Mobile App Development - Android

Working with Android Studio

Jigisha Patel

Agenda

- Setting up Android Studio
- HelloWorld Android Application
- Overview of Project Structure

Android Studio

- Android Studio is the official integrated development environment for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development.

- It is developed by Google and JetBrains. Written in Java, Kotlin and C++.

- Download and install Android Studio (latest version) from https://developer.android.com/studio/
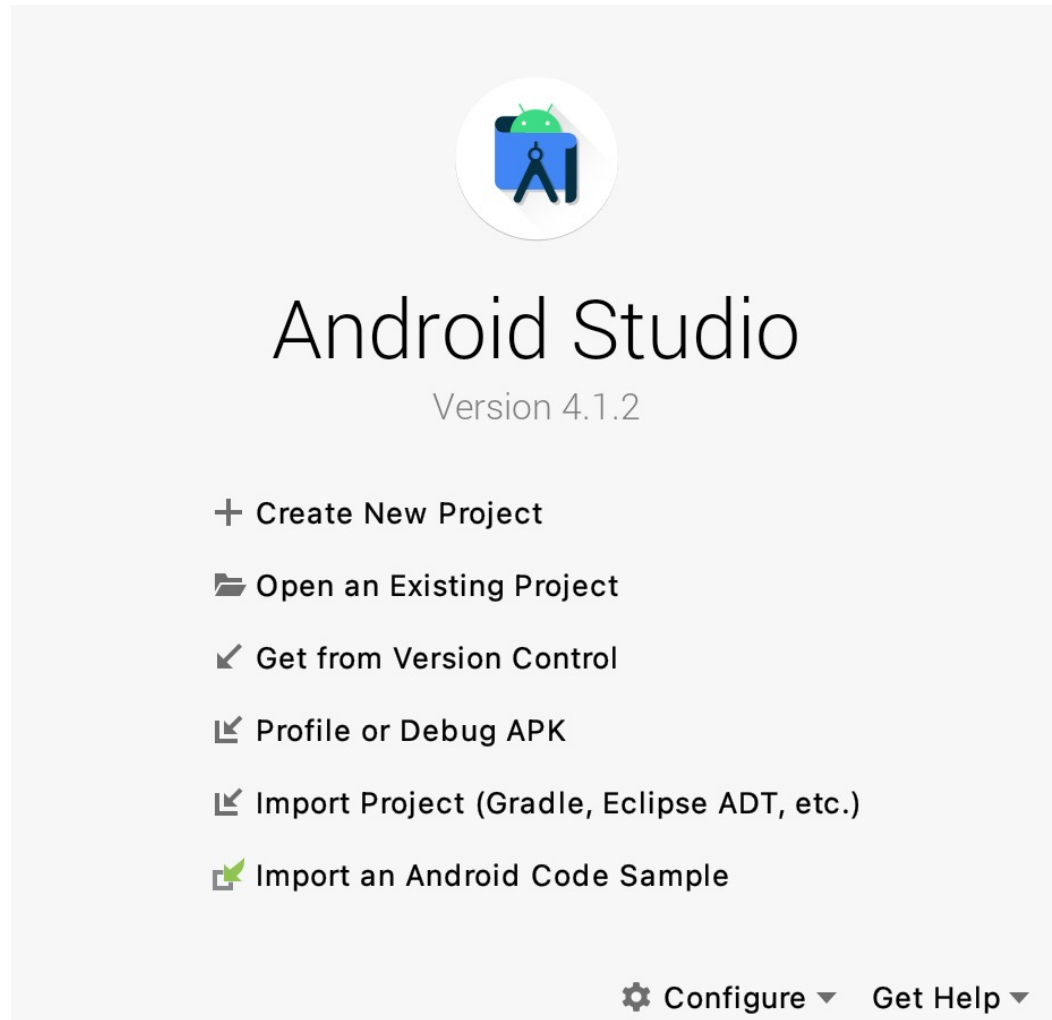
Create new project
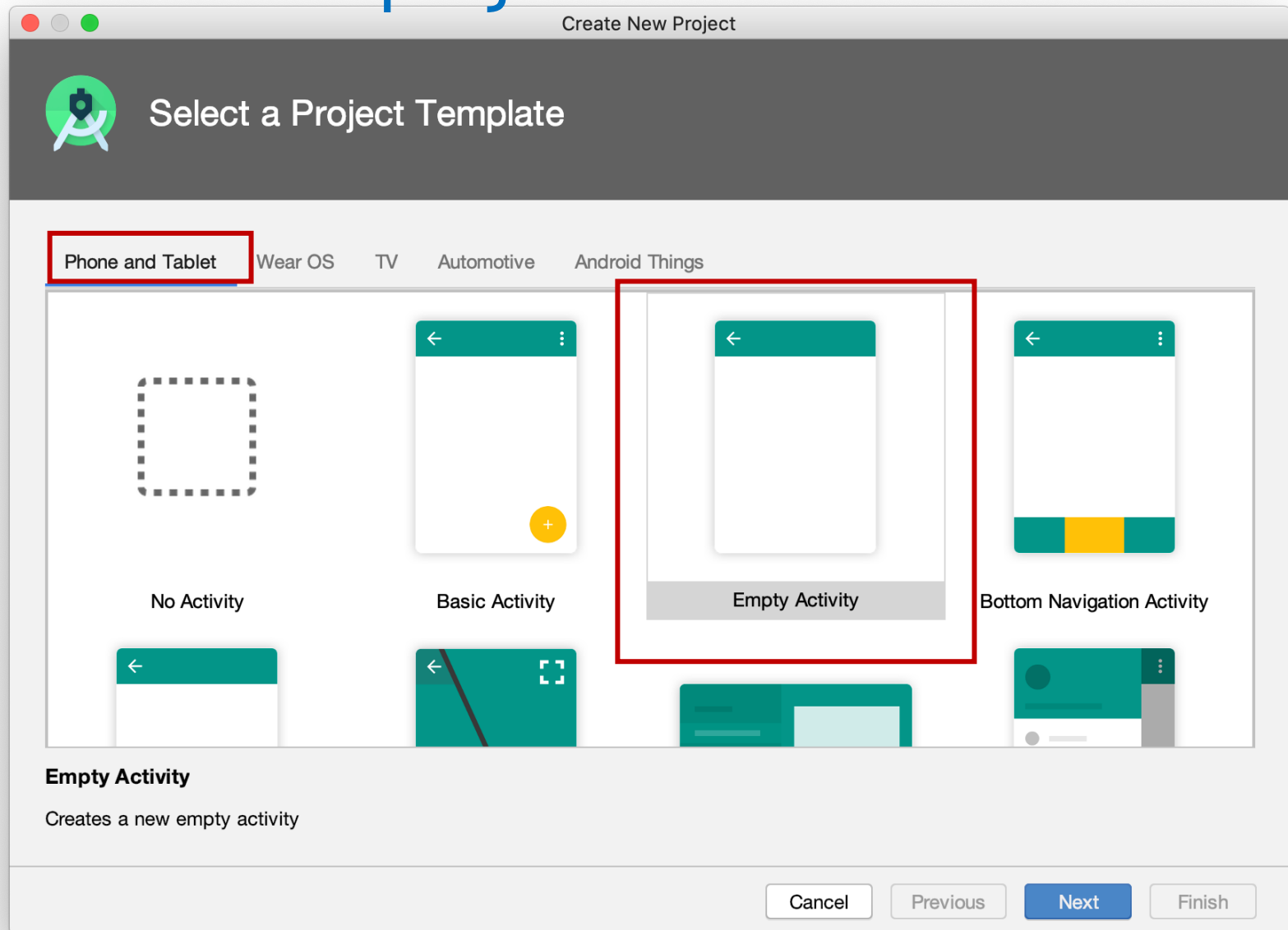
Create new project   cont...

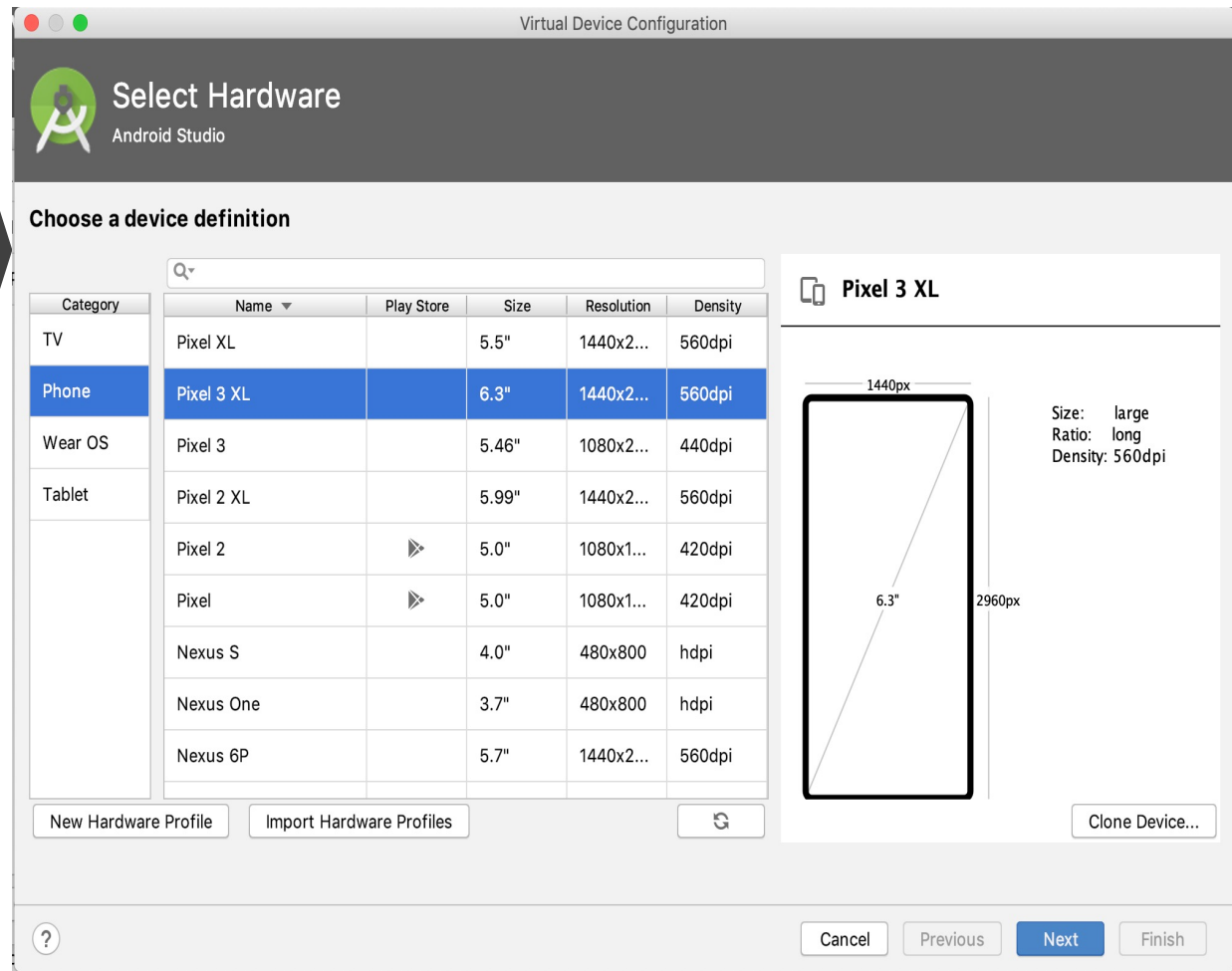# Create new project   cont…

Set up the Emulator

Setting up the Virtual Device

- Open Tools menu
- Select AVD Manager
- Click on Create Virtual Device
- Under the Category pane, Select Phone
- Select Pixel 4 XL
- Click Next

Setting up the Virtual Device

- Click on the download link for the API level 26.
- Wait for the installation to finish.
- Once the installation is complete press Next
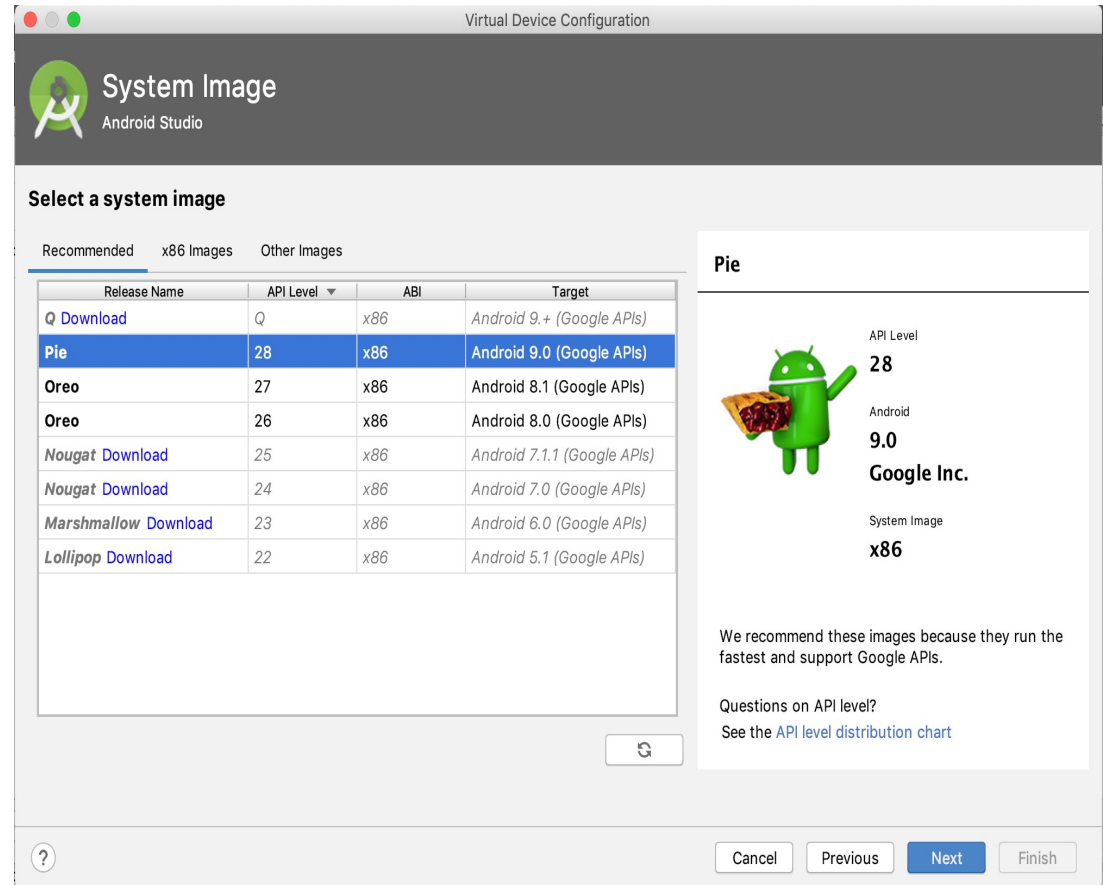
Setting up the Virtual Device

- Set the AVD name to "Pixel 3 XL API 26"
- Press Finish
- Run your app
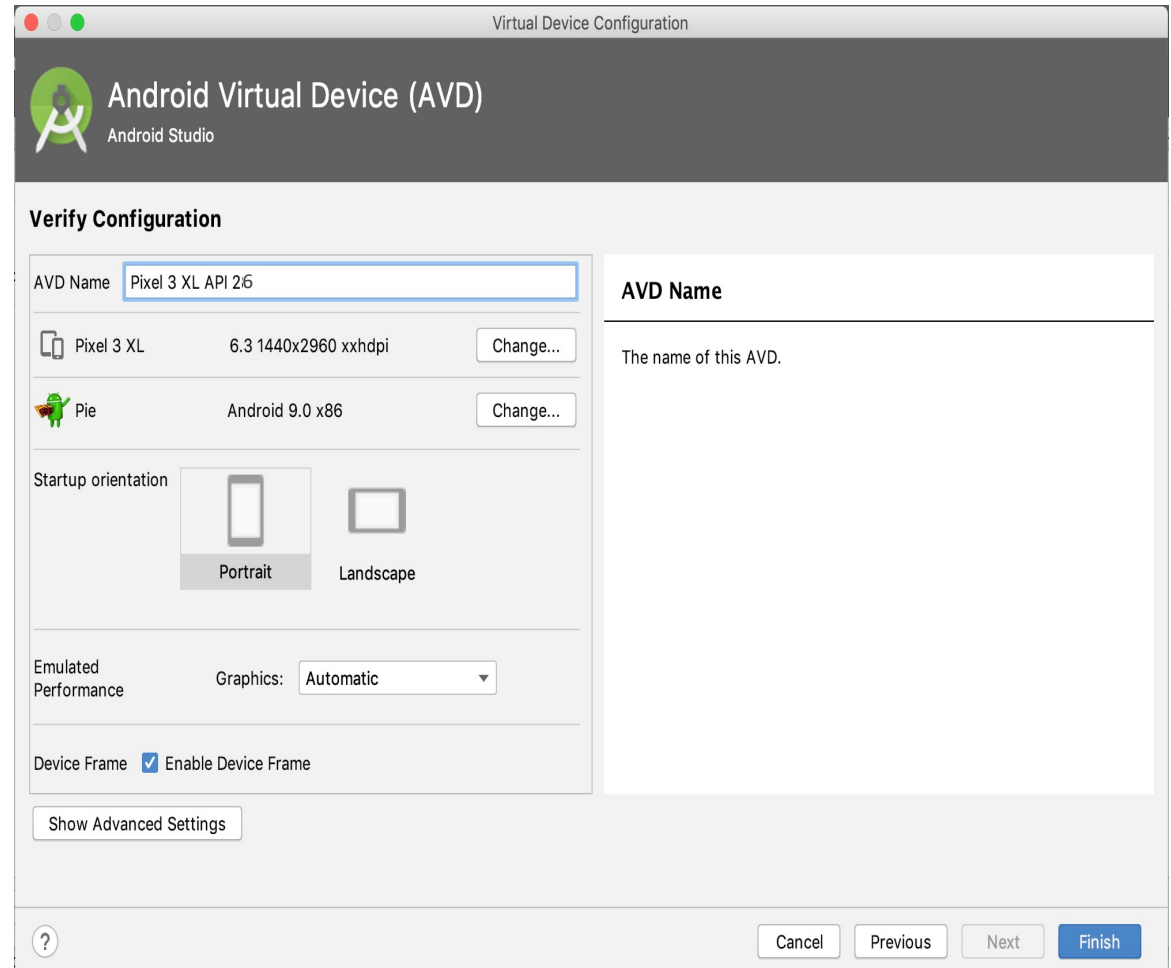
Header Comments for Files

1. On Mac OS choose Android Studio -> Preferences on Windows and Linux choose File -> Settings

2. Then look for Editor -> File and Code Templates in the left-hand pane.

3. Select the Includes tab and choose File Header.

4. Add following template

```
/**
* ${PROJECT_NAME} created by ${USER}
* student ID : your_student_id
* on ${DATE} */
```

5. Press OK.

Understanding the App structure

2. Android Manifest file

3. Drawable resources (images)

6. Gradle Scripts

Android

▼ app
 ▼ manifests
   AndroidManifest.xml
 ▼ java
   ▼ com.jk.helloworld
     C MainActivity
   ▶ com.jk.helloworld (androidTest)
   ▶ com.jk.helloworld (test)
 ▶ generatedJava
 ▼ res
   ▶ drawable
     ic_launcher_background.xml
     ic_launcher_foreground.xml (v24)
   ▼ layout
     activity_main.xml
   ▶ mipmap
     ▶ ic_launcher (6)
     ▶ ic_launcher_round (6)
   ▼ values
     colors.xml
     strings.xml
     styles.xml
 ▼ Gradle Scripts
   build.gradle (Project: HelloWorld)
   build.gradle (Module: app)
   gradle-wrapper.properties (Gradle Version)
   proguard-rules.pro (ProGuard Rules for app)
   gradle.properties (Project Properties)
   settings.gradle (Project Settings)
   local.properties (SDK Location)

1. App Source Code package

4. Layout folder for app screen UI

5. Values folder for resources such as colors, strings, styles, etc.

Java folder

- This contains the source code files for your project.

- By default, it contains an MainActivity.kt file that represents an activity that runs when the app is launched.

- The source code files in this directory are converted to DEX files by ART for executing the app.

- This directory also contains the test packages used while performing unit testing or instrumented testing of the app.

res/drawable

- This directory is used to contain the drawable objects that are used in designing of the app such as images and icons.

- The Android recommends using .png or .svg files for images.

- The name for the images must be in lowercase.

res/layout

- This directory contains the files representing the activities' user interface.

- By default, it contains activity_main.xml file.

- The layout files are written in XML format.

- You can view the layout files in Design View or Text View.

res/values

- This directory contains a set of XML files representing strings, colors and styles used in the app.

```xml
<resources>
    <string name="app_name">HelloWorld</string>
</resources>
```

```xml
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="colorPrimary">#008577</color>
    <color name="colorPrimaryDark">#00574B</color>
    <color name="colorAccent">#D81B60</color>
</resources>
```

AndroidManifest.xml

- This file acts as an interface between Android OS and the app by describing the fundamental characteristics of the app and defining each of its component.

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.jk.helloworld">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="HelloWorld"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

AndroidManifest.xml   cont...

- The <application>...</application> tags encompasses components related to app configuration such as app icon, app title, theme, right to left text support and Activities.

- The <activity> tag represents an activity of the app.

- The name attribute of <activity> tag specifies the Activity subclass that complies with the activity.

- There can be multiple <activity> tags representing multiple activities of app.

AndroidManifest.xml   cont…

- The action for the intent filter is named android.intent.action.MAIN to indicate that this activity serves as the entry point for the application.

- The category for the intent-filter is named android.intent.category.LAUNCHER to indicate that the application can be launched from the device's launcher icon.

- The AndroidManifest.xml file can have several other tags for different Android app components:
  - <activity>elements for activities
  - <service> elements for services
  - <receiver> elements for broadcast receivers
  - <provider> elements for content providers

build.gradle (Module: app)

- This is automatically generated file which contains information required to successfully compile, execute and deploy your app such as follows:
  - compileSdkVersion
  - buildToolsVersion
  - applicationId
  - minSdkVersion
  - targetSdkVersion
  - versionCode
  - versionName
  - And several dependencies.