Windows install

1. [Get started](https://docs.flutter.dev/get-started)
2. [Install](https://docs.flutter.dev/get-started/install)
3. [Windows](https://docs.flutter.dev/get-started/install/windows)

Contents

* [System requirements](https://docs.flutter.dev/get-started/install/windows#system-requirements)
* [Get the Flutter SDK](https://docs.flutter.dev/get-started/install/windows#get-the-flutter-sdk)
  + [Update your path](https://docs.flutter.dev/get-started/install/windows#update-your-path)
  + [Run flutter doctor](https://docs.flutter.dev/get-started/install/windows#run-flutter-doctor)
* [Android setup](https://docs.flutter.dev/get-started/install/windows#android-setup)
  + [Install Android Studio](https://docs.flutter.dev/get-started/install/windows#install-android-studio)
  + [Set up your Android device](https://docs.flutter.dev/get-started/install/windows#set-up-your-android-device)
  + [Set up the Android emulator](https://docs.flutter.dev/get-started/install/windows#set-up-the-android-emulator)
  + [Agree to Android Licenses](https://docs.flutter.dev/get-started/install/windows#agree-to-android-licenses)
* [Windows setup](https://docs.flutter.dev/get-started/install/windows#windows-setup)
  + [Additional Windows requirements](https://docs.flutter.dev/get-started/install/windows#additional-windows-requirements)
* [Next step](https://docs.flutter.dev/get-started/install/windows#next-step)

System requirements

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

* **Operating Systems**: Windows 10 or later (64-bit), x86-64 based.
* **Disk Space**: 1.64 GB (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

*error* **Important:** If you’re in China, read [Using Flutter in China](https://docs.flutter.dev/community/china).

[helpHelp](https://docs.flutter.dev/get-started/install/help)

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

[flutter\_windows\_3.13.6-stable.zip](https://storage.googleapis.com/flutter_infra_release/releases/stable/windows/flutter_windows_3.13.6-stable.zip)

For other release channels, and older builds, check out the [SDK archive](https://docs.flutter.dev/release/archive).

1. 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).

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

*report\_problem* **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.

Update your path

[helpHelp](https://docs.flutter.dev/get-started/install/help#unable-to-find-the-flutter-command)

If you wish to run Flutter commands in the regular Windows console, take these steps to add Flutter to the PATH environment variable:

* From the Start search bar, enter ‘env’ and select **Edit environment variables for your account**.
* 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.

You have to close and reopen any existing console windows for these changes to take effect.

*info* **Note:** The Flutter SDK contains the dart command alongside the flutter command so that you can more easily run Dart command-line programs. Downloading the Flutter SDK also downloads the compatible version of Dart, but if you’ve downloaded the Dart SDK separately, make sure that the Flutter version of dart is first in your path, as the two versions might not be compatible. The following command tells you whether the flutter and dart commands originate from the same bin directory and are therefore compatible.

*content\_copy*

C:\>where flutter dart

C:\path-to-flutter-sdk\bin\flutter

C:\path-to-flutter-sdk\bin\flutter.bat

C:\path-to-dart-sdk\bin\dart.exe :: this should go after `C:\path-to-flutter-sdk\bin\` commands

C:\path-to-flutter-sdk\bin\dart

C:\path-to-flutter-sdk\bin\dart.bat

As shown above, the command dart from the Flutter SDK doesn’t come first. Update your path to use commands from C:\path-to-flutter-sdk\bin\ before commands from C:\path-to-dart-sdk\bin\ (in this case). After restarting your shell for the change to take effect, running the where command again should show that the flutter and dart commands from the same directory now come first.

*content\_copy*

C:\>where flutter dart

C:\dev\src\flutter\bin\flutter

C:\dev\src\flutter\bin\flutter.bat

C:\dev\src\flutter\bin\dart

C:\dev\src\flutter\bin\dart.bat

C:\dev\src\dart-sdk\bin\dart.exe

However, if you are using PowerShell, in it where is an alias of Where-Object command, so you need to use where.exe instead.

*content\_copy*

PS C:\> where.exe flutter dart

To learn more about the dart command, run dart -h from the command line, or see the [dart tool](https://dart.dev/tools/dart-tool) page.

Run flutter doctor

[helpHelp](https://docs.flutter.dev/get-started/install/help)

From a console window that has the Flutter directory in the path (see above), run the following command to see if there are any platform dependencies you need to complete the setup:

*content\_copy*

C:\src\flutter>flutter doctor

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).

For example:

*content\_copy*

[-] Android toolchain - develop for Android devices

• Android SDK at D:\Android\sdk

**✗ Android SDK is missing command line tools; download from https://goo.gl/XxQghQ**

• Try re-installing or updating your Android SDK,

visit https://docs.flutter.dev/setup/#android-setup for detailed instructions.

The following sections describe how to perform these tasks and finish the setup process. Once you have installed any missing dependencies, you can run the flutter doctor command again to verify that you’ve set everything up correctly.

*info* **Note:** If flutter doctor returns that either the Flutter plugin or Dart plugin of Android Studio are not installed, move on to [Set up an editor](https://docs.flutter.dev/get-started/editor?tab=androidstudio) to resolve this issue.

*report\_problem* **Warning:** The Flutter tool might occasionally download resources from Google servers. By downloading or using the Flutter SDK you agree to the [Google Terms of Service](https://policies.google.com/terms).

For example, when installed from GitHub (as opposed to from a prepackaged archive), the Flutter tool downloads the Dart SDK from Google servers immediately when first run, as it is used to execute the flutter tool itself. This also occurs when Flutter is upgraded (for example, by running the flutter upgrade command).

The flutter tool uses Google Analytics to report feature usage statistics and send [crash reports](https://github.com/flutter/flutter/wiki/Flutter-CLI-crash-reporting). This data is used to help improve Flutter tools over time.

Flutter tool analytics are not sent on the very first run. To disable reporting, run flutter config --no-analytics. To display the current setting, use flutter config. If you opt out of analytics, an opt-out event is sent, and then no further information is sent by the Flutter tool.

Dart tools might also send usage metrics and crash reports to Google. To control the submission of these metrics, use the following options on the [dart tool](https://dart.dev/tools/dart-tool):

* --enable-analytics: Enables anonymous analytics.
* --disable-analytics: Disables anonymous analytics.

The Google [Privacy Policy](https://policies.google.com/privacy) describes how data is handled by these services.

Android setup

*info* **Note:** Flutter relies on a full installation of Android Studio to supply its Android platform dependencies. However, you can write your Flutter apps in a number of editors; a later step discusses that.

Install Android Studio

[helpHelp](https://docs.flutter.dev/get-started/install/help#android-setup)

1. Download and install [Android Studio](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.
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 your Android device

[helpHelp](https://docs.flutter.dev/get-started/install/help#android-setup)

To prepare to run and test your Flutter app on an Android device, you need an Android device running Android 5.0 (API level 21) 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. [Optional] To leverage wireless debugging, enable **Wireless debugging** on your device. Detailed instructions are available in the [Android documentation](https://developer.android.com/studio/run/device#wireless).
3. Windows-only: Install the [Google USB Driver](https://developer.android.com/studio/run/win-usb).
4. Using a USB cable, plug your phone into your computer. If prompted on your device, authorize your computer to access your device.
5. 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\_SDK\_ROOT environment variable to that installation directory.

Set up the Android emulator

[helpHelp](https://docs.flutter.dev/get-started/install/help#android-setup)

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#accel-vm) 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](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.

Agree to Android Licenses

[helpHelp](https://docs.flutter.dev/get-started/install/help#android-setup)

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.

*content\_copy*

$ flutter doctor --android-licenses

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

Windows setup

Additional Windows requirements

[helpHelp](https://docs.flutter.dev/get-started/install/help)

For Windows desktop development, you need the following in addition to the Flutter SDK:

* [Visual Studio 2022](https://visualstudio.microsoft.com/downloads/) or [Visual Studio Build Tools 2022](https://visualstudio.microsoft.com/downloads/#build-tools-for-visual-studio-2022) When installing Visual Studio or only the Build Tools, you need the “Desktop development with C++” workload installed for building windows, including all of its default components.

*info* **Note:** **Visual Studio** is different than Visual Studio *Code*.

For more information, see [Building Windows apps](https://docs.flutter.dev/platform-integration/windows/building).

Windows install

1. [Get started](https://docs.flutter.dev/get-started)
2. [Install](https://docs.flutter.dev/get-started/install)
3. [Windows](https://docs.flutter.dev/get-started/install/windows)

Contents

* [System requirements](https://docs.flutter.dev/get-started/install/windows#system-requirements)
* [Get the Flutter SDK](https://docs.flutter.dev/get-started/install/windows#get-the-flutter-sdk)
  + [Update your path](https://docs.flutter.dev/get-started/install/windows#update-your-path)
  + [Run flutter doctor](https://docs.flutter.dev/get-started/install/windows#run-flutter-doctor)
* [Android setup](https://docs.flutter.dev/get-started/install/windows#android-setup)
  + [Install Android Studio](https://docs.flutter.dev/get-started/install/windows#install-android-studio)
  + [Set up your Android device](https://docs.flutter.dev/get-started/install/windows#set-up-your-android-device)
  + [Set up the Android emulator](https://docs.flutter.dev/get-started/install/windows#set-up-the-android-emulator)
  + [Agree to Android Licenses](https://docs.flutter.dev/get-started/install/windows#agree-to-android-licenses)
* [Windows setup](https://docs.flutter.dev/get-started/install/windows#windows-setup)
  + [Additional Windows requirements](https://docs.flutter.dev/get-started/install/windows#additional-windows-requirements)
* [Next step](https://docs.flutter.dev/get-started/install/windows#next-step)

System requirements

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

* **Operating Systems**: Windows 10 or later (64-bit), x86-64 based.
* **Disk Space**: 1.64 GB (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

*error* **Important:** If you’re in China, read [Using Flutter in China](https://docs.flutter.dev/community/china).

[helpHelp](https://docs.flutter.dev/get-started/install/help)

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

[flutter\_windows\_3.13.6-stable.zip](https://storage.googleapis.com/flutter_infra_release/releases/stable/windows/flutter_windows_3.13.6-stable.zip)

For other release channels, and older builds, check out the [SDK archive](https://docs.flutter.dev/release/archive).

1. 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).

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

*report\_problem* **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.

Update your path

[helpHelp](https://docs.flutter.dev/get-started/install/help#unable-to-find-the-flutter-command)

If you wish to run Flutter commands in the regular Windows console, take these steps to add Flutter to the PATH environment variable:

* From the Start search bar, enter ‘env’ and select **Edit environment variables for your account**.
* 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.

You have to close and reopen any existing console windows for these changes to take effect.

*info* **Note:** The Flutter SDK contains the dart command alongside the flutter command so that you can more easily run Dart command-line programs. Downloading the Flutter SDK also downloads the compatible version of Dart, but if you’ve downloaded the Dart SDK separately, make sure that the Flutter version of dart is first in your path, as the two versions might not be compatible. The following command tells you whether the flutter and dart commands originate from the same bin directory and are therefore compatible.

*content\_copy*

C:\>where flutter dart

C:\path-to-flutter-sdk\bin\flutter

C:\path-to-flutter-sdk\bin\flutter.bat

C:\path-to-dart-sdk\bin\dart.exe :: this should go after `C:\path-to-flutter-sdk\bin\` commands

C:\path-to-flutter-sdk\bin\dart

C:\path-to-flutter-sdk\bin\dart.bat

As shown above, the command dart from the Flutter SDK doesn’t come first. Update your path to use commands from C:\path-to-flutter-sdk\bin\ before commands from C:\path-to-dart-sdk\bin\ (in this case). After restarting your shell for the change to take effect, running the where command again should show that the flutter and dart commands from the same directory now come first.

*content\_copy*

C:\>where flutter dart

C:\dev\src\flutter\bin\flutter

C:\dev\src\flutter\bin\flutter.bat

C:\dev\src\flutter\bin\dart

C:\dev\src\flutter\bin\dart.bat

C:\dev\src\dart-sdk\bin\dart.exe

However, if you are using PowerShell, in it where is an alias of Where-Object command, so you need to use where.exe instead.

*content\_copy*

PS C:\> where.exe flutter dart

To learn more about the dart command, run dart -h from the command line, or see the [dart tool](https://dart.dev/tools/dart-tool) page.

Run flutter doctor

[helpHelp](https://docs.flutter.dev/get-started/install/help)

From a console window that has the Flutter directory in the path (see above), run the following command to see if there are any platform dependencies you need to complete the setup:

*content\_copy*

C:\src\flutter>flutter doctor

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).

For example:

*content\_copy*

[-] Android toolchain - develop for Android devices

• Android SDK at D:\Android\sdk

**✗ Android SDK is missing command line tools; download from https://goo.gl/XxQghQ**

• Try re-installing or updating your Android SDK,

visit https://docs.flutter.dev/setup/#android-setup for detailed instructions.

The following sections describe how to perform these tasks and finish the setup process. Once you have installed any missing dependencies, you can run the flutter doctor command again to verify that you’ve set everything up correctly.

*info* **Note:** If flutter doctor returns that either the Flutter plugin or Dart plugin of Android Studio are not installed, move on to [Set up an editor](https://docs.flutter.dev/get-started/editor?tab=androidstudio) to resolve this issue.

*report\_problem* **Warning:** The Flutter tool might occasionally download resources from Google servers. By downloading or using the Flutter SDK you agree to the [Google Terms of Service](https://policies.google.com/terms).

For example, when installed from GitHub (as opposed to from a prepackaged archive), the Flutter tool downloads the Dart SDK from Google servers immediately when first run, as it is used to execute the flutter tool itself. This also occurs when Flutter is upgraded (for example, by running the flutter upgrade command).

The flutter tool uses Google Analytics to report feature usage statistics and send [crash reports](https://github.com/flutter/flutter/wiki/Flutter-CLI-crash-reporting). This data is used to help improve Flutter tools over time.

Flutter tool analytics are not sent on the very first run. To disable reporting, run flutter config --no-analytics. To display the current setting, use flutter config. If you opt out of analytics, an opt-out event is sent, and then no further information is sent by the Flutter tool.

Dart tools might also send usage metrics and crash reports to Google. To control the submission of these metrics, use the following options on the [dart tool](https://dart.dev/tools/dart-tool):

* --enable-analytics: Enables anonymous analytics.
* --disable-analytics: Disables anonymous analytics.

The Google [Privacy Policy](https://policies.google.com/privacy) describes how data is handled by these services.

Android setup

*info* **Note:** Flutter relies on a full installation of Android Studio to supply its Android platform dependencies. However, you can write your Flutter apps in a number of editors; a later step discusses that.

Install Android Studio

[helpHelp](https://docs.flutter.dev/get-started/install/help#android-setup)

1. Download and install [Android Studio](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.
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 your Android device

[helpHelp](https://docs.flutter.dev/get-started/install/help#android-setup)

To prepare to run and test your Flutter app on an Android device, you need an Android device running Android 5.0 (API level 21) 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. [Optional] To leverage wireless debugging, enable **Wireless debugging** on your device. Detailed instructions are available in the [Android documentation](https://developer.android.com/studio/run/device#wireless).
3. Windows-only: Install the [Google USB Driver](https://developer.android.com/studio/run/win-usb).
4. Using a USB cable, plug your phone into your computer. If prompted on your device, authorize your computer to access your device.
5. 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\_SDK\_ROOT environment variable to that installation directory.

Set up the Android emulator

[helpHelp](https://docs.flutter.dev/get-started/install/help#android-setup)

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#accel-vm) 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](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.

Agree to Android Licenses

[helpHelp](https://docs.flutter.dev/get-started/install/help#android-setup)

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.

*content\_copy*

$ flutter doctor --android-licenses

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

Windows setup

Additional Windows requirements

[helpHelp](https://docs.flutter.dev/get-started/install/help)

For Windows desktop development, you need the following in addition to the Flutter SDK:

* [Visual Studio 2022](https://visualstudio.microsoft.com/downloads/) or [Visual Studio Build Tools 2022](https://visualstudio.microsoft.com/downloads/#build-tools-for-visual-studio-2022) When installing Visual Studio or only the Build Tools, you need the “Desktop development with C++” workload installed for building windows, including all of its default components.

*info* **Note:** **Visual Studio** is different than Visual Studio *Code*.

For more information, see [Building Windows apps](https://docs.flutter.dev/platform-integration/windows/building).