

# EXHIBIT FF

ANDROID

# open source project

[Home](#)[Source](#)[Compatibility](#)[Tech Info](#)[Community](#)[About](#)

## Frequently Asked Questions

### In This Document

#### [Frequently Asked Questions](#)

##### [Open Source](#)

[What is the Android Open Source Project?](#)[Why did we open the Android source code?](#)[What kind of open-source project is Android?](#)[Why is Google in charge of Android?](#)[What is Google's overall strategy for Android product development?](#)[How is the Android software developed?](#)[Why are parts of Android developed in private?](#)[When are source code releases made?](#)[What is involved in releasing the source code for a new Android version?](#)[How does the AOSP relate to the Android Compatibility Program?](#)[How can I contribute to Android?](#)[How do I become an Android committer?](#)

##### [Compatibility](#)

[What does "compatibility" mean?](#)[What is the role of Android Market in compatibility?](#)[What kinds of devices can be Android compatible?](#)[If my device is compatible, does it automatically have access to Android Market and branding?](#)[If I am not a manufacturer, how can I get Android Market?](#)[How can I get access to the Google apps for Android, such as Maps?](#)[Is compatibility mandatory?](#)[How much does compatibility certification cost?](#)[How long does compatibility take?](#)[Who determines what will be part of the compatibility definition?](#)[How long will each Android version be supported for new devices?](#)[Can a device have a different user interface and still be compatible?](#)[When are compatibility definitions released for new Android versions?](#)[How are device manufacturers' compatibility claims validated?](#)[What happens if a device that claims compatibility is later found to have compatibility problems?](#)

##### [Compatibility Test Suite](#)

[What is the purpose of the CTS?](#)[What kinds of things does the CTS test?](#)[Will the CTS reports be made public?](#)[How is the CTS licensed?](#)[Does the CTS accept contributions?](#)[Can anyone use the CTS on existing devices?](#)

## Open Source

### What is the Android Open Source Project?

We use the phrase "Android Open Source Project" or "AOSP" to refer to the people, the processes, and the source code that make up Android.

The people oversee the project and develop the actual source code. The processes refer to the tools and procedures we use to manage the development of the software. The net result is the source code that you can use to build cell phone and other devices.

### **Why did we open the Android source code?**

Google started the Android project in response to our own experiences launching mobile apps. We wanted to make sure that there would always be an open platform available for carriers, OEMs, and developers to use to make their innovative ideas a reality. We also wanted to make sure that there was no central point of failure, so that no single industry player could restrict or control the innovations of any other. The single most important goal of the Android Open-Source Project (AOSP) is to make sure that the open-source Android software is implemented as widely and compatibly as possible, to everyone's benefit.

You can find more information on this topic at our [Project Philosophy](#) page.

### **What kind of open-source project is Android?**

Google oversees the development of the core Android open-source platform, and works to create robust developer and user communities. For the most part the Android source code is licensed under the permissive Apache Software License 2.0, rather than a "copyleft" license. The main reason for this is because our most important goal is widespread adoption of the software, and we believe that the ASL2.0 license best achieves that goal.

You can find more information on this topic at our [Project Philosophy](#) and [Licensing](#) pages.

### **Why is Google in charge of Android?**

Launching a software platform is complex. Openness is vital to the long-term success of a platform, since openness is required to attract investment from developers and ensure a level playing field. However, the platform itself must also be a compelling product to end users.

That's why Google has committed the professional engineering resources necessary to ensure that Android is a fully competitive software platform. Google treats the Android project as a full-scale product development operation, and strikes the business deals necessary to make sure that great devices running Android actually make it to market.

By making sure that Android is a success with end users, we help ensure the vitality of Android as a platform, and as an open-source project. After all, who wants the source code to an unsuccessful product?

Google's goal is to ensure a successful ecosystem around Android, but no one is required to participate, of course. We opened the Android source code so anyone can modify and distribute the software to meet their own needs.

### **What is Google's overall strategy for Android product development?**

We focus on releasing great devices into a competitive marketplace, and then incorporate the innovations and enhancements we made into the core platform, as the next version.

In practice, this means that the Android engineering team typically focuses on a small number of "flagship" devices, and develops the next version of the Android software to support those product launches. These flagship devices absorb much of the product risk and blaze a trail for the broad OEM community, who follow up with many more devices that take advantage of the new features. In this way, we make sure that the Android platform evolves according to the actual needs of real-world devices.

### **How is the Android software developed?**

Each platform version of Android (such as 1.5, 1.6, and so on) has a corresponding branch in the open-source tree. At any given moment, the most recent such branch will be considered the "current stable" branch version. This current stable branch is the one that manufacturers port to their devices. This branch is kept suitable for release at all times.

Simultaneously, there is also a "current experimental" branch, which is where speculative contributions, such as large next-generation features, are developed. Bug fixes and other contributions can be included in the current stable branch from the experimental branch as appropriate.

Finally, Google works on the next version of the Android platform in tandem with developing a flagship device. This branch pulls in changes from the experimental and stable branches as appropriate.

You can find more information on this topic at our [Branches and Releases](#).

### **Why are parts of Android developed in private?**

It typically takes over a year to bring a device to market, but of course device manufacturers want to ship the latest software they can. Developers, meanwhile, don't want to have to constantly track new versions of the platform when writing apps. Both groups experience a tension between shipping products, and not wanting to fall behind.

To address this, some parts of the next version of Android including the core platform APIs are developed in a private branch. These APIs constitute the next version of Android. Our aim is to focus attention on the current stable version of the Android source code, while we create the next version of the platform as driven by flagship Android devices. This allows developers and OEMs to focus on a single version without having to track unfinished future work just to keep up. Other parts of the Android system that aren't related to application compatibility are developed in the open, however. It's our intention to move more of these parts to open development over time.

### **When are source code releases made?**

When they are ready. Some parts of Android are developed in the open, so that source code is always available. Other parts are developed first in a private tree, and that source code is released when the next platform version is ready.

In some releases, core platform APIs will be ready far enough in advance that we can push the source code out for an early look in advance of the device's release; however in others, this isn't possible. In all cases, we release the platform source when we feel the version has stabilized enough, and when the development process permits. Releasing the source code is a fairly complex process.

### **What is involved in releasing the source code for a new Android version?**

Releasing the source code for a new version of the Android platform is a significant process. First, the software gets built into a system image for a device, and put through various forms of certification, including government regulatory certification for the regions the phones will be deployed. It also goes through operator testing. This is an important phase of the process, since it helps shake out a lot of software bugs.

Once the release is approved by the regulators and operators, the manufacturer begins mass producing devices, and we turn to releasing the source code.

Simultaneous to mass production the Google team kicks off several efforts to prepare the open source release. These efforts include final API changes and documentation (to reflect any changes that were made during qualification testing, for example), preparing an SDK for the new version, and launching the platform compatibility information.

Also included is a final legal sign-off to release the code into open source. Just as open source contributors are required to sign a Contributors License Agreement attesting to their IP ownership of their contribution, Google too must verify that it is clear to make contributions.

Starting at the time mass production begins, the software release process usually takes around a month, which often roughly places source code releases around the same time that the devices reach users.

### **How does the AOSP relate to the Android Compatibility Program?**

The Android Open-Source Project maintains the Android software, and develops new versions. Since it's open-source, this software can be used for any purpose, including to ship devices that are not compatible with other devices based on the same source.

The function of the Android Compatibility Program is to define a baseline implementation of Android that is compatible with third-party apps written by developers. Devices that are "Android compatible" may participate in the Android ecosystem, including Android Market; devices that don't meet the compatibility requirements exist outside that

ecosystem.

In other words, the Android Compatibility Program is how we separate "Android compatible devices" from devices that merely run derivatives of the source code. We welcome all uses of the Android source code, but only Android compatible devices -- as defined and tested by the Android Compatibility Program -- may participate in the Android ecosystem.

### **How can I contribute to Android?**

There are a number of ways you can contribute to Android. You can report bugs, write apps for Android, or contribute source code to the Android Open-Source Project.

There are some limits on the kinds of code contributions we are willing or able to accept. For instance, someone might want to contribute an alternative application API, such as a full C++-based environment. We would decline that contribution, since Android is focused on applications that run in the Dalvik VM. Alternatively, we won't accept contributions such as GPL or LGPL libraries that are incompatible with our licensing goals.

We encourage those interested in contributing source code to contact us via the AOSP Community page prior to beginning any work. You can find more information on this topic at the [Getting Involved](#) page.

### **How do I become an Android committer?**

The Android Open Source Project doesn't really have a notion of a "committer". All contributions -- including those authored by Google employees -- go through a web-based system known as "gerrit" that's part of the Android engineering process. This system works in tandem with the git source code management system to cleanly manage source code contributions.

Once submitted, changes need to be accepted by a designated Approver. Approvers are typically Google employees, but the same approvers are responsible for all submissions, regardless of origin.

You can find more information on this topic at the [Submitting Patches](#) page.

## **Compatibility**

### **What does "compatibility" mean?**

We define an "Android compatible" device as one that can run any application written by third-party developers using the Android SDK and NDK. We use this as a filter to separate devices that can participate in the Android app ecosystem, and those that cannot. Devices that are properly compatible can seek approval to use the Android trademark. Devices that are not compatible are merely derived from the Android source code and may not use the Android trademark.

In other words, compatibility is a prerequisite to participate in the Android apps ecosystem. Anyone is welcome to use the Android source code, but if the device isn't compatible, it's not considered part of the Android ecosystem.

### **What is the role of Android Market in compatibility?**

Devices that are Android compatible may seek to license the Android Market client software. This allows them to become part of the Android app ecosystem, by allowing users to download developers' apps from a catalog shared by all compatible devices. This option isn't available to devices that aren't compatible.

### **What kinds of devices can be Android compatible?**

The Android software can be ported to a lot of different kinds of devices, including some on which third-party apps won't run properly. The Android Compatibility Definition Document (CDD) spells out the specific device configurations that will be considered compatible.

For example, though the Android source code could be ported to run on a phone that doesn't have a camera, the CDD requires that in order to be compatible, all phones must have a camera. This allows developers to rely on a consistent set of capabilities when writing their apps.

The CDD will evolve over time to reflect market realities. For instance, the 1.6 CDD only allows cell phones, but the 2.1 CDD allows devices to omit telephony hardware, allowing for non-phone devices such as tablet-style music players to be compatible. As we make these changes, we will also augment Android Market to allow developers to retain control over where their apps are available. To continue the telephony example, an app that manages SMS text messages would not be useful on a media player, so Android Market allows the developer to restrict that app exclusively to phone devices.

### **If my device is compatible, does it automatically have access to Android Market and branding?**

Android Market is a service operated by Google. Achieving compatibility is a prerequisite for obtaining access to the Android Market software and branding. Device manufacturers should contact Google to obtain access to Android Market.

### **If I am not a manufacturer, how can I get Android Market?**

Android Market is only licensed to handset manufacturers shipping devices. For questions about specific cases, contact [android-partnerships@google.com](mailto:android-partnerships@google.com).

### **How can I get access to the Google apps for Android, such as Maps?**

The Google apps for Android, such as YouTube, Google Maps and Navigation, Gmail, and so on are Google properties that are not part of Android, and are licensed separately. Contact [android-partnerships@google.com](mailto:android-partnerships@google.com) for inquiries related to those apps.

### **Is compatibility mandatory?**

No. The Android Compatibility Program is optional. Since the Android source code is open, anyone can use it to build any kind of device. However, if a manufacturer wishes to use the Android name with their product, or wants access to Android Market, they must first demonstrate that the device is compatible.

### **How much does compatibility certification cost?**

There is no cost to obtain Android compatibility for a device. The Compatibility Test Suite is open-source and available to anyone to use to test a device.

### **How long does compatibility take?**

The process is automated. The Compatibility Test Suite generates a report that can be provided to Google to verify compatibility. Eventually we intend to provide self-service tools to upload these reports to a public database.

### **Who determines what will be part of the compatibility definition?**

Since Google is responsible for the overall direction of Android as a platform and product, Google maintains the Compatibility Definition Document for each release. We draft the CDD for a new Android version in consultation with a number of OEMs, who provide input on its contents.

### **How long will each Android version be supported for new devices?**

Since Android's code is open-source, we can't prevent someone from using an old version to launch a device. Instead, Google chooses not to license the Android Market client software for use on versions that are considered obsolete. This allows anyone to continue to ship old versions of Android, but those devices won't use the Android name and will exist outside the Android apps ecosystem, just as if they were non-compatible.

### **Can a device have a different user interface and still be compatible?**

The Android Compatibility Program focuses on whether a device can run third-party applications. The user interface components shipped with a device (such as home screen, dialer, color scheme, and so on) does not generally have much effect on third-party apps. As such, device builders are free to customize the user interface as much as they like. The Compatibility Definition Document does restrict the degree to which OEMs may alter the system user interface for areas that do impact third-party apps.

## **When are compatibility definitions released for new Android versions?**

Our goal is to release new versions of Android Compatibility Definition Documents (CDDs) once the corresponding Android platform version has converged enough to permit it. While we can't release a final draft of a CDD for an Android software version before the first flagship device ships with that software, final CDDs will always be released after the first device. However, wherever practical we will make draft versions of CDDs available.

## **How are device manufacturers' compatibility claims validated?**

There is no validation process for Android device compatibility. However, if the device is to include Android Market, Google will typically validate the device for compatibility before agreeing to license the Market client software.

## **What happens if a device that claims compatibility is later found to have compatibility problems?**

Typically, Google's relationships with Android Market licensees allow us to ask them to release updated system images that fix the problems.

## **Compatibility Test Suite**

### **What is the purpose of the CTS?**

The Compatibility Test Suite is a tool used by device manufacturers to help ensure their devices are compatible, and to report test results for validations. The CTS is intended to be run frequently by OEMs throughout the engineering process to catch compatibility issues early.

### **What kinds of things does the CTS test?**

The CTS currently tests that all of the supported Android strong-typed APIs are present and behave correctly. It also tests other non-API system behaviors such as application lifecycle and performance. We plan to add support in future CTS versions to test "soft" APIs such as Intents as well.

### **Will the CTS reports be made public?**

Yes. While not currently implemented, Google intends to provide web-based self-service tools for OEMs to publish CTS reports so that they can be viewed by anyone. CTS reports can be shared as widely as manufacturers prefer.

### **How is the CTS licensed?**

The CTS is licensed under the same Apache Software License 2.0 that the bulk of Android uses.

### **Does the CTS accept contributions?**

Yes please! The Android Open-Source Project accepts contributions to improve the CTS in the same way as for any other component. In fact, improving the coverage and quality of the CTS test cases is one of the best ways to help out Android.

### **Can anyone use the CTS on existing devices?**

The Compatibility Definition Document requires that compatible devices implement the 'adb' debugging utility. This means that any compatible device -- including ones available at retail -- must be able to run the CTS tests.