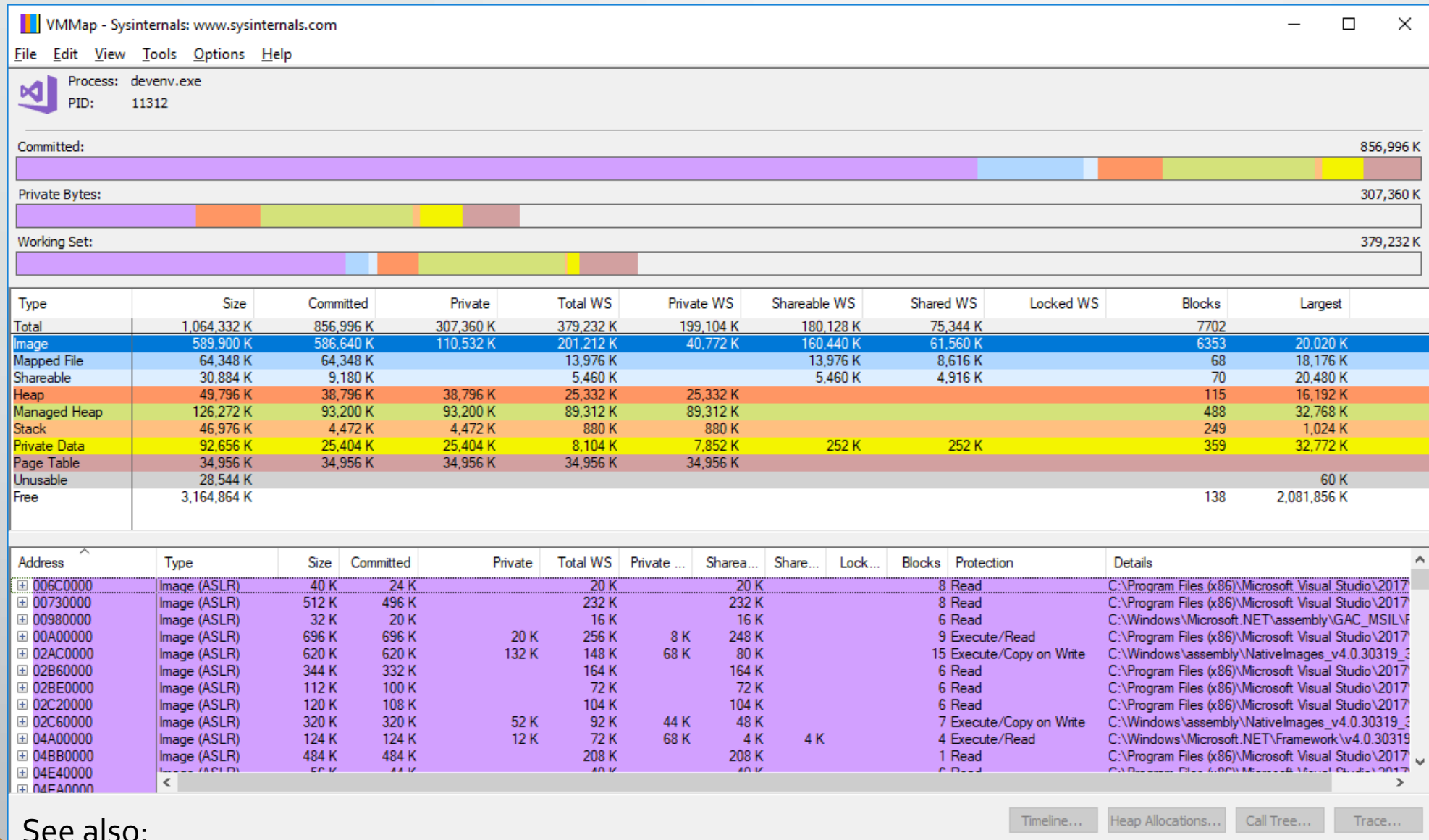
See also:
- http://blogs.microsoft.co.il/sasha/2016/01/05/windows-process-memory-usage-demystified/
- Windows® Sysinternals Administrator's Reference. Pages 216-218

What do you really mean by memory?

Committed:		856,996 K

Private Bytes:		307,360 K

Working Set:		379,232 K

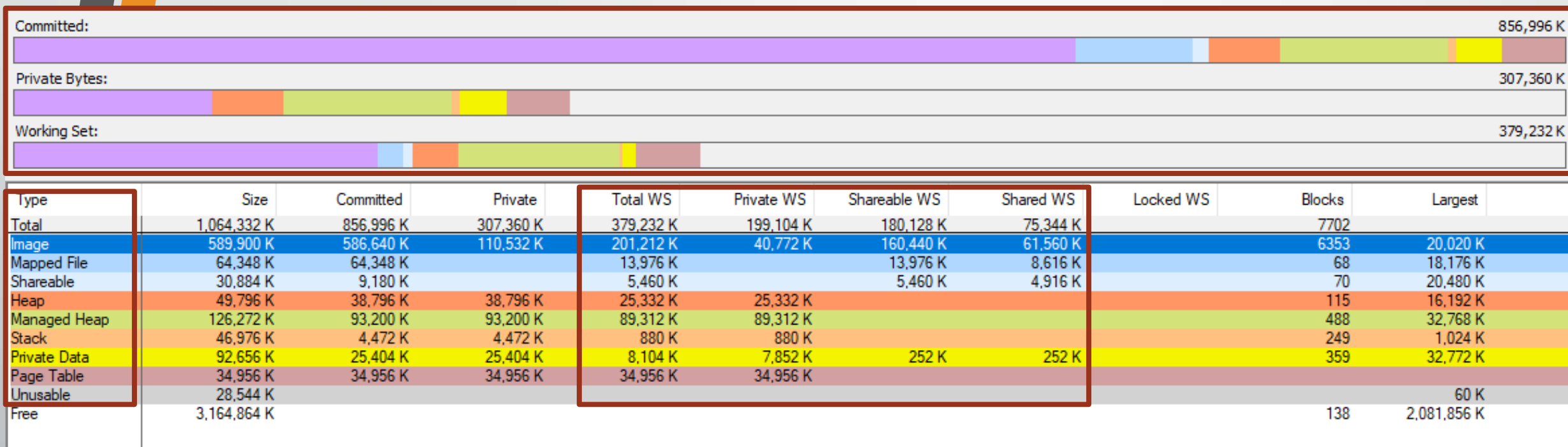| Type	Size	Committed	Private	Total WS	Private WS	Shareable WS	Shared WS	Locked WS	Blocks	Largest
Total	1,064,332 K	856,996 K	307,360 K	379,232 K	199,104 K	180,128 K	75,344 K		7702	
Image	589,900 K	586,640 K	110,532 K	201,212 K	40,772 K	160,440 K	61,560 K		6353	20,020 K
Mapped File	64,348 K	64,348 K		13,976 K		13,976 K	8,616 K		68	18,176 K
Shareable	30,884 K	9,180 K		5,460 K		5,460 K	4,916 K		70	20,480 K
Heap	49,796 K	38,796 K	38,796 K	25,332 K	25,332 K				115	16,192 K
Managed Heap	126,272 K	93,200 K	93,200 K	89,312 K	89,312 K				488	32,768 K
Stack	46,976 K	4,472 K	4,472 K	880 K	880 K				249	1,024 K
Private Data	92,656 K	25,404 K	25,404 K	8,104 K	7,852 K	252 K	252 K		359	32,772 K
Page Table	34,956 K	34,956 K	34,956 K	34,956 K	34,956 K					
Unusable	28,544 K									60 K
Free	3,164,864 K								138	2,081,856 K

See also:

- http://blogs.microsoft.co.il/sasha/2016/01/05/windows-process-memory-usage-demystified/
- Windows® Sysinternals Administrator's Reference. Pages 216-218

VMMap - Sysinternals: www.sysinternals.com

File   Edit   View   Tools   Options   Help

Process: w3wp.exe
PID: 2612

Committed: 345,084 K

Private Bytes: 178,852 K

Working Set: 220,088 K

Type	Size	Committed	Private	Total WS	Private WS	Shareable WS	Shared WS	Locked WS	Blocks	Largest
Total	9,333,852 K	345,084 K	178,852 K	220,088 K	162,852 K	57,236 K	57,152 K		2365	
Managed Heap	8,953,216 K	73,644 K	73,644 K	73,436 K	73,436 K				664	8,912,896 K
Heap	75,824 K	65,732 K	65,668 K	60,072 K	60,068 K	4 K	4 K		73	16,192 K
Image	160,448 K	158,776 K	2,708 K	58,164 K	2,148 K	56,016 K	55,940 K		1407	21,664 K
Page Table	18,580 K	18,580 K	18,580 K	18,580 K	18,580 K					
Private Data	89,668 K	16,708 K	16,708 K	8,028 K	8,024 K	4 K	4 K		100	78,400 K
Mapped File	4,052 K	4,052 K		632 K		632 K	632 K		7	2,900 K
Stack	14,336 K	1,544 K	1,544 K	596 K	596 K				88	512 K
Shareable	9,420 K	6,048 K		580 K		580 K	572 K		26	3,820 K
Unusable	8,308 K									60 K
Free	137,429,638,136 K								75	136,878,830,968 K

Address	Type	Size	Committed	Private	Total WS	Private ...	Sharea...	Share...	Lock...	Blocks	Protection	Details
⊞ 00000079C3FC0000	Managed Heap	8,912,8...	35,480 K	35,480 K	35,272 K	35,272 K				8	Read/Write	GC
⊞ 00007FFB28B80000	Managed Heap	64 K	48 K	48 K	48 K	48 K				6	Execute/Read/Write	Shared Domain
⊞ 00007FFB28B90000	Managed Heap	64 K	36 K	36 K	36 K	36 K				2	Read/Write	Domain 1
⊞ 00007FFB28C30000	Managed Heap	448 K	192 K	192 K	192 K	192 K				5	Execute/Read/Write	Shared Domain Virtual Call Stub
⊞ 00007FFB28CA0000	Managed Heap	384 K	384 K	384 K	384 K	384 K				1	Read/Write	Shared Domain Low Frequency Heap
⊞ 00007FFB28D00000	Managed Heap	64 K	64 K	64 K	64 K	64 K				1	Read/Write	Shared Domain Low Frequency Heap
⊞ 00007FFB28D10000	Managed Heap	64 K	64 K	64 K	64 K	64 K				1	Read/Write	Shared Domain High Frequency Heap
⊞ 00007FFB28D20000	Managed Heap	64 K	28 K	28 K	28 K	28 K				2	Read/Write	Shared Domain
⊞ 00007FFB28D30000	Managed Heap	64 K	64 K	64 K	64 K	64 K				1	Read/Write	Shared Domain Low Frequency Heap
⊞ 00007FFB28D40000	Managed Heap	64 K	64 K	64 K	64 K	64 K				1	Read/Write	Shared Domain Low Frequency Heap
⊞ 00007FFB28D90000	Managed Heap	64 K	64 K	64 K	64 K	64 K				1	Read/Write	Shared Domain High Frequency Heap
⊞ 00007FFB28DA0000	Managed Heap	64 K	52 K	52 K	52 K	52 K				2	Read/Write	Shared Domain
⊞ 00007FFB28DB0000	Managed Heap	64 K	12 K	12 K	12 K	12 K				2	Read/Write	Shared Domain
⊞ 00007FFB28DC0000	Managed Heap	384 K	384 K	384 K	384 K	384 K				1	Read/Write	Shared Domain Low Frequency Heap
⊞ 00007FFB28E20000	Managed Heap	64 K	64 K	64 K	64 K	64 K				1	Read/Write	Shared Domain Low Frequency Heap
⊞ 00007FFB28E30000	Managed Heap	64 K	48 K	48 K	48 K	48 K				2	Read/Write	Shared Domain
⊞ 00007FFB28E40000	Managed Heap	64 K	36 K	36 K	36 K	36 K				2	Read/Write	Shared Domain
⊞ 00007FFB28E50000	Managed Heap	64 K	64 K	64 K	64 K	64 K				1	Read/Write	Shared Domain Low Frequency Heap

Timeline...   Heap Allocations...   Call Tree...   Trace...

The structure of typical application

Sharing assemblies between application domains
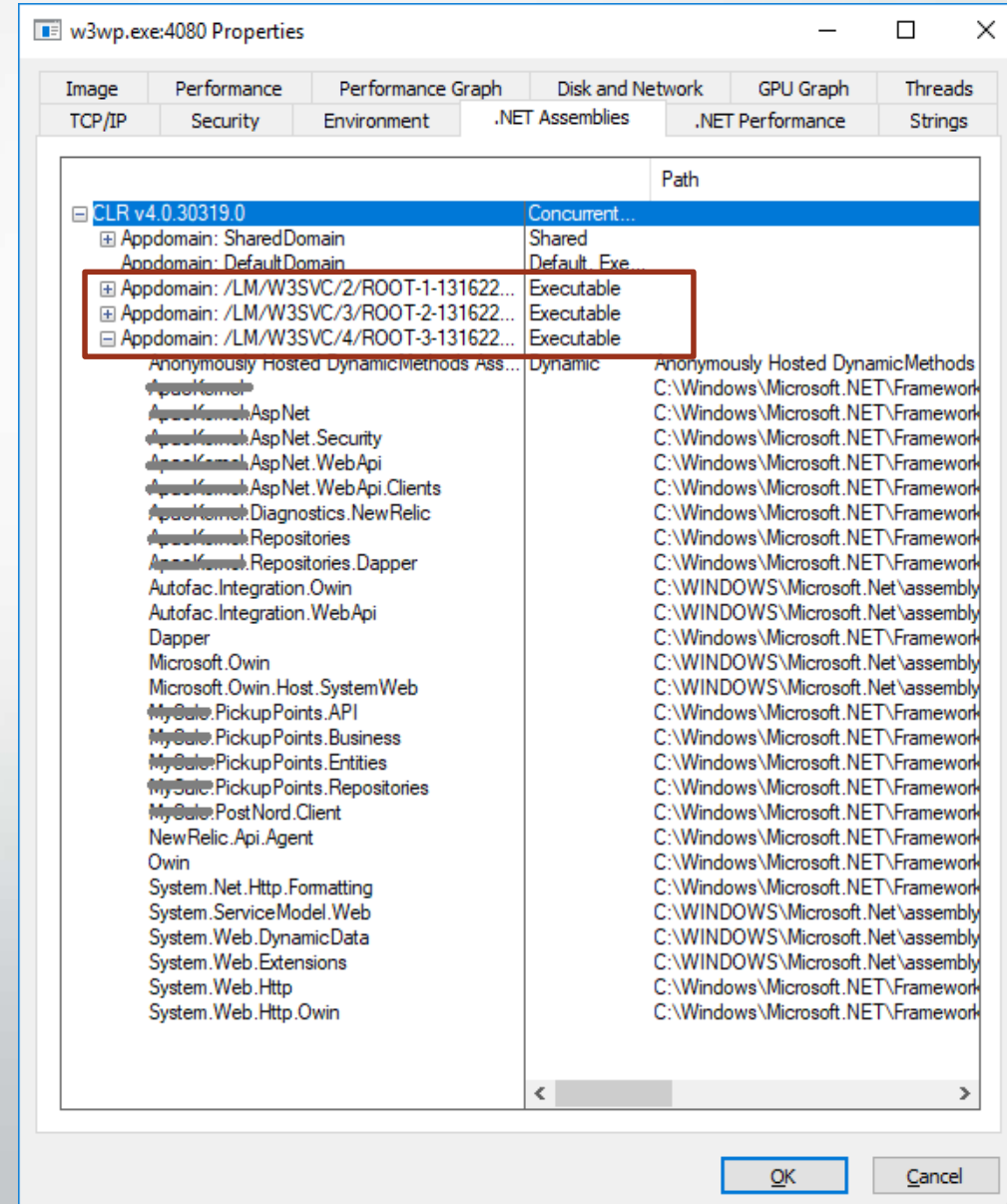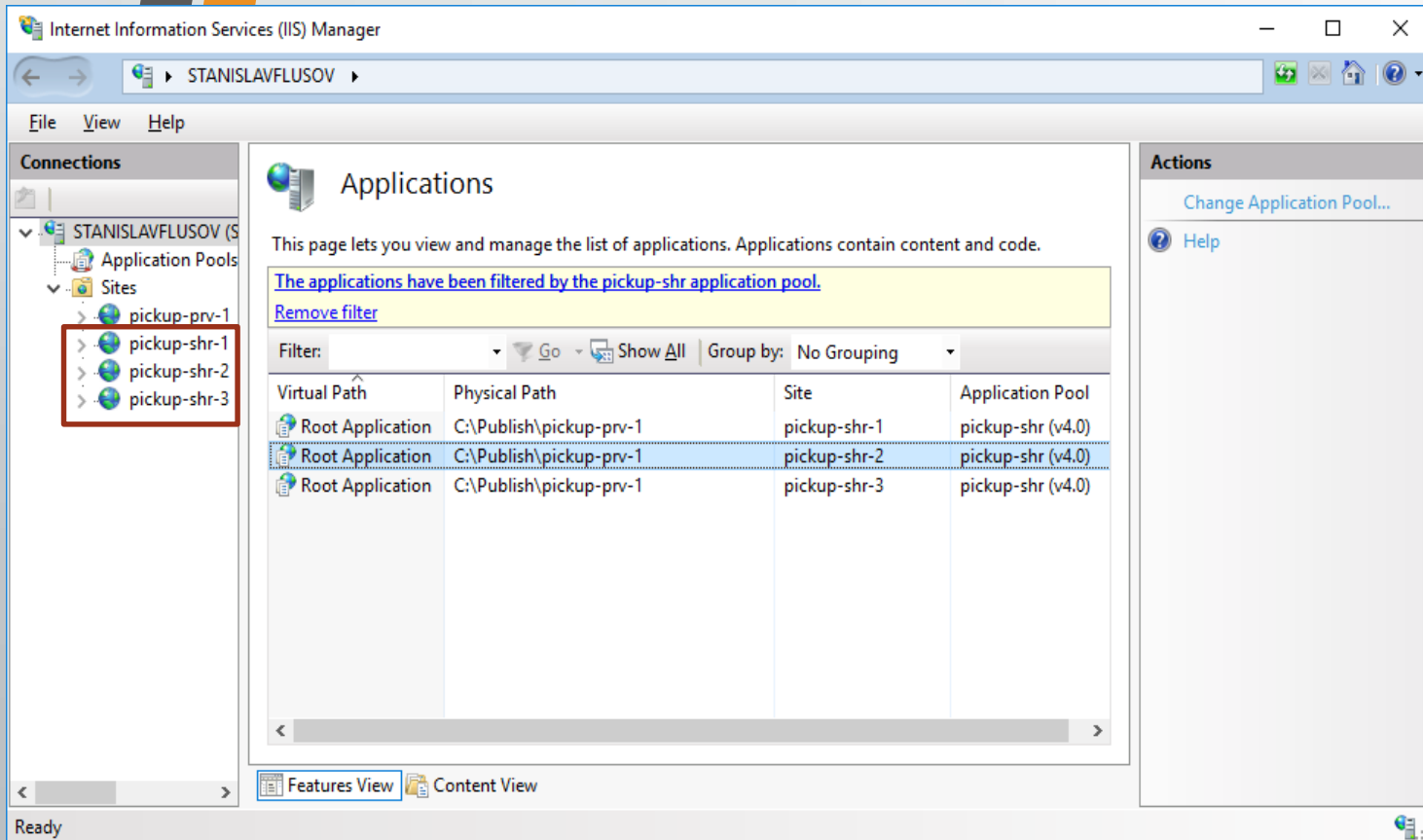

Domain neutral assembly

- It lives across multiple appdomains

- It will be jitted only once

- The jitted code will be shared across appdomains

See also:

https://blogs.msdn.microsoft.com/junfeng/2004/08/05/domain-neutral-assemblies/

Sharing assemblies between application domains

Sharing assemblies between application domains

One site in appPool

Three sites in appPool

Sharing assemblies between application domains
Installing assemblies to GAC

```powershell
 2    #gacutil.exe /i mydll.dll
 3
 4    $asm_path = 'C:\git\IISSharingAssemblies\common-assemblies-legacy'
 5    $gac_path = "C:\Program Files (x86)\Microsoft SDKs\Windows\v10.0A\bin\NETFX 4.7 Tools\gacutil.exe"
 6
 7    #install assembly to GAC
 8    Get-ChildItem -recurse $asm_path | where {$_.extension -eq ".dll"} | ForEach-Object {
 9        Write-Host "Try to install assembly $_"
10        & $gac_path "/i", $_.FullName
11    }
```

Sharing assemblies between processes
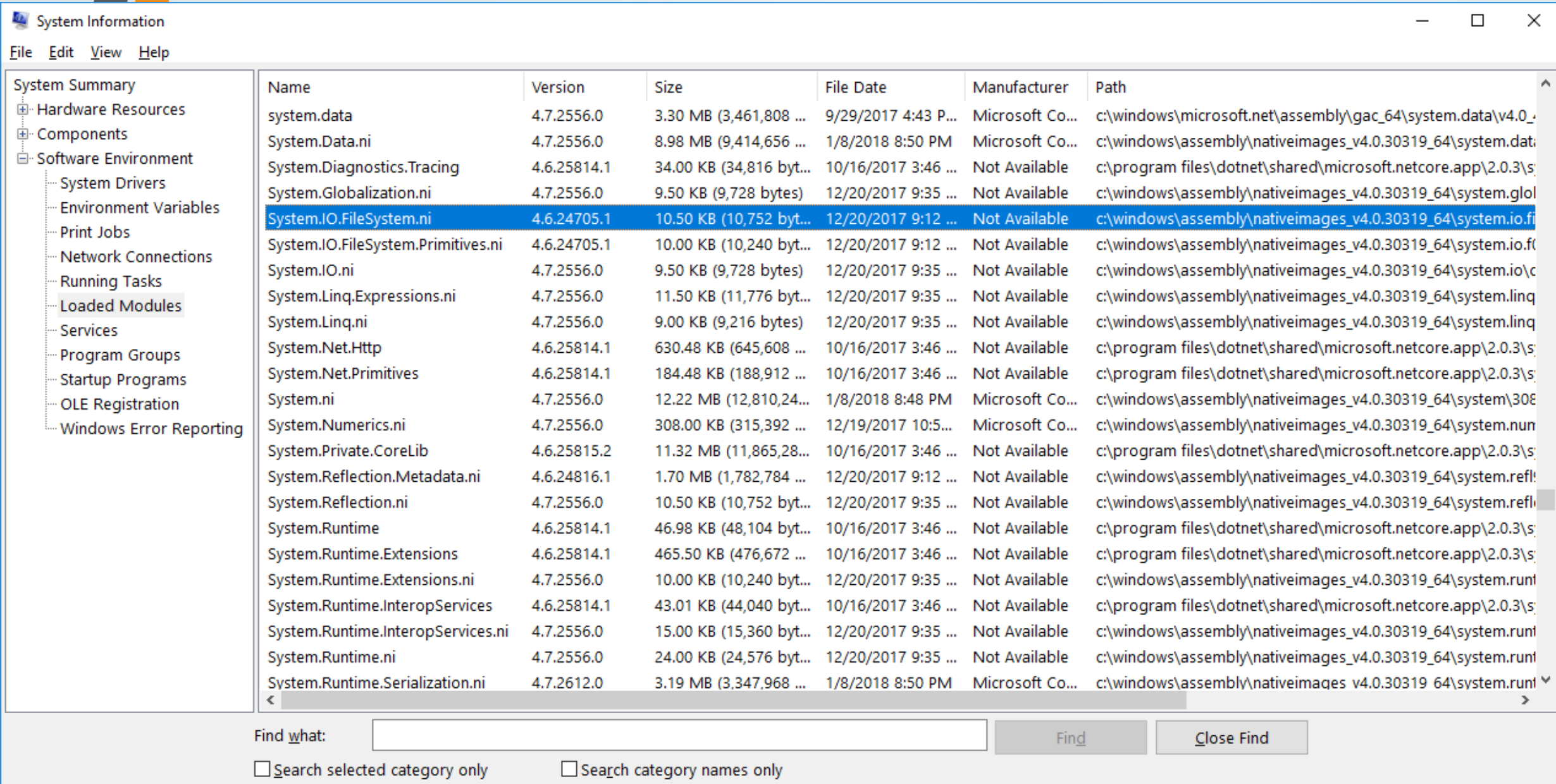
Summary of usage **NGEN**

- Enable code sharing between processes

- Require a smaller initial working set

- Load faster than MSIL

See also:
https://docs.microsoft.com/en-us/dotnet/framework/tools/ngen-exe-native-image-generator
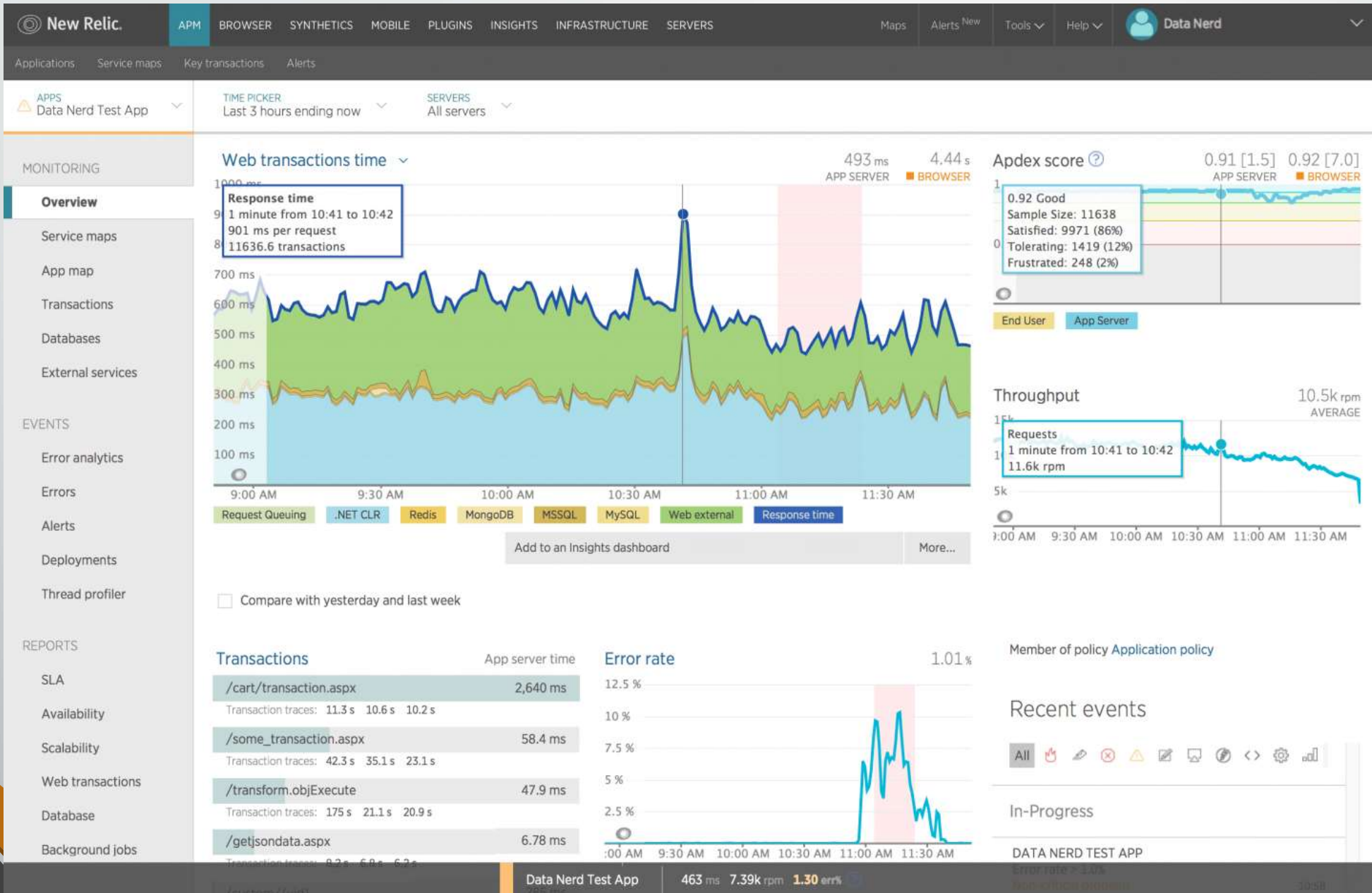
Sharing assemblies between processes

Sharing assemblies between processes

```xml
<runtime>
  <assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">
    <dependentAssembly>
      <assemblyIdentity name="Autofac" publicKeyToken="17863AF14B0044DA" culture="neutral"/>
      <bindingRedirect oldVersion="0.0.0.0-3.5.0.0" newVersion="3.5.0.0"/>
    </dependentAssembly>
    <dependentAssembly>
      <assemblyIdentity name="Microsoft.Owin" publicKeyToken="31bf3856ad364e35" culture="neutral"/>
      <bindingRedirect oldVersion="0.0.0.0-3.1.0.0" newVersion="3.1.0.0"/>
    </dependentAssembly>
    <dependentAssembly>
      <assemblyIdentity name="Newtonsoft.Json" publicKeyToken="30ad4fe6b2a6aeed" culture="neutral"/>
      <bindingRedirect oldVersion="0.0.0.0-9.0.0.0" newVersion="9.0.0.0"/>
    </dependentAssembly>
    <dependentAssembly>
      <assemblyIdentity name="System.Web.Http" publicKeyToken="31bf3856ad364e35" culture="neutral"/>
      <bindingRedirect oldVersion="0.0.0.0-5.2.3.0" newVersion="5.2.3.0"/>
    </dependentAssembly>
    <dependentAssembly>
      <assemblyIdentity name="System.Net.Http.Formatting" publicKeyToken="31bf3856ad364e35" culture="neutral"/>
      <bindingRedirect oldVersion="0.0.0.0-5.2.3.0" newVersion="5.2.3.0"/>
    </dependentAssembly>
    <dependentAssembly>
      <assemblyIdentity name="System.IdentityModel.Tokens.Jwt" publicKeyToken="31bf3856ad364e35" culture="neutral"/>
      <bindingRedirect oldVersion="0.0.0.0-5.1.4.0" newVersion="5.1.4.0"/>
    </dependentAssembly>
    <dependentAssembly>
      <assemblyIdentity name="Microsoft.IdentityModel.Tokens" publicKeyToken="31bf3856ad364e35" culture="neutral"/>
      <bindingRedirect oldVersion="0.0.0.0-5.1.4.0" newVersion="5.1.4.0"/>
    </dependentAssembly>
  </assemblyBinding>
</runtime>
```

New Relic: Digital Performance Monitoring and Management

Summary

Step	Warming time	Total memory usage	Average memory usage per service
Initial conditions **	6 min 43 sec	8 GB	7 GB / 47 = 152 Mb
1. App pools combining 2. Load lib to GAC **	3 min 20 sec	3.5 GB	2.5 GB / 47 = 54 Mb
1. App pools combining 2. Load lib to GAC 3. Make native images	1 min 57 sec	2.0 GB	1.0 GB / 47 = 22 Mb

Remarks:
*   There are some deviations in values, because screenshot was made at different time
** NewRelic agent installed

.NET Core & .NET Standard

```xml
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <system.webServer>
    <handlers>
      <add name="aspNetCore" path="*" verb="*" modules="AspNetCoreModule" resourceType="Unspecified" />
    </handlers>
    <aspNetCore processPath="dotnet" arguments=".\Core.MyService.dll" stdoutLogEnabled="true" stdoutLogFile="C:\tmp-files\logs\output" />
  </system.webServer>
</configuration>
```

⊟ ▣ svchost.exe		5,296 K	10,360 K	3400 Host Process for Windows S...	Microsoft Corporation	64-bit
⊟ ▣ w3wp.exe	0.01	5,356 K	14,000 K	18584 IIS Worker Process	Microsoft Corporation	64-bit
⊟ ▣ dotnet.exe	0.01	95,920 K	41,432 K	14156 dotnet	Microsoft Corporation	64-bit
▣ conhost.exe		5,184 K	9,448 K	17372 Console Window Host	Microsoft Corporation	64-bit
▣ svchost.exe		2,008 K	5,332 K	1324 Host Process for Windows S...	Microsoft Corporation	64-bit

See also:
- https://github.com/dotnet/standard/blob/23ee76bbcd4997c199472a95a9d01f1a74f98e94/docs/faq.md
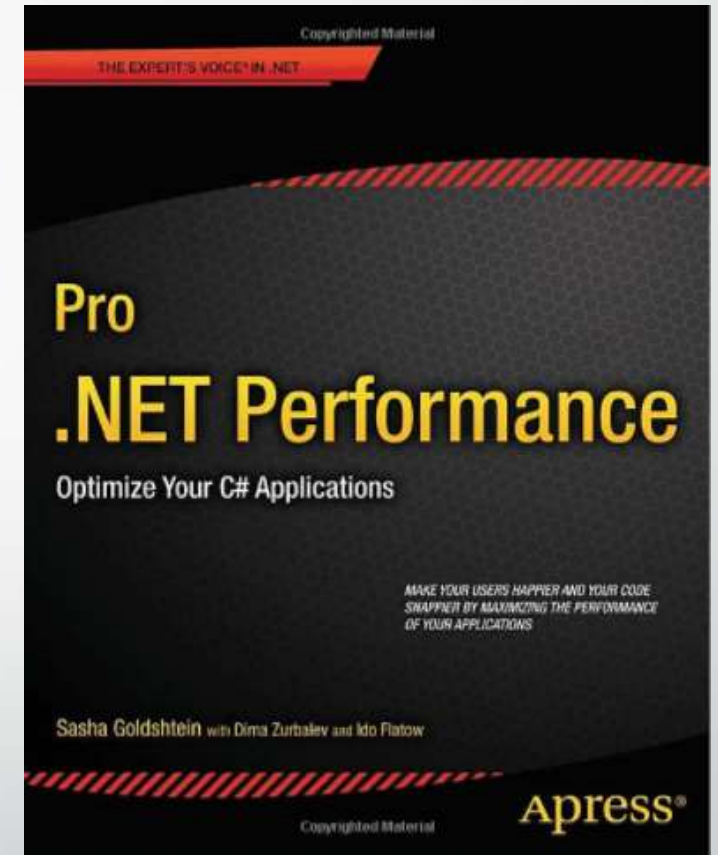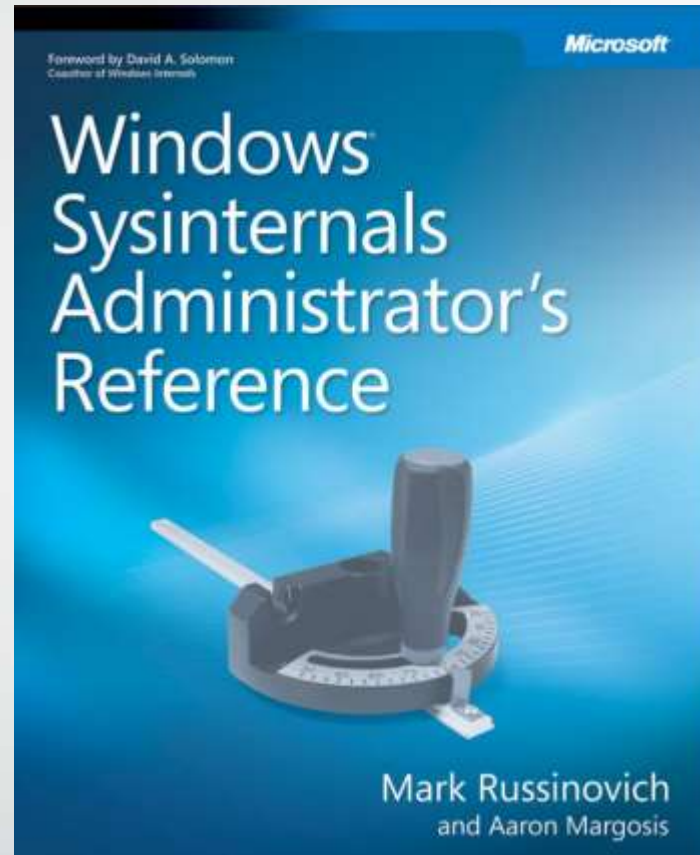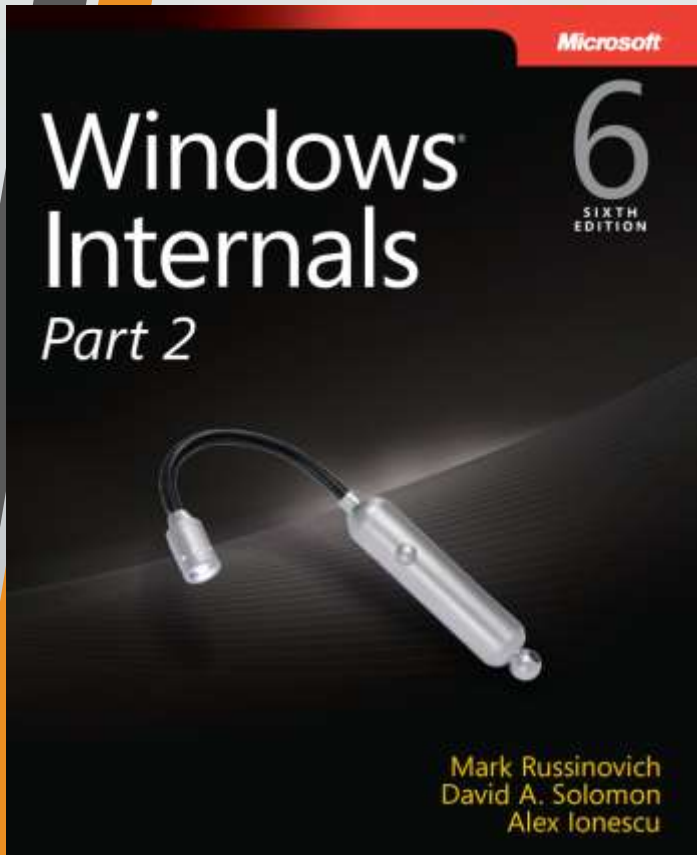
19

.NET Core & .NET Standard

```
83          if (assemblyName.Length == 0 || assemblyName[0] == '\0')
84          {
85              throw new ArgumentException(SR.ZeroLengthString);
86          }
87
88          return assemblyName;
89      }
90
91  public static AppDomain CreateDomain(string friendlyName)
92  {
93      if (friendlyName == null) throw new ArgumentNullException(nameof(friendlyName));
94      throw new PlatformNotSupportedException(SR.PlatformNotSupported_AppDomains);
95  }
96
97  public int ExecuteAssembly(string assemblyFile) => ExecuteAssembly(assemblyFile, null);
98
99  public int ExecuteAssembly(string assemblyFile, string[] args)
100 {
```

See also:

* https://github.com/dotnet/standard/blob/23ee76bbcd4997c199472a95a9d01f1a74f98e94/docs/faq.md

Books recommended for reading

References

- https://www.microsoftpressstore.com/store/windows-internals-part-1-9780735648739
- https://www.microsoftpressstore.com/store/windows-sysinternals-administrators-reference-9780735656727
- http://www.apress.com/us/book/9781430244585

Thank you

- Article: https://habrahabr.ru/post/341950/
- Source code: https://github.com/sflusov/IISSharingAssemblies
- VK: https://vk.com/stanislav_flusov