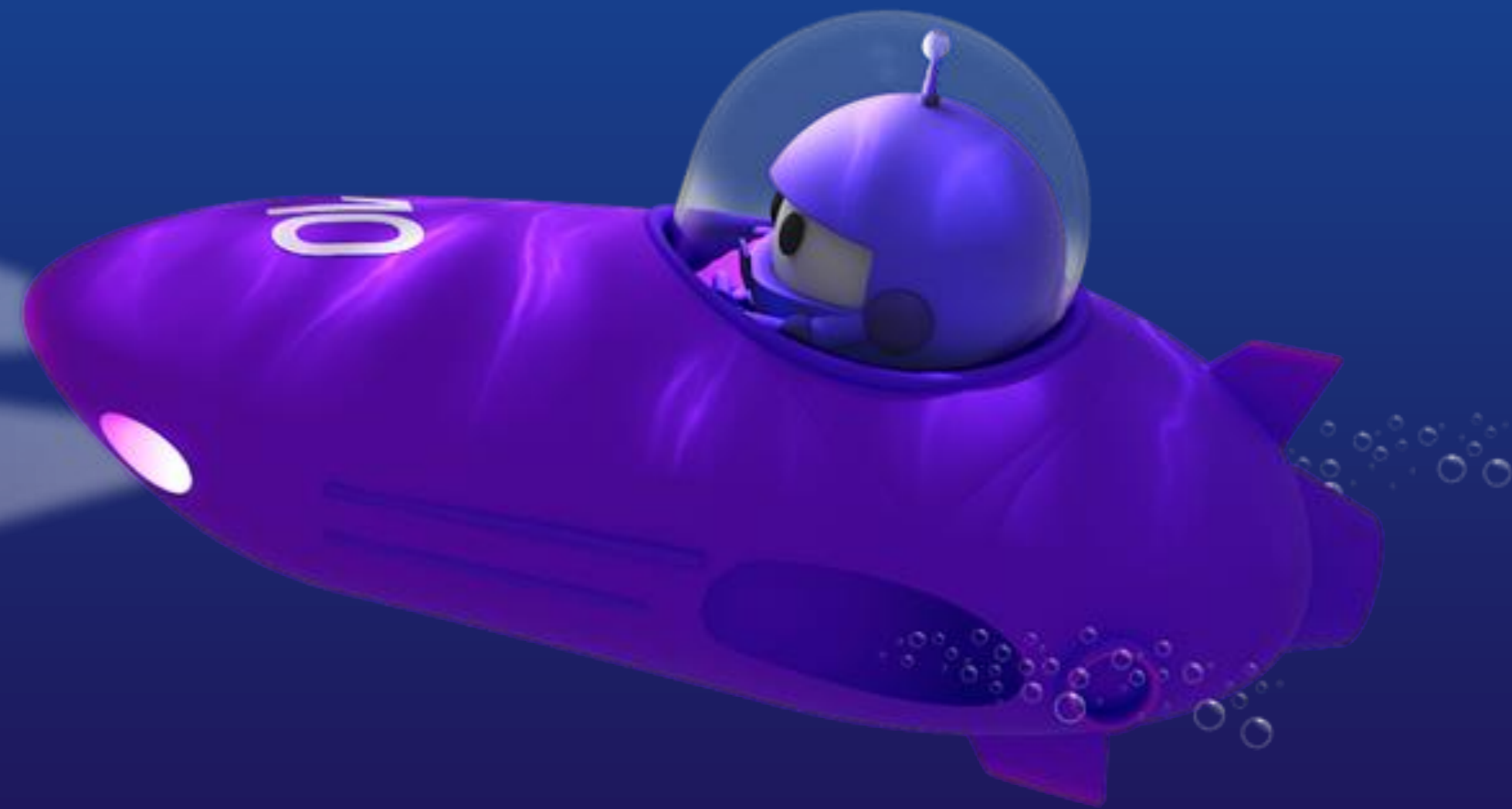


Встречайте Visual Studio

2026



Visual Studio 2026

Быстрая, удобная, умная



Высокая

Ускоренная загрузка решений

- Быстрый переход к отладке



Удобство работы

- Новый UI
- Покрытие тестами
- Inline в режиме отладки



AI работающий с вами

- Агенты
- MCP

Время загрузки решения

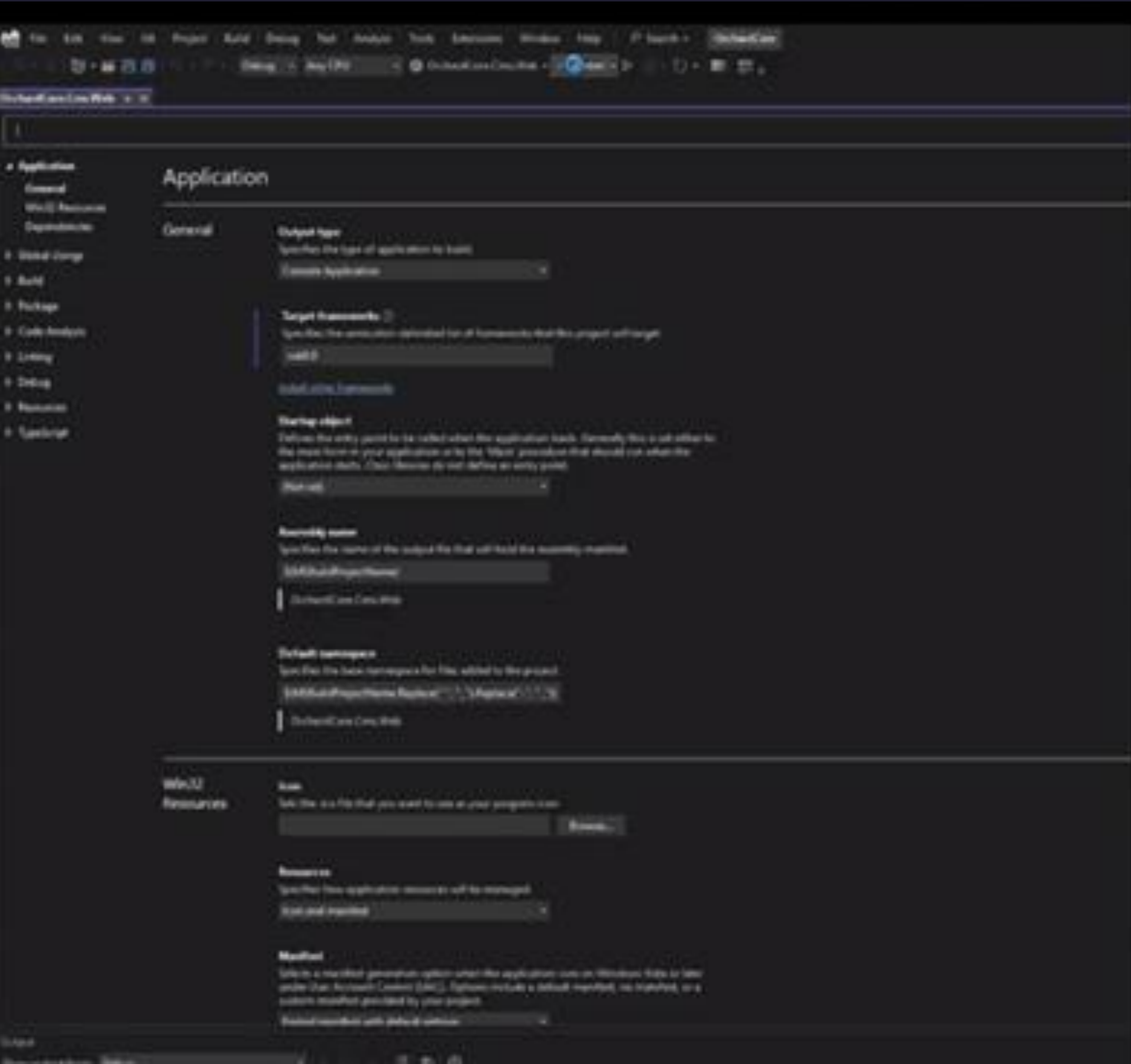
223 проекта



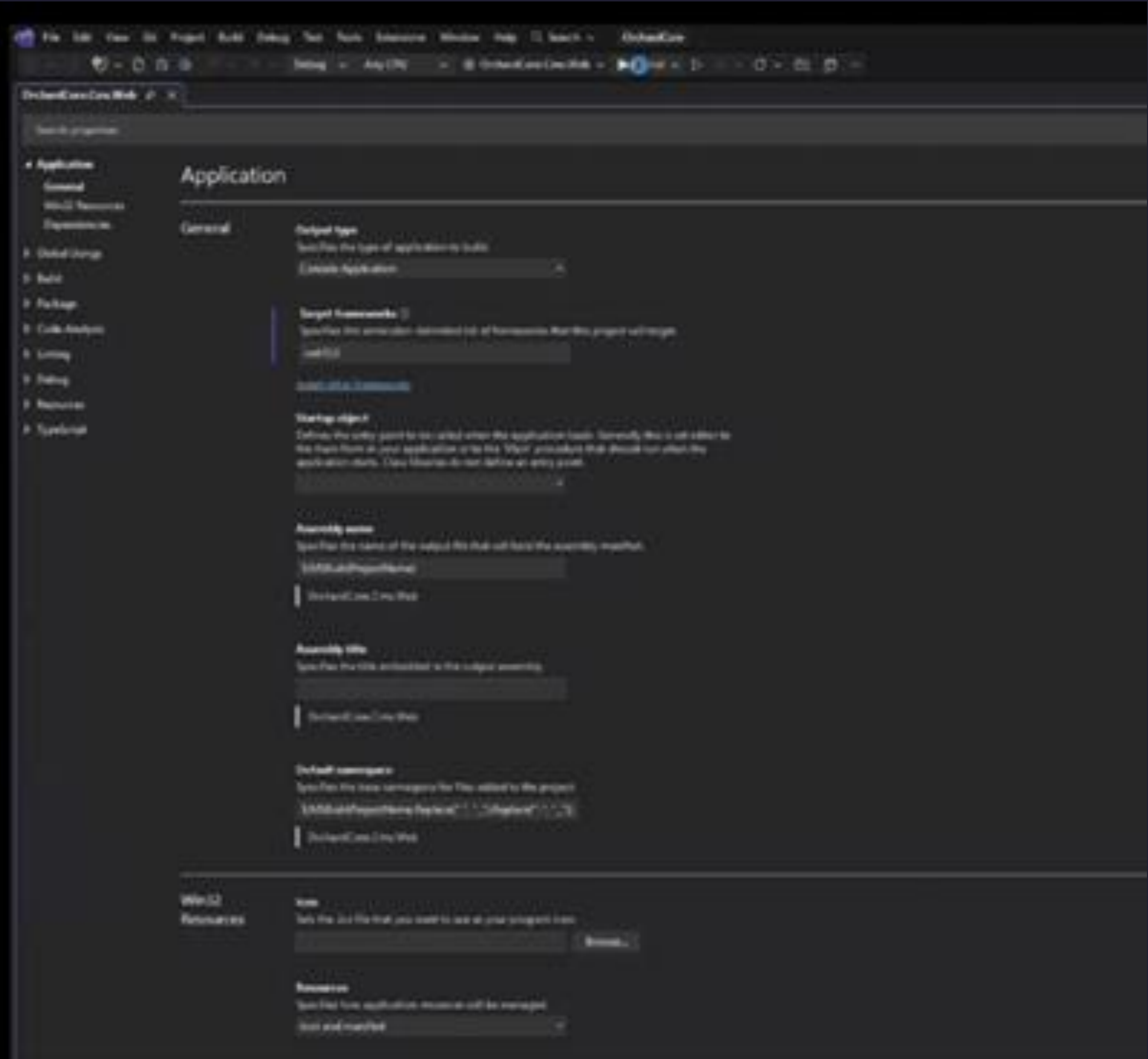
Тестировалось на Orchard Core:
github.com/OrchardCMS/OrchardCore

Переход в отладочный режим

.NET 9



.NET 10



Удобная установка

Установить Visual Studio Community 2026

Выберите способ настройки Visual Studio. Новая установка будет выполнена параллельно с существующими версиями.

Что произойдет в случае выполнения установки?

- ☒ Копировать рабочие нагрузки, компоненты и параметры из предыдущей установки

Visual Studio Community 2022

☐ Включить компоненты, помеченные как неподдерживаемые

☒ Включить установленные расширения из Visual Studio Marketplace ⓘ

- ☐ Использовать файл VSCONFIG ⓘ

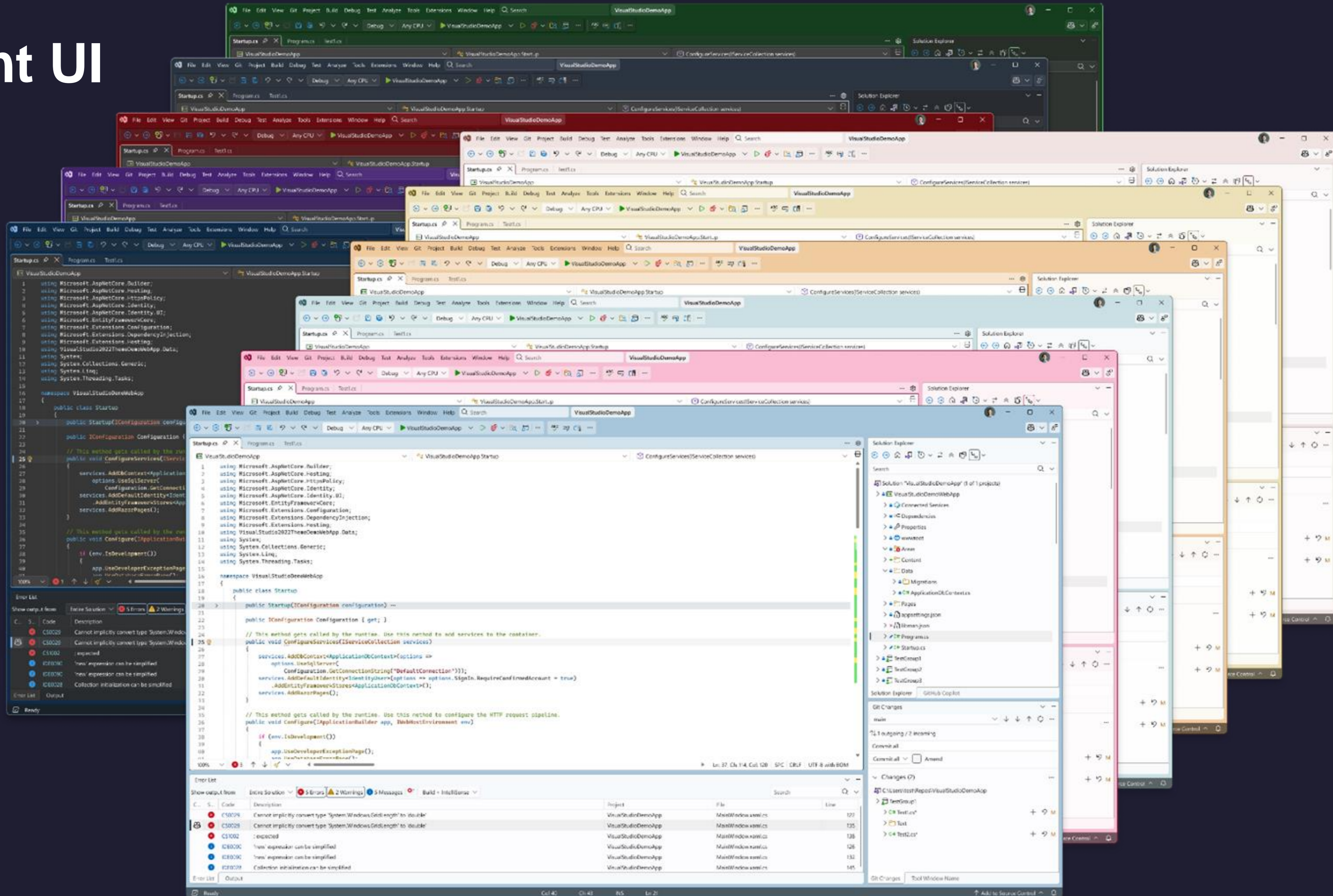
Папка для импорта

C:\Users\darka\OneDrive\Documents\vsconfig

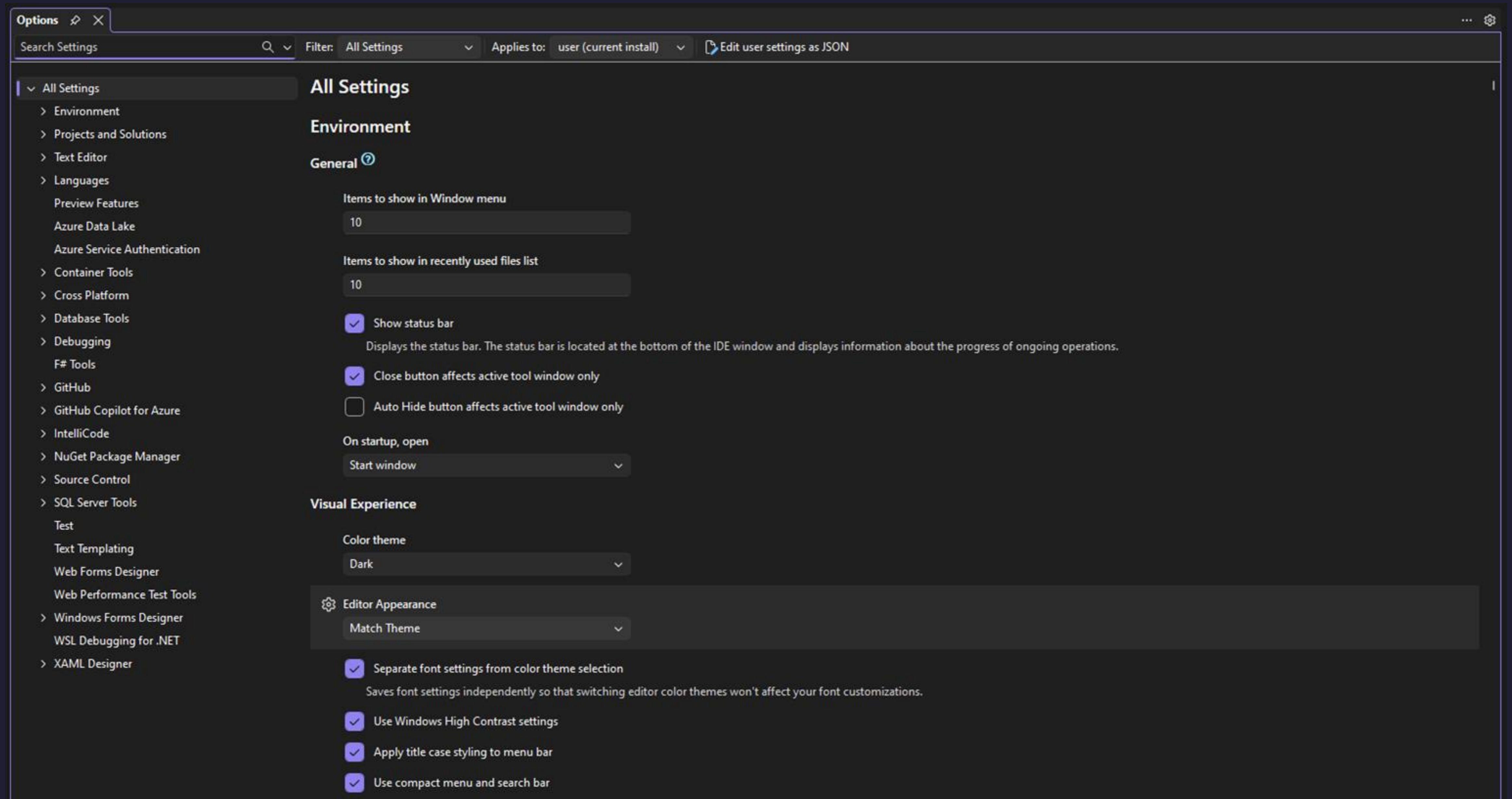
- ☐ Выбрать рабочие нагрузки и компоненты вручную ⓘ

Далее

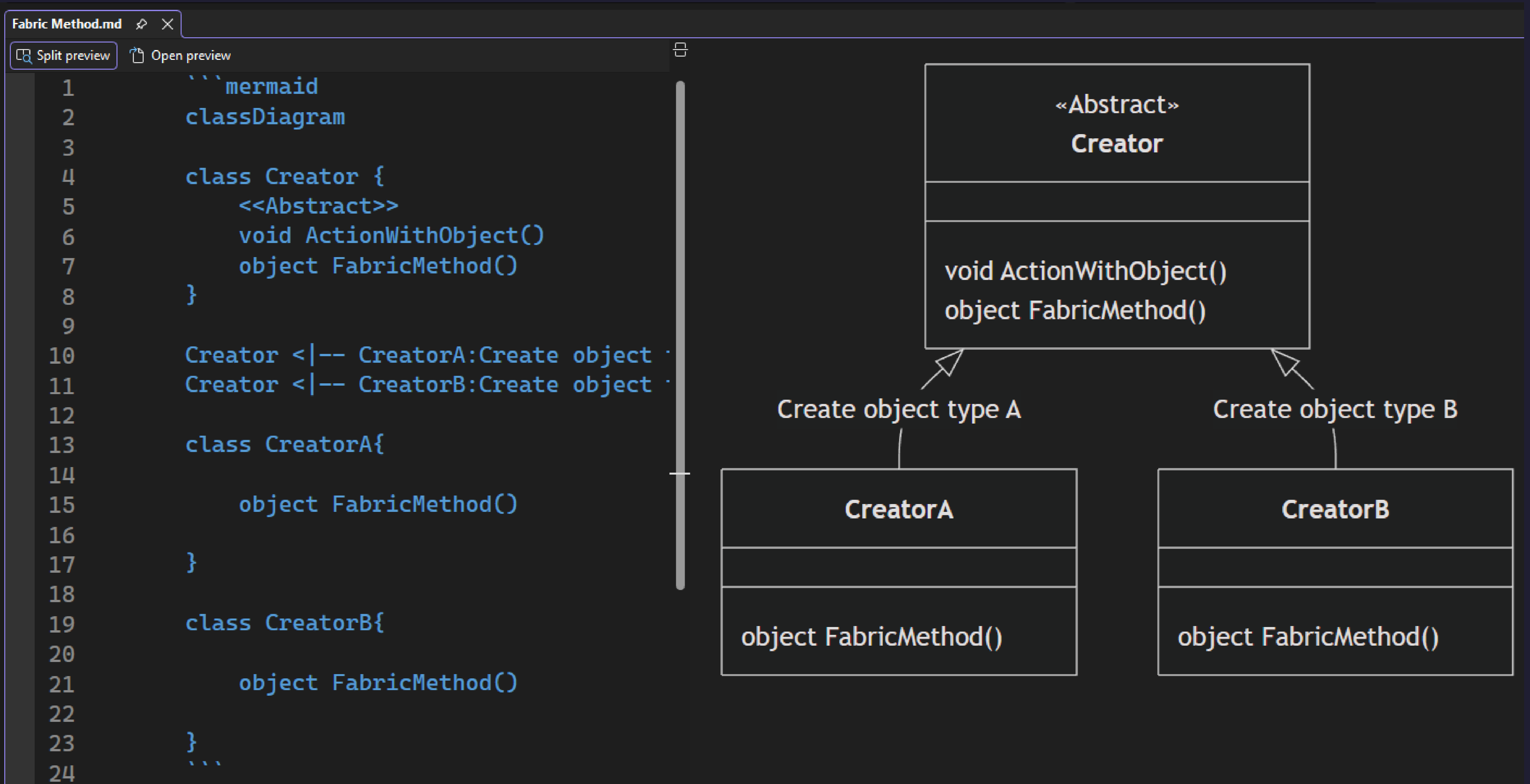
Fluent UI



Новый интерфейс параметров



Mermaid диаграммы



Покрытие кода тестами

Доступно в Community и Proffessional

The screenshot displays the Visual Studio IDE with a C# project. The main editor shows the `MathTests` namespace containing `Unit1.cs`. The code includes a `Router` class with a `SquareRoot` method and a `RouterOneValue` test method. The `RouterTestNegativeInput` method tests the `SquareRoot` method with a negative input. The `Code Coverage Results` window at the bottom shows the coverage for the `mymath.dll` and `mathtests.dll` assemblies.

Code Coverage Results

Hierarchy	Covered (Blocks)	Not Covered (Blocks)	Covered (Lines)	Partially Covered (Lines)	Not Covered (Lines)
SURFACE_2023-12-05.13_17_07.coverage	27	1	36	1	0
mymath.dll	9	0	13	0	0
MyMath	9	0	13	0	0
Router	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
Unit1	17	0	23	0	0
BasicRouterTest()	4	0	7	0	0
RouterValueRange()	6	0	7	0	0

Program

FibonacciIterative(5);

args: {string[0]}

1 reference

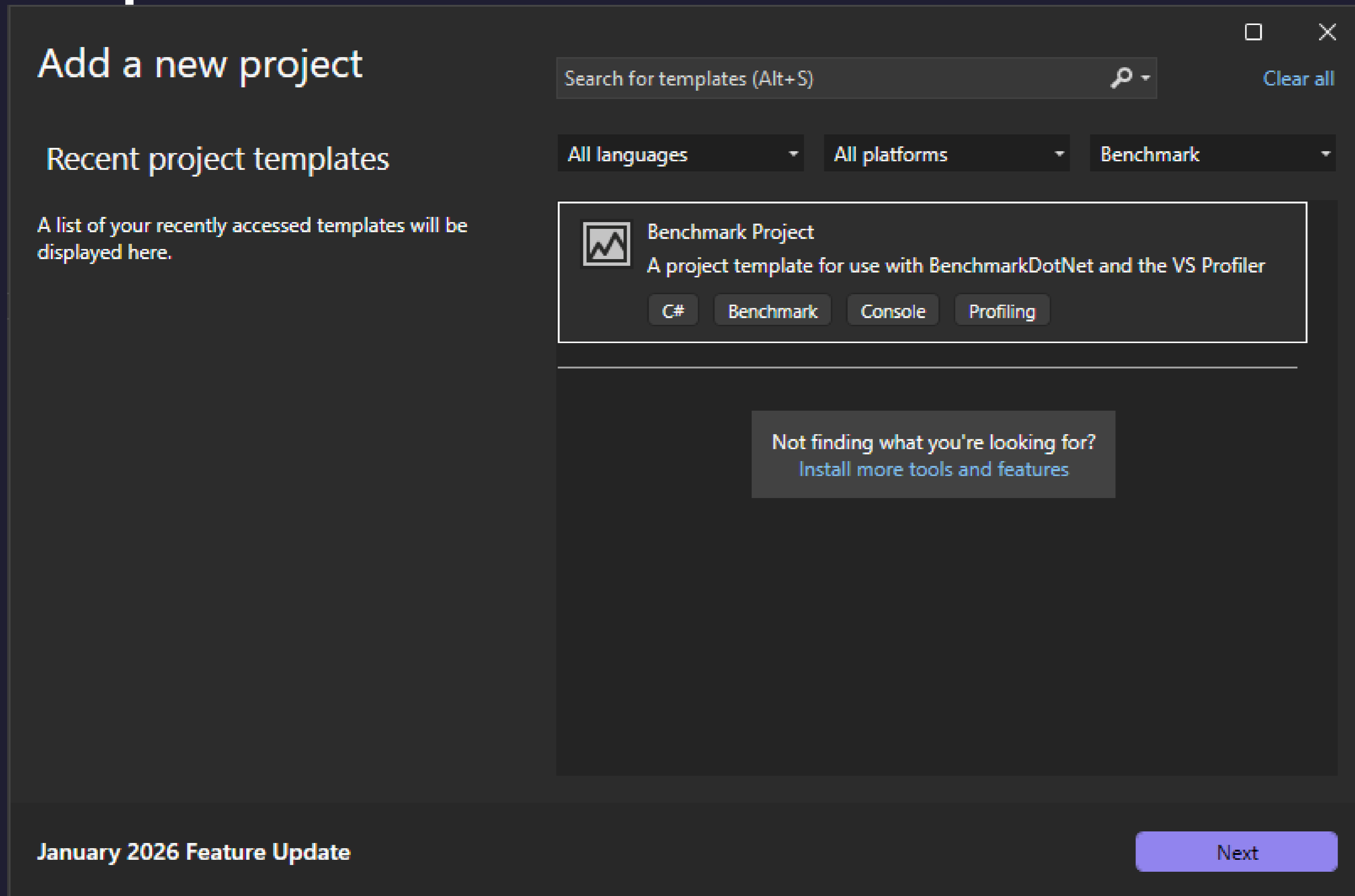
```
static long FibonacciIterative(int n)
{
    if (n <= 1) return n;

    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

☐ .NET Counters ?

Tool to visualize performance counters in .NET applications

☐ CPU Usage ⚙️ ?

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

☐ Database ?

Examine when queries were executed and measure how long they take

☐ Events Viewer ⚙️ ?

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

☐ File IO ?

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

☐ Memory Usage ⚙️ ?

Investigate application memory to find issues such as memory leaks

Start

Интеграция с AI



Visual Studio 2026



Возможности



Отладка



Профилирование



Тестирование



Модернизация



GitHub



Инструменты

Планирование

Microsoft Learn

NuGet

GitHub

MCP server

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)

46

{

47

std::cerr << "Listen failed" << std::endl;

48

return -1;

49

}

50

51

std::cout << "Server is listening on port " << port << std::endl;

52

53

while (true)

54

{

55

// Accept incoming connection

56

new_socket = accept(server_fd, (struct sockaddr*)&address, (socklen_t*)&addrlen);

57

if (new_socket < 0)

58

{

59

std::cerr << "Accept failed" << std::endl;

60

continue;

61

}

62

63

// Send HTTP response

64

send(new_socket, http_response.c_str(), http_response.size(), 0);

65

66

// Close the socket

67

close(new_socket);

68

}

69

70

Tab accept 0;

71

}

CS0103: The name 'listen' does not exist in the current context

CS0201: Only assignment, call, increment, decrement, await, and new object expressions can be used as a statement

CS0432: Alias 'std' not found

CS1002: ; expected

CS0103: The name 'new_socket' does not exist in the current context

CS0201: Only assignment, call, increment, decrement, await, and new object expressions can be used as a statement

CS0139: No enclosing loop out of which to break or continue

CS0103: The name 'send' does not exist in the current context

CS0103: The name 'close' does not exist in the current context

CS1022: Type or namespace definition, or end-of-file expected

CS1022: Type or namespace definition, or end-of-file expected

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help Search

ICSharpCode.SharpZipLib

Release Any CPU ICSharpCode.SharpZipLib ICSharpCode.SharpZipLib

Adler32.cs Adler32.cs Program.cs

ICSharpCode.SharpZipLib.Ben ICSharpCode.SharpZipLib.Ben MultipleRuntimes0

```
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     }
18
19     1 reference | 0 changes | 0 authors, 0 changes
20     class Program
21     {
22         0 references | 0 changes | 0 authors, 0 changes
23         static void Main(string[] args)
24         {
25             BenchmarkSwitcher.FromAssembly(System.Reflection
26             args,
27             DefaultConfig.Instance
28             .AddJob(BenchmarkDotNet.Jobs.Job.De
29             .AddDiagnoser(new Microsoft.VSDiagni
```

125 % No issues found Loc 12, Ch 15 SPC MIXED UTF-8 with BOM

Copilot Chat

I might make mistakes, so check for accuracy.

- Summarize my code
Explain how the code in #class works
- Write unit tests
Write unit tests for all the methods in #file
- Fix my code
How can I improve #method? Suggest code for any improvements

Start with @ to use an extension

/help what can you do?

Total changes

+ Active Document X

Ask Copilot or use @workspace

Agent • GPT-4.1 •

Output

Show output from: Build

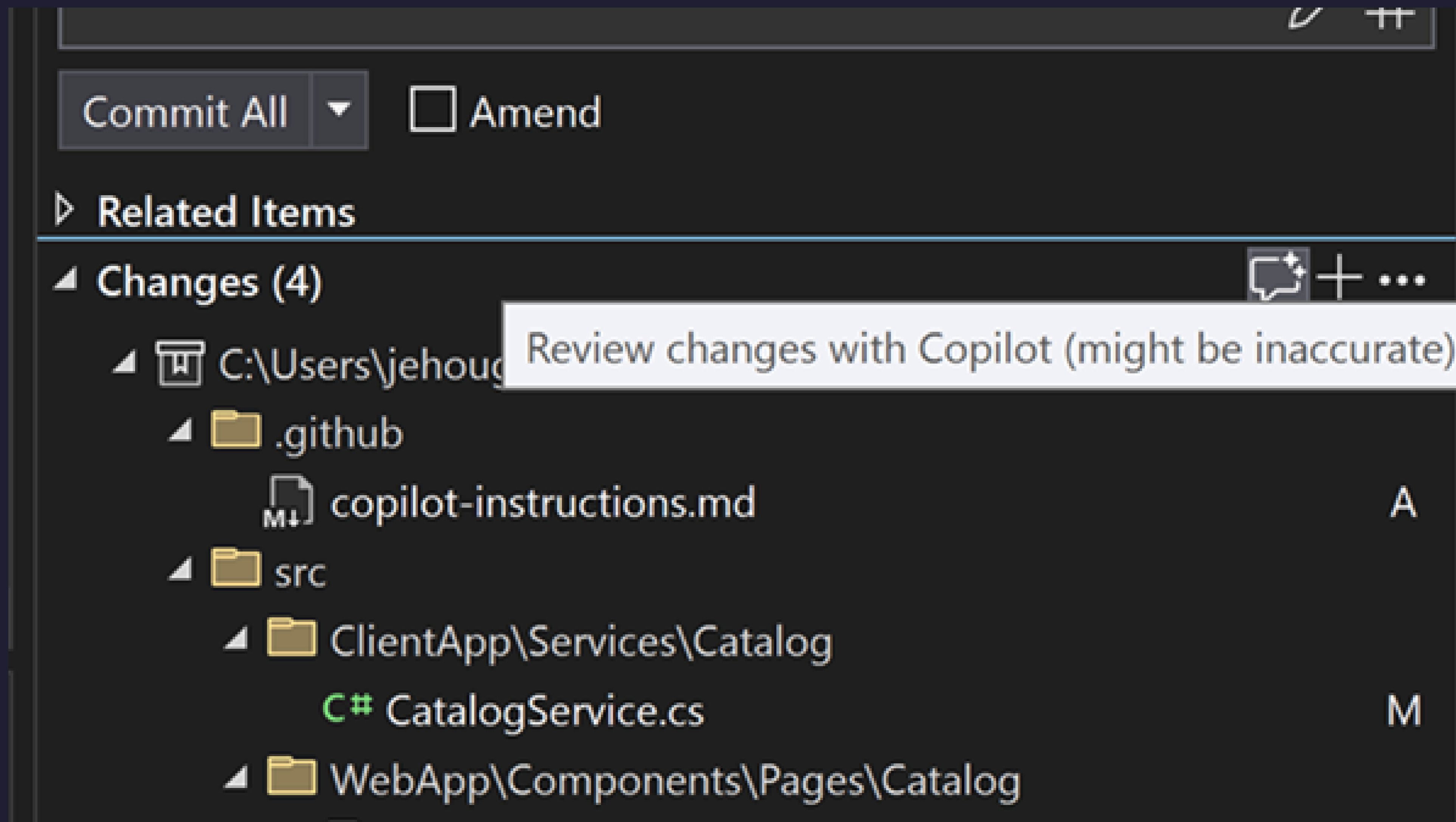
***** build: W succeeded, W failed, 3 up-to-date, W skipped *****

***** Build completed at 3:53 PM and took 00.070 seconds *****

Ready

0/0 2 master SharpZipLib

Ревью кода





Microsoft Visual Studio

INSIDERS


2026

MCP Server

GitHub Copilot Chat

Deploy to Azure

Deploy to Azure
> 1 reference

 GitHub Copilot


> 1 reference

I'll help you deploy this ASP.NET Core web application to Azure. Let me start by gathering some information about your project and then guide you through the deployment process.

✓ Read 1 project


✓ Ran verify_azure_publish_compatibility

Running Azure Best Practices

 Run Azure Best Practices

Tool arguments:

Name	Value
intent	Deploy ASP.NET Core web application to Azure

 This tool is from 'Azure MCP Server' (MCP Server). Note that MCP servers or malicious conversation content may attempt to misuse Visual Studio through tools. Please carefully review any requested actions.

Confirm

Deny

19

Модернизация

1. Оценка
2. Планирование
3. Исполнение



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

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.

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



Active Document



Ask Copilot



Agent



GPT-5 mini



.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

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

+ [D] Active Document X

Ask Copilot

⚡ Agent GPT-5 mini



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%)

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

App Modernization experience

> 5 tasks done

`BlazorSignalRApp.Client.... BlazorSignalRApp\Blazor...` -3 +1

`BlazorSignalRApp.Client.... BlazorSignalRApp\Blazor...` -2 +2

✓ Read `BlazorSignalRApp.csproj`, lines 1-120

`BlazorSignalRApp.csproj BlazorSignalRApp\Blazor...` -2 +2

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

...

▼ Total changes (2) Keep Undo ...

`BlazorSignalRApp.Client.csproj`

`BlazorSignalRApp.csproj`

+ [D] Active Document X

Ask Copilot

⚡ Agent ▼ GPT-5 mini ▼



[Release notes](#)



[Visual Studio
2026 is here](#)