C# 12 for Cloud, Web, and Desktop Applications

Modern concepts, and techniques for software development with C# 12

Thiago Vivas de Araujo

www.bpbonline.com

First Edition 2024

ISBN: 978-93-55519-023

LIMITS OF LIABILITY AND DISCLAIMER OF WARRANTY

The information contained in this book is true to correct and the best of author's and publisher's knowledge. The author has made every effort to ensure the accuracy of these publications, but publisher cannot be held responsible for any loss or damage arising from any information in this book.

All trademarks referred to in the book are acknowledged as properties of their respective owners but BPB Publications cannot guarantee the accuracy of this information.

Dedicated to

My beloved parents:

Carlos Roberto Coutinho de Araujo

Sonia Maria Pereira Vivas

About the Author

Thiago is a Brazilian expert passionate about software development and the complexity that comes with the process of constructing reliable software. Thiago has over 12 years of experience in software development and has worked in big companies like Oi, Accenture, Rio 2016 Olympic Games, Altice Portugal, Vodafone, and Microsoft.

Thiago, as a content creator, has already published more than 60 technical articles that have helped more than 2 million knowledge-seekers, writing mainly about C#, DevOps, and Cloud development.

About the Reviewers

- **Tomasz Kocurek** is a .NET developer. He graduated in Applied Computer Science in the Faculty of Electrical Engineering, Automatics, Computer Science, and Electronics at AGH University of Cracow. He has worked in various industries: artificial intelligence, creating bots, cryptocurrencies, virtual reality, medicine, electronic document circulation, data protection, transport, and navigation. He is keen on artificial intelligence, especially in healthcare. His master's thesis was blood detection using machine learning. He loves reading books. He has tried to find something interesting in every part of science. He likes to talk about new technologies. He has created documents, wiki, and a few applications with information about projects because data is priceless. He is interested in mathematics, playing bridge and football.

- **Denis Kazakov** has spent over 17 years in the software industry with a breadth of experience in different businesses and environments, from corporations to startups and freelance.

  He started as a developer and has worked through technical and development leadership roles. Today Denis's focus is on cloud technologies and development using Azure and .NET. Denis is the founder of the educational startup Similearn.com, where he works now. Denis regularly contributed to the IT community via forums of developers, open source projects, and blog posts. Denis has been awarded by Microsoft as a Most Valuable Professional four times in the Azure category

- **Dirk** is a seasoned software developer with over 19 years of experience utilizing C# and Visual Studio, who has had the privilege of working with various companies and learning from some of the most talented individuals in the industry. In addition to his professional experience, he has published multiple books on topics such as C#, Visual Studio, and ASP.NET Core. Dirk's passion for programming is unwavering, and he is dedicated to staying current with the latest technology and sharing his expertise with others.

Acknowledgement

I am immensely grateful to my beloved parents and sisters for their unconditional support. Their encouragement has sustained me through the ups and downs of this journey. Despite their bewilderment at the intricacies of software development, their unwavering belief in me has been a guiding light. Thanks for everything; I love you.

To my friends from Lisbon, Portugal, thank you for understanding my occasional disappearance while I was lost in the depths of this book. Your patience, camaraderie, and occasional provision of beer made the journey more bearable and the late nights more enjoyable. Cheers to each of you for being part of this adventure; the next drinks round is on me!

To my childhood friends from Complexo da Chumbada, Rio de Janeiro, Brazil, your unwavering support and camaraderie throughout the years have been a source of immense strength and inspiration. From the streets of our neighborhood to the pages of this book, your friendship has been a guiding light. Thank you for the countless memories, laughter, and shared adventures that have shaped who I am today. Your friendship is truly cherished, and I am grateful for the bond we share.

Preface

This book covers many different aspects of web scraping, the importance of automation of web scraping. This book also introduces the importance of web scraping in the field of real time industry. It shows how the data is important for the industries. This book solves the basic understanding of web scraping / crawling in the data world. It also gives importance for python learning. So that python's basic concepts get refreshed. This book gives information about the usefulness of Python in web scraping as well.

This book takes a practical approach for web scraping learners. It covers few real time industry examples as well. It will cover information such as Python Basically used for automation, machine learning. It can be used for easy data manipulating and transforming. Used in different domains for data mining purposes. You can design API access frameworks for end to end automation in different domains like finance, retail etc.

This book is divided into 14 chapters. They will cover Python basics, basics and advance in web scraping, why python is better for web scraping, real time examples in web scraping etc. So, learners can get more interest in web scraping tools as well. The details are listed below:

Chapter 1: Introduction to Visual Studio 2022 - Readers will explore the redesigned user interface, enhanced code analysis tools, and new debugging and testing functionalities. This chapter aims to equip developers with the knowledge and tools needed to leverage Visual Studio's latest features for enhanced productivity and efficiency.

Chapter 2: What is New in C# 12 - Whether experienced or new to C#, this chapter provides valuable insights into the evolving C# landscape, empowering developers to leverage the latest tools and advancements for efficient and effective coding.

Chapter 3: Mastering Entity Framework Core - It explores Database-First and Code-First approaches, covering schema creation and data model generation. It explains data modeling concepts, data annotations, and customization. The chapter also addresses data management tasks like querying and modifying data using LINQ to Entities, with examples.

Chapter 4: Getting Started with Azure Functions - covers triggers and bindings, exploring HTTP, Timer, Blob, and Queue triggers with detailed setup instructions and best practices. The chapter also explains how bindings facilitate interaction with Azure services like Blob Storage and Cosmos DB, demonstrating data access and manipulation through binding examples.

Chapter 5: Azure SQL, Cosmos DB and Blob Storage - providing step-by-step instructions and key concepts for each service. From creating and connecting to Azure SQL Database instances to understanding Cosmos DB APIs and working with Blob Storage containers, readers will gain valuable insights into leveraging these services effectively within their .NET applications.

Chapter 6: Unleashing the Power of Async Operations with Azure Service Bus - explores key concepts like queues, topics, and subscriptions, providing practical instructions and best practices for implementation. From understanding messaging patterns to configuring advanced features like message sessions and dead-letter queues, readers will gain valuable insights into building robust and scalable messaging solutions with Azure Service Bus.

Chapter 7: Securing Your Apps with Azure Key Vault - focus on authentication, the chapter guides readers through creating and configuring Key Vault instances, storing secrets, and accessing sensitive data securely. Step-by-step instructions and coverage of authentication options, including client certificates and Azure Active Directory authentication, equip developers with the knowledge and tools to ensure robust security measures for their applications.

Chapter 8: Building Dynamic Web Apps with Blazor and ASP.NET - Focusing on Blazor's key features like Hot Reload, Security, and Strongly-typed databinding, the chapter presents practical case studies to demonstrate their application in real-world scenarios. Starting with an overview of Blazor and its benefits, it covers project setup with ASP.NET and Visual Studio, exploring both client-side and server-side hosting models

Chapter 9: Real-time Communication with SignalR and ASP.NET - Starting with an overview of SignalR's benefits, it covers project setup with ASP.NET and Visual Studio, exploring various transport protocols like WebSockets and Server-Sent Events

Chapter 10: Implementing MicroServices with Web APIs - covers microservices architecture and the advantages of using Web APIs. It details designing scalable architectures, including service discovery, load balancing, and fault tolerance. Exploring scaling techniques like horizontal, vertical, and auto-scaling, it provides practical instructions, best practices, and pitfalls to avoid, empowering developers to create robust microservices architectures.

Chapter 11: CI/CD with Docker and Azure DevOps - Beginning with an overview of Docker and Azure DevOps, it demonstrates how to automate the CI-CD process. Covering Docker commands for building and deploying containerized apps, it explains CI-CD concepts and their role in streamlining development and deployment. The chapter walks through configuring build pipelines, release pipelines, and environment management using Azure DevOps

Chapter 12: Building Multi-platform Apps with .NET MAUI and Blazor Hybrid - explores its role in creating apps for various platforms. The chapter introduces Blazor Hybrid and its integration with .NET MAUI, comparing it with Blazor to highlight their respective benefits and drawbacks. It explains how Blazor Hybrid enables developers to build responsive UIs seamlessly integrated with .NET MAUI

Chapter 13: Windows UI Library: Crafting Native Windows Experience- Starting with an introduction to WinUI, it delves into how it facilitates the development of responsive desktop apps. Exploring the Fluent Design System's principles, including depth and motion, it demonstrates using WinUI's UI controls to implement engaging user interfaces

Chapter 14: Unit Testing and Debugging - craft effective unit tests using nUnit and xUnit for code reliability. Discover automated testing frameworks for better code quality. Dive into debugging with C# tools, mastering techniques like breakpoints and variable inspection for efficient error resolution, ultimately boosting developer productivity

Code Bundle and Coloured Images

Please follow the link to download the
Code Bundle and the ***Coloured Images*** of the book:

https://rebrand.ly/fcn7t93

The code bundle for the book is also hosted on GitHub at
https://github.com/bpbpublications/CSharp12-for-Cloud-Web-and-Desktop-Applications.
In case there's an update to the code, it will be updated on the existing GitHub repository.

We have code bundles from our rich catalogue of books and videos available at **https://github.com/bpbpublications**. Check them out!

Errata

We take immense pride in our work at BPB Publications and follow best practices to ensure the accuracy of our content to provide with an indulging reading experience to our subscribers. Our readers are our mirrors, and we use their inputs to reflect and improve upon human errors, if any, that may have occurred during the publishing processes involved. To let us maintain the quality and help us reach out to any readers who might be having difficulties due to any unforeseen errors, please write to us at :

errata@bpbonline.com

Your support, suggestions and feedbacks are highly appreciated by the BPB Publications' Family.

Piracy

If you come across any illegal copies of our works in any form on the internet, we would be grateful if you would provide us with the location address or website name. Please contact us at **business@bpbonline.com** with a link to the material.

If you are interested in becoming an author

If there is a topic that you have expertise in, and you are interested in either writing or contributing to a book, please visit **www.bpbonline.com**. We have worked with thousands of developers and tech professionals, just like you, to help them share their insights with the global tech community. You can make a general application, apply for a specific hot topic that we are recruiting an author for, or submit your own idea.

Reviews

Please leave a review. Once you have read and used this book, why not leave a review on the site that you purchased it from? Potential readers can then see and use your unbiased opinion to make purchase decisions. We at BPB can understand what you think about our products, and our authors can see your feedback on their book. Thank you!

For more information about BPB, please visit **www.bpbonline.com**.

Join our book's Discord space

Join the book's Discord Workspace for Latest updates, Offers, Tech happenings around the world, New Release and Sessions with the Authors:

https://discord.bpbonline.com

Table of Contents

Chapter 1
Introduction to Visual Studio 2022

Introduction

This chapter explores the latest features and improvements in Microsoft's interactive development environment. The new version of Visual Studio provides developers with better productivity and enhanced collaboration capabilities. The chapter covers the significant changes, including the introduction of the 64-bit architecture, improved performance, and better integration with Azure services.

This chapter highlights the significant updates of the IDE's user interface, which includes an updated start window, a redesigned search experience, and a refreshed iconography. This chapter also covers the enhanced code analysis capabilities, which include code suggestion, auto-correction, and intelligent code completion. Additionally, the chapter delves into new debugging and testing features, including live unit testing, snapshot debugging, and time travel debugging.

Structure

This chapter covers the following topics:

- Significant changes from Visual Studio 2019
- Live unit testing
- Snapshot debugging
- Time travelling debugging

Objectives

At the end of this chapter, readers will have a solid understanding of the main changes and new features introduced in Visual Studio 2022, including live unit testing, snapshot debugging, and time travelling debugging. They will be equipped with the knowledge necessary to leverage these tools effectively in their C# development workflows for cloud, web, and desktop applications.

Significant changes from Visual Studio 2019

Visual Studio 2022 brings significant improvements and enhancements compared to Visual Studio 2019. These include a more modern and intuitive user interface with refreshed visual themes, simplified toolbars, and menus. The performance has been optimized, offering faster operations such as build acceleration for .NET SDK style projects, faster "Find in Files" functionality, and improved Git tooling speed. Visual Studio 2022 is now a 64-bit application, allowing for better utilization of system resources. The introduction of new features like Hot Reload for code files, enhanced Commit Graph, and improved support for multiple repositories enhances the overall development experience. Additionally, there are advancements in the Razor and C# experiences, including improved code navigation, smarter IntelliCode suggestions, and better code formatting. With these changes, Visual Studio 2022 empowers developers with a more efficient, productive, and enjoyable development environment.

Performance improvements

Visual Studio 2022 introduces a range of performance improvements that enhance the development experience for developers. These improvements target various areas of the IDE, including build acceleration for .NET SDK style projects, external sources de-compilation, the Threads Window, Quick Add Item, code coverage, and Razor & C# experiences.

Let's delve into each of these enhancements:

- **Build acceleration for .NET SDK style projects**: Visual Studio 2022 incorporates optimizations for faster build times, specifically for .NET SDK style projects. By improving the build process up to 80%, developers experience reduced waiting times, allowing for quicker iterations and increased productivity.
- **External sources De-compilation**: This improvement enables developers to navigate and debug through code even when the source files are external 10x faster than Visual Studio 2019.
- **Find in files**: With enhanced search indexing, parallel processing, smarter search algorithms, search result previews, and advanced filtering options, developers can locate the desired information within their codebase 3x faster, reducing time spent on searching and improving overall productivity.

- **Git tooling**: Visual Studio 2022 introduces a new feature called the Commit Graph, which enhances the visualization and navigation of Git commits within the IDE. The Commit Graph in Visual Studio 2022 demonstrates a substantial improvement in performance, with an average loading time for branch history reduced by 70%.
- **Threads window**: The Threads Window in Visual Studio 2022 has undergone performance improvements, resulting in a more responsive and efficient debugging for applications with many threads. Developers can now effortlessly analyze and manage threads, gaining deeper insights into multi-threaded applications and improving overall debugging productivity.
- **Quick add item**: Visual Studio 2022 introduces the "New Item" menu command, which accelerates the process of adding new items to projects. With this enhancement, developers can swiftly add multiple files and folders streamlining the project creation and modification process.
- **Code coverage**: Code coverage analysis in Visual Studio 2022 has been optimized for 35% faster code coverage tests. Developers can now measure the code coverage of their tests more efficiently, allowing them to identify areas of code that lack test coverage and improve overall code quality.
- **Razor and C# experience**: Visual Studio 2022 brings support for code actions, including some helpful shortcuts that are vital for web development. These enhancements enhance the responsiveness and responsiveness of code editing, navigation, and IntelliSense features, enabling developers to write, modify, and navigate code more smoothly.

With these performance improvements in Visual Studio 2022, developers can expect a more efficient and responsive development environment. The build acceleration for .NET SDK style projects, external sources de-compilation, improvements to the Threads Window, Quick Add Item feature, enhanced code coverage analysis, and optimized Razor and C# experiences collectively enhance productivity and streamline the development process, enabling developers to deliver high-quality applications more efficiently.

64 -bit support

One of the significant enhancements in Visual Studio 2022 is its transition to a 64-bit application. This shift from a 32-bit to a 64-bit architecture brings several advantages and benefits to developers.

Let's explore the significance of Visual Studio 2022 being a 64-bit application:

- **Increased memory access**: As a 64-bit application, Visual Studio 2022 can take advantage of the expanded memory address space provided by 64-bit systems. This allows the IDE to access larger amounts of memory, enabling developers to work with larger and more complex projects without facing memory limitations or performance issues. Developers may now work with solutions containing more projects, capitalizing on the extended memory support offered by the 64-bit architecture.

- **Compatibility with 64-bit Tools and Libraries**: Being a 64-bit application, Visual Studio 2022 seamlessly integrates with other 64-bit tools, libraries, and components, ensuring compatibility and interoperability. Developers can take full advantage of the advancements and optimizations provided by 64-bit technologies in the development ecosystem.
- **Future-Proofing**: The transition to a 64-bit application positions Visual Studio 2022 for future growth and scalability. It aligns with the industry's shift towards 64-bit computing and ensures that the IDE can accommodate the evolving demands of modern development environments and technologies.

This transition empowers developers with a more robust and capable development environment, enabling them to tackle complex projects, streamline workflows, and deliver high-quality applications with greater efficiency.

Smarter IntelliCode

Program Synthesis using Examples (PROSE) technology is an automated program synthesis framework developed by Microsoft Research. It enables developers to generate code automatically based on input-output examples, allowing for rapid prototyping and development. PROSE leverages machine learning and programming language techniques to understand and generalize patterns from provided examples, reducing the manual effort required in writing code. It has been applied to various domains, including data wrangling, API usage, and code refactoring. PROSE aims to enhance developer productivity by automating repetitive coding tasks through the power of example-based program synthesis.

IntelliCode in Visual Studio 2022 actively analyzes your code changes as you type, leveraging PROSE technology to identify common patterns in manual code modifications. By recognizing how developers typically perform automatable code changes, IntelliCode offers helpful suggestions to streamline your coding process. When you're in the middle of a code fix or refactoring, IntelliCode presents suggestions from the completions list, allowing you to effortlessly complete the code change with a single click. This intelligent assistance provided by IntelliCode optimizes your coding experience, enabling you to work more efficiently and with greater accuracy.

Razor editor improvements

The new Razor editor introduces additional code fixes and refactoring, including the popular "Add missing usings" refactoring. It also offers Razor-specific refactoring like "Extract block to code behind" for easy extraction of code blocks to separate files.

Navigation support is improved with the "Go to Definition" feature for components, allowing quick navigation within files to understand code better.

Default colors in the new Razor editor have been updated, removing the code background highlight for better readability and reduced visual clutter.

The new Razor editor supports the latest compiler features, providing smarter Razor syntax completions such as "<text>" completion and auto-complete. Diagnostics are streamlined to show only the most important issues and maintain the intended fidelity of compiler-generated diagnostics.

Razor now fully integrates with Visual Studio Live Share, facilitating remote collaboration and code sharing among developers, supporting a shared context for efficient co-programming.

Hot reload with new capabilities

The introduction of the new Hot Reload technology brings seamless code file updates alongside XAML Hot Reload, catering to applications utilizing XAML for their UI. Whether you're working with XAML or .NET, Hot Reload is available, providing a powerful development experience. Hot Reload seamlessly integrates with familiar debugging capabilities like breakpoints, **'edit and continue' (EnC**), and other essential features.

In this release, Hot Reload is compatible with a wide range of application types. It works effortlessly with XAML-powered applications such as WPF, .NET MAUI and WinUI 3, as well as other frameworks like Windows Forms, ASP.NET web apps, Blazor Server, Console apps, and more. Essentially, wherever a modern .NET runtime is used in conjunction with the Visual Studio debugger, Hot Reload can significantly enhance your development workflow.

Multi-repository support

Visual Studio 2022 now offers support for multiple repositories, allowing you to work with up to 10 active Git repositories simultaneously. This feature enables seamless collaboration and efficient management of solutions that span across multiple repositories. You can perform Git operations across various repositories concurrently, streamlining your workflow.

For instance, in a large-scale web project, you might have separate repositories for the frontend, API, database, documentation, and various libraries or dependencies. Previously, managing work across these repositories required opening multiple instances of Visual Studio. However, with the multi-repository support, you can now handle, view, and debug all these repositories within a single instance of Visual Studio. This capability enhances productivity by eliminating the need to switch between different IDE instances and allows for a consolidated and cohesive development experience across multiple repositories.