

# Android App Publishing to Google Play Store

Muhammad Umair

Reg No : 232202013

November 11, 2025

## Contents

|  |          |
|--|----------|
| <b>1 Learning Objectives</b>   | <b>2</b> |
| <b>2 Part 1: APK Versioning</b>  | <b>2</b> |
| 2.1 What are versionCode and versionName, and why are they important? . . . . .          | 2        |
| 2.2 What happens if you don't increase the versionCode when updating your app? . . . . . | 2        |
| 2.3 Where can you find and edit these values in Android Studio? . . . . .                | 2        |
| <b>3 Part 2: Generating a Signed Build</b>   | <b>3</b> |
| 3.1 What is a .jks (keystore) file, and why is it critical? . . . . .                    | 3        |
| 3.2 Difference between .apk and .aab files . . . . .                                     | 3        |
| 3.3 Steps to generate a signed release file in Android Studio . . . . .                  | 3        |
| 3.4 Precautions for keystore and passwords . . . . .                                     | 3        |
| <b>4 Part 3: Publishing to Google Play</b>   | <b>3</b> |
| 4.1 Requirements to open a Google Play Developer account . . . . .                       | 3        |
| 4.2 Major steps to publish an app . . . . .  | 4        |
| 4.3 Common errors or rejections and how to avoid them . . . . .                          | 4        |
| <b>5 Practical Task: Sample Android Application</b>                                      | <b>4</b> |
| 5.1 Assign versionCode and versionName . . . . .   | 4        |
| 5.2 Generate a signed release (AAB) file . . . . .                                       | 4        |
| 5.3 Screenshots (Optional) . . . . .   | 5        |

# 1 Learning Objectives

By completing this assignment, students will be able to:

- Understand how to manage version codes and version names in Android apps.
- Learn how to generate and sign an Android App Bundle (AAB) or APK file.
- Understand the Play Console workflow — including developer account setup and app publishing.
- Produce a LaTeX-based technical report explaining the full Play Store publishing process.

## 2 Part 1: APK Versioning

### 2.1 What are `versionCode` and `versionName`, and why are they important?

`versionCode` is an integer value that represents the version of the application code, used internally by Google Play to determine if one version is more recent than another.

`versionName` is a string displayed to users to identify the release version (e.g., "1.0.0").

They are important because:

- `versionCode` ensures proper app updates; Play Store uses it to detect newer versions.
- `versionName` helps users understand app updates.

### 2.2 What happens if you don't increase the `versionCode` when updating your app?

If the `versionCode` is not increased, the Google Play Store will reject the update as it will be considered older or identical to the currently published version.

### 2.3 Where can you find and edit these values in Android Studio?

You can find and edit `versionCode` and `versionName` in the `build.gradle (Module: app)` file:

```
android {  
    defaultConfig {  
        applicationId "com.example.myapp"  
        minSdkVersion 21  
        targetSdkVersion 33  
        versionCode 2  
        versionName "1.1"  
    }  
}
```

## 3 Part 2: Generating a Signed Build

### 3.1 What is a .jks (keystore) file, and why is it critical?

A .jks file is a Java Keystore file containing security keys used to sign an Android app. It is critical because:

- It authenticates the app's publisher.
- Updates must be signed with the same keystore; otherwise, Play Store rejects them.

### 3.2 Difference between .apk and .aab files

- **APK:** The Android application package, installable directly on devices.
- **AAB:** Android App Bundle, a publishing format that allows Google Play to generate optimized APKs for each device.

### 3.3 Steps to generate a signed release file in Android Studio

1. Open Android Studio and your project.
2. Go to `Build > Generate Signed Bundle / APK`.
3. Choose `Android App Bundle` and click `Next`.
4. Select or create a new keystore (.jks), enter passwords.
5. Select the release build variant and finish.
6. Android Studio generates a signed .aab file in your chosen folder.

### 3.4 Precautions for keystore and passwords

- Keep keystore and passwords secure and backed up.
- Never share keystore publicly.
- Losing it prevents updates to your app.

## 4 Part 3: Publishing to Google Play

### 4.1 Requirements to open a Google Play Developer account

- A Google account
- 25 USD one-time registration fee
- Acceptance of Developer Program Policies

## 4.2 Major steps to publish an app

1. Sign in to Google Play Console.
2. Create a new app and fill out required details (title, description, etc.).
3. Upload the signed .aab file.
4. Complete content rating, pricing, and distribution info.
5. Submit for review and publish.

## 4.3 Common errors or rejections and how to avoid them

- **VersionCode issue:** Ensure it is incremented.
- **Policy violations:** Follow Play Store policies carefully.
- **Missing assets or metadata:** Provide all required screenshots and descriptions.

# 5 Practical Task: Sample Android Application

## 5.1 Assign versionCode and versionName

```
versionCode 1  
versionName "1.0"
```

## 5.2 Generate a signed release (AAB) file

1. Open Build > Generate Signed Bundle / APK.
2. Select Android App Bundle.
3. Choose the keystore and provide passwords.
4. Build and locate the .aab file.

### 5.3 Screenshots (Optional)

```
android {  
    namespace 'com.example.demo'  
    compileSdk 36  
  
    defaultConfig {  
        applicationId "com.example.demo"  
        minSdk 24  
        targetSdk 36  
        versionCode 1  
        versionName "1.0"  
    }  
}
```

Figure 1: build.gradle version settings

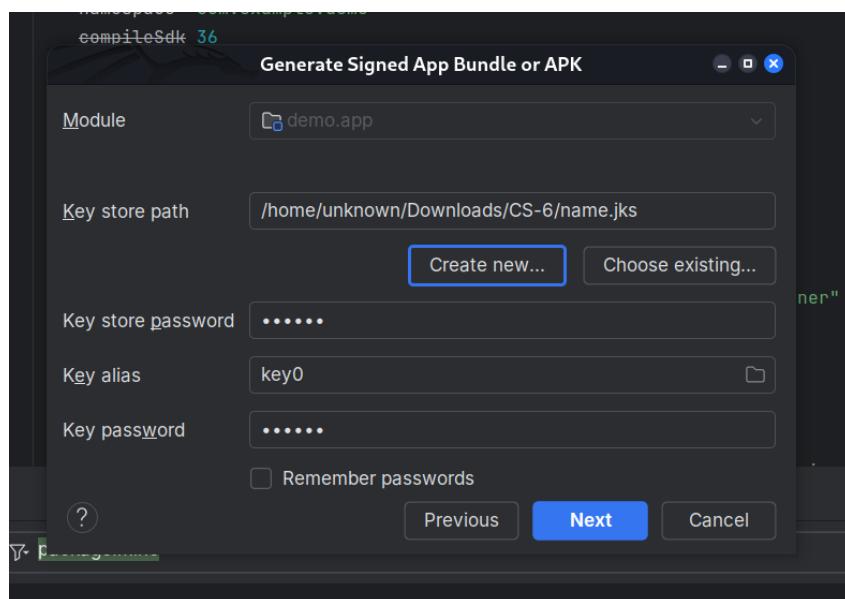


Figure 2: Generate Signed Bundle window

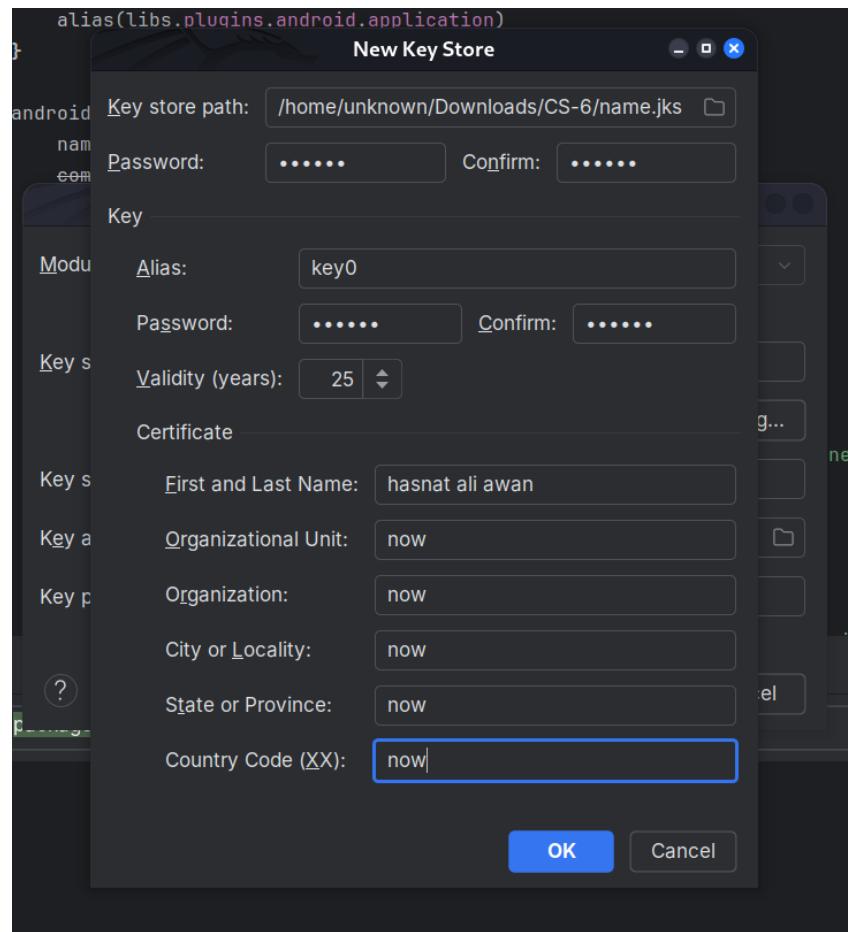


Figure 3: New Keystore creation window

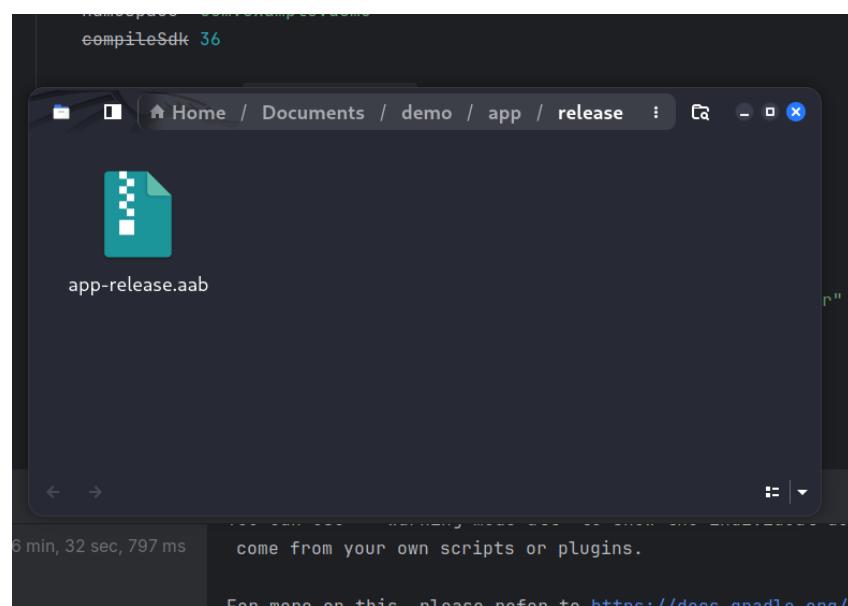


Figure 4: Resulting .aab file