

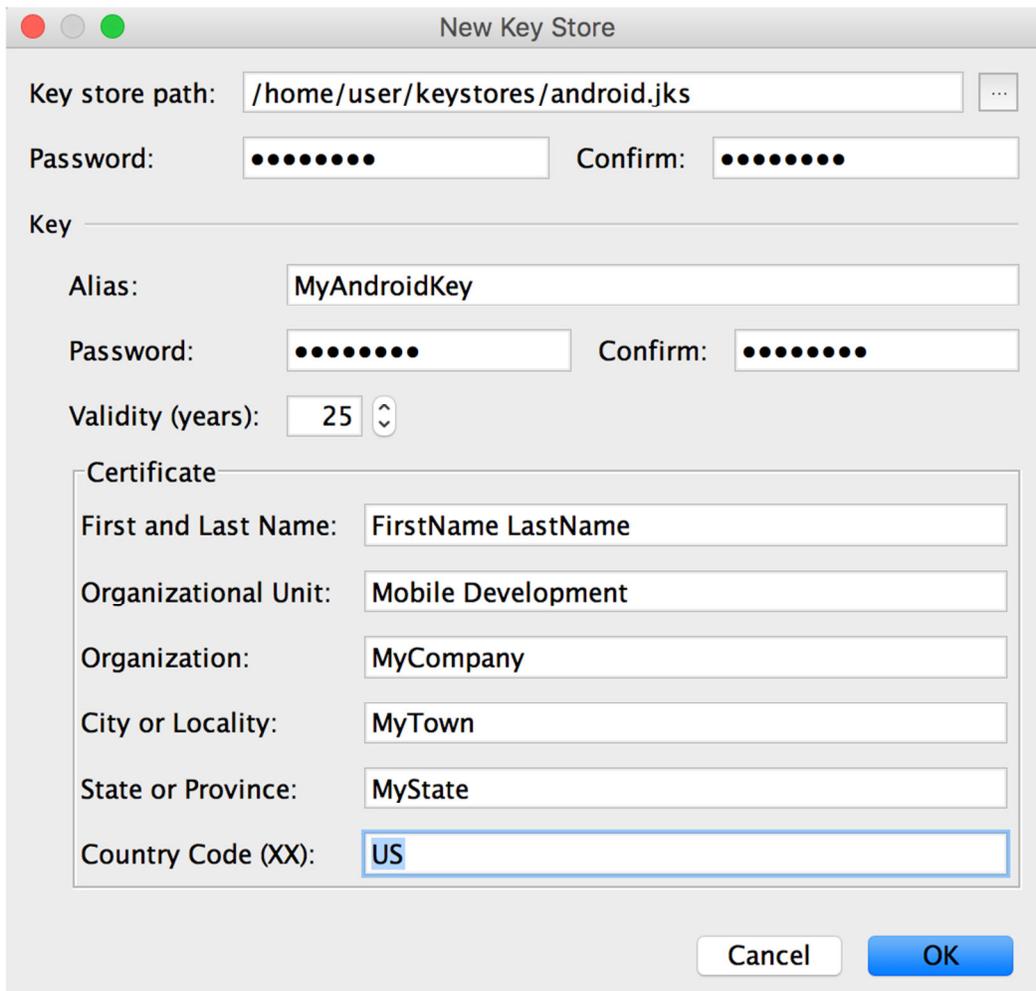
Sign Your Release Build

In order to share your apk file you need to sign it first.

Modified from <https://developer.android.com/studio/publish/app-signing.html#release-mode>

To manually sign your APK for release in Android Studio, follow these steps:

1. In the menu bar, click **Build > Generate Signed APK**.
2. Click Create new Key
3. On the **New Key Store** window, provide the following information for your keystore and key, as shown in figure 1.



The screenshot shows the 'New Key Store' dialog box with the following fields and values:

- Key store path: /home/user/keystores/android.jks
- Password: [masked]
- Confirm: [masked]
- Key section:
 - Alias: MyAndroidKey
 - Password: [masked]
 - Confirm: [masked]
 - Validity (years): 25
- Certificate section:
 - First and Last Name: FirstName LastName
 - Organizational Unit: Mobile Development
 - Organization: MyCompany
 - City or Locality: MyTown
 - State or Province: MyState
 - Country Code (XX): US

Buttons: Cancel, OK

Figure 1. Create a new keystore in Android Studio.

Keystore

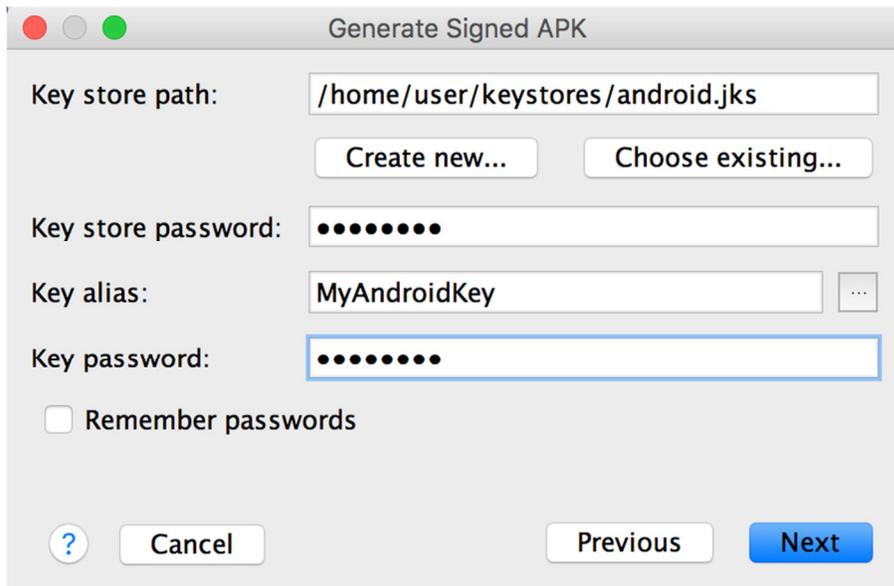
- **Key store path:** Select the location where your keystore should be created.
- **Password:** Create and confirm a secure password for your keystore.

Key

- **Alias:** Enter an identifying name for your key.
- **Password:** Create and confirm a secure password for your key. This should be different from the password you chose for your keystore
- **Validity (years):** Set the length of time in years that your key will be valid. Your key should be valid for at least 25 years, so you can sign app updates with the same key through the lifespan of your app.
- **Certificate:** Enter some information about yourself for your certificate. This information is not displayed in your app, but is included in your certificate as part of the APK.

Once you complete the form, click **OK**.

1. On the **Generate Signed APK Wizard** window, select a keystore, a private key, and enter the passwords for both. (If you created your keystore in the last step, these fields are already populated for you.) Then click **Next**.



The screenshot shows the "Generate Signed APK" dialog box. It has a title bar with standard window controls (red, yellow, green buttons) and the text "Generate Signed APK". Below the title bar, there are four main input sections: "Key store path:" with a text field containing "/home/user/keystores/android.jks" and two buttons "Create new..." and "Choose existing..."; "Key store password:" with a text field containing ten black dots; "Key alias:" with a text field containing "MyAndroidKey" and a small "..." button to its right; and "Key password:" with a text field containing ten black dots. At the bottom left, there is a checkbox labeled "Remember passwords" which is currently unchecked. At the bottom center and right, there are three buttons: a help button with a question mark icon, a "Cancel" button, and a "Next" button which is highlighted in blue.

Figure 2. Select a private key in Android Studio.

- On the next window, select a destination for the signed APK(s), select the build type, (if applicable) choose the product flavor(s), and click **Finish**.

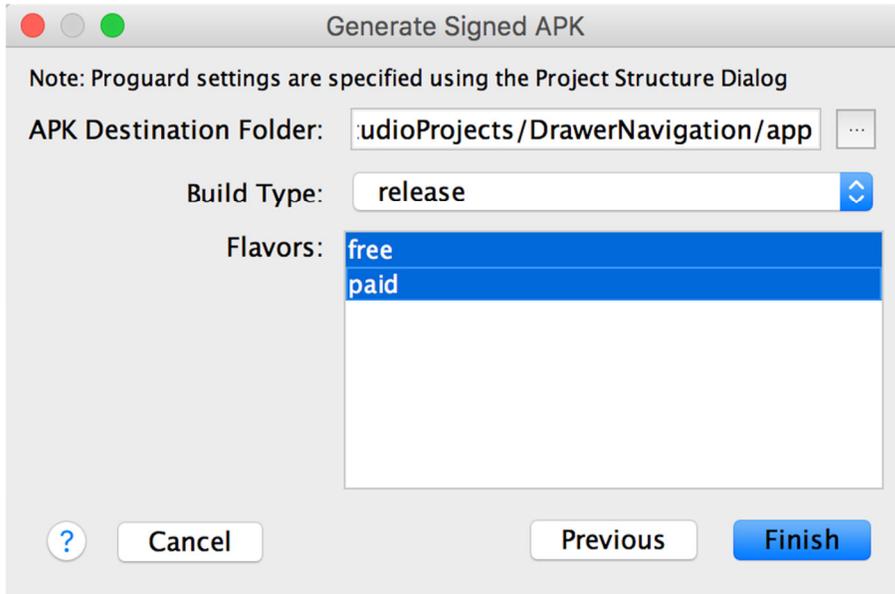


Figure 3. Generate signed APKs for the selected product flavors.

Note: If your project uses product flavors, you can select multiple product flavors while holding down the **Ctrl** key on Windows/Linux, or the **Command** key on Mac OSX. Android Studio will generate a separate APK for each selected product flavor.

When the process completes, you will find your signed APK in the destination folder you selected above. You may now distribute your signed APK through an app marketplace like the Google Play Store, or using the mechanism of your choice.