

.NET

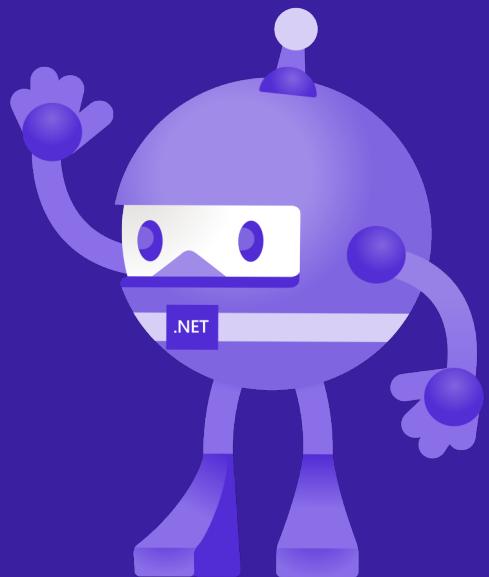
Free. Cross-platform. Open source.
A developer platform for building all your apps.

www.dot.net

Perf Improvements in .NET 6

Stephen Toub

Partner Software Engineer
.NET



.NET 6 == huge release for perf

- > 6500 PRs into the release
 - > 550 primarily for performance
 - > 15% from non-Microsoft contributors
- Obviously not covering them all here
 - [Performance Improvements in .NET 6 - .NET Blog \(microsoft.com\)](#)
- Highlight a few library examples
 - Through the lens of what/why/how...
 - and “Friends”

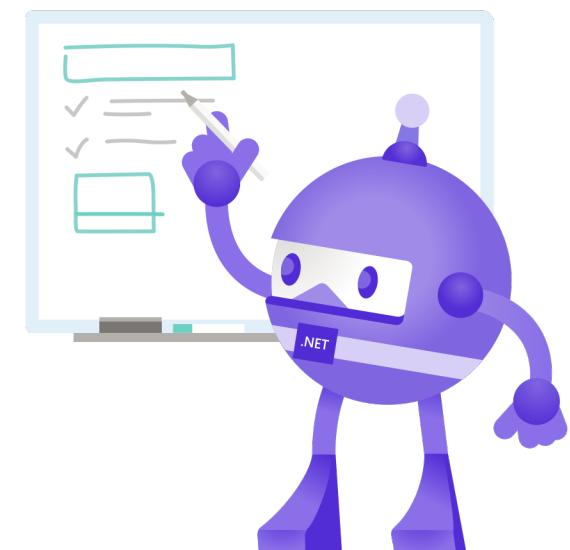


Threading

- CancellationToken{Source}
 - “The One Where We Revisit Past Assumptions”

à: ĀTI ī t „ ā
” əgőjí ñújč ; Ā, j Āgûjgj Ä Bûg ügÄáó
ó
əgjTg ái, i j g t mTAP ; , Ti Äoo, j Jutj uOATPne i Äáó
i j g əgj uOATg Ägûjgj Ä əgnit t mómÄGmTéooð Ágj ügÄáó
Ä

Version	Time	Ratio	Allocation
.NET Framework 4.8	144.218 ns	1.00	385 B
.NET Core 3.1	79.392 ns	0.55	352 B
.NET 5.0	79.431 ns	0.55	352 B
.NET 6.0	56.715 ns	0.39	192 B



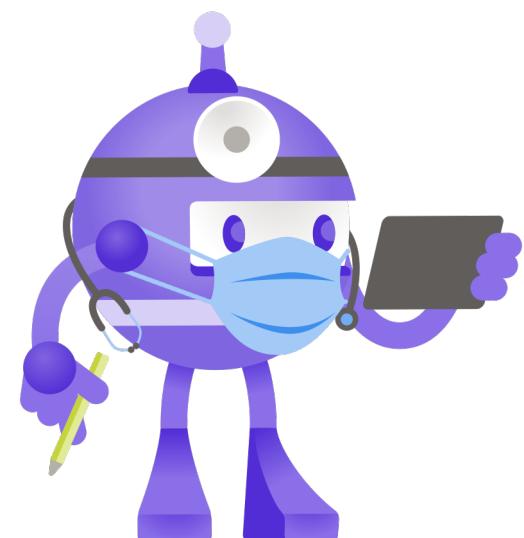
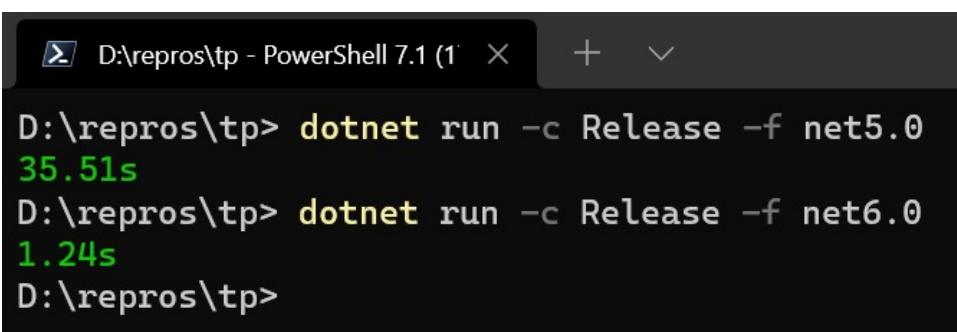
Threading, cont.

- “Sync-over-async” w/ the thread pool
 - “The One Where We Just Roll With It”

əg̊j̊T̊G̊ ꝑC̊g̊j̊ Åt̊ åB̊j̊, G̊T̊ůg̊j̊ J̊i̊ g̊K̊

И, ј і ѡ т м̄тАЛ і, ѡО ѿ ѿј јуТиуе i АæóК
И, ј, ѡОг҃рт м̄тАЛ Аљ Ѥ Ти, ѡА æóК
ѡ ѕТј ј т риVкнм ТиFти ѿТј ѕТј ѿи ѡггү ; ѿТј DzMкнжшØ
ѡ
JТј Jcнrт нùК
j, ѡОг҃ж ѡсса, ѡОкк ѡка ѡи ї ѡк, ѡОк, Jj æóóóK
A
j, ѡОг҃ж ѡсса, ѡОкк ѡка ѡи ї ѡкАл ѡАггёoj æóóóK

И, ѡЛ т м̄П ѿ Л, j i Ткк „ j аАЛæóK
i, ѡОк, Jj ѿсса, ѡОг҃ж ѿ, ѿСæóóK
; ѿТкюОАæт Jj ААУТАæNeøдЛаFо.” ѿАсса ѿи ѿПАи ѿЧкк ѡAæéóK



Threading, cont.

- Async state machine pooling
 - “The One Where We Cache All The Things”

ხელი თუ არ არის დანართული

• კოდი დანართულია
• კოდი დანართულია

გვი ამონ დანართულია

• კოდი დანართულია
• კოდი დანართულია

კოდი დანართულია და კოდი დანართულია

გვი ამონ დანართულია

• კოდი დანართულია
• კოდი დანართულია

კოდი ამონ დანართულია და კოდი დანართულია

კოდი დანართულია და კოდი დანართულია

Type	Allocations
System.Runtime.CompilerServices.AsyncTaskMethodBuilder<>.AsyncStateMachineBox <>	1,003
System.Runtime.CompilerServices.AsyncTaskMethodBuilder<System.Threading.Tasks.VoidTaskResult>.AsyncStateMachineBox <<< Main >>> g_NoPoolingAsync0_2>d <>	1,000
System.Runtime.CompilerServices.AsyncTaskMethodBuilder<System.Threading.Tasks.VoidTaskResult>.AsyncStateMachineBox <<< Main >>> g_With0_1>d <>	1
System.Runtime.CompilerServices.AsyncTaskMethodBuilder<System.Threading.Tasks.VoidTaskResult>.AsyncStateMachineBox <<< Main >>> g_Without0_0>d <>	1
System.Runtime.CompilerServices.AsyncTaskMethodBuilder<System.Threading.Tasks.VoidTaskResult>.AsyncStateMachineBox <<< Main >>> d_0 <>	1
System.Runtime.CompilerServices.PoolingAsyncValueTaskMethodBuilder<System.Threading.Tasks.VoidTaskResult>.StateMachineBox <<< Main >>> g_PoolingAsync0_3>d <>	1
System.Runtime.CompilerServices.PoolingAsyncValueTaskMethodBuilder<System.Threading.Tasks.VoidTaskResult>.SyncSuccessSentinelStateMachineBox <>	1



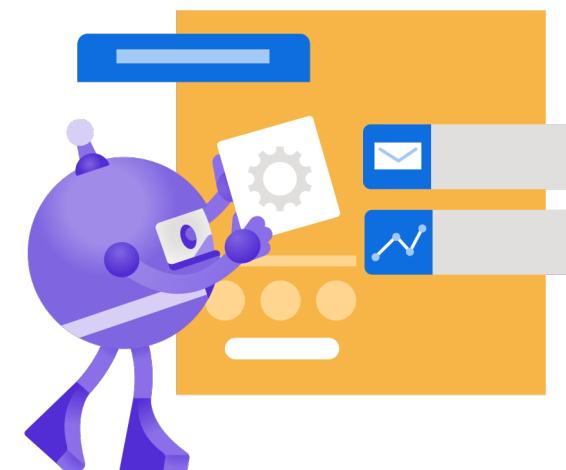
System.*

- Random
 - “The One Where It’s No Longer the 1980s”

„, JИ, j Ă gq ĂăĂmРgoggĂ mт mĂЛ gq ĂăVIVWRIWNAĂ
„, JИ, j Ă ≥, Tčut R , Tčut mт mĂЛ ≥, Tčut aóK

à: ĂTl Ēt „ ĀA
“ eгoJi ИuJc aĂPj : Q Āgao mт i nR , Tčut aĂPj : Q ĀgaoРgoggĂ óK

Version	Time	Ratio
.NET 5.0	72.2 us	1.00
.NET 6.0	1.2 us	0.02



System.*, cont.

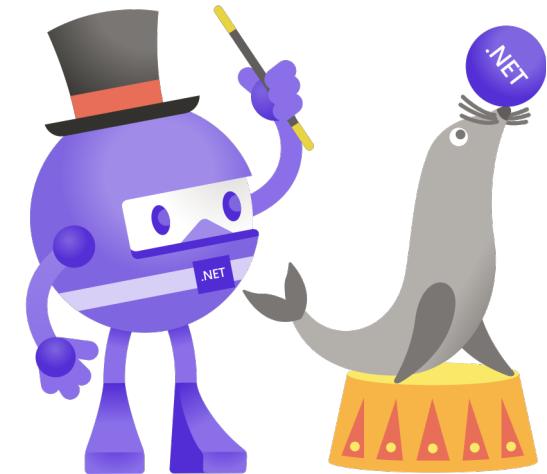
- Guid.Parse
 - “The One Where We’re Card Counters at the Instruction Table”

” , ȐИ, ј А ѕј, ЈТГ Рѹжчнт мј өјчакАлј өјчак ѕуП, ЈТГ ѕОК

а: АТІ Т „ ОА

” өг҃оДІ] өјч „ ѕАсмт т мј өјч ако „ ѕАасРѹжчОКн

Version	Time	Ratio
.NET Framework 4.8	251.88 ns	1.00
.NET Core 3.1	100.78 ns	0.40
.NET 5.0	80.13 ns	0.32
.NET 6.0	33.84 ns	0.13

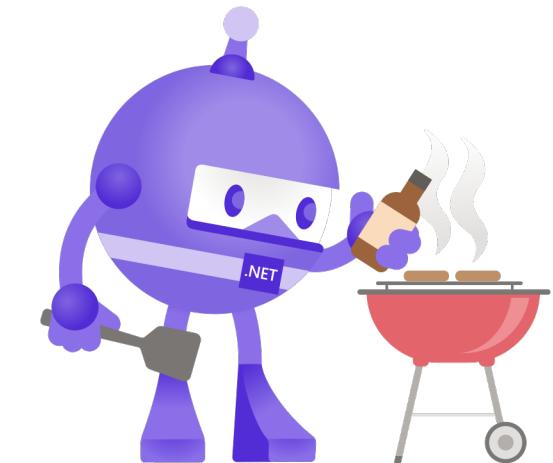


System.*, cont.

- `DateTime.UtcNow`
 - “The One Where We Have Our Cake And Eat It, Too”

à: ǞT̄I T̄t „ ŌA
" ēḡoJ̄I B̄, j̄ Āj̄ J̄t̄ Ā »j̄ I ǟuLǟoñt̄ t̄m̄B̄, j̄ Āj̄ J̄t̄ Ā »j̄ I ǟuLǟK

Version	Time	Ratio
.NET Core 2.1	20.96 ns	1.00
.NET Core 3.0	63.35 ns	3.01
.NET 6.0	19.95 ns	0.95



System.*, cont.

- Analyzers (Strings, Spans, ...)
 - “The One Where We Optimize Your Code”

The screenshot shows a code editor with the following code:

```
1 Console.WriteLine(SayHello("\"Stephen\""));
2
3 static string SayHello(string quotedName) =>
4     "Hello," + quotedName.Substring(1, quotedName.Length - 2);
```

A tooltip is displayed over the line of code at line 4, highlighting the string concatenation. The tooltip contains the following information:

- Use 'AsSpan' with 'string.Concat'
- Use block body for local functions
- Use range operator
- Wrap expression
- Introduce parameter for "Hello," + quotedName.Substring(1, quotedName.Length - ...)
- Convert to interpolated string
- Suppress or Configure issues

At the top right of the tooltip, there is a warning icon and the text "CA1845 Use span-based 'string.Concat' and 'AsSpan' instead of 'Substring'". Below the warning, the code is shown with the recommended changes applied:

```
using System;

Console.WriteLine(SayHello("\"Stephen\""));

static string SayHello(string quotedName) =>
    "Hello," + quotedName.Substring(1, quotedName.Length - 2);
    string.Concat("Hello,", quotedName.AsSpan(1, quotedName.Length - 2));
```

At the bottom right of the tooltip, there are buttons for "Preview changes" and "Fix all occurrences in: Document | Project | Solution".

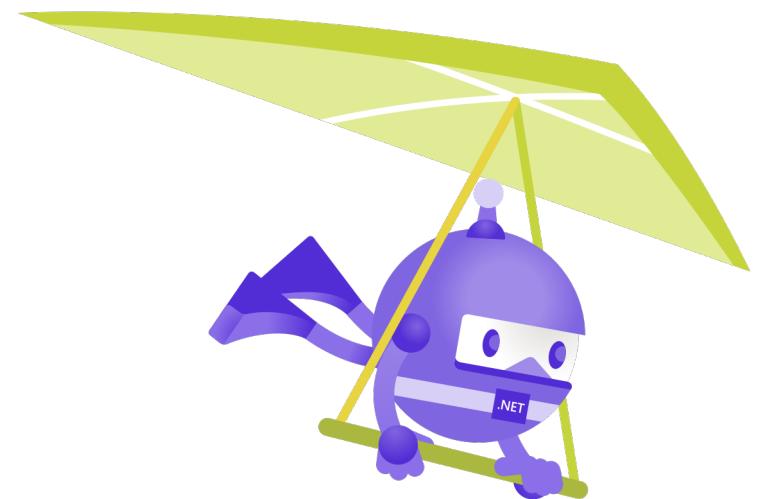
System.*, cont.

- Span/Array.Fill
 - “The One Where We Vectorize”

” „ ÐИј Ă ī Ė „ àÄñR „ , Oñt mĂЛ i Ė „ àVWXAK
” „ ÐИј Ă ī Ė „ Rí mñ mñ ÕK

à: ÄTí Tt „ ÕA
” egóJi iñjč \ Jõõaõmít m „ , Cä JõõaõR „ , o ri ÕX

Version	Time	Ratio
.NET 5.0	55.95 ns	1.00
.NET 6.0	3.81 ns	0.07



System.*, cont.

- String.Replace
 - “The One Where We Special-Case”

”, JI,j Ă gj, JTG Rgj, K

à] óúg, óTÄj é Ā
” egóJi , gCTI i , gO TÄj é aó
ó
egóJTG I, tI t mÄP dj j ; óJÄTj aó
Rgj, m t m L, Jj t i a Äj P, JTG gCTI a t j j g g ÄMPL a e j ÄTgÄ G a y G A i , i t ÄA Ä e g ÄVIVVÄ GWMVW Pj eó
Ä

à: ÄTl t „ OÄm egóJi gj, JTG EÄo o a m t m Rgj, a Ä o, i Ä a e ä T e G r e ö

à: ÄTl t „ OÄm egóJi gj, JTG ; uTl , j a o m t m Rgj, a Ä o, i Ä a e ä T e G r e ö



Benchmark	Version	Time	Ratio
Yell	.NET 5.0	32.85 ms	1.00
Yell	.NET 6.0	16.99 ms	0.52

Benchmark	Version	Time	Ratio
Concat	.NET 5.0	34.36 ms	1.00
Concat	.NET 6.0	22.93 ms	0.67

System.*, cont.

- String Interpolation [\(blog\)](#)
 - “The One Where We Reinvent The Mechanism”

Version	Time	Ratio	Allocation
.NET 5.0	160.4 ns	1.00	192 B
.NET 6.0	42.3 ns	0.26	-

gj , j Ji և Jč „ „ AčæT, JTG eJóčA ցցՕնJ TJ t , Nü ՇնJ TJ t JTy ՇնJ TJ ցԵՋօ՛ՇնJ TJ , ԱՄՇյՌՌt t ցց՛ ” AčæNe ot , Nü Առt JTy ԱռշԵՋօ՛ՇնJ TJ , ԱՄՇյՌՌt eOK

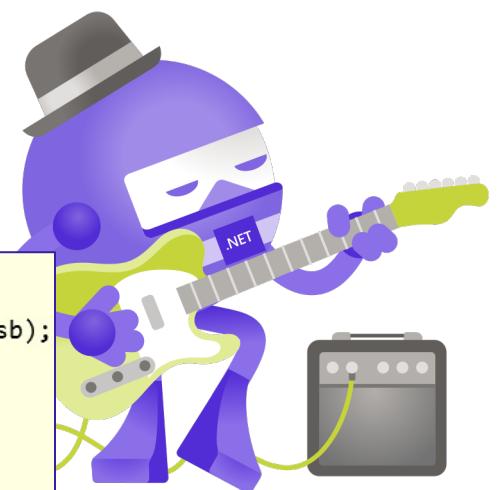
.NET 5 + C# 9

```
private static void Append(StringBuilder sb, int major, int minor, int build, int revision)
{
    sb.Append(string.Format("{0}.{1}.{2}.{3}", major, minor, build, revision));
}
```

```
private static void Append(StringBuilder sb, int major, int minor, int build, int revision)
{
    object[] array = new object[4];
    array[0] = major;
    array[1] = minor;
    array[2] = build;
    array[3] = revision;
    sb.Append(string.Format("{0}.{1}.{2}.{3}", array));
}
```

.NET 6 + C# 10

```
private static void Append(StringBuilder sb, int major, int minor, int build, int revision)
{
    StringBuilder.AppendInterpolatedStringHandler handler = new StringBuilder.AppendInterpolatedStringHandler(3, 4, sb);
    handler.AppendFormatted(major);
    handler.AppendLiteral(".");
    handler.AppendFormatted(minor);
    handler.AppendLiteral(".");
    handler.AppendFormatted(build);
    handler.AppendLiteral(".");
    handler.AppendFormatted(revision);
    sb.Append(ref handler);
}
```



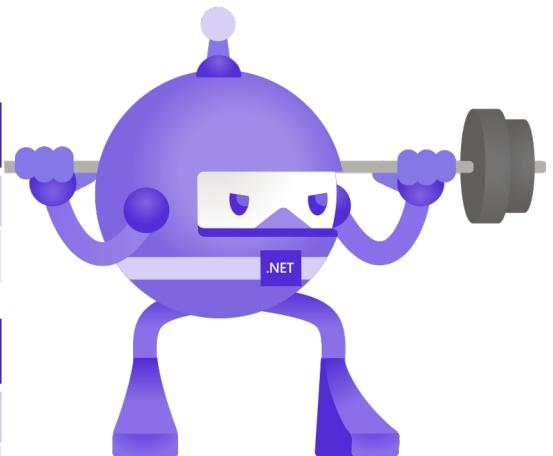
ArrayPool

- Max Array Size
 - “The One Where We Remove The Cliff”

à: ÄTİ T̄ ̄ Öæ , gÄöJ TÄm̄ m̄, eÄöA
" eägöJi ̄ Hñjç ≥ÄTİ ≥Äj e T̄RVIWHDÖKkä öm̄ t̄ m̄, , OaüuöTgQ Ät äT̄ , Äc äÄj e T̄äe
OaüuöTgQ Ät äT̄ , Äc äÄj e T̄äe

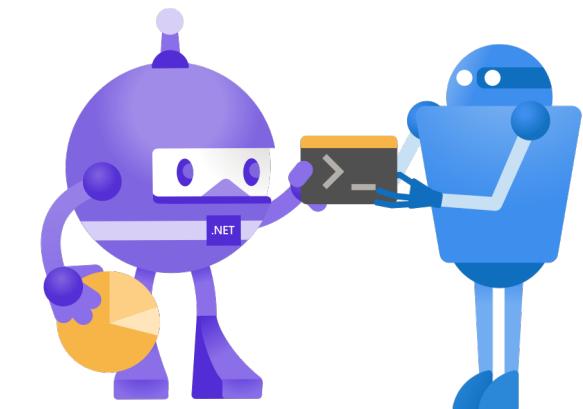
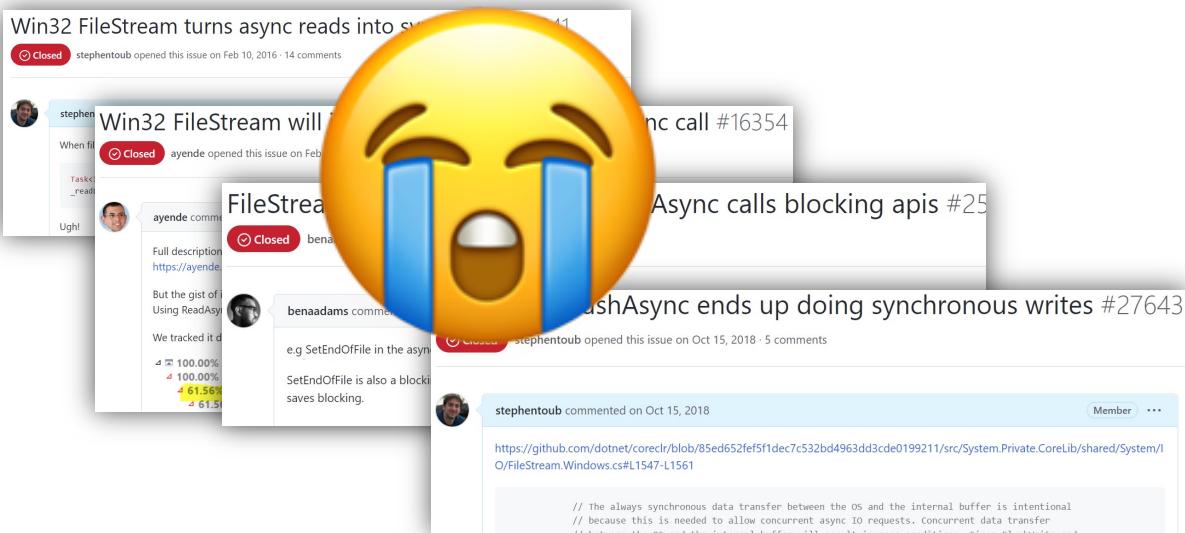
Benchmark	Version	Time	Ratio	Allocated
RentReturn_1048576	.NET 5.0	21.01 ns	1.00	
RentReturn_1048577	.NET 5.0	12,132.90 ns	577.48	1,048,593 B

Benchmark	Version	Time	Ratio	Allocated
RentReturn_1048576	.NET 6.0	16.36 ns	1.00	
RentReturn_1048577	.NET 6.0	16.38 ns	1.00	



File IO

- FileStream [\(blog\)](#)
 - “The One Where We Make Async Actually Async”



File IO

- FileStream
 - “The One Where We Make Async Actually Async”

“ঠা] ওঁগ্ৰ, ওঁটাজ্ এ আ
” এগোজি নুজি টাজ্ এ আ

gó Äääämč, j , m̄t m̄ččL gq ÄäVIVIIVVRIIVVAK
ččL ≥ Tččut, aäVIVIIVVRIIVVAK : q Äggæ, j , óč

gj, JĘG", j i m̄ m̄, j i ā Āj i Āf \JōAā, t̄ ĀāOK
\JōĀāt̄ Jj Ā ō: C̄ Āḡā, j i Ḡm̄c̄, j , OK

Þegjóðum, Á, t, t m̄fAPL \JÓAPL, Á, t æ, j TGN JÓAðučAæ AÐGN JÓA i i ÄggæA, cGN JÓAPL, ÁæA, cGNMGNgA gCTI gñj, eÄK

ବ୍ୟାକୁ ପାଇଁ ଏହିରେ କିମ୍ବା କିମ୍ବା କିମ୍ବା କିମ୍ବା

Þórgjöð ÁPI. Á, t æBjúg úg ÁæóK
\\ JÓAæBÁo Áj Áæþgjöð ÁPI. Á, t áa, t ÁóK

à: ĂTİ ī Ł „ ŌĀ
” eğőöJi „ gCTI i „ gÖ ≥Ă, c gCTI aó

GĂŤčJg
GmNgmHgegg
Árígí míváník

Version	Time	Ratio	Allocation
.NET Framework 4.8	355.670 ms	1.00	3,833,856 B
.NET Core 3.1	262.625 ms	0.74	3,048,120 B
.NET 5.0	259.284 ms	0.73	3,047,496 B
.NET 6.0	119.573 ms	0.34	403 B



Networking

- HttpHeaders.NonValidated
 - “The One Where We Stop Doing Unnecessary Work”

à: ÄTI TŁ „ ØA

” egóJi „ gCTI i „ gØ FŁet Ä , j ÅæØ

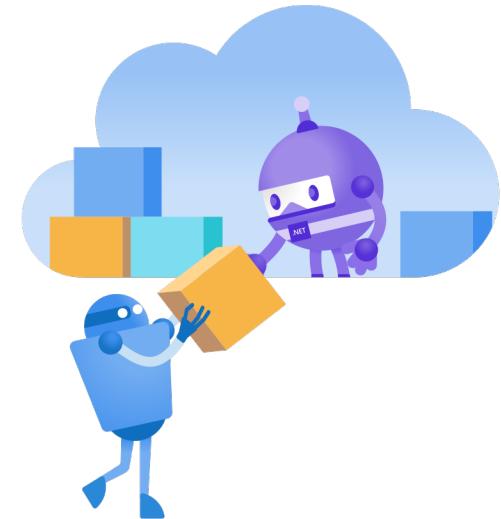
Ø

И „ Ä eÄgj m t mÄPL dj j „ gÄ eÄgj aÄgg, GÄædj j „ äÄj T uÄç Äj GmÄ Rø JØK
eÄgjTøG И „ m Äg m t m L, Jj „ gRø öJÄTj aÄTÄTc gCTI æ Ä eÄgj GmÄ Äg, eöj ØK

gÅ Ä, i T aM „ TÄ, cÄ mÄT, Äg aÄÄ, cÄ gÄDænT, öJc, j ÄcDÄ ØmørñA
gÅ Ä, i T aM „ i uTj ÄTj dÄ, cÄ Jt, Äg aÄ uTj ÄTj aÄÄ, cÄ gÄDænT, öJc, j ÄcDÄ ØmørñA

Ä „ L, Jj „ Äg aÄ uTj ÄTj aÄ u Ci u gCTI aÄT, Ä, t aÄeooØK

Version	Time	Ratio	Allocated
Original	82.70 us	1.00	3 KB
NonValidated	67.36 us	0.81	2 KB



Reflection

- Activator.CreateInstance
 - “The One Where new() Is Faster”

” „ JИ,j Ă i m, Ă,j ĂT i t aómпtĂ Ă i ngmĂП aómt i mĂП i aóK

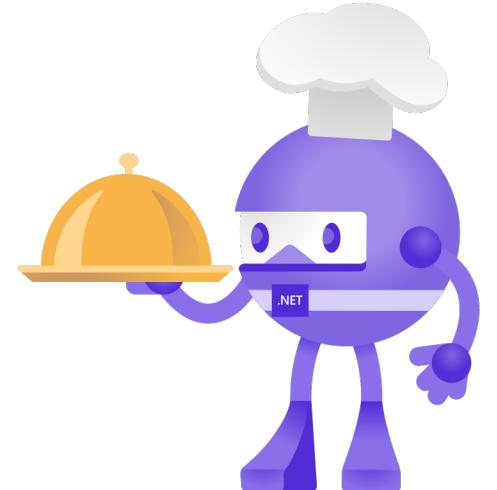
ă: ĂT i T t „ Ā

” egóJi „ ūG , t ; Ă,j Ăaómt i m, Ă,j ĂT „ ūG , t i aóK

Version	Time	Ratio	Allocation
.NET Framework 4.8	49.496 ns	1.00	24 B
.NET Core 3.1	28.296 ns	0.57	24 B
.NET 5.0	26.350 ns	0.53	24 B
.NET 6.0	9.439 ns	0.19	24 B

```
.method private hidebysig
    instance !!T Create<.ctor T> () cil managed
{
    // Method begins at RVA 0x2050
    // Code size 6 (0x6)
    .maxstack 8

    IL_0000: call !!0 [System.Private.CoreLib]System.Activator::CreateInstance<!T>()
    IL_0005: ret
} // end of method Program::Create
```



Reflection, cont.

- Invoke allocation
 - “The One Where We Get Faster At Self-Defense”

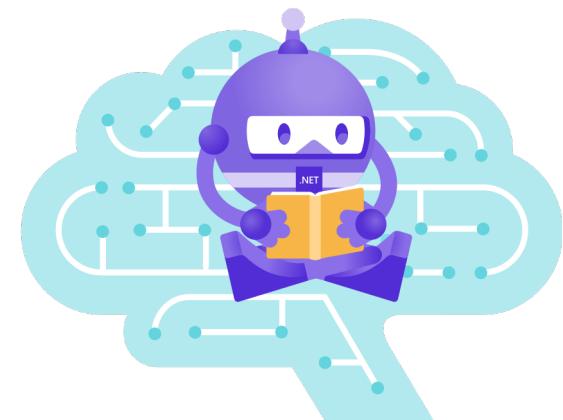
”, ЈИ, я А ёАј Т ўчи Тгү Р, Ај Т ўчмт мј С Аўгас, ўG , т Ок Ај ёАј Т ўчас, ёеOK

” ёг҃оJи ИJc ёаJТ „ GVГmгj, JТG „ GVОmомA

а: AТI Т „ ОA

” ёг҃оJи ИJc i ТИиОАас, мт т нР, Ај Т ўчас ТИиОАа, Т Jг, ОмTЛ uгNAI j аАмомVОмej ЛуемAOK

Version	Time	Ratio	Allocation
.NET Framework 4.8	195.5 ns	1.00	104 B
.NET Core 3.1	156.0 ns	0.80	104 B
.NET 5.0	141.0 ns	0.72	104 B
.NET 6.0	123.1 ns	0.63	64 B



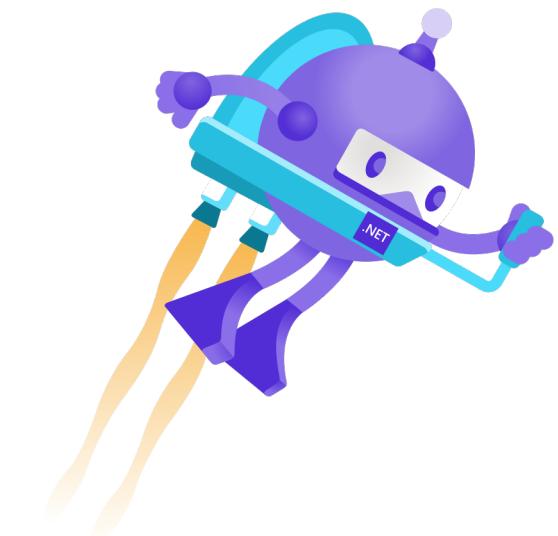
LINQ

- SequenceEqual
 - “The One Where We Go Right To Ludicrous Speed”

" „ JИ́ j Ă i FТet Ă , gоĂTj Тj t nRÁVnít mFТet Ă , gоĂæ , TГAéлVГnMРíVVnRíVVnDæ u , , , Cæók
" „ JИ́ j Ă i FТet Ă , gоĂTj Тj t nRÁVnít mFТet Ă , gоĂæ , TГAéлVГnMРíVVnRíVVnDæ u , , , Cæók

ä: ÄTİ ጥ ዕ Ä
” የጤናን ቁጥር ተስፋ ይችላል እና የሚያስፈልግ የሚያስፈልግ የሚያስፈልግ የሚያስፈልግ

Version	Time	Ratio
.NET Framework 4.8	10,822.6 us	1.00
.NET 5.0	5,421.1 us	0.50
.NET 6.0	150.2 us	0.01



LINQ, cont.

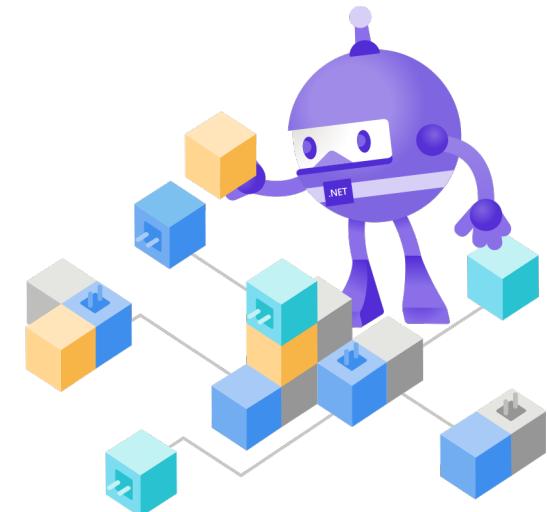
- Distinct
 - “The One Where We Stop Duplicating Code”

”, JИ, j Ă i FТет Ă , gоĂTgj, JТQ nРc, j , mт mFТет Ă , gоĂ , TГААиGнWВиWВиAоAи j , t t rмк bП, JТQ aоAи n , , , cоĂ

а: AĂt t „ ā

” eгoJi JТj Bжgj JТi j ; uеTj aomt nРc, j , aBжgj JТi j aоa uеTj aоK

Version	Time	Ratio	Allocated
.NET 5.0	5.154 ms	1.04	5 MB
.NET 6.0	2.626 ms	0.53	2 MB



Crypto

- ToBase64Transform
 - “The One Where We Stop Being Lazy”

```
public int TransformBlock(byte[] inputBuffer, int inputOffset, int inputCount, byte[] outputBuffer, int outputOffset)
{
    // inputCount < InputBlockSize is not allowed
    ThrowHelper.ValidateTransformBlock(inputBuffer, inputOffset, inputCount, InputBlockSize);

    if (outputBuffer == null)
        ThrowHelper.ThrowArgumentNullException(ThrowHelper.ExceptionArgument.outputBuffer);

    // For now, only convert 3 bytes to 4
    Span<byte> input = inputBuffer.AsSpan(inputOffset, InputBlockSize);
    Span<byte> output = outputBuffer.AsSpan(outputOffset, OutputBlockSize);
```



Version	Time	Ratio	Allocation
.NET Framework 4.8	329.871 ms	1.000	213,976,944 B
.NET Core 3.1	251.986 ms	0.765	213,334,112 B
.NET 5.0	146.058 ms	0.443	974 B
.NET 6.0	1.998 ms	0.006	300 B

JSON

- Source generator [\(blog\)](#)
 - “The One Where We Build At, You Know, Build Time”

The image shows a comparison between a user-defined source code file and a generated source code file. On the left, the 'Program.cs' file contains the following C# code:

```
1 using System.Text.Json.Serialization;
2
3 record Person(string FirstName, string LastName, int Age);
4
5 [JsonSerializable(typeof(Person))]
6 internal partial class MyJsonContext : JsonSerializerContext
```

On the right, the 'MyJsonContext.cs [generated]' file contains the generated serialization logic:

```
This file is auto-generated by the generator 'System.Text.Json.SourceGeneration.JsonSourceGenerator' and cannot be edited.
private static void PersonSerialize(global::System.Text.Json.Utf8Writer writer, Person value)
{
    if (value == null)
    {
        writer.WriteNullValue();
        return;
    }

    writer.WriteStartObject();
    writer.WriteString(PropertyName_FirstName, value.FirstName);
    writer.WriteString(PropertyName_LastName, value.LastName);
    writer.WriteNumber(PropertyName_Age, value.Age);

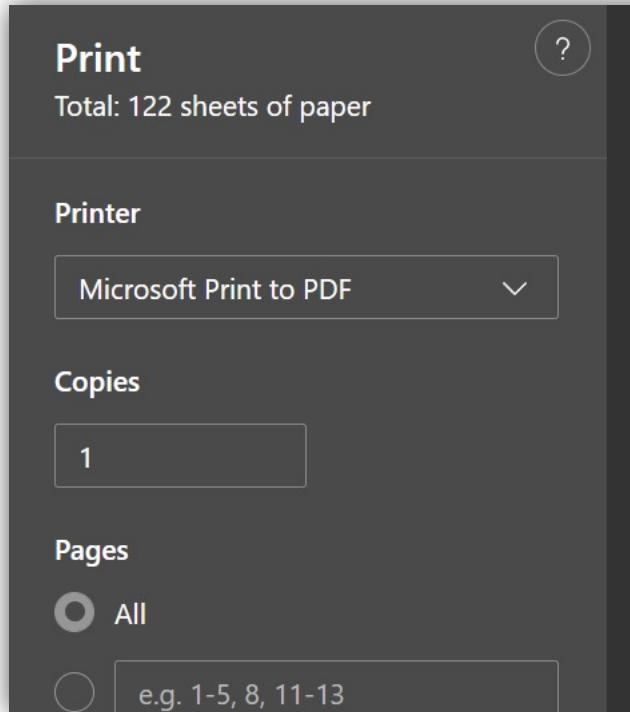
    writer.WriteEndObject();
}

private static global::System.Text.Json.Serialization.Metadata.JsonPropertyMetadata[] GetProperties()
{
    return newJsonPropertyMetadata
    {
        { "FirstName", newJsonPropertyMetadata<string>(PropertyName_FirstName) },
        { "LastName", newJsonPropertyMetadata<string>(PropertyName_LastName) },
        { "Age", newJsonPropertyMetadata<int>(PropertyName_Age) }
    };
}
```

A purple cartoon character wearing a hard hat and holding a briefcase stands to the right of the code editors, representing the build process.

More generators coming in .NET 7...

And so much more...



Performance Improvements in .NET 6



Stephen

August 17th, 2021

f t in

Four years ago, around the time .NET Core 2.0 was being released, I wrote [Performance Improvements in .NET Core](#) to highlight the quantity and quality of performance improvements finding their way into .NET. With its very positive reception, I did so again a year later with [Performance Improvements in .NET Core 2.1](#), and an annual tradition was born. Then came [Performance Improvements in .NET Core 3.0](#), followed by [Performance Improvements in .NET 5](#). Which brings us to today.

The [dotnet/runtime](#) repository is the home of .NET's runtimes, runtime hosts, and core libraries. Since its main branch forked a year or so ago to be for .NET 6, there have been over 6500 merged PRs (pull requests) into the branch for the release, and that's

Thanks for joining!

