Introduction to Android Studio & Gradle

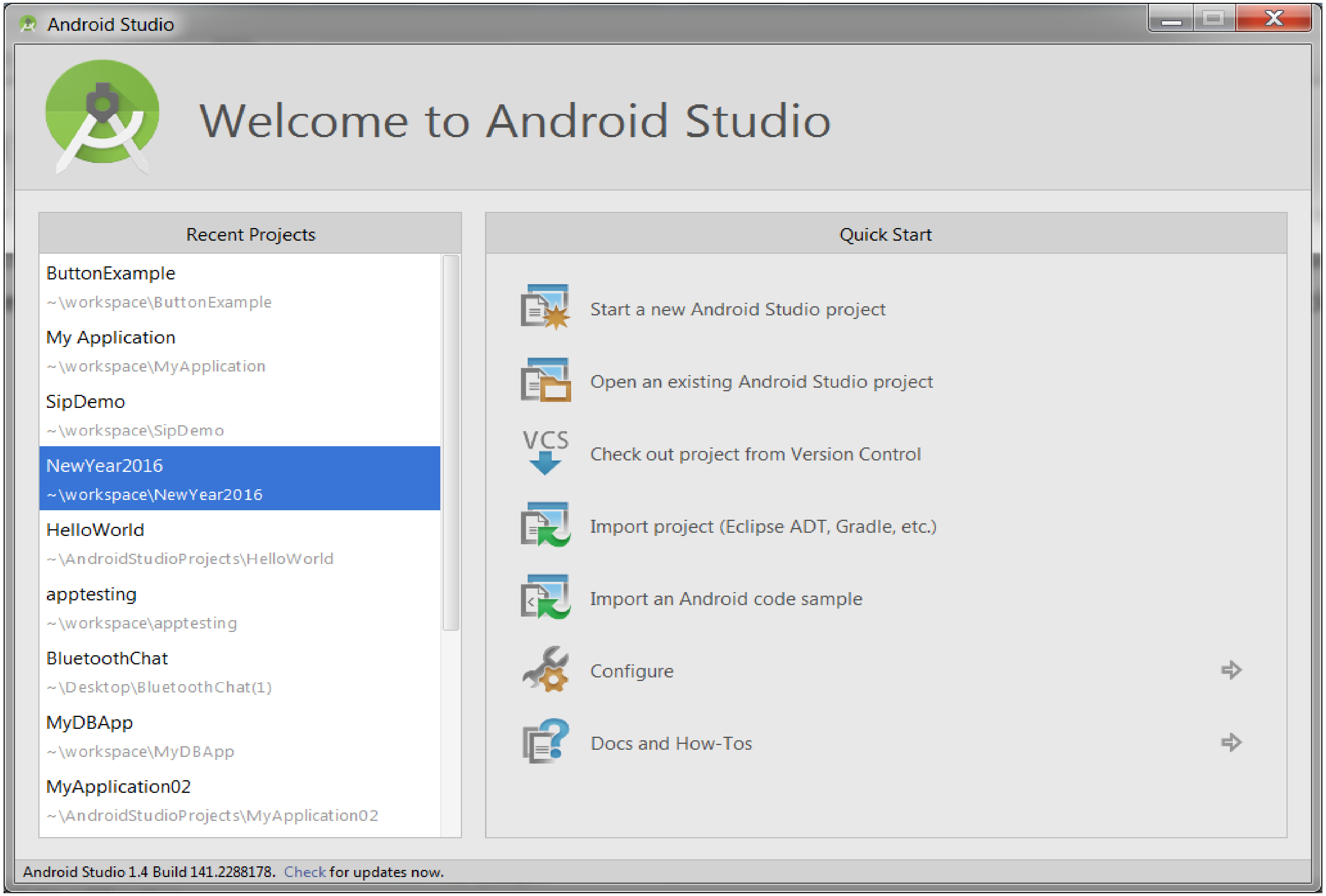
What is Android Studio?

- **Android Studio** is an integrated development environment (IDE) for developing for the Android platform.
- Android Studio is freely available under the Apache License 2.0.
- Based on JetBrains's IntelliJ IDEA software, Android Studio is designed specifically for Android development.
- It is available for download on Windows, MacOx & Linux and replaced Eclipse Android Development Tools (ADT) as Google's primary IDE for native Android application development.
- The first stable build was released in December 2014, starting from version 1.0.

2. Features of Android Studio

- Flexible Gradle-based build system
- Build variants and multiple apk file generation
- Code templates to help you build common app features
- Rich layout editor with support for drag and drop theme editing
- lint tools to catch performance, usability, version compatibility, and other problems
- ProGuard and app-signing capabilities
- Built-in support for Google Cloud Platform, making it easy to integrate Google Cloud Messaging and App Engine
- Template-based wizards to create common Android designs and components.
- Support for building Android Wear apps.
- Meaven Support

Welcome Window


Android Studio

Welcome to Android Studio

Recent Projects

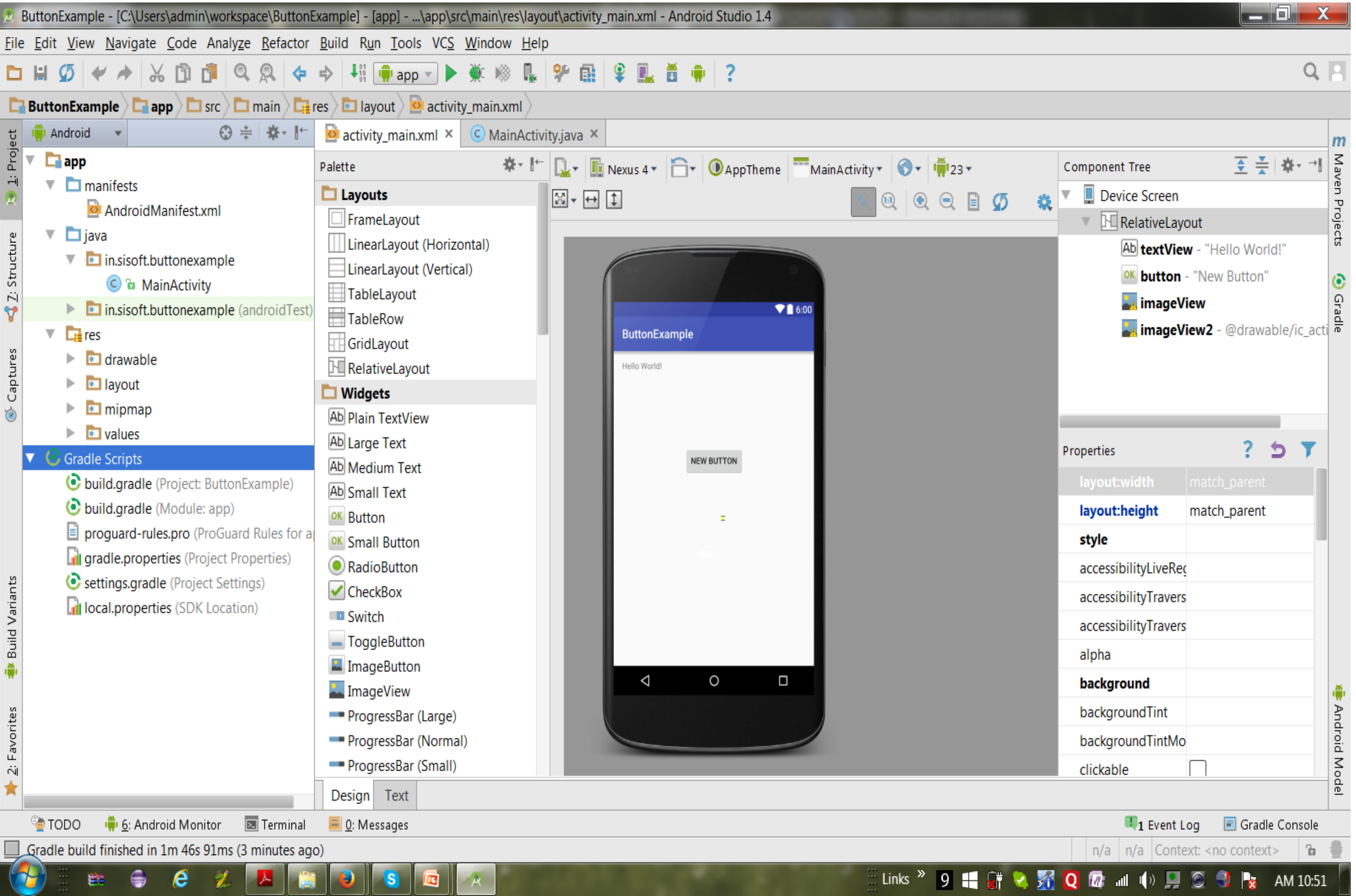
ButtonExample
~\workspace\ButtonExample
My Application
~\workspace\MyApplication
SipDemo
~\workspace\SipDemo
NewYear2016
~\workspace\NewYear2016
HelloWorld
~\AndroidStudioProjects\HelloWorld
apptesting
~\workspace\apptesting
BluetoothChat
~\Desktop\BluetoothChat(1)
MyDBApp
~\workspace\MyDBApp
MyApplication02
~\AndroidStudioProjects\MyApplication02

Quick Start

Start a new Android Studio project
Open an existing Android Studio project
Check out project from Version Control
Import project (Eclipse ADT, Gradle, etc.)
Import an Android code sample
Configure
Docs and How-Tos

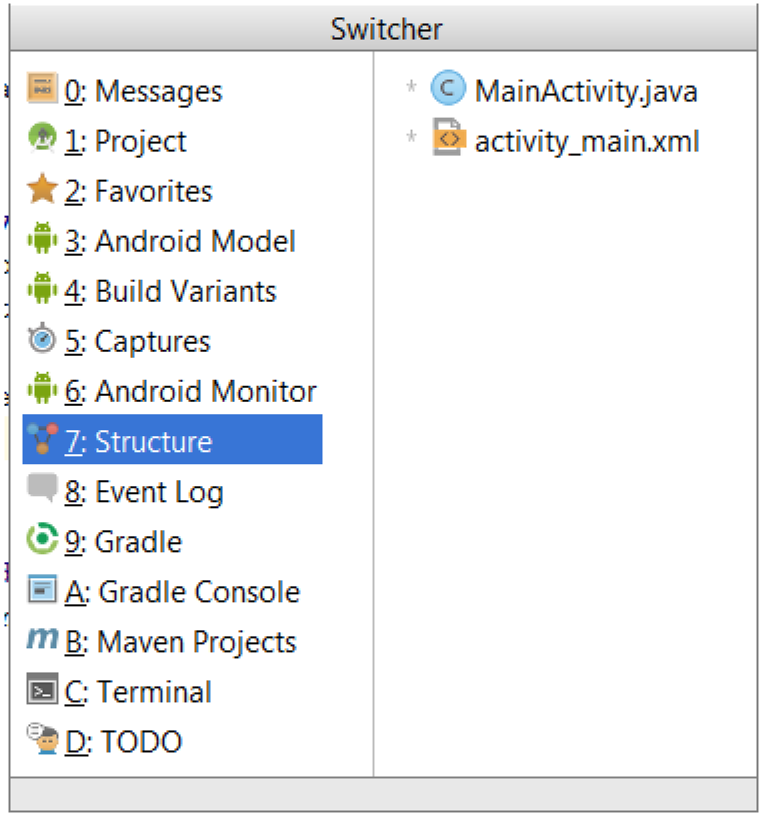
Android Studio 1.4 Build 141.2288178. Check for updates now.

Main Window


ButtonExample - [C:\Users\admin\workspace\ButtonExample] - [app] - ...\app\src\main\res\layout\activity_main.xml - Android Studio 1.4
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
ButtonExample app src main res layout activity_main.xml
Android
app
manifests
AndroidManifest.xml
java
in.sisoft.buttonexample
MainActivity
in.sisoft.buttonexample (androidTest)
res
drawable
layout
mipmap
values
Gradle Scripts
build.gradle (Project: ButtonExample)
build.gradle (Module: app)
proguard-rules.pro (ProGuard Rules for a
gradle.properties (Project Properties)
settings.gradle (Project Settings)
local.properties (SDK Location)
activity_main.xml
MainActivity.java
Palette
Layouts
FrameLayout
LinearLayout (Horizontal)
LinearLayout (Vertical)
TableLayout
TableRow
GridLayout
RelativeLayout
Widgets
Plain TextView
Large Text
Medium Text
Small Text
Button
Small Button
RadioButton
CheckBox
Switch
ToggleButton
ImageButton
ImageView
ProgressBar (Large)
ProgressBar (Normal)
ProgressBar (Small)
Nexus 4
AppTheme
MainActivity
23
ButtonExample
Hello World!
NEW BUTTON
Design
Text
Component Tree
Device Screen
RelativeLayout
textView - "Hello World!"
button - "New Button"
imageView
imageView2 - @drawable/ic_acti
Properties
layout:width match_parent
layout:height match_parent
style
accessibilityLiveReg
accessibilityTravers
accessibilityTravers
alpha
background
backgroundTint
backgroundTintMo
clickable
TODO
6: Android Monitor
Terminal
0: Messages
1 Event Log
Gradle Console
Gradle build finished in 1m 46s 91ms (3 minutes ago)
n/a n/a Context: <no context>
Links
AM 10:51

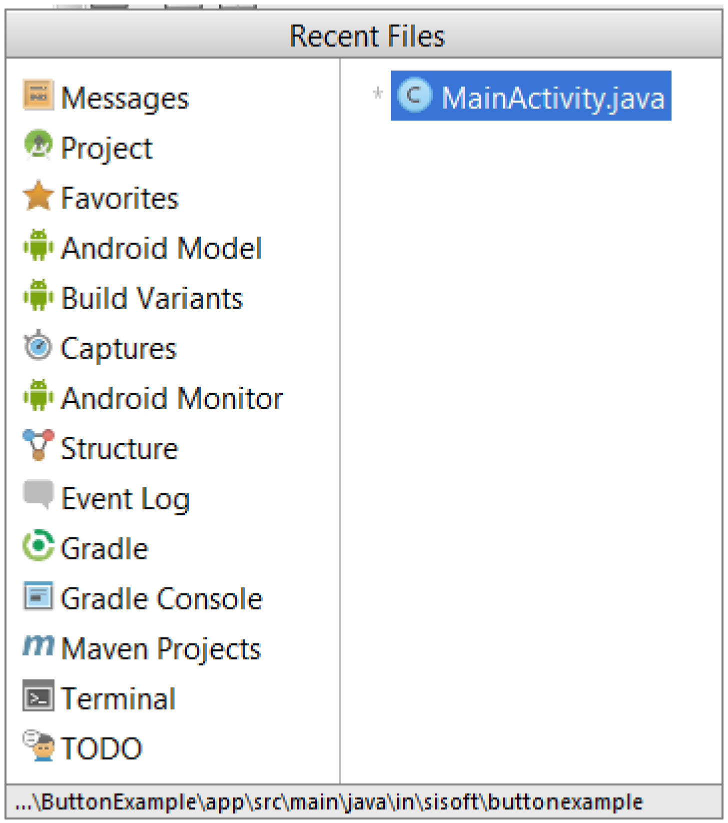
Switcher

Useful feature to navigate within Android Studio, accessed via Crtl-Tab keyboard shortcut. Visible till Crtl key is pressed



Recent Files

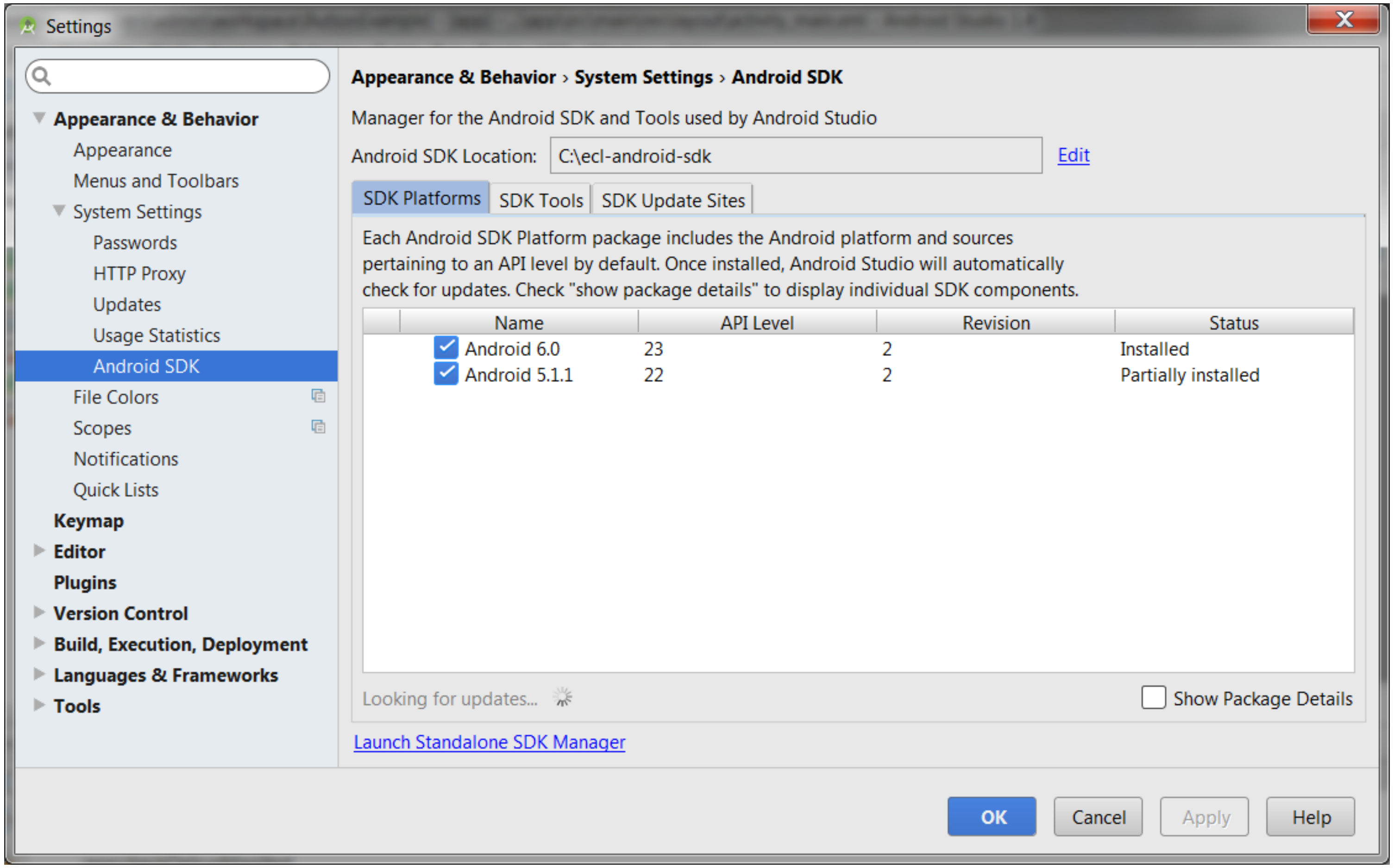
Crtl- E



...\ButtonExample\app\src\main\java\in\sisoft\buttonexample

Android Studio Setting

The overall theme of the Android Studio environment may be changed either from the welcome screen using the *Configure -> Settings option, or via the File -> Settings… menu option of the main window.*



Settings

Appearance & Behavior › System Settings › Android SDK

Manager for the Android SDK and Tools used by Android Studio

Android SDK Location: C:\ecl-android-sdk Edit

SDK Platforms | SDK Tools | SDK Update Sites

Each Android SDK Platform package includes the Android platform and sources pertaining to an API level by default. Once installed, Android Studio will automatically check for updates. Check "show package details" to display individual SDK components.

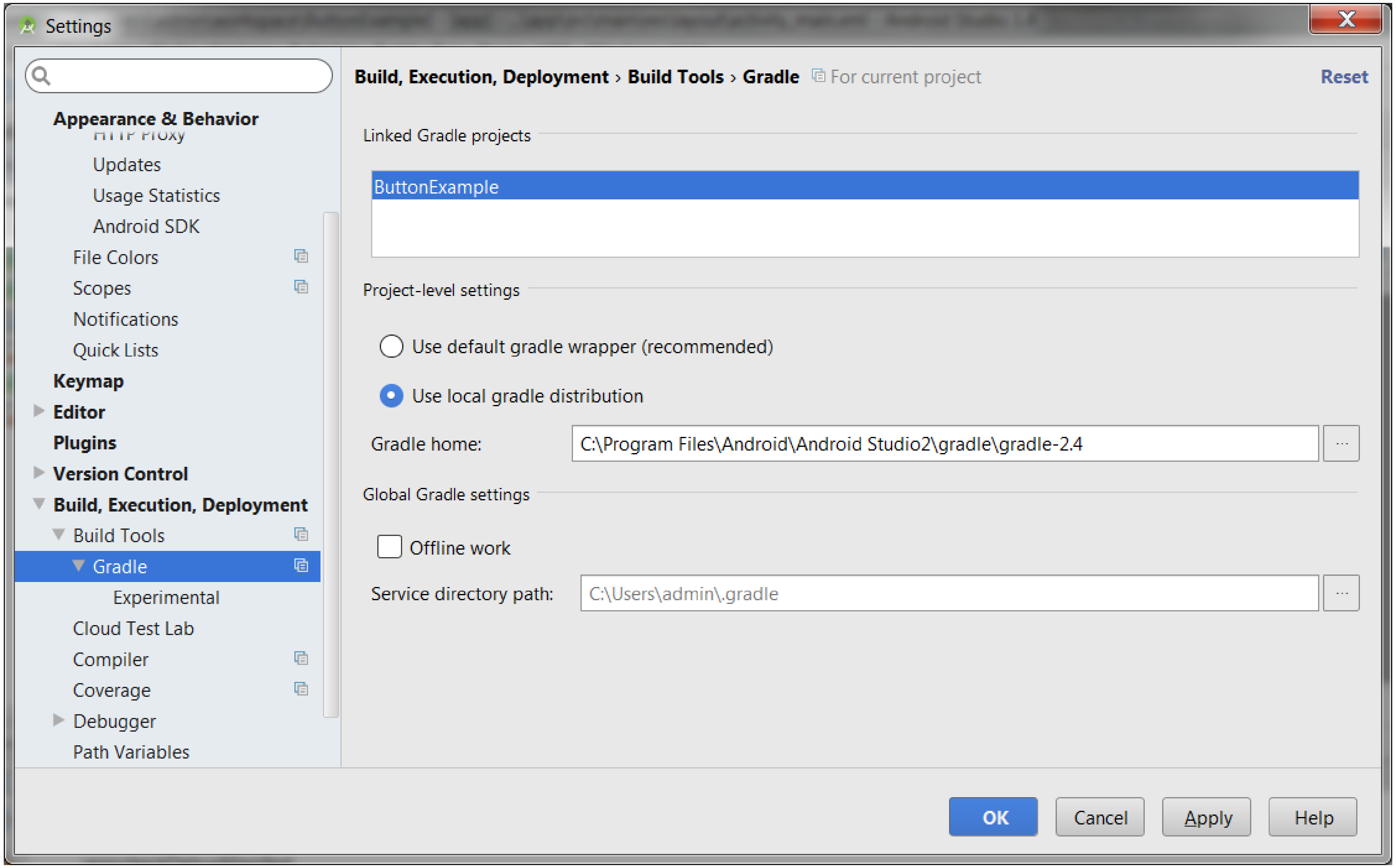
	Name	API Level	Revision	Status
✓	Android 6.0	23	2	Installed
✓	Android 5.1.1	22	2	Partially installed

Looking for updates...

☐ Show Package Details

Launch Standalone SDK Manager

Android Studio Build Tools



Build, Execution, Deployment › Build Tools › Gradle For current project

Linked Gradle projects

ButtonExample

Project-level settings

○ Use default gradle wrapper (recommended)

● Use local gradle distribution

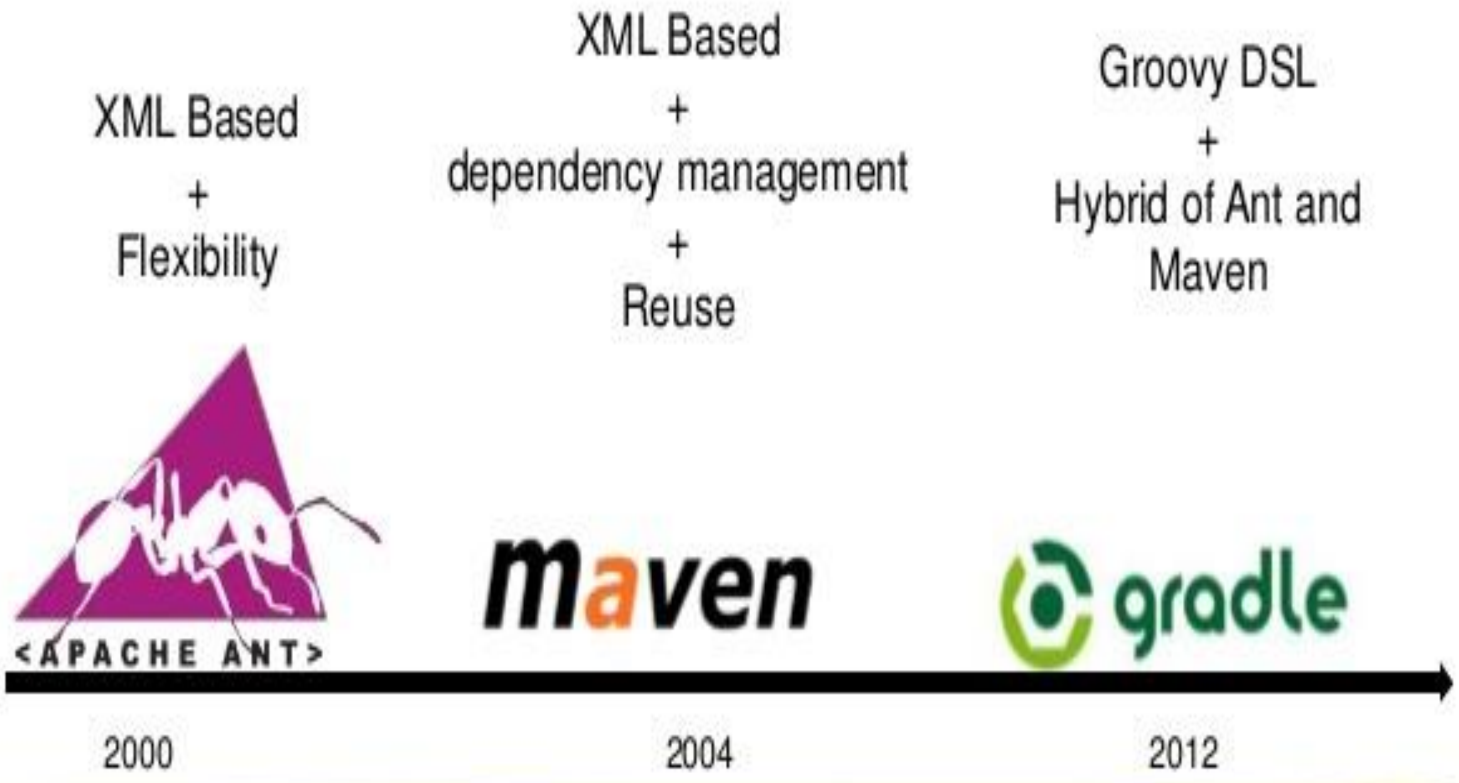
Gradle home: C:\Program Files\Android\Android Studio2\gradle\gradle-2.4

Global Gradle settings

☐ Offline work

Service directory path: C:\Users\admin\.gradle

Java Build Tools



Gradle

- Gradle is an advanced build management system which based on Groovy.

  It is an open source medium & combines the power of Maven & Ant.
- Before Android Studio we used Eclipse for development purposes & the chance may be that we did not know how to build our android project apk without even Eclipse. We can do this on the command line, but we have to learn what each tool (dx, aapt) does in the sdk. Eclipse saved us all from these low level but important, fundamental details by giving us their own build system.Now, in order to automate all these tasks, we need a script to write own build system using shell scripting in linux or batch files syntax in windows.
- Gradle is **build system** that takes the best features from other build systems and combines them into one.
- It is **JVM based build system**, means in which you can write your own script in java, which Android Studio makes use of.
- Also it is **plugin based system**. This means if you have your own programming language and you want to automate the task of building some package (output like jarin case of Java) from sources then you can write a complete plugin in Java or Groovy, and distribute it to rest of world.

2. Features & Benefits of Gradle

- **Dependency Management** : It provide flexibility approach dependency management that can reuse existing repositories or reference local JARS.

- **Simple Build System** : Gradle is the authoritative build across the Ide and the command line.

- **Product Flavors** : Free & Paid Version

- **Build Variants** : Play Store or Amazon or Any othe store.

- **Binary Bundles for Libraries (.aaar) :** It supports the new .aar binary bundle format for the library projects.

Android Plugin for Gradle

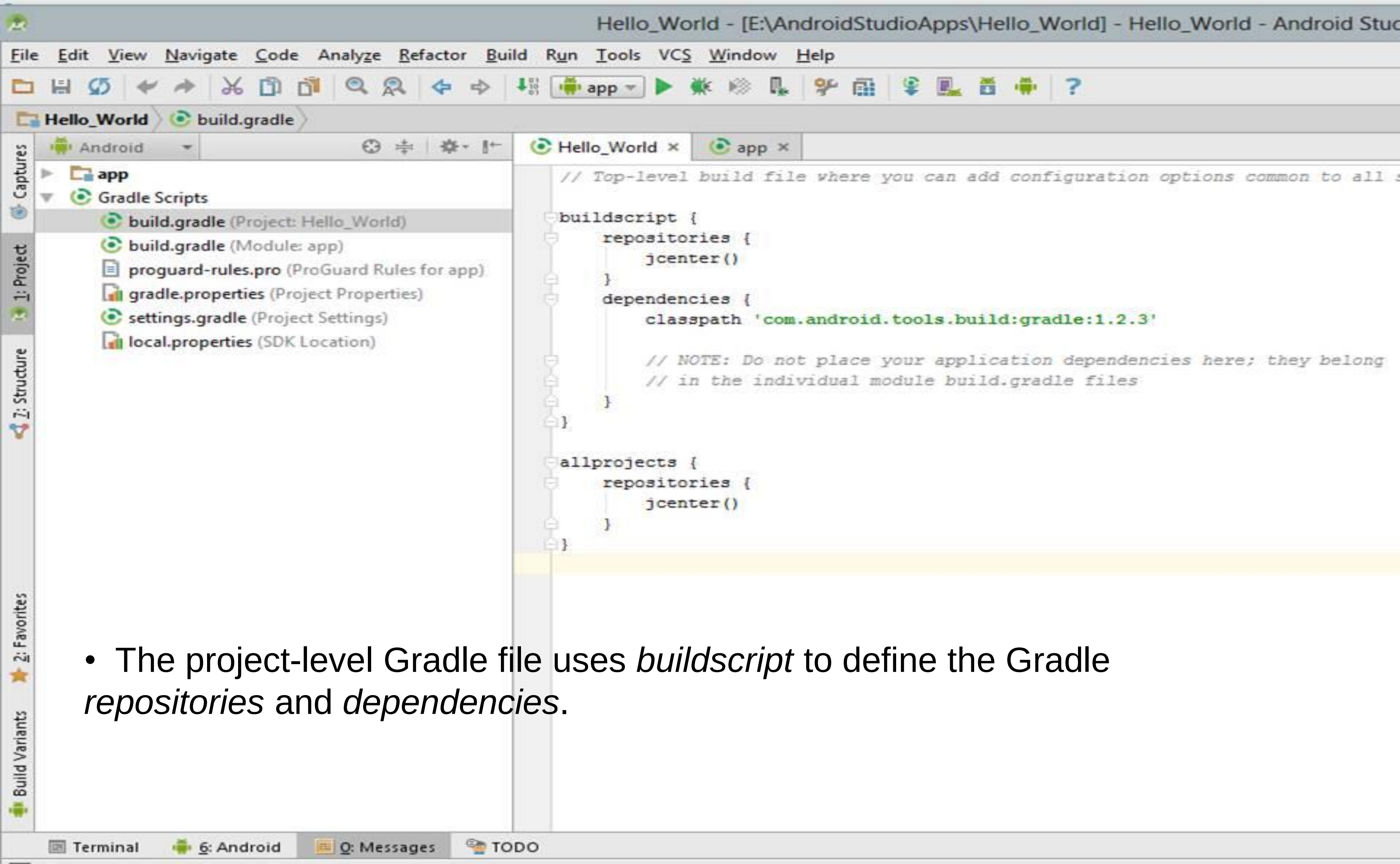
- The Android build system consists of an Android plugin for *Gradle*.
- Android Studio uses a Gradle wrapper to fully integrate the Android plugin for Gradle.
- The Android plugin for Gradle also runs independent of Android Studio. This means that you can build your Android apps from within Android Studio and from the command line on your machine or on machines where Android Studio is not installed (such as continuous integration servers).
- The build configuration for android studio project is defined inside **build.gradle** files, which are plain text files that use the syntax and options from Gradle and the Android plugin to configure
- **Gradle** build files use Domain Specific Language (DSL) to describe and manipulate the build logic through Groovy syntax. Groovy is a dynamic language that you can use to define custom build logic and to interact with the Android-specific elements provided by the Android plugin for Gradle.

Android Plugin for Gradle: properties files

- **Local.properties:** To define the SDK location for the project
  - **ndk.dir=C\:\\Users\\admin\\AppData\\Local\\Android\\sdk3\\ndk-bundle**
  - **sdk.dir=C\:\\ecl-android-sdk**

- **Gradle.properties:** To override the Gradle settings configured thru IDE

build.gradle for project

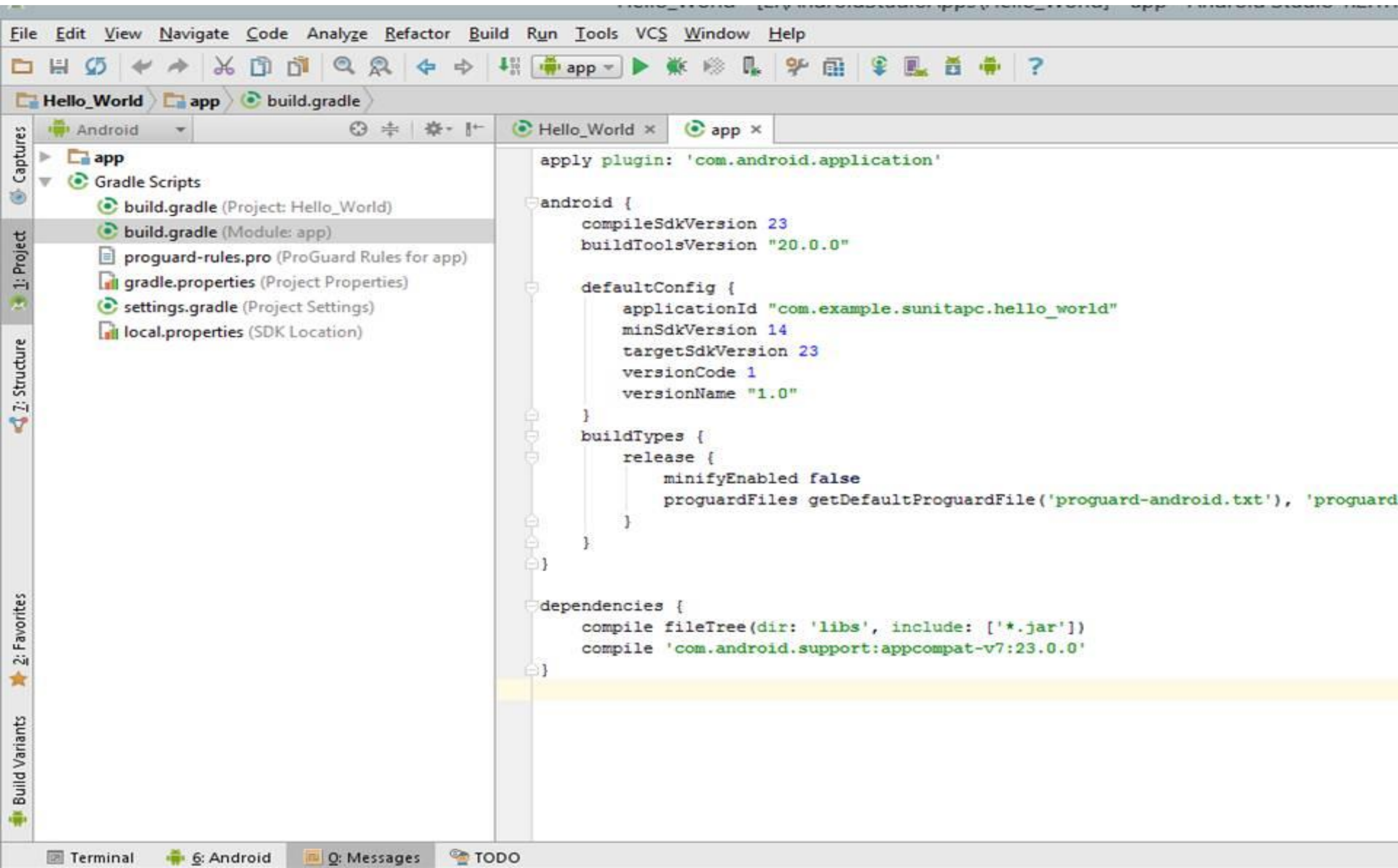


Hello_World - [E:\AndroidStudioApps\Hello_World] - Hello_World - Android Stuc
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
app
Hello_World
build.gradle
Android
app
Gradle Scripts
build.gradle (Project: Hello_World)
build.gradle (Module: app)
proguard-rules.pro (ProGuard Rules for app)
gradle.properties (Project Properties)
settings.gradle (Project Settings)
local.properties (SDK Location)
Hello_World
app
Captures
1: Project
7: Structure
2: Favorites
Build Variants
Terminal
6: Android
0: Messages
TODO

```
// Top-level build file where you can add configuration options common to all s

buildscript {
    repositories {
        jcenter()
    }
    dependencies {
        classpath 'com.android.tools.build:gradle:1.2.3'

        // NOTE: Do not place your application dependencies here; they belong
        // in the individual module build.gradle files
    }
}

allprojects {
    repositories {
        jcenter()
    }
}
```

- The project-level Gradle file uses *buildscript* to define the Gradle *repositories* and *dependencies*.

build.gradle for Module:app


File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
app
Hello_World > app > build.gradle
Android
Hello_World ×
app ×
app
Gradle Scripts
build.gradle (Project: Hello_World)
build.gradle (Module: app)
proguard-rules.pro (ProGuard Rules for app)
gradle.properties (Project Properties)
settings.gradle (Project Settings)
local.properties (SDK Location)
Captures
1: Project
7: Structure
2: Favorites
Build Variants
Terminal
6: Android
0: Messages
TODO

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 23
    buildToolsVersion "20.0.0"

    defaultConfig {
        applicationId "com.example.sunitapc.hello_world"
        minSdkVersion 14
        targetSdkVersion 23
        versionCode 1
        versionName "1.0"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard
        }
    }
}

dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    compile 'com.android.support:appcompat-v7:23.0.0'
}
```

build.gradle for Module:app

- apply plugin: 'com.android.application' applies the Android plugin for Gradle to this build
- android {...} configures all the Android-specific build options:
  - The compileSdkVersion property specifies the compilation target.
  - The buildToolsVersion property specifies what version of the build tools to use. To install several versions of the build tools, use the SDK Manager.
  - **Note:** Always use a build tools version whose major revision number is higher or equal to that of your compilation target and target SDK.

build.gradle for Module:app

- android {...}
  - The defaultConfig element configures core settings and entries in the manifest file (AndroidManifest.xml) dynamically from the build system. The values in defaultConfig override those in the manifest file.
  - The configuration specified in the defaultConfig element applies to all build variants, unless the configuration for a build variant overrides some of these values.
  - The buildTypes element controls how to build and package your app. By default, the build system defines two build types: *debug* and *release*. The debug build type includes debugging symbols and is signed with the debug key. The release build type is not signed by default.
- Dependency {… .} element declares the dependencies for this module.