nouns, verbs, adjectives, adverbs

<https://medium.com/flutter/announcing-flutter-1-22-44f146009e5f>

We’re delighted to introduce our latest release of Flutter, with extensive support for iOS 14 and Android 11. Flutter 1.22 builds on the foundation set by previous releases by enabling developers to build fast, beautiful user experiences for multiple platforms from a single codebase. Our quarterly stable releases package the latest features, performance improvements, and bug fixes, and are suitable for broad production use.

Since this is the season for new mobile OS versions, this release focuses on ensuring that Android 11 and iOS 14 work great with Flutter. Updates for both of these OSes include a lot of under-the-hood work to conform to the latest SDKs and to make sure everything passes our extensive test suite. For iOS 14, this release includes support for the new Xcode 12, new icons, and preview support for the new iOS 14 App Clips feature. For Android 11, the update supports the new types of display cutouts as well as smoother animation when bringing up the soft keyboard.

This release comes two months after our 1.20 release, so it was shorter than most. Even in that short time, we closed 3,024 issues and merged 1,944 PRs from 197 contributors. Of those contributors, 114 (58%) of them were from the community-at-large and they contributed 271 PRs. The largest single contributor was a14n, who makes our top contributor list again with 20 PRs, most of which were done as part of the work to support null safety in Flutter (more on that coming soon).

In addition to the support for the new mobile OS versions, there’s quite a bit more news to share, including a preview of one of the top-requested features for Android: state restoration, a new “universe” of Material buttons, new international and localization support that works with hot reload, a new Navigator, a stable release for Platform Views (the foundation for the Google Maps and WebView plugins), and a switch you can throw in your code to improve scrolling on devices with high frequency displays. We’ve also got a new tool for dissecting app size and for ensuring that the plugins that you’re building support only the platforms that you want to support.

**Targeting iOS 14**

Whenever a new version of a mobile OS is announced, we test it thoroughly, looking for incompatibilities or changes that affect Flutter and its tools.

In the case of iOS 14, we made quite a few changes to Flutter to ensure that it works the way developers want:

* Xcode 12 requires iOS 9.0 or up, so our default template increases its default from 8.0 to 9.0
* iOS 14 specific crashes and font rendering issues were fixed in Flutter 1.22
* Problems deploying to physical devices were fixed as of Flutter 1.20.4
* A new policy that shows uses notifications when apps access their clipboard caused spurious notifications in Flutter apps, and was fixed as of Flutter 1.20.4
* A restriction disables running debug apps on iOS 14 devices except as part of the debugging process
* A new policy around network security for locally debugged Flutter apps causes iOS 14 to show a one-time confirmation dialog (only during development, not for released Flutter apps)

Bottom line: if you’re targeting iOS 14 with your Flutter app, we strongly encourage you to rebuild it with Flutter 1.22 and deploy it to the App Store now to ensure that your iOS 14 users have the best experience.

For more details about targeting iOS 14 with Flutter, including some Add-to-App, deep linking, and notification considerations, refer to the iOS 14 documentation on flutter.dev.

Hopefully, all of this work on the tooling and SDK support allows you to focus on the coding that you care about — taking advantage of new iOS 14 features.

One such feature is updated support for iOS’s new SF Symbols font, which inspired us to spend some time giving the cupertino\_icon package a refresh. Existing uses of CupertinoIcons will automatically map to the new style once you update your cupertino\_icons dependency to the new 1.0 major version. If you use cupertino\_icons 1.0 in conjunction with Flutter 1.22, you’ll also have access to ~900 new icons through the CupertinoIcons API.

You can see the complete list of icons on the cupertino\_icons preview page and a migration detail page on flutter.dev.

Another feature for you to try with Flutter on iOS 14 is App Clips, a new iOS 14 feature that supports quick, no-install app executions of lightweight versions of apps under 10MB. In Flutter version 1.22, we have a preview of App Clip targets built with Flutter.

For more details on how to build App Clips with Flutter, check out the docs on flutter.dev. You could also consult this simple sample project.

**Android 11**

This release of Flutter also coincides with the launch of Android 11 this month. The Flutter framework and engine have been updated to support two new features introduced in the latest version of Android.

Firstly, Flutter now supports exposing the safe insets of Android notches, cutouts and edges of waterfall displays.

By using the MediaQuery and SafeArea APIs, you can ensure that you’re placing active UI and interactive elements in the non-obstructed regions of the device’s display. Also, you’ll want to avoid gesture detectors in the waterfall edge area that may be prone to accidental touches.

Secondly, the animation is synchronized with Android 11 as it displays the software keyboard.