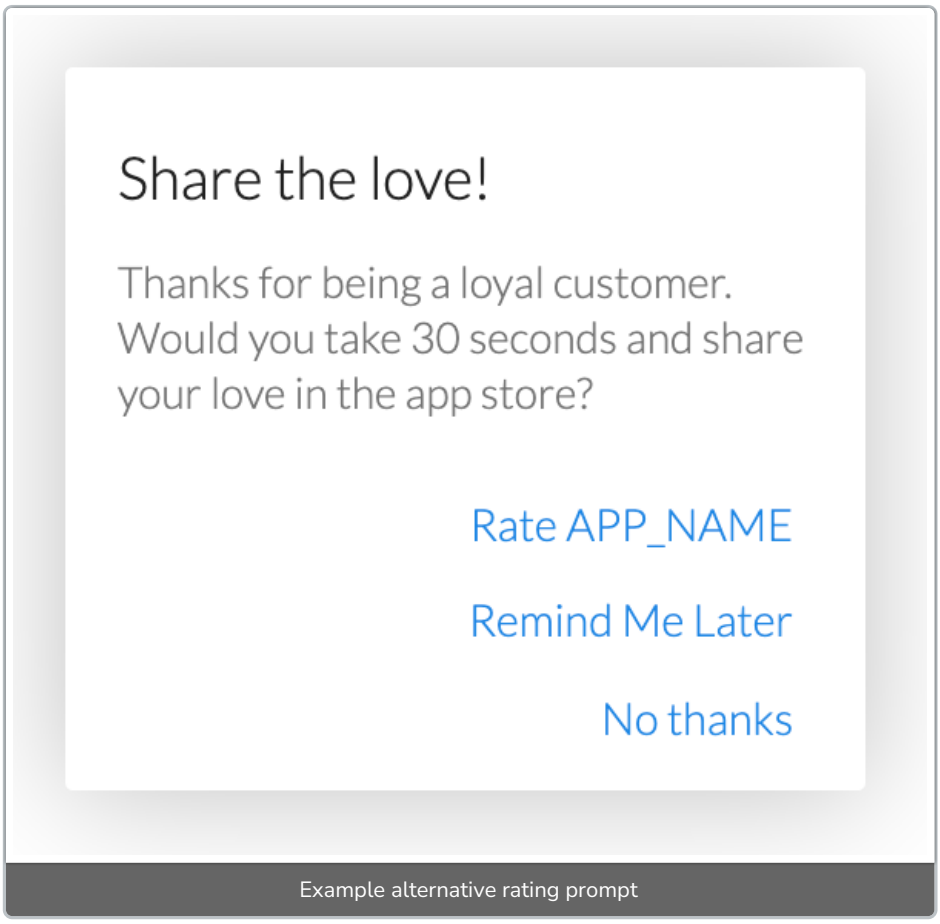
Developer Guide for Alternative Rating Prompt

This document provides an overview of how to use an alternative rating prompt, instead of the Google In-App Review.

> This is for version **6.0.0+** (includes betas) of the **Android** Alchemer Mobile SDK.
>
> The Google In-App Review is the preferred way to ask for reviews for apps in the Play Store and is enabled by default.

What is this rating prompt?

This is an interaction by Alchemer that is primarily used as a solution for alternate app stores but also can serve as an optional alternative to the Google In-App Review prompt.



Example alternative rating prompt

Who is this for?

This rating prompt will mainly be used for these three use cases:

A. Customers who have their app on an app store that is **not** the Play Store

B. Customers who have an app on the Play Store **and** other app stores

C. Customers who don't want to use Google In-App Review

Using this prompt

For a separate app not in the Google Play Store

If you have separate apps for each app store you can set `customAppStoreURL` in each app within your `ApptentiveConfiguration` to utilize the alternative rating prompt.

Tutorial

> Some apps will intercept specific URLs for deep linking. Look for guidance from the app store support on what URL you should send your users to rate.
>
> Make sure you test your URLs in-app with the alternate app store's app installed.

1. Get your app's store URL
2. Create your `ApptentiveConfiguration` within your `Application` class
3. Set your app store URL as a String to `ApptentiveConfiguration`'s `customAppStoreURL`
4. Register Alchemer Mobile `Apptentive.register(this, configuration)`

```
class MyApplication : Application() {
  override fun onCreate() {
    super.onCreate()
    val configuration = ApptentiveConfiguration(
      apptentiveKey = "YOUR_APPTENTIVE_KEY",
      apptentiveSignature = "YOUR_APPTENTIVE_SIGNATURE"
    )
    configuration.customAppStoreURL = "YOUR_APP_STORE_URL"
    Apptentive.register(this, configuration)
  }
}
```

With one app in multiple stores using build variants

If you have one app and deploy it to multiple app stores, you can use build variants to build up different AABs or APKs for different stores.

> When `customAppStoreURL` is set to `null` or just not set, the SDK will default to using Google

In-App Review for the Rating Dialog interaction

Tutorial

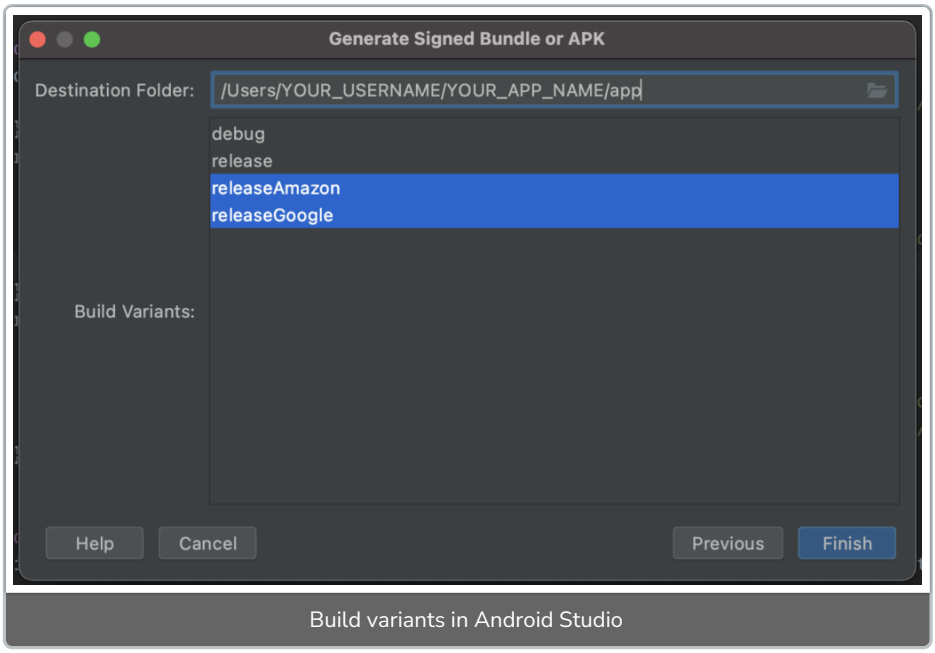
The example code below will show you how to set up release builds using Google In-App Review for the Google release build and the alternative rating prompt for the Amazon release build.

1. Go to your app's `build.gradle` file
2. Create different `buildTypes` for each app store you'll release to
3. Add `buildConfigField` s to each of them
   a. These will allow these variables to be accessed through the generated `BuildConfig` file when the app is built

```
android {
  ...

  buildTypes {
    debug {
      ...
    }
    releaseGoogle {
      buildConfigField "String", "CUSTOM_STORE_URL", "null"
    }
    releaseAmazon {
      buildConfigField "String", "CUSTOM_STORE_URL", "\"YOUR_APP_STORE_URL\""
    }
  }
}
```

4. **Sync** your project with the updated Gradle file
   a. This will create build variants that you can view within the Build Variants tab in Android Studio



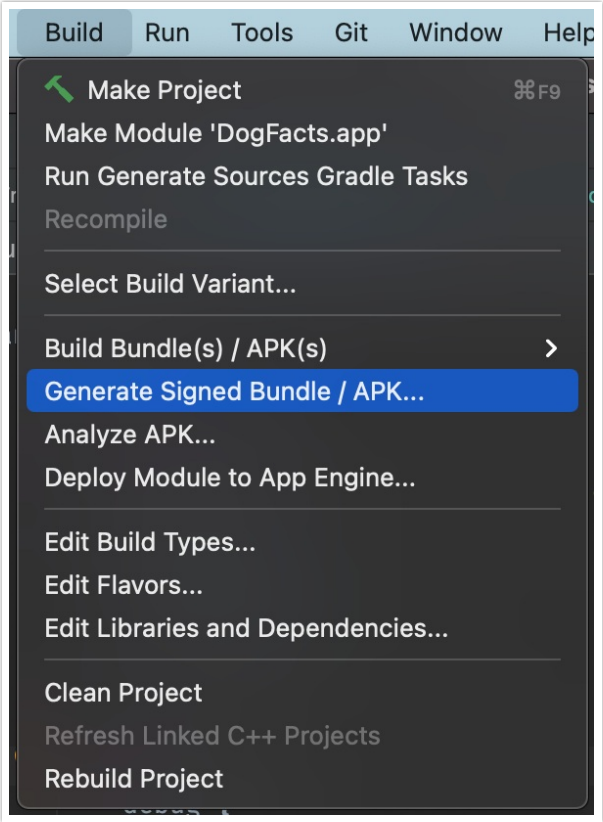
Build variants in Android Studio

5. **Build** your project
   a. This will create the `BuildConfig` file with the `CUSTOM_STORE_URL` variable

6. Set `BuildConfig.CUSTOM_STORE_URL` to `customAppStoreURL` for `ApptentiveConfiguration` within your `Application` file

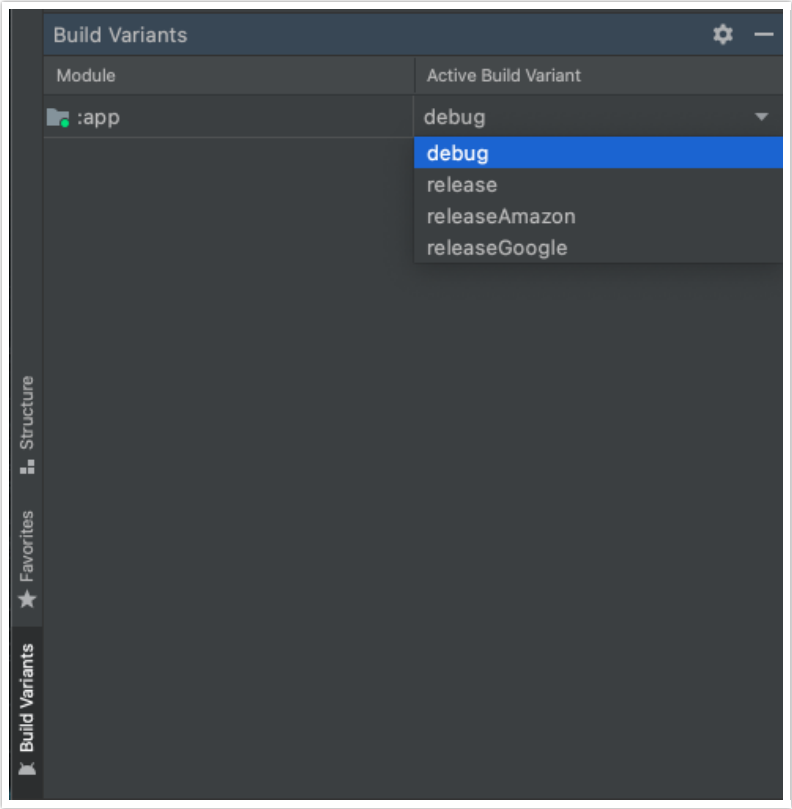
```
class MyApplication : Application() {
    override fun onCreate() {
        super.onCreate()
        val configuration = ApptentiveConfiguration(
            apptentiveKey = "YOUR_APPTENTIVE_KEY",
            apptentiveSignature = "YOUR_APPTENTIVE_SIGNATURE"
        )
        configuration.customAppStoreURL = BuildConfig.CUSTOM_STORE_URL
        Apptentive.register(this, configuration)
    }
}
```

7. Click **Generate Signed Bundle / APK...** in the **Build** dropdown for Android Studio



8. Get to the last part of that flow

9. Select the Build Variants you want to generate
   a. *Command (⌘) + Click* to select multiple Build Variants on Mac
   b. *Control (Ctrl) + Click* to select multiple Build Variants on PC



10. Select **Finish**

11. Collect the builds from the Destination Folder that you set in the **Generate Signed Bundle / APK...** process

As an alternative to Google In-App Review

If you prefer the alternative rating prompt compared to the Google In-App Review, this section help with the why and how.

> Each app build can use the Google In-App Review *or* the alternative rating prompt **Not both.**

Why shouldn't you use the Alternative Rating Prompt instead of In-App Review?

The main reasons why someone would **not** want to use the Rating Dialog instead of Google In-App Review:

- Google In-App Review is **the recommended solution by Google** to collect user reviews
- With Google In-App Review the end customer can review the app without leaving it
- Google In-App Review won't bother you if you've already reviewed it
  – This information is immediately available to Google
  – The Rating Dialog will not show for you again if the end customer selects the **Rate** or

Decline option, but has no knowledge if they did review

Why use the Alternative Rating Prompt instead of In-App Review?

The main reasons why someone would want to use the alternative rating prompt instead of Google In-App Review:

- Google In-App Review has internal limitations on how often it can be shown
  – These cannot be changed or queried about
  – These are there for the benefit of the app user
- You want to provide a more personal message
- You can customize the ***look* and *feel*** of the alternative rating prompt.
  – See the Interface Customization and Cookbook Overview articles for more info and ideas

Tutorial

1. Find your Play Store URL
   a. Google Chrome example: https://play.google.com/store/apps/details?id=com.android.chrome

2. Create your `ApptentiveConfiguration` within your `Application` class

3. Set your Play Store URL as a `String` to `ApptentiveConfiguration`'s `customAppStoreURL`

4. Register Alchemer Mobile `Apptentive.register(this, configuration)`

```
class MyApplication : Application() {
  override fun onCreate() {
    super.onCreate()
    val configuration = ApptentiveConfiguration(
      apptentiveKey = "YOUR_APPTENTIVE_KEY",
      apptentiveSignature = "YOUR_APPTENTIVE_SIGNATURE"
    )
    configuration.customAppStoreURL = "https://play.google.com/store/apps/details?id=com.android.chrome
"
    Apptentive.register(this, configuration)
  }
}
```

Related Articles