

Installing and getting started with flutter in vs code

Ok, so this is my general plan. Lets get started. First step is to install FLUTTER SDK. <https://docs.flutter.dev/get-started/install/windows> Found it here.



1. Download the following installation bundle to get the latest stable release of the Flutter SDK:

`flutter_windows_3.16.0-stable.zip`

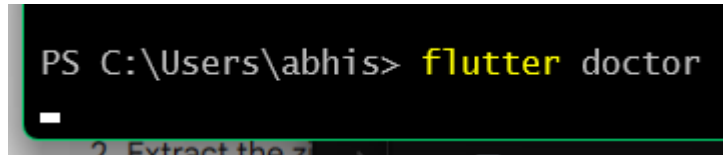
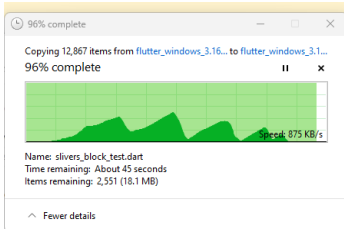
For other release channels, and older builds, check out the [SDK archive](#).

2. Extract the zip file and place the contained `flutter` in the desired installation location for the Flutter SDK (for example, `%USERPROFILE%\flutter, D:\dev\flutter`).

Warning: Do not install Flutter to a path that contains special characters or spaces.

Warning: Do not install Flutter in a directory like `C:\Program Files\` that requires elevated privileges.

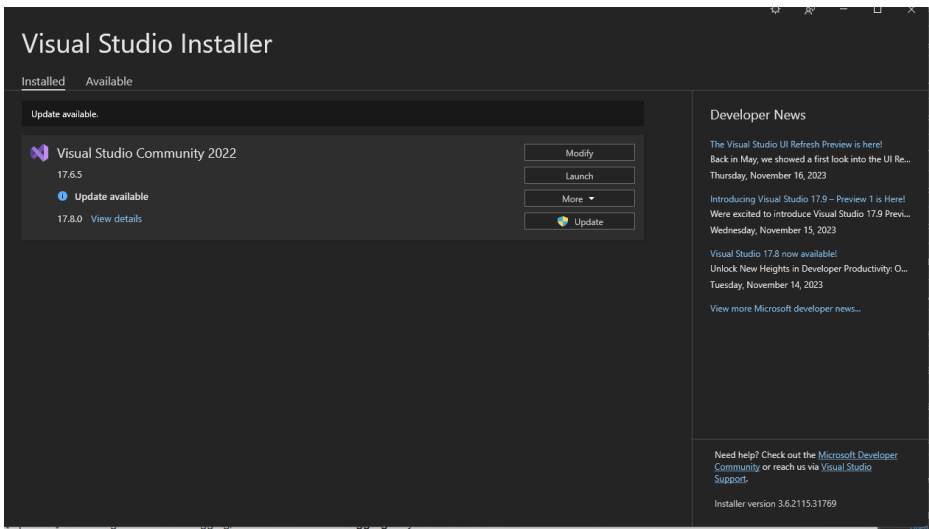
You are now ready to run Flutter commands in the Flutter Console.



```
PS C:\Users\abhis> flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 3.16.0, on Microsoft Windows [Version 10.0.22631.2715], locale en-US)
[✓] Windows Version (Installed version of Windows is version 10 or higher)
[!] Android toolchain - develop for Android devices (Android SDK version 33.0.2)
    X cmdline-tools component is missing
      Run 'path/to/sdkmanager --install "cmdline-tools;latest"'
      See https://developer.android.com/studio/command-line for more details.
    X Android license status unknown.
      Run 'flutter doctor --android-licenses' to accept the SDK licenses.
      See https://flutter.dev/docs/get-started/install/windows#android-setup for more details.
[✓] Chrome - develop for the web
[!] Visual Studio - develop Windows apps (Visual Studio Community 2022 17.6.5)
    X Visual Studio is missing necessary components. Please re-run the Visual Studio installer for the "Desktop
      development with C++" workload, and include these components:
        MSVC v142 - VS 2019 C++ x64/x86 build tools
        - If there are multiple build tool versions available, install the latest
        C++ CMake tools for Windows
        Windows 10 SDK
[✓] Android Studio (version 2022.2)
[✓] VS Code (version 1.84.2)
[✓] Connected device (3 available)
[✓] Network resources

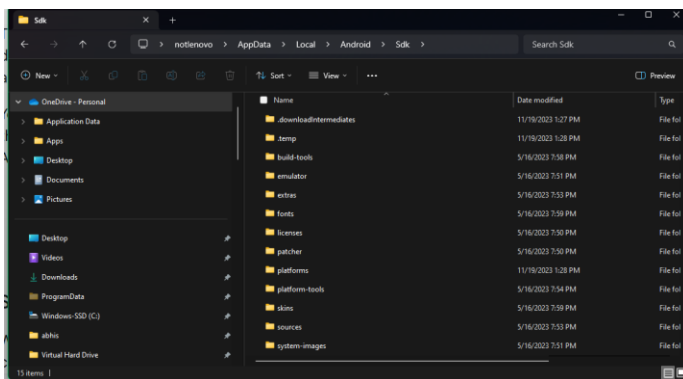
! Doctor found issues in 2 categories.
```

OK need to install the missing pieces. Lets head to Visual Studio first.



Found them and edited the installation!

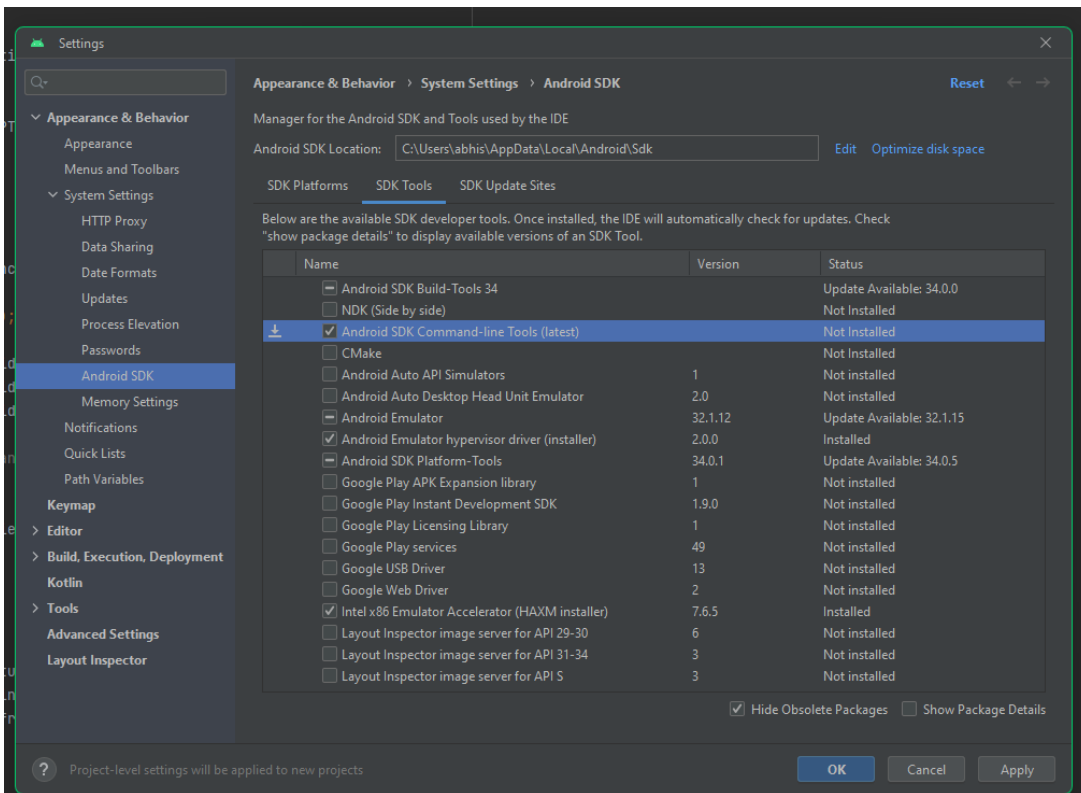
Now let's tackle Cmd line tools (The first one)



available, organized by the packages in which they're delivered.

You can install and update each package using Android Studio's [SDK Manager](#) or the `sdkmanager` command-line tool. All of the packages are downloaded into your Android SDK directory, which you can locate as follows:

1. In Android Studio, click **File > Project Structure**.
2. Select **SDK Location** in the left pane. The path is shown under **Android SDK location**.



Android sdkmanager not found. Update to the latest Android SDK and ensure that the cmdline-tools are installed. To resolve this, run: flutter doctor --android-licenses

```
PS C:\Users\abhis> flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 3.16.0, on Microsoft Windows [Version 10.0.22631.2715], locale en-US)
[✓] Windows Version (Installed version of Windows is version 10 or higher)
[!] Android toolchain - develop for Android devices (Android SDK version 33.0.2)
    ! Some Android licenses not accepted. To resolve this, run: flutter doctor --android-licenses
[✓] Chrome - develop for the web
[✓] Visual Studio - develop Windows apps (Visual Studio Community 2022 17.6.5)
[✓] Android Studio (version 2022.2)
[✓] VS Code (version 1.84.2)
[✓] Connected device (3 available)
[✓] Network resources
```

```
! Doctor found issues in 1 category.
PS C:\Users\abhis> flutter doctor --android-licenses
[=====] 100% Computing updates...
4 of 7 SDK package licenses not accepted.
Review licenses that have not been accepted (y/N)?
```

Path Variables	Google Play APK Expansion library	Version	Status
	<input type="checkbox"/>	1	Not installed
Keymap	<input type="checkbox"/>	1.9.0	Not installed

All SDK package licenses accepted

```
PS C:\Users\abhis> flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 3.16.0, on Microsoft Windows [Version 10.0.22631.2715], locale en-US)
[✓] Windows Version (Installed version of Windows is version 10 or higher)
[✓] Android toolchain - develop for Android devices (Android SDK version 33.0.2)
[✓] Chrome - develop for the web
[✓] Visual Studio - develop Windows apps (Visual Studio Community 2022 17.6.5)
[✓] Android Studio (version 2022.2)
[✓] VS Code (version 1.84.2)
[✓] Connected device (3 available)
[✓] Network resources
```

No issues found!

```
PS C:\Users\abhis>
```

! Some Android licenses not accepted. To resolve this, run: flutter doctor --android-licenses

```
[✓] Chrome - develop for the web
```

Install Android Studio

 [Help](#)

1. Download and install [Android Studio](#).
2. Start Android Studio, and go through the 'Android Studio Setup Wizard'. This installs the latest Android SDK, Android SDK Command-line Tools, and Android SDK Build-Tools, which are required by Flutter when developing for Android.
3. Run `flutter doctor` to confirm that Flutter has located your installation of Android Studio. If Flutter cannot locate it, run `flutter config --android-studio-dir=<directory>` to set the directory that Android Studio is installed to.

Set up the Android emulator

 [Help](#)

To prepare to run and test your Flutter app on the Android emulator, follow these steps:

1. Enable [VM acceleration](#) on your machine.
2. Launch **Android Studio**, click the **Device Manager** icon, and select **Create Device** under **Virtual** tab...
 - In older versions of Android Studio, you should instead launch **Android Studio > Tools > Android > AVD Manager** and select **Create Virtual Device....** (The **Android** submenu is only present when inside an Android project.)
 - If you do not have a project open, you can choose **3-Dot Menu / More Actions > Virtual Device Manager** and select **Create Device...**
3. Choose a device definition and select **Next**.
4. Select one or more system images for the Android versions you want to emulate, and select **Next**. An *x86* or *x86_64* image is recommended.
5. Under Emulated Performance, select **Hardware - GLES 2.0** to enable [hardware acceleration](#).
6. Verify the AVD configuration is correct, and select **Finish**.

For details on the above steps, see [Managing AVDs](#).

7. In Android Virtual Device Manager, click **Run** in the toolbar. The emulator starts up and displays the default canvas for your selected OS version and device.

Agree to Android Licenses

 [Help](#)

Before you can use Flutter, you must agree to the licenses of the Android SDK platform. This step should be done after you have installed the tools listed above.

1. Open an elevated console window and run the following command to begin signing licenses.

```
$ flutter doctor --android-licenses
```

2. Review the terms of each license carefully before agreeing to them.
3. Once you are done agreeing with licenses, run `flutter doctor` again to confirm that you are ready to use Flutter.

Set up an editor

[Get started](#) > Set up an editor

You can build apps with Flutter using any text editor or integrated development environment (IDE) combined with Flutter's command-line tools. The Flutter team recommends using an editor that supports a Flutter extension or plugin, like VS Code and Android Studio. These plugins provide code completion, syntax highlighting, widget editing assists, run & debug support, and more.

You can add a supported plugin for Visual Studio Code, Android Studio, or IntelliJ IDEA Community, Educational, and Ultimate editions. The [Flutter plugin](#) only works with Android Studio and the listed editions of IntelliJ IDEA.

(The [Dart plugin](#) supports eight additional JetBrains IDEs.)

Follow these procedures to add the Flutter plugin to VS Code, Android Studio, or IntelliJ.

If you choose another IDE, skip ahead to the [next step: Test drive](#).

Visual Studio Code Android Studio and IntelliJ

Install VS Code

[VS Code](#) is a code editor to build and debug apps. With the Flutter extension installed, you can compile, deploy, and debug Flutter apps.

To install the latest version of VS Code, follow Microsoft's instructions for the relevant platform:

- [Install on macOS](#)
- [Install on Windows](#)
- [Install on Linux](#)

Install the VS Code Flutter extension

1. Start **VS Code**.
2. Open a browser and go to the [Flutter extension](#) page on the Visual Studio Marketplace.
3. Click **Install**. Installing the Flutter extension also installs the Dart extension.

Validate your VS Code setup

1. Go to **View > Output**.

You can also press **Ctrl / Cmd + Shift + U**.

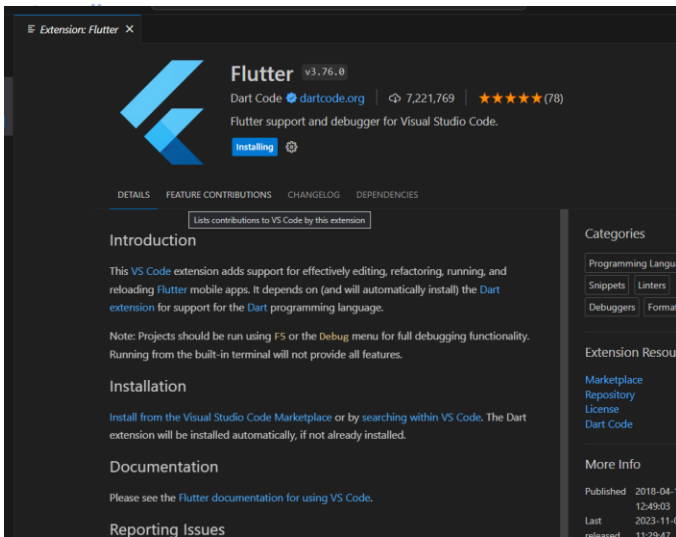
2. In the dropdown on the upper right of the **Output** panel, select **flutter (flutter)**.

3. Go to **View > Command Palette....**

You can also press **Ctrl / Cmd + Shift + P**.

4. Type **doctor**.

5. Select the **Flutter: Run Flutter Doctor**. Flutter Doctor runs and its response displays in the **Output** panel.



[Set up an editor](#)

[Write your first Flutter app](#)

Test drive

[Get started](#) > Test drive

This page describes the following tasks:

1. How to create a new Flutter app from templates.
2. How to run the created Flutter app.
3. How to use "hot reload" after you make changes to the app.

Details for these tasks depend on the integrated development environment (IDE) you use.

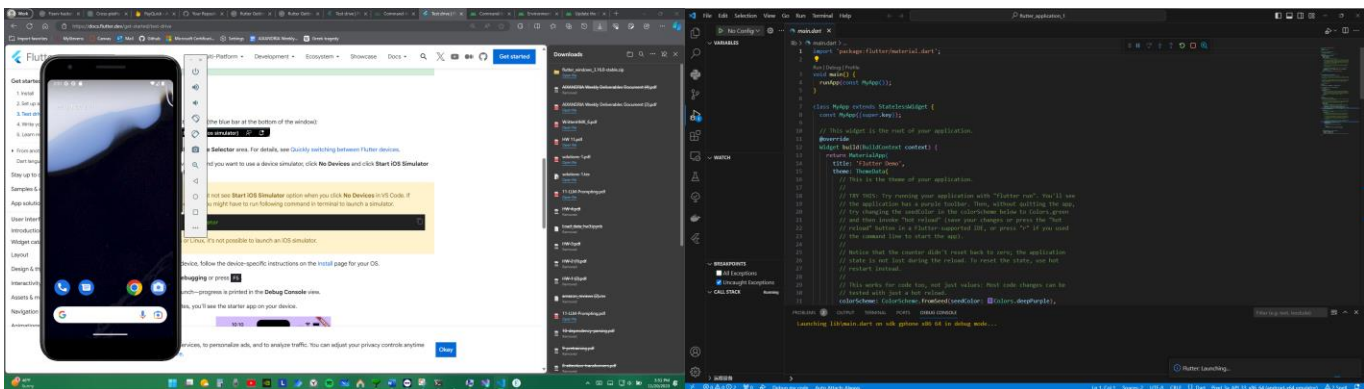
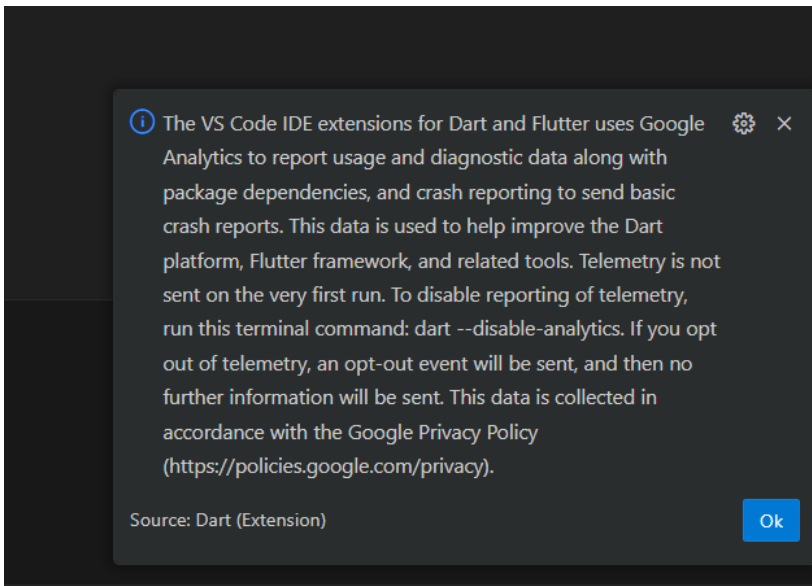
The first two options listed rely on the Flutter plugin for the respective IDE. Visual Studio Code, Android Studio, and IntelliJ IDEA Community, Educational, and Ultimate editions support Flutter development through plugins.

The third option explains how to use an editor of your choice and the terminal to run the commands.

Select your preferred IDE for Flutter apps.

Get started

1. Install
2. Set up an editor
3. **Test drive**
4. Write your first app
5. Learn more



This is really cool.

⚠ Warning: You might not see **Start iOS Simulator** option when you click **No Devices** in VS Code. If you're on Mac, then you might have to run following command in terminal to launch a simulator.

```
$ open -a simulator
```

On Windows or Linux, it's not possible to launch an iOS simulator.

Greedy pigs.

Flutter in 100 seconds

<https://www.youtube.com/watch?v=IHhRhPV--G0&>

Hot reload

Try hot reload

Flutter offers a fast development cycle with *Stateful Hot Reload*, the ability to reload the code of a live running app without restarting or losing app state. Make a change to app source, tell your IDE or command-line tool that you want to hot reload, and see the change in your simulator, emulator, or device.

1. Open `lib/main.dart`.
2. Change the string

```
'You have pushed the button this many times'
```

to

```
'You have clicked the button this many times'
```

Important: Do *not* stop your app. Let your app run.

3. Save your changes: invoke **Save All**, or click **Hot Reload** ⚡.

You'll see the updated string in the running app almost immediately.

Get involved with Flutter Gallery

<https://gallery.flutter.dev/#/>

