

RB Consulting, Inc.
Google Play Publishing Overview
Mobile Applications

Prepared by:
Robert Broadhead

March 27, 2015



RB Consulting Confidential

Revised: Rob Broadhead

No portion of this document may be reproduced, stored in a retrieval system or transmitted in any form by any means without the prior written approval of RB Consulting Inc. Any such requests should be sent to:

RB Consulting, Inc.
309 Inwood Way
Franklin, TN 37064

Contact Name: Rob Broadhead

In no event shall RB Consulting be liable to anyone for special, incidental, collateral, or consequential damages arising out of the use of this information.

Revision: 01 RB Consulting Inc. 2015 All rights reserved.

This document contains RB Consulting sensitive material. Posting or sharing this material outside of RB Consulting should be done only at management discretion.

Printed in the United States

Google Play Overview

The Google store for publishing/purchasing applications is called Google Play. There are other app stores that allow a developer to publish Android applications, but this document will focus only on Google Play. It is important to note that once an application is published it may be run from anywhere in the world so scalability and security are critical for the reputation of the publisher as well as potential reasons for being rejected from the Google Play store.

Development Requirements



The Android developer site has a wealth of documentation around development and should be reviewed by the developers and development management team in order to ensure a smooth and secure publishing process. The steps for publishing an application are listed in the image shown to the left. This is a full SDLC process and should be followed completely to help ensure high quality applications are being produced. Skipping steps may cause serious application issues during the publishing process or beyond.

Google suggests specific development tools and environments, but its open nature has allowed for a large number of tools and utilities to be created that are open source and/or commercial applications. These can be used to increase team productivity. Tool discussion is out of scope in this document, but the latest tools should be searched for and reviewed as time permits. Google suggests Android Studio to ensure the quickest to market and most reliable development process. The

Android environment is constantly evolving and regular reviews of the latest releases and tools will help reduce compatibility issues when the final product is published.

Publishing Steps

There are a number of steps required to publish an application. Like all things Google, a developer account must be linked to a gmail address. Once a gmail address has been setup as a developer account the other steps can be addressed and completed. Details are provided later in this document and on the Android development site, but this is provided as a quick checklist.

- Determine content rating and country (or countries) of distribution
- Verify final app size is below 50MB limit
- Verify platform and screen compatibility
- Price model (amount, in app billing/subscriptions)
- Application Development
- Application Testing
- Promotional graphics, screen shots, and content
- Prepare for Release
- Build and Deploy (decide on whether to use a beta program)
- Complete App store listing
- Final checks and publishing

Publishing Details

The publishing process requires a number of questions to be answered about the application to be published. These answers are critical to how the application will look in Google Play. The content can be edited even after an application has been published, but it is best to get the content correct from the start rather than have to correct typos and other errors. First impressions are important when trying to get users to try an application. There is a concept of Alpha and Beta testing in Google Play. There is also a draft mode available. It is recommended to take advantage of these multiple ways to preview and test out an application before it is released to the general public. An application can also be unpublished, if desired, to remove it from availability.

Once an application is published Google provides an option to stage rollouts. This allows an application to be made available to a percentage of users at a time and ensure users are able to use it properly. Rollbacks are not allowed, but a new version can be released if a release is "broken."

Once an application is built and ready to be published there is a series of groups of data that will need to be entered. This data includes: content/descriptions, images to use as icons while searching for the app in the store, and screen shots.

Preparing for Release

An application development effort is going to be somewhat similar for mobile apps to any other development effort. This means that the general development effort for mobile will follow the development team's standard processes. Once the application is developed and functionally tested the publishing of an Android application falls into the Android processes. There are five main tasks involved in publishing :

1. Gather materials for release
2. Configure application for release
3. Build application for release
4. Prepare remote servers
5. Test application for release

[Gathering materials for release](#) is as much a marketing/sales effort as it is development. This step requires the EULA to be created, cryptographic keys for signing the application (validity must end after 10-22-2033), create an application icon, third party tool license keys, screen shots, and promotional text for the store.

Configuring an application for release is largely a clean-up effort from testing. Make sure all logging and debugging code is turned off and that any related files (log files, trace files, etc) have been removed. Clean up the project directories to remove orphan code/files. Source files for resources and other items should also be removed. This step includes verifying configuration settings as well to make sure the build files are all properly configured for production and final values for things like version numbers and application name are set. Included in this step is a verification that the application is configured to properly support the target devices. Ensure that any URLs are targeted to the correct servers and not development or test servers. Some of the device support is addressed during development, but configuration settings need to match the final target devices and screen sizes as well. Finally, add support for Google Play licensing if that is desired for the application (<http://developer.android.com/google/play/licensing/index.html>).

Building an application for release is just a formality at this point if the other steps have been followed. The build will need to produce a signed and optimized package (.apk file). Once all the proper keys have been acquired it is a matter of incorporating those in the build and then build for production rather than debug (<http://developer.android.com/tools/building/building-studio.html>).

Preparing remote servers may already be done outside of this process. The preparation includes securing the server(s) and ensuring that the server(s) have access to production data. Whether the servers are pre-existing or not testing will need to be done to ensure the application has the access it needs.

Testing for release is covered in more detailed later in this document. The key take away for this step is to mimic realistic conditions including full deployment to actual (not virtual) devices to do the final testing. This will ensure there are no issues that arise with a real device that were not seen in emulators. The emulators are great for initial testing but speed and memory issues often will not appear until testing is done on a physical device.

Store Listing

The store listing has a few tiers of content to provide about the application. This product listing text can be provided in multiple translations for each of the properties. The properties include:

1. Title (30 character limit)
2. Description (4000 character limit, limited html tag support)
3. Promo text (80 character limit)
4. Recent changes/Whats New (500 character limit)

There are also a series of graphic assets that are part of the product listing:

1. Screen shots
2. High-res icon
3. Feature Graphic
4. Banner Asset (Android TV)
5. Promotional Video

[Screen shots](#) can be provided for each of the display types (Phone, 7-inch tablet, 10-inch tablet, and Android TV). A minimum of 2 screen shots are required and the minimum dimension is 320px with a max of 3840px or twice the minimum. Tablet support requires tablet screen shots. Android TV has some additional requirements, but that is outside the scope of this document. It is worth noting that video displays first in the store, then tablet screen shots, and then phone screen shots.

A [high-res icon](#) is required for any application type and is usually a higher res version of the application launcher icon. The icon is required to be a 32-bit PNG with alpha that is 512px by 512px and with a maximum file size of 1024KB. The guidelines suggest a distinct silhouette, 3-D front view with a slight perspective as if viewed from above. It should also have a background that can be used on any background color.

The [feature graphic](#) is required in order for the application to be listed as a featured application on Google Play. This can be optional, but it is highly suggested to not consider it as such. Featured applications are going to be much more likely to be tried out by users. The feature graphic should be a way to draw users to the application and thus is like an ad. You do not have to have it, but lack of one will likely reduce adoption of the application. In order to be considered for featured placement the image must be 1024px by 500px in dimension. The graphic can be JPEG or 24-bit PNG (no alpha). If a promotional video has been provided then a play button will appear on this image to play the video.

A [Banner asset](#) can be provided to allow an application to be listed on Android TV. This must be 320px by 180px and can be a JPEG or 24-bit PNG (no alpha). This is required for an Android TV application.

A [promotional video](#) can be provided via YouTube. This is an optional feature, but provides a great way to advertise and promote an application. This is a video URL in YouTube and not a playlist or channel.

Coding Standards

Publishing an Android application on the store requires a certain level of quality in the code, architecture, stability, and packaging. These requirements are presented on the Android developer site at: <http://developer.android.com/distribute/essentials/quality/core.html>. The highlights are brought out here to provide a quick checklist for easing the process of publishing an application and reducing rejection from the application evaluators.

In Android development one of the key decisions to be made is what versions and devices will be supported. Once a decision has been made then the documentation needs to be reviewed for any older devices or OS versions. Standards have evolved over time and the requirements have as well. It is generally safe to assume “backward compatibility” of requirements so publishing for the newest devices and OS versions will cover older ones, but reviewing those older requirements will help avoid errors that only users of those older versions will see. The notes in this document are based on Android OS version 5.01 (21) and older (or newer) versions might have slightly different features or standards to match.

User Interface

The UI design standards are easy to meet if development is done using standard controls and look-and-feel. Problems may arise if the application tries to do things atypical to the Android application experience. The rule of thumb is to not do things that break the default way applications work in Android. This can be broken down to these specific rules :

- Do not change how the system icons (e.g. back button) work
- Do not add new ways to access the system icons
- Always allow for usage of the back and home buttons from any screen
- Notifications should not contain extra information (ads or unrelated content)
- Stack notifications into a single object where possible
- Use notifications only for a user context change or to inform about an on going event
- When a user presses the back button and will lose data they should be notified and allowed to save their data before proceeding

Behavior and Permissions

Applications can make use of core hardware features like video, audio, SD cards, etc. There are some general rules to follow with regard to these features that boil down to “take only what you need as long as you need it”. Here are some checklist items to look for in evaluating compliance with the requirements :

- App requests only absolute minimum permissions it needs
- App does not request access to sensitive data or pay services (dialer or SMS) unless they are required for a core capability of the app.
- App functions normally on an SD card (unless SD is not supported)
- Audio does not play when is screen is off, over another app, or behind lock screen unless that is a core feature (e.g. a music player)
- Audio either resumes when an application is moved to foreground/active state or the user is shown the audio is in a paused state
- Support both orientations where possible. Support same functionality in all orientations with minor changes acceptable (content or display changes due to orientation).
- Orientations should use screen real estate and not go to letterbox
- App does not leave services running while in background, unless related to core feature of the app.
- App preserves and restores user state when switching from active to inactive app status
- Resources are properly released as soon as the need for them has ended

Performance and Stability

The standards for performance and stability are what a user would expect of any application. In general:

- The display should look correct (no cut-off images or text)
- Audio should work (no crackle, stuttering, etc)
- Exceptions should be handled gracefully
- There should not be any errors that cause an application crash or freeze.

The only key metric that is provided as a hard number is that progress indication should be provided if it takes more than 2 seconds for an app or screen to load.

Testing Notes

A generic test script is beyond the scope of this document, but there are some key points that should be hit during testing an Android application. These steps include:

1. Navigate to all parts/screens of the application
2. Verify any available ports can be in use without breaking the app (keyboard, usb, ssd etc)
3. Verify rotation to and from landscape/portrait works without impacting app.
4. Verify all back buttons
5. Verify pause/sleep and restore functionality works for apps (background app and power off/on)
6. Verify all resources (images, etc) work properly on publish