

Here are some terms that we use in the Flutter project and what they mean:

- **Cold restart.** Quitting and restarting an application and its associated runtime. This contrasts with hot restart, where the process and/or runtime would not be terminated.
- **Dynamic extension loading.** The ability to download some Dart code that wasn't written when the app was first published, which adds a new feature to the app. This could be done on the fly. It may require the core app to be larger since we can't know ahead of time what is needed by each future extension, and therefore can't do tree shaking (whole program optimization).
- **Dynamic patching.** The ability to update the Dart code of an app in the field by downloading a patch (of sorts) and providing it to the Dart VM, probably requiring a restart of the application.
- **Embedder.** The core of the Flutter engine is platform agnostic. But platforms that run Flutter applications need platform specific logic to wire up rendering to the native window toolkit, handle input events, etc.. This platform specific logic is referred to as the Flutter embedder. Embedders interact with the engine using a [very low level C/C++ API](#). While this API exposed by the engine to its embedders is platform agnostic, it is usually not suitable for developers used to the tools, frameworks and libraries on that specific platform (UIKit using Objective-C/Swift on iOS, or, Android APIs using Java on Android). To make interacting with the engine more ergonomic to developers on a specific platform, there are platform specific embedding APIs ([iOS](#) and [Android](#)). These APIs provide higher level abstractions but are optional.
- **Embedding.** Flutter provides "embedders" to port Flutter behavior to specific platforms, as defined elsewhere in this glossary. However, embedders use a [very low level C/C++ API](#) which would feel unnatural to typical developers of a platform, e.g., Android or iOS. A Flutter "embedding" is a counterpart to an "embedder", which introduces common platform tools for interacting with Flutter. The Android embedding offers a `FlutterActivity`, `FlutterFragment`, `FlutterView`, etc: [Android embedding](#). The iOS embedding offers a `FlutterViewController`, etc: [iOS embedding](#). An embedding + an embedder signifies the entire story of how Flutter ports to a specific platform.
- **Engine.** The C++ (and Java and ObjectiveC and...) side of Flutter. Defined in the `engine` repository. Includes Skia, Dart, and other low-level components. Exposed as `dart:ui` and other Dart libraries to author code.
- **Flakiness.** The tendency of a test to occasionally claim that it has passed even though it is really a failing test. Usually caused by race conditions. Please treat flaky tests as failures; attempt to find the cause and fix it.
- **Framework.** The `flutter` library (`packages/flutter` in the main repository) that includes the bulk of the Dart classes exposed by Flutter, such as all the built-in widgets.
- **Hot reload.** During development, causing a running application to update with new logic, without losing any of its state. This contrasts with hot restart because state is maintained despite the code being updated. We consider hot reload to be "hot" if it's subsecond in the normal case, significantly so in favourable conditions (e.g. fast devices, changing a small part of an application's UI).
- **Hot restart.** During development, causing a running application to rerun from the beginning, using new logic, and with all state reset to initial conditions. This contrasts with a cold restart because it does not require the application to shut down, and contrasts with a hot reload because state is lost in the process. We consider hot restart to be "hot" if it's under four seconds in the normal case.
- **Modular application delivery.** The ability to package a single app into multiple separate archives when compiling it, and download them independently as needed.

- **Skia.** The graphics library used by Flutter.
- **Tool.** The `flutter` program (started from `bin/flutter` on Unix systems, `bin/linux.bat` on Windows) that developers use to drive their work when using Flutter.