

SETUP MOBILE APPLICATION ENVIRONMENT

Team ID	PNT2022TMID08097
Project Name	Containment Zone Alerting Application

Build the APK File For The Project

Android Studio allows you to create two kinds of APK files.

First are the **debug APK** files that are generated solely for testing purposes. They will run on your Android mobile. However, they cannot be uploaded to the Play Store or made available to the public.

Secondly, you can generate **signed APK** files. Signed APK files come in handy when you've tested your application and it is ready to be uploaded on the Play Store and released to the general public.

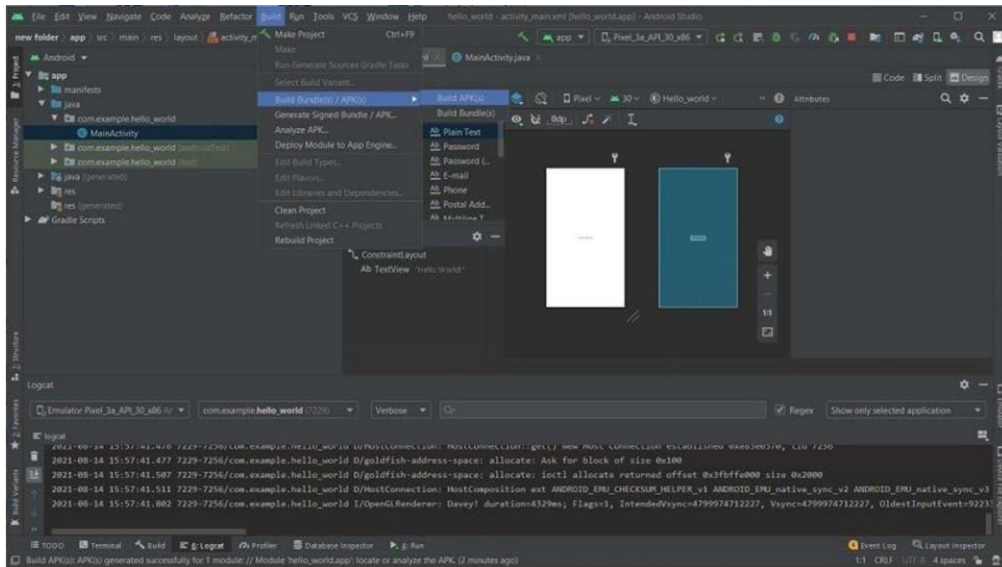
This tutorial will show you how to create an Android app by generating APK files using Android Studio.

First things first: open up a project file in Android Studio. If you don't have a project file yet, simply create a [New Project](#).

Creating an APK file

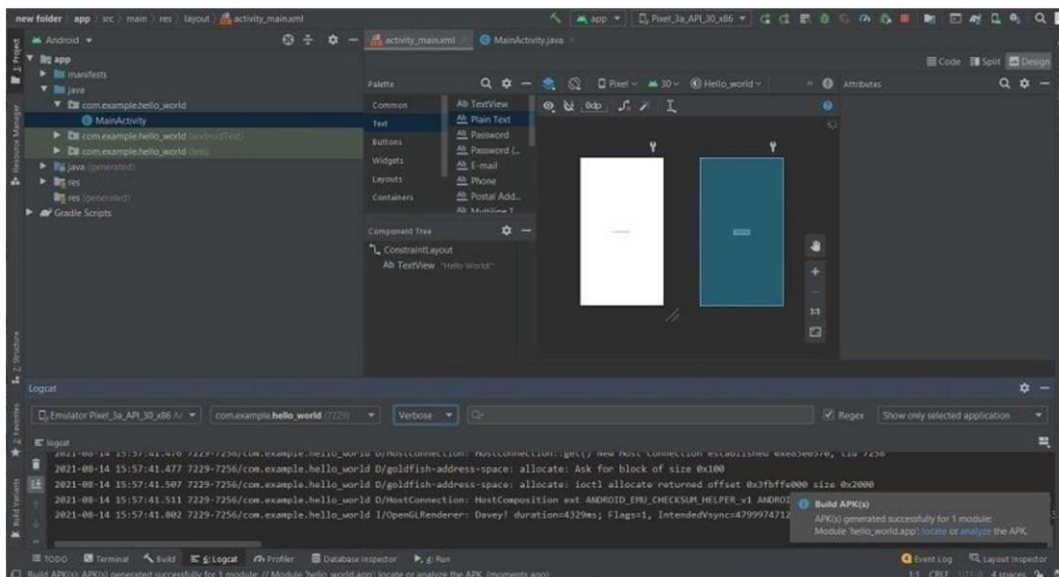
Generating a debug APK file is easy and is a matter of just a few clicks.

First, open up your project or application that you want to import into an APK file. Then, select **Build > Build Bundle(s)/APK(s) > Build APK(s)** from the toolbar menu.



Android Studio will take a few moments to generate an APK file.

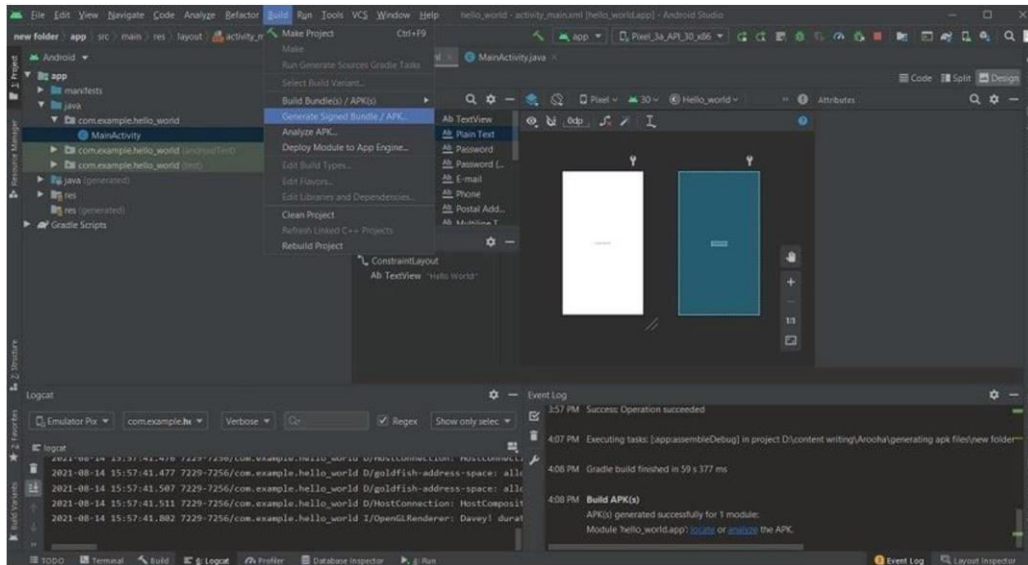
Once the APK build is complete, you'll receive a notification on the bottom right corner of your screen. From that notification, select **Locate** and you will be led to the APK file location.



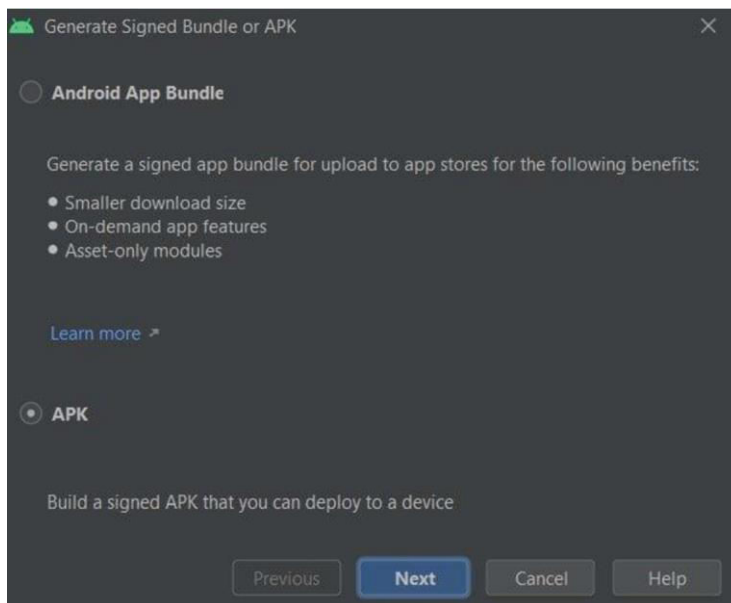
If you miss the notification, you can still locate the APK file in the following path within your project folder: **app/build/outputs/apk/debug**. The file is named **appdebug.apk** by default.

Creating a Signed APK File

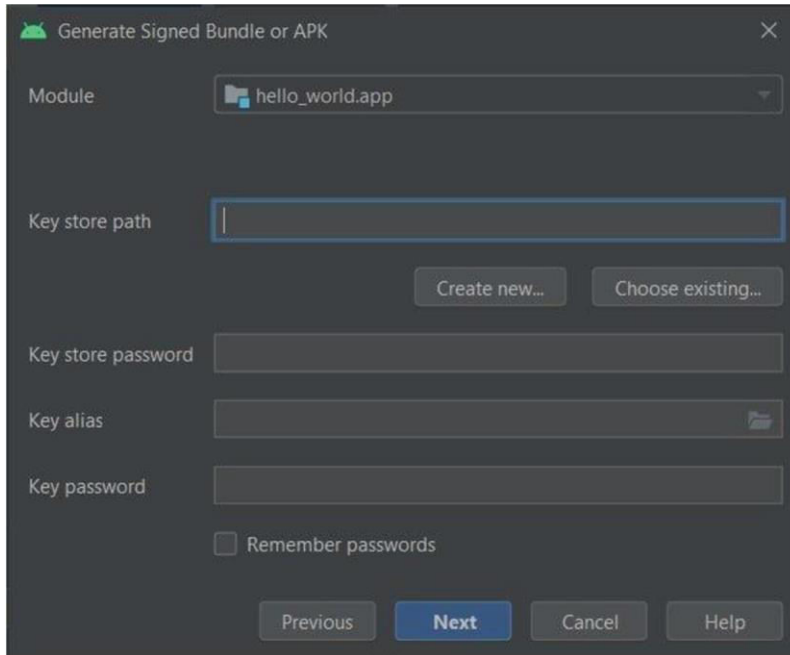
To generate a signed APK file, open the **Build** menu from the toolbar and select **Generate Signed Bundle/APK**.



This opens up a screen where you have to select between creating an [Android App Bundle](#) and creating an APK file. Check the **APK** radio button and proceed to the next window.



In the next window, you will be shown the module (your application) for which the APK file is being generated. You'll be asked about your **Key store path**, **Key store password**, **Key alias**, and the **Key password**.



The screenshot shows the 'Generate Signed Bundle or APK' dialog box. At the top, the title bar says 'Generate Signed Bundle or APK' with a close button. Below the title bar, there is a 'Module' dropdown menu showing 'hello_world.app'. Underneath, there is a 'Key store path' text field, which is currently empty and highlighted with a blue border. To the right of this field are two buttons: 'Create new...' and 'Choose existing...'. Below the 'Key store path' field is a 'Key store password' text field. Below that is a 'Key alias' text field with a folder icon on the right. Below the 'Key alias' field is a 'Key password' text field. At the bottom left, there is a checkbox labeled 'Remember passwords'. At the bottom right, there are four buttons: 'Previous', 'Next' (which is highlighted in blue), 'Cancel', and 'Help'.

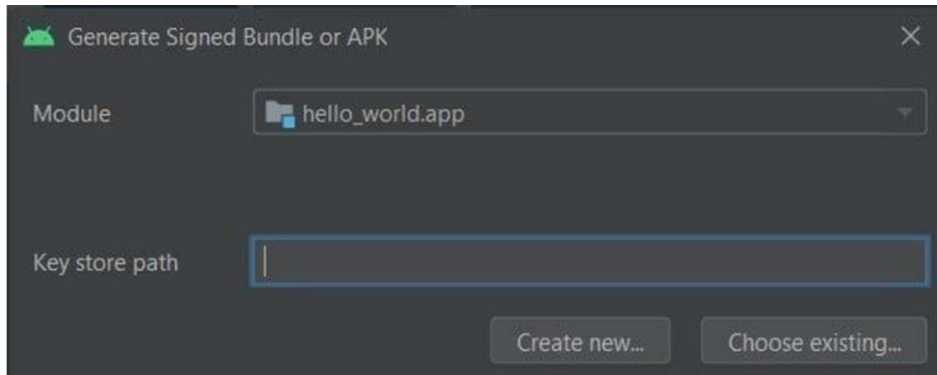
Creating a New Key Store

Assuming that this is the first time you're creating a **Signed APK** file, you will have to create a new key store.

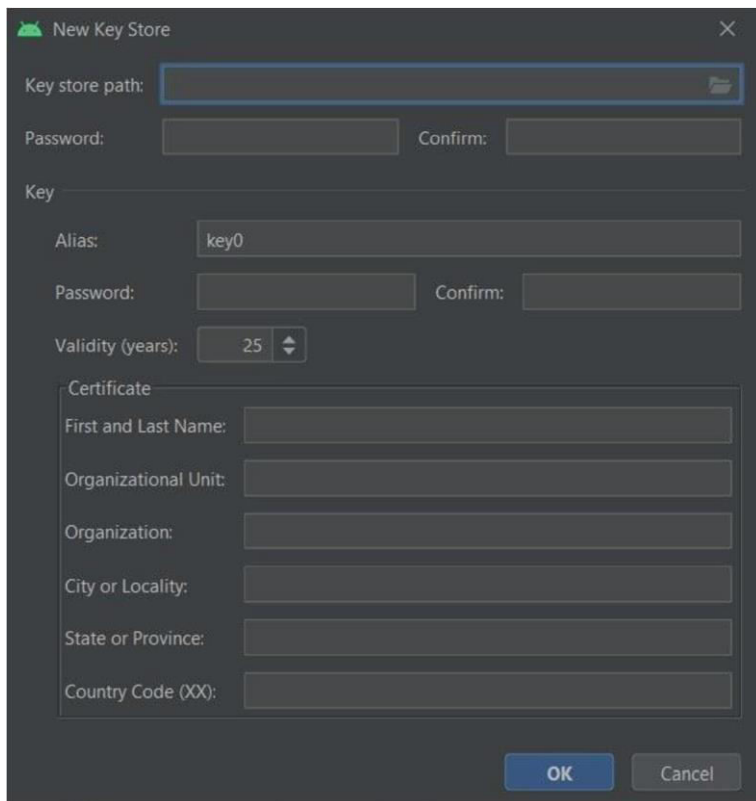
Keys are used by the developer to access their application once it has been uploaded to the Play Store. The need for the keys usually arises when you have to update your application. All of the keys are stored in the key store.

Both the key store and the keys are protected by passwords of their own. The passwords should be at least six characters in length. Also, it is a good practice to keep multiple copies of your keys since they are your only gateway to your application. If the key is lost, you will not be able to access your application or update it.

Creating your own app requires you to create a new key store. To do so, select **Create new**. You will find it underneath the input field where you enter the key store path.



You will then be redirected to a new window.



In the new window, enter the path for your new key store, and then enter a password to protect it.

In the same window, you will also be setting a new key for your application. Enter an identity for your key in the key alias field and then enter a password for it.

You can keep the same password as that of your key store, but it's a good practice to give a new password to each of your keys. The same goes for the key alias.

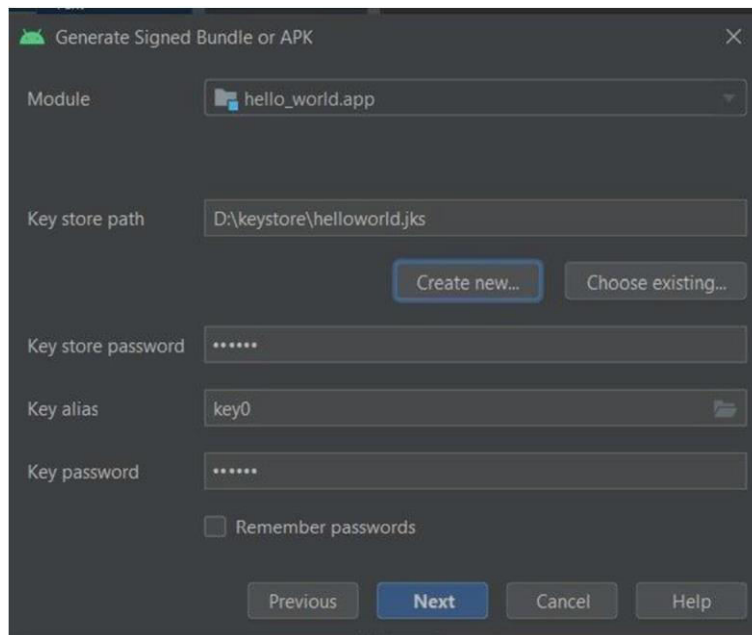
The next field defines the validity of your application. This is the duration after which the key to your application will expire, leaving your application inaccessible. The default validity for a key is 25 years.

For each key that you generate, you're given a certificate that contains all the information about you and your company. You don't necessarily have to fill in all the details-just choose the ones you think should go on your certificate. A key will still be generated, even without filling in each field of the certificate.

Finishing Up

Once you have filled in the details for the certificate, select **OK**. You will then be directed back to the **Generate Signed Bundle or APK** screen.

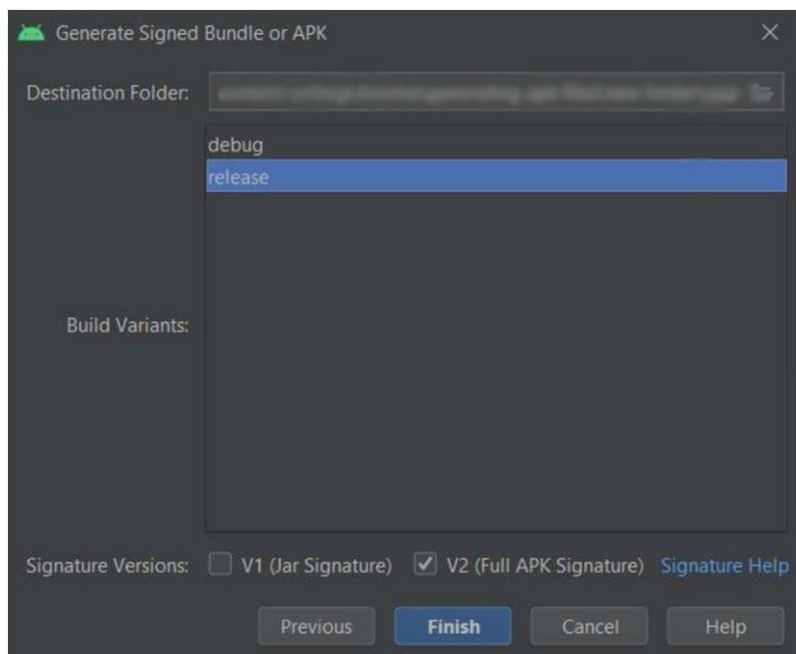
Here, all of the fields will now be pre-filled for you. Go through all the details to stay on the safe side. Then, select **Next**.



On the last screen, you will now be able to see the destination of your **Signed APK** file. Below that, you will see two more options: **Debug** and **Release**.

Debugging is used when the application is still in the testing phase. Since your application has passed the testing phase and is ready for deployment, select **Release**.

There are two more checkboxes towards the bottom of the screen. Select **V2 (Full APK Signature)** and click **Finish**.



You will be notified by Android Studio once the **APK build** is finished. Now, you can click on **Locate** from the notification to open up the file location.

The **Signed APK** file is named **app-release.apk** by default. You will find it in your project folder in the **app/release** directory.

