

Learning .NET MAUI

*Unlock the potential of .NET MAUI
for Cross-Platform app development*

Aleksei Starkov



www.bpbonline.com

Copyright © 2023 BPB Online

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embedded in critical articles or reviews.

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the author, nor BPB Online or its dealers and distributors, will be held liable for any damages caused or alleged to have been caused directly or indirectly by this book.

BPB Online has endeavored to provide trademark information about all of the companies and products mentioned in this book by the appropriate use of capitals. However, BPB Online cannot guarantee the accuracy of this information.

First published: 2023

Published by BPB Online

WeWork

119 Marylebone Road

London NW1 5PU

UK | UAE | INDIA | SINGAPORE

ISBN 978-93-55518-804

www.bpbonline.com

Dedicated to

My beloved wife:

Ekaterina

&

My Son Stepan

About the Author

Aleksei Starkov is a seasoned software developer with over 10 years of experience in the tech industry. After 5 years of working on industrial automation projects that shaped his work ethic, he has been specializing in .NET, Xamarin, and MAUI since 2017. Aleksei's deep understanding of these technologies has been honed by participating in developing multiple commercial projects of different sizes and working for various software companies, from startups to large enterprises. In his free time, Aleksei spends time with his family and enjoys riding a motorcycle.

About the Reviewer

Anton Tereshko, a seasoned, certified software developer, holds a degree in Computer Software Engineering. His extensive career, spanning over eight years in the software development sector, includes three years of demonstrated leadership. He has proficiently led and streamlined the development processes for application services across numerous large-scale businesses operating in complex sectors such as medicine, transport & logistics, and insurance, utilizing his expert knowledge of MAUI, Xamarin, .NET, Flutter, and Microsoft Azure technologies.

Embracing the rise of AI/ML technologies, Anton is actively engaged in the development of applications featuring artificial intelligence. He holds a certificate in Azure Cognitive Service and continues to advance in this direction. In addition to his software development accomplishments, Anton is a serial entrepreneur, having co-founded multiple technology startups.

Acknowledgement

I want to express my deepest gratitude to my family for their endless support and encouragement throughout this book's writing, especially my wife Ekaterina and my son Stepan.

I am also grateful to the team of BPB Publications for their guidance, expertise, and professionalism. It was a long journey, with valuable participation and collaboration of reviewers, technical experts, and editors.

I would also like to acknowledge the valuable contributions of my colleagues, teammates, employers, and clients, who have taught me much and shaped me professionally.

Finally, I would like to thank all the readers who have taken an interest in my book.

Preface

In its time, Xamarin, the predecessor of .NET MAUI, played a crucial role in establishing a sustained trend for cross-platform GUI applications. Business owners, managers, and engineers were captivated by the opportunity to write code once and run it on various platforms and devices, all while leveraging the power of .NET.

When creating my first application with Xamarin, I used only official documentation and some blog posts from experienced colleagues. Time passed, and I worked on many different apps. Some of them I built alone, and for others, I worked as part of a group with more than 20 engineers. This book was born out of my desire to share my knowledge, passion, and experience to help others grow faster in their journey through the world of .NET MAUI.

In writing this book, my goal has been to provide as much useful theoretical and practical knowledge as possible to prepare you for work on real-world commercial projects while minimizing any confusion about the purpose of what you're learning.

Throughout the book, you will refresh your knowledge about .NET and C#, learn about the fundamentals and internals of .NET MAUI, its features, and how to use them to build real-world applications with custom UI and high-quality codebase that are reliable and easy to maintain. You will also learn about best practices and design patterns on the path through the mixture of theory, practical examples, and coding tasks.

This book is intended mostly for developers without commercial Xamarin and .NET MAUI experience and want to learn how to build GUI applications targeting different platforms and devices. However, it would also be helpful for developers with Xamarin experience who want to switch to .NET MAUI and improve their skills in building robust and reliable cross-platform applications.

Welcome aboard; our .NET MAUI adventure begins! I hope you'll enjoy it. See you on the other side.

Chapter 1: Getting Started with .NET MAUI - covers a .NET MAUI environment setup. The chapter introduces Visual Studio 2022 and the NuGet package manager.

Besides, this chapter shows how to create the blank application from the predefined template, and explains platform-specific debugging details and the anatomy of the .NET MAUI solution.

Chapter 2: Upskill Essentials of C# - covers the most important .NET and C# topics knowledge of that are crucial for understanding upcoming chapters. The chapter introduces the concept of modern .NET and explains numerous areas, including data types, Object-Oriented Programming, memory management and generics, and more.

Chapter 3: Exploring .NET MAUI and Its Features - covers the fundamentals of .NET MAUI, starting from the history of Xamarin to how MAUI works. It explains such crucial topics as main .NET MAUI classes, HostBuilder, and application lifecycle. This chapter is where the reader gets familiar with managing application assets and starts creating their first MAUI application.

Chapter 4: In and Out of UI Development - allows the reader to learn fundamental concepts related to user interface implementation. The chapter addresses the main challenges while creating applications targeting different platforms and devices. Within this chapter, the reader also learns the fundamental UI controls, layouts, binding and navigation mechanisms while constantly developing the application from the previous chapter. One of the main areas covered by Chapter 4 is the XAML declarative markup language mainly used in MAUI to describe the user interface.

Chapter 5: Layering with MVVM - specializes the application architecture explaining MVVM, Inversion of Control, addressing in-app navigation challenges, and more. It also covers the usage of third-party dependency injection containers, embedding the NoSQL database, and local persistent storage creation.

Chapter 6: Deep Dive into UI Design - covers more complex areas of UI development. Within this chapter, the reader turns its application into a multilanguage one that handles different color schemes and visual states reacting to data changes. This chapter is where the application UI appearance and behavior become more polished and sophisticated thanks to custom animations, custom controls, and a new group of built-in controls and features.

Chapter 7: Essentials and Community Toolkit - explains extremely helpful libraries nearly every commercial application uses. One of them introduces a bunch of useful, commonly used classes created by the community, like converters, behaviors, and AsyncRelayCommand. The other introduces support for unique

operating systems and platform APIs like Text-to-Speech, Gyroscope, or Secure Storage.

Chapter 8: Accessibility - covers the important topic of accessibility support. The chapter addresses the most popular accessibility issues and explains the tools and APIs provided by operating systems and .NET MAUI to make the application more accessible.

Chapter 9: Native Interactive Features with Shell and Blazor - introduces alternative approaches to .NET MAUI application development. During this chapter, the reader will turn its application into a Shell application learning the differences between the classic approach and Shell. Additionally, they will learn methods to optimize Shell for real-world commercial projects. Besides, this chapter covers Blazor Hybrid applications, their concept, anatomy, and superpower.

Code Bundle and Coloured Images

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

<https://rebrand.ly/2b1m6y2>

The code bundle for the book is also hosted on GitHub at **<https://github.com/bpbpublications/Learning-.NET-MAUI>**. 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.

Did you know that BPB offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at www.bpbonline.com and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at :

business@bpbonline.com for more details.

At **www.bpbonline.com**, you can also read a collection of free technical articles, sign up for a range of free newsletters, and receive exclusive discounts and offers on BPB books and eBooks.

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

1. Getting Started with .NET MAUI.....	1
Introduction.....	1
Structure.....	2
Objectives.....	2
Visual Studio 2022 installation.....	2
Visual Studio 2022 for Mac installation.....	6
Creating .NET MAUI project	8
Let's make it alive.....	10
Visual Studio IDE overview.....	16
Anatomy of .NET MAUI solution.....	18
NuGet package manager	19
Conclusion.....	21
Points to remember	22
Questions	22
2. Upskill Essentials of C#	23
Introduction.....	23
Structure.....	24
Objectives.....	24
.NET 6.....	25
FCL	25
CLR.....	25
C#	30
Value types.....	32
Reference types	35
Classes and objects	38
Class members	38
Memory management.....	39
Garbage collector	40

Strings.....	41
Interfaces.....	42
Inheritance	46
Access modifiers	47
Polymorphism.....	48
Generics.....	50
Delegates.....	51
Events	54
Anonymous methods.....	57
Async Await and TPL.....	58
Exception handling.....	59
IDisposable	60
Coding style.....	61
Conclusion	62
Points to remember	63
Questions	64
3. Exploring .NET MAUI and Its Features.....	65
Introduction.....	65
Structure.....	66
Objectives.....	66
Why did cross-platform development come up?	67
History of Xamarin and Mono.....	68
What does .NET MAUI change and why?.....	72
Where are Renderers?	73
Handlers.....	75
Building an app from a shared code to a native application.....	77
Application life cycle.....	80
Exploring App.xaml.cs.....	80
<i>Properties and methods</i>	81
HostBuilder pattern.....	83
MauiProgram.cs.....	84

MAUI application life cycle events.....	87
Platform-specific life cycle events	88
MessagingCenter	90
Dealing with image resources.....	96
Application icon.....	100
Splash screen	103
Fonts	104
Conclusion.....	106
Points to remember	106
Questions	107
4. In and Out of UI Development.....	109
Introduction.....	109
Structure.....	109
Objectives.....	110
Challenge of screens and sizes: pixels, points, and units.....	111
What is XAML about?.....	114
Markup extensions	121
UI - looking through the developer's eyes	123
Base types of UI elements.....	126
Content page	131
Image.....	131
Label	132
Text input controls.....	135
Button and ImageButton	139
Layouts overview	141
StackLayout.....	142
Bindings Mechanism.....	143
Attached properties.....	149
NavigationPage.....	149
Grid.....	153
BindableLayout and DataTemplate	157

Showing modal pages.....	162
Picker.....	164
Pop-ups.....	166
View-to-view binding.....	166
Resources.....	167
Styles.....	168
Conclusion.....	170
Points to remember.....	171
Questions.....	171
5. Layering with MVVM.....	173
Introduction.....	173
Structure.....	173
Objectives.....	174
What is MVVM?.....	174
ViewModel as a BindingContext.....	176
Commanding and ICommand interface.....	177
Navigation with MVVM.....	181
Dependency Inversion Principle.....	187
Inversion of Control.....	190
Dependency Injection.....	191
Third-Party DI Container Integration (StrongInject).....	194
Invoke platform code from shared code.....	196
LiteDB database.....	202
Local storage implementation.....	203
Conclusion.....	209
Points to remember.....	209
Questions.....	209
6. Deep Dive into UI Design.....	211
Introduction.....	211
Structure.....	211
Objectives.....	212

Handling system themes	212
Custom themes with DynamicResources	215
Multilanguage support.....	220
CarouselView	225
IndicatorView.....	230
Visual State Manager.....	231
Visual State Triggers.....	237
Animations	240
ContentView.....	243
Custom Controls.....	243
FlexLayout	248
Triggers.....	251
OnPlatform and OnIdiom.....	255
Brushes	256
Shadows.....	258
Behaviors.....	259
CollectionView	260
Customizing native controls.....	263
Conclusion.....	266
Points to remember	266
Questions	266
7. Essentials and Community Toolkit.....	267
Introduction.....	267
Structure.....	267
Objectives.....	268
Essentials library overview	268
Connectivity	271
Text-to-Speech (TTS).....	276
Gyroscope	277
Application preferences.....	282
Secure storage.....	283

Community Toolkit library overview.....	284
Snackbar.....	285
InvertedBoolConverter.....	288
NumericValidationBehavior.....	289
ValidationBehavior abstract class.....	290
ColorAnimationExtensions.....	291
UserStoppedTypingBehavior.....	291
AsyncRelayCommand.....	292
Conclusion.....	293
Points to remember.....	294
Questions.....	294
8. Accessibility.....	295
Introduction.....	295
Structure.....	296
Objectives.....	296
Overview.....	296
Font sizing.....	298
Screen readers overview.....	299
Enable TalkBack.....	300
Enable VoiceOver on Mac.....	300
Enable VoiceOver on iPhone/iPad.....	301
Enable Narrator on Windows.....	301
Accessibility Tree.....	301
Semantic properties.....	302
Description.....	303
Hint.....	303
HeadingLevel.....	304
SetSemanticFocus and announce.....	304
SemanticOrderView.....	304
Exclude from the accessibility tree.....	305
Good practices.....	306

Conclusion	307
Points to remember	307
Questions	308
9. Native Interactive Features with Shell and Blazor	309
Introduction.....	309
Structure.....	309
Objectives.....	310
Introduction to Shell.....	310
Shell class	312
Shell life cycle.....	317
Shell navigation routing	317
Passing data.....	320
Making Shell navigation strong.....	322
Base ViewModels.....	325
Bricks of Strong Shell	329
Strong route registration.....	330
ShellNavigationPathBuilder	333
StrongShellNavigation service.....	335
Shell pages	341
Shell Flyout.....	342
What is Blazor?.....	346
Blazor WebAssembly and Blazor Server.....	346
Blazor Hybrid.....	347
BlazorWebView	347
MAUI capabilities from Blazor code.....	348
Conclusion.....	349
Points to remember	350
Questions	350
Index	351-360

CHAPTER 1

Getting Started with .NET MAUI

Introduction

Let us imagine an enthusiastic mechanic who builds a car in the early 1900s in a backyard. They do it right on the ground under the open sky without precise tools, ergonomic toolboxes, and bright lights. Needless to say, ensuring the speed and quality of work requires significant efforts in such conditions.

Programmers from the 70s and 80s were in pretty similar situations. They had to use a set of separate hardware and software tools to write, compile, and debug the code they wrote.

Because of the same reasons why mechanics eventually moved from dusty backyards to shiny workshops, developers ended up with **Integrated Development Environments (IDE)**. The first IDE was created by *Anders Hejlsberg*, engineer of *Borland Ltd.*, in 1983 featuring an integrated code editor and compiler for the first time.

Microsoft Visual Studio is a modern IDE loved by millions of software developers around the world. It provides numerous handy development tools and features and supports multiple development platforms and frameworks. Since Microsoft Visual Studio is an official free IDE for the .NET **Multi-Platform App UI** development framework (**MAUI**). The main task of this chapter is to help you learn how to

establish an MAUI developer environment on both Windows and macOS operational systems, take a look around, and run the blank project of an MAUI application.

Structure

In this chapter, we will cover the following topics:

- Visual Studio 2022 installation
- Visual Studio 2022 for Mac installation
- Creating .NET MAUI project
- Let's make it alive
- Visual Studio IDE overview
- Anatomy of .NET MAUI solution
- NuGet package manager

Objectives

After going through this chapter, an essential set of development tools will be established, making your virtual workplace ready for work on MAUI projects. Besides, you'll be familiarised with the fundamentals of the Visual Studio IDE, MAUI project structure, and NuGet package manager.

Visual Studio 2022 installation

To install Visual Studio 2022 for Windows, go to the official website, that is, <https://visualstudio.microsoft.com/downloads/>, and download the free community version of Visual Studio 2022. At this stage, you might notice that there are three trims of Visual Studio:

- Community
- Professional
- Enterprise

The main idea here is 'the larger the team and project, the more likely it is that a more advanced version of IDE is needed'. The community version is an appropriate choice for students and individual developers. It contains all essential tools and functionalities, and it's free.

The tool you have downloaded is a Visual Studio Installer. It's an answer to the need to have a single place to manage installed Visual Studio versions and modules of each installed version:

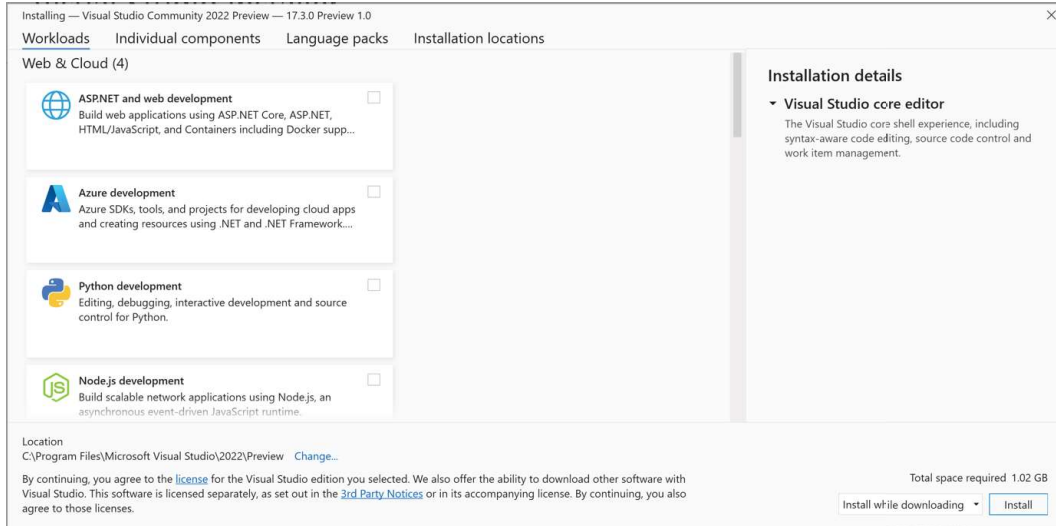


Figure 1.1: First launch of Visual Studio Installer

Since Visual Studio is a modular IDE that allows the creation of various kinds of software products, you can select only the modules you need to make it as lightweight as possible. Figure 1.1 represents the modules selection step of the installation wizard:

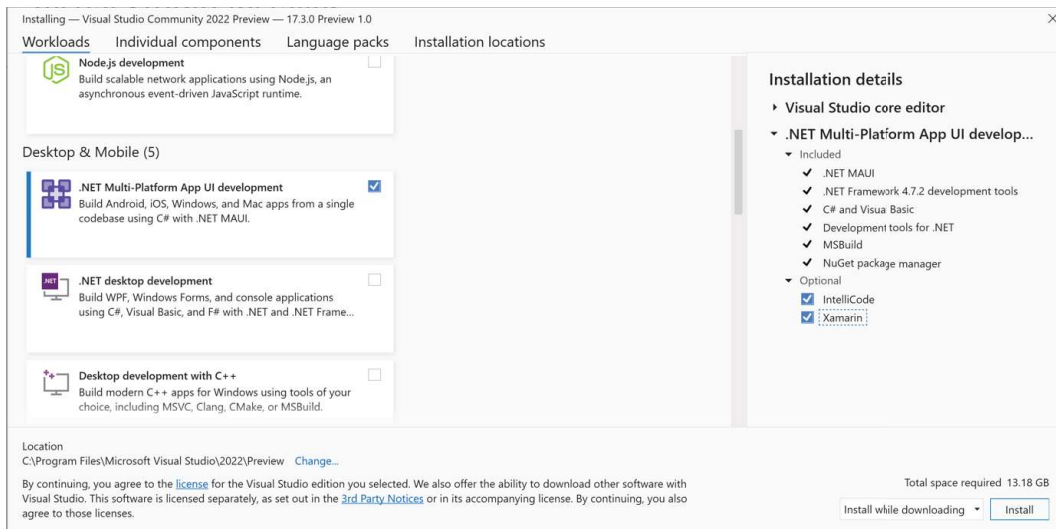


Figure 1.2: Selecting the MAUI module

Let's scroll it down and select **.NET Multi-Platform App UI development**, as shown in Figure 1.2. There are two important things you should pay attention to.

The first thing is the **Preview** version used. This book is being written in May 2022 and MAUI is still unavailable for the regular version of Visual Studio. So, if the regular

version of Visual Studio you downloaded doesn't contain the MAUI module, go to <https://visualstudio.microsoft.com/vs/preview/> and download the preview version of Visual Studio. However, it is believed that MAUI would have been included in stable Visual Studio 2022 before this book goes on sale.

The second thing is the **Xamarin checkbox** you might notice under the **Installation details** section. It's optional, but it is recommended to select it. In the later chapters, we will touch on the differences between Xamarin and MAUI. So, it might be helpful to play around Xamarin project for better understanding. You might probably not know what Xamarin is and how it relates to MAUI. We will talk about it in *Chapter 3, Exploring .NET MAUI and Its Features*.

Now, we are ready to click **Install**. Make sure you have a stable internet connection during the installation process:

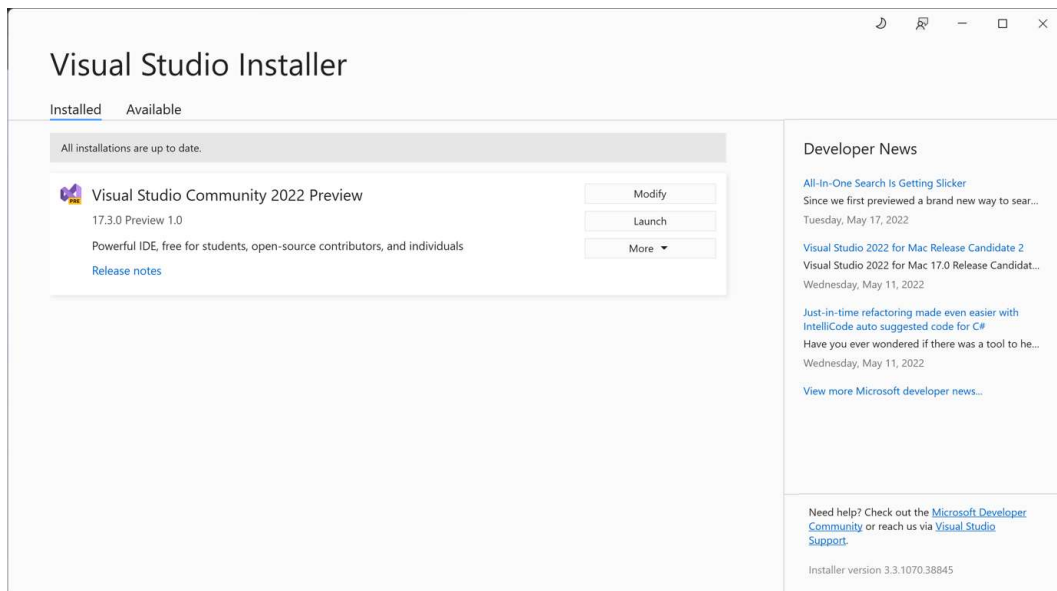


Figure 1.3: Visual Studio Installer with installed Visual Studio 2022

Figure 1.3 shows what the Visual Studio installer should look like after the installation. Call the installer from time to time to check if some updates are available. Besides, you can add or remove specific modules from the menu hidden under the **Modify** button.

The next step to take right after Visual Studio installation is checking the Android SDK Manager configuration. The Android SDK Manager helps developers control and update the installed native Android development tools, such as Android Emulator, build tools, drivers, and SDKs, to target different versions of the Android platform.

Launch the Visual Studio and choose **Continue without code**. Then, go to **Tools > Android > Android SDK Manager**, as shown in *Figure 1.4*:

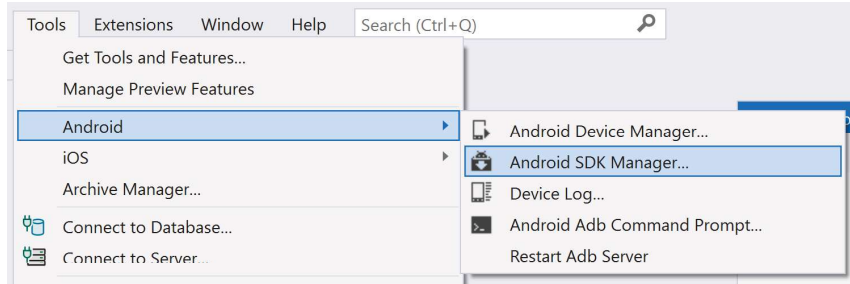


Figure 1.4: Visual Studio tools menu

Right after you have launched the Android SDK Manager, it might offer you to repair and reinstall SDK and SDK tools to make sure the latest versions are installed. Click the **Repair** button and follow the instructions.

General advice: Keep the SDK of the last two versions of Android installed. It's not a rare situation when you are forced to target the previous version of Android instead of the latest one because some other parts of your infrastructure, like the CI/CD pipelines of a build server, are not ready yet to work with the latest version of Android SDK. So, having a previous one hundred percent reliable version of SDK might be useful.

Figure 1.5 shows what the SDK manager should look like when it's configured and ready for work:

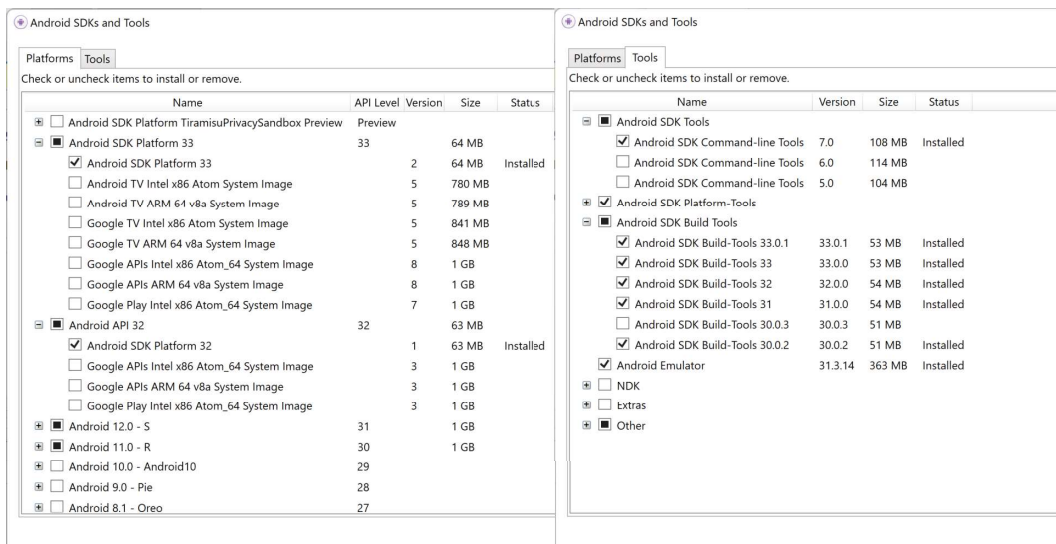


Figure 1.5: Final look after Android SDK installation