

# Overview

## Warning

Android SDK 1.1 is intended to use on Android 8 and newer devices

## Resources

Use Android SDK to develop mobile applications for streaming video and calls.

Download full WCS Android SDK build including examples and API documentation: [Release notes](#)

Read API documentation online: [API docs](#)

Download the source code of the examples: [GitHub](#)

The source code of the examples is located at [GitHub](#) and is used to comment the examples in the present documentation. For instance, this link [line 34](#) refers to the thirty fourth line in the `TwoPlayersActivity.java` class of the 2players example, the revision with the hash of 4ed4c6d77.

To test compiled applications, download the full build bundle with the examples and install the `...-debug.apk` file to your Android device.

## Differences between Android SDK versions

In Android SDK 1.1 WebRTC library `libjingle_peerconnection.jar` is updated to actual version. So Android SDK 1.1 requires Android API 26, i.e. application built with Android SDK 1.1 will run only on Android 8 and higher. Use [Android SDK 1.0](#) only to support previous Android versions.

When publishing in Google Play, [two APKs](#) can be deployed - one for each of the Android SDK versions - for compatibility with devices with API lower and higher than 26.

## Preparing examples for building

If you have some experience in developing Android apps, you can simply download the aar-library and link it to the project manually, then configure building.

Below is how to do this automatically using the `export.sh` script:

## 1. Download the examples

```
git clone -b 1.1 git@github.com:flashphoner/wcs-android-sdk-samples.git
```

## 2. Download the aar library

Download the aar library and put it to the `export` folder

Example (replace `x` by the actual build number):

```
wget http://flashphoner.com/downloads/builds/flashphoner_client/wcs-android-sdk/1.0/wcs-android-sdk-1.1.0.x.aar
cp wcs-android-sdk-1.1.0.x.aar export
```

## 3. Execute the `export.sh` script

Open the `export` folder and execute the `export.sh` script.

The `export.sh` script will prepare configs for further building. The result is placed into the `output` folder.

### Warning

This step is very important because application examples source code is the same for different versions of Android SDK. `export.sh` script automatically sets minimal required Android API version for building examples depending on Android SDK version

```
cd export
./export.sh wcs-android-sdk-1.1.0.x.aar
```

## 4. Edit the `local.properties` file

Edit the `local.properties` file and specify paths to Android SDK and NDK.

Linux environment example:

```
ndk.dir=/opt/android-ndk-r12b
sdk.dir=/opt/android-sdk-linux
```

## Building examples with Gradle

Prepare examples for building, then go to the `output` folder and start building:

```
cd output
gradle build
```

Since Android SDK build [1.1.0.55](#), it is necessary to use Gradle 7 and above, or build examples with Gradle wrapper

```
cd output
gradlew build
```

Also, JDK 11 should be installed.

## Building examples in Android Studio

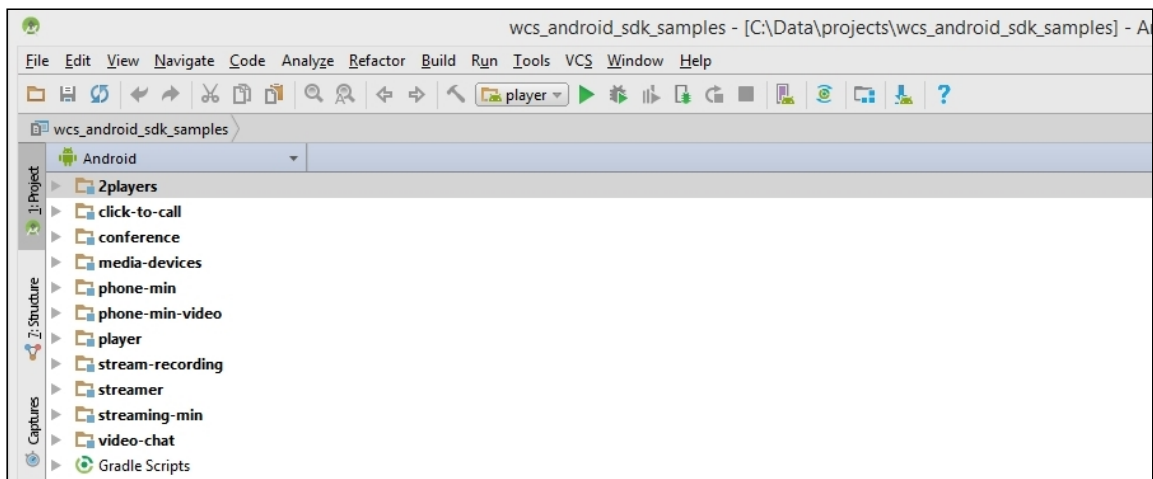
Prepare examples for building, then rename the `output` folder to `wcs_android_sdk_samples`, for instance.

### 1. Install the necessary programs

- [Android Studio with Android SDK](#)
- [Android NDK](#)

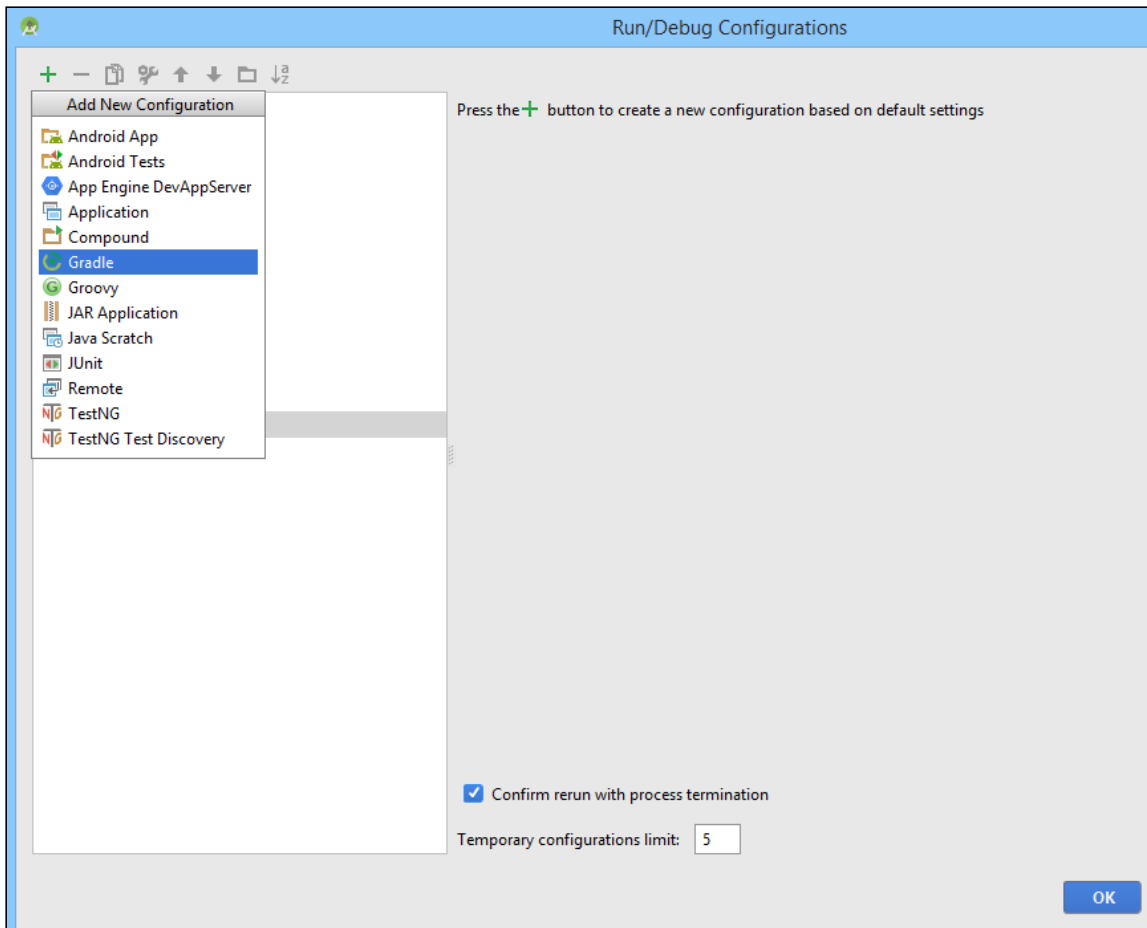
### 2. Open the project in Android Studio

Open the project from the `output` folder in Android Studio



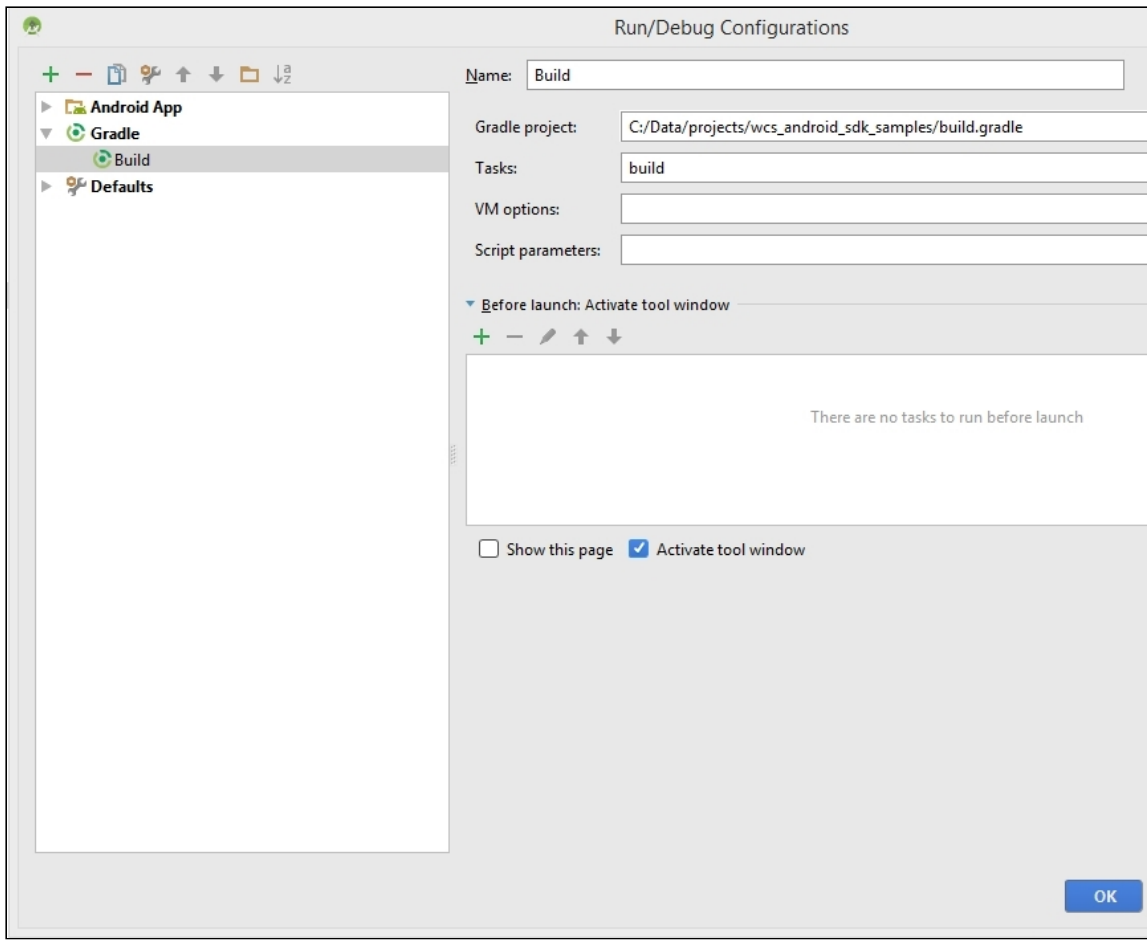
### 3. Add Gradle run configuration

Add Gradle run configuration to the `Run / Debug Configurations` menu

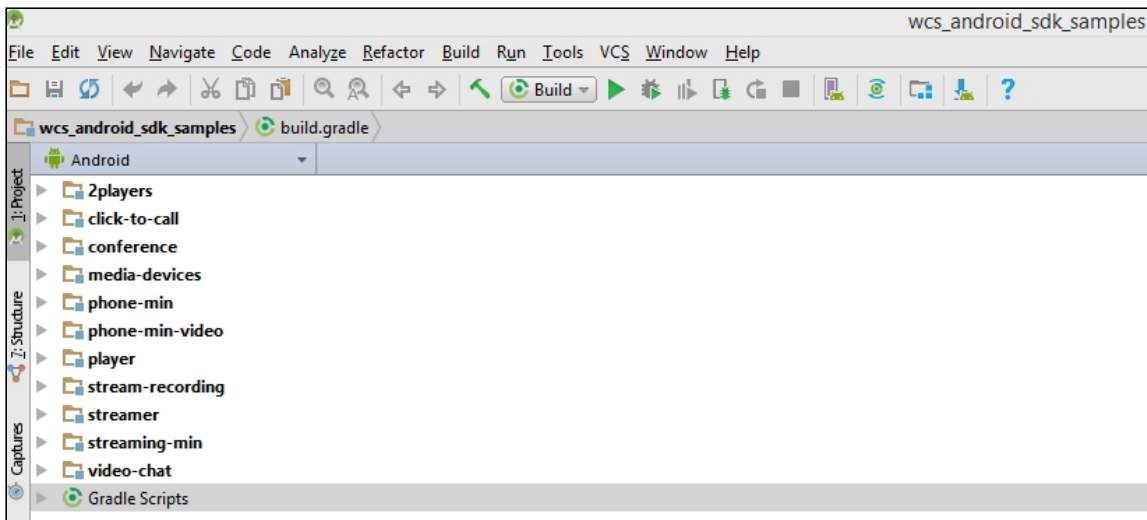


#### 4. Configure the build

In the run configuration specify the `build.gradle` file, the name of the build and set the purpose to `build`



## 5. Run building of examples



The building result is `.apk` files located in the corresponding folders: `2players/build`, `click-to-call/build`, and so on. The Android SDK file is located in the following path in the project: `libs/wcs-android-sdk-1.1.0.x.aar`

## Building a separate example

If you cannot run export script for all the examples, or wish to build a separate example, it can be prepared for building as follows:

### 1. Download examples source code

```
git clone -b 1.1 https://github.com/flashphoner/wcs-android-sdk-samples.git
```

### 2. Copy an example needed to a separate folder

```
cd wcs-android-sdk-samples
cp streaming-min gradle.properties ~/streaming-min
```

### 3. Download aar library and put it to `libs` subfolder

Download aar library and put it to `libs` subfolder in the example folder

```
wget https://flashphoner.com/downloads/builds/flashphoner_client/wcs-android-
sdk/1.1/wcs-android-sdk-1.1.0.x.aar
mkdir ~/streaming-min/libs
cp wcs-android-sdk-1.1.0.x.aar ~/streaming-min/libs
```

### 4. Add the `buildscript` section to the `build.gradle` file

Add the `buildscript` section to the beginning of `build.gradle` file in the example folder:

```
buildscript {
    repositories {
        jcenter()
        mavenCentral()
        maven { url 'https://maven.google.com' }
        google()
    }
    dependencies {
        classpath 'com.android.tools.build:gradle:7.0.0'
        classpath 'com.github.triplet.gradle:play-publisher:1.1.5'
    }
}
```

### 5. Add the `repositories` section to the `build.gradle` file

Add the `repositories` section to the `build.gradle` file in the example folder:

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

```
mavenCentral()
maven { url 'https://maven.google.com' }
google()
flatDir{
    dirs 'libs'
}
}
```

## 6. Replace the string in `dependencies` section of the `build.gradle` file

Replace the string in `dependencies` section of the `build.gradle` file in the example folder

```
implementation project(':fp_wcs_api')
```

to

```
implementation 'com.flashphoner.fpwcsapi:wcs-android-sdk-1.1.0.x@aar'
```

where `wcs-android-sdk-1.1.0.x` is the aar file name downloaded at step 3.

## 7. The `build.gradle` file example

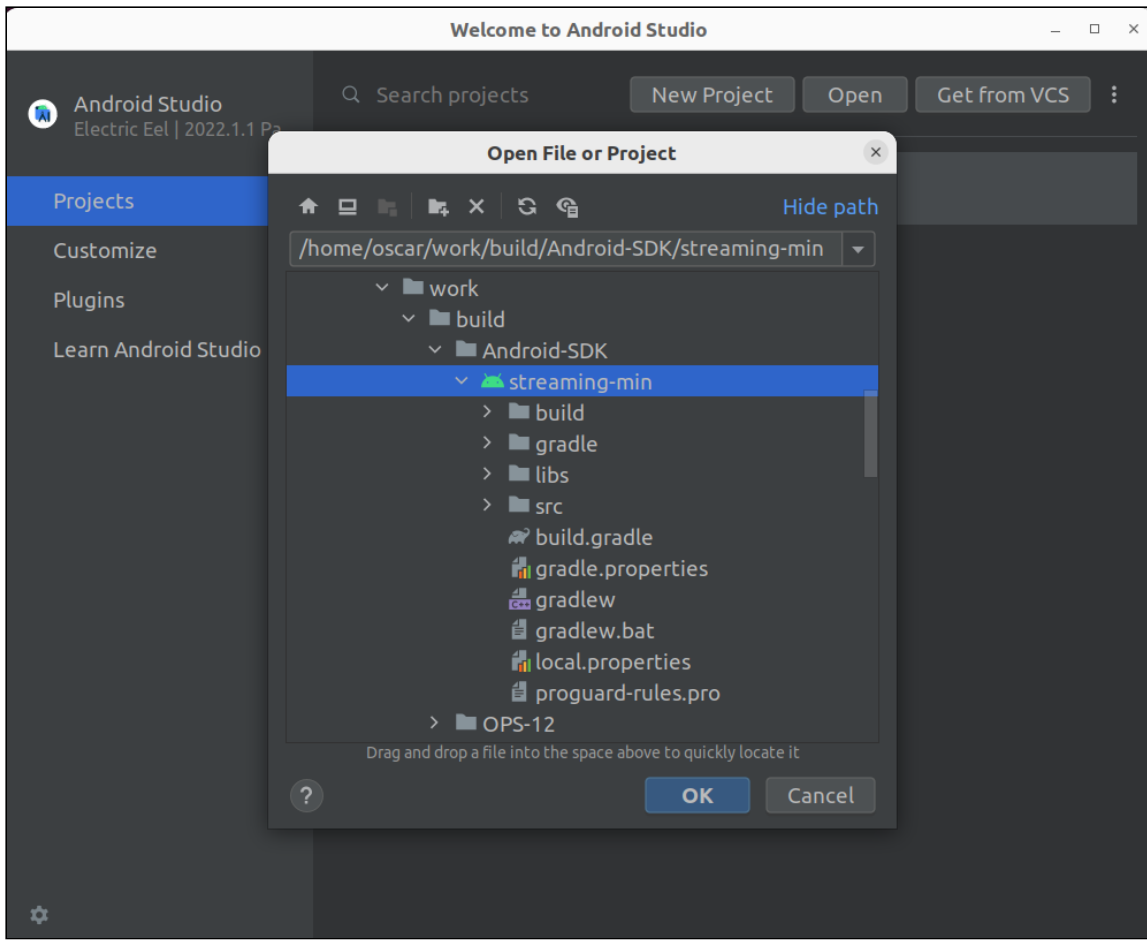
build.gradle Expand source

 **build.gradle**



## 8. Building the example in Android Studio

Open the example folder in Android Studio to build the example. Android Studio will do Gradle sync and install gradle version needed



## 9. Building the example with Gradle

To build the example with Gradle install Gradle wrapper and run

```
gradlew build
```

## Known issues

1. It is impossible now to set microphone gain in Android SDK while publishing stream.