.NET

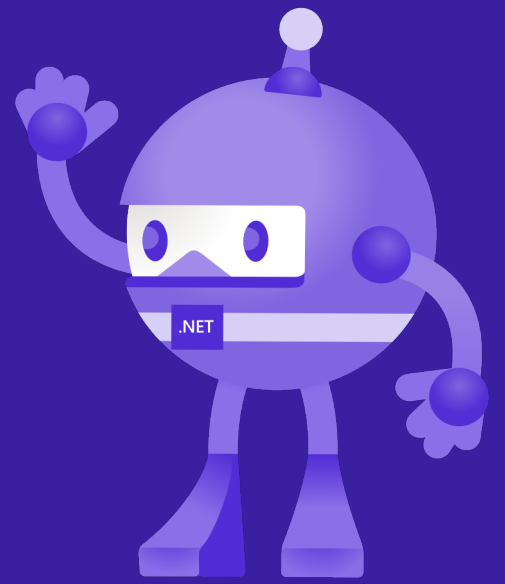
Free. Cross-platform. Open source.

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)

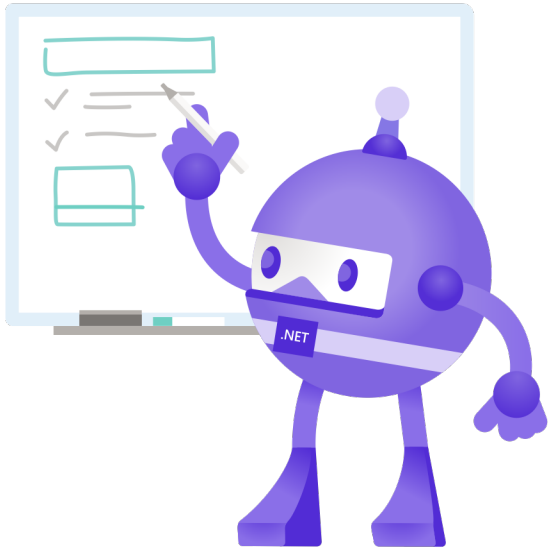
- Highlight a few library examples
  - Through the lens of what/why/how…
    - and “Friends”

Threading

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

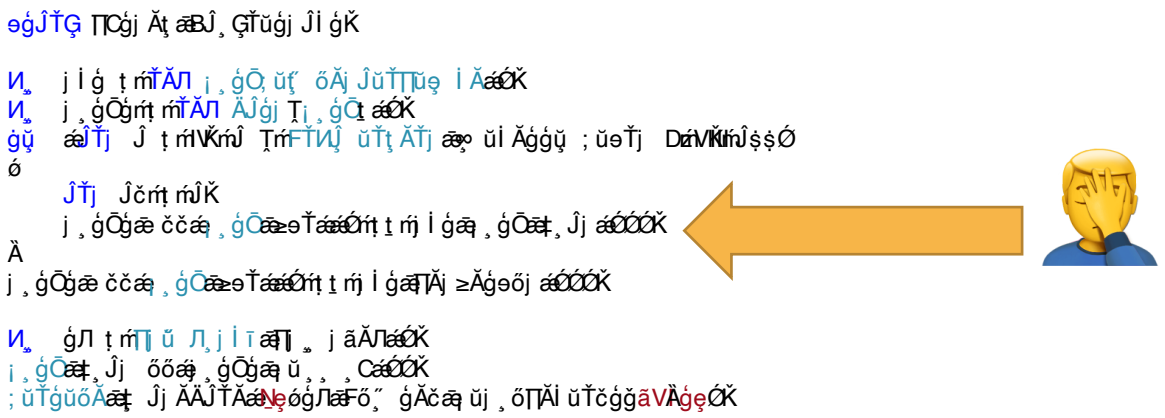
à: ĂŤİ ī ţ ˛ ŌÂ
˝ ɵğőĴİ ИŭĴč ; Ă¸ j Ă≥ĂĢĴģj Ą BĴğ ŭģĂáæØ
ó
ɵģĴŤĢ áæИ˛ İ j ģ ţ mŤĂЛ ; ¸ŤİĂőő¸ j ĴŭŤ¡ ŭŌĂŤŢŢŭę İ Ăáæ ØØ
İ j ģāę ŭŌĂŤāæ≥ĂĢĴģj Ą áæģmţ ţ mømÀĞmŤɵőőØāæBĴğ ŭģĂáæØǨ
À

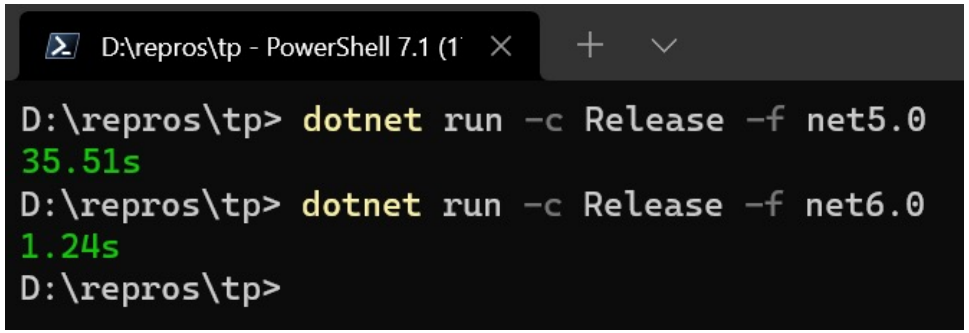
Version	Time	Ratio	Allocation
.NET Framework 4.8	144.218 ns	1.00	385 B
.NET Core 3.1	79.392 ns	0.55	352 B
.NET 5.0	79.431 ns	0.55	352 B
.NET 6.0	56.715 ns	0.39	192 B



Threading, cont.

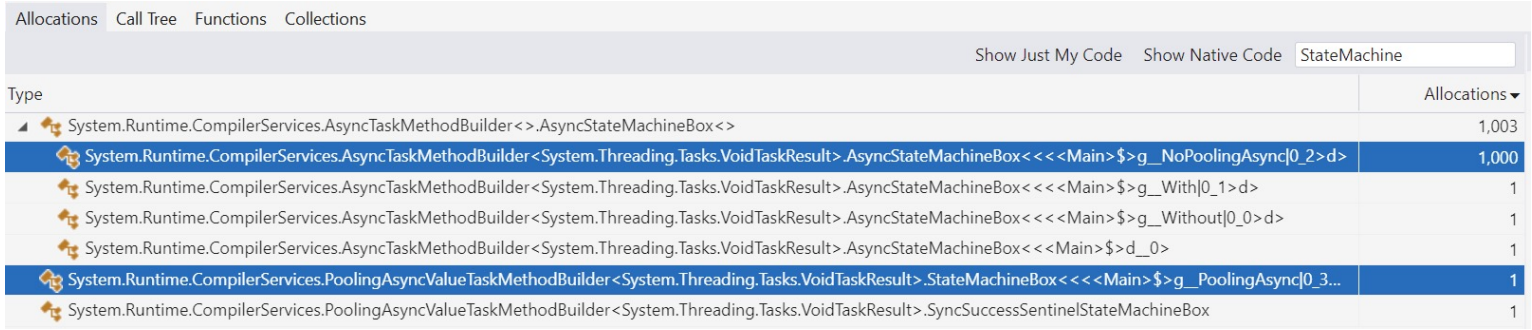
- “Sync-over-async” w/ the thread pool
  - “The One Where We Just Roll With It”

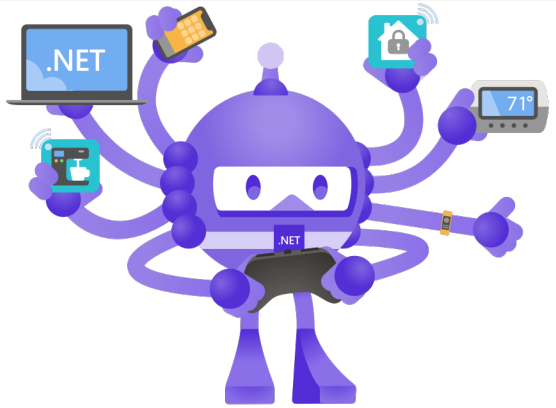




Threading, cont.

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





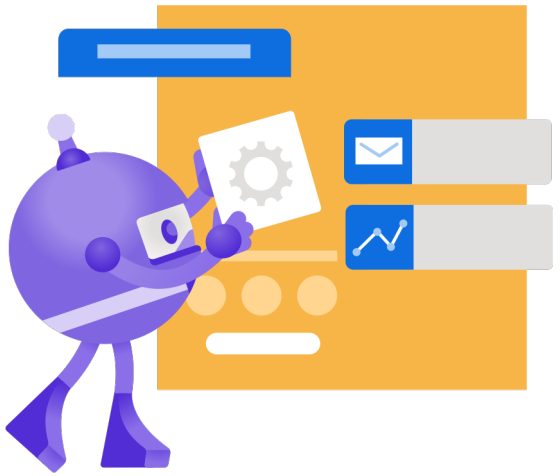
System.*

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

˝ ¸ ĴИ¸j Ă ğCj ĂàẤŔğəġġĄ ńţ ḿŤĂЛ ğCj ĂàVIVŔIVWVŔIVWVÂǨ
˝ ¸ ĴИ¸j Ă ≥¸Ťčŭţ Ŗ ¸Ťčŭţ ńţ ḿŤĂЛ ≥¸Ťčŭţ æǾǨ

à: ĂŤİ īţ ˛ ŌÂ
˝ əğőĴİ ИŭĴč ãĂPj : Cj ĂģáǾńţ ţ ḿŖ ¸Ťčŭţ ææĂPj : Cj ĂģáŔğəġġĄ ǾǨ

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



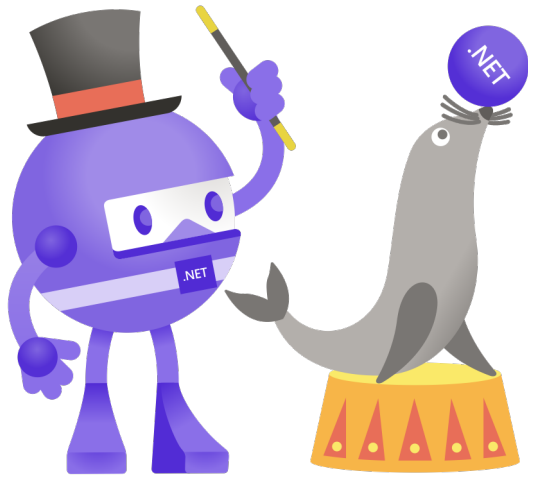
System.*, cont.

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

˝ ¸ ĴИ¸ j Ă ģj¸ ĴŤĢ ŘĢəĴčḿṭ ḿ] əĴčāãĂЛ] əĴčáǾāę ŭТ]¸ ĴŤĢáǾǨ

à: ĂŤİ ī ţ „ ŌÂ
˝ əğőĴİ] əĴč ∞„ ģĂáǾḿṭ ţ ḿ] əĴčā»„ ģĂáŘĢəĴčǾǨḿ

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

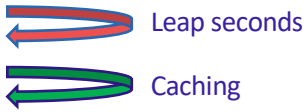


System.*, cont.

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

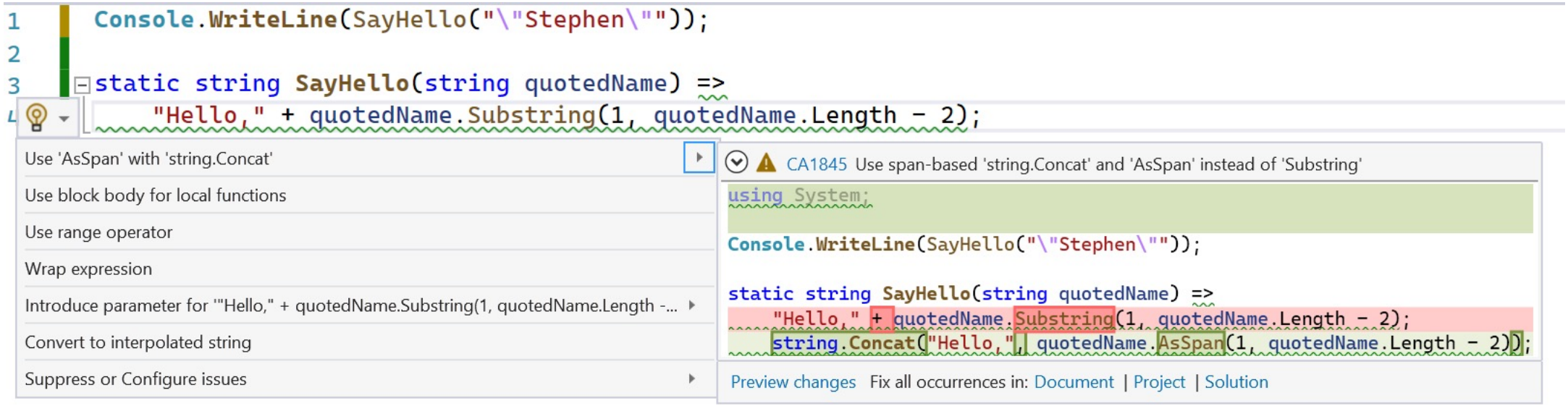
à: ĂŤİ ī ţ ˛ ŌÂ
˝ ɵğőĴİ B˛ j Ăį Ĵţ Ă »j İ ãŭЛæØḿţ ţ ḿB˛ j Ăį Ĵţ Ăæ›j İ ãŭЛĶ

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”



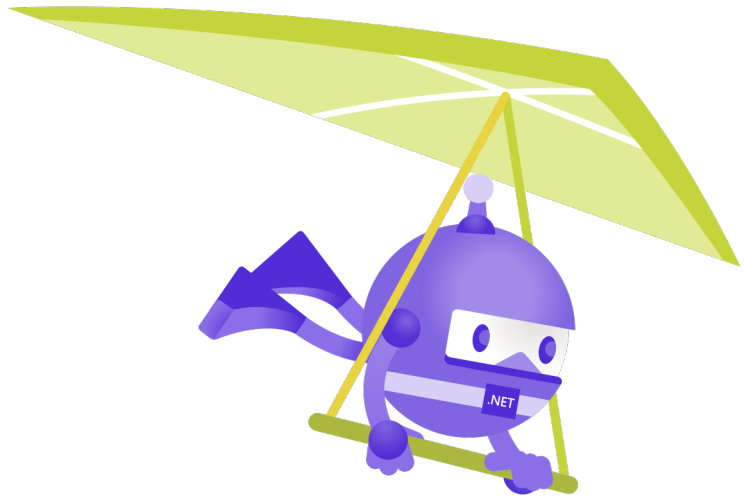
System.*, cont.

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

```
˝ ¸ ĴИ¸j Ă İ ī „ àÂńŘ„ ¸ ¸ Cńţ ńŤĂЛ İ ī „ àVVXÂĂŇ
˝ ¸ ĴИ¸j Ă İ ī „  Řİ ńţ ńÖİ ÖŇ

 à: ĂŤİ ī ţ „ ŌÂ
˝ əğőĴİ  ИŭĴč \ Ĵőőáə Ńţ ţ ń ¸ ¸ Cæà Ĵőőáə Ř„ ¸ ¸ CĞńŘİ ŃŇ
```

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”

```
˝ ¸ ĴИ¸ j Ă ǵj¸ ĴŤĢ Ȑǵj¸ Ǩ

à] őŭğ¸ őΠĂj ő Â
˝ əğőĴİ ¸ǵCŤİ ¡ ¸ǵŌ ΠĂj ő áǾ
ǿ
      əǵĴŤĢ И¸ ī İ ţ ḿŤĂЛ dj ĵ˝ ; őĴĂŤj áǾǨ
      Ȑǵj¸ ḿţ ḿ¸ Л¸ Ĵj ī İ áǭ Ăj Π¸ ĴŤĢ ǵCŤİ áęī j ĵ˝ ǵğǺǺЛЛāĢəj ĂŤğĀ Ģāǘ ĢǺİ ¸ İ ī ĂǺĂ əğǺVШVǺ ĢVШVāę Pj ęǾǨ
À

à: ĂŤİ ī ţ ¸ ŌÂḿ əğőĴİ ǵj¸ ĴŤĢ ÊĂőőáǾḿţ ţ ḿȐǵj¸ āěĂ ő¸ İ ĂáęāęǦḿęǪęǾǨ

à: ĂŤİ ī ţ ¸ ŌÂḿ əğőĴİ ǵj¸ ĴŤĢ ; ŭŤİ ¸ j áǾḿţ ţ ḿȐǵj¸ āěĂ ő¸ İ ĂáęáŤęǦḿęęǾǨ
```

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	-

ģj ¸j ĴÌ ИŭĴč ˝˝ ĂŤčæ¶¸ ĴŤÇ: əĴőčĂ ģğĞńĴŤj ţ¸Ņŭ ĞńĴŤj ţĴŤŭ ĞńĴŤj ğəĴőčĞńĴŤj ¸ ĂИĴģĴüŤÓńţţ
ģğæ˝ ĂŤčáÑ¸ótţ¸Ņŭ ÀǣótĴŤŭ ÀǣóğəĴőčÀǣó ĂИĴģĴüŤÀ¸ÓǨ

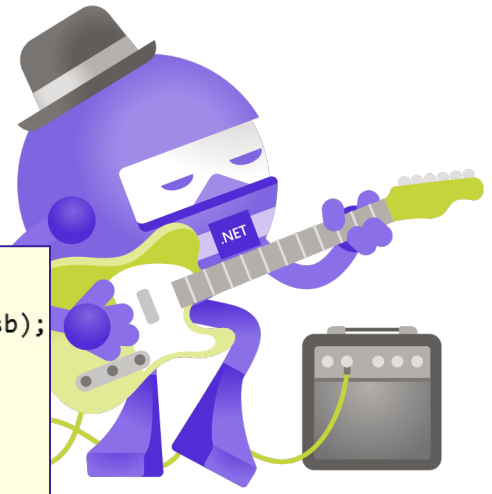
.NET 5 + C# 9

```
private static void Append(Str
{
    sb.Append(string.Format("{
}
```

```
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

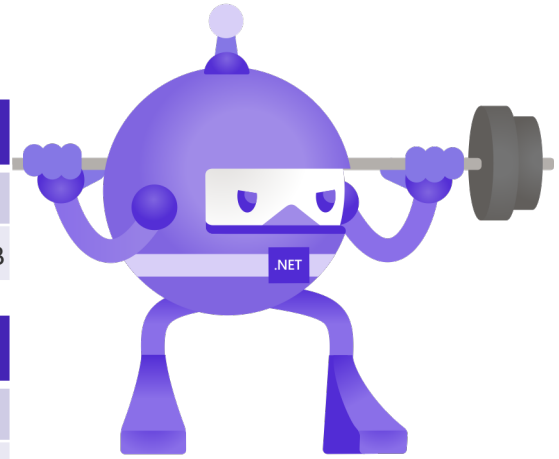
- Max Array Size
  - “The One Where We Remove The Cliff”

```
à: ĂŤİ ī ţ „ Ōáé ¸ ģĂőĴŤĂṁṭ ṁ¸ əĂØÂ
˝ əğőĴİ ИŭĴč ≥ĂŤj ≥Ăj ę ŤŘVIWWXIDXXKáØṁṭ ţ ṁ¸ ¸ ¸C∞ŭŭőŢğCį Ăţ æĘTī „ Ăčāə≥Ăj ę Ťáé
    ¸ ¸ ¸C∞ŭŭőŢğCį Ăţ æĘTī „ Ăčāə≥ĂŤj æVIWWVIIDźVIWWVØØK̂

à: ĂŤİ ī ţ „ ŌÂ
˝ əğőĴİ ИŭĴč ≥ĂŤj ≥Ăj ę ŤŘVIWWXIDXXKáØṁṭ ţ ṁ¸ ¸ ¸C∞ŭŭőŢğCį Ăţ æĘTī „ Ăčāə≥Ăj ę Ťáé
    ¸ ¸ ¸C∞ŭŭőŢğCį Ăţ æĘTī „ Ăčāə≥ĂŤj æVIWWVIIDźVIWWVIIţṁVØØK̂
```

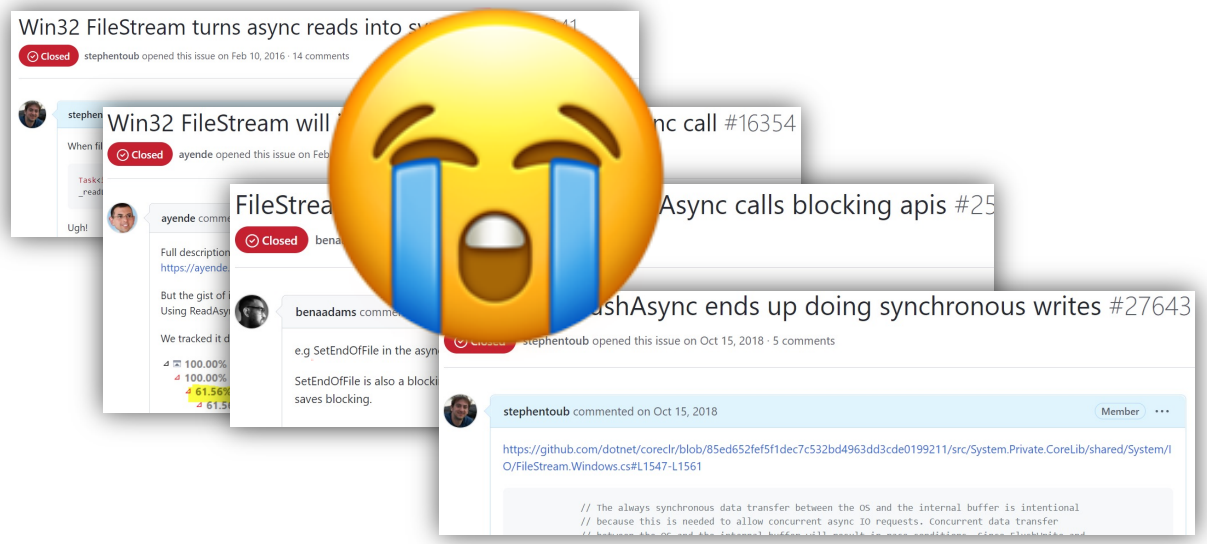
Benchmark	Version	Time	Ratio	Allocated
RentReturn_1048576	.NET 5.0	21.01 ns	1.00	
RentReturn_1048577	.NET 5.0	12,132.90 ns	577.48	1,048,593 B

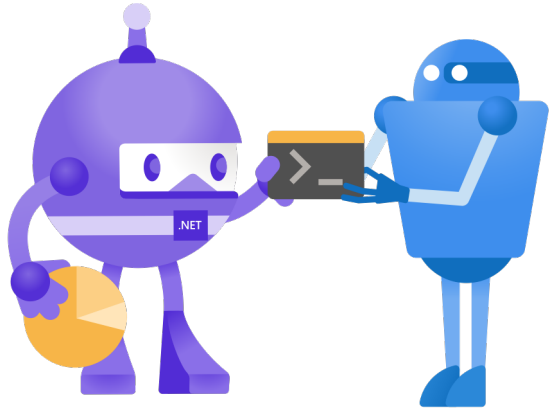
Benchmark	Version	Time	Ratio	Allocated
RentReturn_1048576	.NET 6.0	16.36 ns	1.00	
RentReturn_1048577	.NET 6.0	16.38 ns	1.00	



File IO

- FileStream (blog)
  - “The One Where We Make Async Actually Async”



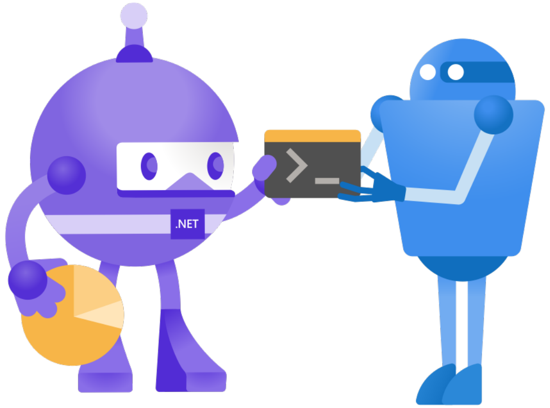


File IO

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

[illegible]

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”

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”

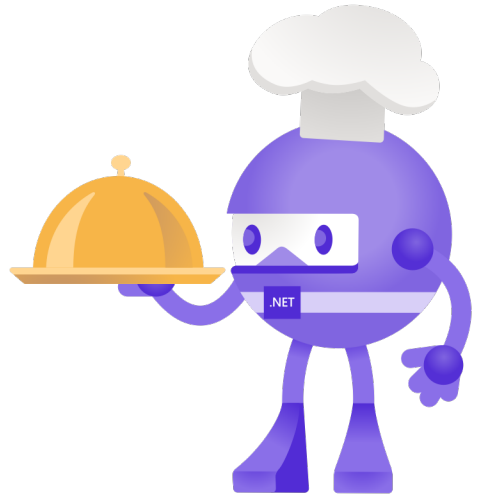
˝ ¸ ĴИ¸ j Ă ¡ ḿ¸ Ă¸ j ĂŢ¡ ţ áǾḿЛτ Ą Ă ¡ ḿğḿŤĂЛáǾḿţ ţ ḿŤĂЛ ¡ áǾǨ

à: ĂŤİ ī ţ „ ŌÂ
˝ əğőĴİ ∞ ŭĢ ¸ ţ ;¸ Ă¸ j ĂáǾḿţ ţ ḿ¸ Ă¸ j ĂŢ∞ ŭĢ ¸ ţ ţ áǾǨ

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”

˝ ¸ ĴИ¸ j Ă äĂj ī ŭči Ťġŭ ŘŢ Ăj ī ŭčḿṭ ḿj C Ăŭġáə» ŭĢ ¸ ţ Ćāą Ăj äĂj ī ŭčáęäęĆǨ

˝ əğőĴİ ИŭĴč äáəĴŤj „ ĢVĞḿǵj¸ ĴŤĢ „ ĢVĆḿómÀ

à: ĂŤİ ī ţ „ ŌÂ
˝ əğőĴİ ИŭĴč i ŤИŭŌĂáĆḿṭ ţ ḿŘŢ Ăj ī ŭčāə ŤИŭŌĂą ī Ĵǵ ĞḿŤĂЛ ŭğŊĂİ j àÂḿómVĞḿęj ЛŭęḿÀĆǨ

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”

```
˝ ¸ ĴͶ¸ j Ă i FŤəţ Ą̆ ¸ ğőĂŢ̂ĴŤj ṯ ḿȐĂVḿț ḿFŤəţ Ą̆ ¸ ğőĂǣ≥¸ ŤÇĂǽVǦḿVȐIVVVȐIVVVǾǣ ŭ ¸ ¸ ¸ CǽǾǨ
˝ ¸ ĴͶ¸ j Ă i FŤəţ Ą̆ ¸ ğőĂŢ̂ĴŤj ṯ ḿȐĂVḯḿț ḿFŤəţ Ą̆ ¸ ğőĂǣ≥¸ ŤÇĂǽVǦḿVȐIVVVȐIVVVǾǣ ŭ ¸ ¸ ¸ CǽǾǨ

ȁ: ĂŤİ ī ţ ̮ ŌÂ
˝ əğőĴİ ğŭŭő ŢĄ̆ əĂŤİ ĂF̦ ə¸ őǽǾḿț ṯ ḿȐĂVǣŢĄ̆ əĂŤİ ĂF̦ ə¸ őǽȐĂVǾǨ
```

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

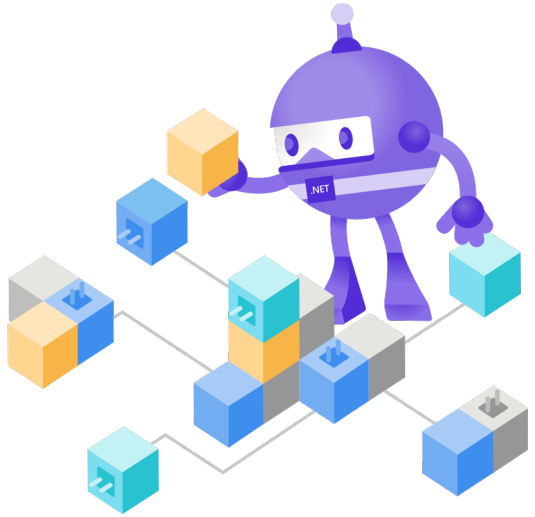
LINQ, cont.

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

˝ ¸ ĴИ¸ j Ă i FŤəţ Ą ¸ ğőĂŢģj¸ ĴŤÇţ mŘč¸ j ¸ mţ mFŤəţ Ą ¸ ğőĂæ≥¸ ŤÇĂáWĞmVIVWŘIVWÓãŢĂőĂİ j áąĴ ţ t mĴãę üŢĮ¸ ĴŤÇáéÓÓãę ü¸ ¸ ¸ CáéÓǨ

à: ĂŤİ ī ţ ¸ ŌÂ
˝ əğőĴİ ĴŤj BĴģj ĴŤİ j ; ũəŤj áéÓmţ t mŘč¸ j ¸ æBĴģj ĴŤİ j áéÓãę ũəŤj áéÓǨ

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”

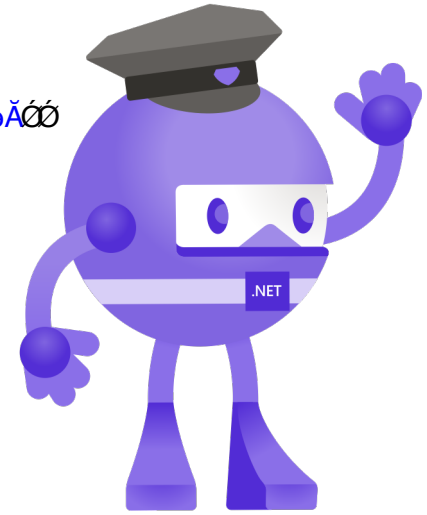
```
public int TransformBlock(byte[] inputBuffer, int inputOffset, int inputCount, byte[] outputBuffer, int outputOffset)
{
    // inputCount < InputBlockSize is not allowed
    ThrowHelper.ValidateTransformBlock(inputBuffer, inputOffset, inputCount, InputBlockSize);

    if (outputBuffer == null)
        ThrowHelper.ThrowArgumentNull(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

ĂŤğḿ, ɘĂØØ

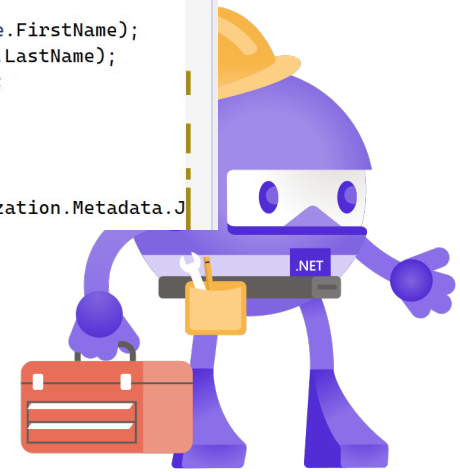


JSON

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

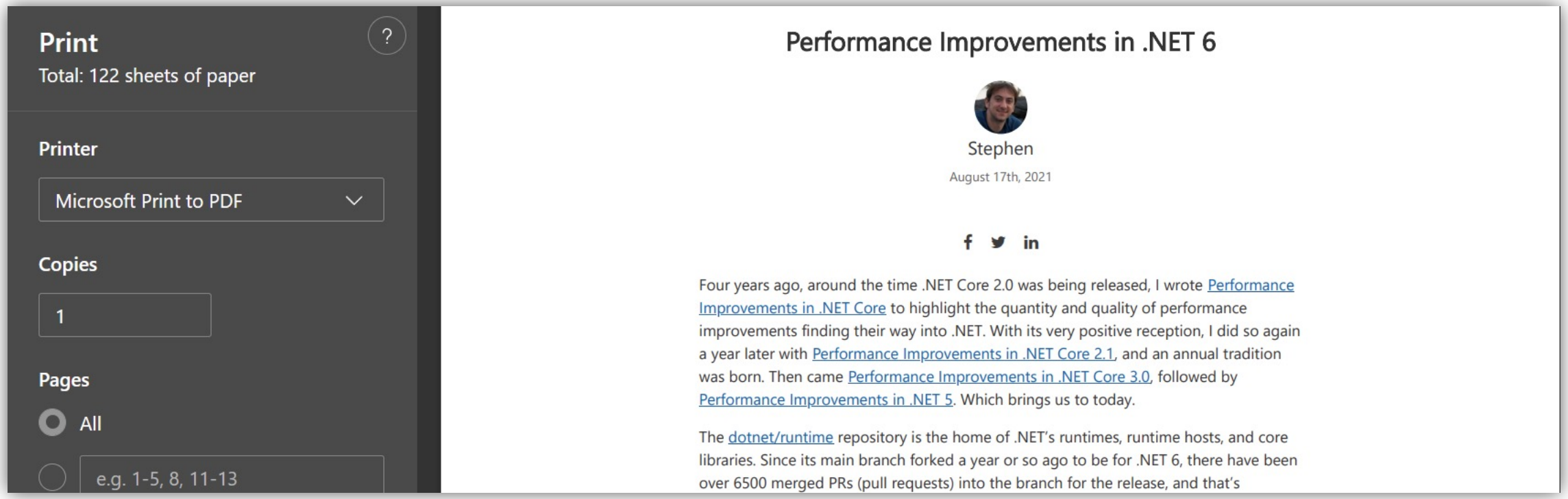
```
Program.cs
ConsoleApp66 | MyJsonContext | String
1   using System.Text.Json.Serialization;
2
    17 references
3   record Person(string FirstName, string LastName, int Age);
4
5   [JsonSerializable(typeof(Person))]
    13 references
6   internal partial class MyJsonContext : JsonSerializerContext
```

```
MyJsonContext...cs [generated]
This file is auto-generated by the generator 'System.Text.Json.SourceGeneration.JsonSourceGenerator' and cannot be edited.
Miscellaneous Files | MyJsonContext | Person
95   private static void PersonSerialize(global::System.Text.Json.Utf
96   {
97       if (value == null)
98       {
99           writer.WriteNullValue();
100          return;
101      }
102
103      writer.WriteStartObject();
104      writer.WriteString(PropName_FirstName, value.FirstName);
105      writer.WriteString(PropName_LastName, value.LastName);
106      writer.WriteNumber(PropName_Age, value.Age);
107
108      writer.WriteEndObject();
109  }
110
111  private static global::System.Text.Json.Serialization.Metadata.J
```



More generators coming in .NET 7…

And so much more…



Thanks for joining!