|  |  |
| --- | --- |
| Document number 文档编号 | Confidentiality level 密级 |
|  | [内部公开] |
| Document version 文档版本 | Total pages 共 页 |
|  |

Android APP构建

|  |  |  |  |
| --- | --- | --- | --- |
| Prepared by  拟制 |  | Date  日期 |  |
| Reviewed by  评审人 |  | Date  日期 | yyyy-mm-dd |
| Approved by  批准 |  | Date  日期 | yyyy-mm-dd |

Revision Record 修订记录

| Date  日期 | Revision Version  修订版本 | Sec No.  修改章节 | Change Description  修改描述 | Author  作者 |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Table of Contents 目录

[1 简介 5](#_Toc139204793)

[1.1 目的 5](#_Toc139204794)

[1.2 范围 5](#_Toc139204795)

[1.3 构建环境 5](#_Toc139204796)

[1.4 Android下载地址 5](#_Toc139204797)

[2 APP构建流程 5](#_Toc139204798)

[2.1 创建密钥 5](#_Toc139204799)

[2.2 配置Cocos Creator环境 7](#_Toc139204800)

[2.3 创建发布任务 8](#_Toc139204801)

[2.4 编辑android工程 13](#_Toc139204802)

[2.5 构建与发布 15](#_Toc139204803)

[3 定制android APP图标 19](#_Toc139204804)

Table List 表目录

**未找到目录项。**

# 简介

## 目的

文档用于指导Android APP开发、构建、测试等。

## 范围

开发人员、测试人员、项目经理。

## 构建环境

|  |  |
| --- | --- |
| 构建工具 | 版本号 |
| android studio | Android Studio 4.2.2 |
| android build tools | 29.0.2 |
| android studio SDK | 32 |
| android studio NDK | 19.2.5345600 |
| android studio CMake | 3.10.2.4988404 |
| Cocos Creator | 3.7.3 |

安装指导参考官方网站安装原生环境。

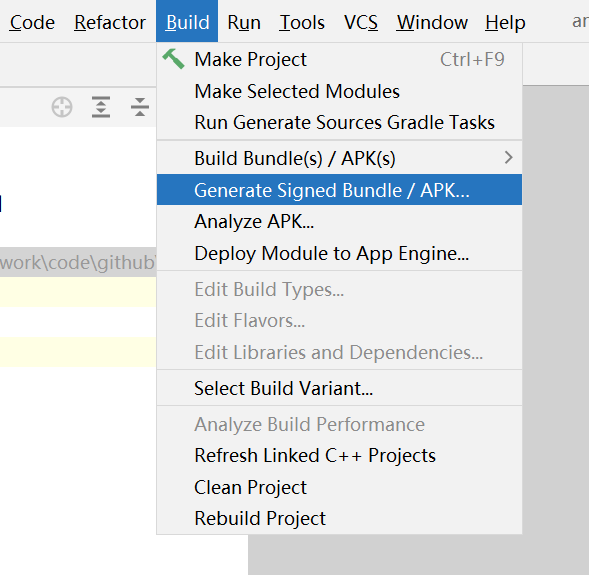
## Android下载地址

https://developer.android.google.cn/studio/archive?hl=zh-cn

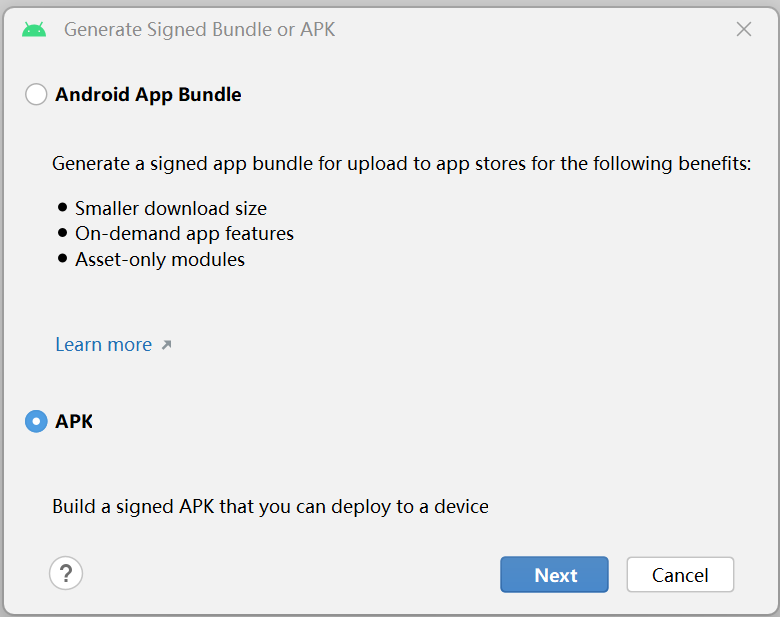
# APP构建流程

## 创建密钥

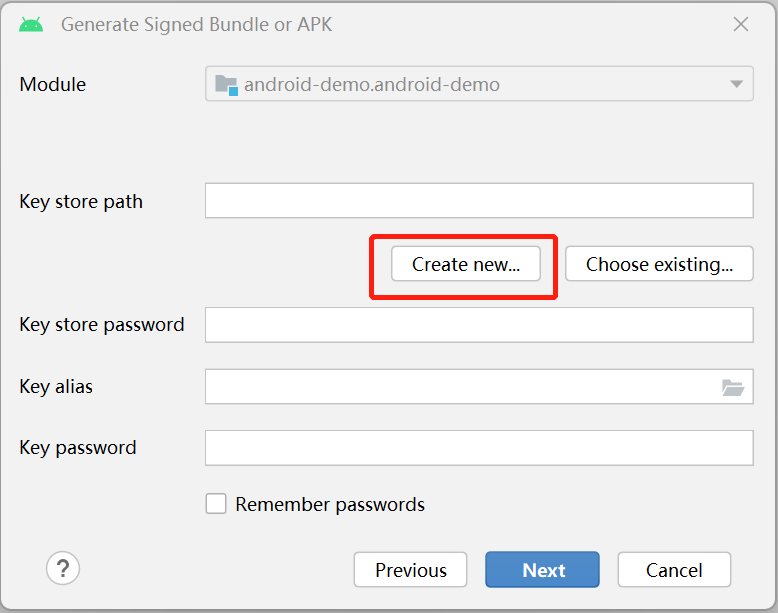
1. 在 Android Studio 中点击 Build 菜单，选中 Generate Signed Bundle / APK



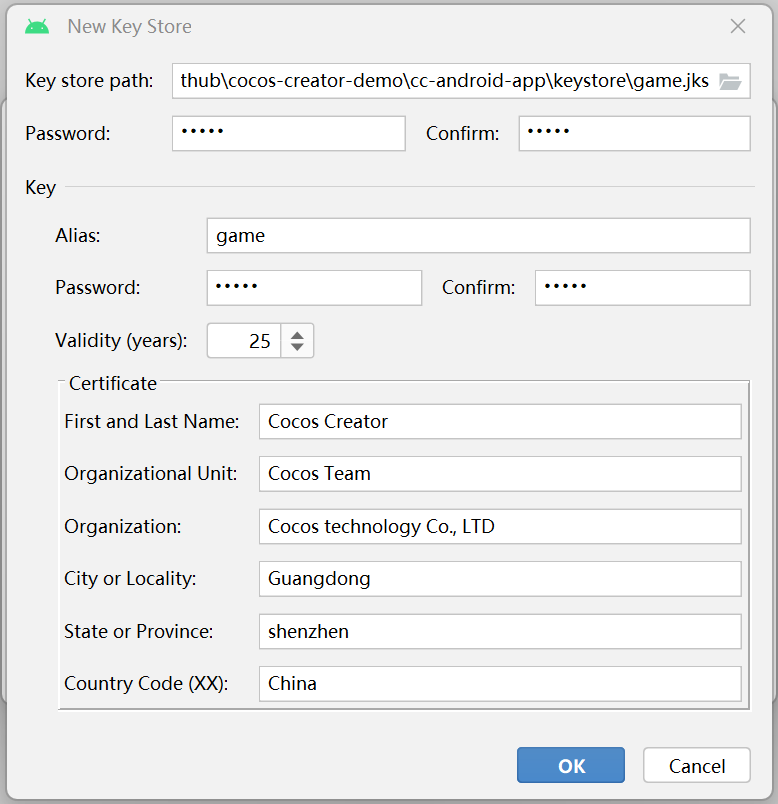
1. 在弹出菜单的窗口中选择 APK，点击 Next 按钮



1. 在引导的窗口中点击 Create new



1. 在弹出的窗口中，填好相应信息即可



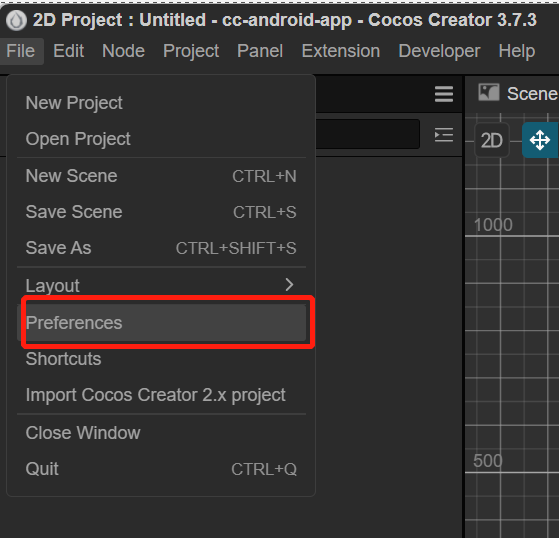
不同的项目，建议使用不同的密钥。可以将每个项目的密钥保存在项目根目录下。

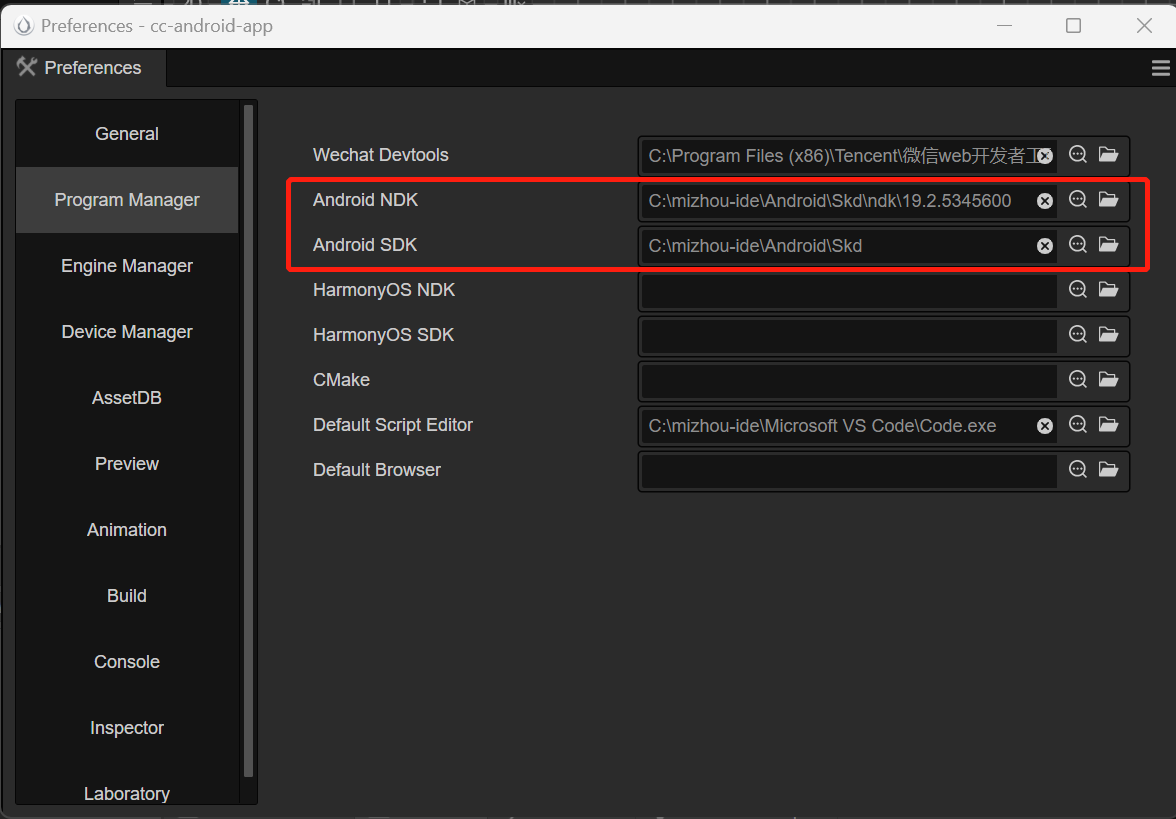
注意：密码要输入英文、数字和特殊字符组合，8位字符以上。

密钥文件及密码等信息要存储好，并且不要泄露给其他人。

## 配置Cocos Creator环境

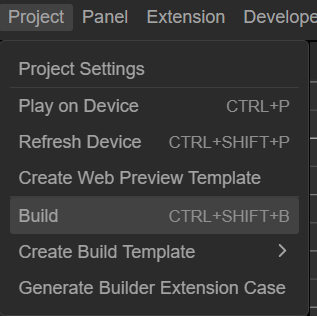
1. 配置android NDK路径



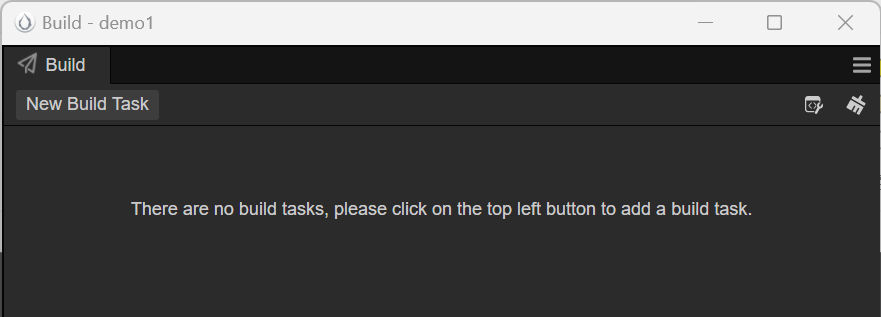


## 创建发布任务

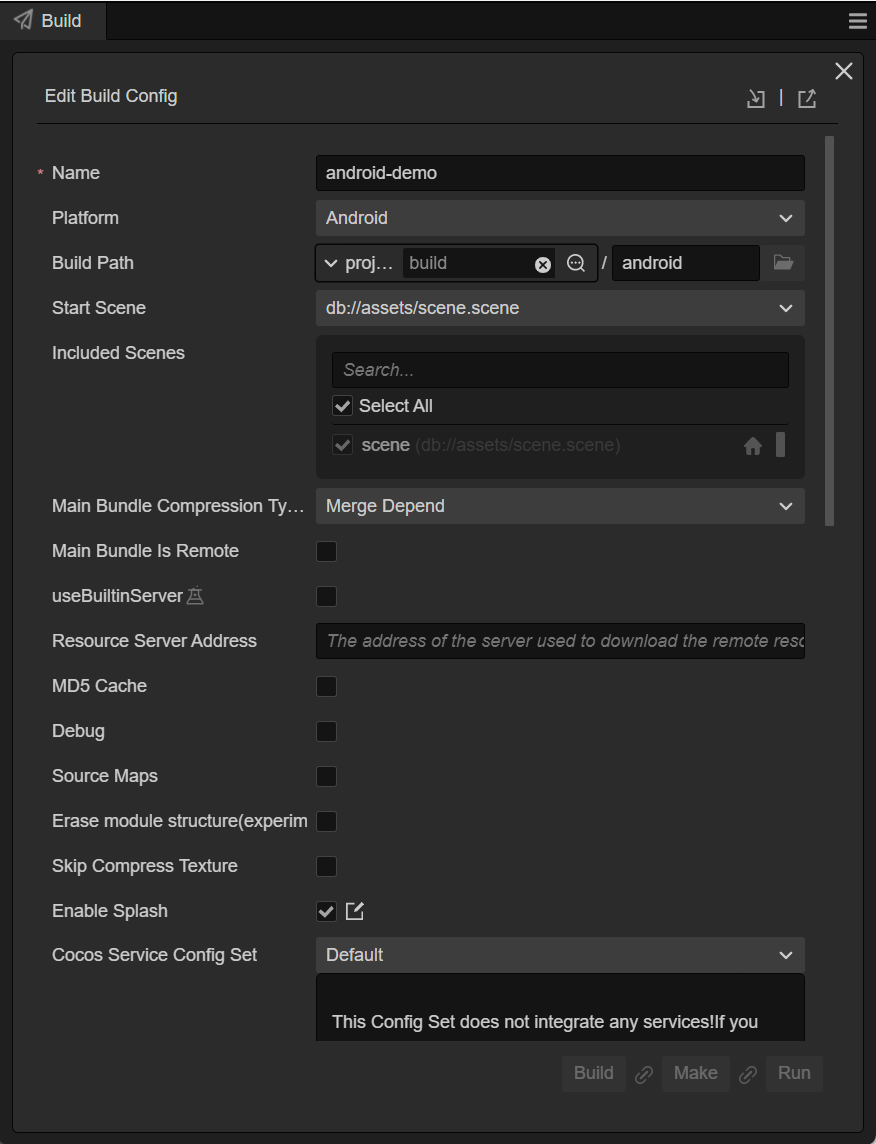
1. 选择 项目 -> 构建 菜单，打开构建面板



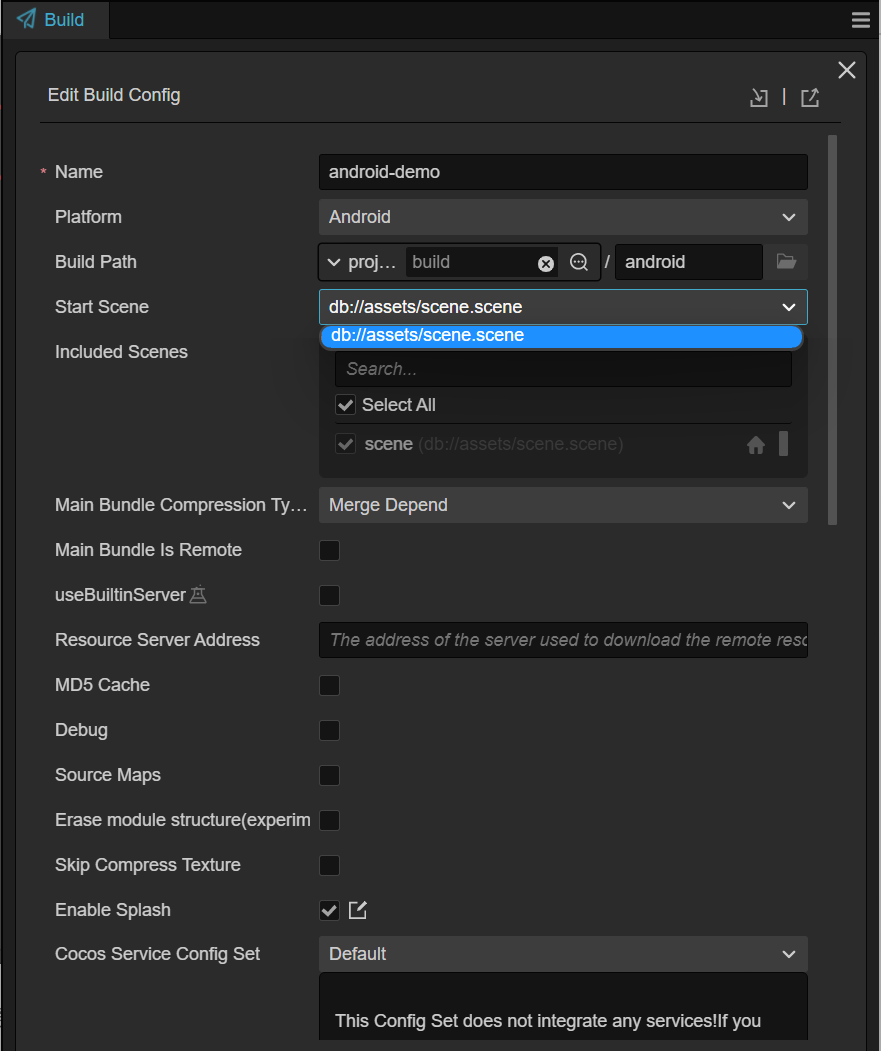
1. 点击面板上的 新建构建任务 选项



1. 选择构建平台为安卓



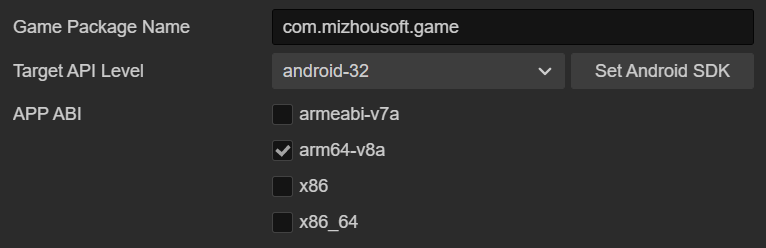
1. 选择至少一个场景作为应用载入的首场景，当仅有一个场景时会被默认添加



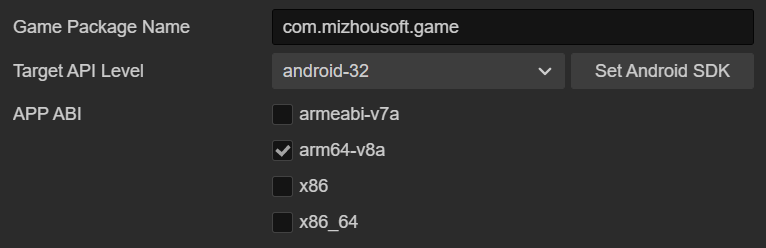
1. 参考 Android 平台构建选项 - 渲染后端 选择渲染后端



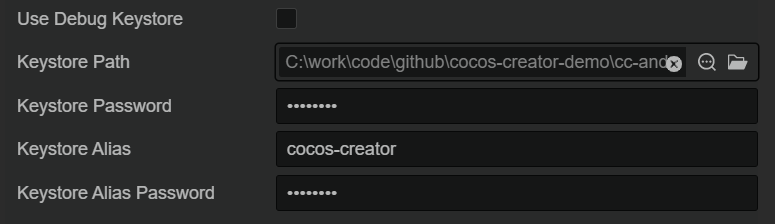
1. 填入包名



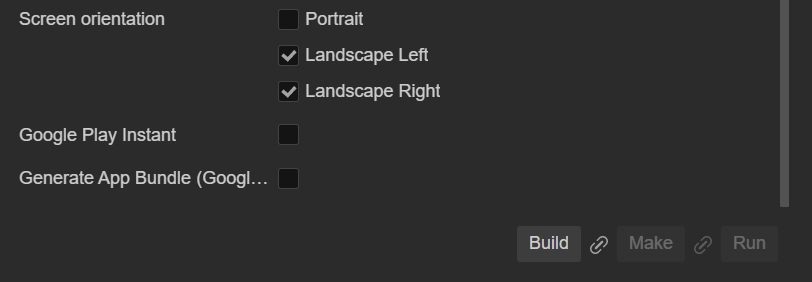
1. 选择 Target API Level为android-32



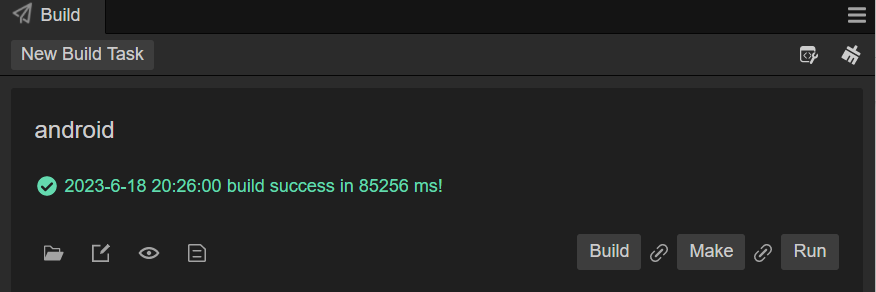
1. 输入密钥信息



1. 点击build，保存及构建



1. 等待项目构建完成



## 编辑android工程

1. 打开项目所在目录，进入 {项目路径}\native\engine\android 目录下，编辑build.gradle文件，替换以下脚本块

buildscript {

repositories {

maven { url 'https://maven.aliyun.com/repository/public' }

maven { url 'https://maven.aliyun.com/repository/google' }

maven { url 'https://maven.aliyun.com/repository/gradle-plugin' }

maven { url 'https://repo.huaweicloud.com/repository/maven' }

maven { url 'https://jitpack.io' }

maven { url 'https://developer.huawei.com/repo/'}

mavenCentral()

// jcenter() // keeped as anchor, will be removed soon

}

dependencies {

classpath 'com.android.tools.build:gradle:4.1.0'

// NOTE: Do not place your application dependencies here; they belong

// in the individual module build.gradle files

}

}

allprojects {

repositories {

maven { url 'https://maven.aliyun.com/repository/public' }

maven { url 'https://maven.aliyun.com/repository/google' }

maven { url 'https://maven.aliyun.com/repository/gradle-plugin' }

maven { url 'https://repo.huaweicloud.com/repository/maven' }

maven { url 'https://jitpack.io' }

maven { url 'https://developer.huawei.com/repo/'}

mavenCentral()

// jcenter() // keeped as anchor, will be removed soon

flatDir {

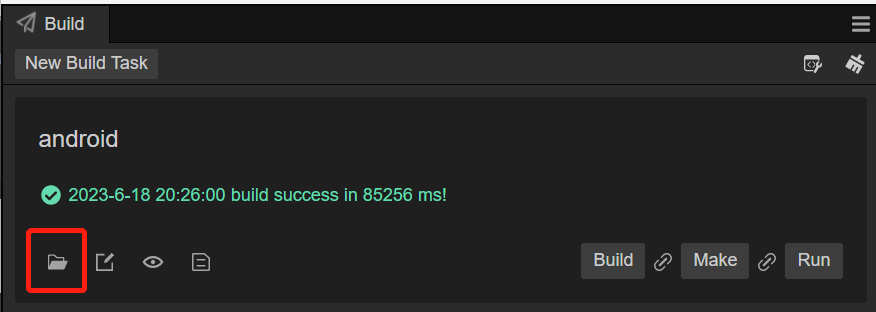
dirs 'libs'

}

}

}

1. 在点击下面的按钮打开生成好的 Android Studio 工程



1. 找到工程对应的目录{项目路径}\build\android



1. 编辑{项目路径}\build\android\proj\build.gradle文件，替换buildscript脚本块

buildscript {

repositories {

maven { url 'https://maven.aliyun.com/repository/public' }

maven { url 'https://maven.aliyun.com/repository/google' }

maven { url 'https://maven.aliyun.com/repository/gradle-plugin' }

maven { url 'https://repo.huaweicloud.com/repository/maven' }

maven { url 'https://jitpack.io' }

maven { url 'https://developer.huawei.com/repo/'}

mavenCentral()

// jcenter() // keeped as anchor, will be removed soon

}

dependencies {

classpath 'com.android.tools.build:gradle:4.1.0'

// NOTE: Do not place your application dependencies here; they belong

// in the individual module build.gradle files

}

}

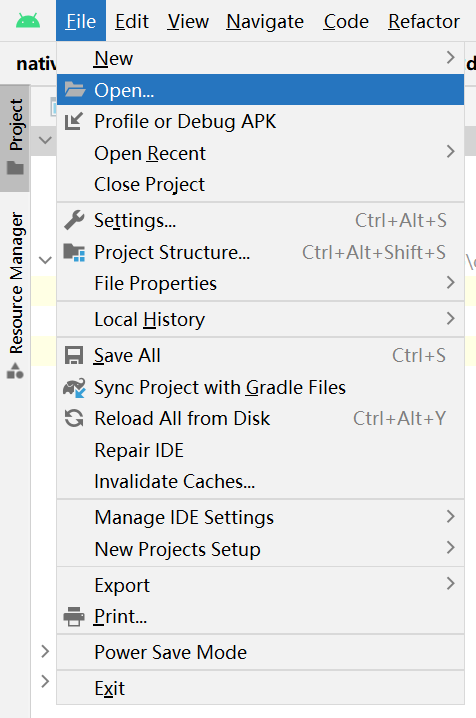
1. 编辑{项目路径}\build\android\proj\gradle.properties，替换构建工具版本号及NDK版本号

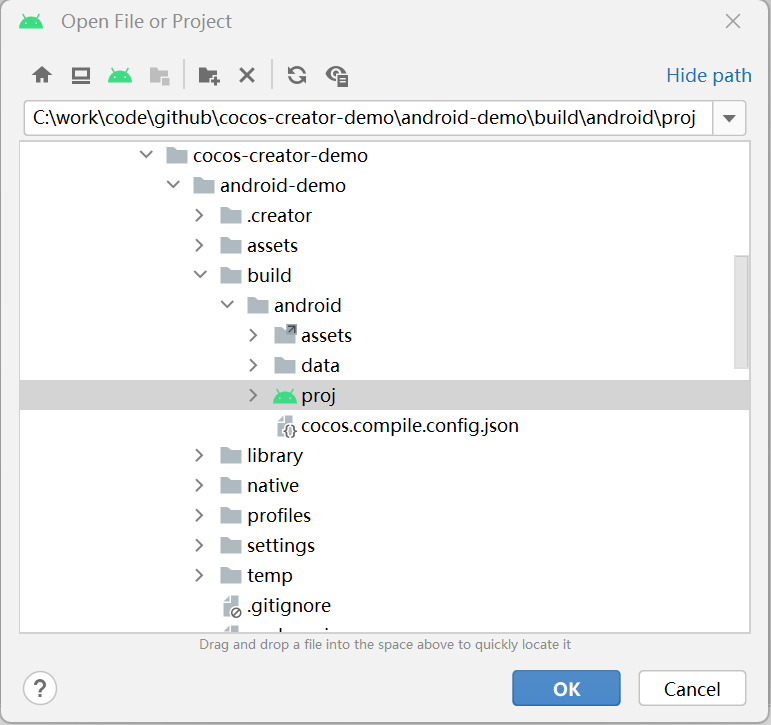
PROP\_BUILD\_TOOLS\_VERSION=29.0.2

PROP\_NDK\_PATH=C:\\mizhou-ide\\Android\\Skd\\ndk\\19.2.5345600

## 构建与发布

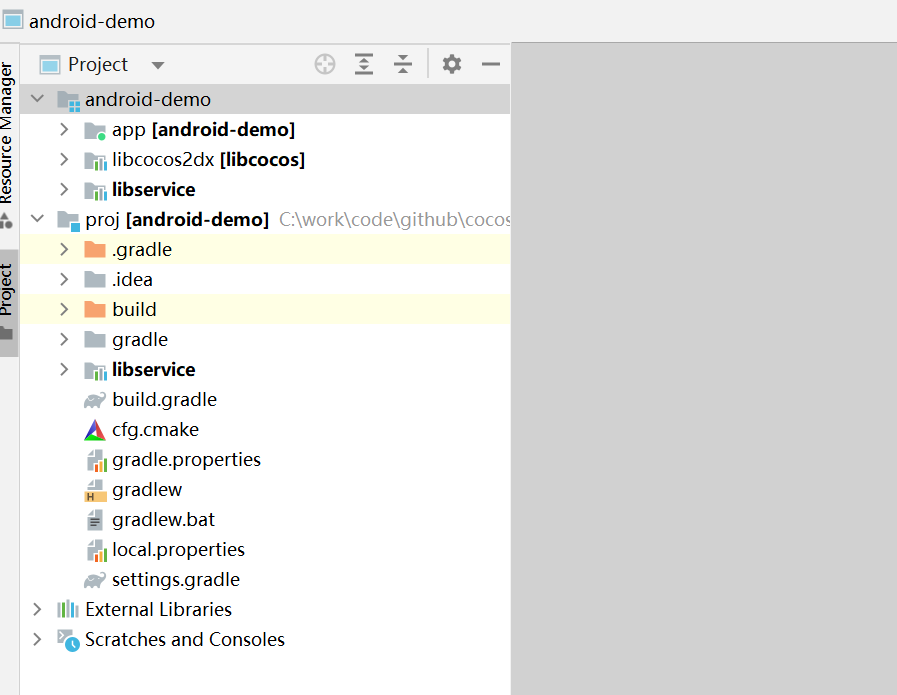
1. 打开 Android Studio，打开已经构建好的项目，{项目路径}/build/android/proj



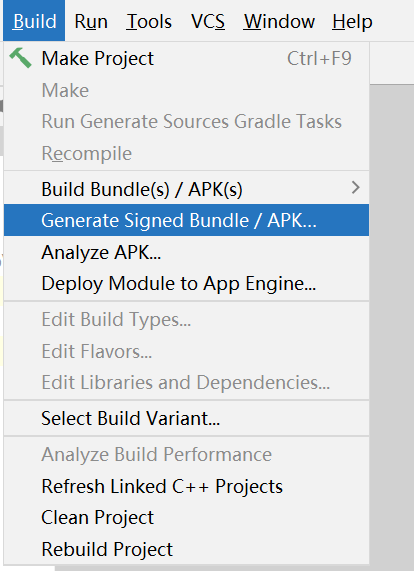


1. 使用 Android Studio 构建 APK

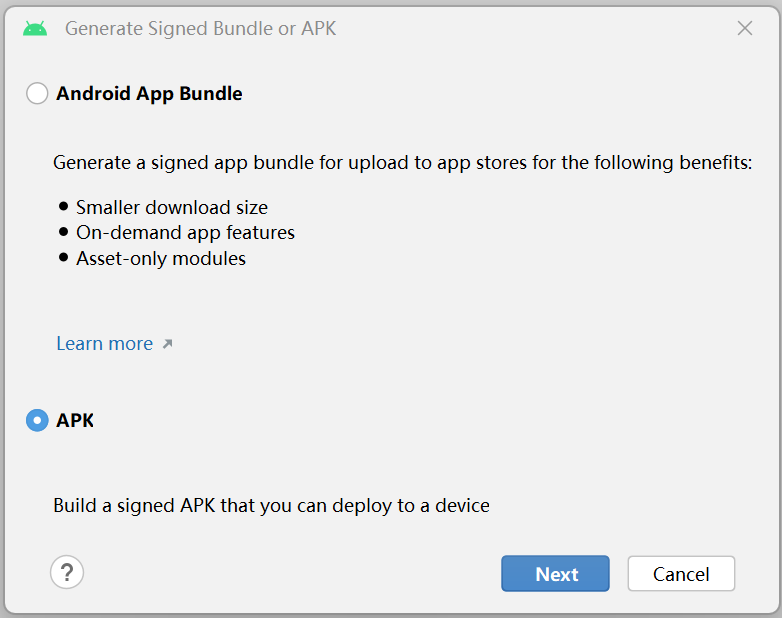
打开 Android Studio 后，会花一段时间进行准备工作，待 Android Studio 将项目准备完成后，即可打包 APK。准备过程可能会耗时较久，如果长时间无响应，请检查网络，或者切换到其他镜像。



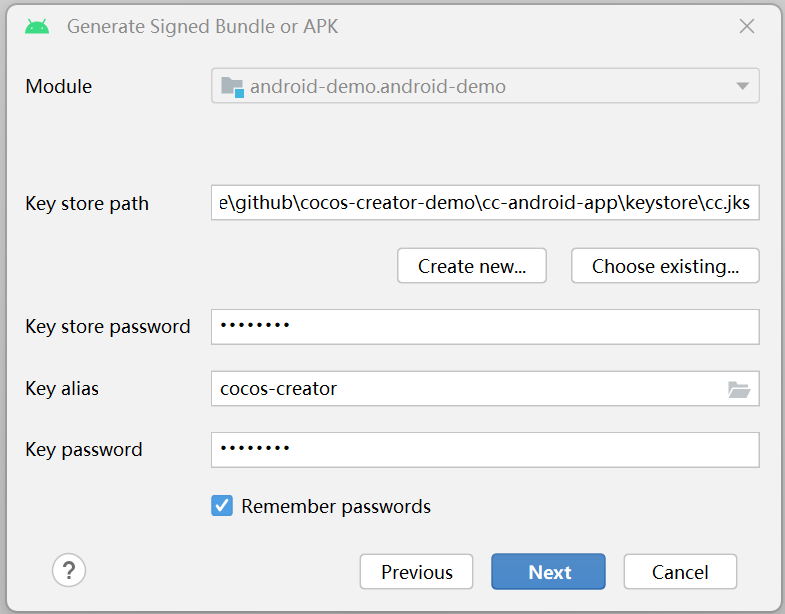
1. 打开 Build 菜单选择 Build Bundle(s) / APK(s)



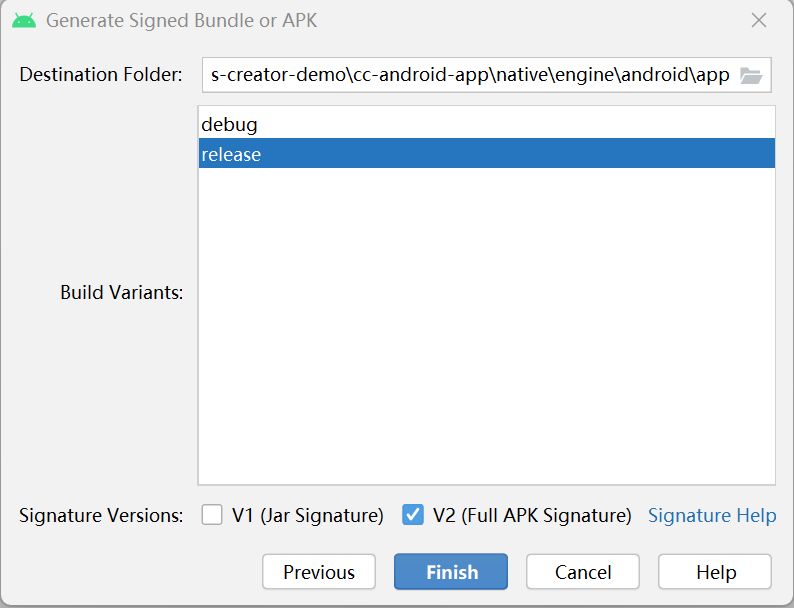
1. 选择APK



1. 选择创建好的密钥



1. 选择release构建，签名版本V2



1. 发布成功后可以在 {项目路径} 目录内找到APK：



# 定制android APP图标

进入 {项目路径}\native\engine\android\res 目录，准备5个不同尺寸ic\_launcher.png图标，覆盖下面5个目录下的文件即可

