

# Visual Studio 2022 version 17.7 Release Notes

Article 10/10/2023

## Tip

[Watch the recordings of the Visual Studio 2022 launch event](#) to learn about what's new, hear tips & tricks, and download free digital swag.

[Developer Community](#) | [Visual Studio 2022 Roadmap](#) | [System Requirements](#) |  
[Compatibility](#) | [Distributable Code](#) | [Release History](#) | [License Terms](#) | [Blogs](#) | [Latest Release Known Issues](#) | [What's New in Visual Studio Docs](#)

Click a button to download the latest version of Visual Studio 2022. For instructions on installing and updating Visual Studio 2022, see [Update Visual Studio 2022 to the most recent release](#). Also, see instructions on [how to install offline](#).

[Download Community 2022 ↓](#)

[Download Professional 2022 ↓](#)

[Download Enterprise 2022 ↓](#)

Visit the [Visual Studio](#) site to download other Visual Studio 2022 products.

## Visual Studio 2022 version 17.7 Releases

- Oct 10th 2023 — [Visual Studio 2022 version 17.7.5](#) NEW!
- Sep 12th 2023 — [Visual Studio 2022 version 17.7.4](#)
- Aug 29th 2023 — [Visual Studio 2022 version 17.7.3](#)
- Aug 22nd 2023 — [Visual Studio 2022 version 17.7.2](#)
- Aug 15th 2023 — [Visual Studio 2022 version 17.7.1](#)
- Aug 8th 2023 — [Visual Studio 2022 version 17.7](#)

## Visual Studio 2022 Blog

The Visual Studio 2022 Blog is the official source of product insight from the Visual Studio Engineering Team. You can find in-depth information about the Visual Studio 2022 releases in the following posts:

- [Visual Studio 2022 version 17.7](#)
- 

## Visual Studio 2022 version 17.7.5

*released Oct 10th, 2023*

### Summary of What's New in this Release of Visual Studio 2022 version 17.7.5

- This fixes an issue where the configuration that was selected in previous Visual Studio sessions wasn't being restored.
- The CMake cache no longer generates automatically when "Never run configure step automatically" is selected in the settings.
- Fixed C++ modules bugs and added several modules throughput improvements.  
When using LTCG (/GL), in certain rare cases, the compiler could fail to realize that a global variable might be modified during program execution and instead incorrectly conclude it's constant, and improperly propagate its initial value to locations where the global variable is used.
- Updates .NET 7 MAUI to SR8. Fixes an issue where MAUI apps cannot compile with Xcode 15 when Linker was disabled. For full release notes please visit:  
<https://github.com/xamarin/xamarin-macios/releases>
- Fixed several issues related to display DPI changes that could result in a crash or hang.
- Fixed an issue where opening a folder would cause an error related to "Process Telemetry" to be displayed if certain Visual Studio Workloads or Individual Components are not installed.

### Developer Community

- Variable CMAKE\_INSTALL\_PREFIX is defined multiple times
- Build hangs for ASP.NET Core project when using Visual Studio 2022 version 17.7.3
- document outline function so slow.
- HTTP Editor Special char handling is wrong

- System.Windows.Extensions types are not supported on platform .NET6 with Studio Version 17.6.5
- Version 17.7.0 Preview 2.0 - WebApp unable to connect to SQL Managed Instance using Active Directory Default Authentication and VisualStudioCredential
- Indexing sometimes hangs on 2022 Preview 2
- Remote Terminal Exception: Method 'ClosePtyAsync' does not have an implementation
- "Format on Save" extension no longer works as of VS 17.7
- Toggling breakpoints ruins undo stacks
- Visual Studio 17.6 doesn't remember which CMakePresets.json configuration was in use after restarting
- WebApp unable to connect to SQL Managed Instance using Active Directory Default Authentication and VisualStudioCredential
- DTE2.Documents Exception 0x8000000A
- Visual Studio 2022 crashes when restored after minimizing to taskbar
- 'Find in files' dialog doesn't open
- Cmake cache configuration runs automatically even when "never run configure automatically" is tickened



## Visual Studio 2022 version 17.7.4

*released Sep 12th, 2023*

### Summary of What's New in this Release of Visual Studio 2022 version 17.7.4

- Updated version of Git used by Visual Studio to v 2.41.0.3.
- Fixed a crash while applying changes with hot reload on a decompiled module.
- Ensures the security of the VS Installer Elevation Service by adding quotes to the path, preventing process hijacking along the path.

### Developer Community

- Code Lens TypeScript/JavaScript references animation
- Visual Studio not asserting for 0xC0000005: Access violation

## Security Advisories Addressed

- [CVE-2023-36799](#) A vulnerability exists in .NET where reading a maliciously crafted X.509 certificate may result in Denial of Service. This issue only affects Linux systems.
- [CVE-2023-36796](#) This security update addresses a vulnerability in DiaSymReader.dll when reading a corrupted PDB file which can lead to Remote Code Execution.
- [CVE-2023-36794](#) This security update addresses a vulnerability in DiaSymReader.dll when reading a corrupted PDB file which can lead to Remote Code Execution.
- [CVE-2023-36793](#) This security update addresses a vulnerability in DiaSymReader.dll when reading a corrupted PDB file which can lead to Remote Code Execution.
- [CVE-2023-36792](#) This security update addresses a vulnerability in DiaSymReader.dll when reading a corrupted PDB file which can lead to Remote Code Execution.
- [CVE-2023-36759](#) This security update removes pgodriver.sys, where reading a malicious file can lead to Elevation of Privilege.
- [CVE-2023-36758](#) This security update addresses a vulnerability in VSInstallerElevationService when running a malicious executable which can lead to Elevation of Privilege.



## Visual Studio 2022 version 17.7.3

*released Aug 29th, 2023*

### Summary of What's New in this Release of Visual Studio 2022 version 17.7.3

- Added support to list simulators and devices with Xcode 15 and removed the NewsstandKit hard link dependency.
- Fixed an issue preventing the Visual Studio Installer from running when there is a symlink in the installer path.
- The new Include Cleanup feature for C++ which was added in 17.7 has been causing crashes for users, so we have disabled it for now, with a view to re-enable in 17.8

## Developer Community

- [Opening a solution hangs if any files from a certain project were open](#)
- [VS 2022 crashes when attempting to open .sql file](#)

- Kestrel Url containing wildcards not working on Visual Studio 2022 17.7
  - VSIX installation failure on VS2022
  - With New VS 2022 Update 17.7.0 Functional test are not working
- 



## Visual Studio 2022 version 17.7.2

*released Aug 22nd, 2023*

### Summary of What's New in this Release of Visual Studio 2022 version 17.7.2

- Certain endpoint security and application control programs were inducing instability in Visual Studio. This has been addressed.
- Fixed crashes experienced by some users of the "Install WSL For Me" experience.
- Users who opened a folder containing a CMake project may have experienced an indefinite hang in Visual Studio. The race condition causing this hang has been identified and fixed.

### Developer Community

- Remove Github actions from solution explorer
  - Diagnostic Suppressors are not taken into account in the QuickInfo tooltip with VS 2022 17.7
- 



## Visual Studio 2022 version 17.7.1

*released Aug 15th, 2023*

### Summary of What's New in this Release of Visual Studio 2022 version 17.7.1

### Developer Community

- Commented out code is not colored correctly
- Output Window for Application disappear

- Fixes Csproj Editor does not open due to AndroidPackageFormat
- Solution Folders cannot be renamed



## Visual Studio 2022 version 17.7

released Aug 8th, 2023

### Summary of What's New in this Release of Visual Studio 2022 version 17.7

#### *Productivity Features*

Features	Community Suggestions
Comparing Files	– 543 votes
Copy and Trim Indentation	– 86 votes
Multi-branch Git Graph	– 149 votes
Create new GitHub Actions	
Parallel Stack Filtering	
Collapse All Descendants for Solution Explorer	– 7 votes

#### *Performance Improvements*

Features	Community Suggestions
F5 Speed	
All-in-One Search Responsiveness	
Find in Files Search Improvements	– 122 votes
C# Spell Checker Memory Consumption	
C++ Unreal Engine Time to IntelliSense	

Features	Community Suggestions
Light Bulb Responsiveness	
Solutions Load CPU Consumption	

## .NET development

Features	Community Suggestions
Auto-decompilation for External .NET Code	 – 29 votes
New Auto Insights for the CPU usage tool	

## C++ and game development

Features	Community Suggestions
Unreal Engine Macros Formatting for C++	
Build Insights in Visual Studio (C++)	 – 141 votes
Unreal Engine Naming Convention Checker for C++	
Improved Doxygen overload resolution	
Unreal Engine Blueprint Find All References	
Address Sanitizer continue_on_error Support	
Include Cleanup	

## Linux and embedded development with C++

Features	Community Suggestions
WSL Automatic Acquisition	
Remote File Explorer Updates	

## [Other C++ Updates...]

### *Web Development*

Features	Community Suggestions
npm Dependency Management in ASP.NET projects	
Supporting Vite for React and Vue new project creation	
HTTP Editor	
Power Platform connected services	

### *F#*

### *SSDT (SQL)*

Features	Community Suggestions
Support for Serverless SQL Pool in SSDT	

### *Installation and Updates*

Features	Community Suggestions
Rollback now preserves all user settings	

### *Enterprise Management and Acquisition*

Features	Community Suggestions
Ability for standard users to update and modify Visual Studio	 - 97 votes

### *Microsoft 365 development*

Features	Community Suggestions
Microsoft Teams development tools (Teams Toolkit)	

## Other Developer Community Suggestions: You asked, we listened!

- Cannot import anymore tables into edmx. "Running transformation: System.NullReferenceException: Object reference not set to an instance of an object." 46 votes
- Selection in Batch Build was not remembered after restart 41 votes
- .vs folder grows unexpectedly large 32 votes
- Call stack and other windows not rendering correctly 21 votes
- SignTool Error: The /p option requires the /f option 12 votes
- In MAUI, Can't Use F7 To Xaml and Use Shift+ F7 To CS 10 votes
- Clicking pinned tab drags window out and unpin 9 Votes
- MSVC compiler hangs on user defined operator overload 8 votes
- HLSL Tools will crash Visual Studio when viewing shader files in Unreal Engine and the virtual file mapping has been provided 7 votes
- error C1010 with VS 2022 version 17.6.0 Preview 3 and precompiled headers
- SSDT IN VS 2022 IS TOTALLY BROKEN
- Compiler crashes for analysis of code including specialization of class template with nested non-template class referring to containing class
- VS failed to launch with random JSON dialog
- Azure Function's host.json being ignored unless a manual rebuild is done or code is modified

## Productivity Features:

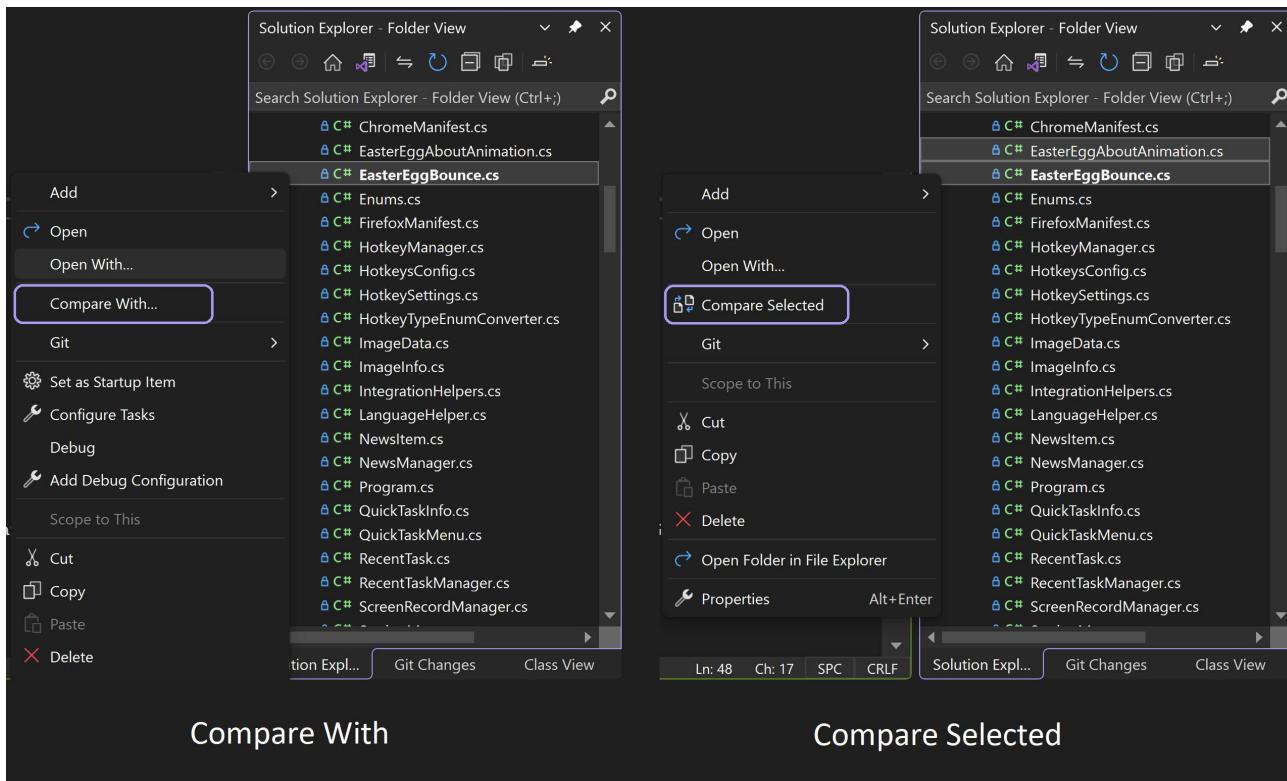
### Compare Files

You no longer need to leave Visual Studio and rely on other tools to compare files. Now you can easily compare any file in Solution Explorer with other files by either:

- Right-clicking on a single file, selecting "Compare With..." from the context menu which will bring up File Explorer. Navigating to any file on disk and selecting it for

comparison.

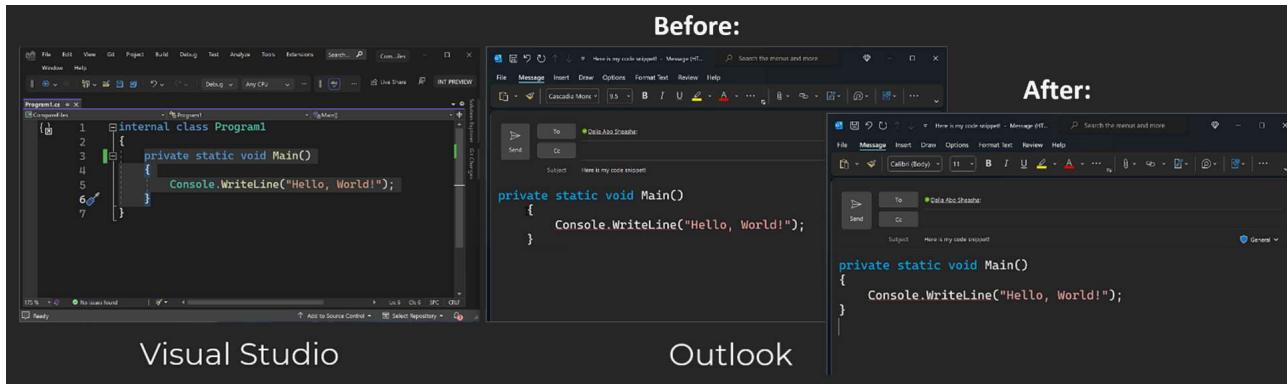
- Multi-selecting two files by holding down the Ctrl button then right-clicking and selecting “Compare Selected” from the context menu.



Please share your feedback [here](#). We'd love to hear your thoughts on this new experience.

## Copy and Trim Indentation

When you copy code from Visual Studio to another program (Outlook, Teams, Slack, etc), Visual Studio will now automatically fix the indentation for you.



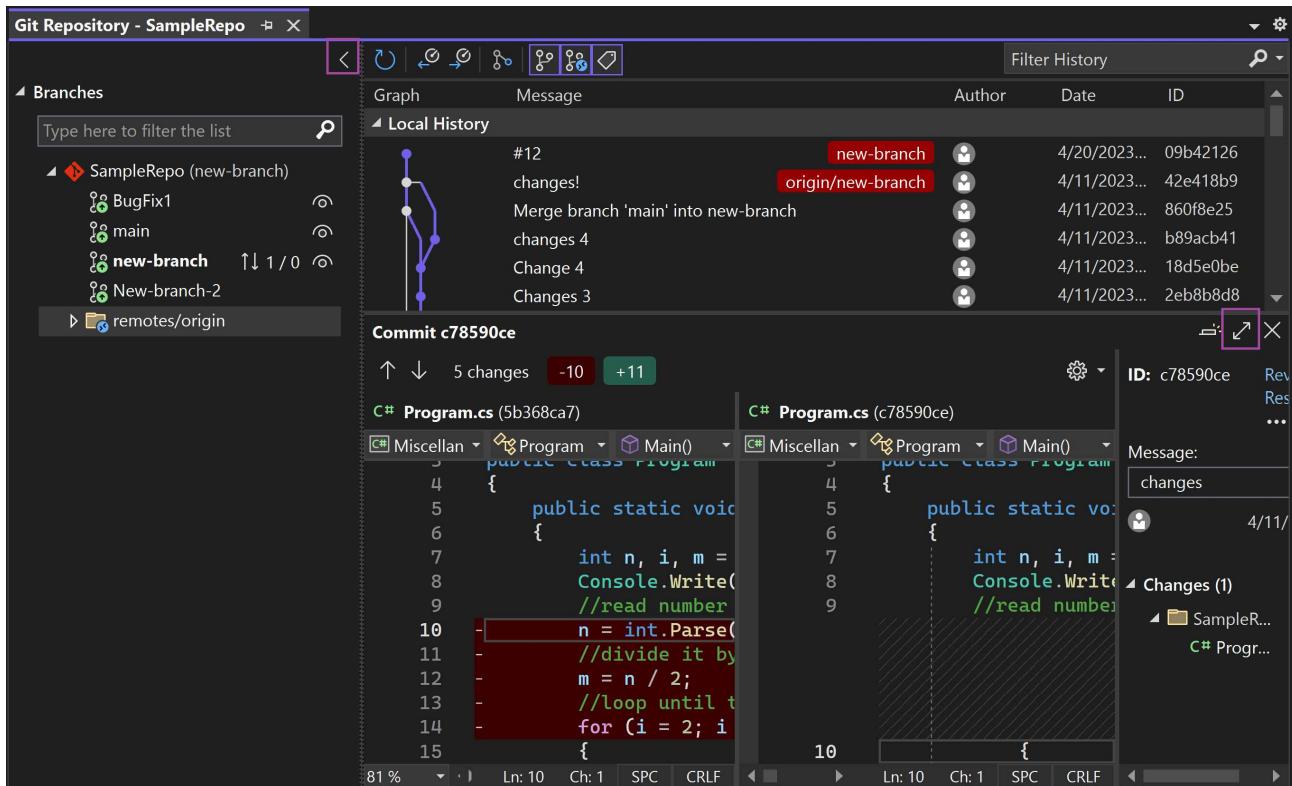
## Multi-branch Git Graph

Building on the work in preview, we've implemented the multi-branch graph to enable visualizing multiple branches in the same graph along with other UI improvements.

The screenshot shows the Git Repository interface for the 'ShareXDemo' repository. On the left, there's a sidebar with a 'Branches' section listing branches like 'ShareXDemo (featurebranch5)', 'bugfix1', 'develop', 'featurebranch1', 'featurebranch2', 'featurebranch3', 'featurebranch4', 'featurebranch5', 'master', 'misc', and 'remotes/origin'. The 'featurebranch5' branch is currently selected. The main area features a 'Graph' view where three branches are shown: 'featurebranch5', 'featurebranch4', and 'develop'. Each branch has a vertical timeline of commits represented by circles. A commit in 'featurebranch5' is highlighted with a blue box. The commits are listed in the table below, showing details like message, author, date, and ID. The table columns are Branch / Tag, Graph, Message, Author, Date, and ID.

Branch / Tag	Graph	Message	Author	Date	ID
featurebranch5	More updates	Jessie H	6/5/2023...	fd11e21a	
featurebranch5	New change	Jessie H	6/5/2023...	be84ca05	
featurebranch5	Add fix for #10	Jessie H	5/10/2023...	7641220d	
featurebranch5	closes #2 similar...	Jessie H	5/24/2023...	98297cd6	
featurebranch5	Followed insight...	Jessie H	5/10/2023...	d824ca7b	
featurebranch5	Update Z	Jessie H	5/10/2023...	cd0b48d0	
featurebranch5	Update Y and Z	Jessie H	5/10/2023...	c6db62ae	
featurebranch5	Fix for Y	Jessie H	5/10/2023...	da90a2d2	
featurebranch5	Followed insight...	Jessie H	5/10/2023...	7cf2edb1	
featurebranch5	Update Z	Jessie H	5/10/2023...	864f9677	
featurebranch5	Update Y and Z	Jessie H	5/10/2023...	016ea334	
featurebranch5	Fix for Y	Jessie H	5/10/2023...	0b633c3d	
featurebranch5	Add some chan...	Jessie H	5/10/2023...	e60edf14	
featurebranch5	Merge PR into d...	Jessie H	5/10/2023...	ccb0c686	
featurebranch5	Hotfix YZ	Jessie H	5/10/2023...	919c895c	
featurebranch5	Add X, Y, and Z	Jessie H	5/10/2023...	0198f0d8	
featurebranch5	Fix for X	Jessie H	5/10/2023...	533aef30	
featurebranch5	More changes	Jessie H	5/10/2023...	ce74313d	
featurebranch5	Some changes	Jessie H	5/10/2023...	d1f86c57	

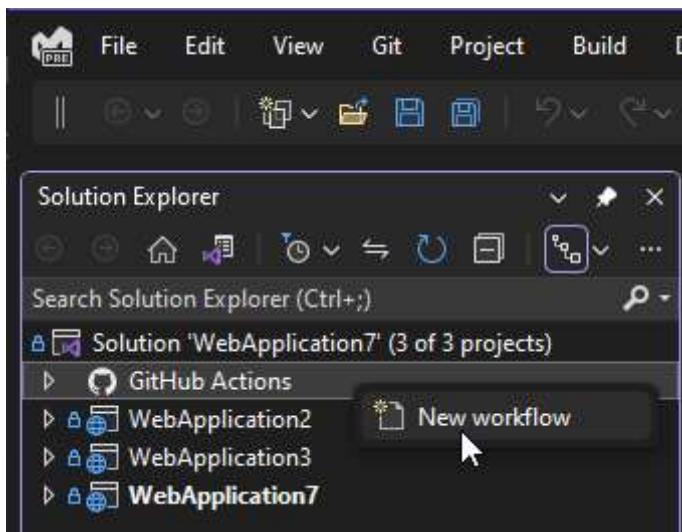
You can select other branches to compare to the currently checked out branch or use the hover eye icon to toggle branches on and off in the graph, keeping track of your incoming and outgoing branches on the currently checked out branch with the new summary icons. Additionally, you can see avatars for the authors in Azure DevOps repos, get smoother graph rendering updates, and maximize your space by collapsing the branches panel or opening your commit details.



Please share your feedback on this feature [here](#)

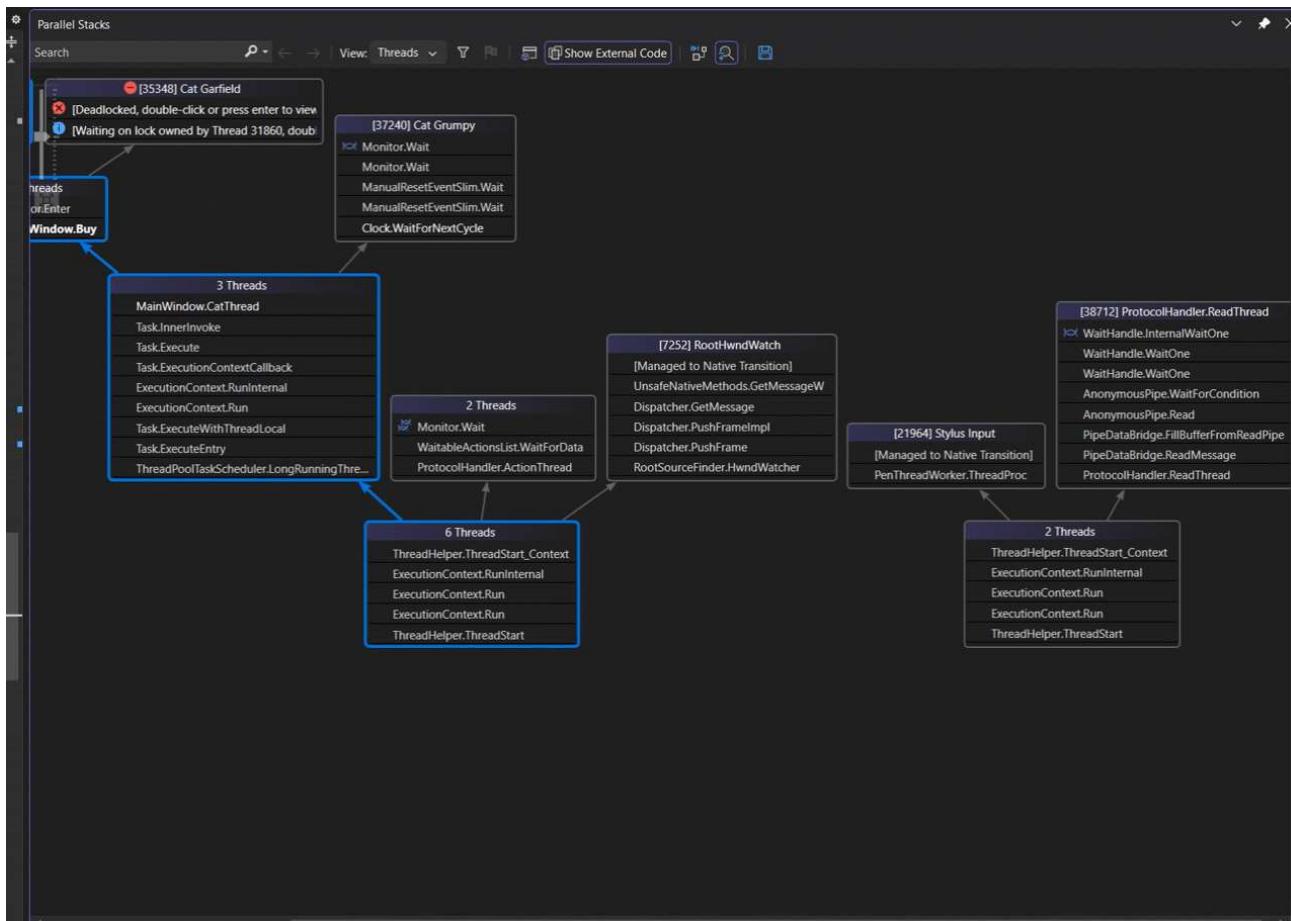
## Create new GitHub Actions

You can now create new GitHub Actions workflows from Solution Explorer. It supports Azure Container Apps and Azure Kubernetes Service as destinations, and it supports deploying multiple projects with a single workflow. When your project is under source control on GitHub.com, a new node appears in Solution Explorer representing GitHub Actions (if you are not seeing the new node, make sure "Show GitHub Actions in Solution Explorer" is checked under Tools > Options > Preview Features). Right-click and select "Create new" to try out the new experience.



## Parallel Stacks Filter

The Parallel Stack Window now supports filtering options. With filtering, you can easily focus on just the frames and threads that matter to you, so you can achieve greater clarity and get more done in less time. In addition to filtering we have now added the ability to Drag Select frames from the parallel stack window. So now you can easily perform operations like copy/paste, flag/unflag or freeze/thaw for all the selected frames at once.



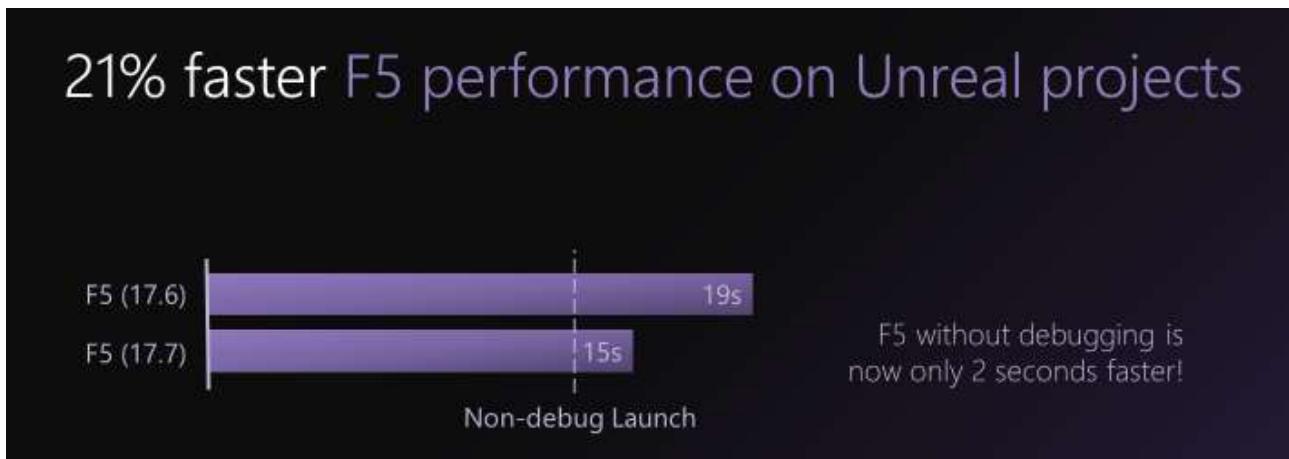
## Collapse All Descendants for Solution Explorer

- We've added a "Collapse All Descendents" command to the context menu when you right-click on a node in Solution Explorer. You can use this to collapse the selected node as well as its descendent nodes. We've also mapped this as a shortcut via the Ctrl+Left arrow keys.

## Performance Improvements:

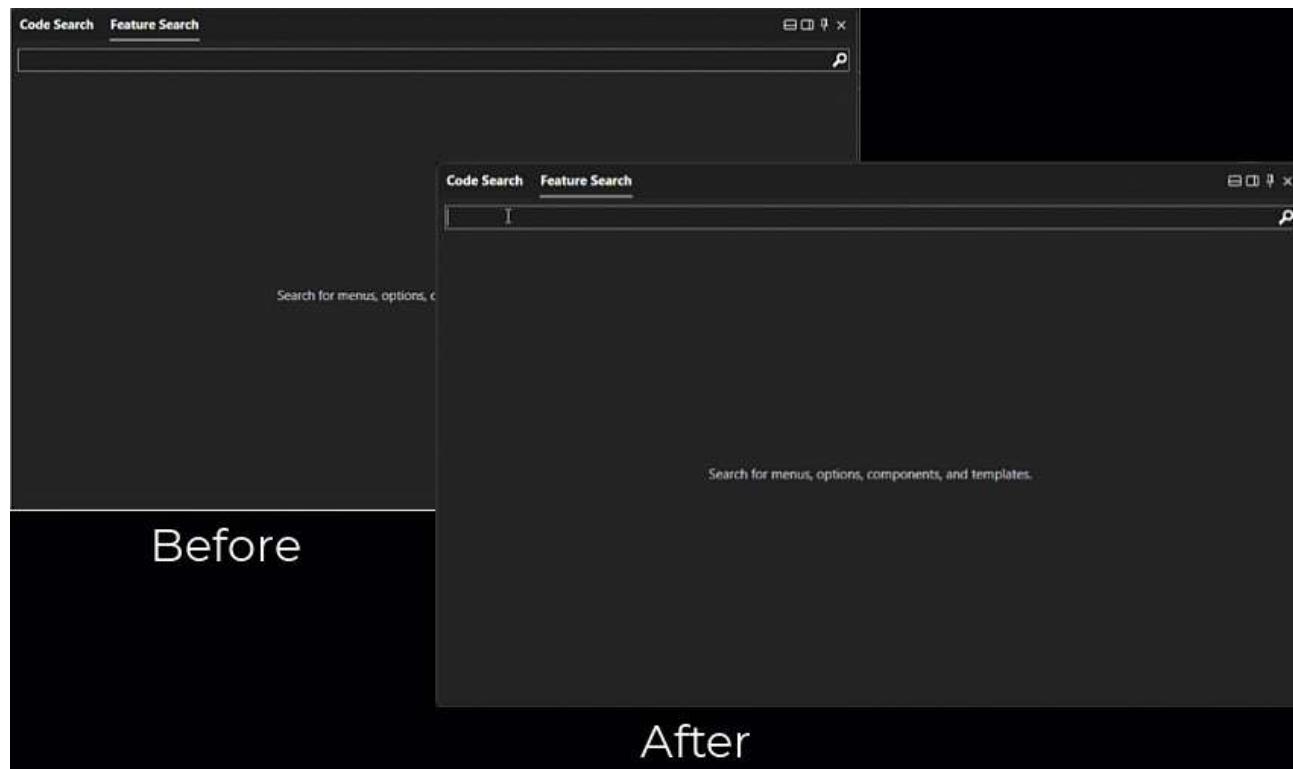
### F5 Speed

We recognize that performance of Visual Studio is critical to you. With this in mind, we have optimized Program Database (PDB) files, resulting in improving debugging sessions and decreasing the project selection screen loading time for Unreal projects for example by 21% (~4 sec gain). Additionally, the F5 path is optimized, leading to a 5-10% improvement in the debugger's startup time and process launch.



## All-in-One Search Responsiveness

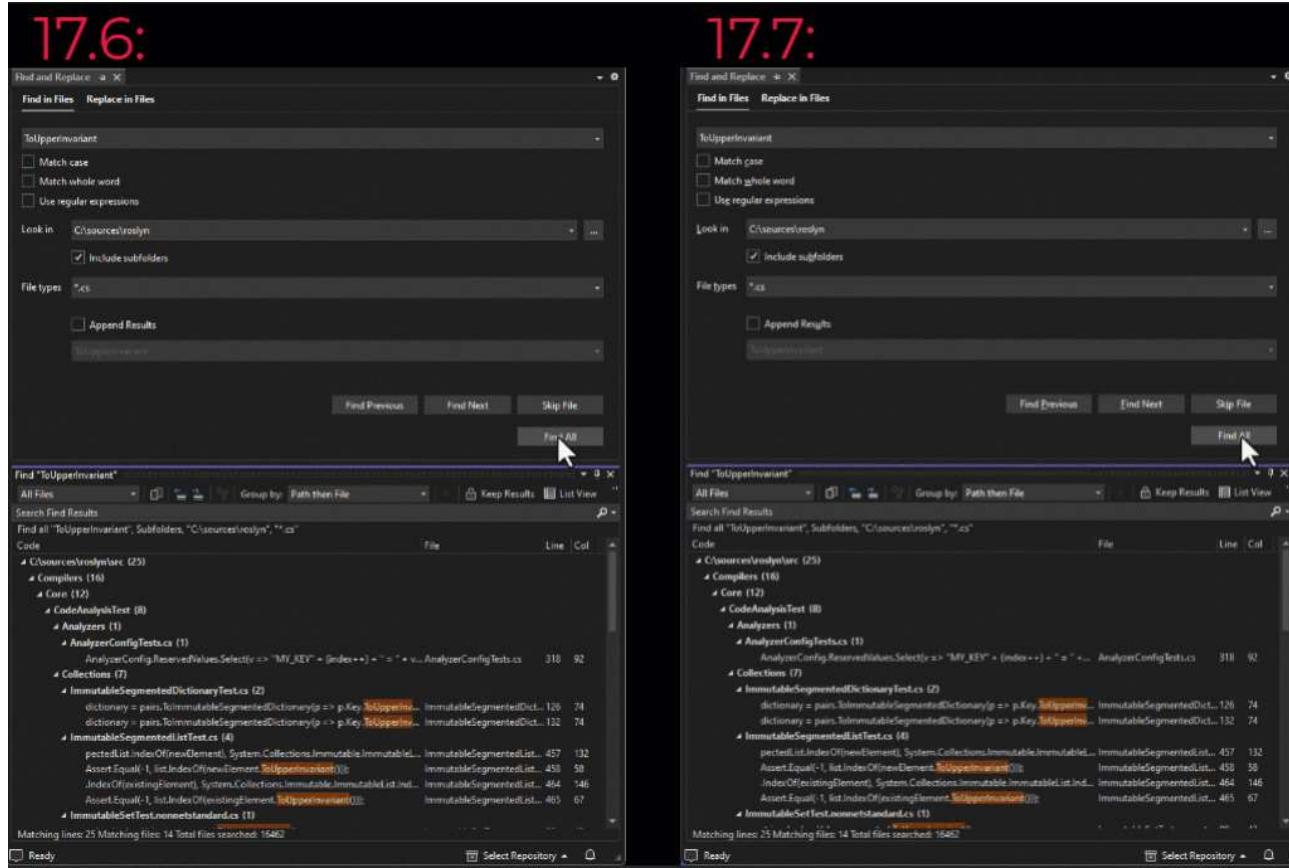
Several accuracy and performance enhancements are coming to All-in-One Search. For example, exact matches now show higher in the results list and results update more accurately in real time as you type your query.



Try out the All-In-One Search experience by making sure it is enabled in Tools > Manage Preview Features > "New Visual Studio Search experience (restart required)" and using the Ctrl + T or the Ctrl + Q keyboard shortcut to launch it. If you have feedback to share with us about this feature, let us know [here](#).

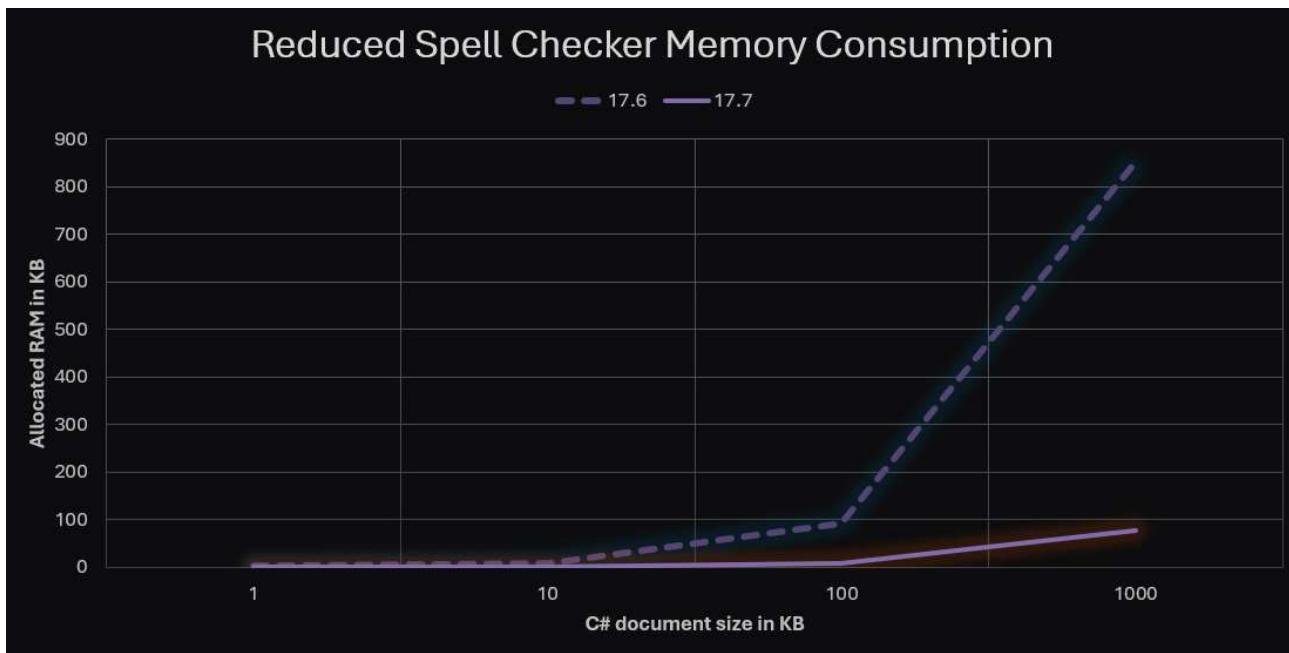
## Find in Files Search Time

Find in Files search time has been significantly improved with various optimizations, resulting in a remarkable ~50% reduction in search time!



## C# Spell Checker Memory Consumption

The LSP (Language Server Protocol) for the C# Spell Checker underwent significant improvements. By implementing a more efficient data structure and enabling streaming support, we achieved a remarkable 90% reduction in the memory footprint of the JSON transferred between the LSP Server and LSP Client. This enhancement leads to faster performance, increased efficiency and scalability for the spell-checking functionality.



## C++ Unreal Engine Time to IntelliSense

In Unreal Engine projects, the time for IntelliSense and colorization to become ready in newly opened C++ files is significantly reduced. The generation of IntelliSense cache (IPCH) is now 30% faster in Unreal Engine 5.1 and 5.2 projects and 15% faster in Unreal Engine 4.27 projects.



## Light Bulb Responsiveness

Performance enhancements to the light bulb feature in Roslyn have resulted in faster fixes for slow diagnostics like Fix Formatting and Simplify Type Name. Enjoy immediate suggestions and fixes, even in large documents or after making edits.

## Solutions Load CPU Consumption

Loading large solutions is now more efficient, with reduced CPU consumption thanks to optimized hash creation methods. This enhancement is particularly beneficial for machines

with multiple cores.

## .NET Development

### Auto-decompilation for External .NET Code

The Visual Studio Debugger supports Autodecompilation for external .NET code. Whenever you step into code that has been implemented externally, the debugger will automatically decompile it and display the point of execution.

This feature is particularly useful when analyzing call stacks, as you can double-click any stack frame and the debugger will navigate directly to the code. Additionally, you can debug the decompiled code and set breakpoints easily. All of the decompiled code is also shown under the External Sources node in Solution Explorer, making it easy to browse through the external files if needed.

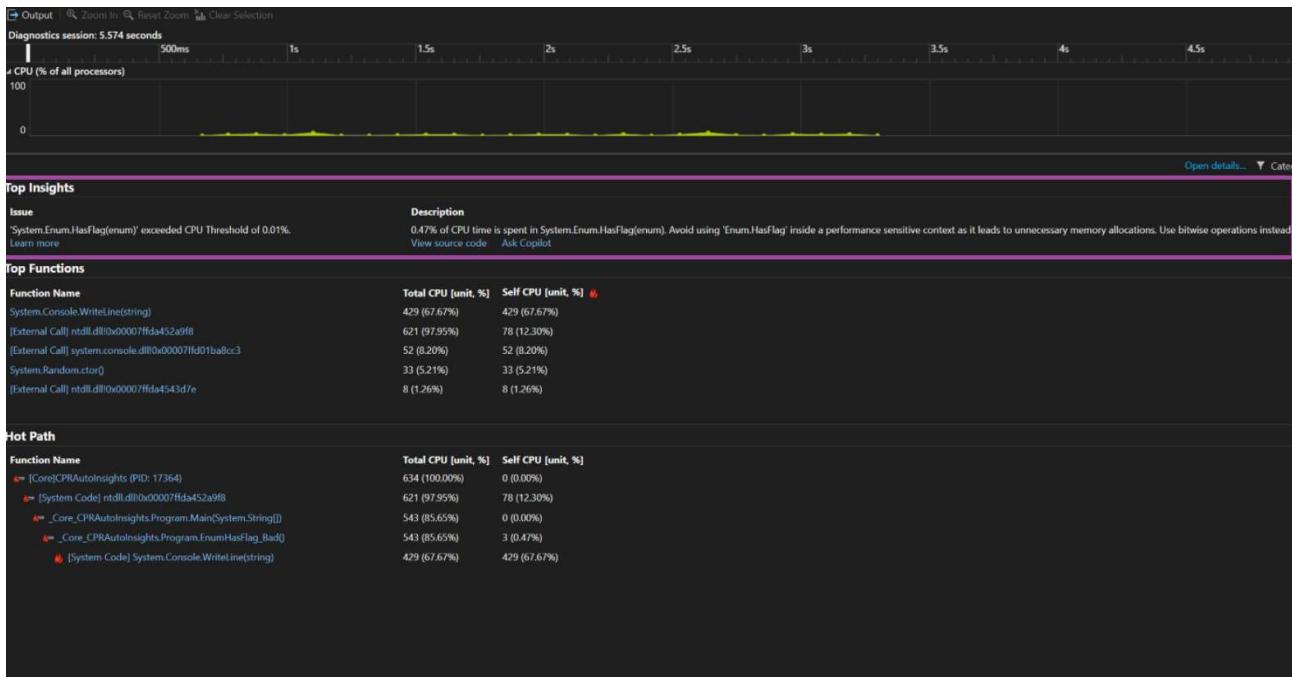
If you wish to disable the automatic decompilation of external code, simply clear the "Automatically decompile to source when needed (managed only)" option under Tools > Options > Debugging.



## New Auto Insights for CPU Usage tool

The CPU usage tool now provides additional insights and details for scenarios involving certain methods and properties, such as `Enum.HasFlag`, `Enum.ToString`, `String.StartsWith`, `String(char[])`, `ConcurrentDictionary.Count`, `ConcurrentDictionary.Keys/Values`, and `DateTime.Now`.

Additionally, there is now an 'Ask Copilot' option that uses the Copilot chat extension. When triggered, Copilot chat will explain why the issue occurred and suggest a fix.



## C++ and game development

### Unreal Engine Macros Formatting for C++

Visual Studio will detect and follow indentation and formatting rules for your Unreal Engine reflection macros. To turn on Unreal Engine specific formatting, search "Unreal Engine" in Options or go to Tools > Options > Text Editor > C/C++ > Code Style > Formatting > Unreal Engine and check "Apply special formatting for Slate declarative syntax and Unreal Engine reflection macros."

```
54     static FRPGGameplayEffectContainerSpec AddTargetsSlot()
55
56     /** Applies container spec that was made from an ability */
57     UFUNCTION(BlueprintCallable, Category = Ability)
58     static TArray<FActiveGameplayEffectHandle> ApplyExternalEffects();
59
60     //Returns the project version set in the 'Project Settings' panel
61     UFUNCTION(BlueprintPure, Category = "Project")
62     static FString GetProjectVersion();
63
64     SLATE_BEGIN_ARGS(SHeaderRow)
65         : _Style()
66     {}
67     SLATE_EVENT(FOnClicked, OnClicked)
68     SLATE_NAMED_SLOT(FArguments, FSimpleSlot, Content)
69     SLATE_DEFAULT_SLOT(FArguments, HeaderContent)
70     SLATE_END_ARGS()
71 }
```

## Build Insights in Visual Studio (C++)

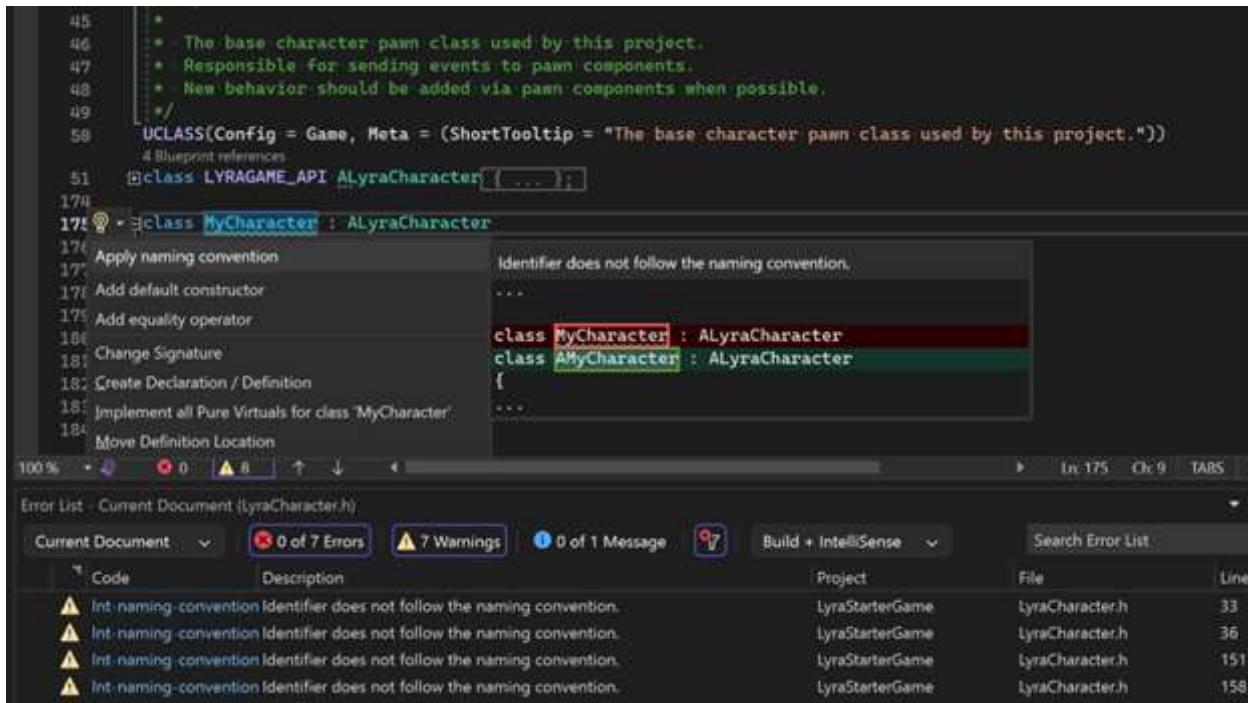
Thanks to your [feedback](#), we are happy to announce that [Build Insights](#) is now integrated with Visual Studio 2022! Build Insights provides you with valuable information needed when optimizing your C++ build times. Start your Build Insights .etl trace capture with the click of a button for your solution or projects. After compilation, Build Insights will create a diagnostic report that allows you to see expensive includes and navigate directly to header files. For power users, you can open Windows Performance Analyzer from within the report.

Included Files	Include Tree	Diagnostics Session: 1.936 seconds Build: 1.533 seconds			Filter Files
File Path		Time [sec, %]	Include Count	Project	
▶ 🔥 C:\src\TestConsoleApp\ConsoleApplication3\Console...		0.430 (28.0%)	1	ConsoleApplication3	
▶ 🔥 C:\Program Files\Microsoft Visual Studio\2022\Mai...		0.412 (26.9%)	2	ConsoleApplication3	
▶ 🔥 C:\Program Files\Microsoft Visual Studio\2022\...		0.396 (25.8%)	1	ConsoleApplication3	
▷ 🔥 C:\Program Files\Microsoft Visual Studio\2022...		0.392 (25.6%)	1	ConsoleApplication3	
◀ C:\Program Files\Microsoft Visual Studio\2022\Main...		0.014 (0.9%)	2	ConsoleApplication3	
◀ C:\Program Files\Microsoft Visual Studio\2022\M...		0.012 (0.8%)	2	ConsoleApplication3	
C:\Program Files\Microsoft Visual Studio\2022\...		0.007 (0.5%)	0	ConsoleApplication3	
◀ C:\Program Files\Microsoft Visual Studio\2022\...		0.002 (0.1%)	1	ConsoleApplication3	
C:\Program Files\Microsoft Visual Studio\202...		0.000 (0.0%)	0	ConsoleApplication3	
C:\Program Files\Microsoft Visual Studio\2022\...		0.000 (0.0%)	0	ConsoleApplication3	
▶ 🔥 C:\src\TestConsoleApp\ConsoleApplication2\Console...		0.200 (13.0%)	1	ConsoleApplication2	
▶ 🔥 C:\Program Files\Microsoft Visual Studio\2022\Mai...		0.196 (12.8%)	2	ConsoleApplication2	
▷ 🔥 C:\Program Files\Microsoft Visual Studio\2022\...		0.193 (12.6%)	1	ConsoleApplication2	
◀ C:\Program Files\Microsoft Visual Studio\2022\Main...		0.002 (0.1%)	2	ConsoleApplication2	
▷ C:\Program Files\Microsoft Visual Studio\2022\...		0.001 (0.1%)	2	ConsoleApplication2	
C:\Program Files\Microsoft Visual Studio\2022\Main...		0.000 (0.0%)	0	ConsoleApplication2	
▶ 🔥 C:\src\TestConsoleApp\ConsoleApplication1\Console...		0.186 (12.1%)	1	ConsoleApplication1	
▶ 🔥 C:\Program Files\Microsoft Visual Studio\2022\Mai...		0.183 (11.9%)	2	ConsoleApplication1	
▶ 🔥 C:\Program Files\Microsoft Visual Studio\2022\...		0.180 (11.7%)	1	ConsoleApplication1	
▷ 🔥 C:\Program Files\Microsoft Visual Studio\2022...		0.176 (11.5%)	1	ConsoleApplication1	
◀ C:\Program Files\Microsoft Visual Studio\2022\Main...		0.002 (0.1%)	2	ConsoleApplication1	
◀ C:\Program Files\Microsoft Visual Studio\2022\...		0.001 (0.1%)	2	ConsoleApplication1	

[Open in WPA](#)

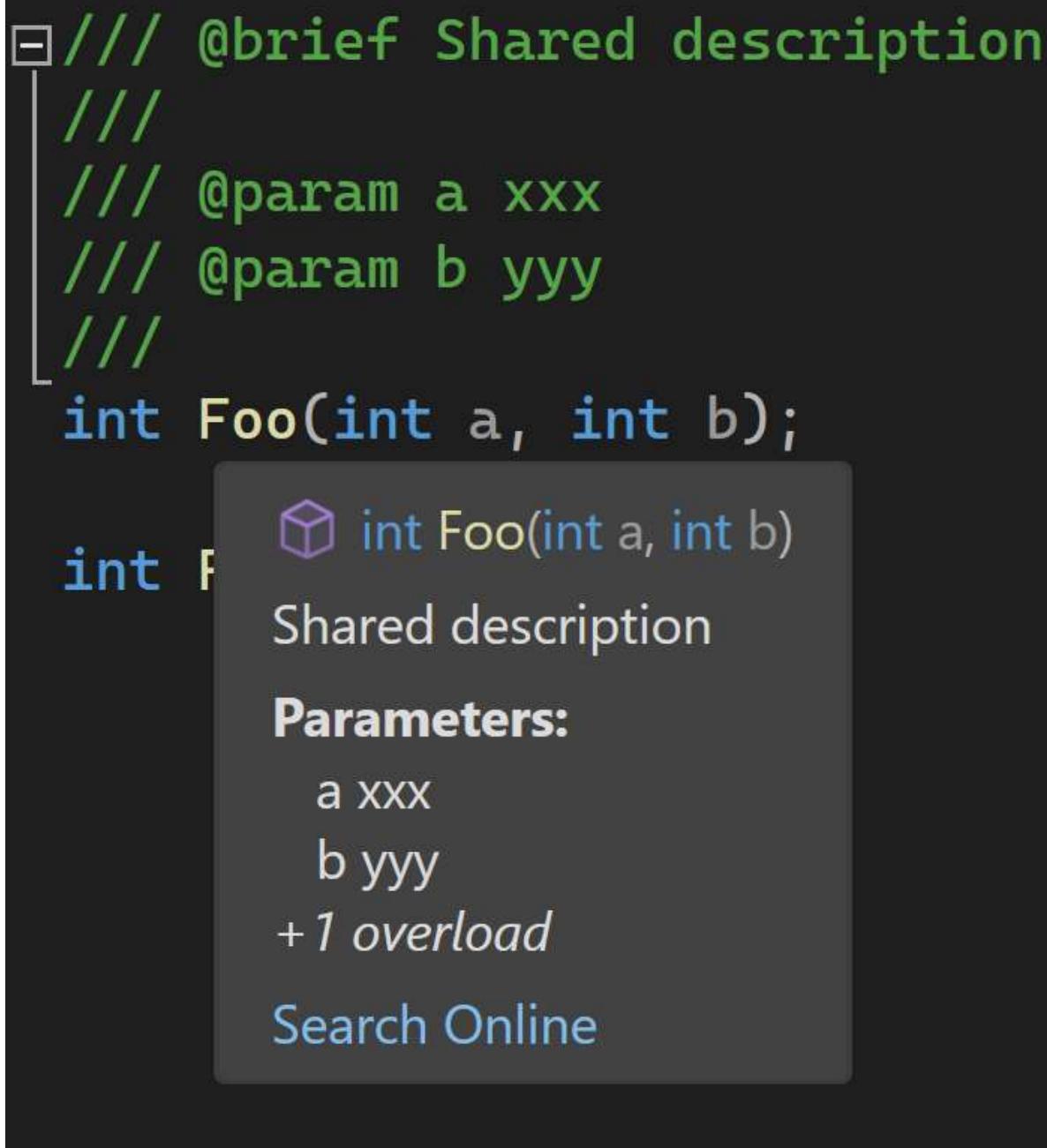
## Unreal Engine Naming Convention Checker for C++

With the Unreal Engine Naming Convention checker, Visual Studio will now give you hints when your symbols are not conformant to the Unreal Engine coding style. After adding an [.editorconfig file](#) to your Unreal Engine project, Visual Studio will display naming convention warnings in the Error List and provide a Quick Fix option upon right-clicking.



## Improved Doxygen overload resolution

Previously, when a specific function had a Doxygen comment, it did not appear in the Quick Info for each overload except for the one immediately following the comment. We now allow the documentation from another overload to be displayed in the Quick Info for all associated overloads.



## Unreal Engine Blueprint Find All References

You can now search for references of `UFUNCTION` in Unreal Engine Blueprints with the new Blueprint Find All References. To start, right-click on a symbol with Blueprint references. Then, click "Find All Blueprint References" to start the search. The latest version of the Unreal Engine plugin "[Visual Studio Integration Tool](#)" is required.

Blueprint References - ULyraGameplayAbility::GetLyraPlayerControllerFromActorInfo - Entire solution

The screenshot shows the 'Blueprint References' tool in Visual Studio. The title bar says 'Blueprint References - ULyraGameplayAbility::GetLyraPlayerControllerFromActorInfo - Entire solution'. The interface includes a dropdown for 'Entire solution', search and filter icons, and a 'Group by: Project then Definition' dropdown. Below the title bar are tabs for 'Code' and 'File'. The main area displays a hierarchical tree of references:

- LyraStarterGame (6)**
  - GAB\_ShowWidget\_WhenInputPressed\_C (1)
    - GAB\_ShowWidget\_WhenInputPressed\_C
  - GAB\_ShowWidget\_WhileInputHeld\_C (1)
    - GAB\_ShowWidget\_WhileInputHeld\_C
  - GA\_ADS\_C (1)
    - GA\_ADS\_C
  - GA\_Grenade\_C (1)
    - GA\_Grenade\_C
  - GA\_WeaponNetShooter\_C (1)
    - GA\_WeaponNetShooter\_C
  - GA\_Weapon\_Fire\_C (1)

On the right side of the tree, file names are listed: GAB\_ShowWidget.h, GAB\_ShowWidget.cpp, GA\_ADS.uasset, GA\_Grenade.uasset, GA\_WeaponNetShooter.h, and GA\_WeaponNetShooter.cpp.

At the bottom of the tool window, there are tabs for 'Blueprint References', 'CodeLens', and 'Error List'.

## Address Sanitizer continue\_on\_error Support

We have expanded our Address Sanitizer support with a new `continue_on_error` mode. This new runtime mode diagnoses and reports hidden memory safety errors, with zero false positives, as your app runs. Output to `stdout` with set

`ASAN_OPTIONS=continue_on_error=1`, or to `stderr` with set

`ASAN_OPTIONS=continue_on_error=2`. See [AddressSanitizer continue\\_on\\_error](#) for more details.

```
== Files in priority order ==
File: c:\msvc\secure_coincidence.cpp Unique call stacks: 2
== Source Code Details: Unique errors caught at instruction offset from source line number, in functions, in the same file. ==
File: c:\msvc\secure_coincidence.cpp
  Func: func()
    Line: 6 Unique call stacks (paths) leading to error at line 6 : 2
          Bug: heap-buffer-overflow at instr 173 bytes from start of line
>>>Total: 2 Unique Memory Safety Issues (based on call stacks not source position) <<
#0 c:\msvc\secure_coincidence.cpp Function: func(Line:6)
  Raw HitCnt: 1 On Reference: 1-byte-read-global-buffer-overflow
#1 c:\msvc\secure_coincidence.cpp Function: func(Line:6)
  Raw HitCnt: 1 On Reference: 1-byte-write-heap-buffer-overflow
```

## Include Cleanup

Added new "Include Cleanup" feature. This provides suggestions to add direct includes whenever indirect includes are detected, and identifies any unused includes. This feature is disabled by default. To enable it, navigate to Tools > Options > Text Editor > C/C++ >

IntelliSense and select "Enable #include cleanup". You can learn more about this feature in the [Include Cleanup blog post](#).

The screenshot shows a code editor window with C++ code. The code includes several #include directives for Windows-specific headers like `<atlbase.h>`, `<atlcom.h>`, and `<atlcomcli.h>`. A tooltip is displayed over the `#include <string>` directive, containing two options: "Remove #include <new>" and "Remove all unused includes". The "Remove all unused includes" option is highlighted with a mouse cursor.

```
#include <iostream>

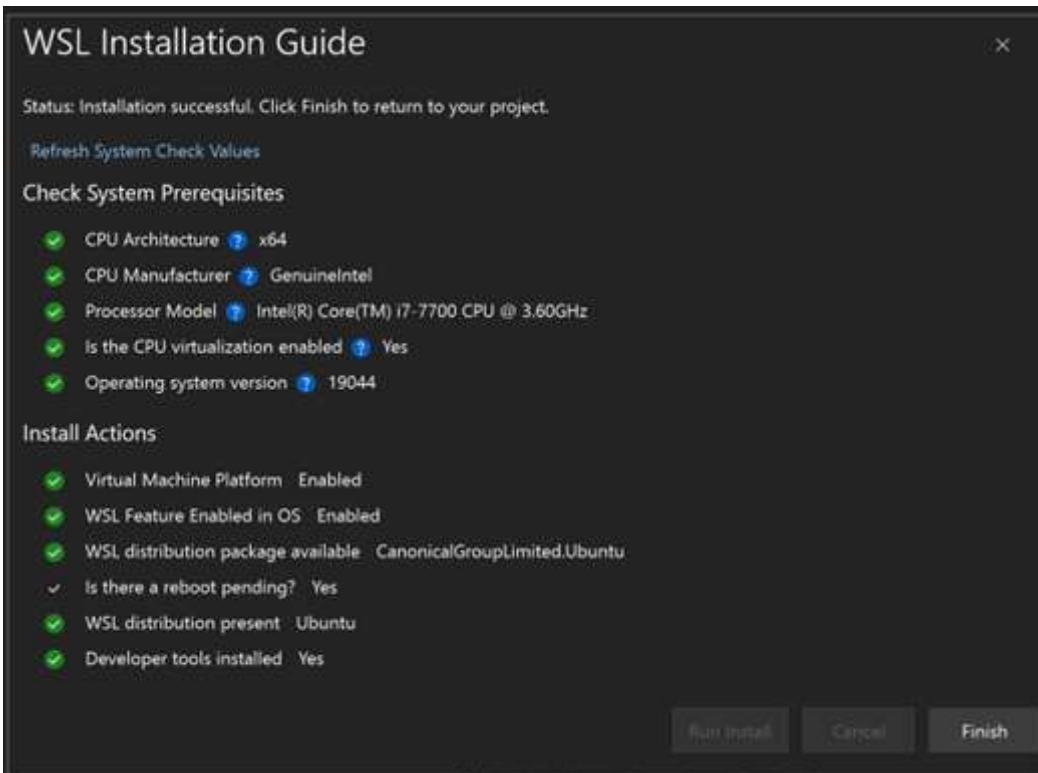
#include <atlbase.h>
#include <atlcom.h>
#include <atlcomcli.h>

#include <processsthreadsapi.h>
#include <minwindef.h>
#include <new>
#include <string>
```

## Linux and embedded development with C++

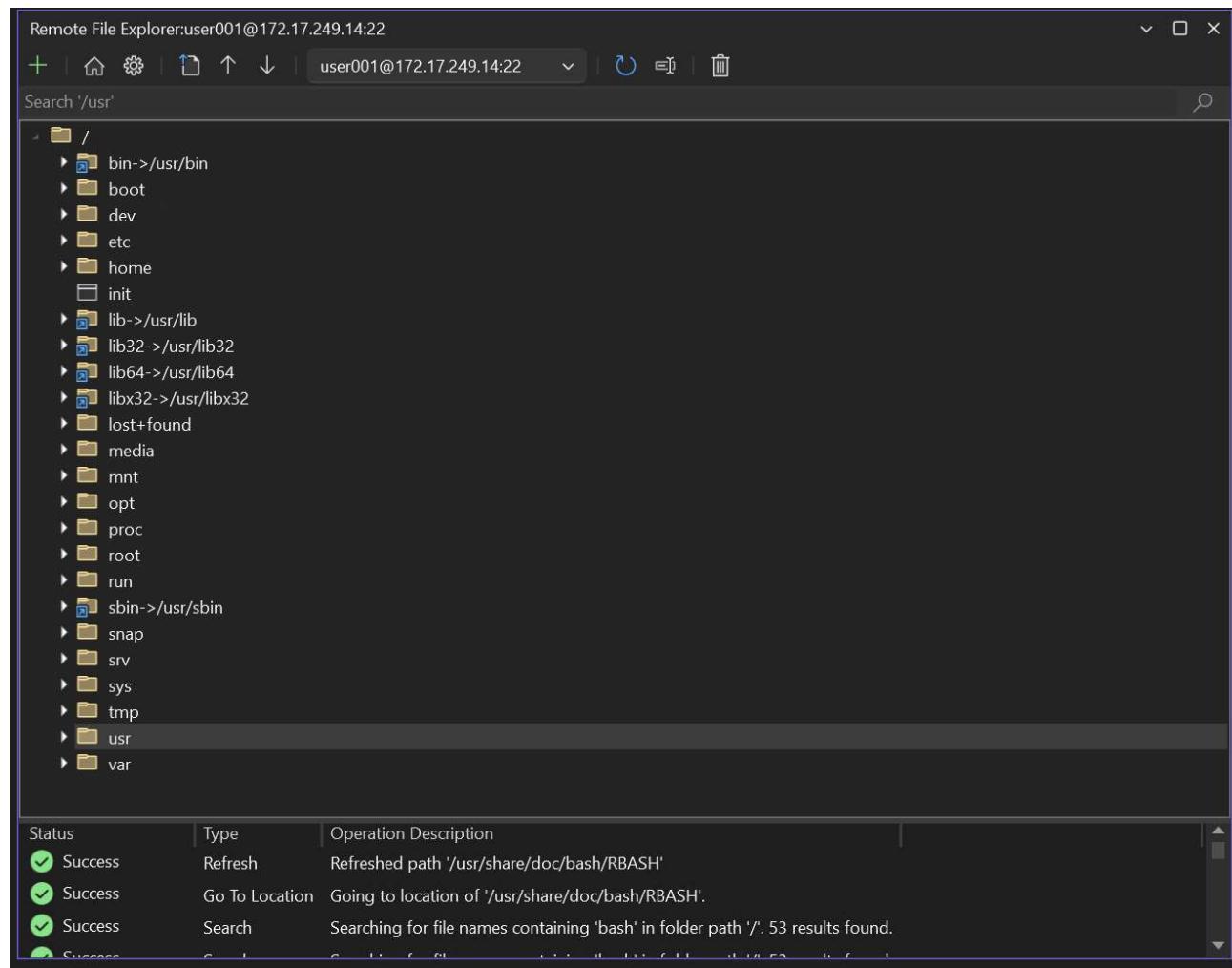
### WSL Automatic Acquisition

One-click downloads of WSL are now available with the Linux and Embedded Workload. When you open a CMake project that can target Linux, a gold bar will open for you to click and open a WSL installation process. By default, it will install an Ubuntu machine. Additionally, you can open the dialog by navigation at any time to Project > Install WSL for Me.



## Remote File Explorer Updates

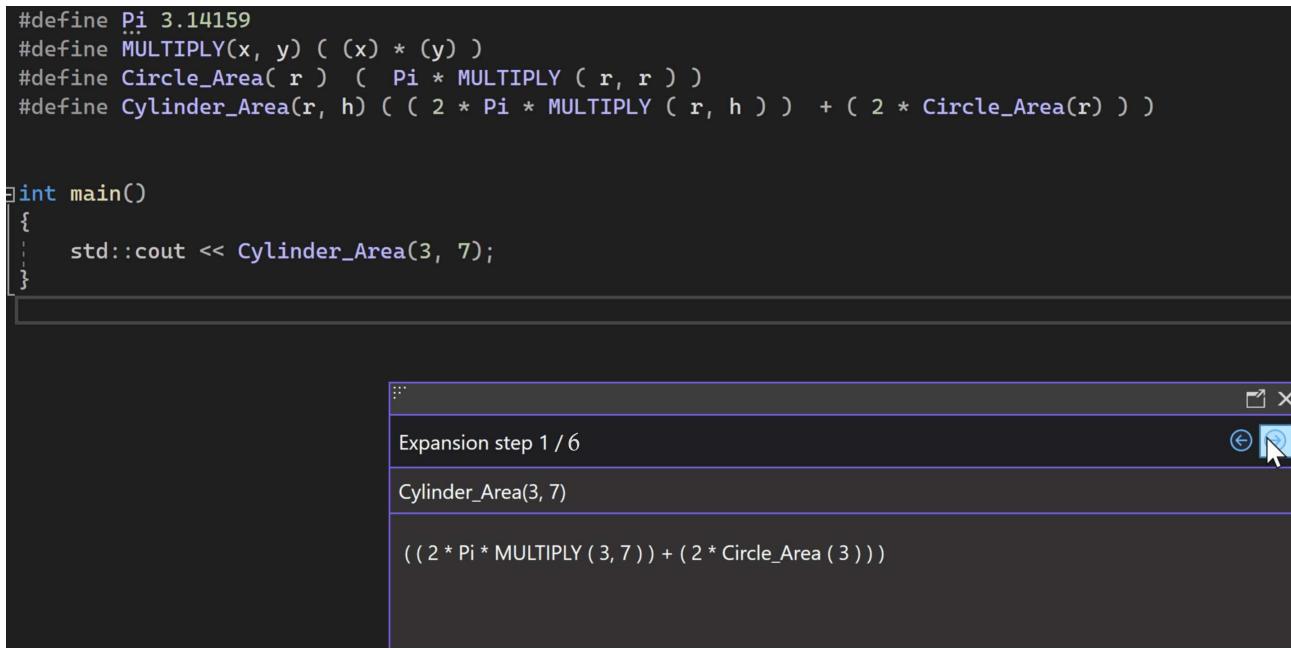
We have added a search capability to the Remote File Explorer. You are now able to search for files based on where you are in your remote machine along the top bar, and then right-click to go to any found file's location. Additionally, we have added two new buttons to the toolbar to improve the Remote File Explorer experience, one to go to the home directory and one to open tools options.



## Other C++ updates

- Added support for CMakePresets.json version 6. See the [CMake documentation](#) for details on what is available in this version.
- Upgraded the version of LLVM shipped with Visual Studio to version 16. See the [LLVM](#) and [Clang](#) release notes for what is available in this version.
- Added new intrinsics for overflow detection, such as `_add_overflow_i32`.
- The Connection Manager API is now publicly exposed for programmatically managing your connections.
- We have changed the linking model used by ASan. Instead of each runtime library setting (`/MD`, `/MT`, `/MDd`, `/MTd`) having its own way of providing the ASan runtime, all of those settings now depend on single DLL: `clang_rt.asan_dynamic-<arch>.dll`. This means that programs that load binaries that link to the C Runtime differently will be able to be run under ASan without any compatibility issues, and `/MT(d)` DLLs no longer need the host EXE to be built with ASan in order to run. Note that this means

- /MT(d) binaries now also rely on the `clang_rt.asan_dyanmic-<arch>.dll` being present on path for that program, so testing environments will need to be updated.
- You can now use the Visualize Expansion option on macros to view the expansion of nested macros step-by-step.



The screenshot shows a code editor with the following C++ code:

```
#define Pi 3.14159
#define MULTIPLY(x, y) ( (x) * (y) )
#define Circle_Area( r ) ( Pi * MULTIPLY ( r, r ) )
#define Cylinder_Area(r, h) ( ( 2 * Pi * MULTIPLY ( r, h ) ) + ( 2 * Circle_Area(r) ) )

int main()
{
    std::cout << Cylinder_Area(3, 7);
}
```

A tooltip window titled "Expansion step 1 / 6" is displayed over the call to `Cylinder_Area(3, 7)`. The tooltip contains the expanded code:

```
(( 2 * Pi * MULTIPLY ( 3, 7 ) ) + ( 2 * Circle_Area ( 3 ) ))
```

The tooltip has a close button (X) and a refresh/circular arrow icon in the top right corner.

## Web Development:

### npm Dependency Management in ASP.NET projects

In your ASP.NET projects that leverage npm packages, Visual Studio will now show those packages from your `package.json` in the Dependencies node in Solution Explorer. This now enables you to get a clear view of all your solution dependencies in one place for .NET with references, NuGet packages, and now npm packages. You can add new packages or restore (`npm install`) all packages by right clicking on the npm node.

Solution Explorer

Search Solution Explorer (Ctrl+;)

Solution 'Project3' (1 of 1 project)

Project3

- Connected Services
- Dependencies
- Analyzers
- Frameworks
- npm
  - ajv (8.12.0)
  - bootstrap (5.2.3)
  - cross-env (7.0.3)
  - eslint (8.39.0)
  - eslint-config-react-a
  - eslint-plugin-flowtyp
  - eslint-plugin-import
  - eslint-plugin-jsx-a11
  - eslint-plugin-react (7)
  - http-proxy-middlew
  - jquery (3.6.4)
  - merge (2.1.1)
  - nan (2.17.0)
  - oidc-client (1.11.5)
  - react (18.2.0)
  - react-dom (18.2.0)
  - react-router-bootstrap (0.26.2)
  - react-router-dom (6.11.0)
  - react-scripts (5.0.1)
  - reactstrap (9.1.9)
  - rimraf (5.0.0)
  - typescript (4.9.5)
  - web-vitals (3.3.1)
  - workbox-background-sync (6.5.4)
  - workbox-broadcast-update (6.5.4)
  - workbox-cacheable-response (6.5.4)
  - workbox-core (6.5.4)
  - workbox-expiration (6.5.4)
  - workbox-google-analytics (6.5.4)
  - workbox-navigation-preload (6.5.4)
  - workbox-precache (6.5.4)

Install New npm Packages

Restore Packages

Configure External Tools...

Open package.json

Open In File Explorer

Code Cleanup >

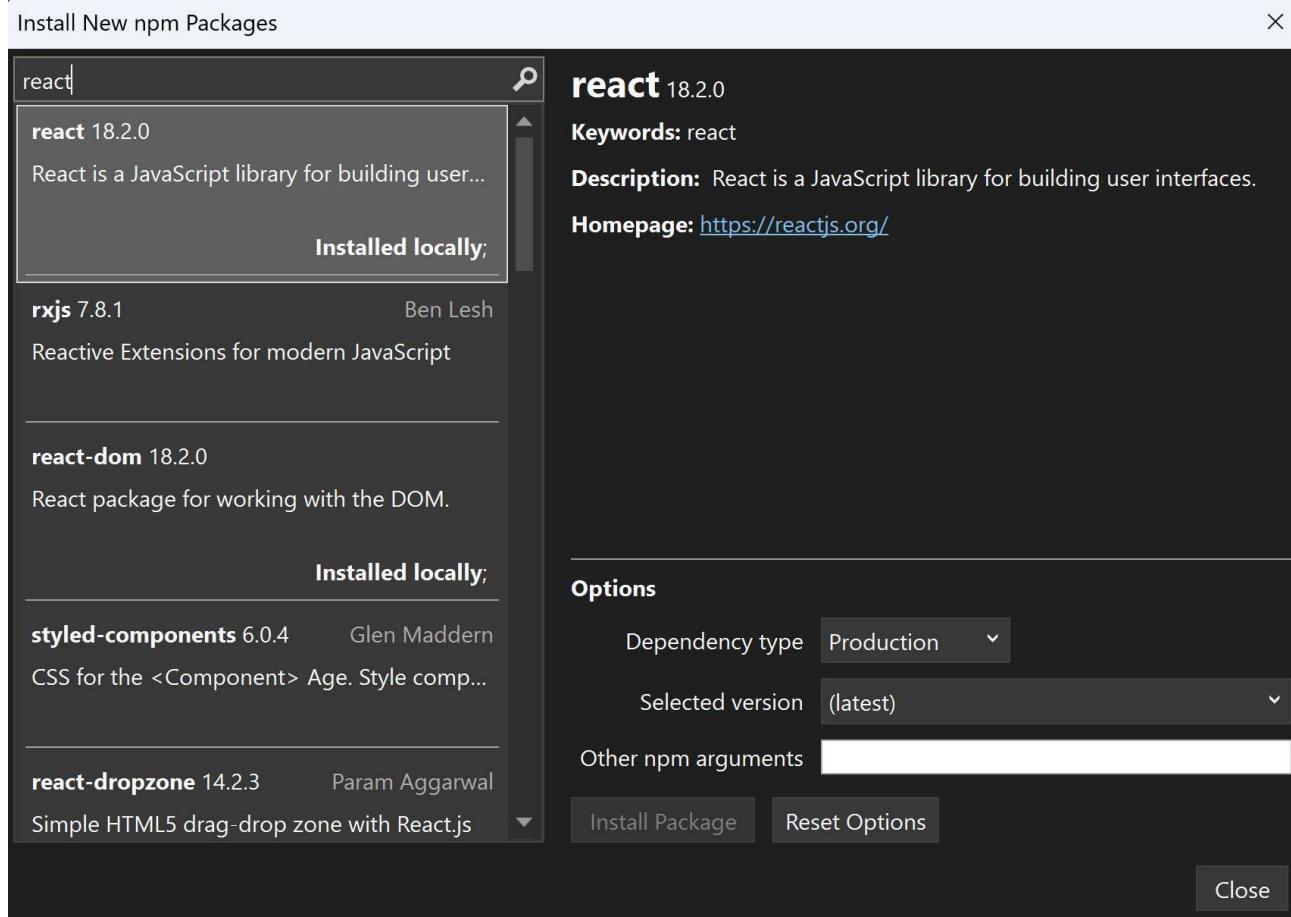
Collapse All Descendants Ctrl+Left Arrow

Scope to This

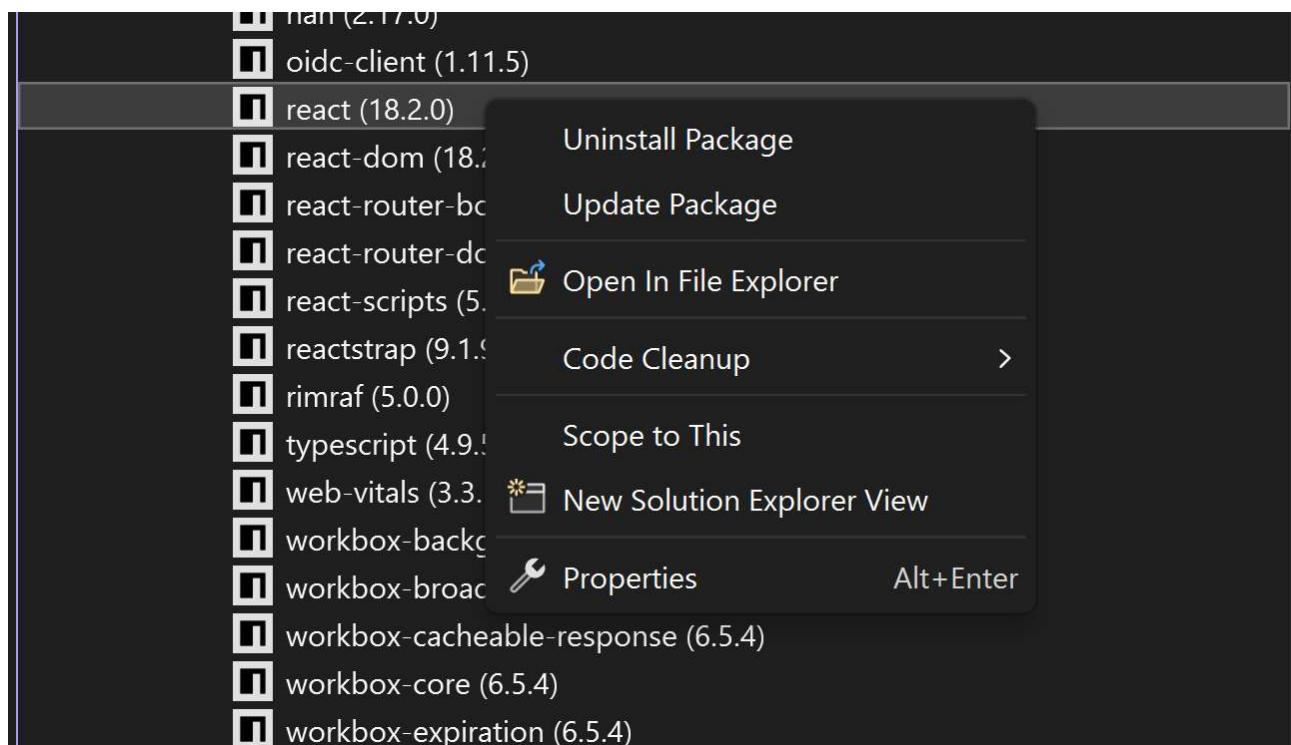
New Solution Explorer View

Properties Alt+Enter

You can search and select which package you would like to add using the UI.



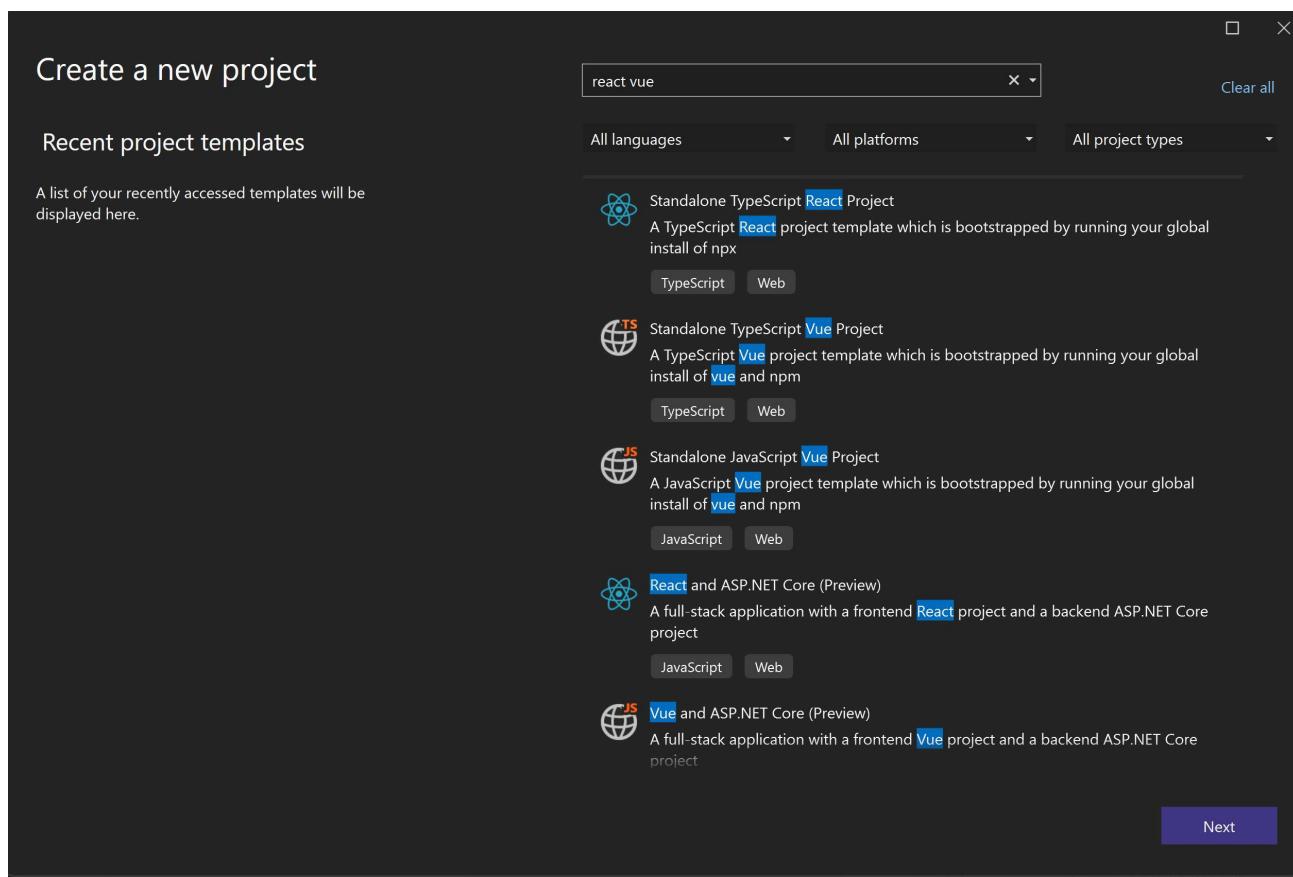
You can update or uninstall a specific package by right clicking on it.



## Supporting Vite for React and Vue new project creation

When we [announced](#) the JavaScript/TypeScript and .NET combined project templates in Visual Studio, we received community feedback that both `create-react-app` and the `vue cli` are no longer the preferred way of creating new applications. We now utilize [Vite](#) for both React and Vue new project creation for our JavaScript and TypeScript project templates in Visual Studio. Project build and run will also benefit from Vite, such as faster server start and updates.

This change also gives a performance boost to the process, resulting in a much faster time to create these types of projects. It also addresses the creation failure problem when you don't have the necessary dependencies installed locally on your machine.



## HTTP Editor

- We've made updates to the [HTTP Editor](#) including a new Response View. The new Response View includes support for JSON highlighting as well as the ability to view; the raw response, the headers of the request and the request as it was sent to the web server.
- The green play button to send the request was replaced by code lens actions.

## Power Platform connected services

- We've added [Connected Services support for Microsoft Power Platform](#).
- You can create a custom connector to your [Power Platform environment](#) and create a [dev tunnel](#) to locally test and debug your Web API project.

## F#

- Added inline return type hints:

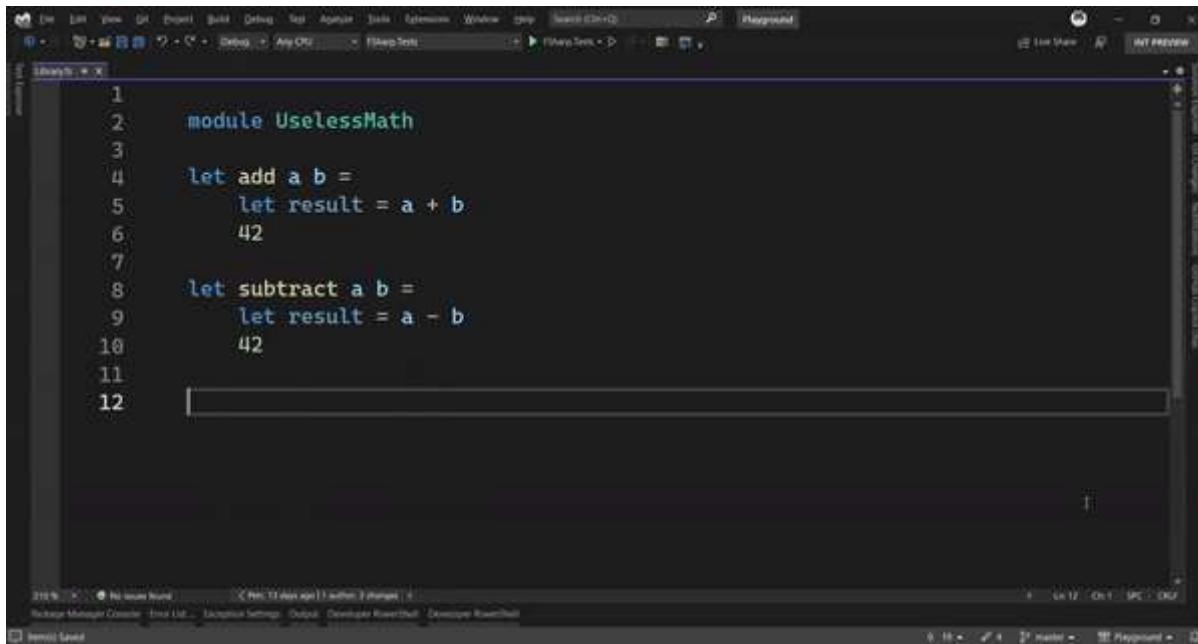
```
let add x y : int = x + y

let getTheAnswer() : int = 42

let setConsoleOut : unit = System.Console.SetOut
```

Since they are in preview and disabled by default. Enable them via **Tools -> Options -> Text Editor -> F# -> Advanced -> Display return type hints**

- Added new bulk code fixes:



See the full list of these code fixes in [#15082](#)

- Resolved an issue with parameter name hints crashing when using multi-line arguments

- Added an option to customize Quick Info description width: **Tools -> Options -> Text Editor -> F# -> QuickInfo -> Preferred description width in characters**. See more details in [#14964](#)
- Incorporated activity log into the output pane (for more information, refer to [#15028](#) )
- Fixed bulk code fix for removing unused opens
- Improved the speed of Quick Info link navigation
- Added support for the new interpolated strings syntax with multiple dollar signs ([details and RFC](#) ):

```

105
106 [;<HookComponent>]
107 let TodoEl dispatch (edit: Todo option): Todo option = 
108   Hook.useHmr(token = hmr)
109   let transitionMs: int = 500
110   let className: string = Hook.use_scoped_css rules = $$"""
111   :host {
112     transition-duration: {{transitionMs}}ms;
113     border: 2px solid lightgray;
114     border-radius: 10px;
115     margin: 5px 0;
116   }
117   :host.transition-enter {
118     opacity: 0;
119     transform: scale(2);
120   }
121   :host.transition-leave {
122     opacity: 0;
123     transform: scale(0.1);
124   }
125   .is-clickable {
126     user-select: none;
127   }
128 """
129

```

- Introduced tooltips for inline hints:

#### A- Type hints:

```

type Song = { Artist: string; Title: string }
let whoSings song: Song : string = song.Artist
let artist: string = whoSings song = { Artist = "U2"; Title = "One" }

```

#### B- Parameter name hints:

```
type Song = { Artist: string; Title: string }
let whoSings song : string = song.Artist

let artist: string = whoSings song = { Artist = "U2"; Title = "One" }

parameter song
```

- Fixed parameter name hints for nested function calls ([#15156](#)) and custom operations ([#15119](#))
- Fixed return type hints for cases where annotations are present alongside generic or type parameters ([#15124](#))

## SSDT (SQL)

### Support for Serverless SQL Pool in SSDT

We are providing the capability of importing a database from Serverless SQL Pool into SSDT. It can be done by the same process by which a regular database is imported.

Import Database

?

X



# Import Database

Source database connection:

cicd-syn-dev-on-demand.sql.azuresynapse.net.SQLODPOOLDEV (sqladmin)

Select Connection...

Target project

Name:

Location:

Browse... Create new solution Create directory for solution Add to source control

## Import Settings

 Import application-scoped objects only Import referenced logins Import permissions Import database settings

Folder structure:

Schema\Object Type

Maximum files per folder:

1000

StartCancel

Import Database

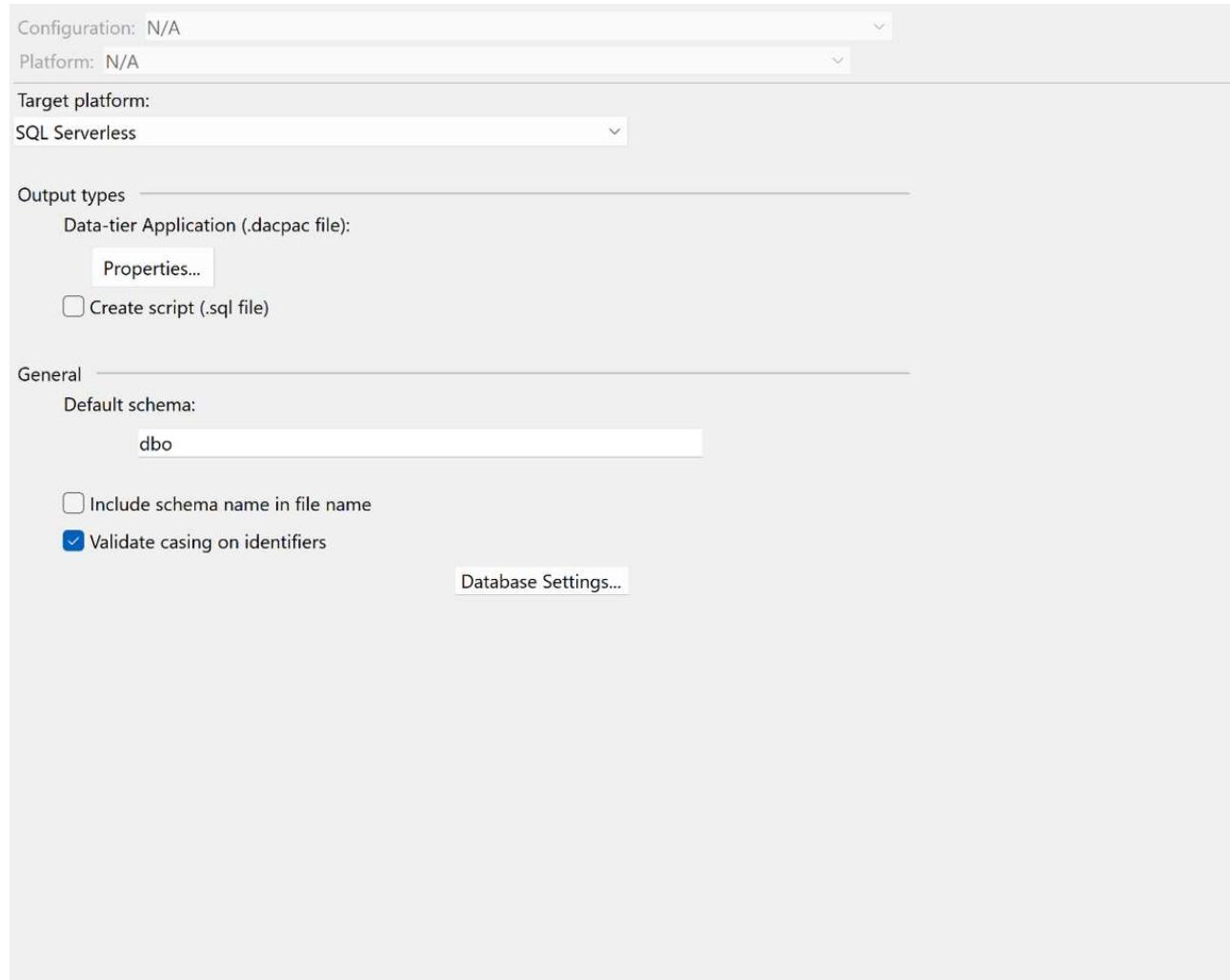
## Summary

Status: Click Finish to continue...

6/7/2023 1:01:26 AM	Import of database schema has started.
6/7/2023 1:01:38 AM	Gathering logins
6/7/2023 1:01:38 AM	Gathering database credentials
6/7/2023 1:01:38 AM	Gathering database options
6/7/2023 1:01:38 AM	Gathering generic database scoped configuration option
6/7/2023 1:01:39 AM	Gathering users
6/7/2023 1:01:39 AM	Gathering roles
6/7/2023 1:01:39 AM	Gathering application roles
6/7/2023 1:01:39 AM	Gathering role memberships
6/7/2023 1:01:39 AM	Gathering symmetric keys
6/7/2023 1:01:39 AM	Gathering encrypted symmetric keys
6/7/2023 1:01:39 AM	Gathering schemas
6/7/2023 1:01:39 AM	Gathering functions
6/7/2023 1:01:39 AM	Gathering encrypted functions
6/7/2023 1:01:39 AM	Gathering procedures
6/7/2023 1:01:39 AM	Gathering encrypted procedures
6/7/2023 1:01:39 AM	Gathering views
6/7/2023 1:01:39 AM	Gathering encrypted views
6/7/2023 1:01:39 AM	Gathering statistics
6/7/2023 1:01:39 AM	Gathering defaults
6/7/2023 1:01:39 AM	Gathering data constraint uddt bindings
6/7/2023 1:01:39 AM	Gathering external data sources
6/7/2023 1:01:39 AM	Gathering external file formats
6/7/2023 1:01:39 AM	Gathering external tables
6/7/2023 1:01:43 AM	Gathering logins

**Finish** **Cancel**

Users will also be able to publish a whole new project in the Serverless SQL Pool. The target platform needs to be set as *SQL Serverless*



Regular SSDT operations like Schema Compare, Data Compare, Addition of external tables can be performed on these tables.

- Fixed issue with publish using Azure Interactive Dir to an Azure Debugger.
- Changed the Nomenclature of SQL Serverless in the Target Platform to Azure Synapse Analytics Serverless SQL Pool.

## Installation and Updates

### Rollback

Visual Studio supports the ability to return to your previously installed version. Rolling back will return Visual Studio to the version installed prior to the last applied update, but it will keep any modifications to selected components chosen since that last update. Your common synchronized settings such as themes, fonts, key bindings, and windows layout will be restored. Similarly, your roamed settings will always be restored. A recent addition

to this feature is that now your local customizations such as your user settings will be restored only if you are rolling back to version 17.7 Preview 3 or higher. For more information, please visit the [Rollback blogpost](#).

## Enterprise Management and Acquisition

### Ability for standard users to update and modify Visual Studio

This version contains a new feature that will help enterprises adhere to security best practices while enabling their developers to update and modify their development environment. Administrators can now delegate installer functionality to standard users. So, if an administrator follows security best practices and restricts user permissions, then the administrator can now grant standard users access to installer functionality, thus allowing developers to update and modify the product.

To enable this configuration, an administrator will need to perform two simple steps on the client machine: install the latest 17.7 installer, and set the new global policy.

- You can acquire the latest installer on the client machine by updating the product.
- Set the global policy '[AllowStandardUserControl](#)' REG\_DWORD.
  - A value of 1 enables just update and rollback.
  - A value of 2 enables all installer functionality, including modify and install from Available tab.

This functionality applies to all versions Visual Studio 2017 and greater.

[Many enterprise users have requested this capability](#) and we hope that our solution meets your needs. [Please try it out and let us know what you think!](#)

## Microsoft 365 development

### Microsoft Teams development tools (Teams Toolkit)

We have fully tested the preview features in 17.6 Preview channel and now ship the preview features officially in 17.7 GA version. Including:

- Teams Toolkit update the Teams app templates project structure to make it more transparent and flexible to developers.

- In the new project, we introducing a new config file `teamsapp.yml`, where users can see what actions Teams Toolkit performs behind the scenes when lifecycle commands like debug, provision and deploy etc are triggered. Users can clearly see the actions being conducted by Teams Toolkit and can customize the actions by editing the `yml` file to change the Teams Toolkit behavior.
- Now you can use VS native Dev Tunnel feature to create tunneling service for Teams app. NgRok is no longer the only way to create tunnel for Teams projects.
- TeamsFx .NET SDK updates
  - Support to get a paginated list of targets where the bot is installed in notification bot.
  - Support to get a paginated list of members of one-on-one, group, or team conversation in notification bot.
- Simplified Teams Tab app template is provided in this version.
- Several bug fix and UI improvements to ensure better user experience on Teams Toolkit