

Описание

- [Ресурсы](#)
- [Различия между версиями Android SDK](#)
- [Подготовка примеров для сборки](#)
- [Сборка примеров с помощью Gradle](#)
- [Сборка примеров в Android Studio](#)
- [Сборка отдельного примера](#)
- [Известные ограничения](#)

Ресурсы

Для разработки мобильных приложений потокового видео и звонков используется Android SDK.

Скачать полную сборку WCS Android SDK с примерами и API документацией
<https://docs.flashphoner.com/display/ANDROIDSDK11RU/Android+SDK+release+notes>

Смотреть API документацию онлайн
<http://flashphoner.com/docs/api/WCS5/client/android-sdk/1.1/latest>

Скачать исходный код примеров
<https://github.com/flashphoner/wcs-android-sdk-samples>

Исходный код примеров расположен на [Github](#) и будет использоваться в дальнейшем для пояснения работы примеров в данной документации. Например, данная ссылка [line 34](#) указывает на тридцать четвертую строку класса TwoPlayersActivity.java примера 2players ревизии с хешем 4ed4c6d77.

Для тестирования готовых приложений, скачайте полную сборку с примерами и установите .apk файл на ваше Android-устройство.

Различия между версиями Android SDK

В версии Android SDK 1.1 обновлена WebRTC библиотеки `libjingle_peerconnection.jar`. В связи с этим, минимальная версия Android API, поддерживаемая Android SDK 1.1, увеличена до 26, т.е. приложения на базе Android SDK 1.1 будут работать в Android 8 и выше. Для поддержки предшествующих версий Android используйте [Android SDK 1.0](#).

При публикации в Google Play для совместимости приложения с устройствами с API ниже и выше 26 можно выгрузить [две APK](#) - под каждую версию Android SDK.

Подготовка примеров для сборки

Если у вас есть опыт разработки под Android, вы можете просто скачать aar-библиотеку и вручную присоединить ее к проекту и сконфигурировать для сборки.

Ниже показано как это сделать автоматически с помощью скрипта `export.sh`:

1. Скачайте примеры

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

2. Скачайте aar библиотеку и положите ее в папку 'export'

Пример:

```
wget https://flashphoner.com/downloads/builds/flashphoner_client/wcs-android-sdk/1.1/wcs-android-sdk-1.1.0.x.aar
cp wcs-android-sdk-1.1.0.x.aar export
```

3. Перейдите в папку 'export' и запустите скрипт `export.sh`, который подготовит конфиги для дальнейшей сборки. Результат будет находиться в папке 'output'.

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



Данный шаг очень важен, поскольку исходные тексты примеров одинаковы для Android SDK 1.0 и 1.1. Скрипт `export.sh` автоматически, в зависимости от версии aar библиотеки, устанавливает для сборки примеров минимальную версию Android API.

4. Скорректируйте файл `local.properties` и пропишите пути к Android SDK и NDK

Пример:

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

Сборка примеров с помощью Gradle

Подготовьте примеры для сборки, перейдите в папку `output` и запустите сборку:

```
cd output
gradle build
```

Начиная со сборки Android SDK [1.1.0.55](#), необходимо использовать Gradle 7 и выше, либо запускать сборку с помощью Gradle wrapper

```
cd output
gradlew build
```

Также для сборки необходимо наличие JDK 11.

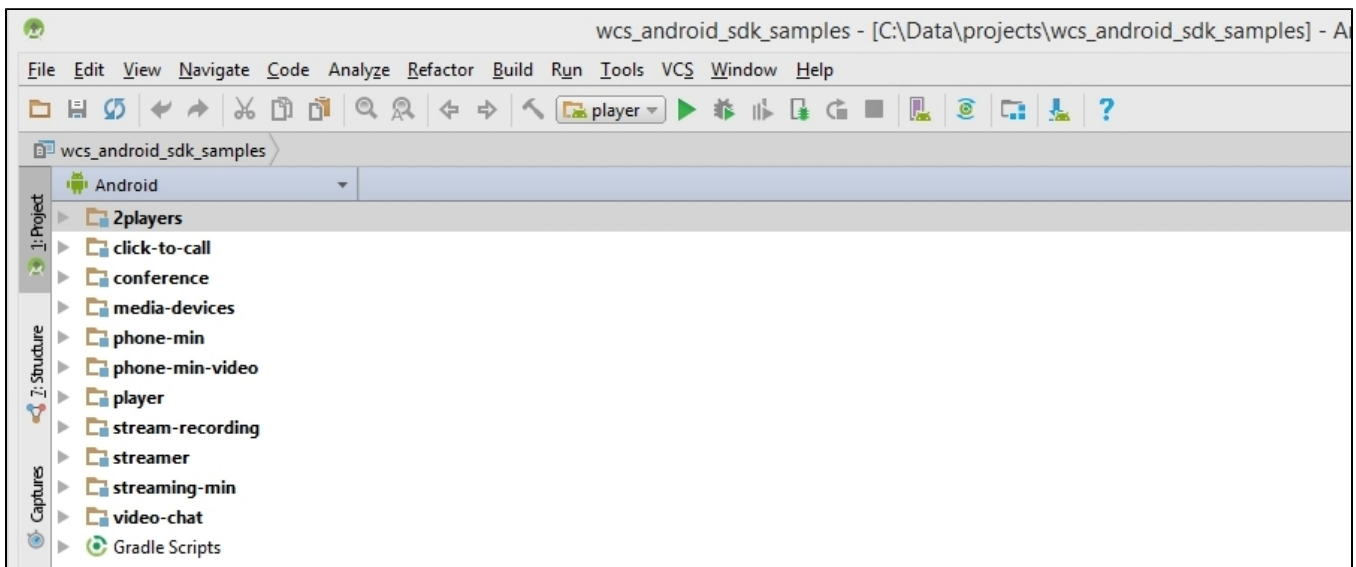
Сборка примеров в Android Studio

Подготовьте примеры для сборки в папку `output` и переименуйте ее, например, в `'wcs_android_sdk_samples'`

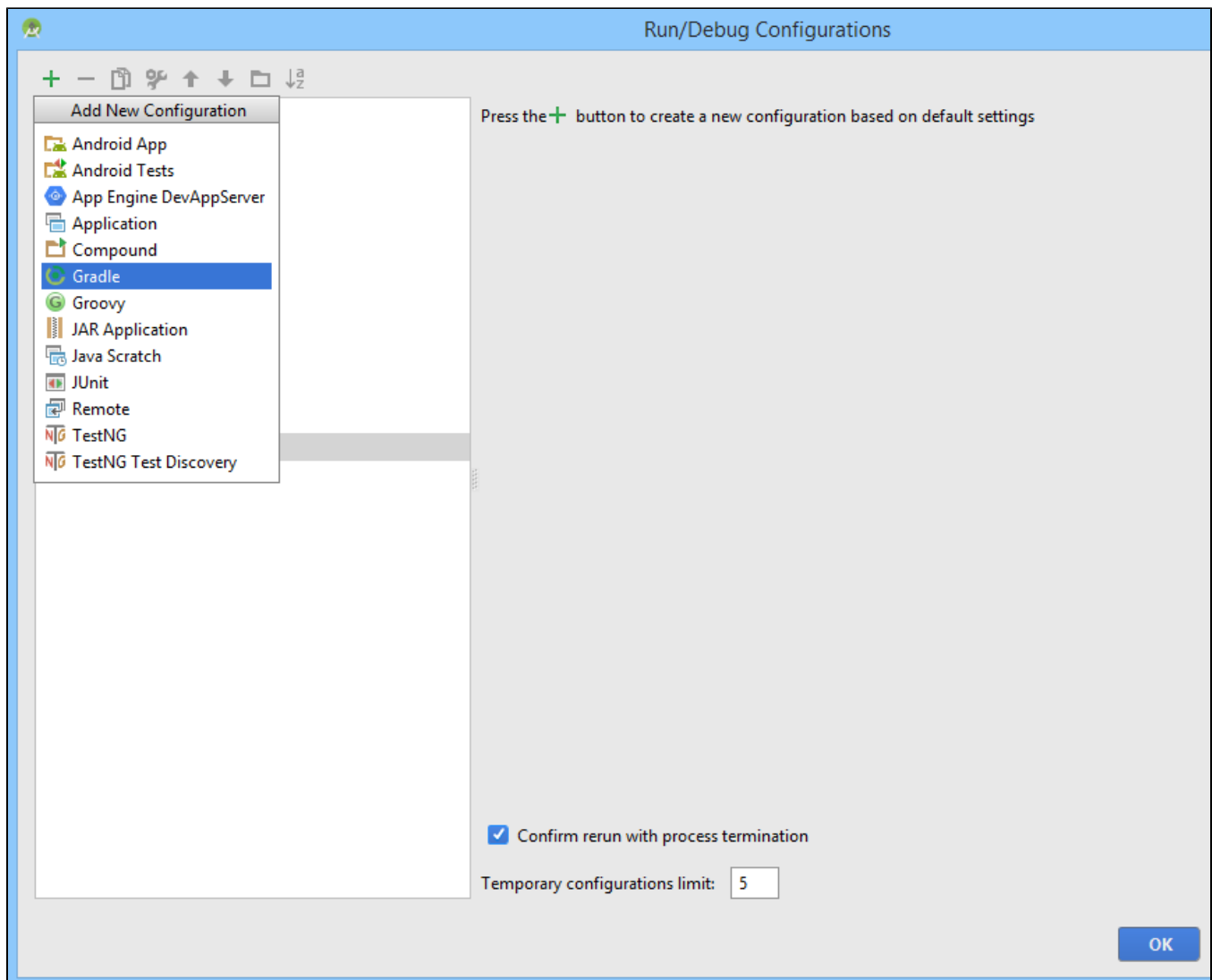
1. Установите следующие программы:

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

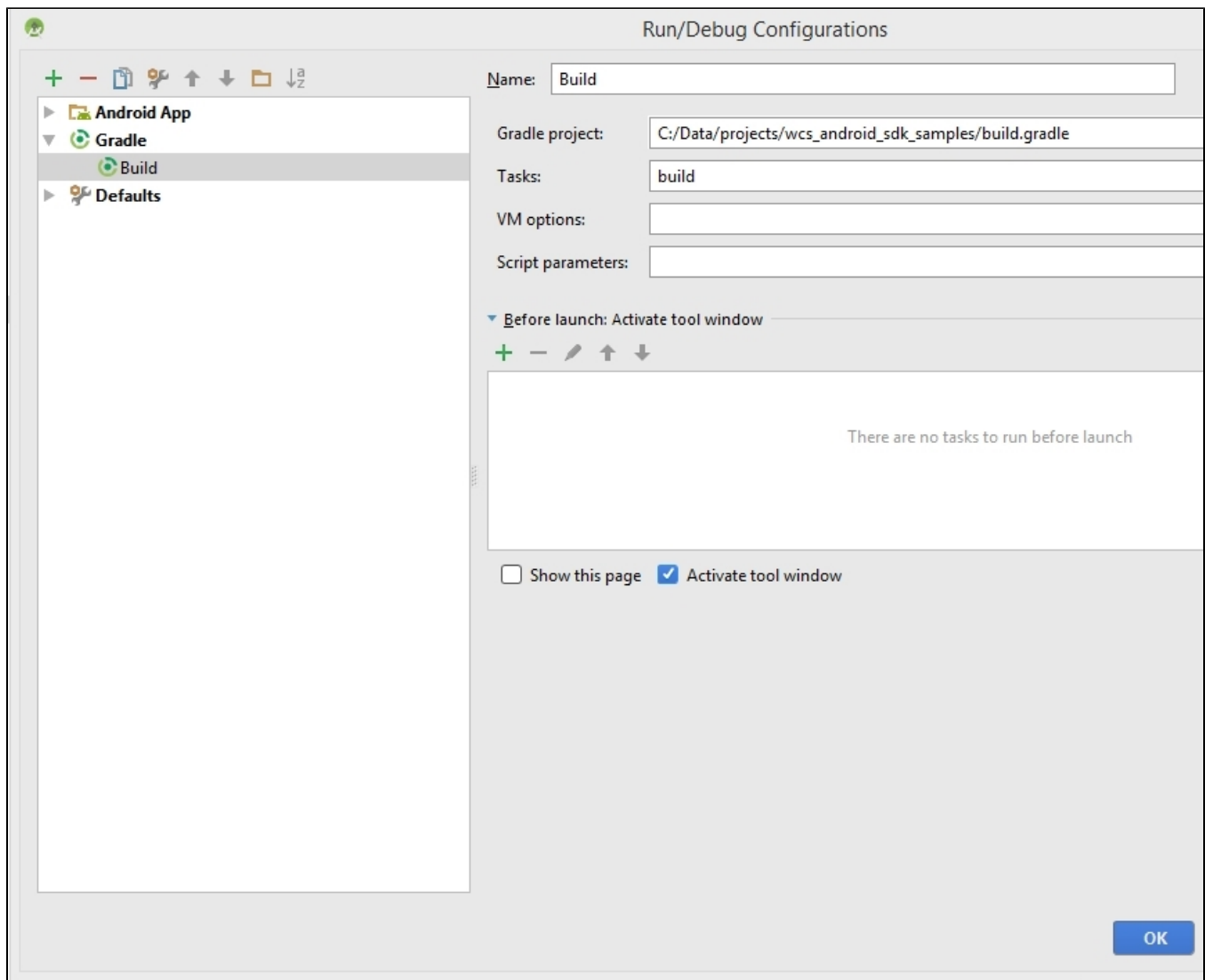
2. Откройте проект из папки `output` в Android Studio



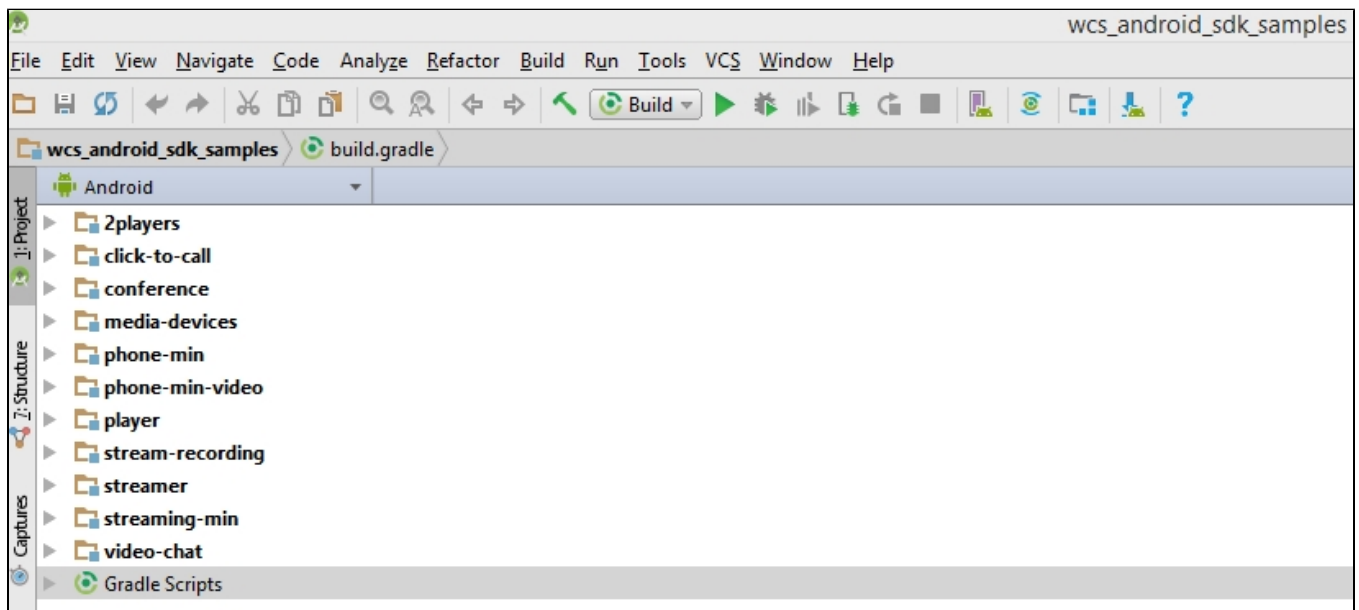
3. Добавьте конфигурацию запуска Gradle в меню `Run / Debug Configurations`



4. Для конфигурации запуска укажите файл сборки build.gradle, название сборки и назначение: build



5. Запустите сборку примеров.



Результатом сборки будут .apk файлы в соответствующих директориях 2players/build, click-to-call/build, и т.д. Файл Android SDK находится в проекте по следующему пути: libs/wcs-android-sdk-1.0.1.2.aar

Сборка отдельного примера

Если необходимо собрать отдельный пример, или нет возможности выполнить скрипт экспорта всех примеров, пример для сборки может быть подготовлен вручную следующим образом:

1. Скачайте примеры

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

2. Скопируйте нужный пример в отдельный каталог

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

3. Скачайте aar библиотеку и положите ее в папку libs в примере

```
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. Добавьте в начало файла build.gradle в каталоге примера секцию buildscript :

```
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. Добавьте в файл build.gradle в каталоге примера секцию repositories :

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

6. Замените в секции dependencies файла build.gradle в каталоге примера строку

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

на

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

где wcs-android-sdk-1.1.0.x - имя aar файла, скачанного на шаге 3.

7. Пример файла build.gradle

build.gradle

```
apply plugin: 'com.android.application'
apply plugin: 'com.github.triplet.play'

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'
    }
}

Properties signingProps = null

android {
    compileSdkVersion 31
    buildToolsVersion '31.0.0'
    defaultConfig {
        applicationId "com.flashphoner.wcsexample.two_way_streaming"
        minSdkVersion 26
        targetSdkVersion 31
        versionCode 2600001
        versionName "1.1"
    }
    def signingPropsFile = System.getenv('SIGNING_PROPERTIES')
    if (signingPropsFile && new File(signingPropsFile).exists()) {
        signingProps = new Properties()
        signingProps.load(new FileInputStream(file(signingPropsFile)))

        signingConfigs {
            release {
                storeFile file(signingProps['keystore'])
                storePassword signingProps['keystore.password']
                keyAlias signingProps['key.alias']
                keyPassword signingProps['key.password']
            }
        }
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'

            if (signingProps) {
                signingConfig signingConfigs.release
            }
        }
    }
    compileOptions {
        sourceCompatibility JavaVersion.VERSION_1_8
        targetCompatibility JavaVersion.VERSION_1_8
    }
}

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

```

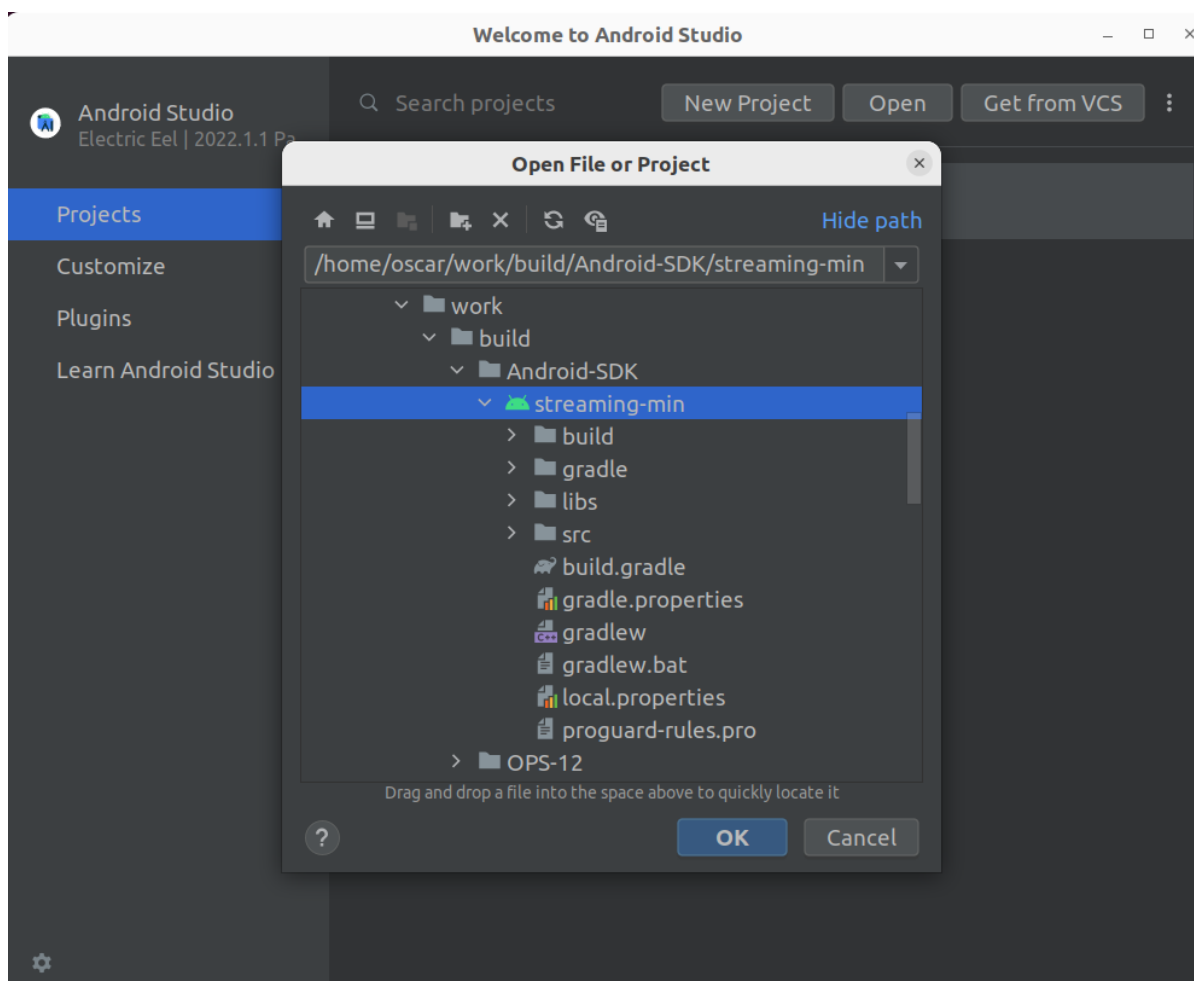
        dirs 'libs'
    }
}

dependencies {
    implementation 'com.flashphoner.fpwcsapi:wcs-android-sdk-1.1.0.x@aar'
    implementation fileTree(include: ['*.jar'], dir: 'libs')
    testImplementation 'junit:junit:4.12'
    implementation 'androidx.appcompat:appcompat:1.4.0'
    implementation 'com.google.android.material:material:1.4.0'
}

play {
    if(signingProps) {
        track = 'production'
        serviceAccountEmail = signingProps['service.account.email']
        pk12File = file(signingProps['service.account.pk12file'])
    }
}
}

```

8. Для сборки в Android Studio откройте каталог примера. Android Studio выполнит Gradle sync и установит необходимую версию gradle



9. Для сборки с помощью Gradle установите Gradle wrapper и выполните команду

```
gradlew build
```

Известные ограничения

1. Невозможно управлять уровнем громкости микрофона при публикации потока в Android SDK.