



What's new in Visual Studio 2019 and some //Build announcements

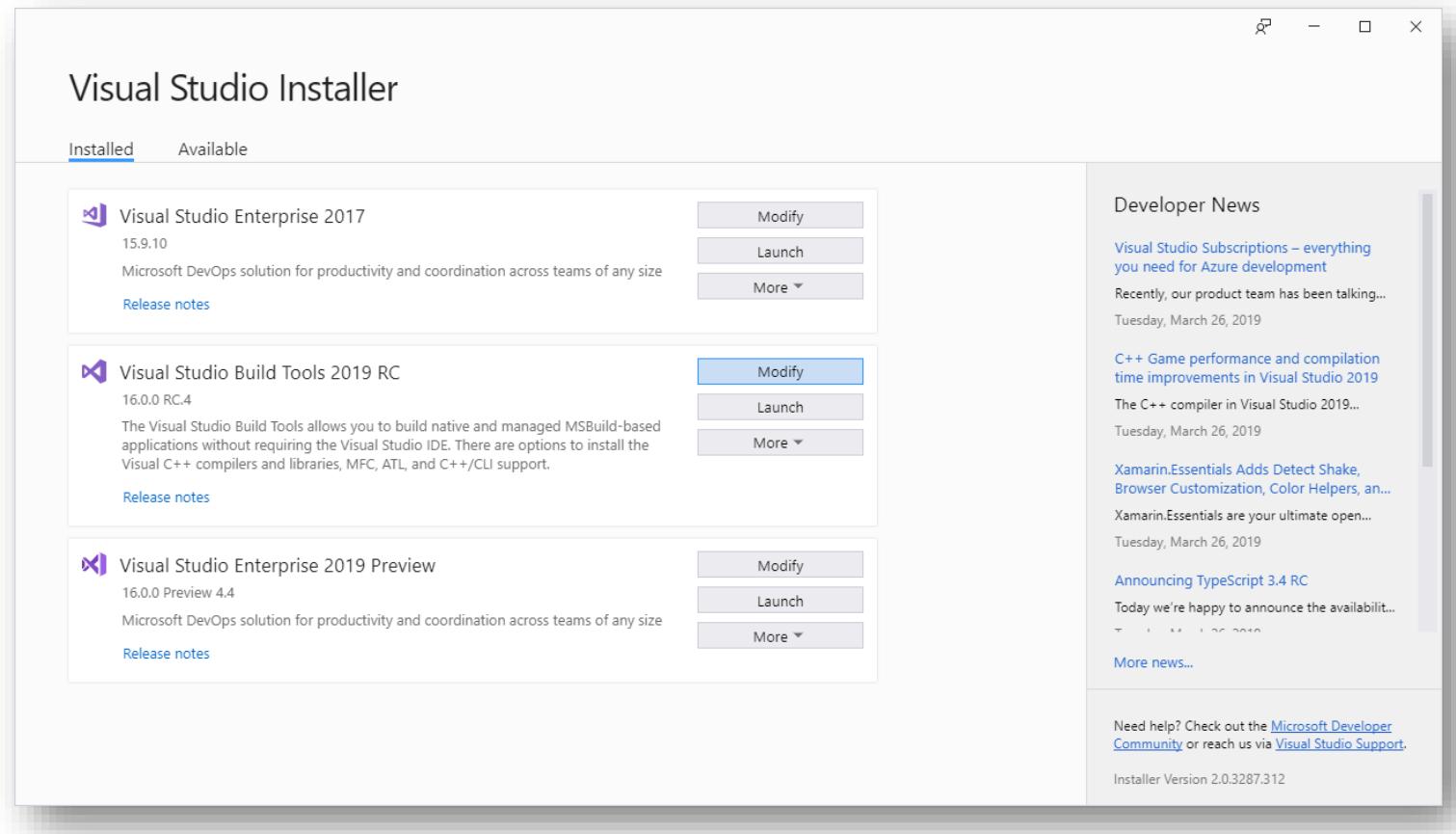


Gianni Rosa Gallina
 @giannirg

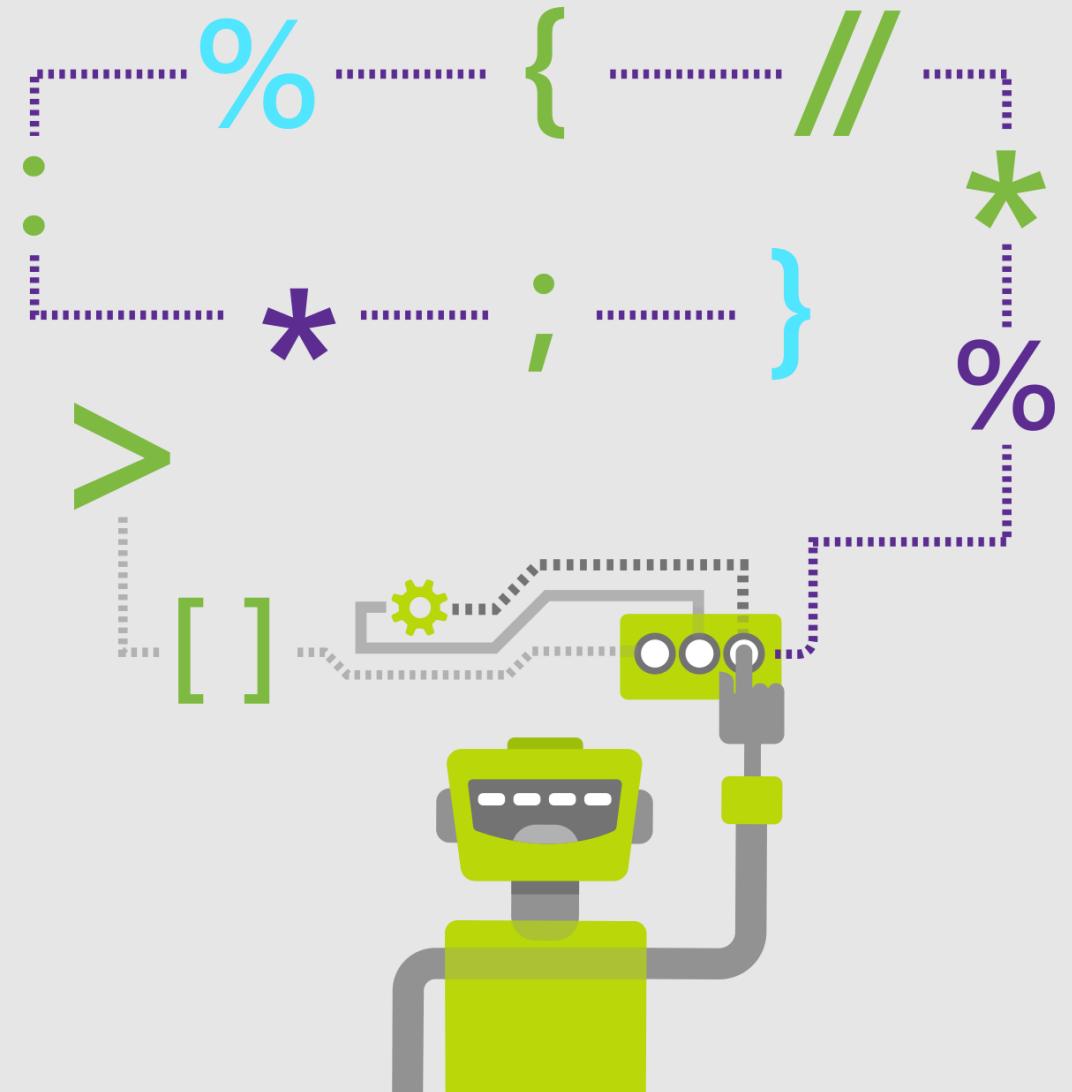
Visual Studio 2019 launch

Pain-free upgrades

Install the latest IDE side-by-side with your current one
Updates download in *background*

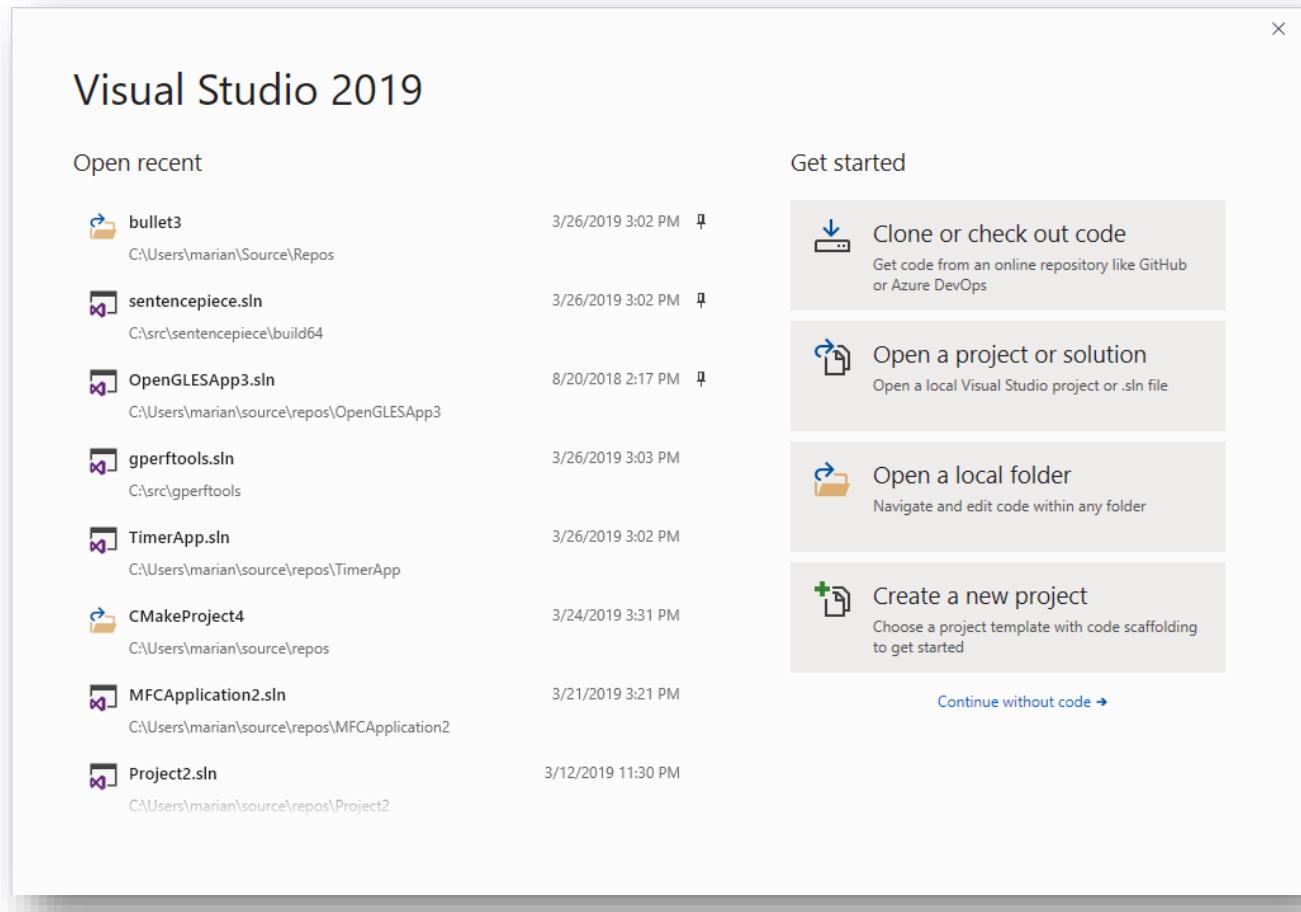


UX and Shell



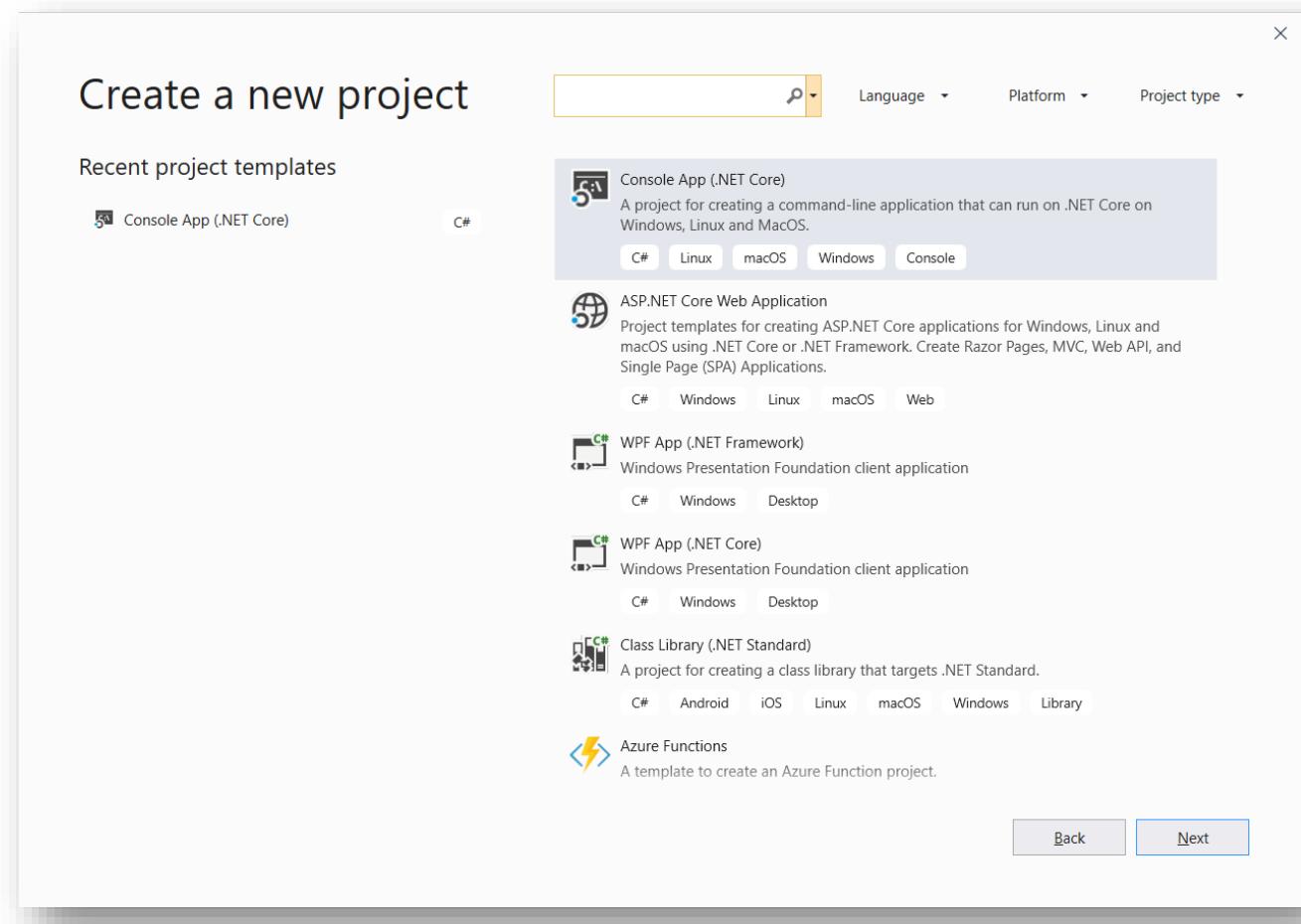
UX and Shell

New Start Window



UX and Shell

New Project Window



UX and Shell

Redesigned UX and Blue Theme

New classification colors (similar to the Visual Studio Code)

The screenshot shows the SmartEditor IDE interface. The main area is a code editor displaying a C# file named `TextAnalyticsClient.cs`. The code implements an asynchronous method `AnalyzeTextAsync` that performs language detection and entity extraction. The Solution Explorer on the right shows a project structure with files like `BingSearchClient.cs`, `TextAnalyticsAnalysisResult.cs`, and `TextAnalyticsClient.cs`. The Properties window is also visible at the bottom.

```
61     result.Entities = mappedEntities.DistinctBy(e => e.Text).ToList();
62     result.Categories = analysisResult.Categories;
63     result.Score = 0;
64     result.Language = analysisResult.Language;
65     result.AnnotatedHTML = analysisResult.AnnotatedHTML;
66     return result;
67   }
68
69   private async Task<TextAnalyticsAnalysisResult> AnalyzeTextAsync(string text, string language = "")
70   {
71     var result = new TextAnalyticsAnalysisResult();
72
73     if (string.IsNullOrWhiteSpace(language))
74     {
75       var detectedLanguages = await DetectLanguageAsync(text).ConfigureAwait(false);
76       if (detectedLanguages.documents.Count > 0 && detectedLanguages.documents[0].detectedLanguages.Count >
77       {
78         language = detectedLanguages.documents[0].detectedLanguages[0].iso6391Name;
79       }
80     }
81
82     result.Language = language;
83     if (!string.IsNullOrWhiteSpace(language))
84     {
85       var entitiesResult = await GetEntitiesAsync(text, language).ConfigureAwait(false);
86       if (entitiesResult.documents.Count > 0)
87       {
88         result.Entities = entitiesResult.documents[0].entities;
89       }
89     }
90   }

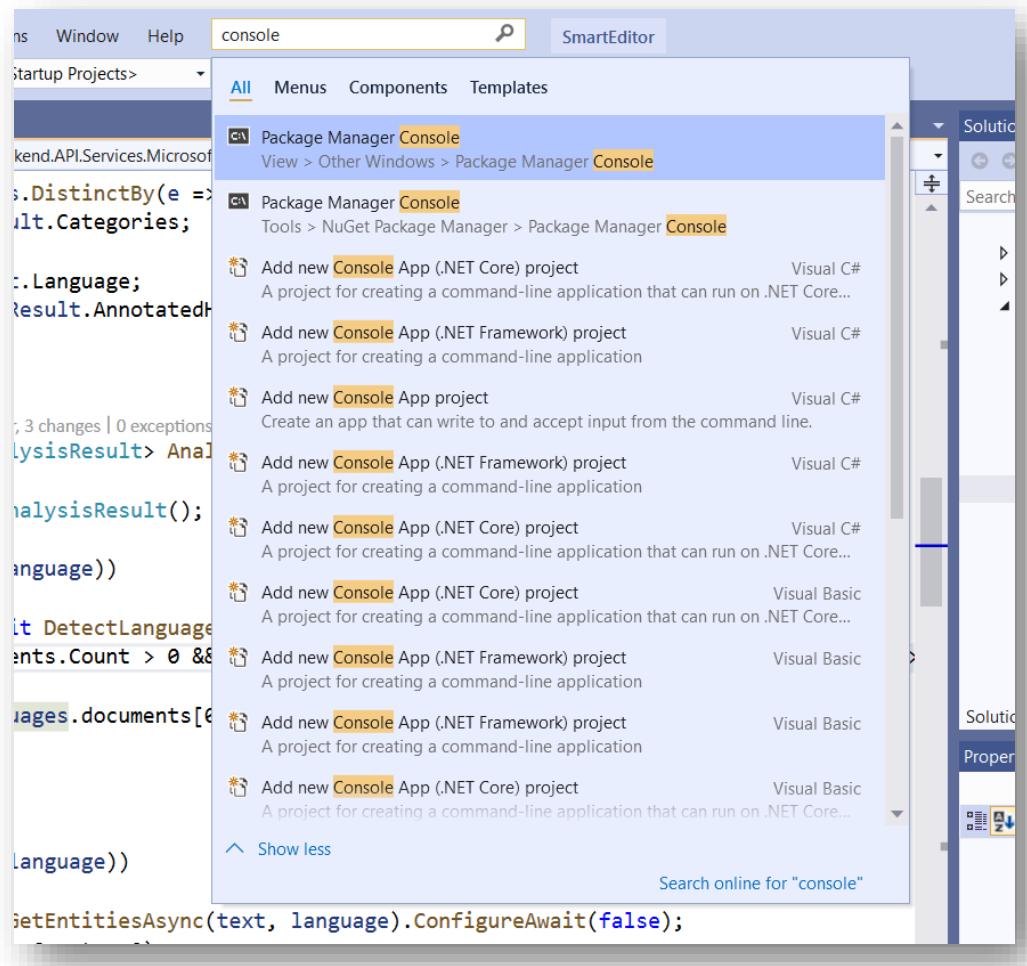
```



UX and Shell

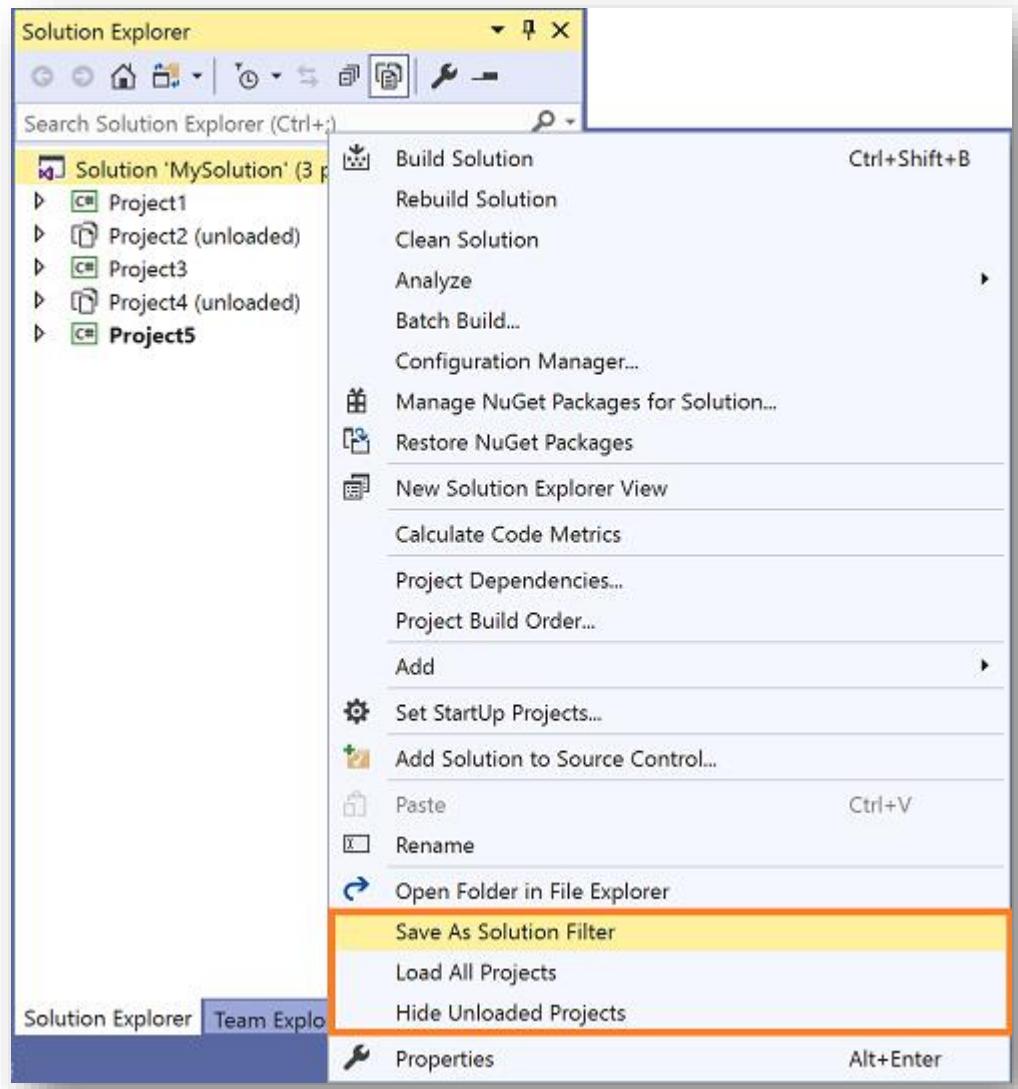
Visual Studio Search experience

CTRL+Q



UX and Shell

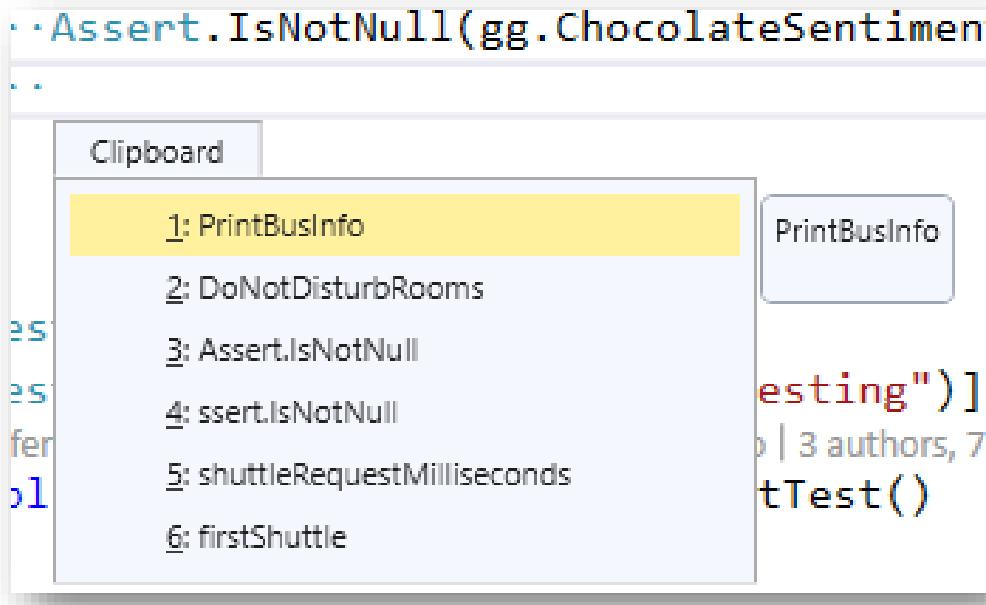
Solution Filters



UX and Shell

Clipboard Content History

CTRL + SHIFT + V



UX and Shell

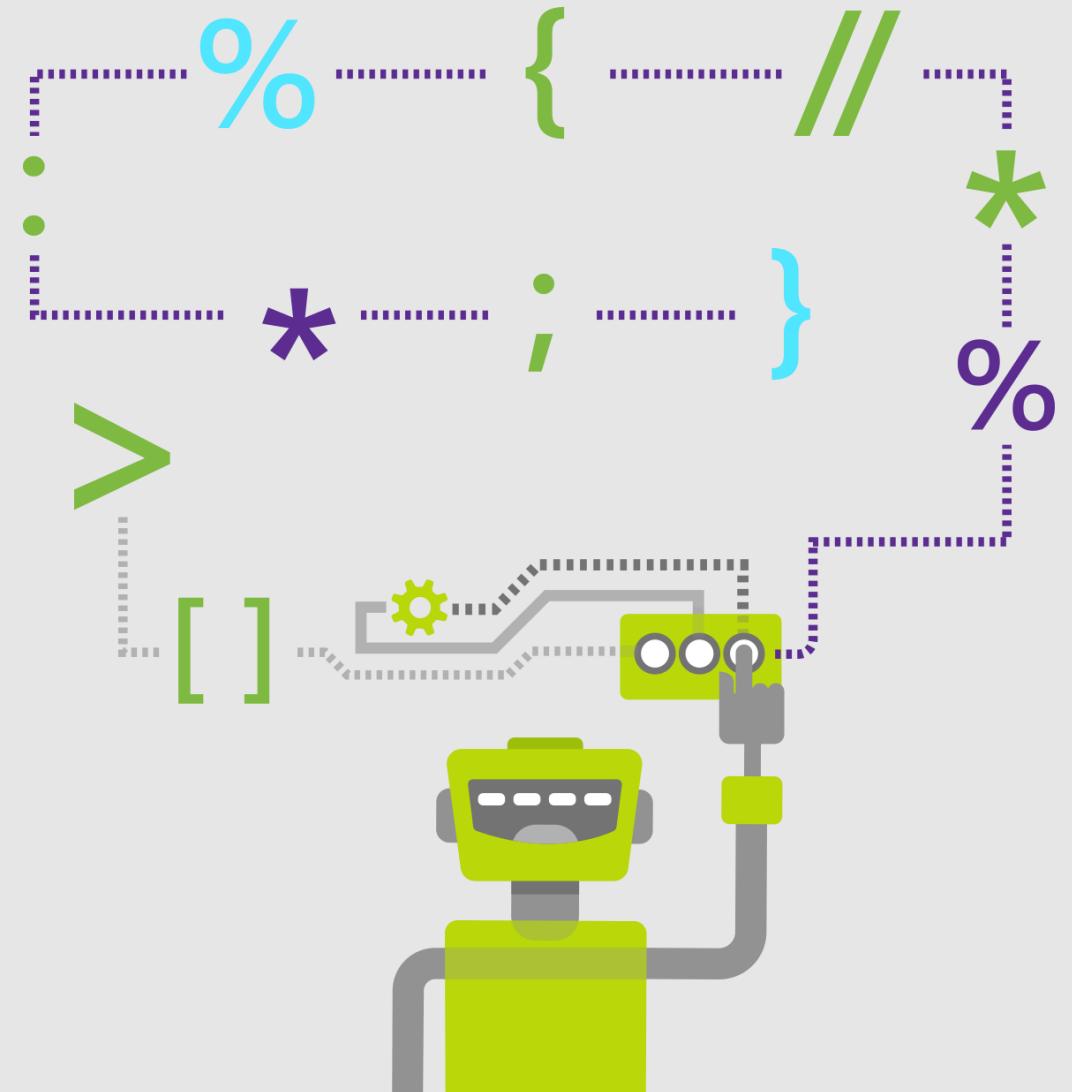
- CodeLens is now in *Community edition*
- Tags in Extensions and Updates (Preview/Trial)
- Per monitor awareness (PMA)
 - Render more sharply regardless of your display configuration and/or scaling

<https://docs.microsoft.com/en-us/visualstudio/releases/2019/release-notes#ide>

<https://docs.microsoft.com/en-us/visualstudio/releases/2019/release-notes-v16.0#ide>

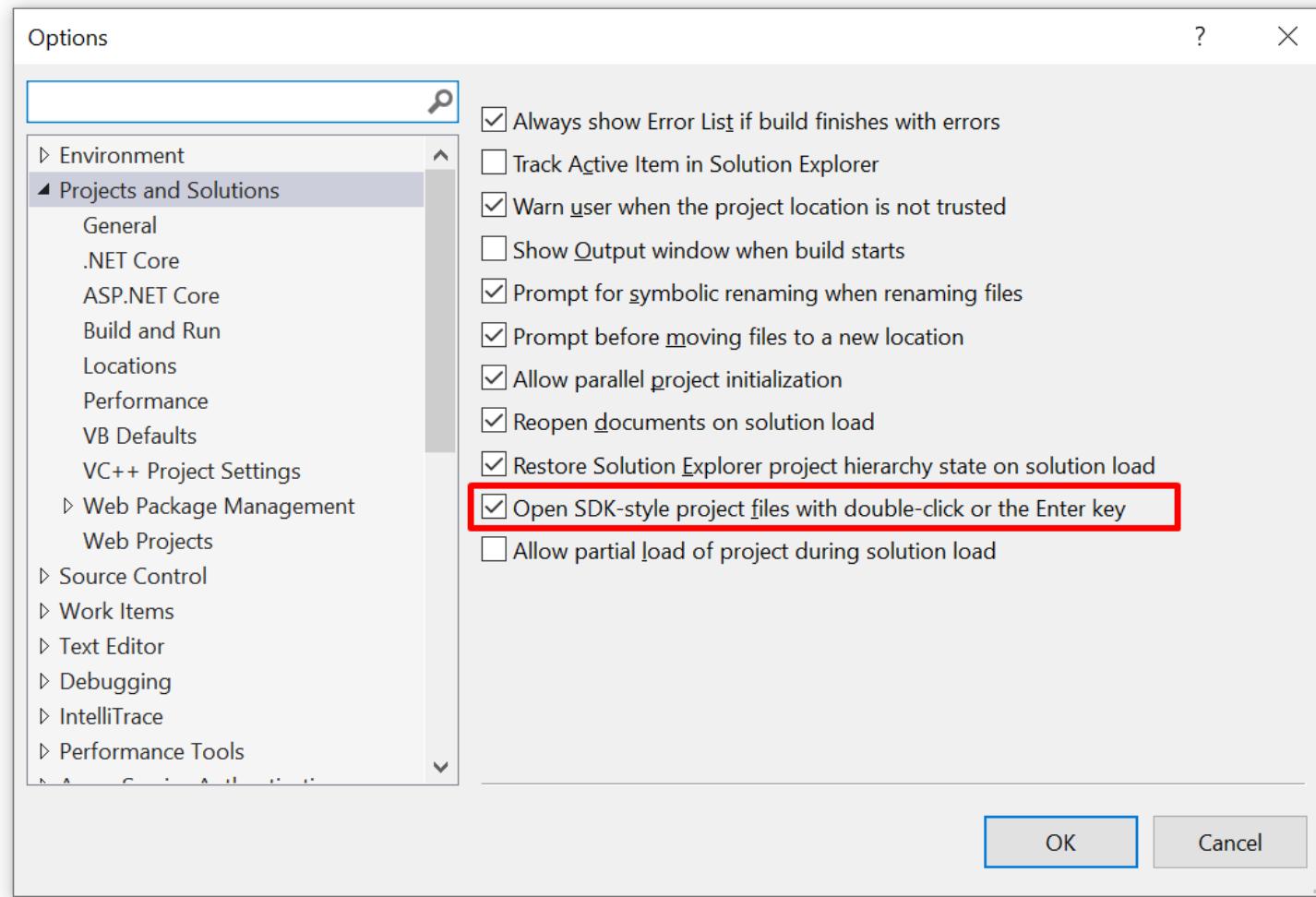


Tooling Improvements



Tooling Improvements

Double-click Project file editing (*SDK-style projects*)



Tooling Improvements

Find all references can now filter by Read/Write, Types, Namespaces

The screenshot shows a code editor with a search results pane at the bottom. The code editor displays a C# file with several lines of code. The search results pane is titled "'language' references" and shows a list of 5 references for the variable 'language'. The results are grouped by project and definition, and the 'Code' tab is selected. The results table includes columns for File, Line, Col, Project, and Kind. The 'language' variable is used both as a parameter and as a local variable.

File	Line	Col	Project	Kind
TextAnalyticsClient.cs	73	43	SmartEditor.Backend.API	Read
TextAnalyticsClient.cs	82	31	SmartEditor.Backend.API	Read
TextAnalyticsClient.cs	83	44	SmartEditor.Backend.API	Read
TextAnalyticsClient.cs	85	67	SmartEditor.Backend.API	Read
TextAnalyticsClient.cs	78	21	SmartEditor.Backend.API	Write

Tooling Improvements

Search local/watch windows during debugging

The screenshot shows a debugger interface with the following elements:

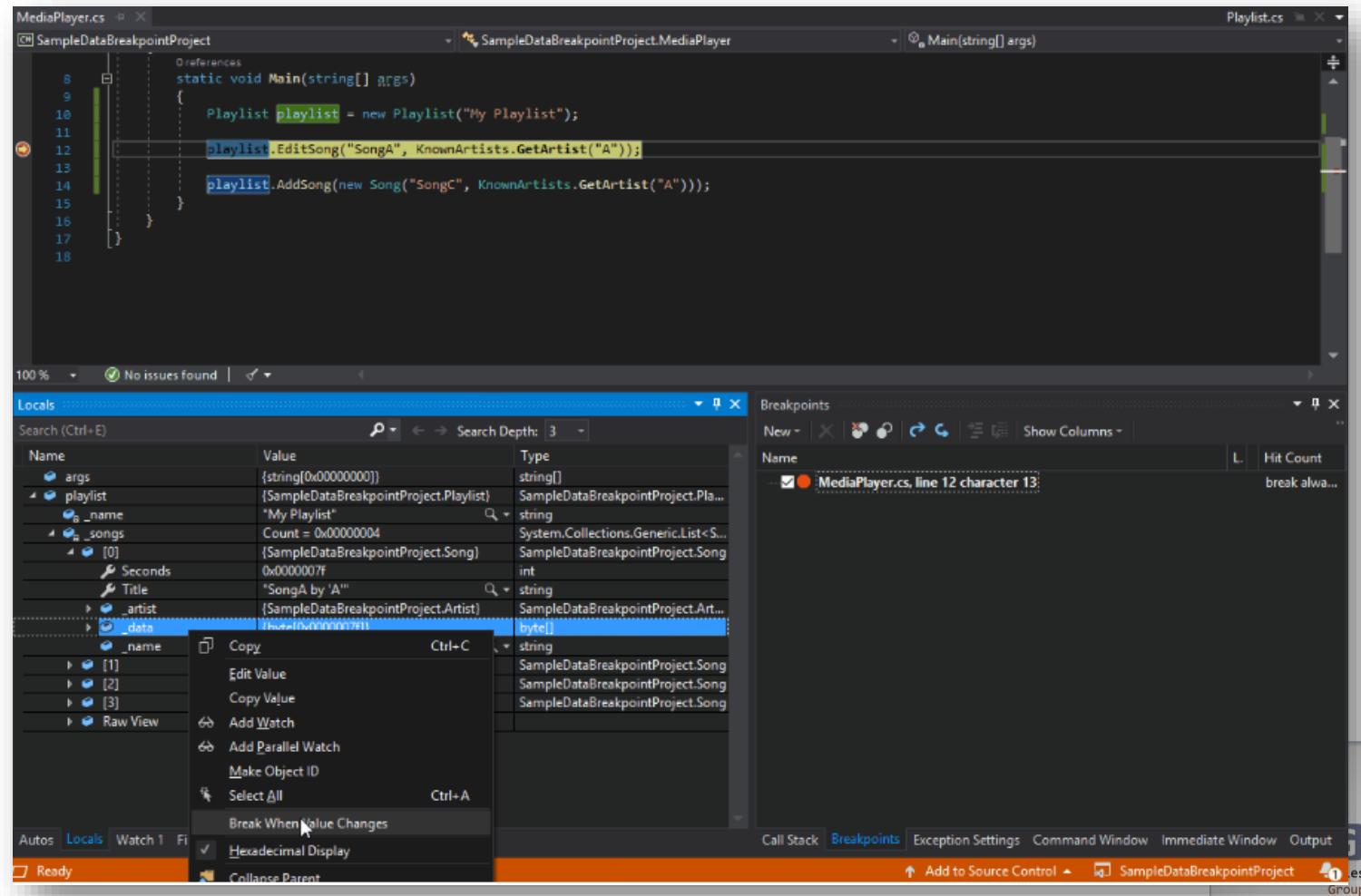
- Code Editor:** Displays C# code for a `AnalyzeTextAsync` method. A red dot at line 57 indicates the current line of execution. The code handles language detection and entity extraction.
- Watch 1 Window:** Shows the variable `language` with a value of "en". It also lists `result`, `AnnotatedHTML`, `Categories`, `Entities`, and `Language`. The `Entities` variable is expanded to show a count of 33 and a partial string value.
- Call Stack Window:** Lists the call stack entries, starting with `SmartEditor.Backend.API.dll!SmartEditor.Backend.API.Services.SpaCy.S`.
- Status Bar:** Shows "121 %", "No issues found", and a timer indicating "< 102ms elapsed".

Tooling Improvements

Managed Data Breakpoints

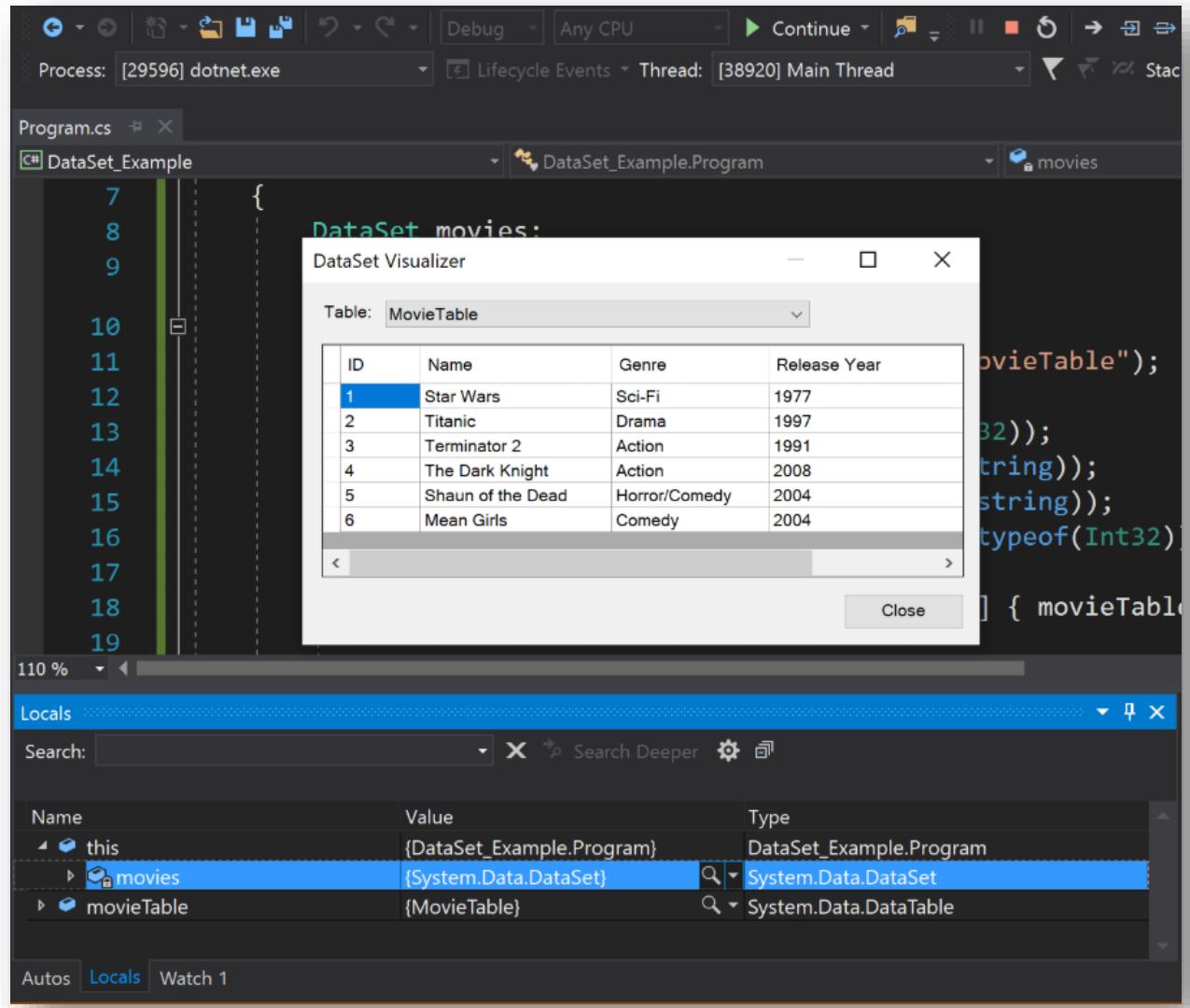
Break when a specific object's property value changes in .NET Core (3.0 or higher)

Already available for C++ projects



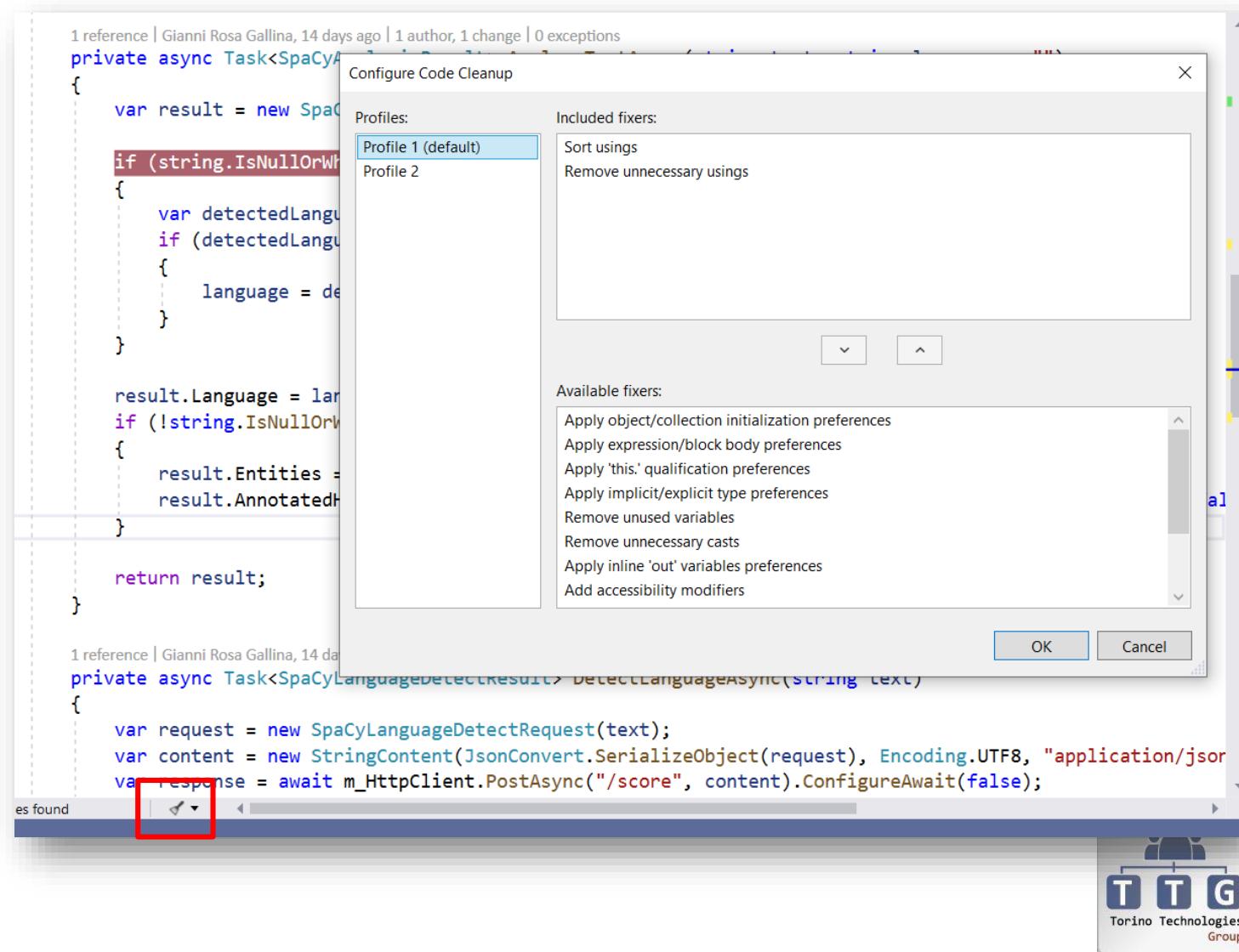
Tooling Improvements

Dataset Visualizer (.NET Core 3.0 or higher)



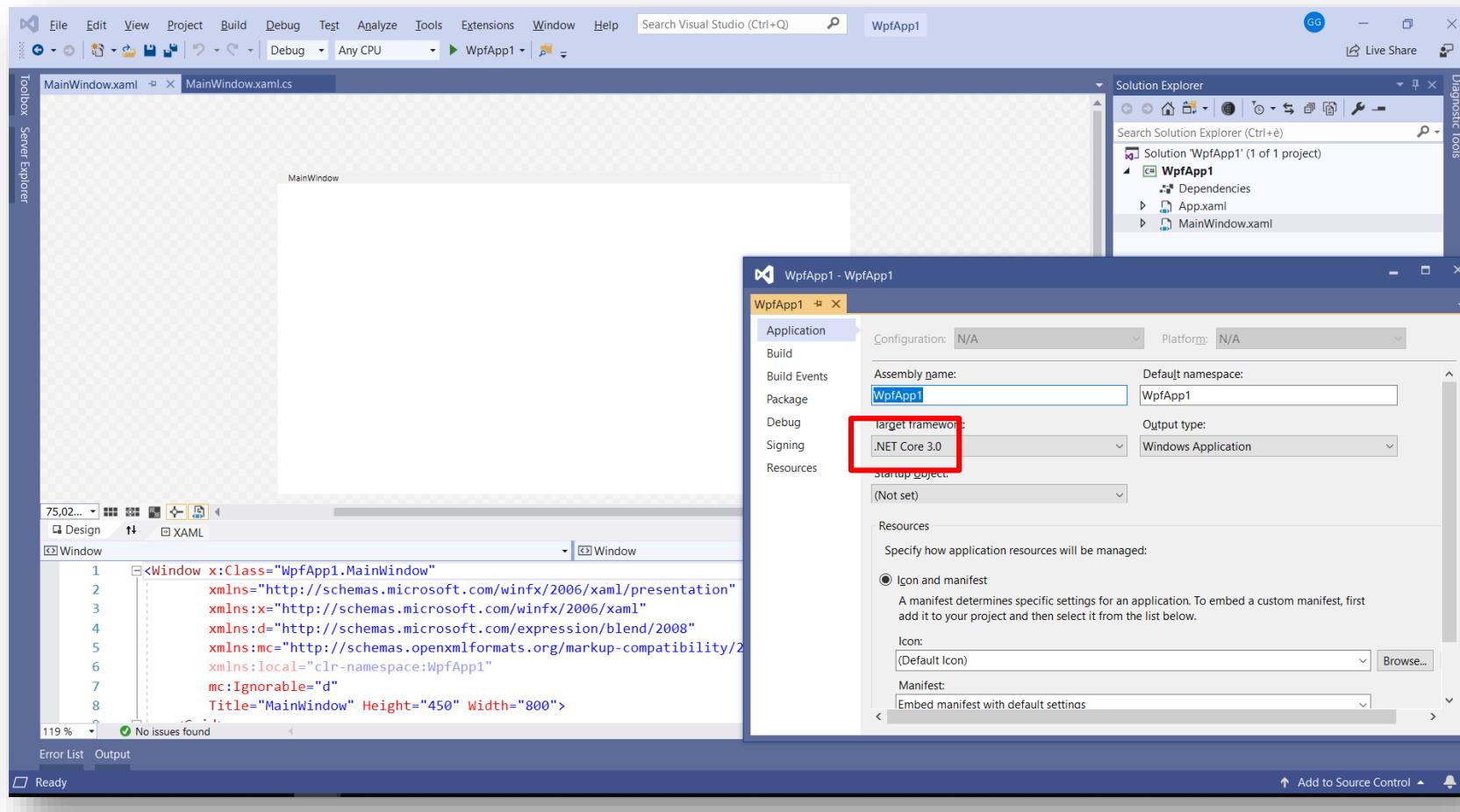
Tooling Improvements

- Export code style in Tools > Options to .editorconfig
- Code cleanup
- dotnet format command line tool
<https://github.com/dotnet/format>



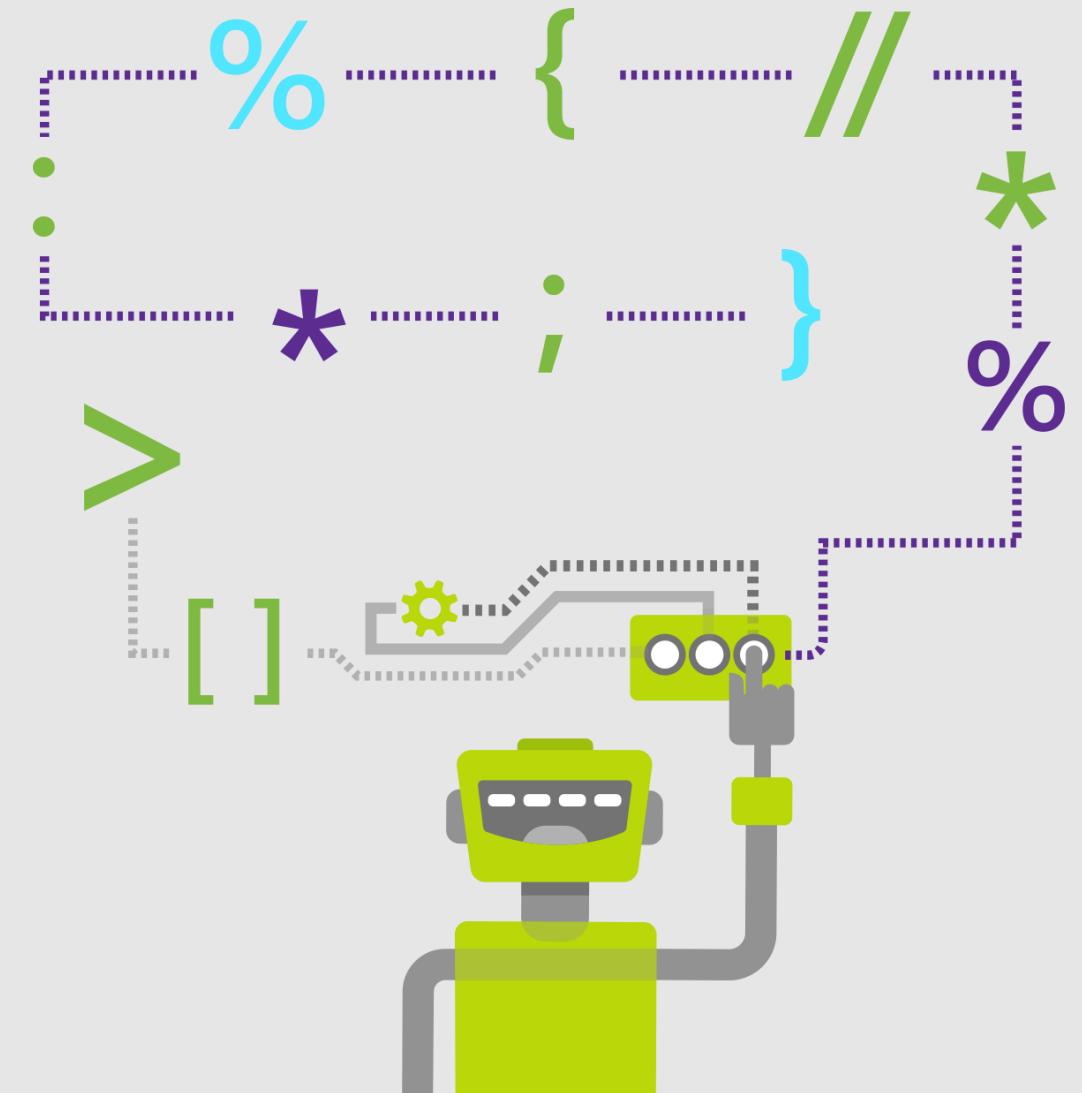
Tooling Improvements

WPF Designer (Preview) (.NET Core 3.0 or higher)

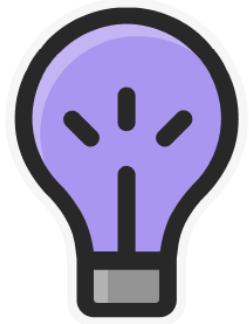


IntelliCode

AI-assisted IntelliSense



IntelliCode – AI-assisted IntelliSense



Contextual IntelliSense recommendations

- Leverage the wisdom of the community
- Works with C#, XAML (C++, TypeScript/JavaScript - in *preview*)

A screenshot of the Visual Studio code editor illustrating IntelliCode's contextual suggestions. The code being edited is:

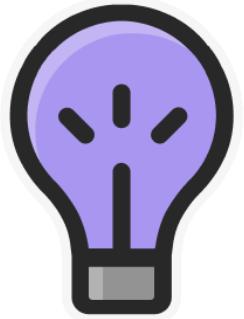
```
private async Task<TextAnalyticsAnalysisResult> AnalyzeTextAsync(string text, string language = "")  
{  
    var result = new TextAnalyticsAnalysisResult();  
  
    if (string.IsNullOrWhiteSpace(language))  
    {  
        var detectedLanguages = await DetectLanguageAsync(text).ConfigureAwait(false);  
        if (detectedLanguages != null) // local variable TextAnalyticsLanguageDetectResult detectedLanguages  
        {  
            language = detectedLanguages.documents[0].detectedLanguages[0].iso6391Name;  
            language = language.  
        }  
    }  
  
    result.Language = language;  
    if (!string.IsNullOrEmpty(result.Language))  
    {  
        var entitiesResult = await GetEntitiesAsync(text, language).ConfigureAwait(false);  
        if (entitiesResult != null && entitiesResult.Count > 0)  
        {  
            result.Entities = entitiesResult.documents[0].entities;  
        }  
    }  
}
```

The cursor is positioned at the end of the line `language = language.`. A tooltip shows the following suggestions:

- ★ Trim (+ 2 overloads)
- ★ Split
- ★ Replace
- ★ ToLower
- Aggregate<>
- All<>
- Any<>
- Append<>
- AsEnumerable<>

The first suggestion, `Trim`, is highlighted. The tooltip also includes the text: "Removes all leading and trailing white-space characters from the current `string` object." and "★ IntelliCode suggestion based on this context".

IntelliCode – AI-assisted IntelliSense



Contextual IntelliSense recommendations

- Leverage the wisdom of the community
- Works with C#, XAML (C++, TypeScript/JavaScript - in *preview*)

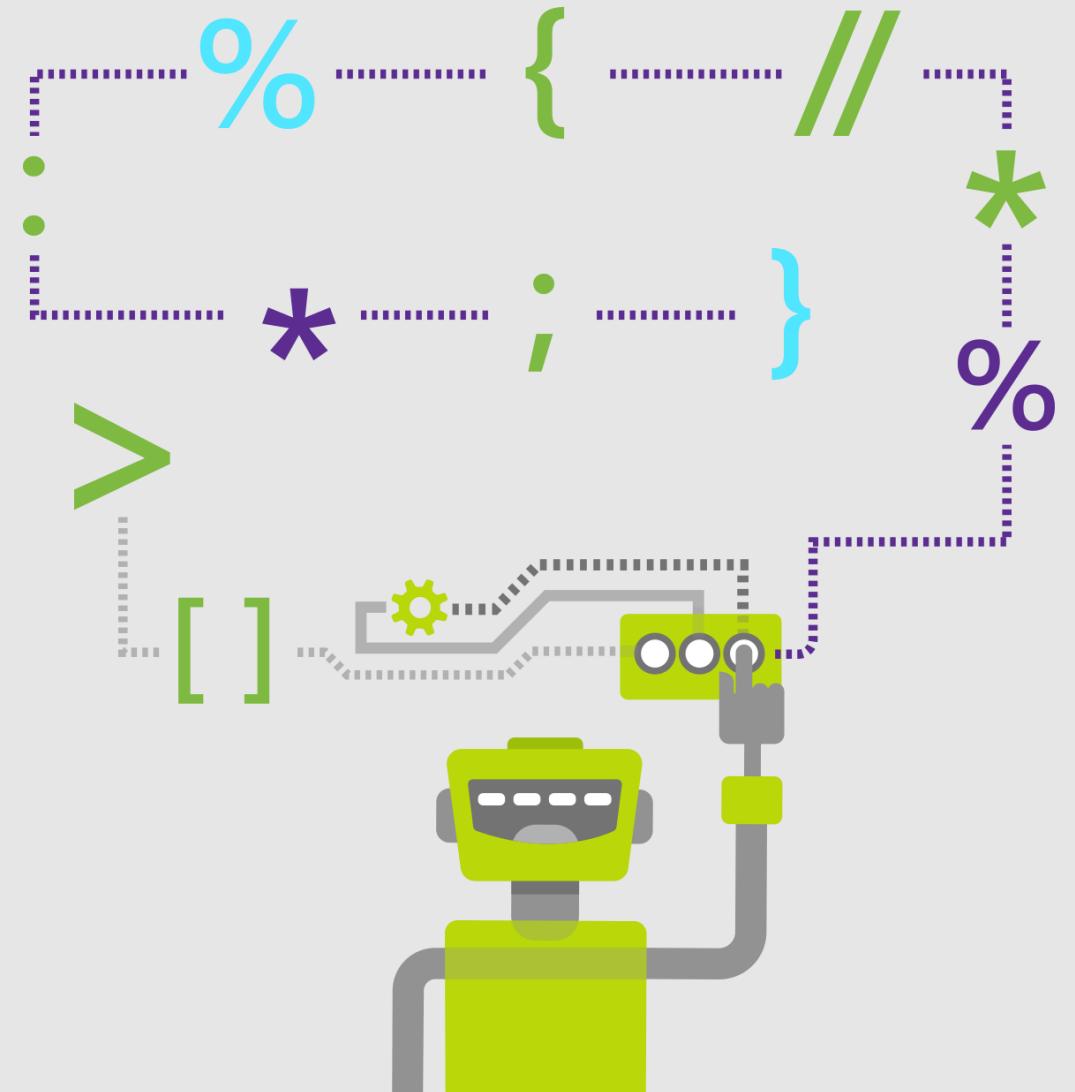
Custom models

- Get recommendations from your *private* codebases
- Distribute models to your team to enhance everyone's productivity
- Available for C#

Code style Inference (preview)

- Create EditorConfig settings from code style convention from *your* codebase
- Formatting/style only
- Available for C# only

.NET Refactoring and Code Fixes



.NET Refactorings and Code Fixes

Foreach to LINQ (and back!)

```
--  
36  | foreach (var image in bingResult.images.value)  
37 | {  
38 | | var newItem = new SearchResultItem  
39 | | {  
40 | | | Id = string.Empty,  
41 | | | Kind = SearchResultItemKind.Image,  
42 | | | Title = image.name,  
43 | | | ThumbnailUrl = image.thumbnailUrl,  
44 | | | Url = image.contentUrl,  
45 | | | Description = string.Empty,  
46 | | | Summary = string.Empty,  
47 | | };  
48 | | mappedResults.Add(newItem);  
49 }
```

.NET Refactorings and Code Fixes

Foreach to LINQ (and back!)

```
38     mappedResults.AddRange(from image in bingResult.images.value
39                             let newItem = new SearchResultItem
40                             {
41                                 Id = string.Empty,
42                                 Kind = SearchResultItemKind.Image,
43                                 Title = image.name,
44                                 ThumbnailUrl = image.thumbnailUrl,
45                                 Url = image.contentUrl,
46                                 Description = string.Empty,
47                                 Summary = string.Empty,
48                             }
49                             select newItem);
50 }
```

.NET Refactorings and Code Fixes

Invert conditional expressions

```
66         result.Language = language;
67         if (!string.IsNullOrWhiteSpace(language))
68     {
69         result.Entities = await GetEntitiesAsync(text, language).ConfigureAwait(false);
70         result.AnnotatedHTML = await GetAnnotatedHtmlAsync(text, language, EntityTypes).ConfigureAwait(false);
71     }
72
73     return result;
74 }
```

.NET Refactorings and Code Fixes

Invert conditional expressions

```
66         result.Language = language;
67         if (string.IsNullOrWhiteSpace(language))
68         {
69             return result;
70         }
71         result.Entities = await GetEntitiesAsync(text, language).ConfigureAwait(false);
72         result.AnnotatedHTML = await GetAnnotatedHtmlAsync(text, language, EntityTypes).ConfigureAwait(false);
73
74         return result;
75     }
```

.NET Refactorings and Code Fixes

Split/Merge IF statements

The screenshot shows a code editor with the following code:

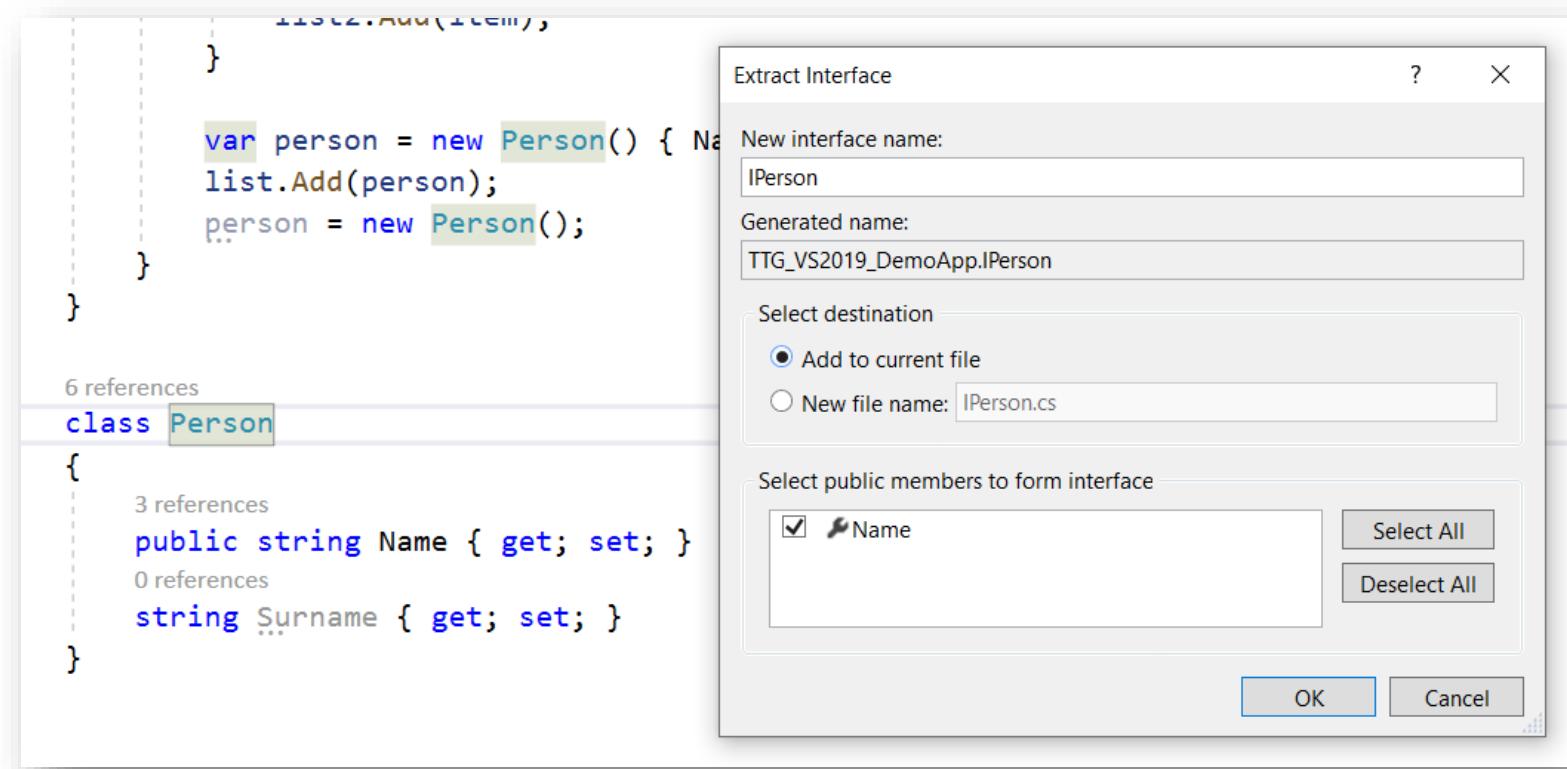
```
96     if (tweets.statuses != null && tweets.statuses.Count > 0)
97
98     Replace '&&' with '||'
99     Split into nested 'if' statements ▶
100    Introduce local
101
102    Invert if
103    Wrap expression
104
105    }
106
107    return Json(data)
```

A context menu is open at line 99, with "Split into nested 'if' statements" selected. The code has been partially refactored:

```
if (tweets.statuses != null && tweets.statuses.Count > 0)
if (tweets.statuses != null)
{
    if (tweets.statuses.Count > 0)
    {
        foreach (var status in tweets.statuses)
        {
            data.documents.Add(new TwitterDocument(id.ToString(), status.text));
            results.Add(id.ToString(), new Result() { id = id.ToString(), text = status.text, userName = status.user.name });
            id++;
            Console.WriteLine("Id: {0} , tweet: {1}", id, status.text);
        }
    }
}
```

.NET Refactorings and Code Fixes

Extract Interface to same file



.NET Refactorings and Code Fixes

Pull members up

The screenshot shows a code editor with the following code:

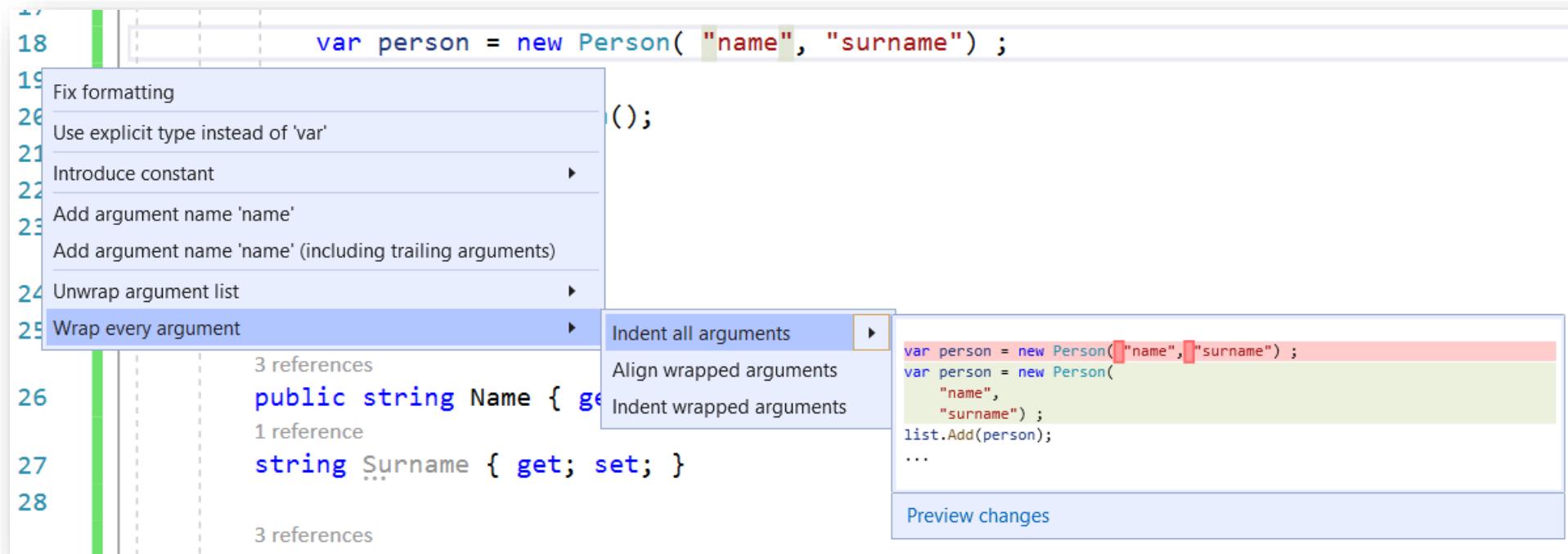
```
24      interface IPerson
25  {
26      string Name { get; set; }
27  }
28
29  class Person : IPerson
30  {
31      public string Name { get; set; }
32      string Surname { get; set; }
33  }
```

A context menu is open at line 32, item 32, over the member `Surname`. The menu items are:

- Remove unused member
- Replace 'Surname' with methods
- Convert to full property
- Pull 'Surname' up ▶** (highlighted)
- Pull members up to base type...
- Add accessibility modifiers
- Suppress IDE0051 ▶

.NET Refactorings and Code Fixes

Wrap/indent/align parameters/arguments



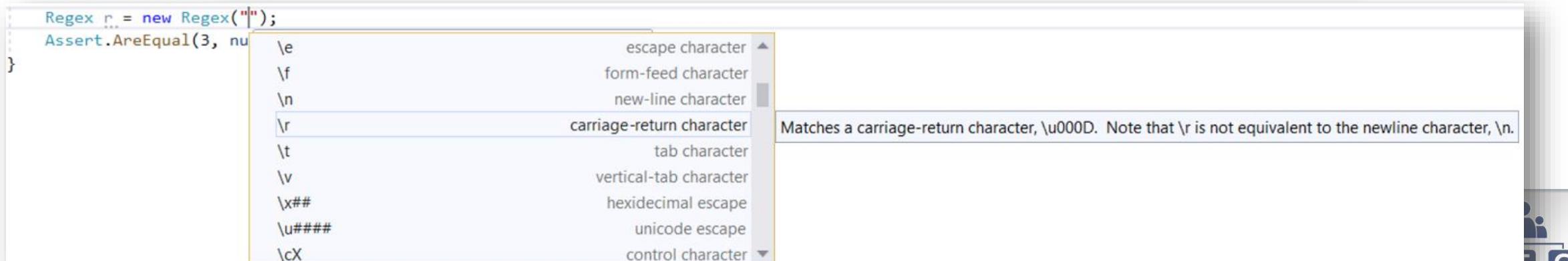
.NET Refactorings and Code Fixes

Regex language support

```
Regex emailRegex = new Regex(@"^([\w\.\-]+)@([\w\-]+)((\.(\\w){2,3})+)$");

// language=regex,ExplicitCapture
var regexString = @"^([\w\.\-]+)@([\w\-]+)((\.(\\w){2,3})+)$";

var email = "gianni@mail.com";
var valid = emailRegex.IsMatch(email) ? "valid" : "not valid";
Console.WriteLine($"{email} is {valid} according to '{regexString}'");
```



.NET Refactorings and Code Fixes

Convert anonymous type to tuple or class

The screenshot shows a code editor with the following code:

```
32 var anonymousClass = new { str = "Hello" };
33
34 Use discard '_'
35 Convert to class ▶
36 Introduce local ▶
37 Suppress IDE0059 ▶
38
39
40
41
42
43
44
45
46
47
48
49
50 }
```

A context menu is open at line 35, with the "Convert to class" option highlighted. A preview window shows the resulting code:

```
var anonymousClass = new { str = "Hello" };
var anonymousClass = new NewClass("Hello");
...
internal class NewClass
{
    public string Str { get; }

    public NewClass(string str)
    {
        Str = str;
    }

    public override bool Equals(object obj)
    {
        return obj is NewClass other &&
               Str == other.Str;
    }

    public override int GetHashCode()
    {
        return HashCode.Combine(Str);
    }
}
```

At the bottom of the preview window, there is a "Preview changes" button.

.NET Refactorings and Code Fixes

- Sync namespace and folder name
- Add missing reference for unimported types
- (Experimental) IntelliSense for *unimported* types
- Move type to namespace or folder
- Use Expression/block body for lambda
- Remove unused expression values and parameters
- Toggle Block Comment/Uncomment
- Unsealing a sealed class

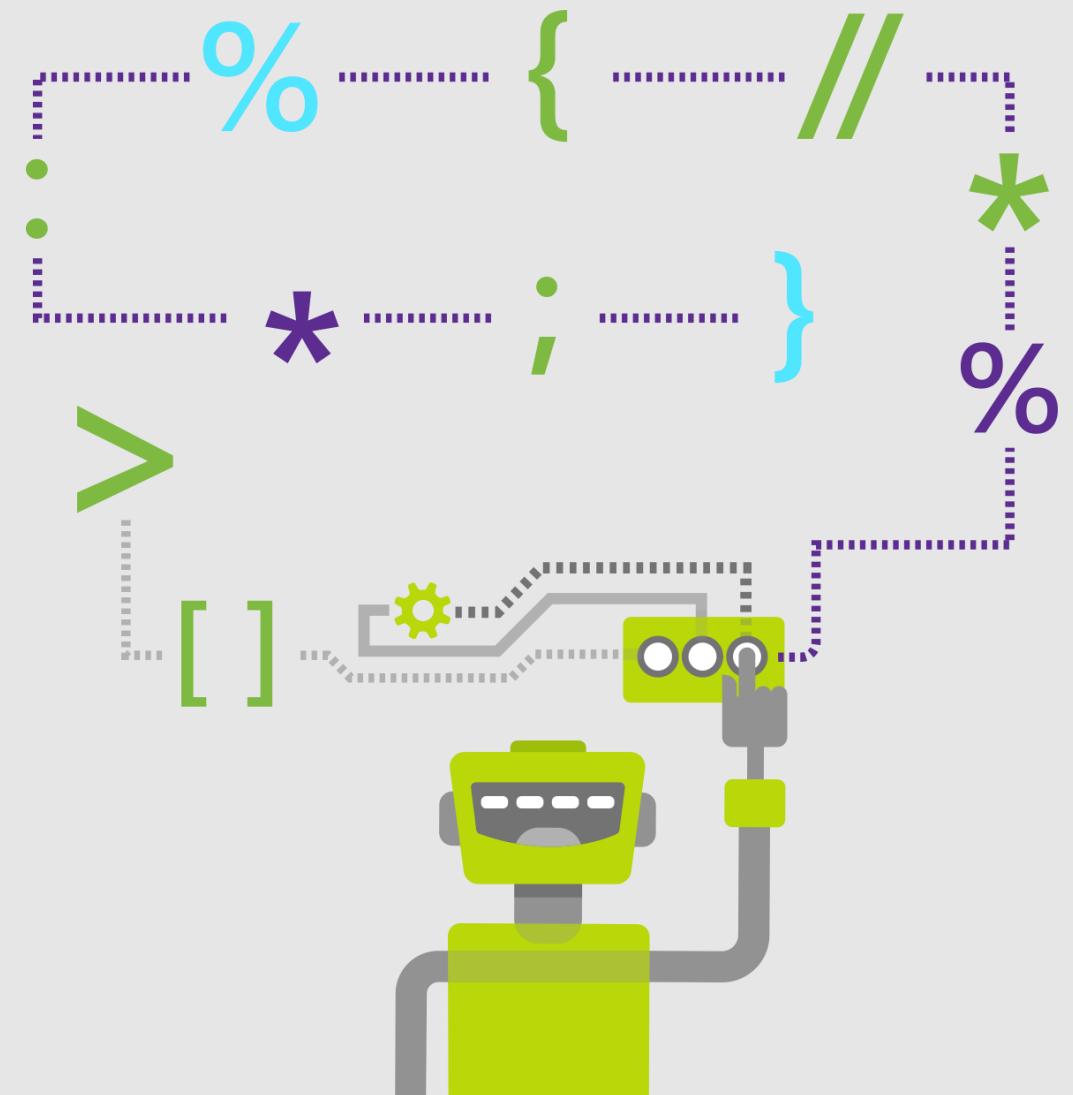
<https://docs.microsoft.com/en-us/visualstudio/releases/2019/release-notes#net-productivity>

<https://docs.microsoft.com/en-us/visualstudio/releases/2019/release-notes-v16.0#net-productivity>



LiveShare

Real-Time Collaboration



LiveShare – Real-Time Collaboration



Work from your own tools

- No need to clone repos or set up environments
- Pair program from across the team room, from home or on the other side of the world
- Visual Studio 2019 / Visual Studio Code
- C#, C++, Python

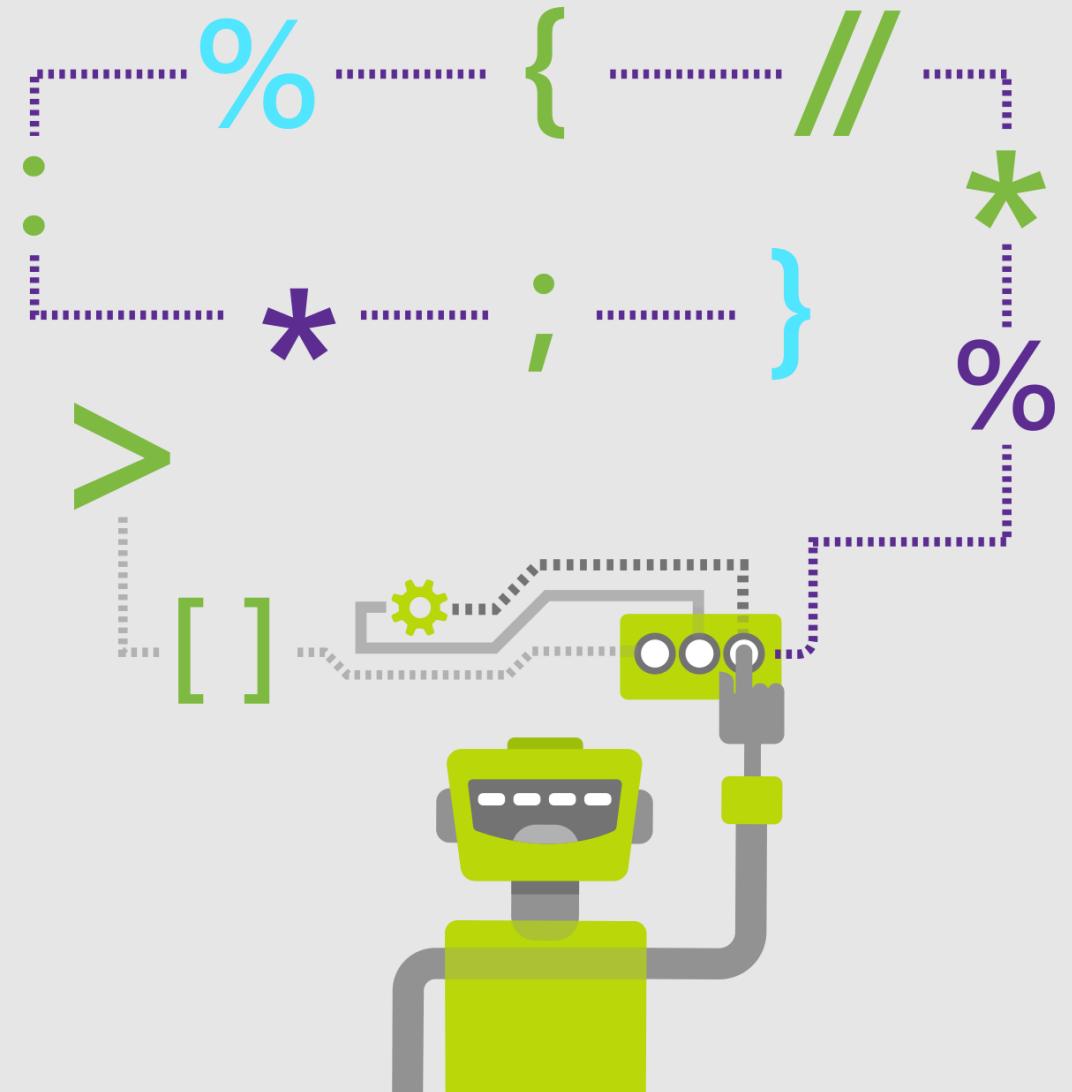
Collaborative development

- Co-edit and debug with your team mates
- Language service features, localhost, terminals and more forwarded to the guest

<https://visualstudio.microsoft.com/services/live-share/>

<https://devblogs.microsoft.com/visualstudio/live-share-now-included-with-visual-studio-2019/>

Python Development



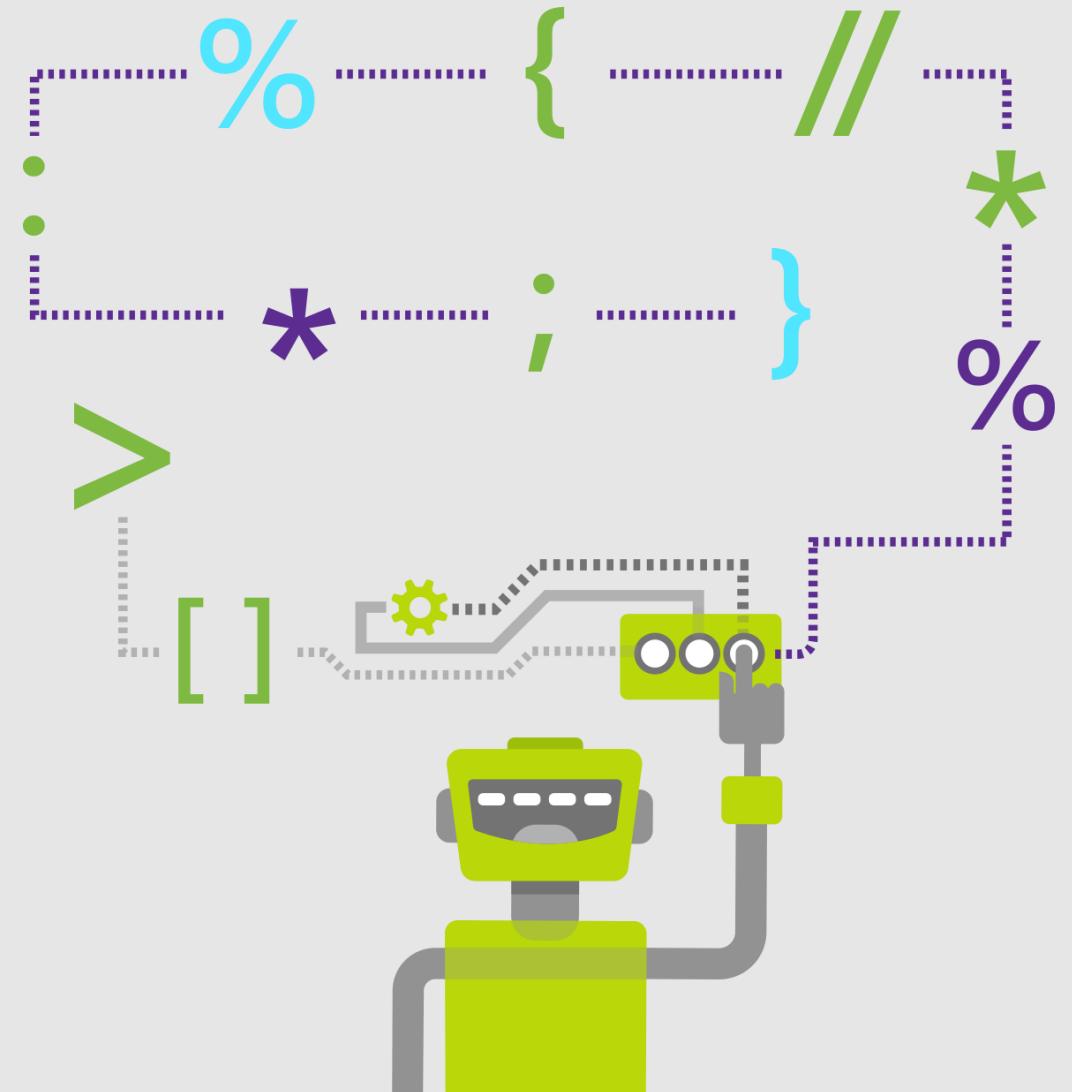
What's New in Visual Studio 2019?

- Python Toolbar
- Open Folder Support
- Add Environment dialog
- LiveShare support (VS and VSCode)
- Embedded Python in other projects (C#, C++)
- New/Updated Project Templates

<https://docs.microsoft.com/en-us/visualstudio/releases/2019/release-notes-v16.0#Python>
<https://devblogs.microsoft.com/python/python-in-visual-studio-2019-preview-2/>



C++ Development



One IDE for any developer, any app, any platform

- Freedom to target any platform from one single IDE
Azure Sphere, ARM microcontrollers, Android, Cygwin/MinGW, iOS, Linux, UWP, Windows
- Easy to get started
Keep your CMake/make/Ninja, No conversion to VS solutions, Easy C++ library acquisition
- Continue to use your C++ tools of choice, all integrated in the IDE
Choice of LLVM/GCC/MSVC, Clang-format, CTest, Google Test, Boost.Test
- Rich and familiar C++ code editing and debugging experiences
IntelliSense, Refactoring, Conditional breakpoints, Debug visualization, Just-my-code debugging



Pain-free upgrades

1. Use any MSVC toolset inside the Visual Studio 2019 IDE
2. Maintain binary compatibility with 3rd party binaries when you upgrade to the latest MSVC toolset
3. Access the full collection of OSS libraries available in Vcpkg

```
c:\src> git clone https://github.com/Microsoft/vcpkg.git & cd vcpkg  
c:\src\vcpkg> .\bootstrap-vcpkg.bat  
c:\src\vcpkg> .\vcpkg install curl
```

Conformance

- C++98 complete under /permissive-
- C++11 complete
- C++14 complete
- C++17 under /std:c++17
 - Compiler feature complete
 - Most complete standard library implementation
- C++20 partial support under /std:c++latest
 - Coroutines (partial support) under /await
 - Modules (partial support) under /experimental:module
 - `<=>` three-way comparison operator (partial support)
 - Feature-test macros
 - `remove_cvref`
 - Prevent aggregate-init with user-declared or deleted constructors



New Code Analysis checks

Concurrency checks

Use-after-move check off-by-default, part of C++ Core Check Experimental Rules

Coroutine checks

Lifetime profile checks off-by-default, part of C++ Core Check Lifetime Rules

Concurrency check example

```
bool HandleRequest(int Id, Request* request)
{
    EnterCriticalSection(&cs_);
    if (cache_.find(Id) != cache_.end())
        return false; // C26115: Failing to release lock 'this->cs_' in function
    cache_[Id] = request;
    LeaveCriticalSection(&cs_);
}
```

Learn more at <https://aka.ms/cpp/code-analysis>



Productivity in Visual Studio 2019

Browse & Edit

- Duplicate code with Ctrl + D
- Add macro to hint file (quick fix)
- ClangFormat support
- Expanded macros in Quick Info
- Macro to constexpr (quick fix)
- IntelliSense in C++ templates
- Live Share for C++
- IntelliCode for C++
- Improved Visual Studio search
- Document Health indicator
- Context menu for clipboard ring
- Add missing #include (quick fix)
- Add missing namespace (quick fix)
- * to & / & to * (quick fix)
- Go to Document on F12
- NULL to nullptr (quick fix)

Commit

- Improved branch switch performance
- Git stash support

Compile

- C++17 feature complete (except charconv)
- Improved compile times
- C++ 20 partial support

Diagnose

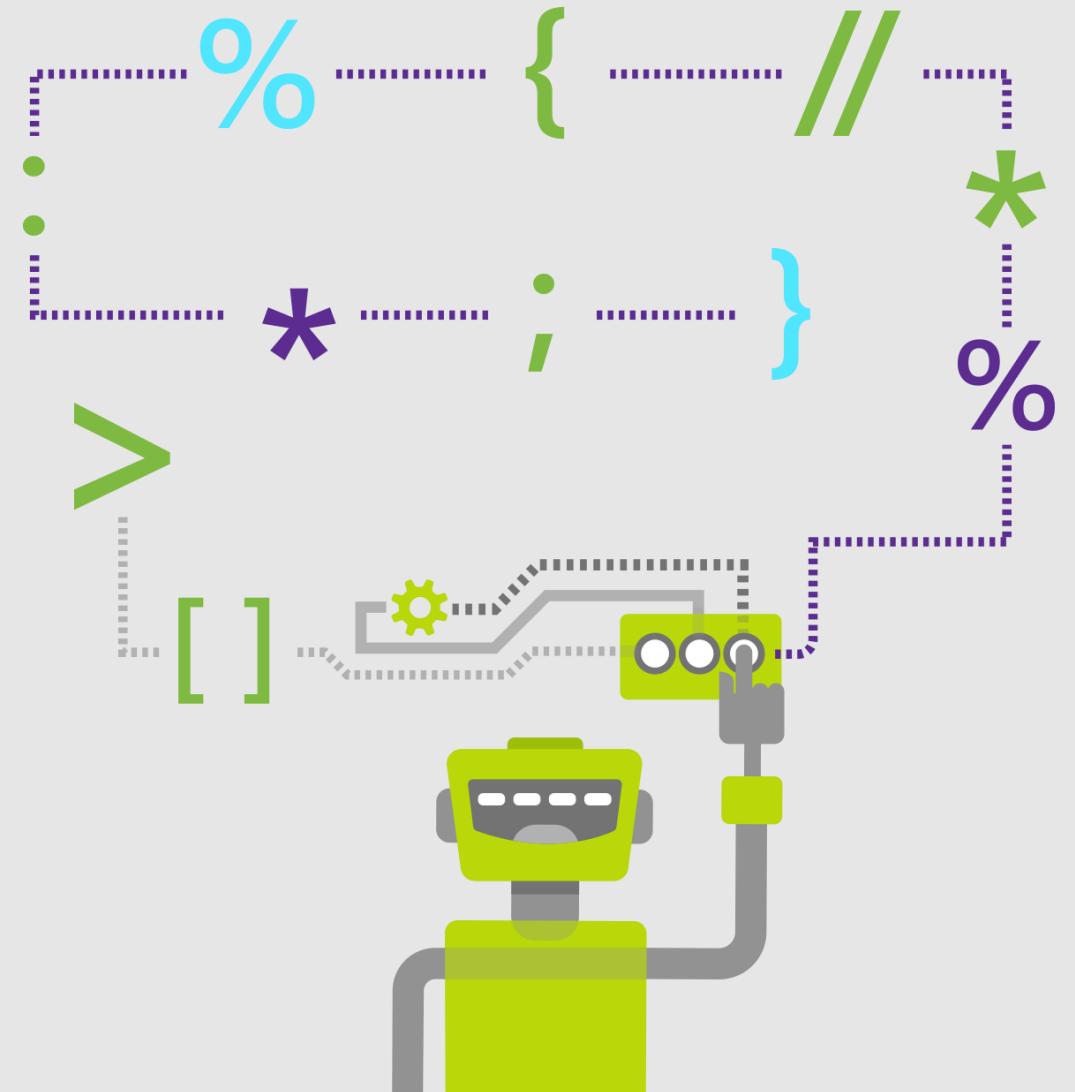
- Single file code analysis
- Just My Code debugging
- Step Back debugging
- Data breakpoints support in debugging tool windows
- 64-bit out-of-process debugger process
- Code analysis running in background
- New C++ Core Checker rules
- Lifetime and Concurrency Check rules

Test

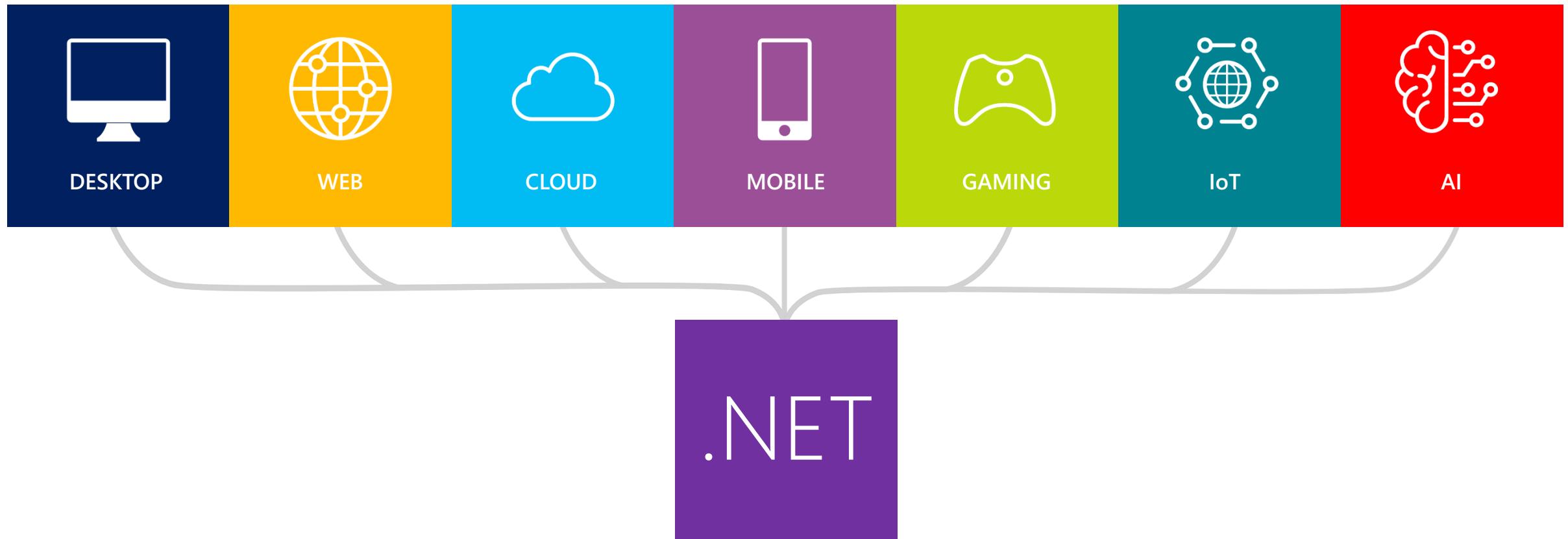
- CTest, Google Test, Boost.Test support in CMake projects
- Boost.Test Add > New Item template
- Test Explorer hierarchical view
- CodeLens support for unit testing



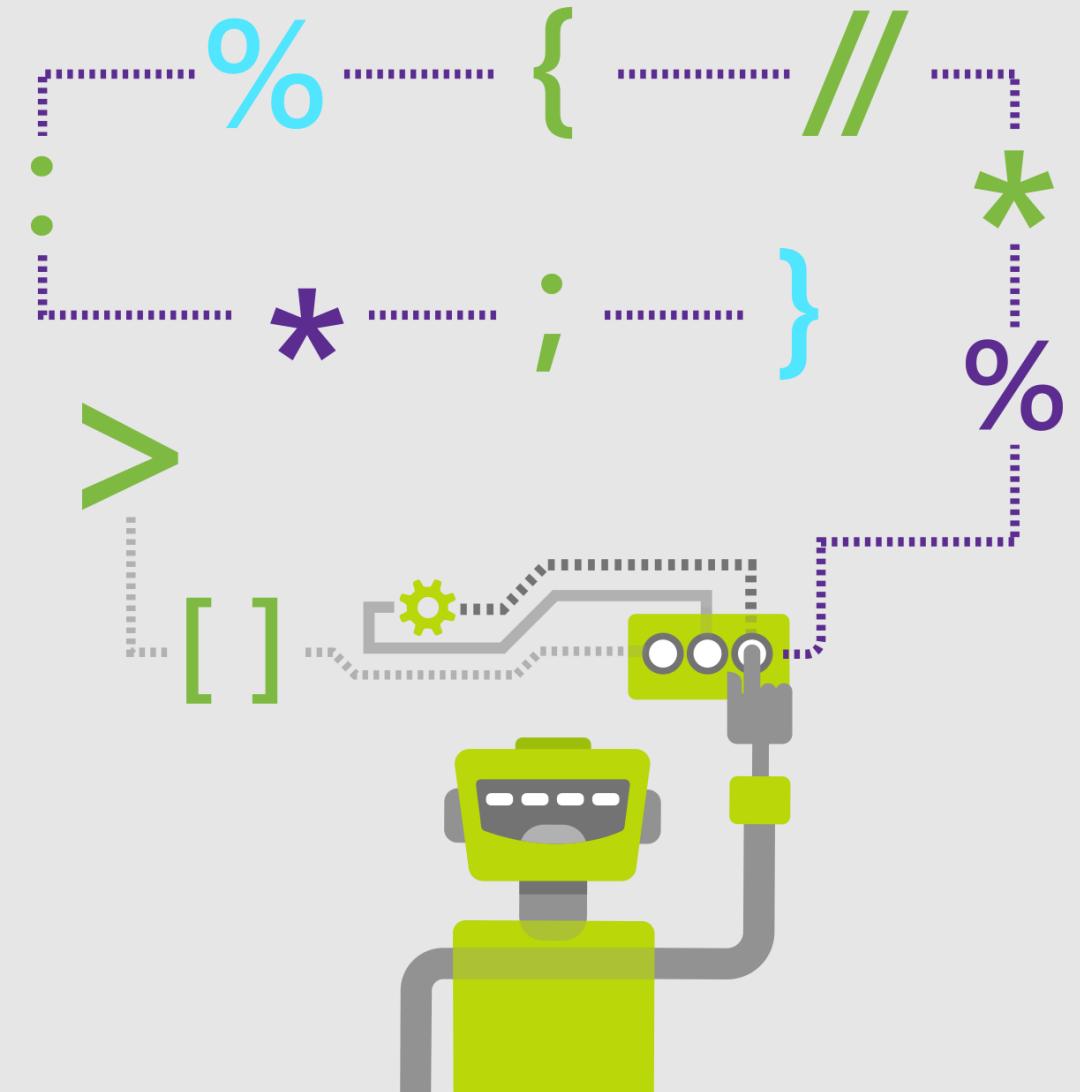
What's new in .NET ecosystem



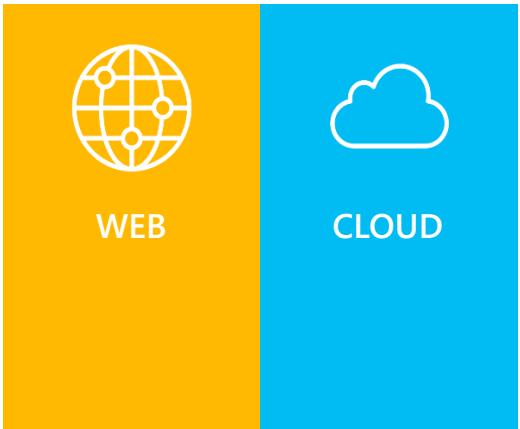
Your platform for building anything



.NET Core the Future of .NET



.NET Core 2



.NET Core is perfectly suited for the requirements
of cloud-native, cross-platform workloads

.NET CORE

LIBRARIES

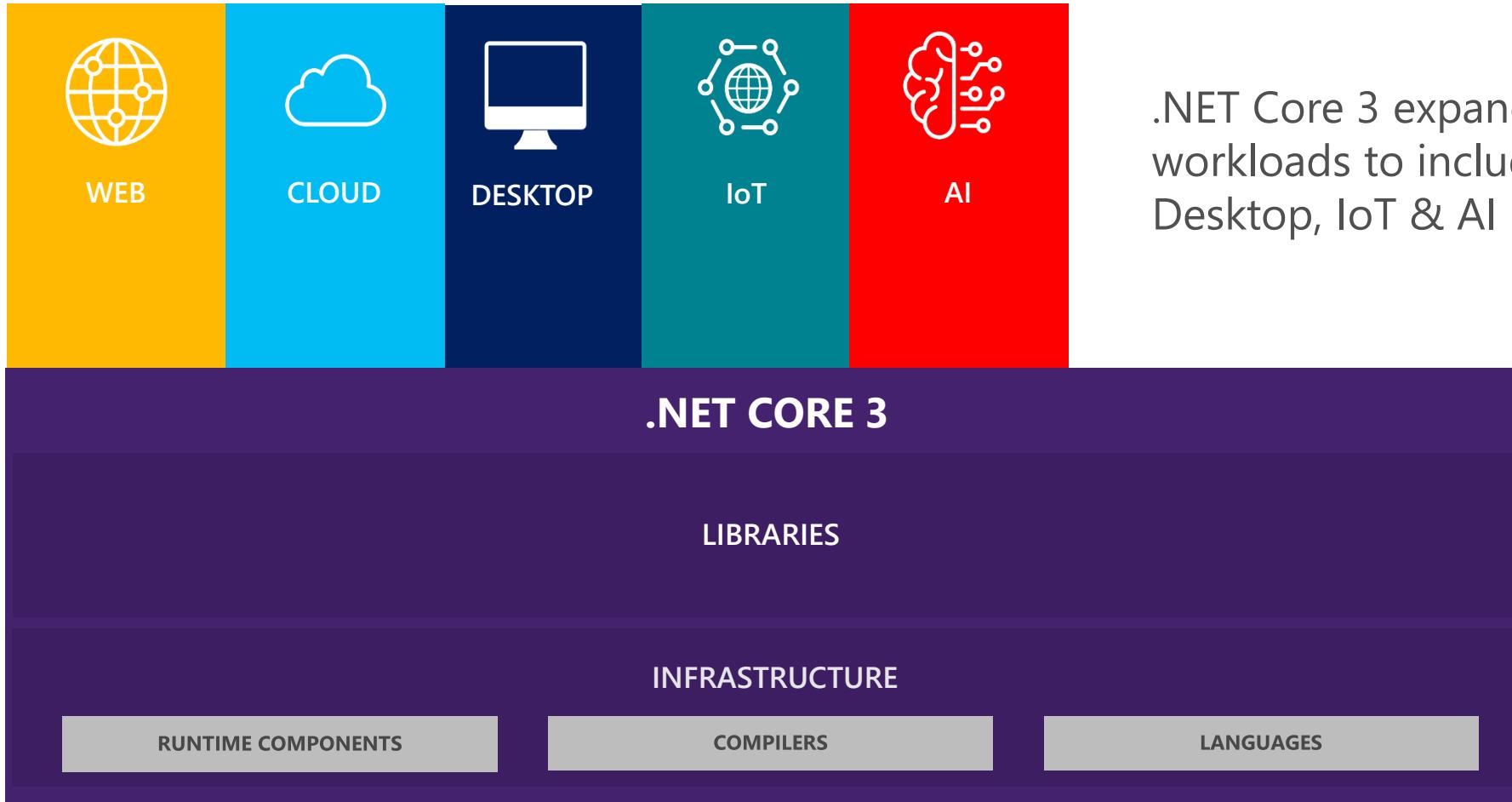
INFRASTRUCTURE

RUNTIME COMPONENTS

COMPILERS

LANGUAGES

.NET Core 3



.NET Core 3.0

- Release planned for September 2019
- Implements .NET Standard 2.1
- No more features from .NET Framework
 - Feature complete to support WPF, WinForm, Entity Framework
 - More than 50k APIs
 - If needed, use alternative open-source/community projects for .NET Core:
ex. Windows Workflow <https://github.com/UiPath/corewf>
- All future investments in .NET will be in .NET Core
 - Runtime, JIT, AOT, GC, BCL (Base Class Library), C#, VB.NET, F#, ASP.NET, Entity Framework, ML.NET, WinForms, WPF and Xamarin

<https://docs.microsoft.com/en-us/dotnet/core/whats-new/dotnet-core-3-0>
<https://devblogs.microsoft.com/dotnet/net-core-is-the-future-of-net/>



C# 8.0

Ships with .NET Core 3.0

Nullable reference types

References must be declared nullable, to be null-assigned

```
string? s = null;
```

```
<LangVersion>8.0</LangVersion>
<NullableContextOptions>enable</NullableContextOptions>
```

Async streams

Introduce **IAsyncEnumerable<T>**, an asynchronous version of **IEnumerable<T>**, with await foreach and yield return support
<https://www.infoq.com/articles/Async-Streams>

C# 8.0

Ranges, Indices and Recursive Patterns

New index '^' and range '..' operators, similar to Python behavior

<https://www.infoq.com/articles/cs8-ranges-and-recursive-patterns>

Default implementations of Interface members

<https://www.infoq.com/articles/default-interface-methods-cs8>

C# 8.0

Ranges, Indices and Recursive Patterns

New index '^' and range '..' operators, similar to Python behavior

<https://www.infoq.com/articles/cs8-ranges-and-recursive-patterns>

Def

```
interface IDefaultInterfaceMethod
{
    public void DefaultMethod()
    {
        Console.WriteLine("I am a default method in the interface!");
    }
}

class AnyClass : IDefaultInterfaceMethod
{
}
```

<https://docs.microsoft.com/en-us/dotnet/csharp/whats-new/csharp-8>



.NET Framework 4.8

- Included in the Windows 10 May 2019 Update
- Last major version of .NET Framework
- Continued support/tooling/bug fix, NO new features
- NO .NET Standard 2.1

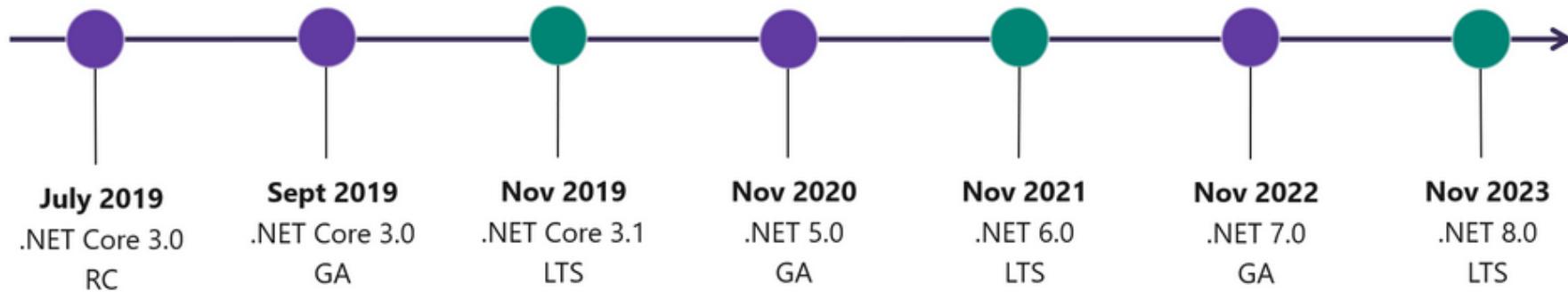
<https://docs.microsoft.com/en-us/dotnet/framework/whats-new/>
<https://devblogs.microsoft.com/dotnet/announcing-the-net-framework-4-8/>



.NET 5

Next release after .NET Core 3.0

One .NET going forward, and you will be able to use it to target Windows, Linux, macOS, iOS, Android, tvOS, watchOS, WebAssembly and more...



.NET 5



.NET 5

Will continue to exist:

- Open source and community-oriented on GitHub
- Cross-platform implementation
- Support for leveraging platform-specific capabilities, such as Windows Forms and WPF on Windows and the native bindings to each native platform from Xamarin.
- High performance
- Side-by-side installation
- Small project files (SDK-style)
- Capable command-line interface (CLI)
- Visual Studio, Visual Studio for Mac and Visual Studio Code integration.

.NET 5

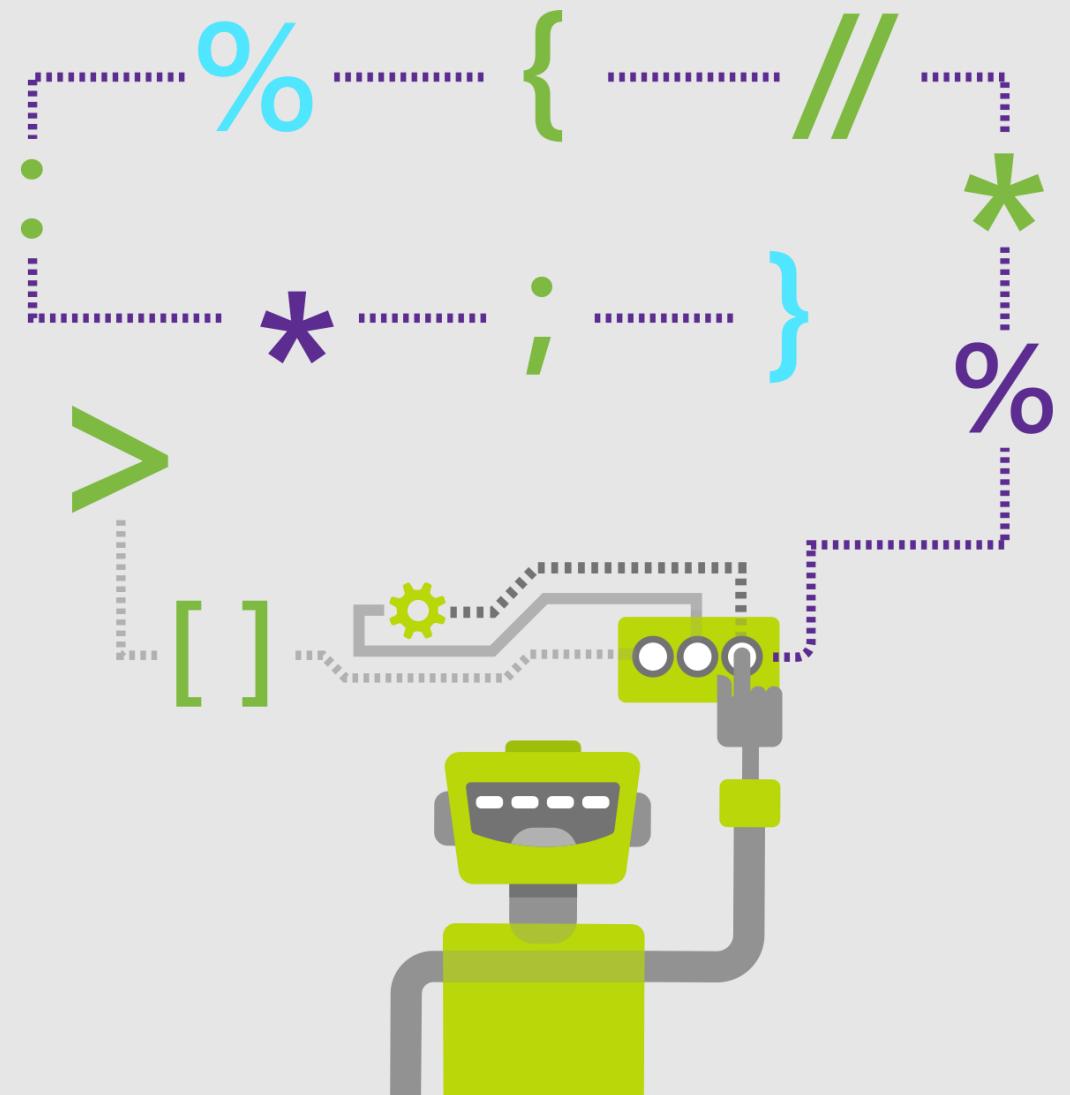
Will be new:

- More choice on runtime experiences (Mono, CoreCLR)
- Java interoperability will be available on all platforms
- Objective-C and Swift interoperability will be supported on multiple operating systems
- CoreFX will be extended to support static compilation of .NET (ahead-of-time – AOT), smaller footprints and support for more operating systems.

<https://devblogs.microsoft.com/dotnet/introducing-net-5/>



ML.NET



ML.NET

Released ML.NET 1.0

<https://devblogs.microsoft.com/dotnet/announcing-ml-net-1-0/>

<https://github.com/dotnet/machinelearning-samples>

Support for many ML scenarios

Binary/Multiclass Classification

Clustering

Regression

Ranking

Anomaly Detection

Clustering

Recommendation

Deep Learning (e.g. TensorFlow, ONNX)

Helpful data readers, learners and transforms

Learners: FFM, LightGBM, Ensemble, SymSGD, ...

Transforms: LightLDA, Word Embeddings, ...

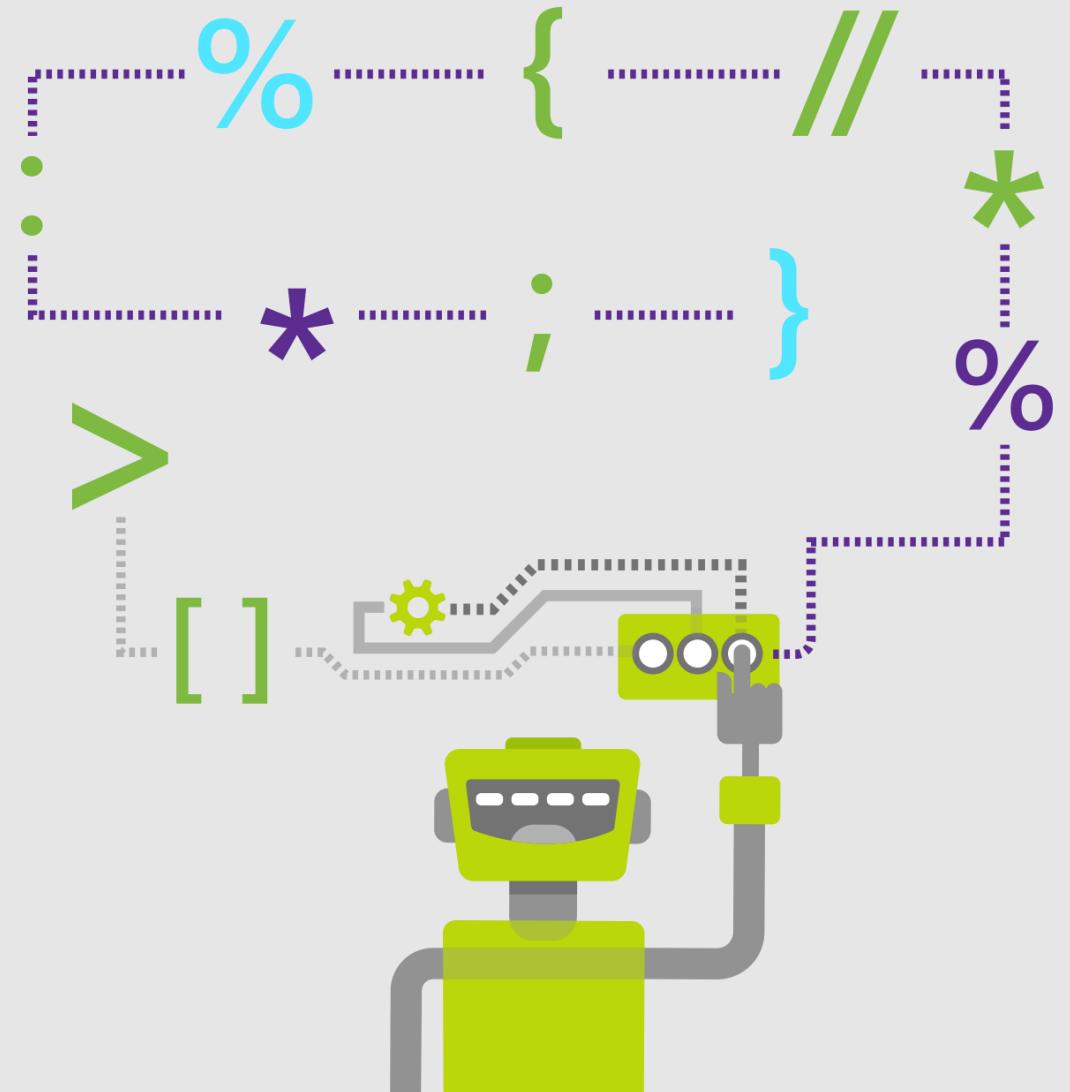
Data Pipeline customization

AutoML, ML.NET CLI, ML.NET Model Builder (in Preview)



Blazor

Client-side web apps with .NET Core 3.0



Web apps with .NET Core 3.0

Build client-side web UI with .NET

Reusable web UI components with C# and Razor

Share .NET code with client and server

Call into JavaScript libraries & browser APIs as needed

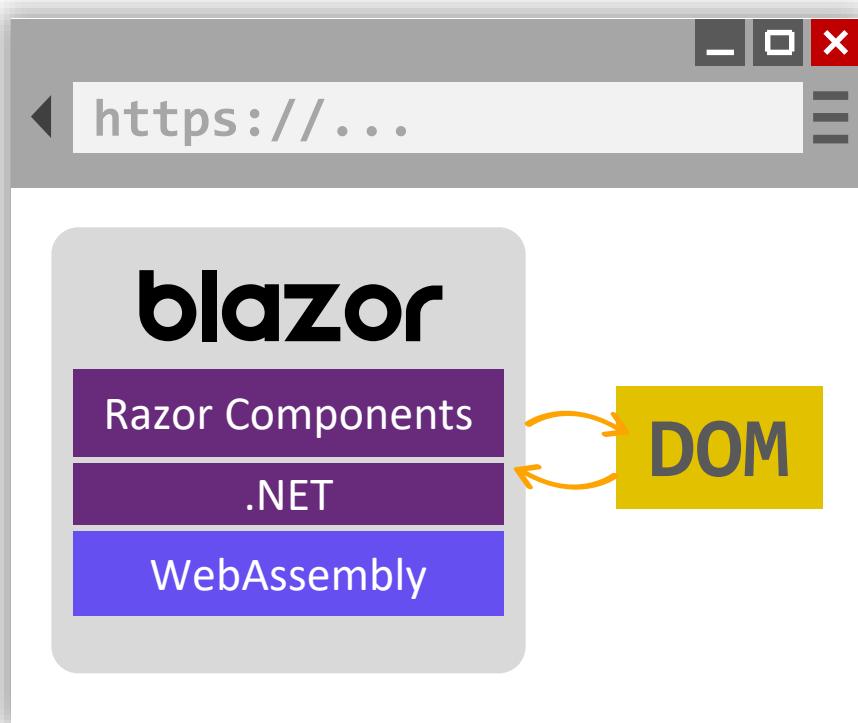
Handle client UI interactions on the server over SignalR or directly in the browser via WebAssembly (future)



<https://blazor.net>

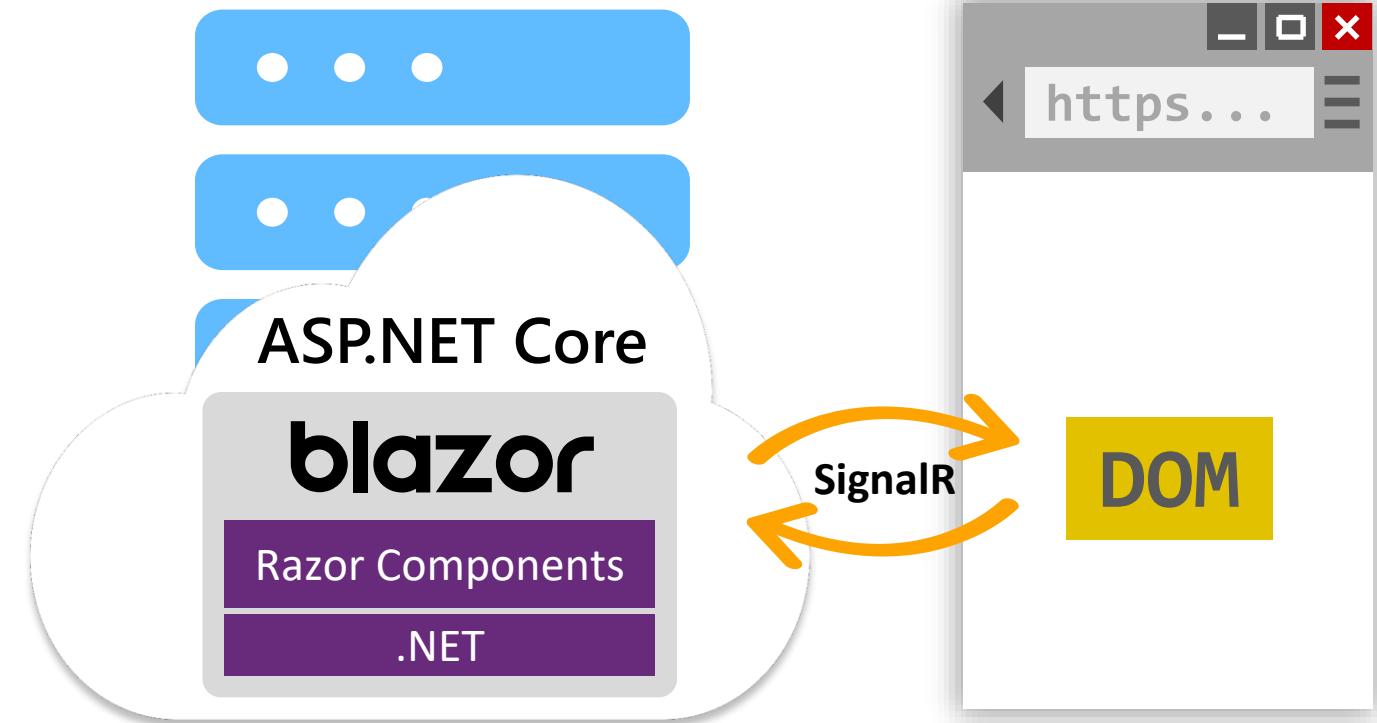
Blazor on client or server

Client-side



Future

Server-side



.NET Core 3.0

Reference Links

<https://docs.microsoft.com/en-us/visualstudio/ide/whats-new-visual-studio-2019?view=vs-2019>

<https://docs.microsoft.com/en-us/visualstudio/releases/2019/release-notes>

<https://devblogs.microsoft.com/visualstudio/>

<https://devblogs.microsoft.com/dotnet/>

<https://devblogs.microsoft.com/xamarin/>

<https://devblogs.microsoft.com/python/>

<https://devblogs.microsoft.com/cppblog/>

<https://devblogs.microsoft.com/devops/>

Thank You

ευχαριστώ Salamat Po متشرّم شكرًا Grazie

благодаря ありがとうございます Kiitos Teşekkürler 谢谢

ឧបម្ពុណមរំបែក Obrigado شكريه Terima Kasih Dziękuję

Hvala Köszönöm Tak Dank u wel дякую Tack

Mulțumesc спасибо Danke Cám ơn Gracias

多謝晒 Ďakujem הַדּוֹת දේකුඩ් Děkuji 감사합니다

Q&A

About



Microsoft Specialist
Programming in C#
Programming in HTML5
with JavaScript & CSS3

Microsoft CERTIFIED
Solutions Developer
Windows Store Apps Using C#
Web Applications



Ing. Gianni ROSA GALLINA

gianni@rosagallina.com

R&D Specialist, Senior Software Engineer @ Deltatre



- AI, Machine Learning (Google/Microsoft ML APIs, PyTorch, fastai)
- Virtual Reality (Oculus Rift, Gear VR, WinMR, Unity 3D)
- Augmented/Mixed Reality (HoloLens, Magic Leap)
- Immersive video streaming and 3D graphics for sport events
- NUI Prototypes (Microsoft Kinect, Leap Motion)
- Mobile App developer (Windows / Android / Xamarin)
- Cloud solutions with Microsoft Azure (serverless, video workflow)



PLURALSIGHT Author

<http://gianni.rosagallina.com>





**Attribution-NonCommercial-ShareAlike
4.0 International (CC BY-NC-SA 4.0)**