**Get Started with Flutter framework**

## System requirements:

To install and run Flutter, your development environment must meet these minimum requirements:

* **Operating Systems**: Windows 7 SP1 or later (64-bit)
* **Disk Space**: 400 MB (does not include disk space for IDE/tools).
* **Tools**: Flutter depends on these tools being available in your environment.
  + [Windows PowerShell 5.0](https://docs.microsoft.com/en-us/powershell/scripting/install/installing-windows-powershell) or newer (this is pre-installed with Windows 10)
  + [Git for Windows](https://git-scm.com/download/win) 2.x, with the **Use Git from the Windows Command Prompt** option.

If Git for Windows is already installed, make sure you can run git commands from the command prompt or PowerShell.

**Get the Flutter SDK:**

1. Download the following installation bundle to get the latest stable release of the Flutter SDK from below link: <https://storage.googleapis.com/flutter_infra/releases/stable/windows/flutter_windows_v1.12.13+hotfix.8-stable.zip>

Or you can download it from

<https://flutter.dev/docs/get-started/install/windows>

1. Extract the zip file and place the contained flutter in the desired installation location for the Flutter SDK (for example, C:\src\flutter; do not install Flutter in a directory like C:\Program Files\ that requires elevated privileges).

### Update path of environment variable:

Under **User variables** check if there is an entry called **Path**:

* If the entry exists, append the full path to flutter\bin using ; as a separator from existing values.
* If the entry doesn’t exist, create a new user variable named Path with the full path to flutter\bin as its value.

## Run flutter doctor:

Open commend prompt and run: cd “C:\src\flutter”

You will be in this “C:\src\flutter” directory. Now type: flutter doctor

And press Enter.

This command checks your environment and displays a report of the status of your Flutter installation. Check the output carefully for other software you might need to install or further tasks to perform (shown in **bold** text).

## Android setup:

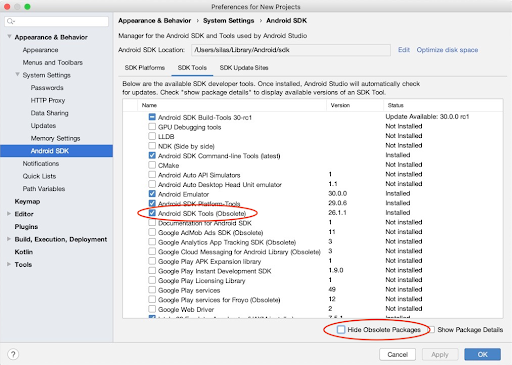
### Install Android Studio

1. Download and install <https://developer.android.com/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.

**Warning:** In Android Studio 3.6 or later, you need to manually add the old version of the Android SDK Tools for Flutter to work. To do this:

1. Open the **Android Studio SDK Manager**
2. In the Android SDK tab, uncheck **Hide Obsolete Packages**
3. Check **Android SDK Tools (Obsolete)**

The dialog below shows the appropriate settings:



This is a [known issue](https://github.com/flutter/flutter/issues/51712) that will be addressed in an upcoming version of Flutter.

### Set up your Android device:

To prepare to run and test your Flutter app on an Android device, you’ll need an Android device running Android 4.1 (API level 16) or higher.

1. Enable **Developer options** and **USB debugging** on your device. Detailed instructions are available in the [Android documentation](https://developer.android.com/studio/debug/dev-options).
2. Windows-only: Install the [Google USB Driver](https://developer.android.com/studio/run/win-usb).
3. Using a USB cable, plug your phone into your computer. If prompted on your device, authorize your computer to access your device.
4. In the terminal, run the flutter devices command to verify that Flutter recognizes your connected Android device. By default, Flutter uses the version of the Android SDK where your adb tool is based. If you want Flutter to use a different installation of the Android SDK, you must set the ANDROID\_HOME environment variable to that installation directory.

### Set up the Android emulator:

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

1. Enable [VM acceleration](https://developer.android.com/studio/run/emulator-acceleration) on your machine.
2. Launch **Android Studio > Tools > Android > AVD Manager** and select **Create Virtual Device**. (The **Android** submenu is only present when inside an Android project.)
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](https://developer.android.com/studio/run/emulator-acceleration).
6. Verify the AVD configuration is correct, and select **Finish**.

For details on the above steps, see [Managing AVDs](https://developer.android.com/studio/run/managing-avds).

1. 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.

**Run Project in the Emulator:**

## Check out the project from Git.

## Open the project in android studio marked menu.

## 

## Select the project folder which is downloaded from Git repository.

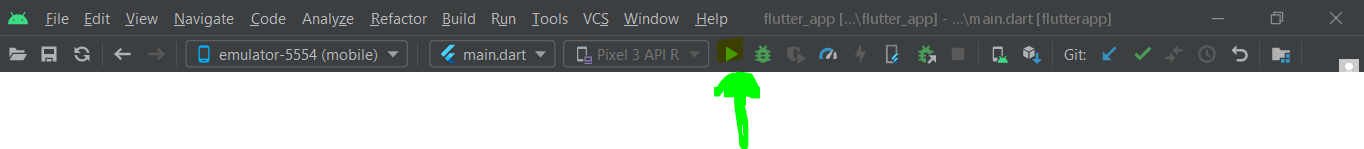
## Download all dependencies by clicking Get dependencies

## 

## Click here (marked (no devices)) and select your device/emulator

## 

1. Then click on the run icon for run the project in the device/emulator (show in the image given below).



Then the project will successfully run in the Emulator/device. Some screen shoot of the app:

