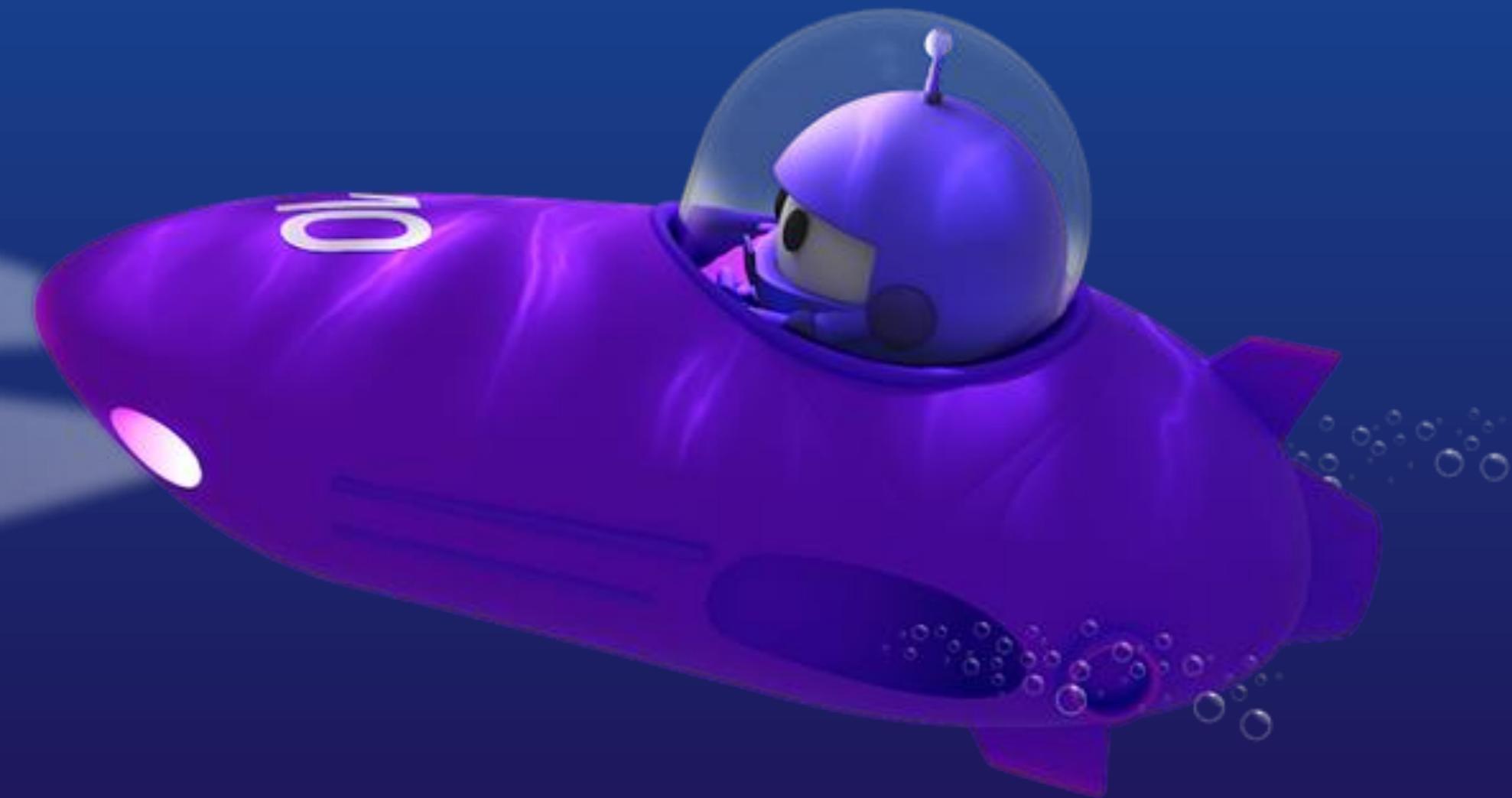


Встречайте
Visual Studio

2026



Visual Studio 2026

Быстрая, удобная, умная



Высокая
~~предусмотренная для загрузки~~
скорость загрузки решений
• Быстрый переход к отладке



Удобство работы

- Новый UI
- Покрытие тестами
- Inline в режиме отладки



AI работающий с вами

- Агенты
- MCP

Время загрузки решения

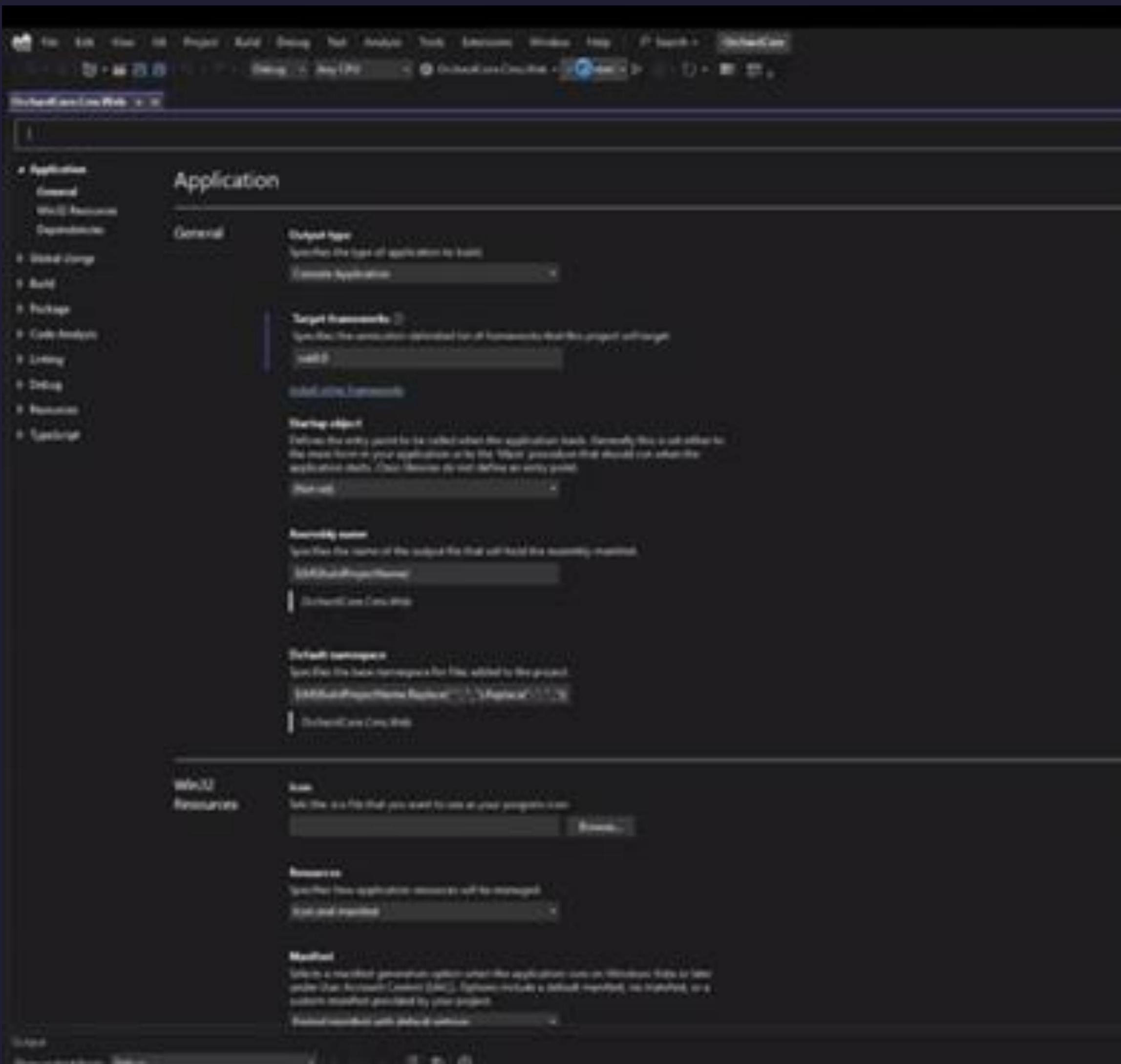
223 проекта



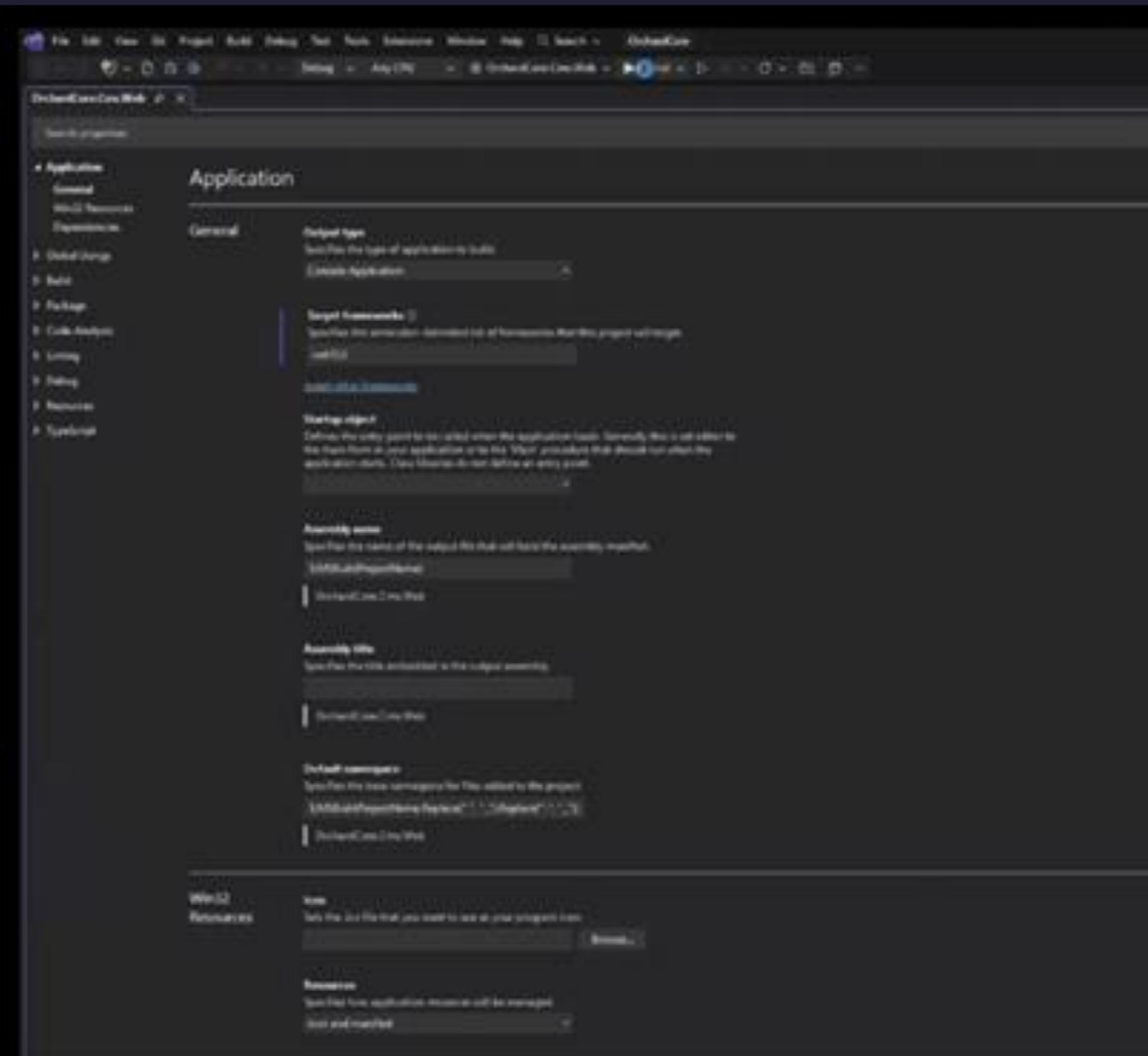
Тестировалось на Orchard Core:
github.com/OrchardCMS/OrchardCore

Переход в отладочный режим

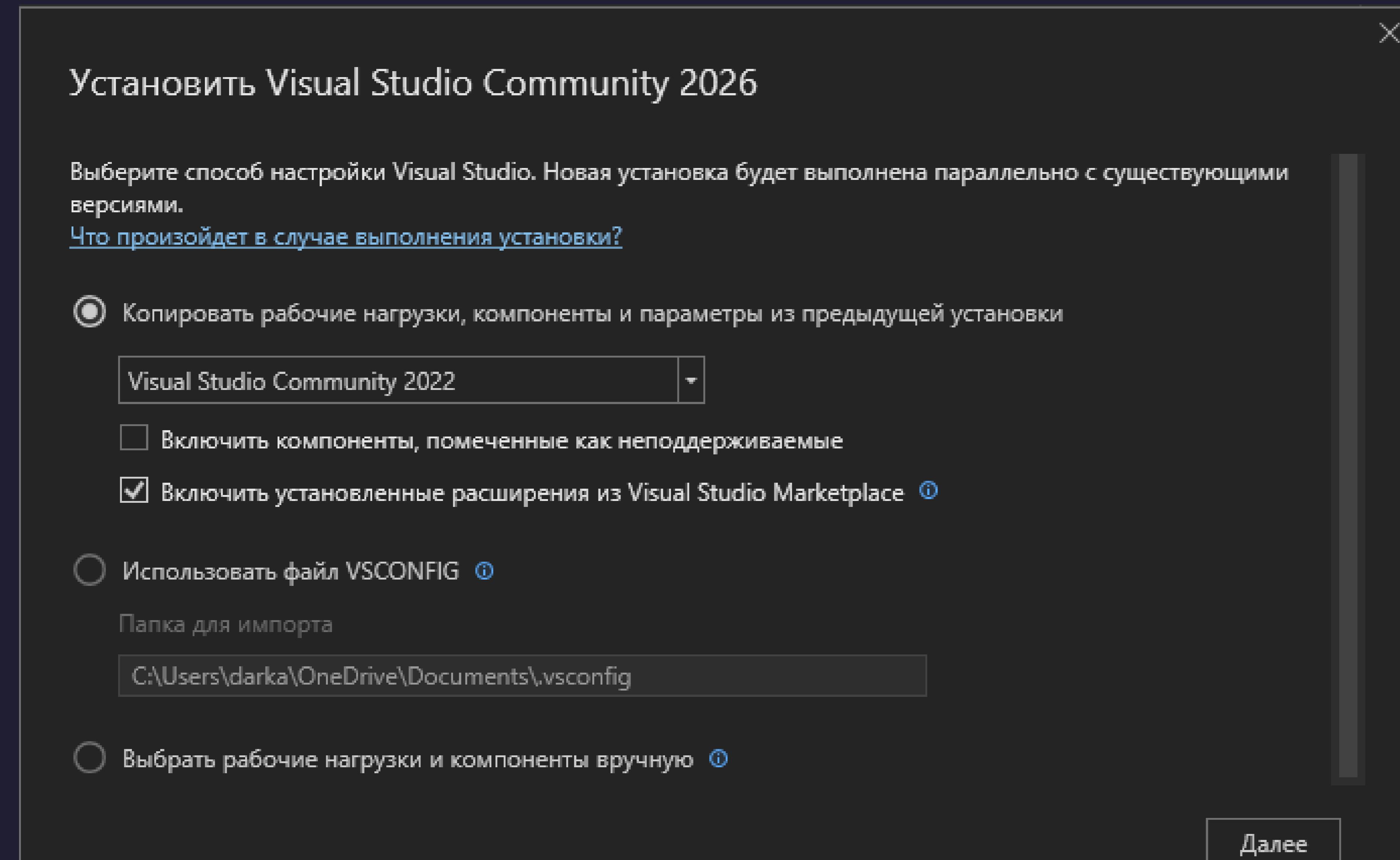
.NET 9



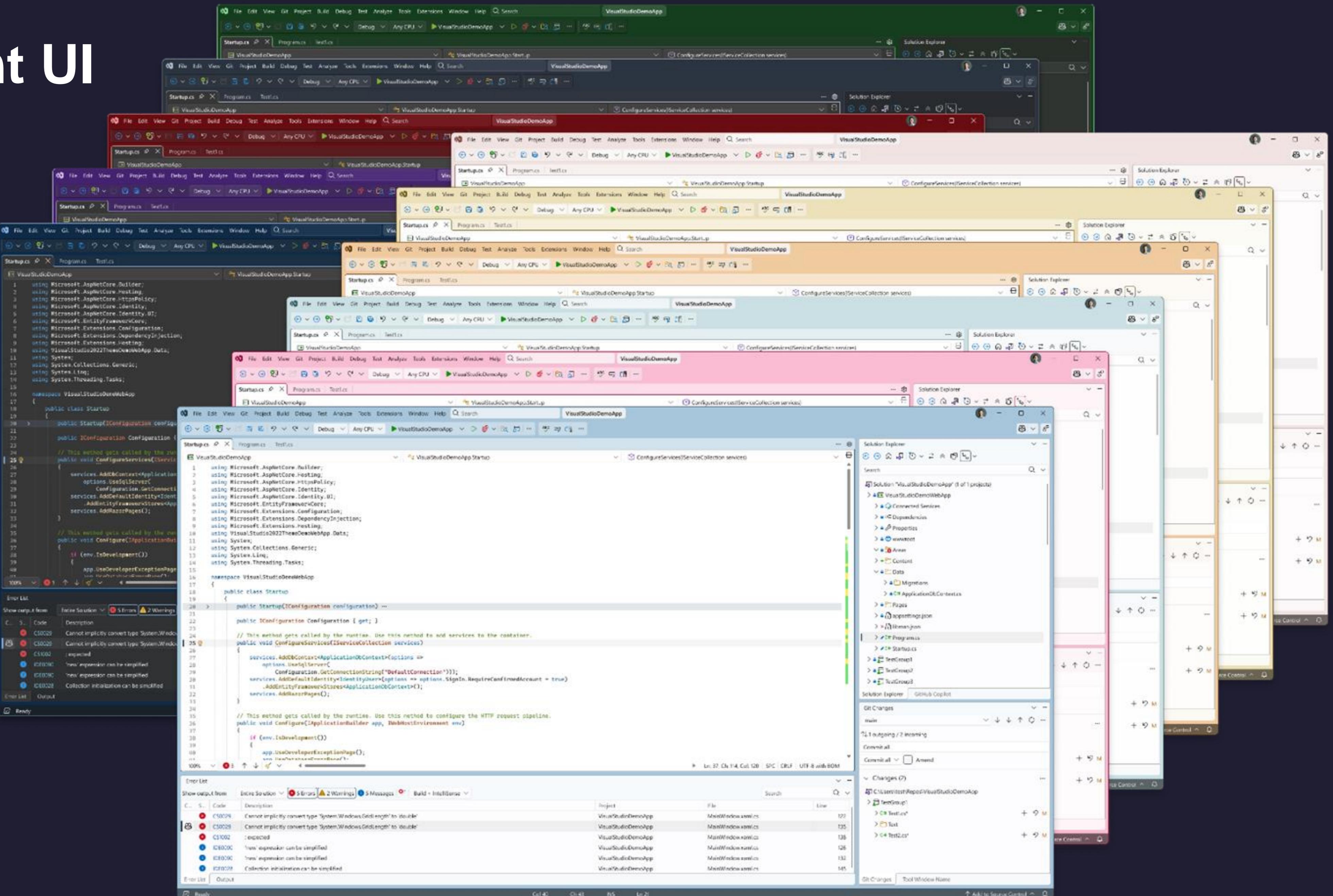
.NET 10



Удобная установка



Fluent UI



Новый интерфейс параметров

The screenshot shows the 'All Settings' view in the Visual Studio Options dialog. The left sidebar lists various settings categories under 'All Settings'. The main area displays the 'Environment' section, specifically the 'General' settings. It includes options like 'Items to show in Window menu' (set to 10), 'Items to show in recently used files list' (set to 10), and checkboxes for 'Show status bar' (checked), 'Close button affects active tool window only' (checked), and 'Auto Hide button affects active tool window only' (unchecked). Below these are sections for 'On startup, open' (set to 'Start window') and 'Visual Experience' (color theme set to 'Dark'). The 'Editor Appearance' section is also visible, with a checkbox for 'Separate font settings from color theme selection' (checked). At the bottom, there are three more checked checkboxes: 'Use Windows High Contrast settings', 'Apply title case styling to menu bar', and 'Use compact menu and search bar'.

Options X

Search Settings Filter: All Settings Applies to: user (current install) Edit user settings as JSON

All Settings

Environment

General

Items to show in Window menu

10

Items to show in recently used files list

10

Show status bar

Displays the status bar. The status bar is located at the bottom of the IDE window and displays information about the progress of ongoing operations.

Close button affects active tool window only

Auto Hide button affects active tool window only

On startup, open

Start window

Visual Experience

Color theme

Dark

Editor Appearance

Match Theme

Separate font settings from color theme selection

Saves font settings independently so that switching editor color themes won't affect your font customizations.

Use Windows High Contrast settings

Apply title case styling to menu bar

Use compact menu and search bar

Mermaid диаграммы

Fabric Method.md X

Split preview Open preview

```
1  ````mermaid
2  classDiagram
3
4  class Creator {
5      <<Abstract>>
6      void ActionWithObject()
7      object FabricMethod()
8  }
9
10 Creator <|-- CreatorA:Create object
11 Creator <|-- CreatorB:Create object
12
13 class CreatorA{
14     object FabricMethod()
15 }
16
17 class CreatorB{
18     object FabricMethod()
19 }
20
21     ...
22
23 }
```

The diagram illustrates the Fabric Method design pattern. At the top is an abstract class **Creator** with two subclasses, **CreatorA** and **CreatorB**. Each subclass has its own **FabricMethod()** object. The **Creator** class also defines an **ActionWithObject()** method.

```
classDiagram
    class Creator {
        <<Abstract>>
        void ActionWithObject()
        object FabricMethod()
    }
    class CreatorA {
        object FabricMethod()
    }
    class CreatorB {
        object FabricMethod()
    }
    Creator <|-- CreatorA:Create object
    Creator <|-- CreatorB:Create object
```

Annotations on the right side of the diagram identify the methods:
- **Create object type A** points to the **Create object** association between **Creator** and **CreatorA**, and to the **FabricMethod()** object in **CreatorA**'s compartment.
- **Create object type B** points to the **Create object** association between **Creator** and **CreatorB**, and to the **FabricMethod()** object in **CreatorB**'s compartment.

Покрытие кода тестами

Доступно в Community и
Professional

The screenshot shows the Visual Studio IDE interface. The top part displays a C# code editor with several test methods. The bottom part shows the 'Code Coverage Results' window, which includes a tree view of covered files and a detailed table of coverage statistics.

```
22 [TestMethod]
23     public void RooterValueRange()
24     {
25         // Create an instance to test.
26         Rooter rooter = new Rooter();
27
28         // Try a range of values.
29         for (double expected = 1e-8; expected < 1e+8; expected *= 3.2)
30         {
31             RooterOneValue(rooter, expected);
32         }
33     }
34
35     private void RooterOneValue(Rooter rooter, double expectedResult)
36     {
37         double input = expectedResult * expectedResult;
38         double actualResult = rooter.SquareRoot(input);
39         Assert.AreEqual(expectedResult, actualResult, delta: expectedResult / 1000);
30
31     }
32 [TestMethod]
33     public void RooterTestNegativeInput()
34     {
35         Rooter rooter = new Rooter();
36         Assert.ThrowsException<ArgumentOutOfRangeException>(() => rooter.SquareRoot(-1));
37     }
38 }
```

Code Coverage Results

	Covered (Blocks)	Not Covered (Blocks)	Covered (Lines)	Partially Covered (Lines)	Not Covered (Lines)
SURFACE_2023-12-05.13_17.coverage	27	1	36	1	0
mymath.dll	9	0	13	0	0
MyMath	9	0	13	0	0
Rooter	9	0	13	0	0
SquareRoot(double)	9	0	13	0	0
mathtests.dll	18	1	23	1	0
MathTests	18	1	23	1	0
UnitTest1	17	0	23	0	0
BasicRooterTest()	4	0	7	0	0
RooterValueRange()	6	0	7	0	0

The screenshot shows the 'Code Coverage Results' window in Visual Studio. The window displays a hierarchical tree of covered files and a detailed table of coverage statistics. A red box highlights the 'Search' button in the toolbar.

	Covered (Blocks)	Not Covered (Blocks)	Covered (Lines)	Partially Covered (Lines)	Not Covered (Lines)
SURFACE_2023-12-05.13_17.coverage	27	1	36	1	0
mymath.dll	9	0	13	0	0
MyMath	9	0	13	0	0
Rooter	9	0	13	0	0
SquareRoot(double)	9	0	13	0	0
mathtests.dll	18	1	23	1	0
MathTests	18	1	23	1	0
UnitTest1	17	0	23	0	0
BasicRooterTest()	4	0	7	0	0
RooterValueRange()	6	0	7	0	0

```
Program <top-level-statements-entry-point>
FibonacciIterative(5); args: {string[0]}

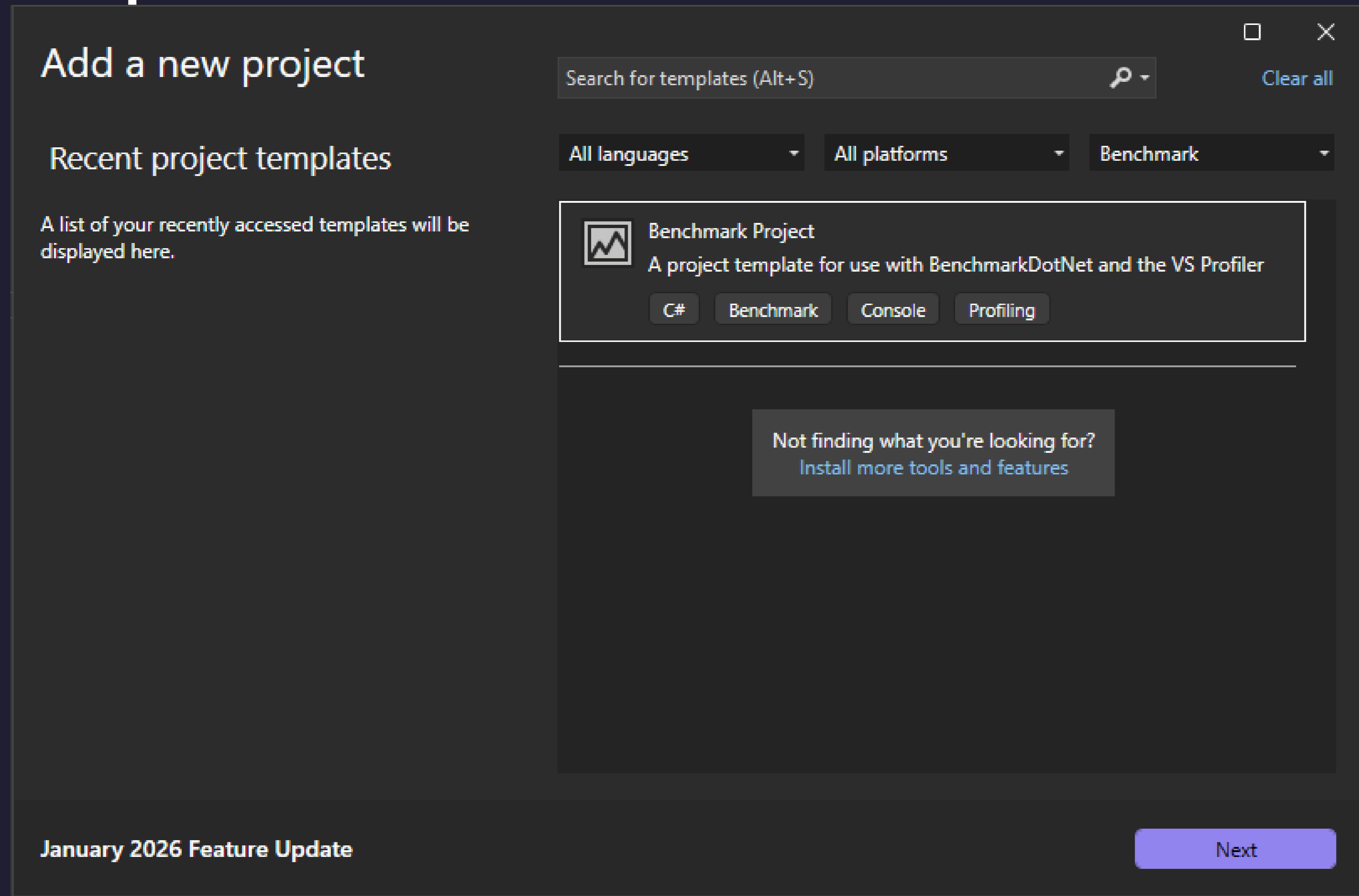
1 reference
static long FibonacciIterative(int n)
{
    if (n <= 1) return n;

    long a = 0, b = 1, result = 0;

    for (int i = 2; i <= n; i++)
    {
        result = a + b;
        a = b;
        b = result;
    }

    return result;
}
```

Шаблон проекта BenchmarkDotNet



Performance Profiler

Select a target and diagnostic tools for performance analysis. Ensure the solution configuration is set to **Release** for more accurate results.

Analysis target: Startup Project ▾

Target name: ContosoPizza

Get guided performance analysis:  Try Profiler Agent

Flexible Standalone  Recommended

Diagnose with multiple tools that can run simultaneously

.NET Async 

Tool to investigate `async/await` usage in .NET applications

Database 

Examine when queries were executed and measure how long they take

Memory Usage 

Investigate application memory to find issues such as memory leaks

.NET Counters 

Tool to visualize performance counters in .NET applications

Events Viewer  

See the events (ETW or NetTrace) that occurred during the session, such as log messages, exceptions and HTTP requests

CPU Usage  

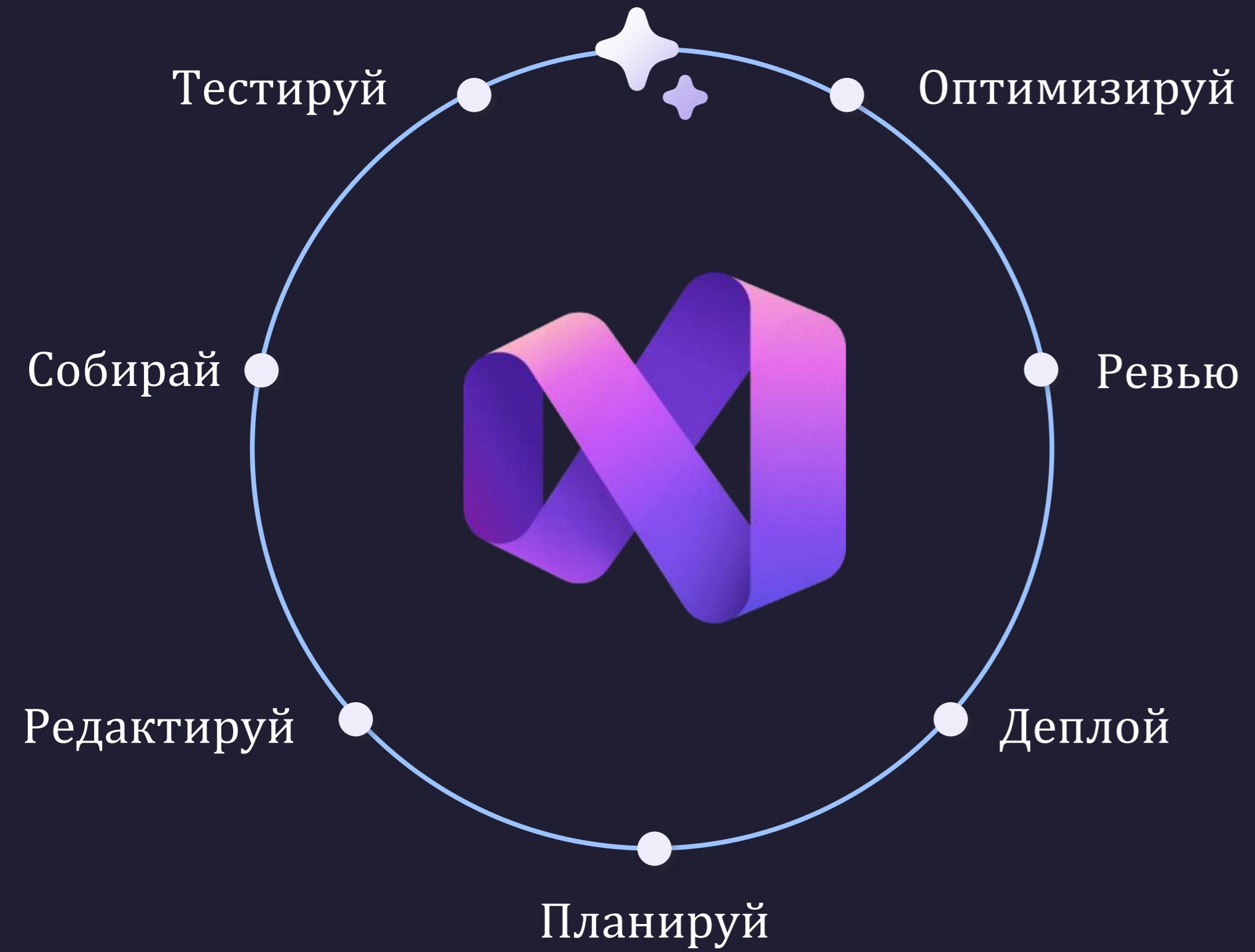
See where the CPU is spending time executing your code. Useful when the CPU is the performance bottleneck

File IO 

See what File I/O operations are being performed, how long they take, and how much data they're processing

Start

Интеграция с AI



Visual Studio 2026



Возможности

- Отладка
- Профилирование
- Тестирование
- Модернизация
- GitHub



Инструменты

- Планирование
- Microsoft Learn
- NuGet
- GitHub
- MCP server

```
1.cs* ➔ X UseCase1_PascalCase.cs | CodeAnswers.cs | Calculator.cs | Math.cs | UseCase1_underscores.cs | Test1.cs | copy.txt | SmartPasteTestProject.csproj | CodeAnswers.json | port.cpp
SmartPasteTestProject
    ↴ sockaddr_in
41     return -1;
42 }
43
44 // Listen for connections
45 if (listen(server_fd, 3) < 0) ⚡ CS0103: The name 'listen' does not exist in the current context
46 {
47     std::cerr << "Listen failed" << std::endl; ⚡ CS0201: Only assignment, call, increment, decrement, await, and new object expressions can be used as a statement
48     return -1;
49 }
50
51 std::cout << "Server is listening on port " << port << std::endl; ⚡ CS0432: Alias 'std' not found
52
53 while (true)
54 {
55     // Accept incoming connection
56     new_socket = accept(server_fd, (struct sockaddr*)&address, (socklen_t*)&addrlen); ⚡ CS1002: ; expected
57     1 reference
58     if (new_socket < 0) ⚡ CS0103: The name 'new_socket' does not exist in the current context
59     {
60         std::cerr << "Accept failed" << std::endl; ⚡ CS0201: Only assignment, call, increment, decrement, await, and new object expressions can be used as a statement
61         continue; ⚡ CS0139: No enclosing loop out of which to break or continue
62     }
63
64     // Send HTTP response
65     send(new_socket, http_response.c_str(), http_response.size(), 0); ⚡ CS0103: The name 'send' does not exist in the current context
66
67     // Close the socket
68     close(new_socket); ⚡ CS0103: The name 'close' does not exist in the current context
69     } ⚡ CS1022: Type or namespace definition, or end-of-file expected
70     Tab accept(); ⚡ CS1022: Type or namespace definition, or end-of-file expected
71 }
```

Copilot Chat

Thread

 Copilot Chat

I might make mistakes, so check for accuracy.

Explain how the code in #class works.

Write unit tests for all the methods in #file.

How can I improve #method? Suggest code for any improvements.

Use # for references, / for commands, and @ for extensions.

what can you do?

Ask or use @workspace

Ask GPT-4.1

Active Document X

Copilot or use @workspace

Ask GPT-4.1

File Solution Explorer Properties Git Changes 100% No issues found

Tests.cs

VariableAnalysisTests

UnitTestProject

[TestClass]

[TestMethod]

public void VerifyCompoundInterestCalculation()

double totalAmount = VariableAnalysisTests.ComputeCompoundInterest(1000.0, 0.05, 10);
Assert.IsTrue(totalAmount > 1620);

[TestMethod]

public void VerifyAddressSerializationDeserialization()

Address address1 = new Address("123 Main St", "Springfield", "IL", "12345");
Address address2 = address.FromJson(address1ToJson());
Assert.AreEqual(address1, address2);

[TestMethod]

public void VerifyProductListInitialization()

List<Product> products = Product.InitData("https://dummyjson.com/products");
Assert.AreEqual(30, products.Count);

[TestMethod]

public void VerifyProductListInitialization2()

var rootObject = Product.InitData2("https://dummyjson.com/products");
Assert.AreEqual(30, rootObject.Products.Count);

[TestMethod]

public void VerifyProductListInitialization3()

Line 21, Ch 13 SPC CRLF UTF-8 with BOM

Test Explorer

Test run finished: 1 Tests (0 Passed, 1 Failed, 0 Warnings, 0 Errors)

Test	Duration	Results
VariableAnalysisTests (1)	1.6 sec	Passed
UnitTestProject (0)	1.6 sec	Passed
Tests (0)	1.6 sec	Passed
VerifyAddressSerializationDeserialization	28 ms	Failed
VerifyCompoundInterestCalculation	32 ms	Passed
VerifyProductListInitialization	129 ms	Passed
VerifyProductListInitialization2	181 ms	Passed
VerifyProductListInitialization3	11 ms	Passed
VerifyUpdateInterest	32 ms	Passed

Test Detail Summary

VerifyAddressSerializationDeserialization

Source: Tests.cs line 17

Duration: 28 ms

Message:

Assert.AreEqual failed. Expected:<VariableAnalysisTests>.Actual:<1620>

Adler32.cs Adler32.cs Programs.cs

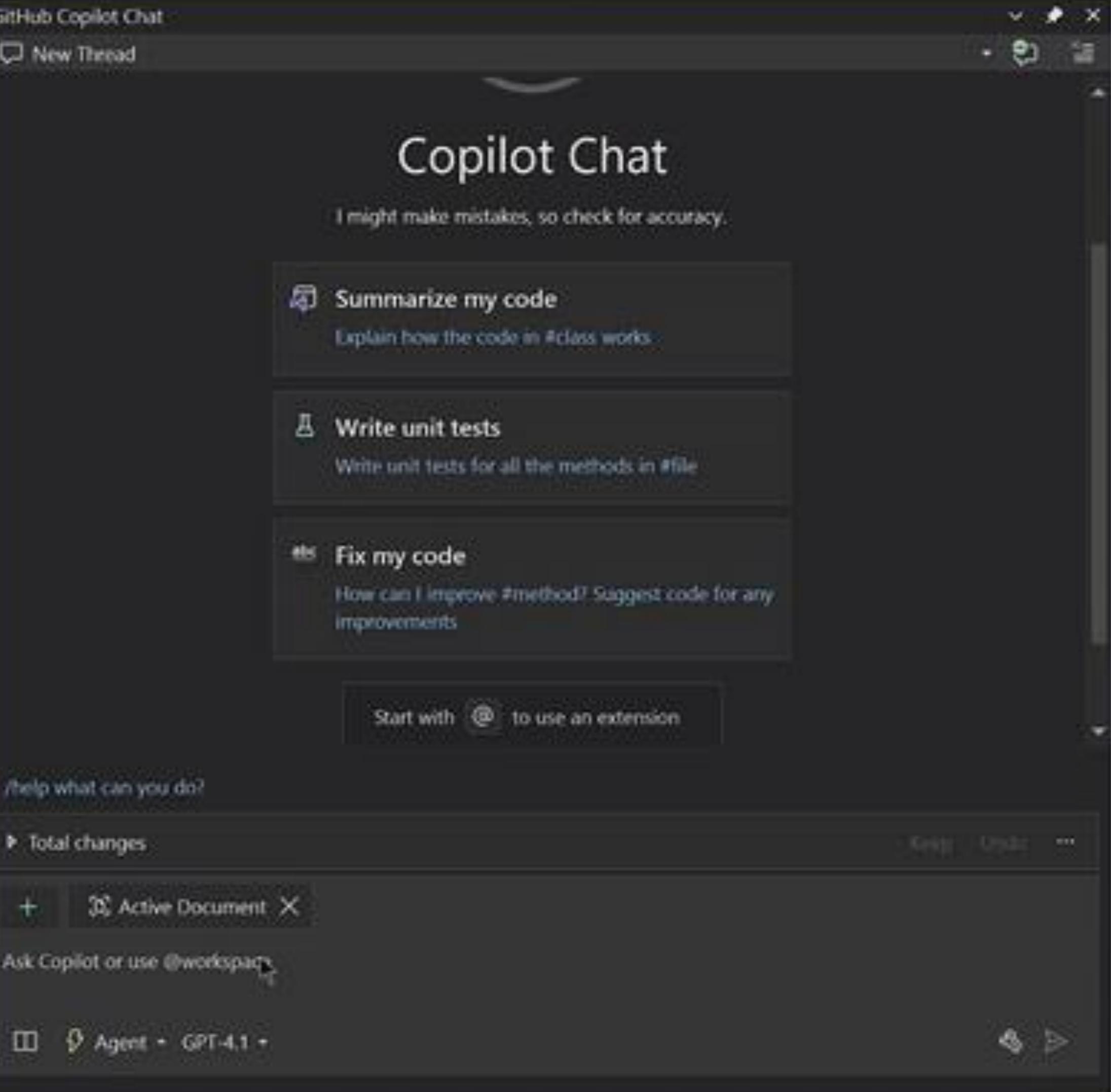
```
1  using BenchmarkDotNet.Configs;
2  using BenchmarkDotNet.Jobs;
3  using BenchmarkDotNet.Running;
4  using BenchmarkDotNet.Toolchains.CsProj;
5
6  namespace ICSharpCode.SharpZipLib.Benchmark
7  {
8      10 references | 0 changes | 0 authors, 0 changes
9      public class MultipleRuntimes : ManualConfig
10     {
11         0 references | 0 changes | 0 authors, 0 changes
12         public MultipleRuntimes()
13         {
14             //AddJob(Job.Default.WithToolchain(CsProjCl)
15             AddJob(Job.Default.WithToolchain(CsProjCore'
16
17         1 reference | 0 changes | 0 authors, 0 changes
18         class Program
19         {
20             0 references | 0 changes | 0 authors, 0 changes
21             static void Main(string[] args)
22             {
23                 BenchmarkSwitcher.FromAssembly(System.Refl
24                     args,
25                     DefaultConfig.Instance
26                         .AddJob(BenchmarkDotNet.Jobs.Job.De
27                         .AddDiagnoser(new Microsoft.VSDiagn
```

125% No issues found

Show output from: Build

```
===== build: 0 succeeded, 0 failed, 3 up-to-date, 0 skipped =====
===== Build completed at 3:53 PM and took 00.078 seconds =====
```

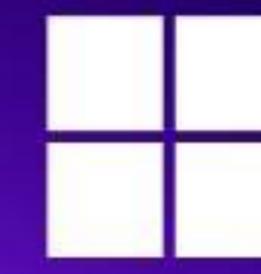
Ready



Ревью кода

The screenshot shows a dark-themed code review interface. At the top, there are buttons for 'Commit All' (with a dropdown arrow) and 'Amend'. Below these are sections for 'Related Items' and 'Changes (4)'. A tooltip 'Review changes with Copilot (might be inaccurate)' is visible over the 'Changes (4)' section. The 'Changes (4)' section lists the following files:

- `C:\Users\jehou.../copilot-instructions.md` (A)
- `src/ClientApp\Services\Catalog/CatalogService.cs` (M)
- `WebApp\Components\Pages\Catalog`

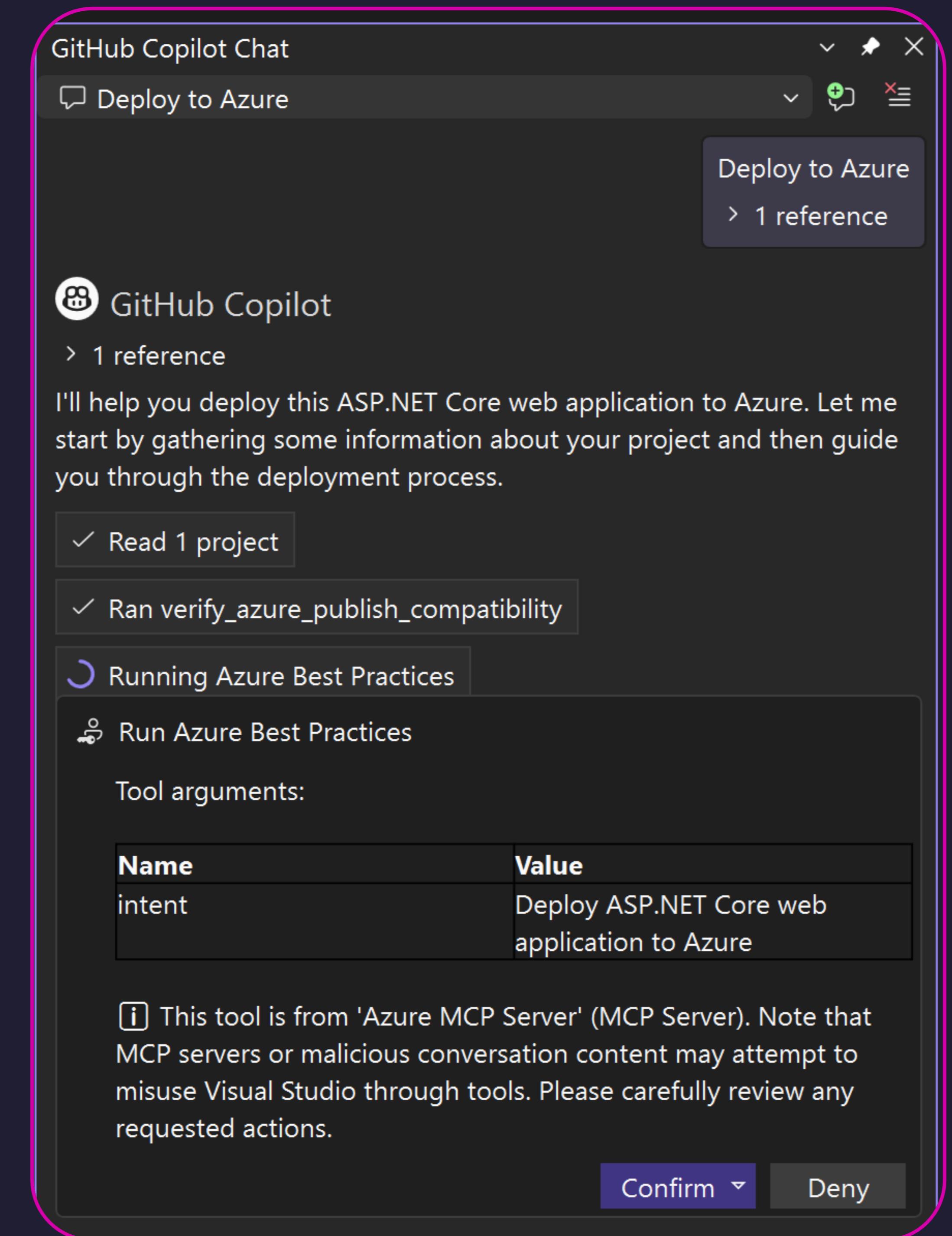


Microsoft Visual Studio

INSIDERS

2026

MCP Server



Модернизация

GitHub Copilot Chat

App Modernization experience

The screenshot shows the GitHub Copilot Chat interface with the title "App Modernization experience". It features a large white icon of a smiling face with a double arrow pointing up and to the right. Below the icon, the text "Modernize with Copilot" is displayed in large, bold, white font. A subtitle below it says "I might make mistakes, so check for accuracy." Three rectangular buttons are listed: "Upgrade to a newer version of .NET" (with a .NET icon), "Migrate to Azure" (with an Azure icon), and "Explore more modernization options" (with a magnifying glass icon). At the bottom, there is a text input field containing "/help what can you do?" and a footer bar with icons for a new document, the active document (labeled "Active Document"), and settings.

Modernize with Copilot

I might make mistakes, so check for accuracy.

Upgrade to a newer version of .NET

Migrate to Azure

Explore more modernization options

/help what can you do?

Active Document

Ask Copilot

Agent GPT-5 mini

20

BlazorSignalRApp - assessment.md [Preview]

assessment.md [Preview] Edit Markdown

Projects and dependencies analysis

This document provides a comprehensive overview of the projects and their dependencies in the context of upgrading to .NETCoreApp, Version=v10.0.

Table of Contents

- Executive Summary
 - Highlevel Metrics
 - Projects Compatibility
 - Package Compatibility
 - API Compatibility
- Aggregate NuGet packages details
- Top API Migration Challenges
 - Technologies and Features
 - Most Frequent API Issues
- Projects Relationship Graph
- Project Details
 - BlazorSignalRApp\BlazorSignalRApp.Client\BlazorSignalRApp.Client.csproj

The assessment is complete and documented in [assessment.md](#). The next stage is Planning, where a detailed migration plan will be created based on this assessment.

Would you like to:

- review or adjust the assessment, or
- proceed to the Planning stage?

[Proceed to Planning stage](#)

[Review assessment.md findings](#)

[Adjust analysis scope](#)

Ask Copilot

Agent GPT-5 mini

GitHub Copilot Chat Solution Explorer Git Changes

GitHub Copilot Chat

App Modernization experience

- The analyzer ran against the solution and recorded findings in [assessment.md](#). Key findings include required target-framework updates for the client, server, and test projects, recommended NuGet package upgrades, and API compatibility / behavioral-change warnings for server and client projects.
- No NuGet packages with security vulnerabilities were reported by the analysis.

plan.md [Preview] X

Edit Markdown

.github/upgrades/plan.md

Table of Contents

- Executive Summary
- Migration Strategy
- Detailed Dependency Analysis
- Project-by-Project Plans
- Package Update Reference
- Breaking Changes Catalog
- Testing & Validation Strategy
- Risk Management
- Complexity & Effort Assessment
- Source Control Strategy
- Success Criteria
- Appendices

GitHub Copilot Chat

App Modernization experience

Upgrade. Now I'll create a detailed specification explaining *how* to make those changes.

What I'll Create

A comprehensive [plan.md](#) that transforms assessment findings into an actionable roadmap:

- **Migration Strategy** — The approach best suited to your codebase (incremental phases or coordinated update)
- **Execution Sequence** — Dependency-ordered steps ensuring nothing breaks along the way
- **Risk Mitigation** — Strategies to handle potential issues before they occur
- **Validation Checkpoints** — Clear success criteria for each phase

This plan becomes the blueprint for the Execution stage, where actual

[Show recommended .csproj changes](#)[Generate tasks.md for execution](#)[Drill into flagged issues from assessment.md](#)

Active Document X

Ask Copilot



Agent ▾ GPT-5 mini ▾



GitHub Copilot Chat

Solution Explorer

Git Changes

tasks.md [Preview] X

BlazorSignalRApp .NET 10 Upgrade Tasks

Overview

This document lists executable tasks to upgrade the `BlazorSignalRApp.sln` solution from `net9.0` to `net10.0`. The workflow verifies prerequisites, performs the atomic project/package upgrade and compilation fixes, then runs tests and creates the final commit per the plan.

Progress: 1/4 tasks complete (25%) 

Tasks

[✓] TASK-001: Verify prerequisites (Completed: 2026-01-27 20:49)

References: Plan §4 (Project-by-Project Plans - Shared prerequisites), Plan §11 (Appendices - quick checklist)

- [✓] (1) Verify .NET 10 SDK is installed on the build machine (`dotnet --list-sdks`) per Plan §4
- [✓] (2) Runtime/sdk version meets minimum requirements (Verify)
- [✓] (3) Check `global.json` (if present) for SDK compatibility with `net10.0` per Plan §4 (do not modify files in this step)
- [✓] (4) `global.json` is compatible or requires update (Verify)

[▶] TASK-002: Atomic framework and package upgrade with compilation fixes

References: Plan §4 (Project-by-Project Plans), Plan §5 (Package Update Reference), Plan §6 (Breaking Changes Catalog)

- [▶] (1) Update `<TargetFramework>` to `net10.0` in all projects listed in Plan §4:
 - `BlazorSignalRApp\BlazorSignalRApp.Client\BlazorSignalRApp.Client.csproj`
 - `BlazorSignalRApp\BlazorSignalRApp\BlazorSignalRApp.csproj`
 - `BlazorSignalRApp\BlazorSignalRApp.Tests\BlazorSignalRApp.Tests.csproj`
- (2) Update package references per Plan §5 (key updates: `Microsoft.AspNetCore.Components.WebAssembly` → `10.0.2` for client, `Microsoft.AspNetCore.SignalR.Client` → `10.0.2` for client, `Microsoft.AspNetCore.Components.WebAssembly.Server` → `10.0.2` for server; see Plan §5 for full matrix)
- (3) Restore dependencies across the solution (`dotnet restore`) per Plan §7
- (4) Build the solution and fix all compilation errors found (address source-incompatible changes such as `TimeSpan.FromSeconds(long)` and other items from Plan §6)
- (5) Solution builds with 0 errors (Verify)

[!] TASK-003: Run test suite and validate upgrade

 Edit Markdown

GitHub Copilot Chat

App Modernization experience

5 tasks done

c# BlazorSignalRApp.Client.... BlazorSignalRApp\Blazor... -3 +1

c# BlazorSignalRApp.Client.... BlazorSignalRApp\Blazor... -2 +2

✓ Read BlazorSignalRApp.csproj, lines 1-120

c# BlazorSignalRApp.csproj BlazorSignalRApp\Blazor... -2 +2

✓ Read BlazorSignalRApp.Tests.csproj, lines 1-200

...

▼ Total changes (2)

Keep Undo ...

c# BlazorSignalRApp.Client.csproj

c# BlazorSignalRApp.csproj

+ Active Document X

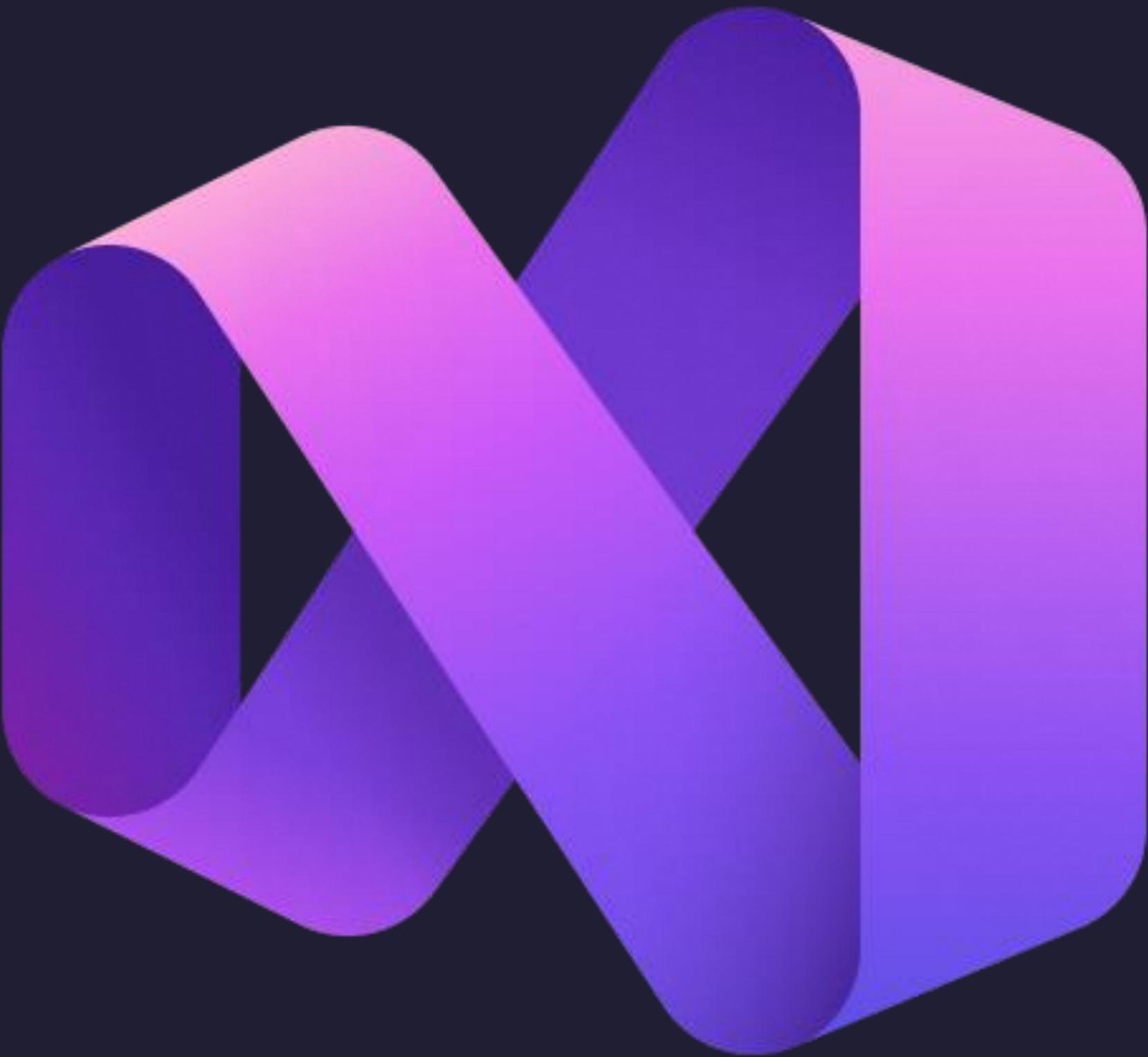
Ask Copilot

Agent GPT-5 mini





Release notes



Visual Studio
2026 is here