apply plugin: 'android'

android {

compileSdkVersion 19

buildToolsVersion "19.1"

defaultConfig {

minSdkVersion 15

targetSdkVersion 19

versionCode 5

versionName "0.4.4"

}

compileOptions {

sourceCompatibility JavaVersion.VERSION\_1\_7

targetCompatibility JavaVersion.VERSION\_1\_7

}

signingConfigs {

release

}

buildTypes {

release {

// runProguard true

proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.txt'

signingConfig signingConfigs.release

}

}

}

dependencies {

compile fileTree(dir: 'libs', include: ['\*.jar'])

compile 'com.example:example-lib:1.0.0'

}

def propFile = file('../signing.properties')

if( propFile.canRead() ) {

def Properties p = new Properties()

p.load(new FileInputStream(propFile))

if( p!=null

&& p.containsKey("STORE\_FILE")

&& p.containsKey('STORE\_PASSWORD')

&& p.containsKey('KEY\_ALIAS')

&& p.containsKey('KEY\_PASSWORD')

) {

println "RELEASE\_BUILD: Signing..."

android.signingConfigs.release.storeFile = file( p['STORE\_FILE'] )

android.signingConfigs.release.storePassword = p['STORE\_PASSWORD']

android.signingConfigs.release.keyAlias = p['KEY\_ALIAS']

android.signingConfigs.release.keyPassword = p['KEY\_PASSWORD']

} else {

println "RELEASE\_BUILD: Required properties in signing.properties are missing"

android.buildTypes.release.signingConfig = null

}

} else {

println "RELEASE\_BUILD: signing.properties not found"

android.buildTypes.release.signingProperties = null

}