# Installation

Article • 01/06/2025

Developing native, cross-platform .NET Multi-platform App UI (.NET MAUI) apps requires Visual Studio 2022 17.12 or greater, or the latest Visual Studio Code with the .NET MAUI extension.

Visual Studio Code

To start developing native, cross-platform .NET MAUI apps on Windows, macOS, or Linux, install the latest Visual Studio Code by following the installation steps.

## **Prerequisites**

To build, sign, and deploy .NET MAUI apps for iOS and Mac Catalyst, you'll need:

- A Mac that is compatible with Xcode. For more information, see Apple's minimum requirements documentation .
- A specific version of Xcode, which depends on the version of .NET MAUI that you're using. For information, see Release versions .
- An Apple ID and paid Apple Developer Program enrollment. An Apple ID is required to deploy apps to devices, and to submit apps to the Apple Store.

## **Install Visual Studio Code**

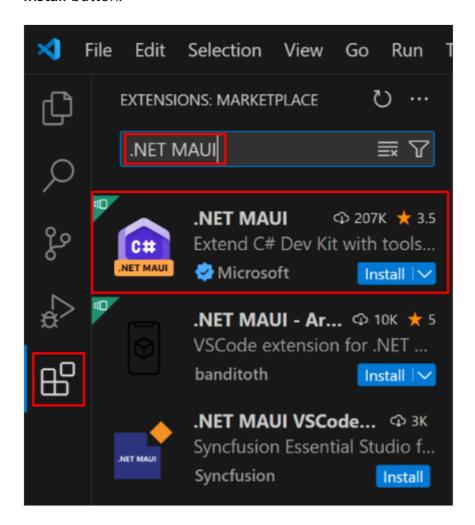
- 1. To create .NET MAUI apps, you'll need to download the latest version of Visual Studio Code:
  - Download Visual Studio Code
- 2. Install Visual Studio Code. For detailed instructions on how to install Visual Studio Code, see Visual Studio Code on Windows , Visual Studio Code on macOS , and Visual Studio Code on Linux .

## Install the .NET MAUI extension

Before you can create .NET MAUI apps in Visual Studio Code you'll need to install the .NET MAUI extension:

1. Launch Visual Studio Code.

2. In Visual Studio Code, navigate to the **Extensions** tab and search for ".NET MAUI". Then select the .NET MAUI extension and install it by pressing the **Install** button:

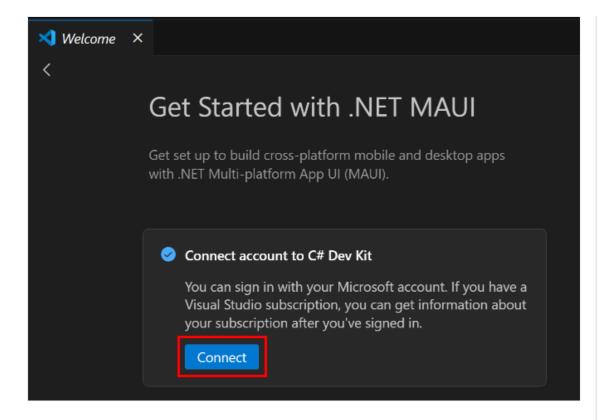


The .NET MAUI extension automatically installs the C# Dev Kit and C# extensions, which are required for the .NET MAUI extension to run. For more information about C# Dev Kit, see C# Dev Kit for Visual Studio Code.

## Connect your account to C# Dev Kit

Using C# Dev Kit requires you to sign in with a Microsoft account that has an active Visual Studio subscription:

1. In the **Welcome** tab for getting started with .NET MAUI, press the **Connect** button:



Follow the prompts to sign into your Microsoft account. For more information, see Signing in to C# Dev Kit .

For more information about C# Dev Kit licensing, see C# Dev Kit FAQ .

# Set up your .NET environment

You'll need the .NET SDK installed on your machine to develop .NET MAUI apps. If you don't have the .NET SDK installed on your machine, the preferred approach to installing it on Windows is through the Visual Studio Installer. For more information, see Installation.

Alternatively, to manually install the .NET SDK:

- 1. Download the .NET installer .
- 2. Install the .NET SDK by running the .NET installer. For more information, see Install .NET on Windows, Linux, and macOS.

∏ Tip

On Linux, you can install the .NET SDK using the <u>scripted installation</u> <u>instructions</u>.

To verify that the .NET SDK is installed:

- 1. Open a terminal.
- 2. In the terminal, run the following command:

```
.NET CLI

dotnet --version
```

You should see the version of the .NET SDK that you've installed.

① Note

It may be necessary to restart your machine before verifying that the .NET SDK is installed.

# Set up your .NET MAUI environment

You'll need the .NET MAUI SDK installed on your machine to develop .NET MAUI apps. If you don't have the .NET MAUI SDK installed on your machine, the preferred approach to installing it on Windows is through the Visual Studio Installer. For more information, see Installation.

Alternatively, to manually install the .NET MAUI SDK:

- 1. Open a terminal.
- 2. In the terminal on Windows, run the following command:

```
.NET CLI

dotnet workload install maui
```

In the terminal on macOS, run the following command:

```
.NET CLI
sudo dotnet workload install maui
```

In the terminal on Linux, run the following command:

.NET CLI

```
dotnet workload install maui-android
```

To verify that the .NET MAUI SDK is installed:

- 1. Open a terminal.
- 2. In the terminal, run the following command:

```
.NET CLI

dotnet workload list
```

On Windows and macOS, you should see the maui workload ID listed alongside the installed version. However, if you've installed it through the Visual Studio Installer on Windows the following workload IDs are listed:

```
android
maui-windows
maccatalyst
ios
```

On Linux, you should see the maui—android workload ID listed alongside the installed version.

# Set up target platforms

To build and debug a .NET MAUI app, you'll need to have a valid target platform relative to your development machine's operating system. The following table lists the supported target platforms on each operating system:

Expand table

Your Operating System	Supported Target Platforms	
Windows	Windows, Android	
macOS	Android, iOS, macOS	
Linux	Android	

Building a .NET MAUI app for Android, and for Apple platforms, requires you to perform additional set up and configuration.

## **Android**

To process to set up your machine for .NET MAUI development on Android with Visual Studio Code is:

- Download and install the Java SDK For more information, see Download and install the Java JDK.
- Download and install the Android SDK. For more information, see Download and install the Android SDK.
- Download and install an Android emulator. For more information, see
   Download and install an Android emulator.

#### Download and install the Java SDK

To download and install the Java SDK, and configure Visual Studio Code to use it:

1. Download and install Microsoft OpenJDK 17. For information about installing the OpenJDK, see Install the Microsoft Build of OpenJDK.

Alternatively, rather than manually downloading and installing the Java SDK, you can use the InstallAndroidDependencies build target to install the Java SDK (and the Android SDK). For more information, see Using the InstallAndroidDependencies target.

## (i) Important

Ensure that you note the location that the OpenJDK is installed to, as this is required in the next step.

- 2. Ensure that you've configured the path to OpenJDK via one of the following approaches:
  - a. Set the JAVA\_HOME environment variable to define the Java SDK path for your machine. This is the recommended approach, which defines the Java SDK path at the machine level.

(!)	Note
-----	------

If you install the OpenJDK on Windows via MSI, you can opt into the installer setting the JAVA\_HOME environmental variable.

- b. In Visual Studio Code, press CTRL+SHIFT+P on Windows, or CMD+SHIFT+P on macOS, and then select .NET MAUI: Configure Android, followed by Select Java SDK location to set the Java SDK path at the user/workspace level.
- c. Configure the Java SDK path in your .csproj file by setting the \$(JavaSdkDirectory) MSBuild property to the OpenJDK path. This will define the Java SDK path at the project level.

#### Download and install the Android SDK

To download and install the Android SDK, and configure Visual Studio Code to use it:

- 1. Download and install the Android SDK via one of the following approaches:
  - a. Download and install the Android SDK by creating a new .NET MAUI project and then use the InstallAndroidDependencies build target, which helps set up your Android environment. This is the recommended approach. For more information, see Using the InstallAndroidDependencies target.
  - b. Download and install the Android SDK on Windows by installing the .NET MAUI development workload in Visual Studio, and then creating and running a .NET MAUI app on Android. This process will ensure that the Android SDK and an Android emulator are installed.
  - c. Download and install the Android SDK through Android Studio. For more information, see Install Android Studio on developer.android.com.
  - d. Download and install the Android SDK through your preferred package manager on Linux.
- 2. Ensure that you've configured the path to the Android SDK via one of the following approaches:
  - a. Set the ANDROID\_HOME environment variable to define the Android SDK path for your machine. This is the recommended approach, which defines the Android SDK path at the machine level.
  - b. In Visual Studio Code, press CTRL+SHIFT+P on Windows, or CMD+SHIFT+P on macOS, and then select .NET MAUI: Configure Android, followed by Select Android SDK location to set the Android SDK path at the user/workspace level.

- c. Configure the Android SDK path in your .csproj file by setting the \$(AndroidSdkDirectory) MSBuild property to the Android SDK path. This will define the Android SDK path at the project level.
- 3. In Visual Studio Code, verify that your Android environment is configured correctly by pressing CTRL+SHIFT+P on Windows, or CMD+SHIFT+P on macOS, and then selecting .NET MAUI: Configure Android, followed by Refresh Android environment. Any detected errors must be addressed:
  - In the command palette, select .NET MAUI: Configure Android followed by both Select Android SDK location and Select Android SDK location and validate that they correctly point to installations of each. On Windows, if you install the SDKs via Visual Studio, OpenJDK will be located at C:\Program Files\Microsoft and the Android SDK will be located at C:\Program Files (x86)\Android\android-sdk.
  - Ensure that your Android SDK folder has sub-folders such as *build-tools*, *cmdline-tools*, and *platform-tools*.
  - Ensure that your OpenJDK folder has sub-folders such as *bin*, *lib*, and more.
  - Ensure that the ANDROID\_HOME environment variable is set to your Android SDK path.
  - Ensure that the JAVA\_HOME environment variable is set to the your Java SDK path.
  - If Android licenses haven't been accepted, in an elevated terminal navigate to your Android SDK's *cmdline-tools/{version}/bin* folder and run sdkmanager —licenses and then follow the CLI prompts.

## Using the InstallAndroidDependencies target

The recommended approach to installing the required dependencies for your .NET MAUI project on Android is to run the InstallAndroidDependencies MSBuild target. This target will install the Android SDK for you, if it isn't already installed.

In a terminal, create a new .NET MAUI project:

```
.NET CLI

dotnet new maui -n "MyMauiApp"
```

In a terminal, change directory to *MyMauiApp*, and build the app while specifying the InstallAndroidDependencies build target:

```
.NET CLI
```

```
cd MyMauiApp
dotnet build -t:InstallAndroidDependencies -f:net9.0-android -
p:AndroidSdkDirectory="/path/to/sdk" -
p:AcceptAndroidSDKLicenses=True
```

#### ① Note

The InstallAndroidDependencies MSBuild target can also install the Java SDK if the JavaSdkDirectory MSBuild property is provided.

#### In the command above:

- -p:AndroidSdkDirectory="/path/to/sdk" installs or updates Android dependencies to the specified absolute path. Suggested paths are %LOCALAPPDATA%/Android/Sdk on Windows, and \$HOME/Library/Android/sdk on macOS.
- -p:AcceptAndroidSDKLicenses=True accepts the required Android licenses for development.
- (optional) -p:JavaSdkDirectory="/path/to/sdk" installs the Java SDK to the specified absolute path.

Try to avoid using paths that contain spaces or non-ASCII characters.

#### Download and install an Android emulator

To download and install an Android emulator on which to run your apps:

- 1. In a terminal, navigate to the {YOUR\_ANDROID\_SDK\_FOLDER}/cmdline-tools/{version}/bin/.
- 2. In a terminal, use the sdkmanager command to download and install an Android emulator:

On Windows, run the following commands:

```
sdkmanager --install emulator
sdkmanager --install "system-images;android-
35;google_apis;x86_64"
```

On macOS, run the following commands:

#### Console

- ./sdkmanager --install emulator
- ./sdkmanager --install "system-images;android-
- 35;google\_apis;arm64-v8a"

#### ① Note

The above command assumes an Apple Silicon Mac. For an Intel Mac, replace arm64–v8a with x86\_64.

For more information about the sdkmanager command, see sdkmanager on developer.android.com.

3. In a terminal, use the avdmanager command to create a new Android emulator:

On Windows, run the following commands:

#### Console

avdmanager create avd -n MyAndroidVirtualDevice-API35 -k
"system-images;android-35;google\_apis;x86\_64"

On macOS, run the following commands:

#### Console

./avdmanager create avd -n MyAndroidVirtualDevice-API35 -k
"system-images;android-35;google\_apis;arm64-v8a"

#### ① Note

The above command assumes an Apple Silicon Mac. For an Intel Mac, replace arm64–v8a with x86\_64.

For more information about the avdmanager command, see avdmanager on developer.android.com.

## iOS and macOS

To set up your Mac for .NET MAUI development on iOS and Mac Catalyst with Visual Studio Code:

- 1. Install the version of Xcode that's required by the version of .NET MAUI that you're using. For information, see Release versions . The latest stable Xcode release can be downloaded from the Apple App Store .
- 2. In a terminal, run the following command to acquire the Xcode command line tools:

```
Console

xcode-select --install
```

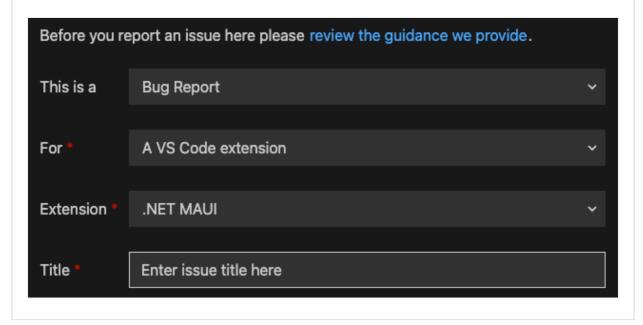
- 3. Launch Xcode and accept any license agreements. If simulators don't start installing, navigate to **Xcode** > **Settings** > **Components** and install your chosen simulator runtimes.
- 4. In Visual Studio Code, verify that your Apple environment is configured correctly by pressing CMD+SHIFT+P and then selecting .NET MAUI: Configure Apple, followed by Refresh Apple environment. Any detected errors must be addressed:
  - Ensure you've ran xcode-select —install in a terminal.
  - If you receive an error that Xcode hasn't been found, run xcode-select
     -p in a terminal and check that it returns a path to your Xcode installation.
  - Open Xcode to ensure it loads correctly, and then navigate to Xcode > Settings > Location and check that the Command Line Tools field is pointing to the correct Xcode installation.

# **Troubleshooting**

If you encounter issues when installing the .NET MAUI extension in Visual Studio Code, more information about the issues can be found by navigating to the **Output** window (CTRL+SHIFT+U) on Windows or CMD+SHIFT+U on macOS) and selecting .NET MAUI in the drop-down.

## Provide feedback

To provide feedback about the .NET MAUI extension from inside Visual Studio Code, navigate to the **Help > Report Issue** dialog. Then, ensure you select "Bug Report" as the value of the **This is a** drop-down, "A VS Code extension" as the value of the **For** drop-down, and ".NET MAUI" as the value of the **Extension** drop-down:



# **Next steps**

To learn how to create and run your first .NET MAUI app in Visual Studio 2022 on Windows, or Visual Studio Code, click the button below.

Build your first app