

What's New in Visual Studio 2019

Mark Michaelis
CEO, IntelliTect



Meet Mark

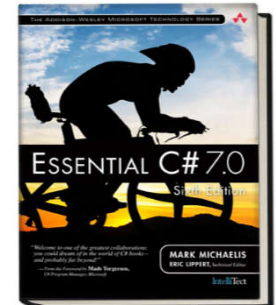


Mark@IntelliTect.com

fb.com/Mark.Michaelis

Twitter: @MarkMichaelis

@IntelliTect, fb.com/IntelliTect



Refactoring Galore

The image shows a sequence of steps in Visual Studio for refactoring a code snippet. The code is:

```
string fileName = args[0]?.Remove(0, 5);
```

Step 1: A lightbulb icon indicates a suggestion: "(local variable) string fileName" with the message "Unnecessary assignment of a value to 'fileName'".

Step 2: The IDE0059 error is shown: "Unnecessary assignment of a value to 'fileName'". The code is highlighted, and the refactoring menu is open, showing options like "Use discard '_'".

Step 3: The "Preview Changes" dialog is shown, displaying the refactored code:

```
if (... args[0]?.ToLower().StartsWith("file:") ?? false)
```

Step 4: The "Preview Code Changes" dialog is shown, displaying the original code with the refactored code overlaid:

```
11 + "Use:\n\tfind.exe file:<file  
12 }  
13 else  
14 {  
15     if (  
16         args[0]?.ToLower().StartsWith(  
17         args[0]?.ToLower().StartsWith(  
18     )  
19 }
```

The "Preview Code Changes" dialog also shows "No issues found" and "Ln: 27 Ch: 1 MIXED".

Apply Styles/Format issues in Bulk

The screenshot shows a code editor with the following C# code snippet:

```
string fileName = args[0]?Remove(0, 5);
```

The code has a red squiggly line under the `args[0]?Remove(0, 5);` line, indicating a formatting issue. A context menu is open over the code, showing the following options:

- Fix formatting
- Generate method 'Program.Remove'
- Wrap every argument
- Unwrap and indent all arguments
- Configure or Suppress issues

The 'Fix formatting' option is selected, and a sub-menu is open showing the following options:

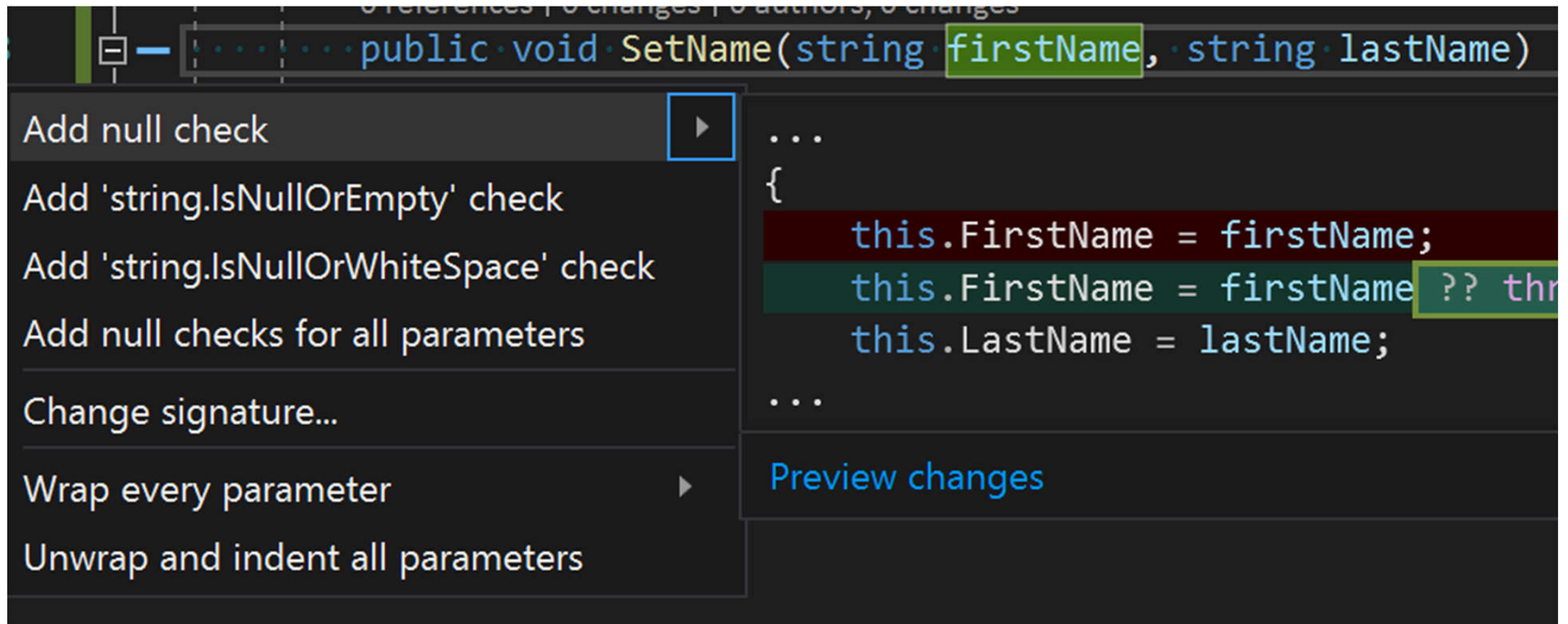
- IDE0055 Fix formatting

The sub-menu also shows a preview of the code after formatting:

```
...  
args[0]?.ToLower().StartsWith("file:")??false)  
{  
    string fileName = args[0]?Remove(0, 5);  
    string fileName = args[0]?Remove(0, 5);  
    // ...  
...  
Preview changes  
Fix all occurrences in: Document | Project | Solution
```

The 'Fix all occurrences in: Document | Project | Solution' option is circled in red.

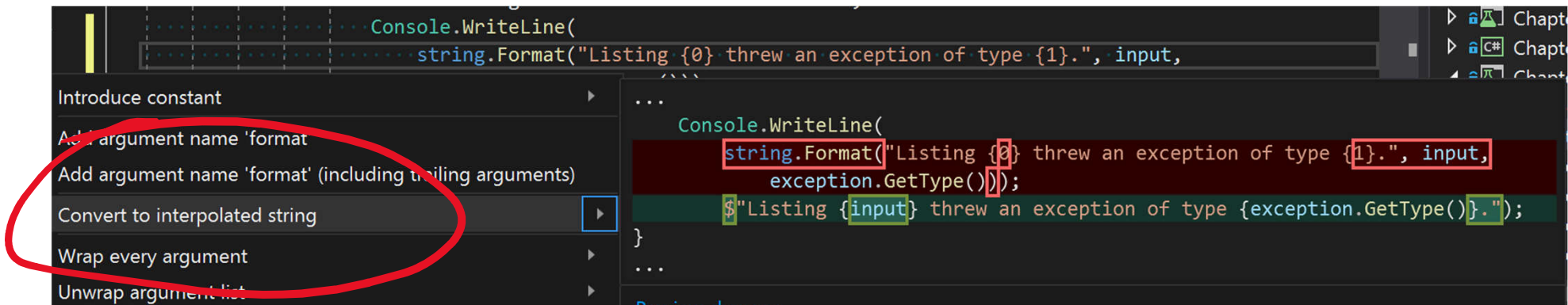
Add Parameter Null Checking



The screenshot shows an IDE interface with a context menu open over a code snippet. The code snippet is a method signature: `public void SetName(string firstName, string lastName)`. The `firstName` parameter is highlighted in green. The context menu is open, showing several options: "Add null check", "Add 'string.IsNullOrEmpty' check", "Add 'string.IsNullOrWhiteSpace' check", "Add null checks for all parameters", "Change signature...", "Wrap every parameter", and "Unwrap and indent all parameters". The "Add null check" option is selected, and a blue box highlights the right-pointing arrow next to it. To the right of the menu, the method body is visible, showing the assignment `this.FirstName = firstName;` on a line that is highlighted in green. The `firstName` variable in this line is also highlighted in green. Below the code, there is a "Preview changes" button.

```
public void SetName(string firstName, string lastName)
{
    this.FirstName = firstName;
    this.FirstName = firstName ?? throw new ArgumentNullException(nameof(firstName));
    this.LastName = lastName;
}
```

Convert to Interpolated String

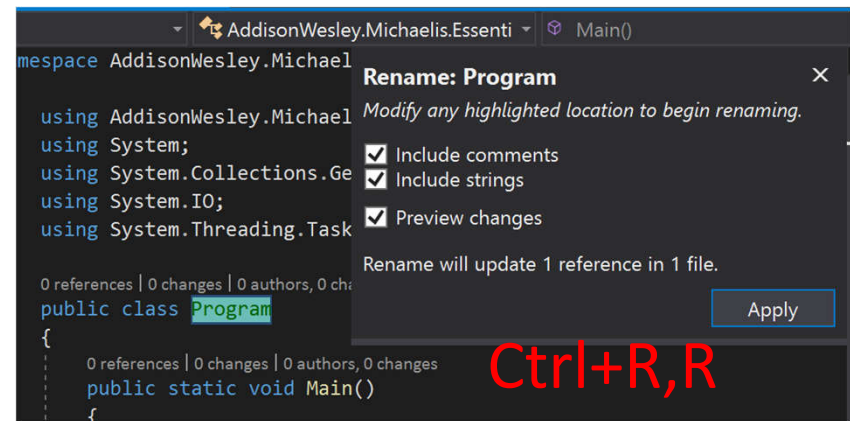


The screenshot shows a Visual Studio code editor with a context menu open over a C# code snippet. The menu item "Convert to interpolated string" is highlighted with a red circle. The code in the background is as follows:

```
... Console.WriteLine(  
    string.Format("Listing {0} threw an exception of type {1}.", input,  
    ...  
    ...  
    Console.WriteLine(  
        string.Format("Listing {0} threw an exception of type {1}.", input,  
            exception.GetType());  
        $"Listing {input} threw an exception of type {exception.GetType()}."  
    }  
    ...  
    ...
```

csproj Level Renames

- Project Rename Updates Binary and Namespace default
- Renaming file when updating interface, enum, class (Ctrl+R,R)
- Introducing local variables after creating an initializer



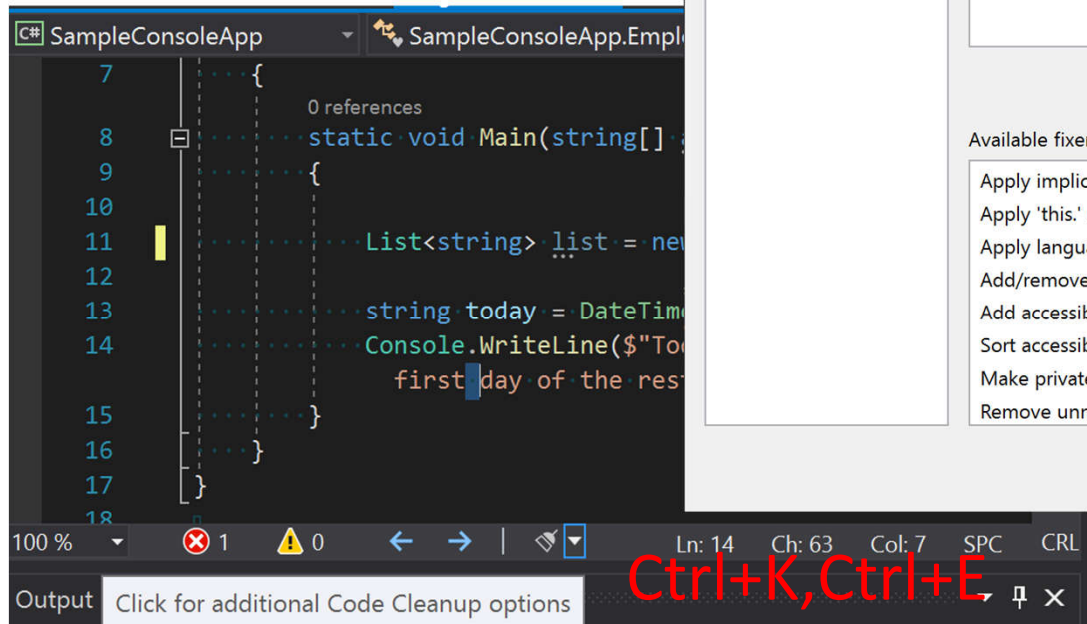
Document Health Indicator

```
C# SampleConsoleApp SampleConsoleApp.Emplc Main(string[] args)
7  ...{
8  0 references
9  ...static void Main(string[] args)
10 ...{
11 ...List<string> list = new List<string> [];
12 ...
13 ...string today = DateTime.Now.ToString("D");
14 ...Console.WriteLine($"Today, {today} is the
15 ...first day of the rest of your life.");
16 ...}
17 ...}
18 ...}
```

Alt+PgUp/PgDn

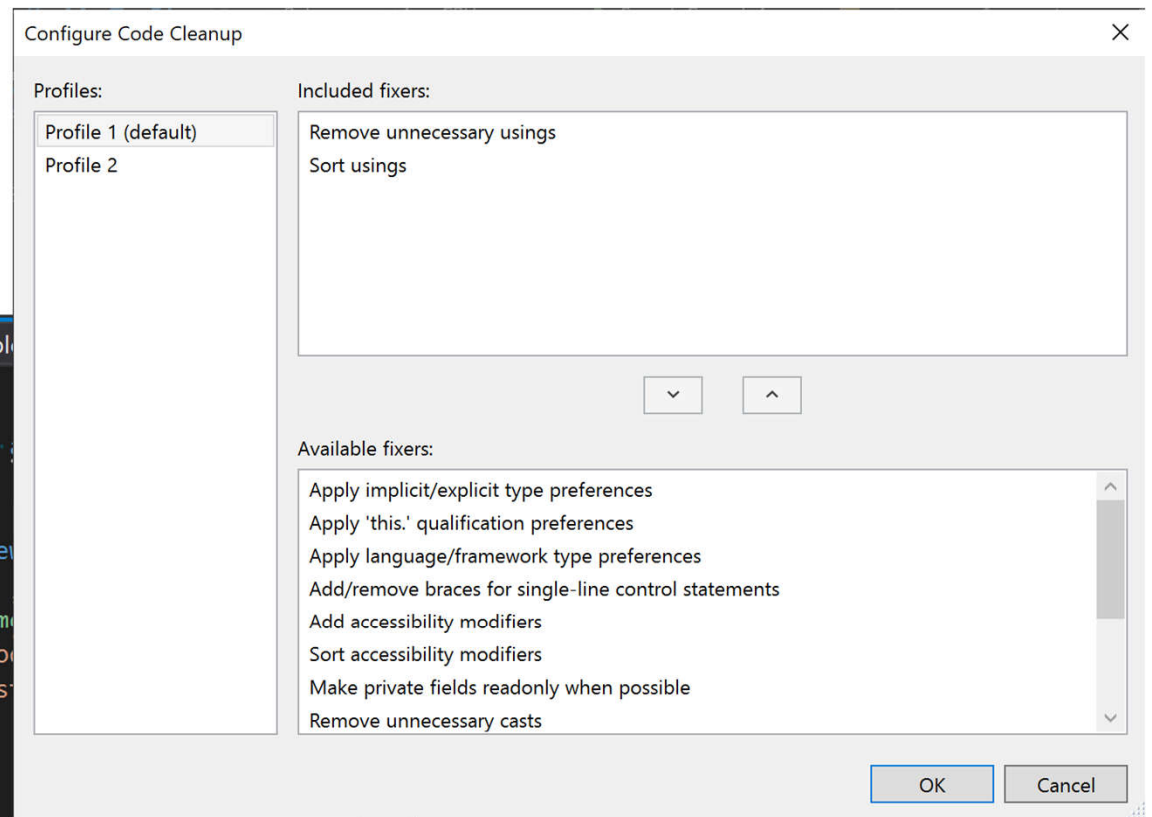
100 % 1 0 Ln: 14 Ch: 63 Col: 7 SPC CRL

Code Cleanup



The screenshot shows the Visual Studio IDE with a C# file named `SampleConsoleApp` open. The code is in a dark theme. The `Main` method is visible, starting at line 8. The code includes a `List<string>` and a `DateTime` object. The status bar at the bottom shows the current cursor position: `Ln: 14 Ch: 63 Col: 7`. The `Output` window is open, showing a message: `Click for additional Code Cleanup options`. A red text overlay `Ctrl+K, Ctrl+E` is positioned over the `Output` window.

```
7     ... {
8     ... static void Main(string[] args)
9     ... {
10    ...
11    ... List<string> list = new List<string>()
12    ...
13    ... string today = DateTime.Now.ToString("d")
14    ... Console.WriteLine($"Today is {today}, the first day of the rest of the year.")
15    ... }
16    ... }
17 }
18
```



The screenshot shows the `Configure Code Cleanup` dialog box. It has a title bar with a close button. The dialog is divided into two main sections: `Profiles:` and `Included fixers:`. The `Profiles:` section contains a list with `Profile 1 (default)` and `Profile 2`. The `Included fixers:` section contains a list with `Remove unnecessary usings` and `Sort usings`. Below these sections are two buttons: `↓` and `↑`. The `Available fixers:` section contains a list of fixers: `Apply implicit/explicit type preferences`, `Apply 'this.' qualification preferences`, `Apply language/framework type preferences`, `Add/remove braces for single-line control statements`, `Add accessibility modifiers`, `Sort accessibility modifiers`, `Make private fields readonly when possible`, and `Remove unnecessary casts`. At the bottom right, there are `OK` and `Cancel` buttons.

Clipboard Ring

```
Console.WriteLine($"Hello, my name is .");
```

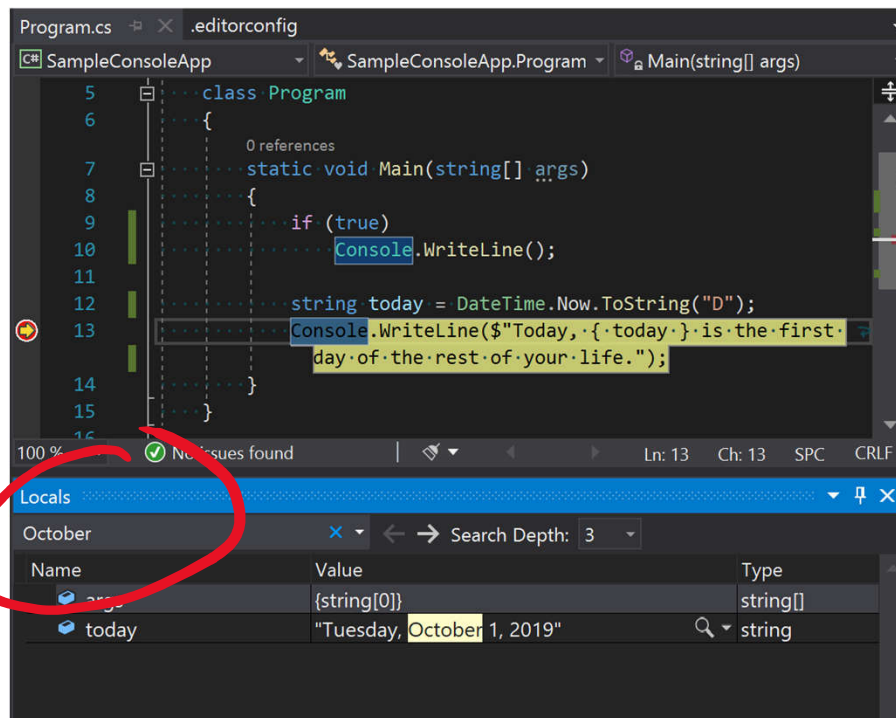
Ctrl+Shift+V

Clipboard

1: Inigo

2: Montoya

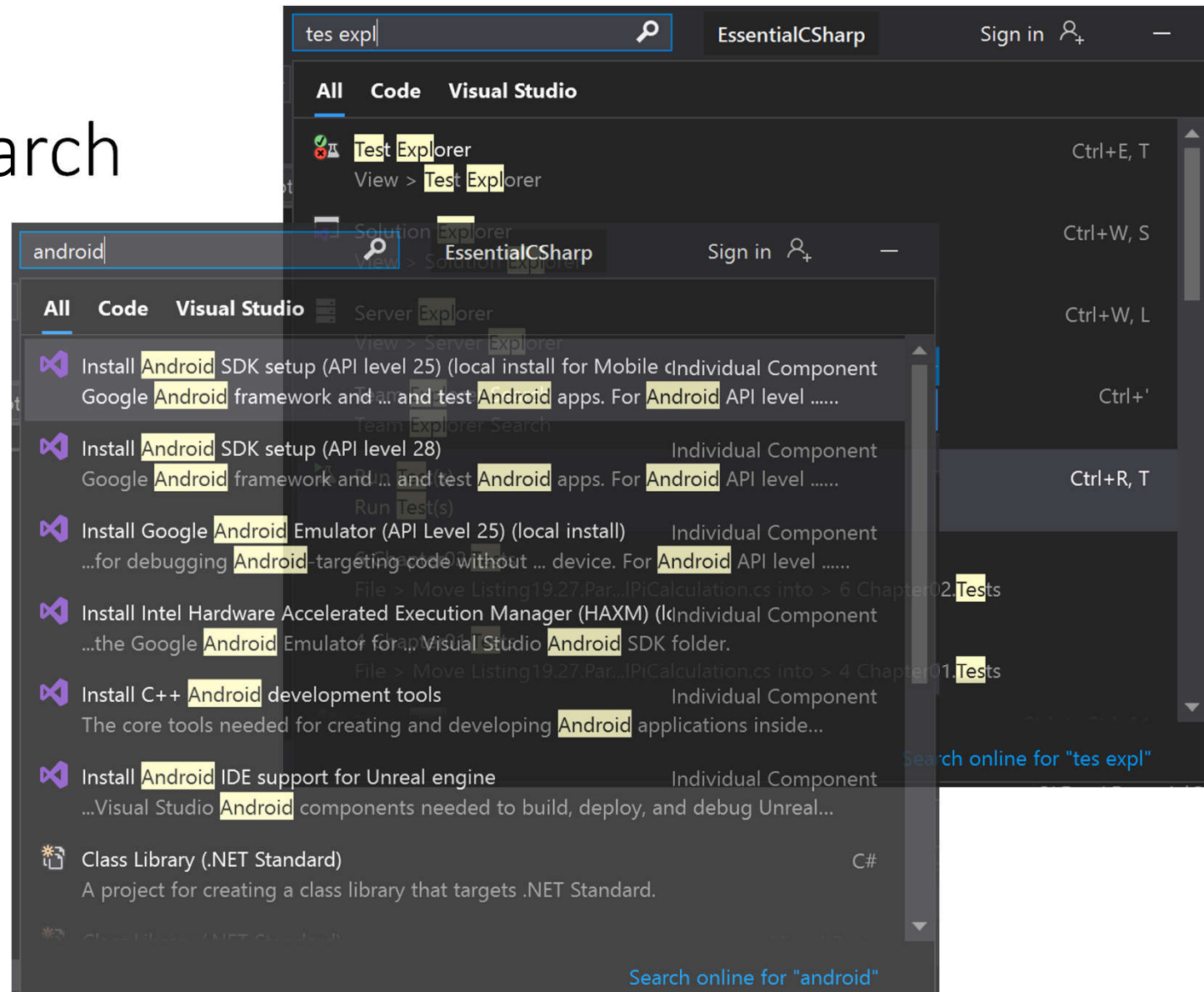
Search Variable Values When Debugging



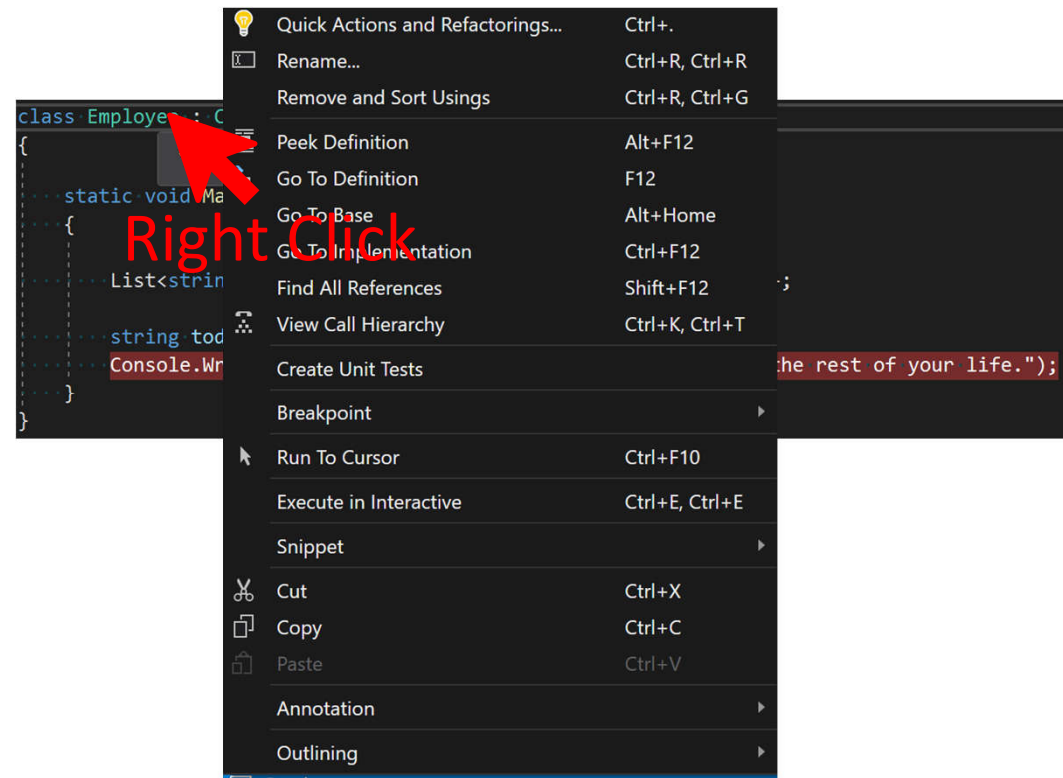
- Work in:
 - autos,
 - locals, and
 - watch window

Improved search

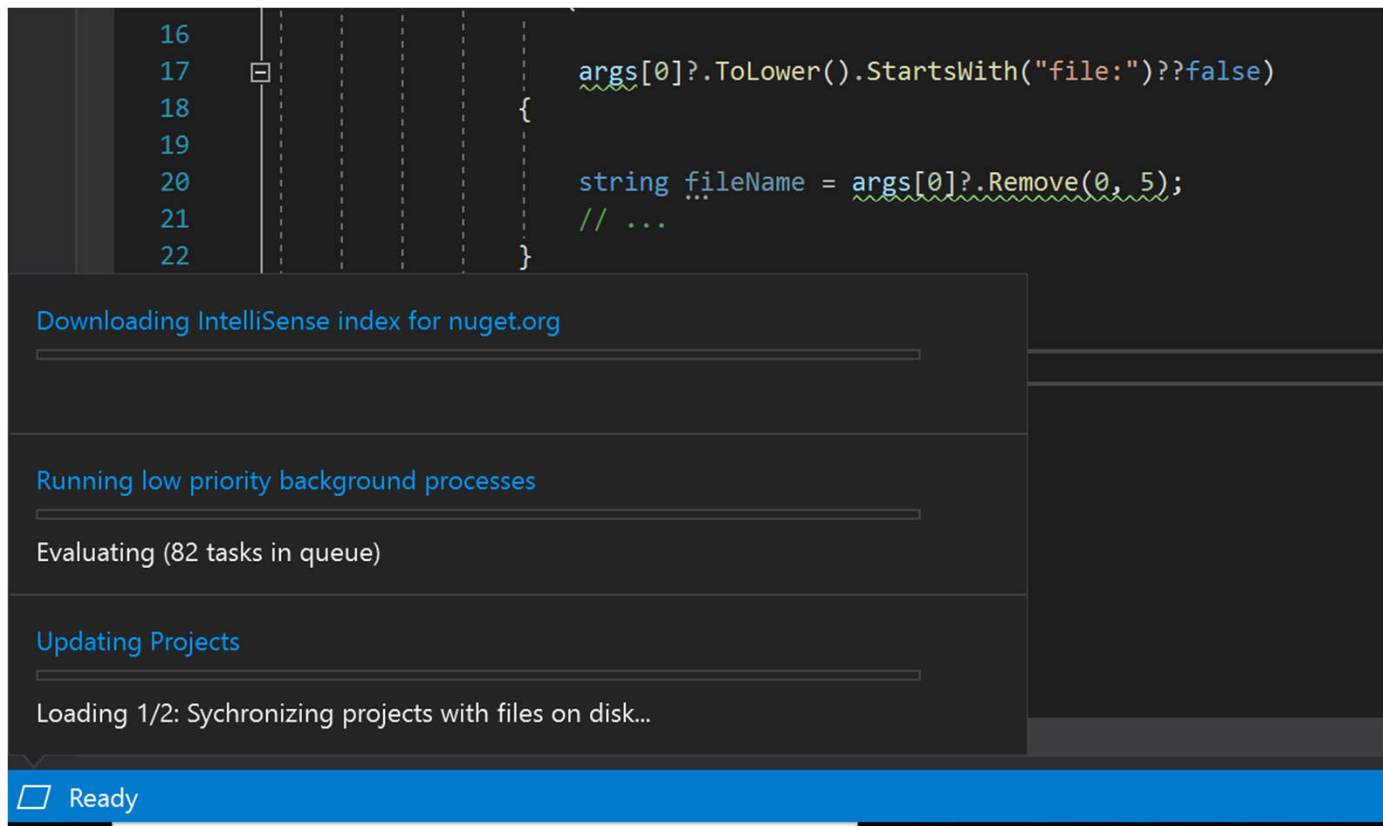
- Settings
- Commands
- Install Options



Navigation Between Constructs



Solution Loading Performance Improvements



The image shows a screenshot of a code editor with a progress window overlaid. The code editor displays the following C# code:

```
16  
17  args[0]?.ToLower().StartsWith("file:")??false)  
18 {  
19   
20 string fileName = args[0]?.Remove(0, 5);  
21 // ...  
22 }
```

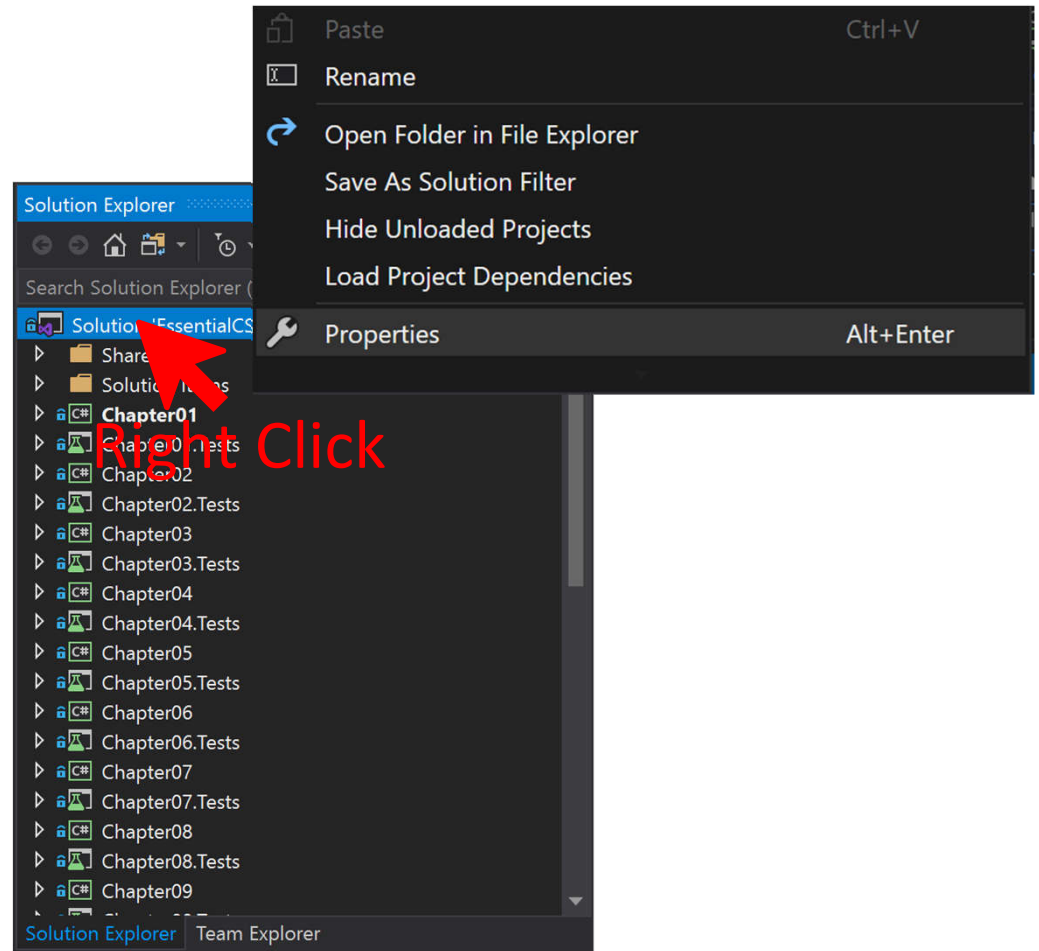
The progress window contains the following items:

- Downloading IntelliSense index for nuget.org
- Running low priority background processes
- Evaluating (82 tasks in queue)
- Updating Projects
- Loading 1/2: Synchronizing projects with files on disk...

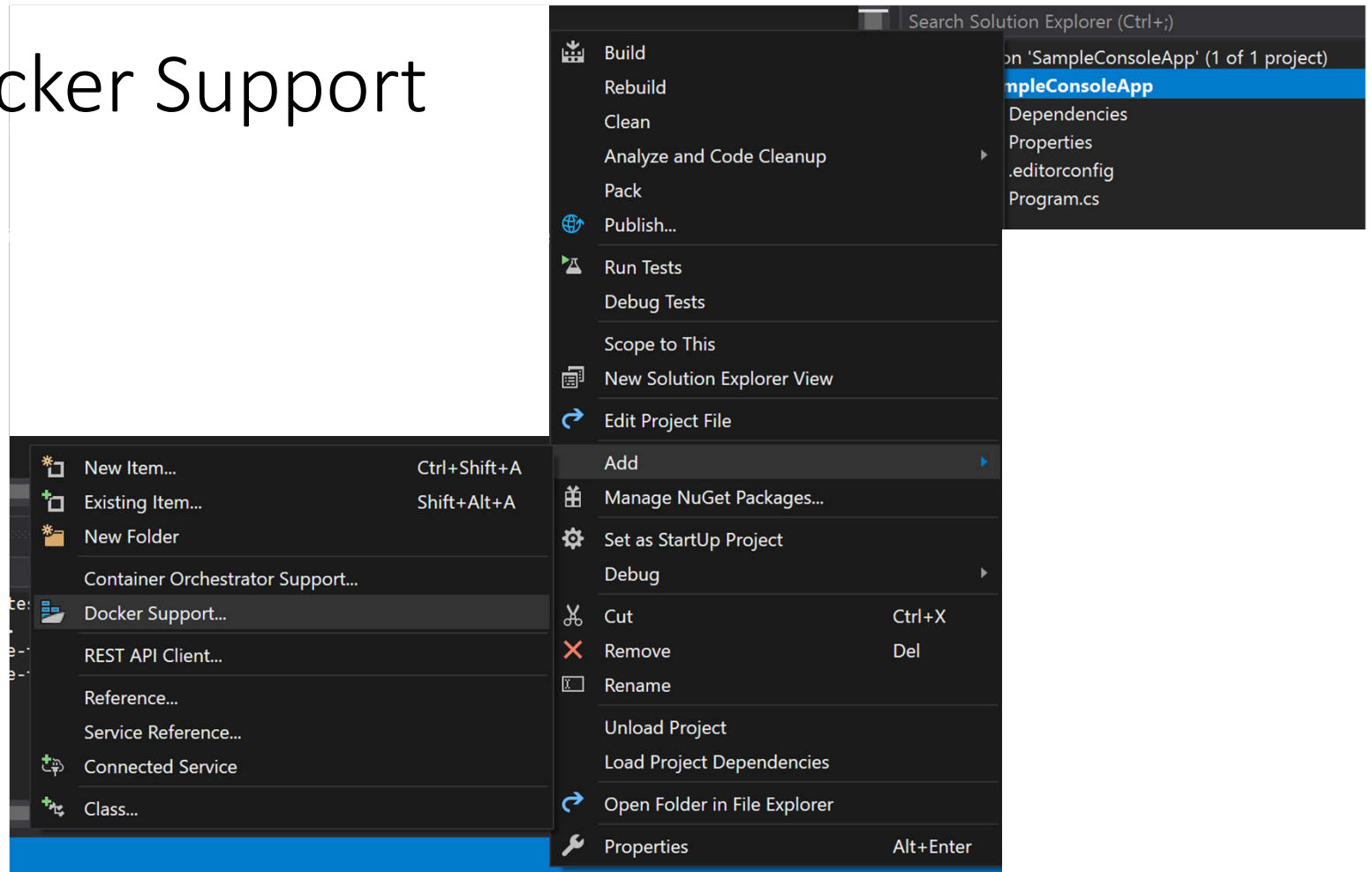
At the bottom of the window, there is a blue bar with a white square icon and the text "Ready".

Solution Filters

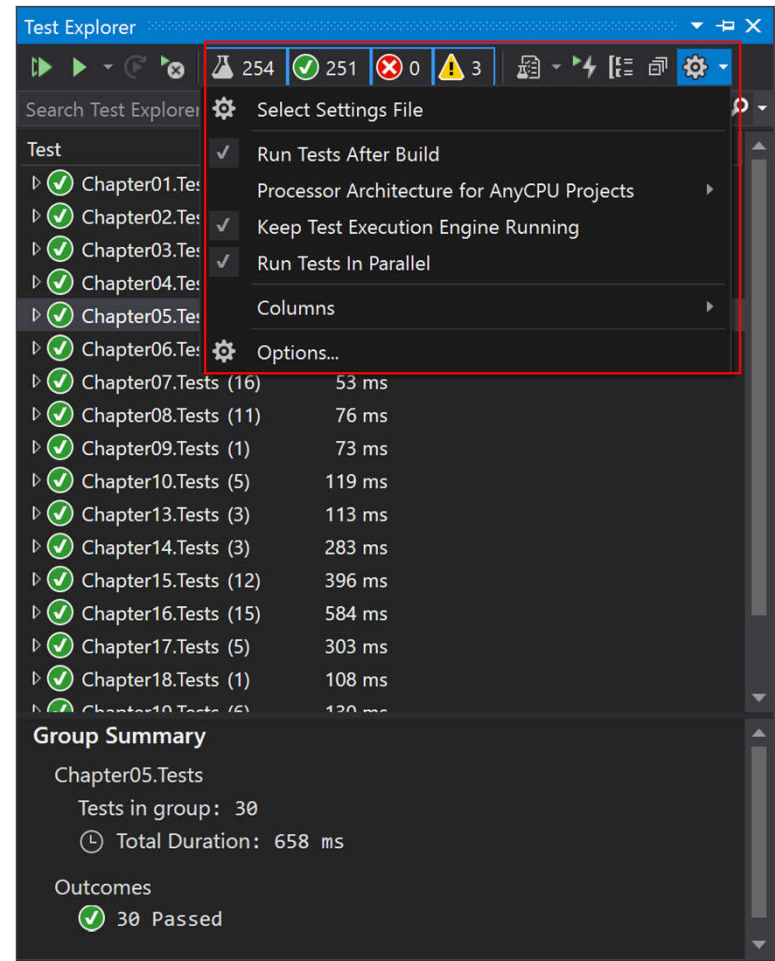
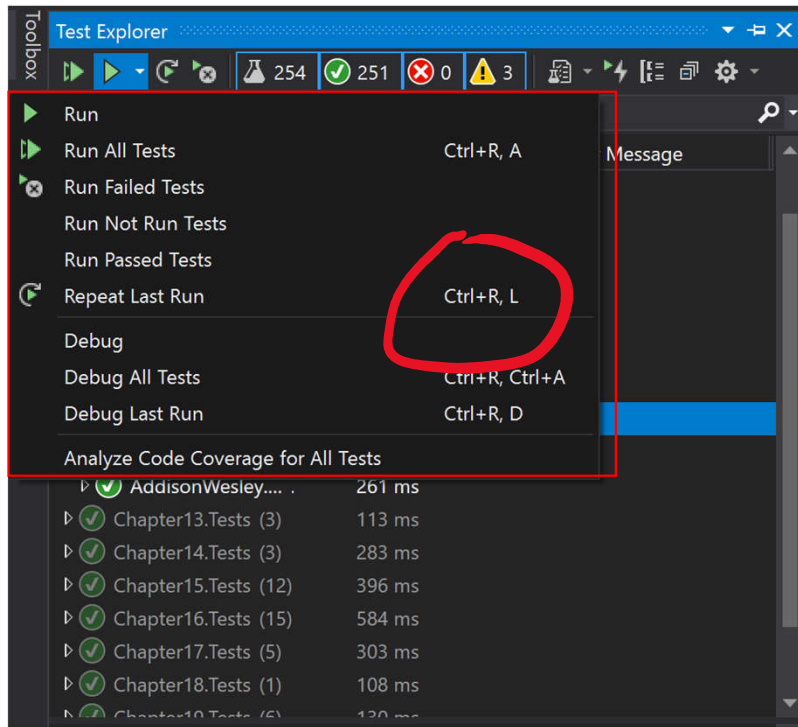
- Load solutions with projects unloaded
- Load project dependencies
- Save solution filters (slfn) – a JSON file



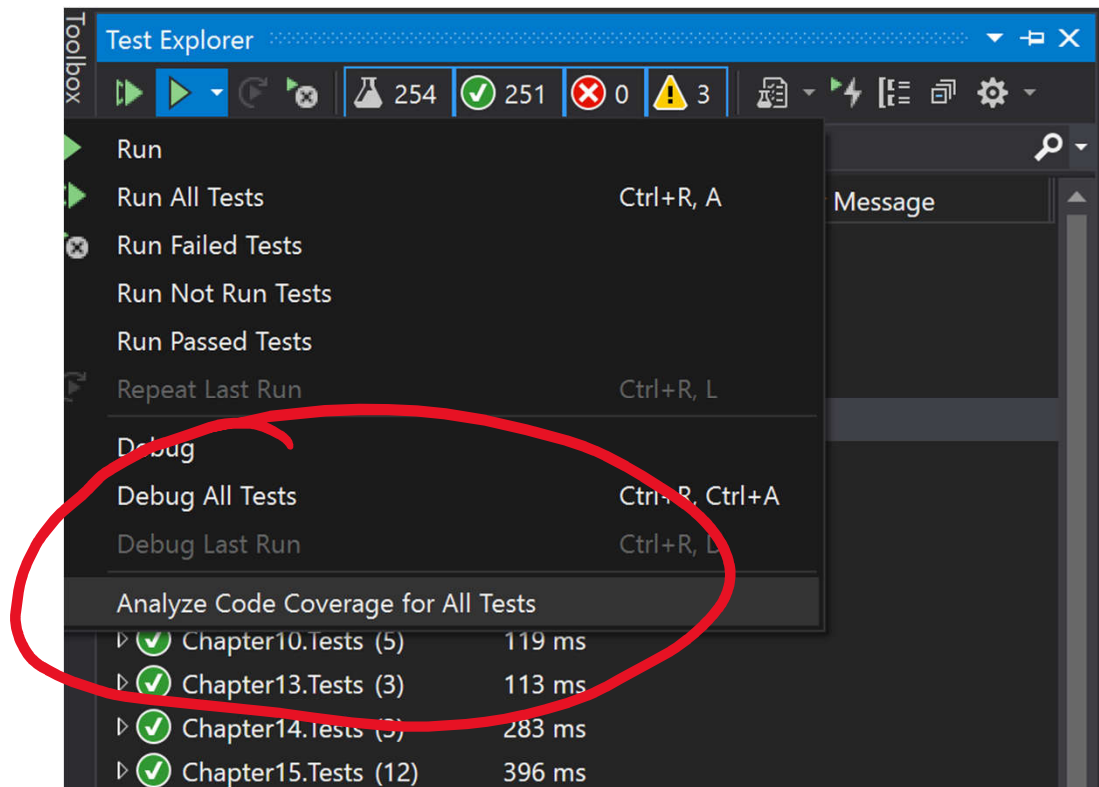
Add Docker Support



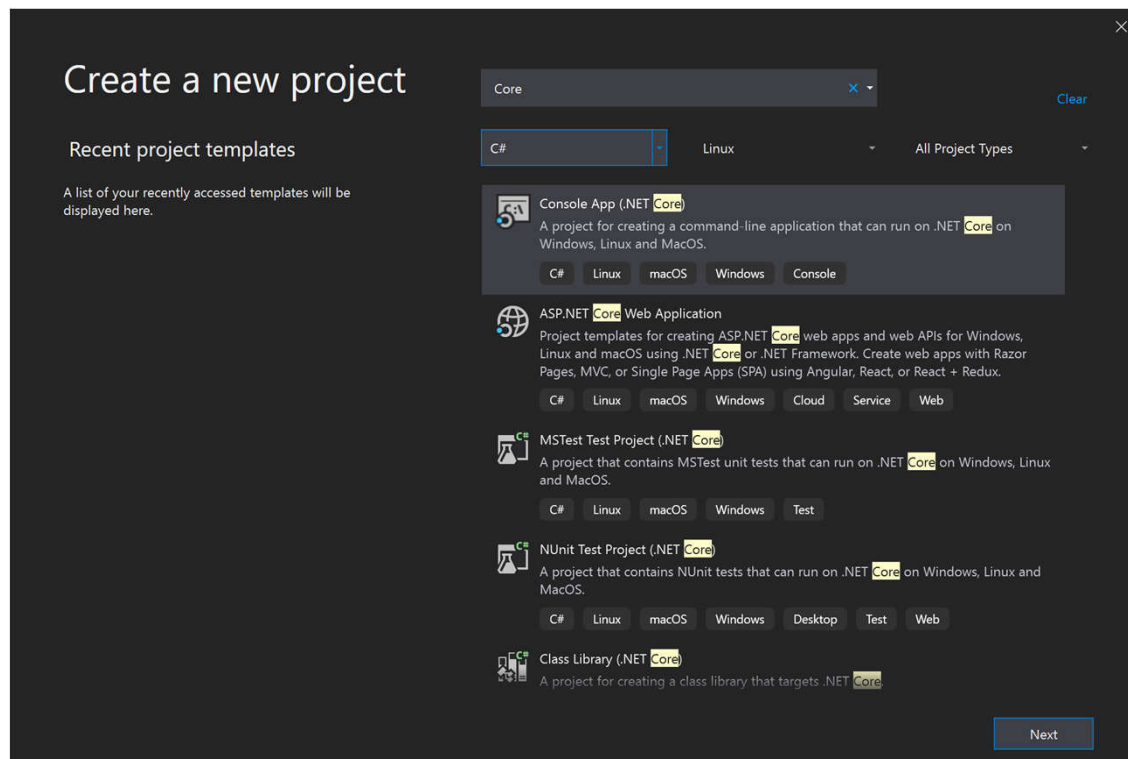
New and Improved Test Explorer



Code Coverage for .NET Core Unit Testing



New Project Dialog



Visual Studio 2019

Open recent

As you use Visual Studio, any projects, folders, or files that you open will show up here for quick access.

You can pin anything that you open frequently so that it's always at the top of the list.

Get started



Clone or check out code

Get code from an online repository like GitHub or Azure DevOps



Open a project or solution

Open a local Visual Studio project or .sln file



Open a local folder

Navigate and edit code within any folder



Create a new project

Choose a project template with code scaffolding to get started

[Continue without code →](#)

Code Analysis

SampleConsoleApp.ruleset NuGet: SampleConsoleApp Sample

Group by: Analyzer ID

ID	Name
<input type="checkbox"/>	Managed Binary Analysis
<input type="checkbox"/>	Microsoft.CodeAnalysis.Ana...
<input checked="" type="checkbox"/>	Microsoft.CodeAnalysis.CSh...
<input checked="" type="checkbox"/>	Microsoft.CodeAnalysis.CSh...
<input checked="" type="checkbox"/>	RS1002 Missing kind argument when registering
<input checked="" type="checkbox"/>	RS1003 Unsupported SymbolKind argument wh
<input checked="" type="checkbox"/>	RS1005 ReportDiagnostic invoked with an unsup
<input checked="" type="checkbox"/>	RS1006 Invalid type argument for DiagnosticAna
<input checked="" type="checkbox"/>	RS1008 Avoid storing per-compilation data into
<input checked="" type="checkbox"/>	RS1012 Start action has no registered actions.
<input checked="" type="checkbox"/>	RS1013 Start action has no registered non-end a
<input checked="" type="checkbox"/>	RS1014 Do not ignore values returned by metho
<input checked="" type="checkbox"/>	RS1022 Do not use types from Workspaces asse
<input checked="" type="checkbox"/>	RS1023 Upgrade MSBuildWorkspace
<input checked="" type="checkbox"/>	Microsoft.CodeAnalysis.CSh...
<input checked="" type="checkbox"/>	Microsoft.CodeAnalysis.Feat...

Solution Explorer

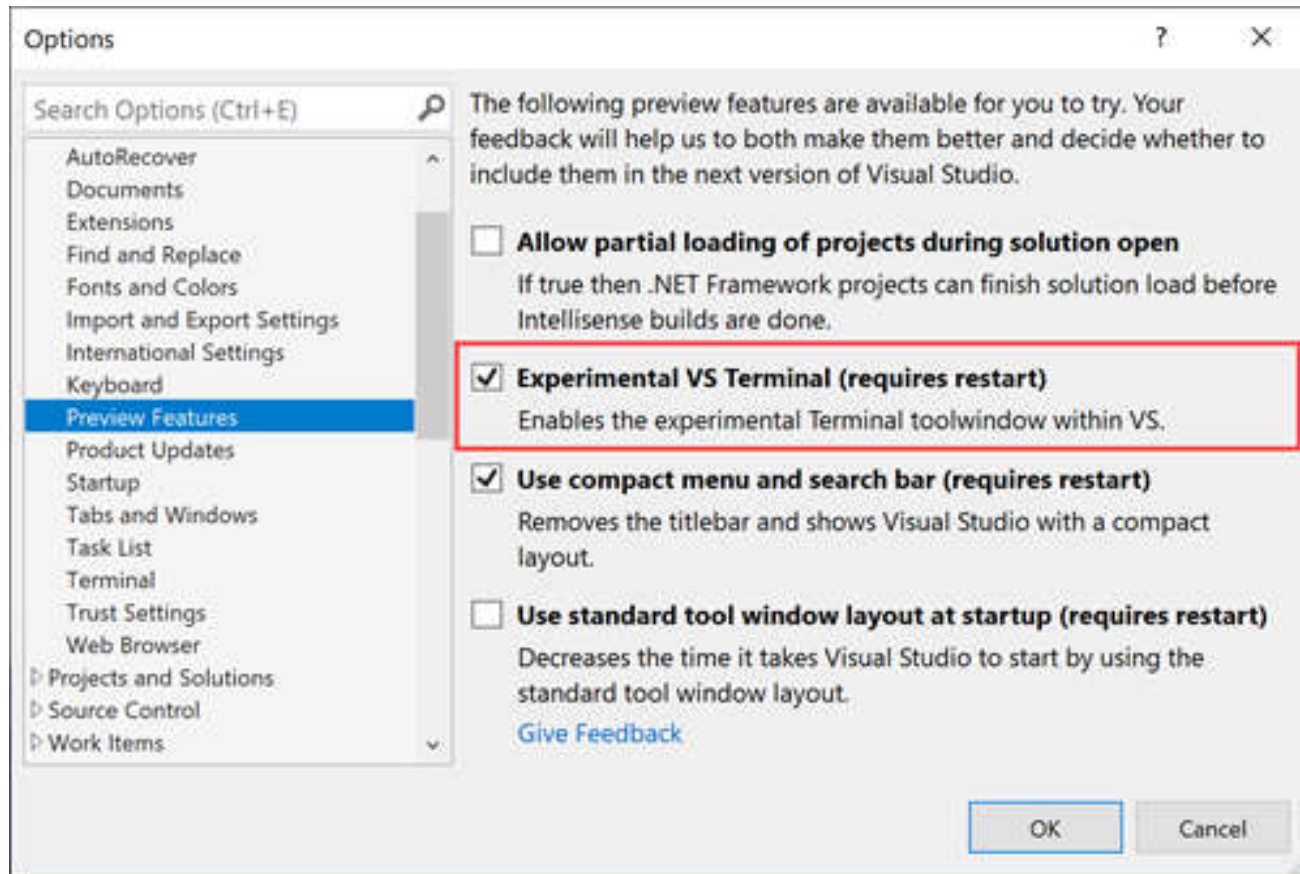
Search Solution Explorer (Ctrl+;)

- Solution 'SampleConsoleApp' (1 of 1 project)
 - SampleConsoleApp
 - Dependencies
 - Analyzers
 - Microsoft.CodeAnalysis.Analyzers
 - Microsoft.CodeAnalysis.CSharp.Analyzers
 - RS1002: Missing kind argument when i
 - RS1003: Unsupported SymbolKind argu
 - RS1005: ReportDiagnostic invoked with
 - RS1006: Invalid type argument for Diag
 - RS1008: Avoid storing per-compilation
 - RS1012: Start action has no registered .
 - RS1013: Start action has no registered
 - RS1014: Do not ignore values returned
 - RS1022: Do not use types from Worksp
 - RS1023: Upgrade MSBuildWorkspace

Multiple

Multiple

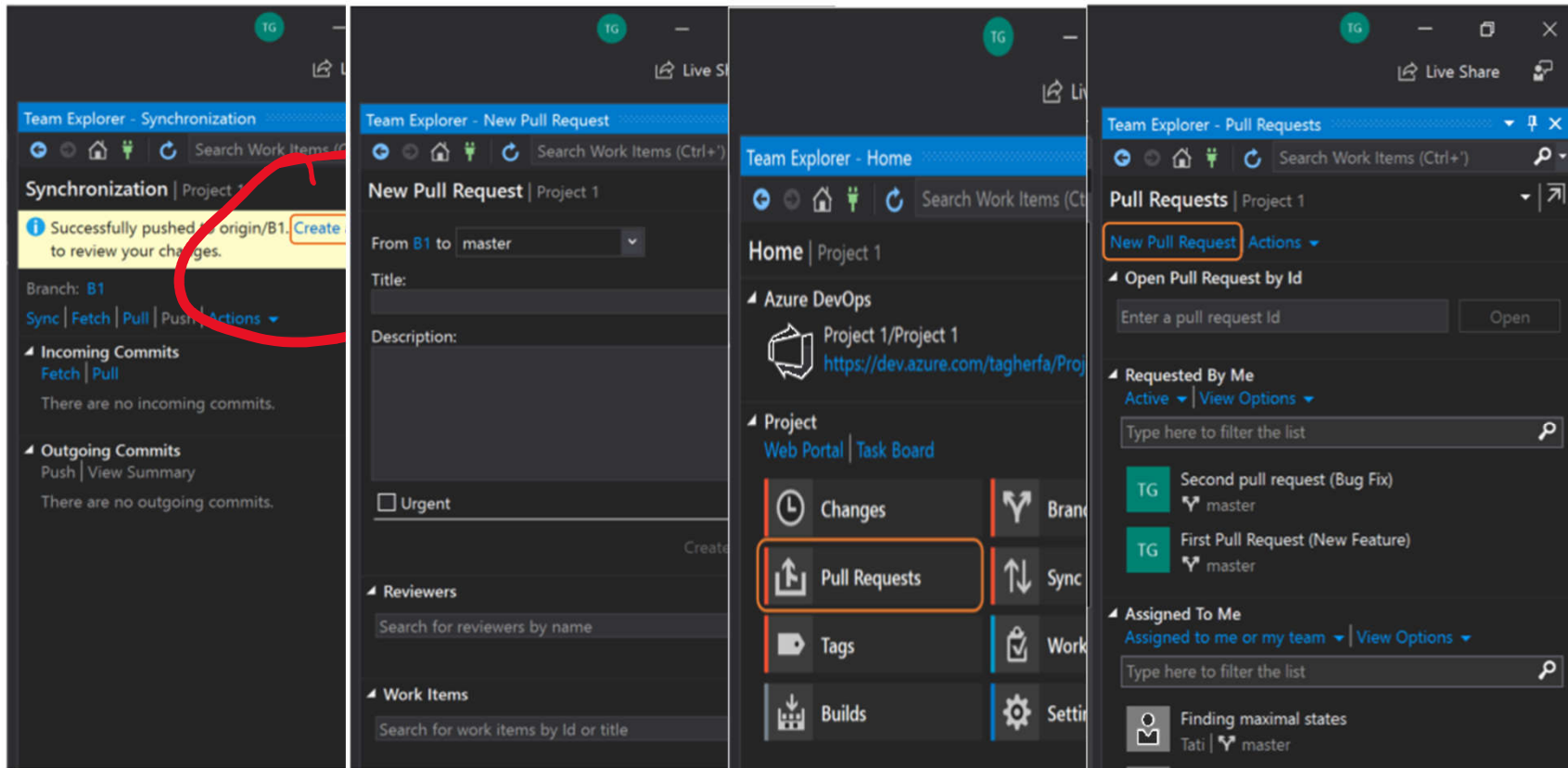
Visual Studio Terminal



.NET Core Updates

- Support for Nullable
- .NET Core 3.0
- .NET Framework 4.8
- F# 4.7

Pull Request: GitHub/Azure DevOps Integration



Creating a pull request right a

New pull request form

Team Explorer's home page

Pull requests page

<https://devblogs.microsoft.com/visualstudio/code-reviews-using-the-visual-studio-pull-requests-extension/>

Pull Request: Review Pull Requests

The screenshot illustrates a pull request review in Visual Studio. The main editor shows a diff of Python code. A comment box is open over the code, displaying a thumbs-up reaction and the text "Great Work". The pull request details panel on the right shows the title "Finding maximal states" and the reviewer "Tatli". The bottom panel shows the pull request comments table.

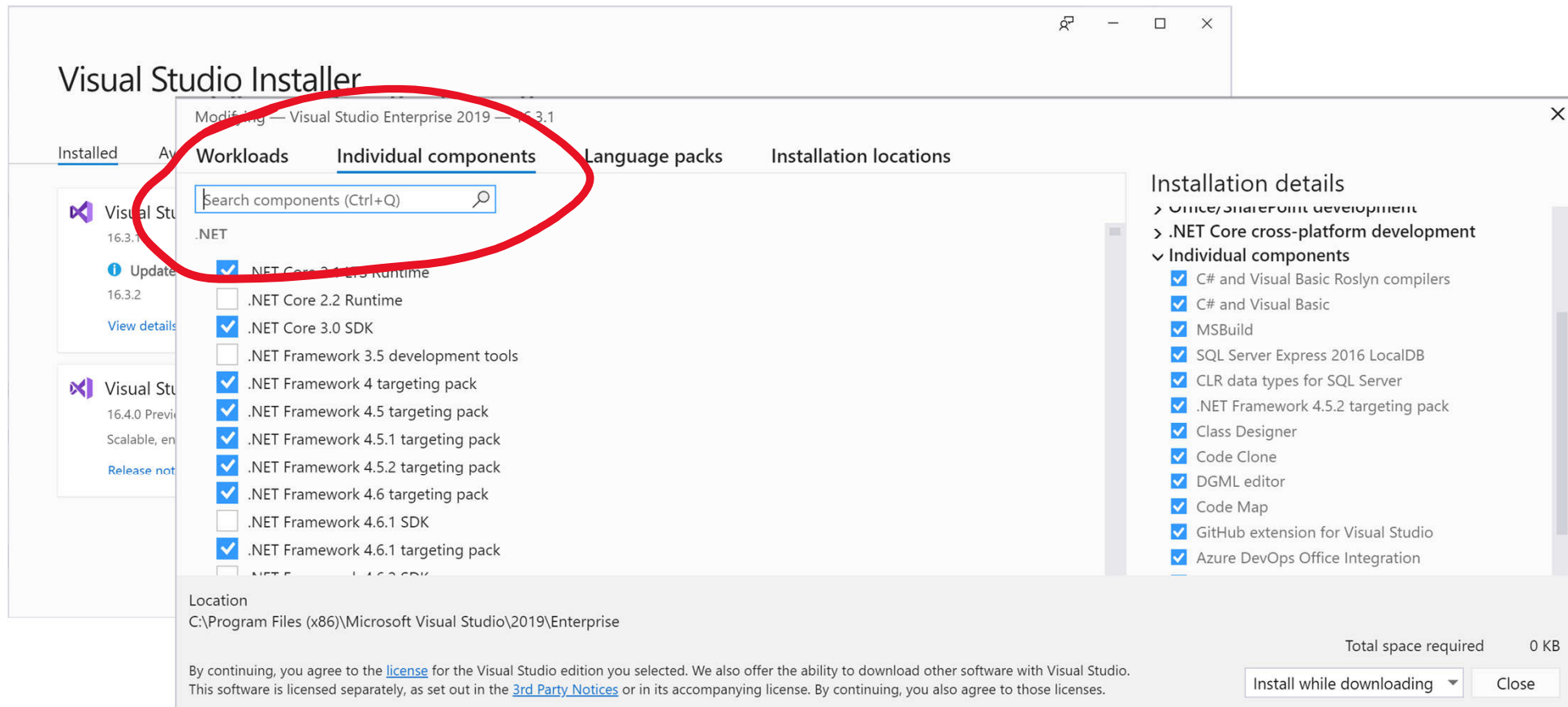
```
running = int (thread.get ("running"))
frame = thread.find ("frame")
if frame is not None:
    address = frame.get ("address")
    states = address_states.setdefault (address, [0, 0])
    states[running] += 1

# Find maximal states
for address, states in list (address_states.items ()):
    n = sum (states)
    if n >= DEDUCE_MIN_PERCENTAGE:
        state = 0
        m = states[0]
        for i in range (1, len (states)):
            if states[i] > m:
                state = i
                m = states[i]
        percentage = m / n
        if percentage >= DEDUCE_MIN_PERCENTAGE:
            address_states[address] = state
        else:
            del address_states[address]
    else:
        del address_states[address]
```

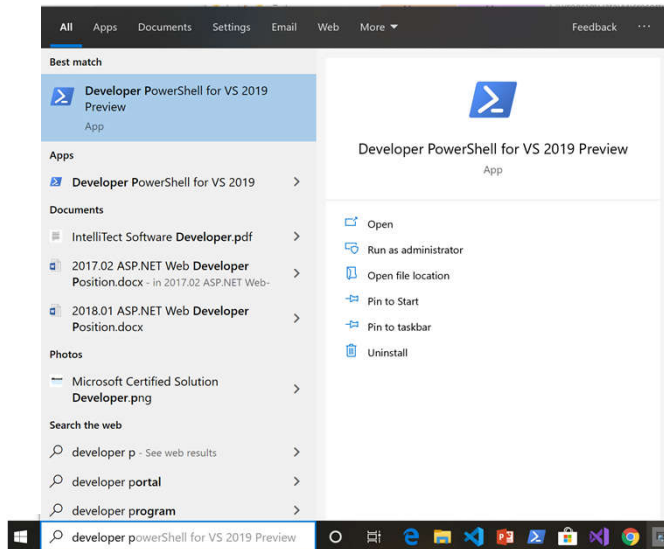
Status	Participant	Update	Search discussions
Active	/PythonApplication1/PythonApplication1/PythonApplication1.py	Reviewer	Last updated
Active	Taysser Gherfal	Sat 5:21 PM	1

<https://devblogs.microsoft.com/visualstudio/code-reviews-using-the-visual-studio-pull-requests-extension/>

Installer... side-by-side, search,



Developer PowerShell



```
Developer PowerShell for VS 2019 Preview
PS C:\Users\mark\source\repos> h

Id CommandLine
-----
1 &{Import-Module "C:\Program Files (x86)\Microsoft Visual Studio\2019\Pre
view\Common7\Tools\Microsoft.VisualStudio.DevShell

PS C:\Users\mark\source\repos> get-module

ModuleType Version Name ExportedCommands
-----
Manifest 3.1.0.0 Microsoft.PowerShell.Utility {Add-Member, Ad...
Binary 16.0.0.0 Microsoft.VisualStudio.DevShell {Enter-VsDevShe...
Script 2.0.0 PSReadline {Get-PSReadLine...

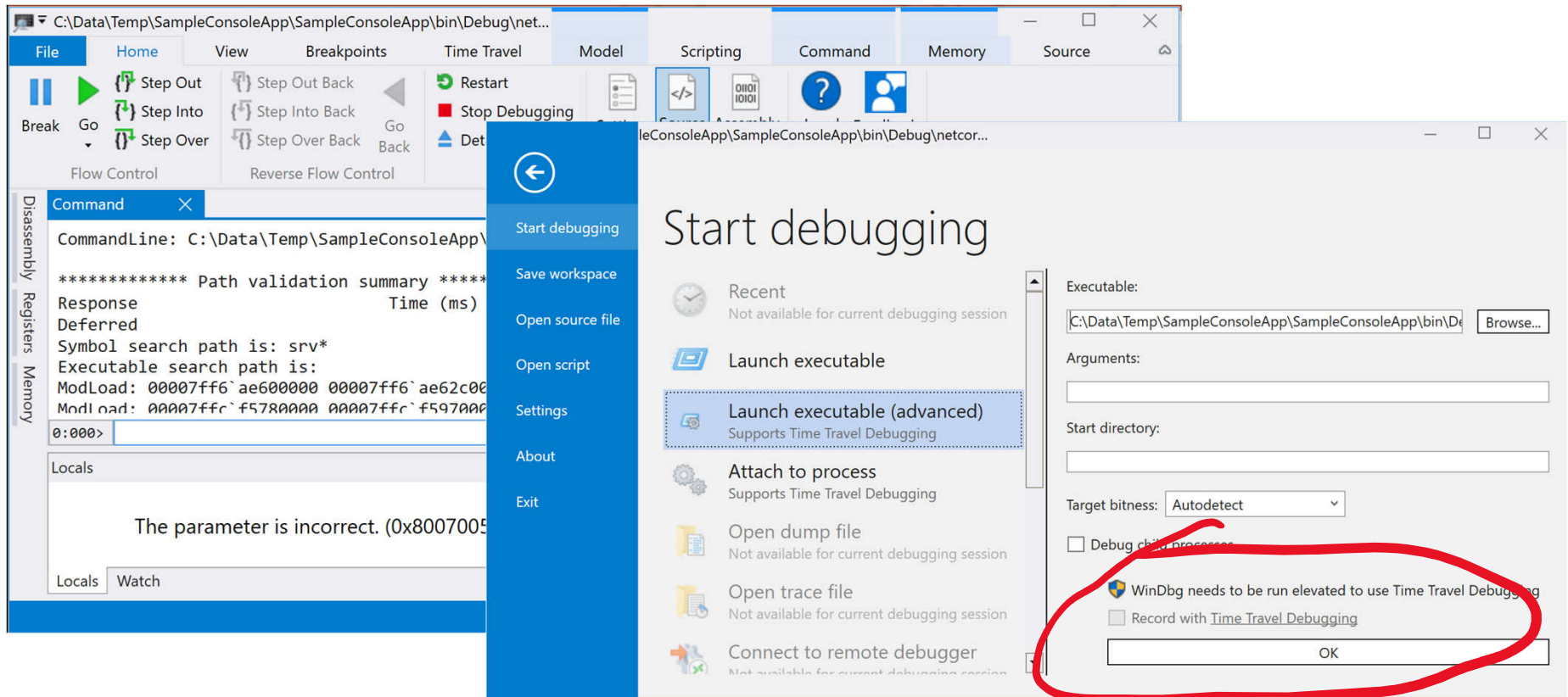
PS C:\Users\mark\source\repos> get-command -module Microsoft.VisualStudio.Dev
Shell

CommandType Name Version
-----
Cmdlet Enter-VsDevShell 16.0.0.0
Cmdlet Send-VsDevShellTelemetry 16.0.0.0

PS C:\Users\mark\source\repos>
```

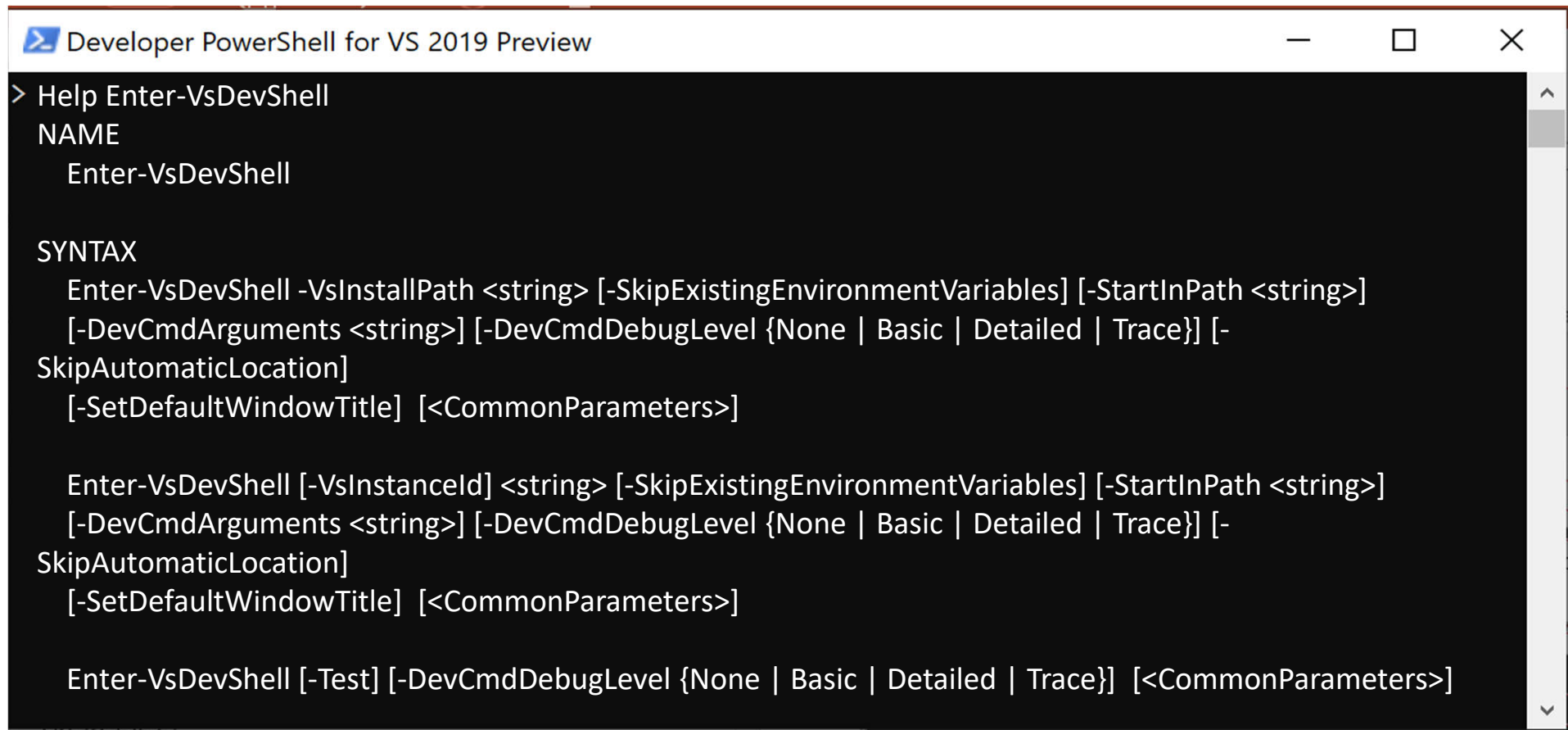
```
C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell
.exe -NoExit -Command "& { Import-Module
.\Common7\Tools\vsdevshell\Microsoft.VisualStudio.Dev
Shell.dll; Enter-VsDevShell -InstanceId ba5a33e2}"
```

Time Travel Debugging (TTD)



<https://docs.microsoft.com/en-us/windows-hardware/drivers/debugger/debugging-using-windbg-preview>

Enter-VsDevShell



```
Developer PowerShell for VS 2019 Preview
> Help Enter-VsDevShell
NAME
    Enter-VsDevShell

SYNTAX
    Enter-VsDevShell -VsInstallPath <string> [-SkipExistingEnvironmentVariables] [-StartInPath <string>]
    [-DevCmdArguments <string>] [-DevCmdDebugLevel {None | Basic | Detailed | Trace}] [-
    SkipAutomaticLocation]
    [-SetDefaultWindowTitle] [<CommonParameters>]

    Enter-VsDevShell [-VsInstanceId] <string> [-SkipExistingEnvironmentVariables] [-StartInPath <string>]
    [-DevCmdArguments <string>] [-DevCmdDebugLevel {None | Basic | Detailed | Trace}] [-
    SkipAutomaticLocation]
    [-SetDefaultWindowTitle] [<CommonParameters>]

    Enter-VsDevShell [-Test] [-DevCmdDebugLevel {None | Basic | Detailed | Trace}] [<CommonParameters>]
```

Visual Studio Enterprise



Live Unit Testing

The screenshot displays the Live Unit Testing window in Visual Studio. The top bar shows a summary of test results: 254 tests passed (green checkmark), 0 tests failed (red X), and 3 tests were skipped (yellow warning triangle). The main area is divided into two panes. The left pane is a table listing test groups with their durations. The right pane shows a detailed summary for the selected group, Chapter02.Tests.

Test	Duration	Traits	Err...
Chapter01.Tests (17)	65 ms		
Chapter02.Tests (26)	83 ms		
Chapter03.Tests (7)	59 ms		
Chapter04.Tests (57)	104 ms		
Chapter05.Tests (30)	458 ms		
Chapter06.Tests (30)	143 ms		
Chapter07.Tests (16)	94 ms		
Chapter08.Tests (11)	92 ms		
Chapter09.Tests (1)	67 ms		
Chapter10.Tests (5)	117 ms		
Chapter13.Tests (3)	76 ms		
Chapter14.Tests (3)	95 ms		
Chapter15.Tests (12)	231 ms		
Chapter16.Tests (15)	212 ms		
Chapter17.Tests (5)	99 ms		
Chapter18.Tests (1)	34 ms		

Group Summary

Chapter02.Tests

Tests in group: 26

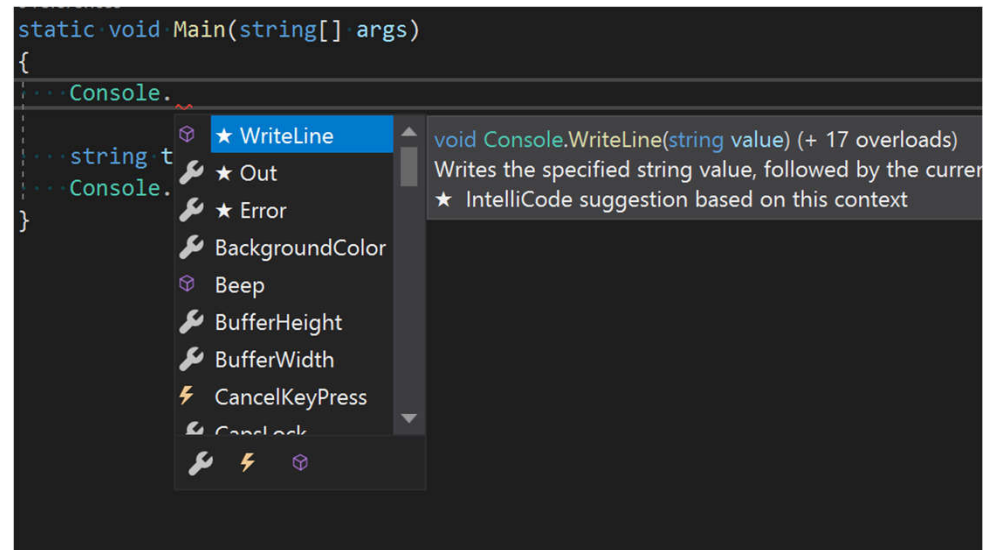
Total Duration: 83 ms

Outcomes

- 25 Passed
- 1 Skipped

IntelliCode

- Get AI-assisted recommendations
- Based on code scans of Github and your own local code base to create a machine learning model.



The screenshot shows a code editor with a C# method signature: `static void Main(string[] args)`. Below the signature, the `Console.` namespace is expanded, and a list of methods is shown. The `WriteLine` method is highlighted in blue. To the right of the list, a tooltip provides details for the selected method: `void Console.WriteLine(string value) (+ 17 overloads)`, followed by the description: "Writes the specified string value, followed by the current line and terminator." Below the description, a star icon indicates it is an "IntelliCode suggestion based on this context".

```
static void Main(string[] args)
{
    Console.
    string t
    Console.
}

★ WriteLine
★ Out
★ Error
BackgroundColor
Beep
BufferHeight
BufferWidth
CancelKeyPress
CancelLock
```

void Console.WriteLine(string value) (+ 17 overloads)
Writes the specified string value, followed by the current line and terminator.
★ IntelliCode suggestion based on this context

IntelliCode

Recommendations for your types, based on your code (Preview -C#)

Argument Completion (Preview)

Inferring code style and formatting conventions (Preview)

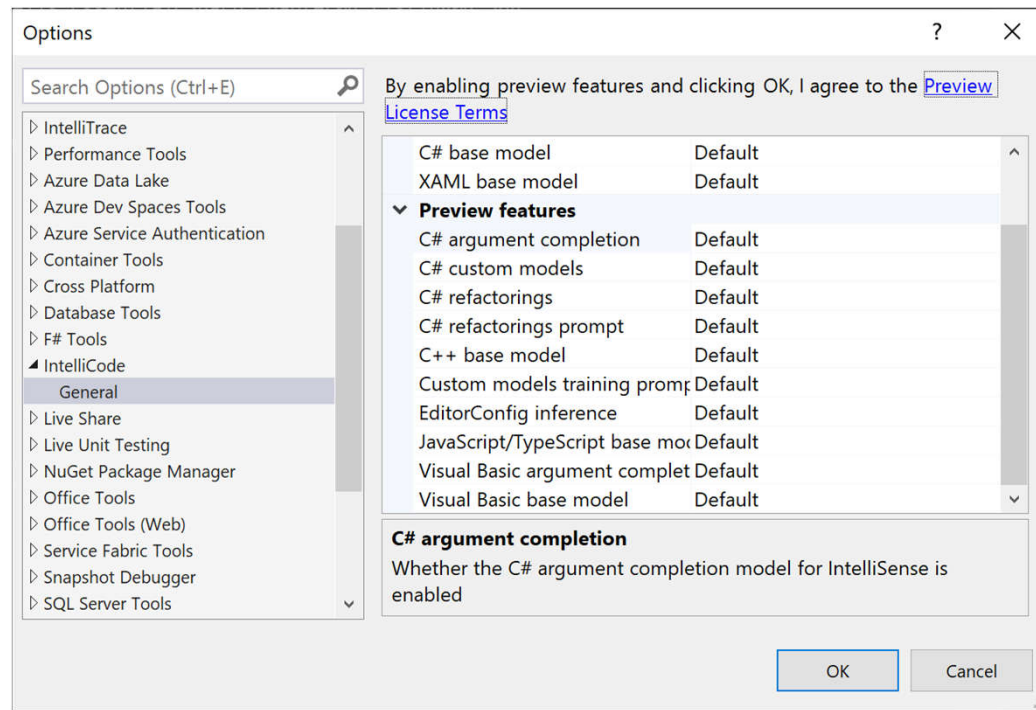
Deliver context-aware code completions

Guide developers to adhere to the patterns and styles of their team

Find difficult-to-catch code issues

Focus code reviews by drawing attention to areas that really matter

IntelliCode Configuration





Meet Mark



- Mark@IntelliTect.com
- fb.com/Mark.Michaelis
- Twitter: [@MarkMichaelis](https://twitter.com/MarkMichaelis)
- [@IntelliTect](https://twitter.com/IntelliTect), fb.com/IntelliTect

