

Launch Checklist

Before you publish your app on Google Play and distribute it to users, you need to get the app ready, test it, and prepare your promotional materials.

This document helps you understand the publishing process and get ready for a successful product launch on Google Play. It summarizes some of the tasks you'll need to complete before publishing your app on Google Play, such as creating a signed, release-ready APK, understanding the requirements of the app, and creating the product page and graphic assets for your app.

The preparation and publishing tasks are numbered to give you a rough idea of sequence. However, you can handle the tasks in any sequence that works for you or you can skip steps as appropriate.

As you move toward publishing, a variety of support resources are available to you. Relevant links are provided in each step.

1. Understand the publishing process

Before you begin the steps in this checklist, you should take a moment to read and understand the overall publishing workflow and become familiar with how the process works. In particular, you or your development team will need to prepare your app for release using a process common to all Android apps. The [Publishing Workflow \(/tools/publishing/publishing_overview.html\)](/tools/publishing/publishing_overview.html) documents provide the details on how publishing works and how to get an APK ready for release.

Once you are familiar with publishing in general, read this document to understand the issues that you should consider when publishing an app on Google Play.

Related resources:

- [General Publishing Overview](#) — Start here for an overview of publishing options for Android apps.
- [Preparing for Release](#) — Developer documentation on how to build the signed, release-ready APK. This process is the same for all Android apps.

2. Understand Google Play policies and agreements

Make sure that you understand and follow the Google Play program policies that you accepted when registering. Google Play actively enforces the policies and any violations can lead to suspension of your app or, for repeated violations, termination of your developer account.

Related resources:

- [Policy and Best Practices](#) — Help Center document describing various content policies and processes.

3. Test for Core App Quality

Before you publish an app on Google Play, it's important to make sure that it meets the basic quality expectations for all Android apps, on all of the devices that you are targeting. You can check your app's quality by setting up a test environment and testing the app against a short set of **core app quality criteria**. For complete information, see the [Core App Quality Guidelines \(/distribute/googleplay/quality/core.html\)](/distribute/googleplay/quality/core.html).

If your app is targeting tablet devices, make sure that it delivers a rich, compelling experience to your tablet customers. See the [Tablet App Quality Checklist \(/distribute/googleplay/quality/tablet.html\)](/distribute/googleplay/quality/tablet.html) for recommendations on ways to optimize your app for tablets.

Related resources:

- [Core App Quality Guidelines](#) — A set of core quality criteria that all Android apps should meet on all targeted devices.

CHECKLIST

1. [Understand the publishing process](#)
2. [Understand Google Play policies](#)
3. [Test for core app quality](#)
4. [Determine your content rating](#)
5. [Determine country distribution](#)
6. [Confirm the app's overall size](#)
7. [Confirm app compatibility ranges](#)
8. [Decide on free or priced](#)
9. [Consider In-app Billing](#)
10. [Set prices for your apps](#)
11. [Start localization early](#)
12. [Prepare promotional graphics](#)
13. [Build the release-ready APK](#)
14. [Plan a beta release](#)
15. [Complete the product details](#)
16. [Use Google Play badges](#)
17. [Final checks and publishing](#)
18. [Support users after launch](#)

- [Tablet App Quality Checklist](#) — A set recommendations for delivering the best possible experience to tablet users.

4. Determine your app's content rating

Google Play requires you to set a content rating for your app, which informs Google Play users of its maturity level. Before you publish, you should confirm what rating level you want to use. The available content rating levels are:

- Everyone
- Low maturity
- Medium maturity
- High maturity

On their Android devices, Android users can set the desired maturity level for browsing. Google Play then filters apps based on the setting, so the content rating you select can affect the app's distribution to users. You can assign (or change) the content rating for your app in the Developer Console, so no changes are required in your app binary.

Related resources:

- [Rating your application content for Google Play](#) — Help Center document describing content ratings levels and how to choose the appropriate one for your app.

5. Determine country distribution

Google Play lets you control what countries and territories your app is distributed to. For widest reach and the largest potential customer base, you would normally want to distribute to all available countries and territories. However, because of business needs, app requirements, or launch dependencies, you might want to exclude one or more countries from your distribution.

It's important to determine the exact country distribution early, because it can affect:

- The need for localized resources in the app
- The need for a localized app description in the Developer Console
- Legal requirements for the app that may be specific to certain countries
- Time zone support, local pricing, and so on.

With your country targeting in mind, you should assess what your localization needs are, both in your app and in its Google Play listing details, and start the work of localization well in advance of your launch target date.

See [Localization Checklist \(/distribute/googleplay/publish/localizing.html\)](#) for key steps and considerations in the localizing process.

Related resources:

- [Localization Checklist](#) — Overview of key steps and considerations for localizing your Android app.
- [Supported locations for distributing applications](#) on Google Play. .

6. Confirm the app's overall size

The overall size of your app can affect its design and how you publish it on Google Play. Currently, the maximum size for an APK published on Google Play is 50 MB. If your app exceeds that size, or if you want to offer a secondary download, you can use [APK Expansion Files \(/google/play/expansion-files.html\)](#), which Google Play will host for free on its server infrastructure and automatically handle the download to devices.

- The maximum size for an APK published on Google Play is 50 MB.
- You can use up to two (2) APK Expansion Files, each up to 2 GB in size, for each APK.

Using APK Expansion files is a convenient, cost-effective method of distributing large apps. However, the use of APK Expansion Files requires some changes in your app binary, so you will need to make those changes before creating your release-ready APK.

To minimize the size of your app binary, make sure that you run the [Proguard \(/tools/help/proguard.html\)](#) tool on your code when building your release-ready APK.

Related resources:

- [APK Expansion Files](#) — Developer documentation describing APK Expansion Files and how to support them in your app.
- [ProGuard](#) — Developer documentation describing how to use ProGuard to shrink, optimize, and obfuscate your code prior to release.

7. Confirm the app's platform and screen compatibility ranges

Before publishing, it's important to make sure that your app is designed to run properly on the Android platform versions and device screen sizes that you want to target.

From an app-compatibility perspective, Android platform versions are defined by [API level](#) ([/guide/topics/manifest/uses-sdk-element.html#ApiLevels](#)). You should confirm the minimum version that your app is compatible with (`<minSdkVersion>` ([/guide/topics/manifest/uses-sdk-element.html](#))), as that will affect its distribution to Android devices once it is published.

For screen sizes, you should confirm that the app runs properly and looks good on the range of screen sizes and densities that you want to support. You should confirm the minimum screen-size and density support that your app declares (`<supports-screens>` ([/guide/topics/manifest/supports-screens-element.html](#))), since that can affect its distribution to Android devices once it is published.

To get a better understanding of the current device penetration of Android platform versions and screen sizes across all Android devices, see the [Device Dashboard](#) ([/about/dashboards/index.html](#)) charts.

Related resources:

- [Device Dashboard](#) — A chart showing global percentages of devices by Android version, screen size, and level of OpenGL ES support.
- [Android API Levels](#) — A definition of API Levels and a list of which Android platform versions they are associated with.

8. Decide whether your app will be free or priced

On Google Play, you can publish apps as free to download or priced. Free apps can be downloaded by any Android user in Google Play. Paid apps can be downloaded only by users who have registered a form of payment in Google Play, such as a credit card or Direct Carrier Billing.

Deciding whether your app will be free or paid is important because, on Google Play, **free apps must remain free**.

- Once you publish your app as a free app, you cannot ever change it to being a priced app. However, you can still sell in-app products and subscriptions through Google Play's In-app Billing service.
- If you publish your app as a priced app, you *can* change it at any time to being a free app (but cannot then change it back to priced). You can also sell in-app products and subscriptions.

If your app is be priced, or if you'll be selling in-app products, you need set up a Google Wallet merchant account before you can publish.

Related resources:

- [In-app Billing](#) — Developer introduction to Google Play In-app Billing.

9. Consider using In-app Billing

Google Play [In-app Billing](#) ([/google/play/billing/index.html](#)) lets you sell digital content in your applications. You can use the service to sell a wide range of content, including downloadable content such as media files or photos, and virtual content such as game levels or potions. In-app Billing service lets you sell one-time purchases and subscriptions from inside your app. This can help you to monetize the app over its installed lifetime.

If your are looking for more ways to monetize your app and build engagement, you should consider In-app Billing. The service has become very popular with both users and developers. To use In-app Billing, you need to make changes to your app binary, so you will need to complete and test your implementation before creating your release-ready APK.

Related resources:

- [In-app Billing](#) — Developer documentation describing In-app Billing and how to support it in your app.

10. Set prices for your products

If your app is priced or you will sell in-app products, Google Play lets you set prices for your products in a variety of currencies, for users in markets around the world. You can set prices individually in different currencies, so you have the flexibility to adjust your price according to market conditions and exchange rates.

Before you publish, consider how you will price your products and what your prices will be in various currencies. Later, you can set prices in all available currencies through the Developer Console.

Related resources:

- [Selling Apps in Multiple Currencies](#) — Help Center document describing how pricing works in Google Play.
- [Prices and supported currencies](#) — Help Center document listing supported currencies for pricing your apps.
- [Transaction Fees](#) — Help Center document describing transaction fees for priced apps and in-app products.
- [Specifying tax rates](#) — Help Center document describing how to set tax rates for different countries.

11. Start localization

With your country targeting in mind, it's a good idea to assess your localization needs and start the work of localizing well in advance of your target launch date.

There are at least three aspects of localization to consider:

- Localizing the strings, images, and other resources in your app
- Localizing your app's store listing details on Google Play
- Localizing the app's graphic assets, screenshots, and videos that accompany your store listing.

See [Localization Checklist \(/distribute/googleplay/publish/localizing.html\)](#) for key steps and considerations in the localizing process.

To localize your store listing, first create and finalize your app title, description, and promotional text. Collect and send all of these for localization. You can optionally translate the "Recent Changes" text for app updates as well. Later you can add your localized listing details in the Developer Console, or you can choose to let Google Play auto-translate your listing details into the languages you support.

A key part of making your app listing attractive to a global customer base is creating localized versions of your promotional graphics, screenshots and videos. For example, your app's feature graphic might include text that should be translated, for maximum effectiveness. You can create different versions of your promotional graphics for each language and upload them to the Developer Console. If you offer a promotional video, you can create localized versions of it and then add a link to the correct localized video for each language you support.

When your translations are complete, move them into your app resources as needed and test that they are loaded properly. Save your app's translated listing details for later, when you upload assets and configure your product details.

Related resources:

- [Localization Checklist](#) — Overview of key steps and considerations for localizing your Android app.
- [Localizing with Resources](#) — Developer guide to localizing resources in your app.

12. Prepare promotional graphics, screenshots, and videos

When you publish on Google Play, you can supply a variety of high-quality graphic assets to showcase your app or brand. After you publish, these appear on your product details page, in store listings and search results, and elsewhere. These graphic assets are key parts of a successful product details page that attracts and engages users, so you should consider having a professional produce them for you. Screen shots and videos are also very important, because they show what your app looks like, how it's used or played, and what makes it different.

All of your graphic assets should be designed so that they are easy to see and highlight your app or brand in a colorful, interesting way. The assets should reference the same logo and icon as users will actually find in the All Apps launcher once they have downloaded the app. Your graphic assets should also fit in well with the graphic assets of other apps published by you, which will be also be displayed to users on your product details page.

To help you market your app more effectively to a global audience, Google Play lets you create localized versions of your promotional graphics, screenshots, and videos and upload them to the Developer Console. When a user visits your app's store listing, Google Play displays the promotional graphic, screenshots and video that you've provided for the user's

language.

To localize your promotional graphics, you can translate any embedded text, use different imagery or presentation, or change your marketing approach to best address the needs of users in specific languages. For example, if your feature or promotional graphic includes an embedded product name or tag line, you can translate that text and add it to a localized version of the promotional graphic.

Because your localized graphic assets and videos are so important, you should get started on creating them and localizing them well in advance of your target publishing date.

Note: Localized promotional graphics and videos are supported only in the new Developer Console design.

Related resources:

- [Graphic Assets for your Application](#) — Details about the graphic assets you need to upload before publishing.
- [Google Play Featured Image Guidelines](#) — Blog post that highlights key design considerations for your app's featured image.

13. Build and upload the release-ready APK

When you are satisfied that your app meets your UI, compatibility, and quality requirements, you can build the release-ready version of the app. The release-ready APK is what you will upload to the Developer Console and distribute to users.

The process for preparing a release-ready APK is the same for all apps, regardless of how they are distributed. Generally the process includes basic code cleanup and optimization, building and signing with your release key, and final testing. When you are finished preparing your application for release, you'll have a signed APK file that you can upload to the Developer Console for distribution to users.

For complete details on how to create a release-ready version of your app, read [Preparing for Release](#) ([/tools/publishing/preparing.html](https://tools/publishing/preparing.html)).

Once you have the release-ready APK in hand, you can upload it to the Developer Console. If necessary, you can replace the APK with a more recent version before publishing.

Related resources:

- [Preparing for Release](#) — Essential information for preparing and packaging your app properly for distribution.

14. Plan a beta release

Before launching your app, it's always valuable to get real-world feedback from users — even more so when you are launching a new app. It's highly recommended that you distribute a pre-release version of your app to users across your key markets and provide an easy means for them to provide feedback and report bugs.

Google Play can help you set up a beta program for your app. After you sign in to the Developer Console and upload your APK, you can set up groups of users for alpha testing and beta testing the app. You can start with a small group of alpha testers, then move to a larger group of beta testers. Once users are added, they access your app's store listing and install the app. User feedback from alpha and beta testers goes directly to you and is not posted as public reviews.

The feedback you receive will help you adjust your UI, translations, and store listing to ensure a great experience for users.

Easy beta testing

Google Play now lets you set up groups of alpha and beta testers, anywhere around the world. Check out this powerful feature next time you sign in to the Developer Console.

15. Complete the app's product details

On Google Play, your app's product information is shown to users on its product details page, the page that users visit to learn more about your app and the page from which they will decide to purchase or download your app, on their Android devices or on the web.

Google Play gives you a variety of ways to promote your app and engage with users on your product details page, from colorful graphics, screenshots, and videos to localized descriptions, release details, and links to your other apps. As you

prepare to publish your app, make sure that you take advantage of all that your product details page can offer, making your app as compelling as possible to users.

You should begin planning your product page in advance of your target launch date, arranging for localized description, high-quality graphic assets, screenshots and video, and so on.

As you get near your target publishing date, you should become familiar with all the fields, options, and assets associated with the product details configuration page in the Developer Console. As you collect the information and assets for the page, make sure that you can enter or upload it to the Developer Console, until the page is complete and ready for publishing.

After you've set your app's geographic targeting in the Developer Console, remember to add your localized product details, promotional graphics, and so on, for all of the languages that you support.

If your app is targeting tablet devices, make sure to include at least one screen shot of the app running on a tablet, and highlight your app's support for tablets in the app description, release notes, promotional campaigns, and elsewhere.

Related resources:

- [Category types](#) — Help Center document listing available categories for apps.
- [Graphic Assets for your Application](#) — Help Center document describing the various graphics you can add to your product listing.

16. Use Google Play badges and links in your promotional campaigns

Google Play badges give you an officially branded way of promoting your app to Android users. Use the [Google Play Badge generator \(/distribute/googleplay/promote/badges.html\)](#) to quickly create badges to link users to your products from web pages, ads, reviews, and more. You can also use special [link formats \(/distribute/googleplay/promote/linking.html\)](#) to link directly to your product details page, to a list of your products, or to search results.

To help your app get traction after launch, it's strongly recommended that you support launch with a promotional campaign that announces your product through many channels as possible, in as many countries as possible. For example, you can promote the launch using ad placements, social network or blog posts, video and other media, interviews and reviews, or any other channel available.

Related resources:

- [Google Play Badges](#) — Generate a badge to bring users to your app in Google Play.
- [Linking to Your Products](#) — Link formats that you can use to bring users to your app in Google Play.

17. Final checks and publishing

When you think you are ready to publish, sign in to the Developer Console and take a few moments for a few final checks.

Make sure that:

- Your developer profile has the correct information and is linked to the proper Google Wallet merchant account (if you are selling products).
- You have the right version of the app uploaded.
- All parts of your Product Details are ready, including all graphic assets, screenshots, video, localized descriptions, and so on.
- You have set your app's pricing to free or priced.
- You have set country (and carrier) targeting and priced your products (if appropriate) in buyer currencies
- "Compatible devices" shows that your app is actually reaching the devices that you are targeting. If not, you should check with your development team on the apps requirements and filtering rules.
- You have provided the correct link to your web site and the correct support email address.
- Your app does not violate content policy guidelines.
- You have acknowledged that your app meets the guidelines for Android content on Google Play and also US export laws.

Your app is now ready to publish!

If you are releasing an update, make sure to read the [requirements for publishing updates](#) (<http://support.google.com/googleplay/android-developer/bin/answer.py?hl=en&answer=113476&topic=2365760&ctx=topic>).

When you are ready, click the **Publish** button in the Developer Console. Within a few hours, your app will become available to users and your product page will appear in Google Play for browsing, searching, or linking from your promotional

campaigns.

Related resources:

- [Google Play Developer Program Policies](#) — Guidelines for what is acceptable content in Google Play. Please read and understand the policies before publishing.
- [Updates](#) — Requirements for app updates in Google Play.
- [Developer Support](#) — Support resources that you can use to find answers and report issues.

18. Support users after launch

After you publish an app or an app update, it's crucial for you to support your customers. Prompt and courteous support can provide a better experience for users that results in better ratings and more positive reviews for your products. Users are likely to be more engaged with your app and recommend it if you are responsive to their needs and feedback. This is especially true after publishing if you are using a coordinated promotional campaign.

There are a number of ways that you can keep in touch with users and offer them support. The most fundamental is to provide your *support email address* on your product details page. Beyond that, you can provide support in any way you choose, such as a forum, mailing list or a Google+ page. The Google Play team does provide user support for downloading, installing and payments issues, but issues that fall outside of these topics will fall under your domain. Examples of issues you can support include: feature requests, questions about using the app and questions about compatibility settings.

After publishing, plan to:

- Check your ratings and reviews frequently on your app's product details page. Watch for recurring issues that could signal bugs or other issues.
- Be mindful of new Android platform version launches, as compatibility settings for your apps might need to be updated.
- Put a link to your support resources on your web site and set up any other support such as forums.
- Provide an appropriate support email address on your product details page and respond to users when they take the time to email you.
- Beyond the automatic refund window offered by Google Play, be generous with your own refund policy, as satisfied users will be more likely to purchase in the future.
- Acknowledge and fix issues in your app. It helps to be transparent and list known issues on your product details page proactively.
- Publish updates as frequently as you are able, without sacrificing quality or annoying users with too-frequent updates.
- With each update, make sure to provide a summary of what's changed. You can enter this information in the Developer Console. Users will read it and appreciate that you are serious about improving the quality of your app.

Related resources:

- [Supporting your users](#) — Help Center document describing options for supporting users.
- [In-app Billing](#) — Help Center document describing how to correctly set up In-app Billing.
- [Issuing Refunds](#) — -- Help Center document describing how to issue refunds.