

Xilinx Zynq FPGA, TI DSP, MCU Programming 및 Circuit Design Expert 과정

강사 - Innova Lee(이상훈)
gcccompil3r@gmail.com

Android NDK

Dev Env Setting for Linux

Android Development Environment Configuration

과거의 복잡함

1. <http://developer.android.com/sdk/>
2. <http://www.eclipse.org/downloads/>
3. <http://developer.android.com/sdk/installing/installing-adt.html>
4. <http://developer.android.com/sdk/index.html>

위 링크의 모든 것을 다운받고 설정을 맞춰줘야 하는 복잡함이 존재했음

현재

옛 부분에 Windows 에서 설치하는 과정이 적혀 있는데 그냥 편하게 리눅스에서 하는 것을 권장한다.

현재 Vivado HLS, Code Composer Studio, Linux Kernel(DSP) 등의 모든 개발 환경이 리눅스에 최적화되어 있을 수 밖에 없다.

우리가 Android Programming 을 하는 이유는 오직 하나 휴대폰 조종기 때문이다.

별 것 없으므로 작업을 최대한 빨리 끝내도록 한다.

설치하는 과정도 리눅스가 훨씬 간편하다.

우선 안드로이드 스튜디오를 다운 받기 위해 아래 링크에 들어간다.

<https://developer.android.com/studio/index.html>



로 돌아가기

이

이드

|

Android Studio

Android용 공식 IDE

Android Studio는 모든 유형의 Android 기기
가장 빠른 도구를 제공합니다.

최고 수준의 코드 편집, 디버깅, 성능 도구, 유
턴트 빌드/배포 시스템을 사용하여 뛰어난 품질
집중할 수 있습니다.

다운로드 **ANDROID STUDIO**
2.3.2 FOR LINUX (438 MB)

Android Studio 다운로드

다운로드하기 전에, 다음 사용 약관에 동의해야 합니다.

사용 약관

Android 소프트웨어 개발 키트 라이선스 계약

1. 소개

1.1 Android 소프트웨어 개발 키트(라이선스 계약에서는 'SDK'라고 하며 특히 Android 시스템 파일, 패키지 API 및 Google API 부가기능을 포함)의 사용은 SDK 라이선스 계약의 약관에 따라 귀하에게 라이선스를 부여합니다. 라이선스 계약은 SDK 사용과 관련하여 귀하와 Google 간에 체결하는 법적 구속력을 가집니다.

1.2 'Android'는 Android 오픈소스 프로젝트(<http://source.android.com/>에서 확인할 수 있음)에 따라 사용할 수 있도록 만들어진, 기기를 대상으로 하는 Android 소프트웨어 스택을 의미하며 수시로 업데이트될 수 있습니다.

1.3 '호환 가능한 구현'은 (i) Android 호환성 웹사이트(<http://source.android.com/compatibility>)에서 확인할 수 있으며 수시로 업데이트될 수 있습니다.

☐ 본인은 상기 사용 약관을 읽었으며 이에 동의합니다.

ANDROID STUDIO FOR LINUX 다운로드

터미널 창에 아래 명령어 입력

```
PROCESSING TRIGGERS FOR libc-bin (2.25-0ubuntu1) ...  
kwrg@kwrg-Samsung-DeskTop-System:~/gproj/RCBattleShip$ sudo apt-get install  
libz1:i386 libncurses5:i386 libbz2-1.0:i386 libstdc++6:i386  
more (8) Role-playing Games
```

적당한 디렉토리 잡고 압축 해제

```
kwrg@kwrg-Samsung-DeskTop-System:~/and$ ls  
android-studio  android-studio-ide-162.3934792-linux.zip  
kwrg@kwrg-Samsung-DeskTop-System:~/and$ cd android-studio/  
kwrg@kwrg-Samsung-DeskTop-System:~/and/android-studio$ ls  
Install-Linux-tar.txt  NOTICE.txt  build.txt  jre  license  
LICENSE.txt           bin          gradle     lib  plugins  
kwrg@kwrg-Samsung-DeskTop-System:~/and/android-studio$
```

위치를 /usr/local 로 변경

```
kwrg@kwrg-Samsung-DeskTop-System:~/and$ ls
android-studio  android-studio-ide-162.3934792-linux.zip
kwrg@kwrg-Samsung-DeskTop-System:~/and$ sudo cp -r android-studio /usr/local
/
kwrg@kwrg-Samsung-DeskTop-System:~/and$
```

이제 추가적인 SW 들을 설치하도록 한다.

```
kwrg@kwrg-Samsung-DeskTop-System:/usr/local/android-studio/bin$ sudo apt-get
```

```
install python-software-properties
```

패키지 목록을 읽는 중입니다... 완료

의존성 트리를 만드는 중입니다

상태 정보를 읽는 중입니다... 완료

다음 패키지가 자동으로 설치되었지만 더 이상 필요하지 않습니다:

```
linux-headers-4.4.0-64 linux-headers-4.4.0-64-generic
```

```
linux-headers-4.4.0-66 linux-headers-4.4.0-66-generic
```

```
linux-image-4.4.0-64-generic linux-image-4.4.0-66-generic
```

```
linux-image-extra-4.4.0-64-generic linux-image-extra-4.4.0-66-generic
```

```
linux-signed-image-4.4.0-64-generic linux-signed-image-4.4.0-66-generic
```

Use 'sudo apt autoremove' to remove them.

The following additional packages will be installed:

```
python-apt python-pycurl
```

제안하는 패키지:

```
python-apt-dbg python-apt-doc libcurl4-gnutls-dev python-pycurl-dbg
```

```
python-pycurl-doc
```

다음 새 패키지를 설치할 것입니다:

```
python-apt python-pycurl python-software-properties
```

0개 업그레이드, 3개 새로 설치, 0개 제거 및 148개 업그레이드 안 함.

203 k바이트 아카이브를 받아야 합니다.

이 작업 후 930 k바이트의 디스크 공간을 더 사용하게 됩니다.

계속 하시겠습니까? [Y/n]


```
kwrg@kwrg-Samsung-DeskTop-System:/usr/local/android-studio/bin$ sudo add-apt-  
-repository ppa:webupd8team/java
```

Oracle Java (JDK) Installer (automatically downloads and installs Oracle JDK7 / JDK8 / JDK9). There are no actual Java files in this PPA.

More info (and Ubuntu installation instructions):

- for Oracle Java 7: <http://www.webupd8.org/2012/01/install-oracle-java-jdk-7-in-ubuntu-via.html>
- for Oracle Java 8: <http://www.webupd8.org/2012/09/install-oracle-java-8-in-ubuntu-via-ppa.html>

Debian installation instructions:

- Oracle Java 7: <http://www.webupd8.org/2012/06/how-to-install-oracle-java-7-in-debian.html>
- Oracle Java 8: <http://www.webupd8.org/2014/03/how-to-install-oracle-java-8-in-debian.html>

Oracle Java 9 (for both Ubuntu and Debian): <http://www.webupd8.org/2015/02/install-oracle-java-9-in-ubuntu-linux.html>

```
kwrg@kwrg-Samsung-DeskTop-System:/usr/local/android-studio/bin$ sudo apt-get
update
기존:1 http://kr.archive.ubuntu.com/ubuntu xenial InRelease
기존:2 http://kr.archive.ubuntu.com/ubuntu xenial-updates InRelease
기존:3 http://kr.archive.ubuntu.com/ubuntu xenial-backports InRelease
받기:4 http://security.ubuntu.com/ubuntu xenial-security InRelease [102 kB]
기존:5 http://ppa.launchpad.net/mc3man/gstffmpeg-keep/ubuntu xenial InReleas
e
받기:6 http://ppa.launchpad.net/webupd8team/java/ubuntu xenial InRelease [17
.6 kB]
받기:7 http://ppa.launchpad.net/webupd8team/java/ubuntu xenial/main amd64 Pa
ckages [2,892 B]
받기:8 http://ppa.launchpad.net/webupd8team/java/ubuntu xenial/main i386 Pac
kages [2,892 B]
받기:9 http://ppa.launchpad.net/webupd8team/java/ubuntu xenial/main Translat
ion-en [1,260 B]
내려받기 127 k바이트, 소요시간 2초 (60.1 k바이트/초)
패키지 목록을 읽는 중입니다... 완료
```

```
kwrg@kwrg-Samsung-DeskTop-System:/usr/local/android-studio/bin$ sudo apt-get
```

```
install oracle-java8-installer
```

패키지 목록을 읽는 중입니다... 완료

의존성 트리를 만드는 중입니다

상태 정보를 읽는 중입니다... 완료

다음 패키지가 자동으로 설치되었지만 더 이상 필요하지 않습니다:

```
linux-headers-4.4.0-64 linux-headers-4.4.0-64-generic
```

```
linux-headers-4.4.0-66 linux-headers-4.4.0-66-generic
```

```
linux-image-4.4.0-64-generic linux-image-4.4.0-66-generic
```

```
linux-image-extra-4.4.0-64-generic linux-image-extra-4.4.0-66-generic
```

```
linux-signed-image-4.4.0-64-generic linux-signed-image-4.4.0-66-generic
```

Use 'sudo apt autoremove' to remove them.

The following additional packages will be installed:

```
gsfonts-x11 java-common oracle-java8-set-default
```

제안하는 패키지:

```
visualvm ttf-baekmuk | ttf-unfonts | ttf-unfonts-core ttf-kochi-gothic
```

```
| ttf-sazanami-gothic ttf-kochi-mincho | ttf-sazanami-mincho
```

```
ttf-arphic-uming
```

다음 새 패키지를 설치할 것입니다:

Android Studio 가 설치될 것이다.

```
kwrg@kwrg-Samsung-DeskTop-System:/usr/local/android-studio/bin$ ./studio.sh
Looking in classpath from com.intellij.util.lang.UrlClassLoader@23ab930d for
/com/sun/jna/linux-x86-64/libjnidispatch.so
Found library resource at jar:file:/usr/local/android-studio/lib/jna.jar!/co
m/sun/jna/linux-x86-64/libjnidispatch.so
Trying /home/kwrg/.AndroidStudio2.3/system/tmp/jna2736621312571416915.tmp
Found jnidispatch at /home/kwrg/.AndroidStudio2.3/system/tmp/jna273662131257
tmp
```

oid
O

IntelliJ Platform



Welcome

Android Studio

Welcome! This wizard will set up your development environment for Android Studio.
Additionally, the wizard will help port existing Android apps into Android Studio
or create a new Android application project.



Previous

Next

Cancel

Finish



Install Type

Choose the type of setup you want for Android Studio:

☒ Standard

Android Studio will be installed with the most common settings and options.
Recommended for most users.

☐ Custom

You can customize installation settings and components installed.

Previous

Next

Cancel

Finish



Verify Settings

If you want to review or change any of your installation settings, click Previous.

Current Settings:

Total Download Size:

832 MB

SDK Components to Download:

Android SDK Build-Tools 24.0.2	47.2 MB
Android SDK Platform 24	78.8 MB
Android SDK Platform-Tools 24.0.3	2.86 MB
Android SDK Tools 25.2.2	288 MB
Android Support Repository	276 MB
Google Repository	109 MB
Intel x86 Emulator Accelerator (HAXM installer)	1.73 MB
Sources for Android 24	28.9 MB

Previous

Next

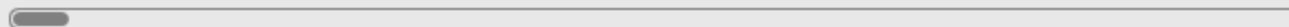
Cancel

Finish



Downloading Components

Downloading...



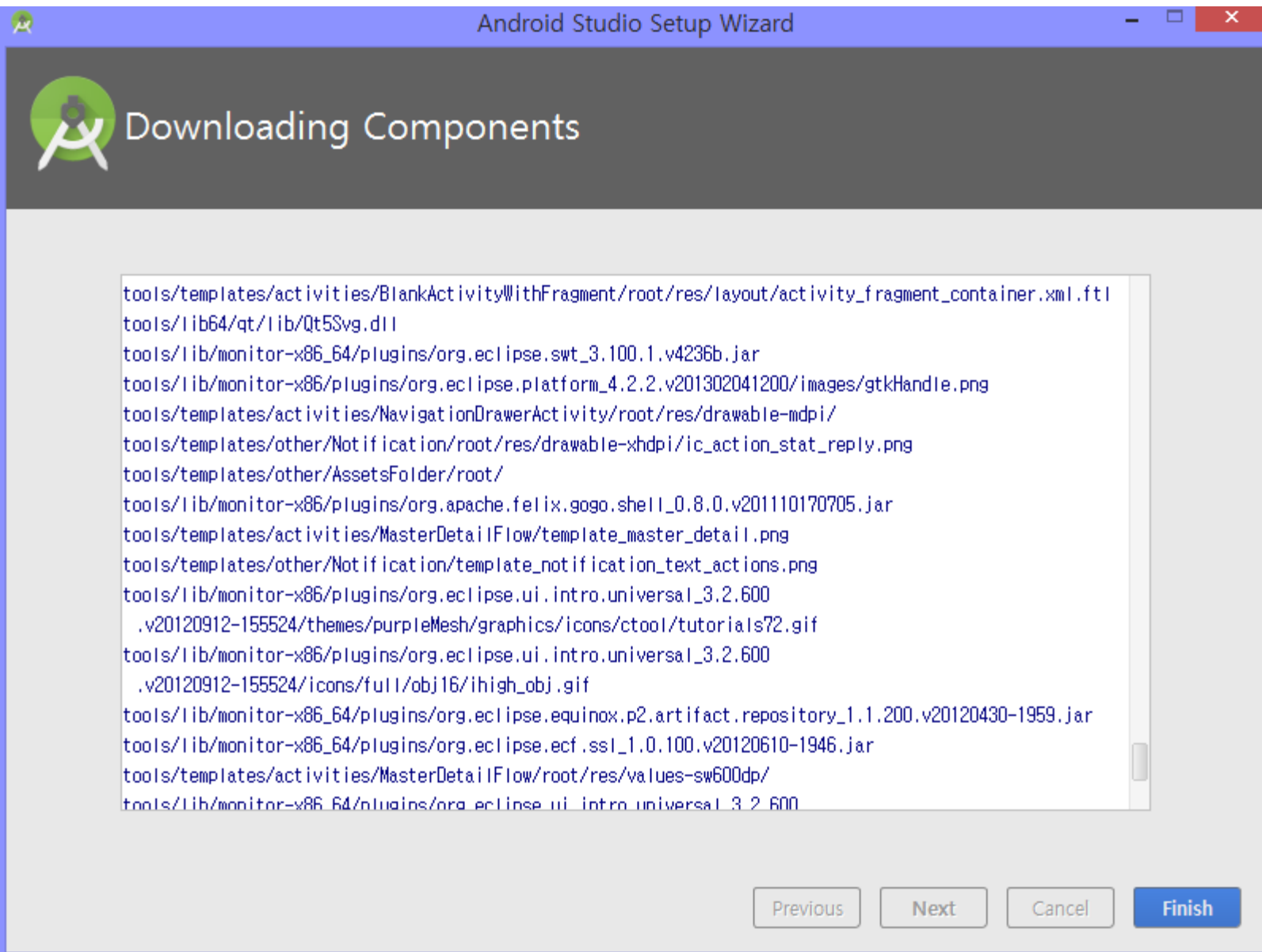
Show Details

Previous

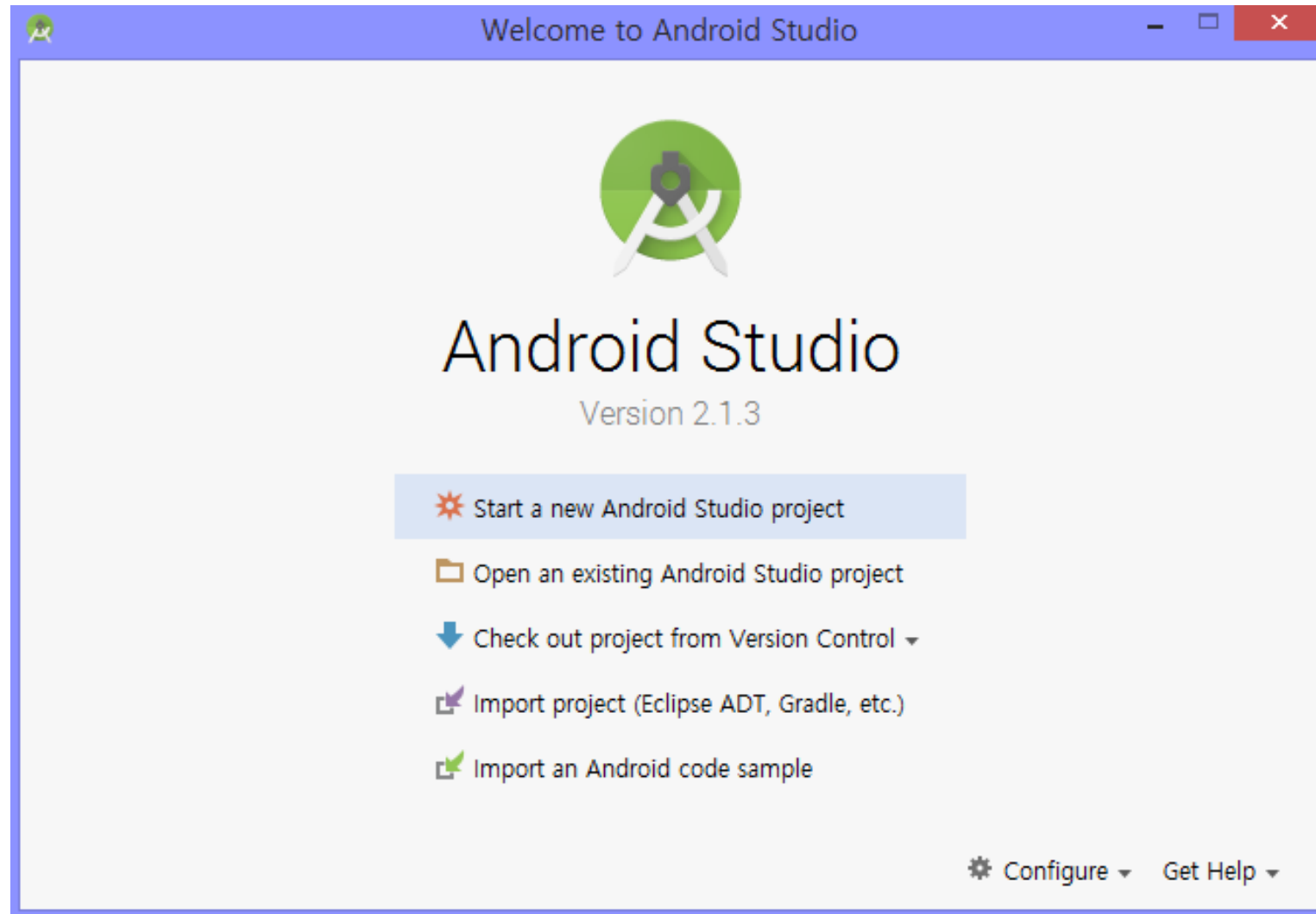
Next

Cancel

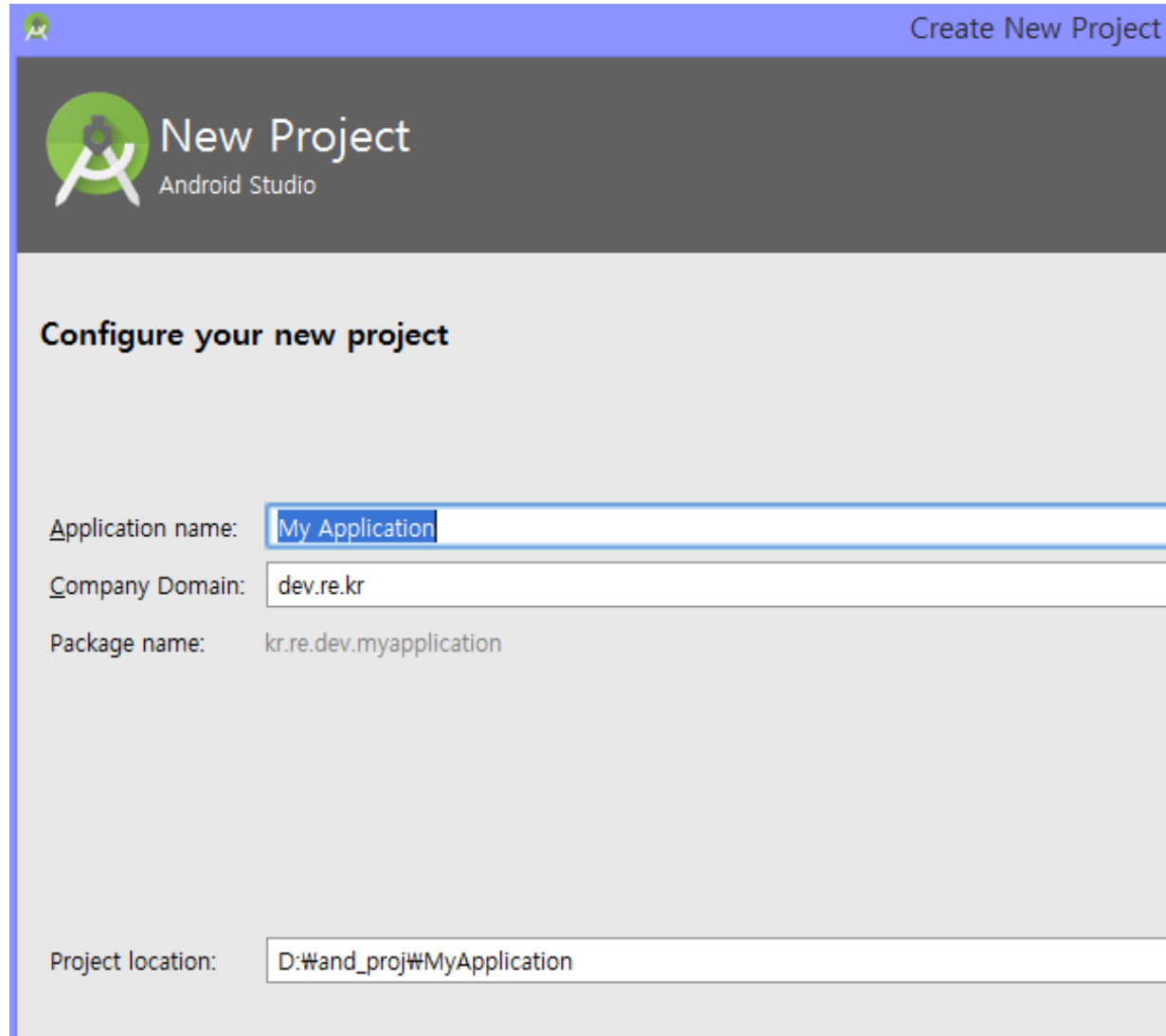
Finish



이제 **Start a new Android Studio project** 를 누르도록 한다.



우선 기본적으로 Android SDK 자체가 잘 구동되는지 확인할 필요가 있으므로 디폴트 값으로 전부 ok ok 하도록 한다.



The screenshot shows the 'Create New Project' dialog in Android Studio. The dialog has a blue header bar with the text 'Create New Project' on the right. Below the header, there is a dark gray section with the Android Studio logo and the text 'New Project' and 'Android Studio'. The main area is light gray and contains the text 'Configure your new project'. Below this, there are four input fields: 'Application name' with the value 'My Application', 'Company Domain' with the value 'dev.re.kr', 'Package name' with the value 'kr.re.dev.myapplication', and 'Project location' with the value 'D:\wand_proj\MyApplication'.

Create New Project

New Project
Android Studio

Configure your new project

Application name: My Application


Company Domain: dev.re.kr

Package name: kr.re.dev.myapplication

Project location: D:\wand_proj\MyApplication

역시 디폴트 값으로 잡고 넘어가도록 한다.

Create New Project

 Target Android Devices

Select the form factors your app will run on

Different platforms may require separate SDKs

☒ Phone and Tablet

Minimum SDK

Lower API levels target more devices, but have fewer features available.
By targeting API 16 and later, your app will run on approximately **95.2%** of the devices that are active on the Google Play Store.
[Help me choose](#)

☐ Wear

Minimum SDK

☐ TV

Minimum SDK

☐ Android Auto

☐ Glass

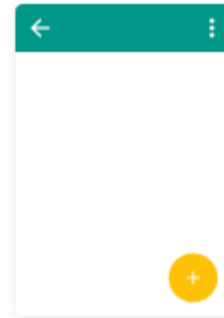
Minimum SDK



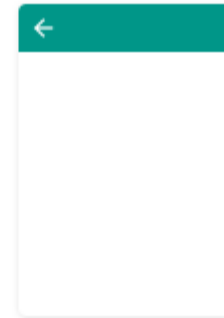
Add an Activity to Mobile



Add No Activity



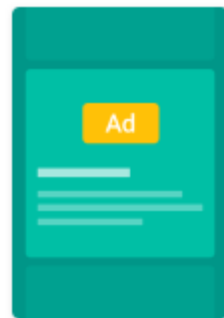
Basic Activity



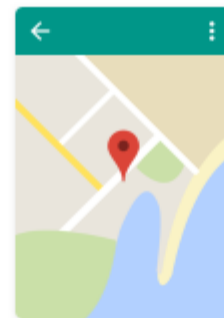
Empty Activity



Fullscreen Activity



Google AdMob Ads Activity



Google Maps Activity



Login Activity



Master/Detail Flow

Previous

Next

Cancel

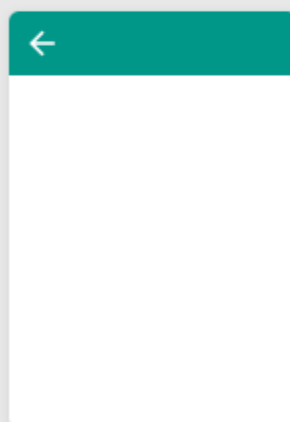
Finish



Customize the Activity



Creates a new empty activity



Empty Activity

Activity Name: MainActivity

☒ Generate Layout File

Layout Name: activity_main

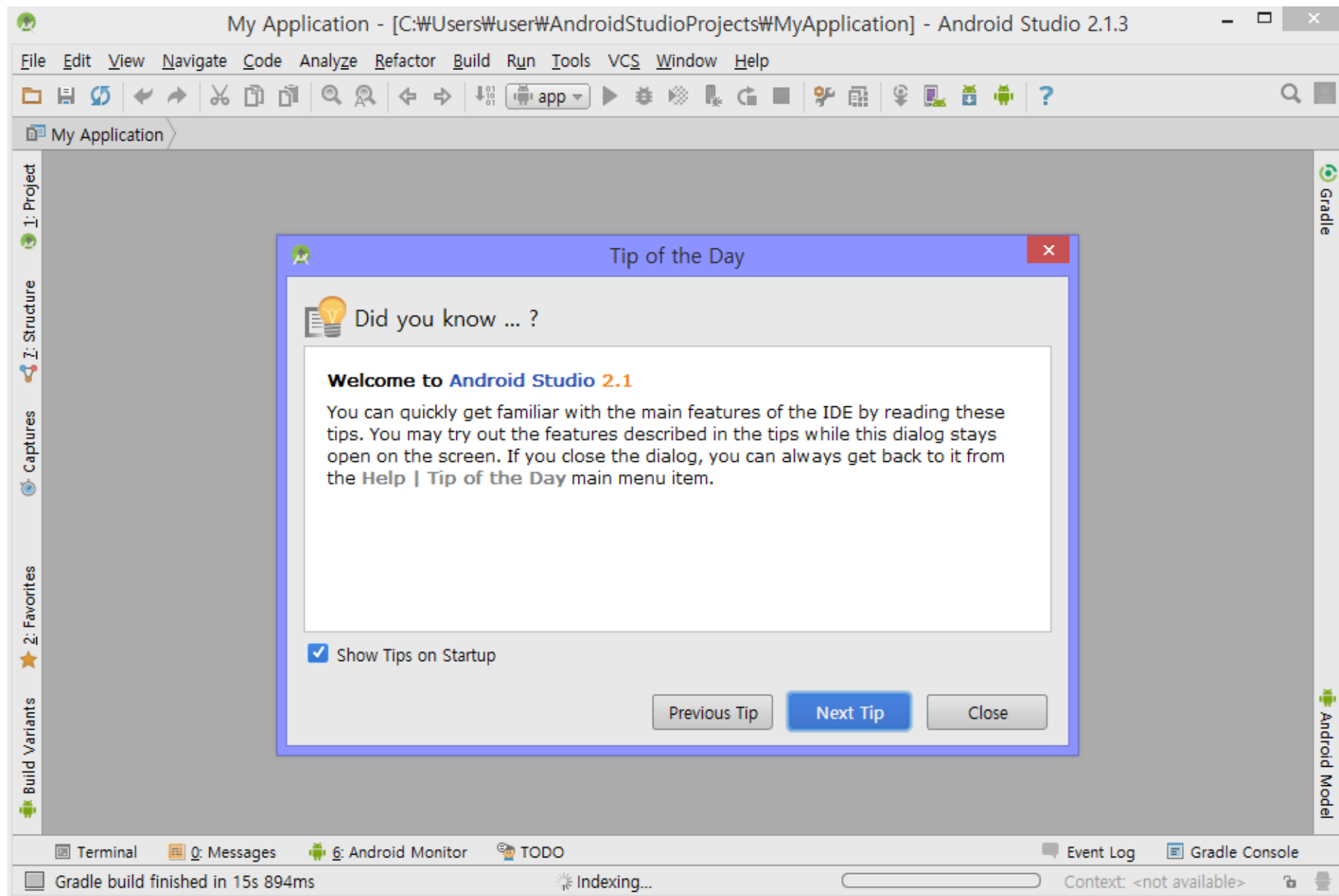
The name of the activity class to create

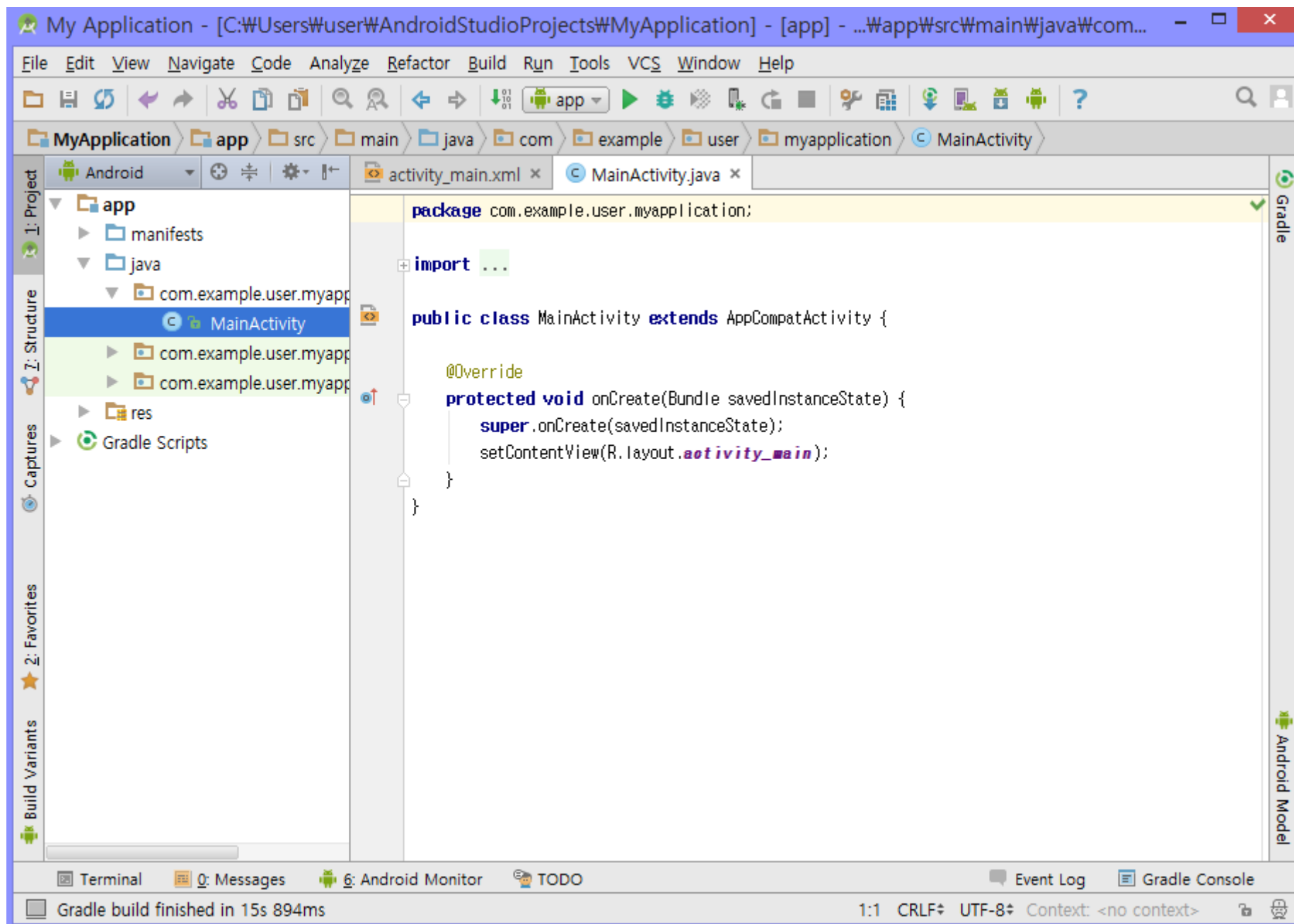
Previous

Next

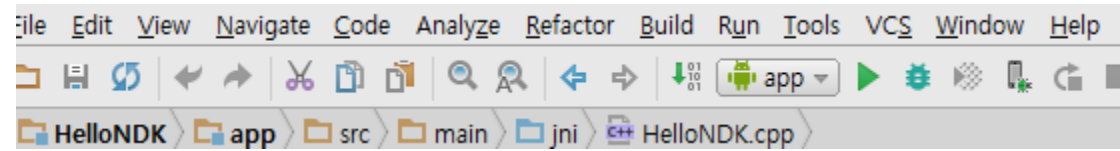
Cancel

Finish

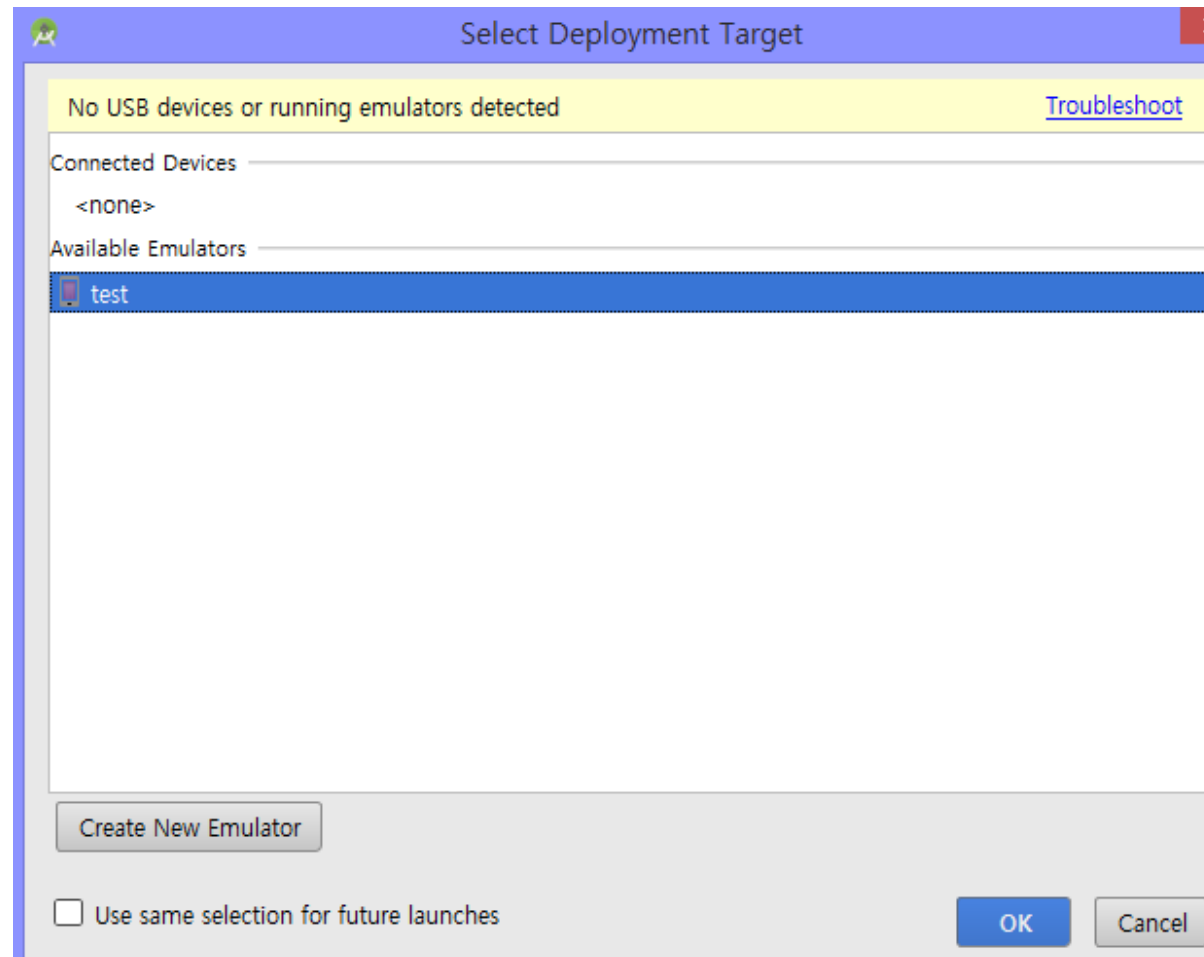




우선 초기 형태를 가지고 바로 위쪽에 플레이 버튼이 보일텐데 이것을 눌러서 구동을 확인해보도록 한다.



그러면 아래와 같은 창이 나타날텐데 Create New Emulator 를 클릭하도록 한다.







그러면 화면 크기등이 적당하게 나오게 되는데 적절한 크기를 선택해주면된다.
 (본인 컴퓨터에 무리가 가지 않을 적당한 선에서 선택하는 것이 좋으며 컴퓨터가 좋다면 무관하다)

Phone				
Tablet	Nexus 5X	5.2"	1080x1920	420dpi
	Nexus 5	4.95"	1080x1920	xxhdpi
	Nexus 4	4.7"	768x1280	xhdpi
	Galaxy Nexus	4.65"	720x1280	xhdpi
	5.4" FWVGA	5.4"	480x854	mdpi
	5.1" WVGA	5.1"	480x800	mdpi
	4.7" WXGA	4.7"	720x1280	xhdpi
	4.65" 720p (Galaxy Nexus)	4.65"	720x1280	xhdpi
	4" WVGA (Nexus S)	4.0"	480x800	hdpi
	3.7" WVGA (Nexus One)	3.4"	480x800	hdpi
	3.7" FWVGA slider	3.7"	480x854	hdpi
	3.4" WQVGA	3.4"	240x432	ldpi
	3.3" WQVGA	3.3"	240x400	ldpi
	3.2" QVGA (ADP2)	3.2"	320x480	mdpi
	3.2" HVGA slider (ADP1)	3.2"	320x480	mdpi





다음으로 OS 버전이라고 할 수 있는 부분을 설정하면 되는데
 본인 휴대폰의 버전에 맞게 해도 되고 최신 버전으로 테스트해봐도 된다.
 무난하게 Jelly Bean(하위 호환) 혹은 Nougat 이나 Marshmallow 등(최신)을 선택하는 것이 좋다.

Recommended x86 Images Other Images			
Release Name	API Level ▼	ABI	Target
<i>null</i> Download	25	<i>x86_64</i>	<i>Android null (with Google APIs)</i>
<i>null</i> Download	25	<i>x86</i>	<i>Android null (with Google APIs)</i>
<i>Nougat</i> Download	24	<i>x86_64</i>	<i>Android 7.0 (with Google APIs)</i>
<i>Nougat</i> Download	24	<i>x86</i>	<i>Android 7.0 (with Google APIs)</i>
Marshmallow	23	x86	Android 6.0 (with Google APIs)
<i>Marshmallow</i> Download	23	<i>x86_64</i>	<i>Android 6.0 (with Google APIs)</i>
<i>Lollipop</i> Download	22	<i>x86</i>	<i>Android 5.1 (with Google APIs)</i>
<i>Lollipop</i> Download	22	<i>x86_64</i>	<i>Android 5.1 (with Google APIs)</i>
<i>KitKat</i> Download	19	<i>x86</i>	<i>Android 4.4 (with Google APIs)</i>
<i>Jelly Bean</i> Download	18	<i>x86</i>	<i>Android 4.3 (with Google APIs)</i>
<i>Jelly Bean</i> Download	17	<i>x86</i>	<i>Android 4.2 (with Google APIs)</i>
Jelly Bean	16	x86	Android 4.1 (with Google APIs)
<i>Gingerbread</i> Download	10	<i>x86</i>	<i>Android 2.3.3 (with Google APIs)</i>

아래와 같이 적당한 이름을 지어주고 Finish 하면 가상 머신이 만들어진다.

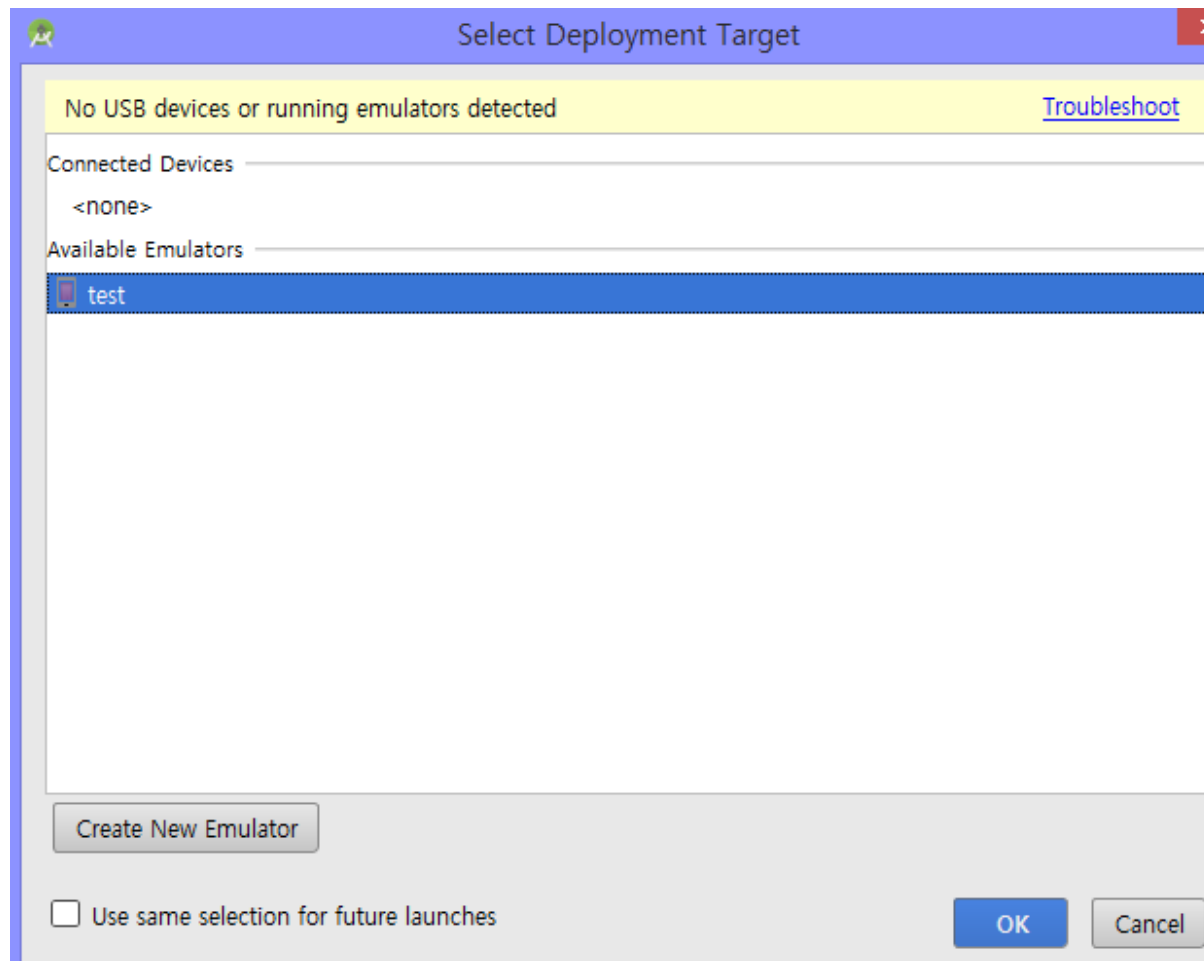
AVD Name	<input type="text" value="QVGA (ADP2) API 23"/>		
 3.2" QVGA (ADP2)	3.2" 320x480 mdpi	<button>Change...</button>	
 Marshmallow	Android 6.0 x86	<button>Change...</button>	
Startup size and orientation	Scale:	<input type="text" value="Auto"/> ▼	
	Orientation:	<div><div> Portrait</div><div> Landscape</div></div>	
Emulated Performance	Graphics:	<input type="text" value="Auto"/> ▼	
Device Frame	<input type="checkbox"/> Enable Device Frame		

아래와 같이 적당한 이름을 지어주고 Finish 하면 가상 머신이 만들어진다.
이 이름을 test 로 만들면 test 가 나오게 되며
실제 휴대폰을 연결할 경우 Connected Devices 에 당사자의 휴대폰이 잡히게 된다.
(USB 를 활용하여 휴대폰과 PC 를 연결하면 됨)

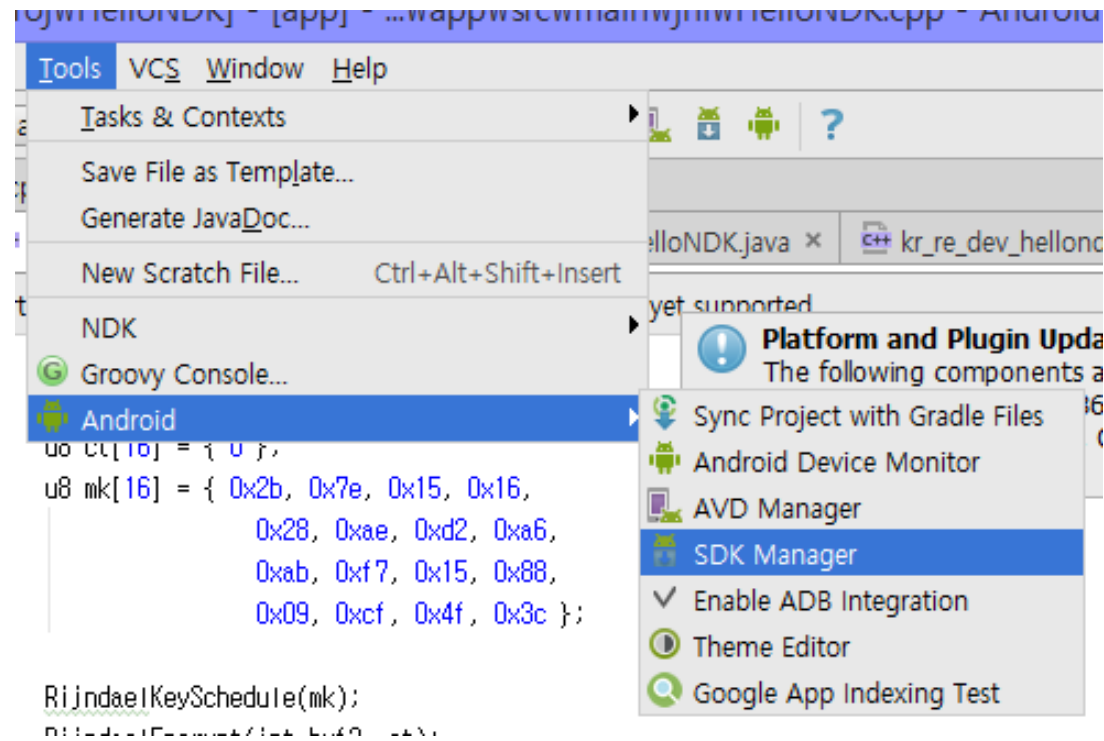
AVD Name	<input type="text" value="QVGA (ADP2) API 23"/>		
 3.2" QVGA (ADP2)	3.2" 320x480 mdpi	<button>Change...</button>	
 Marshmallow	Android 6.0 x86	<button>Change...</button>	
Startup size and orientation	Scale:	<input type="text" value="Auto"/>	
	Orientation:	<div><div> Portrait</div><div> Landscape</div></div>	
Emulated Performance	Graphics:	<input type="text" value="Auto"/>	
Device Frame	<input type="checkbox"/> Enable Device Frame		

아무튼 다시 플레이 버튼을 눌러보면 아래와 같은 창이 나타날 것이다.
맨 처음에는 Available Emulators 가 아무것도 없었겠지만
현재는 우리가 추가했기 때문에 나타날 것이다.
이를 선택하고 OK 누르면 해당 가상 머신에서 안드로이드가 구동하는 것을 볼 수 있을 것이다.

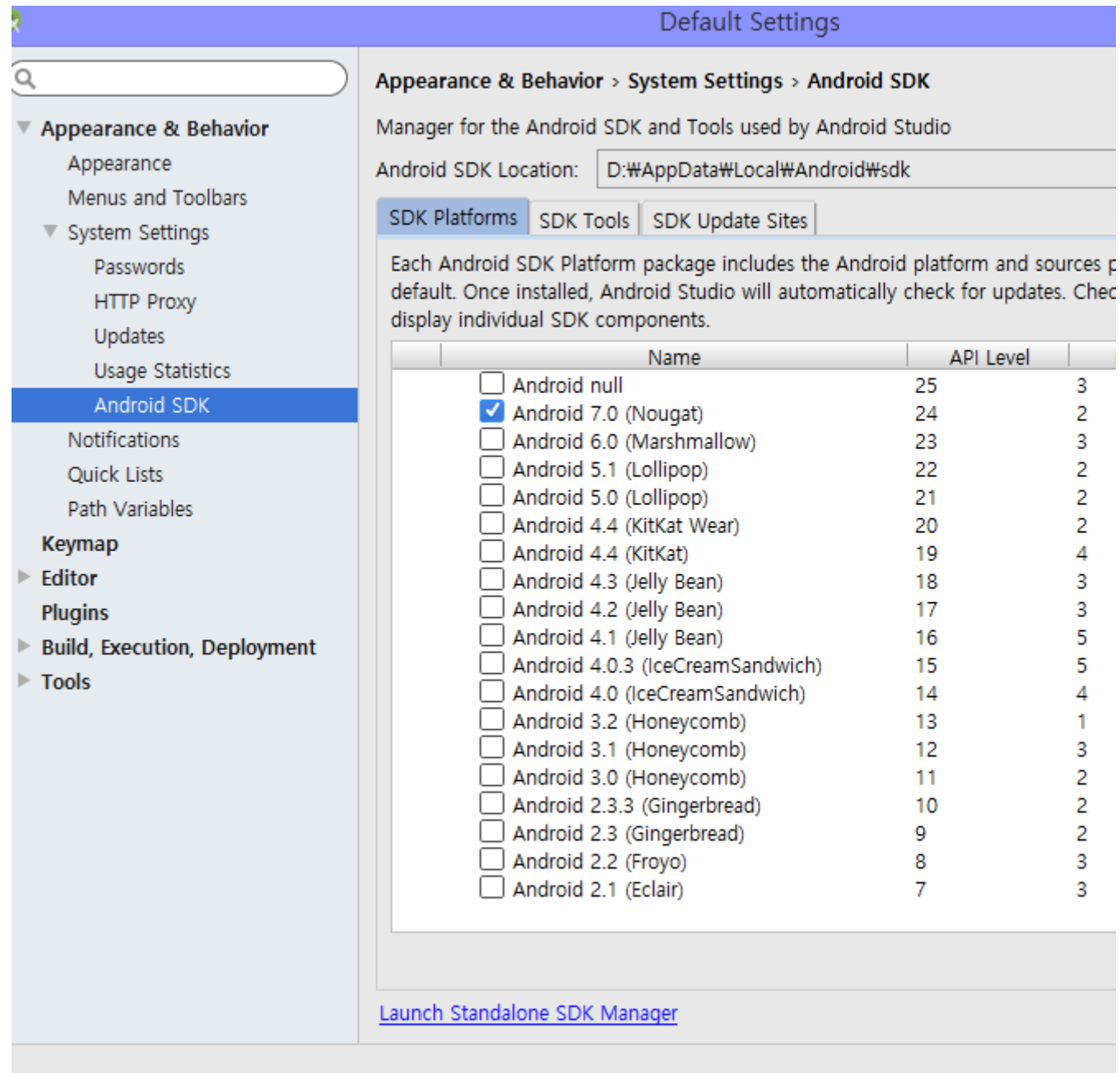
이때 주의할 것은 Instant Run 을 설치하라고 나오는데 Install and Continue 를 누른다.
그리고 반드시 Done 이 나오기 전까지 기다렸다가 나오면 Finish 를 누르고 실행하도록 한다.
정상적으로 잘 수행되었다면 이제 다음으로 NDK 를 설정하면 되겠다.



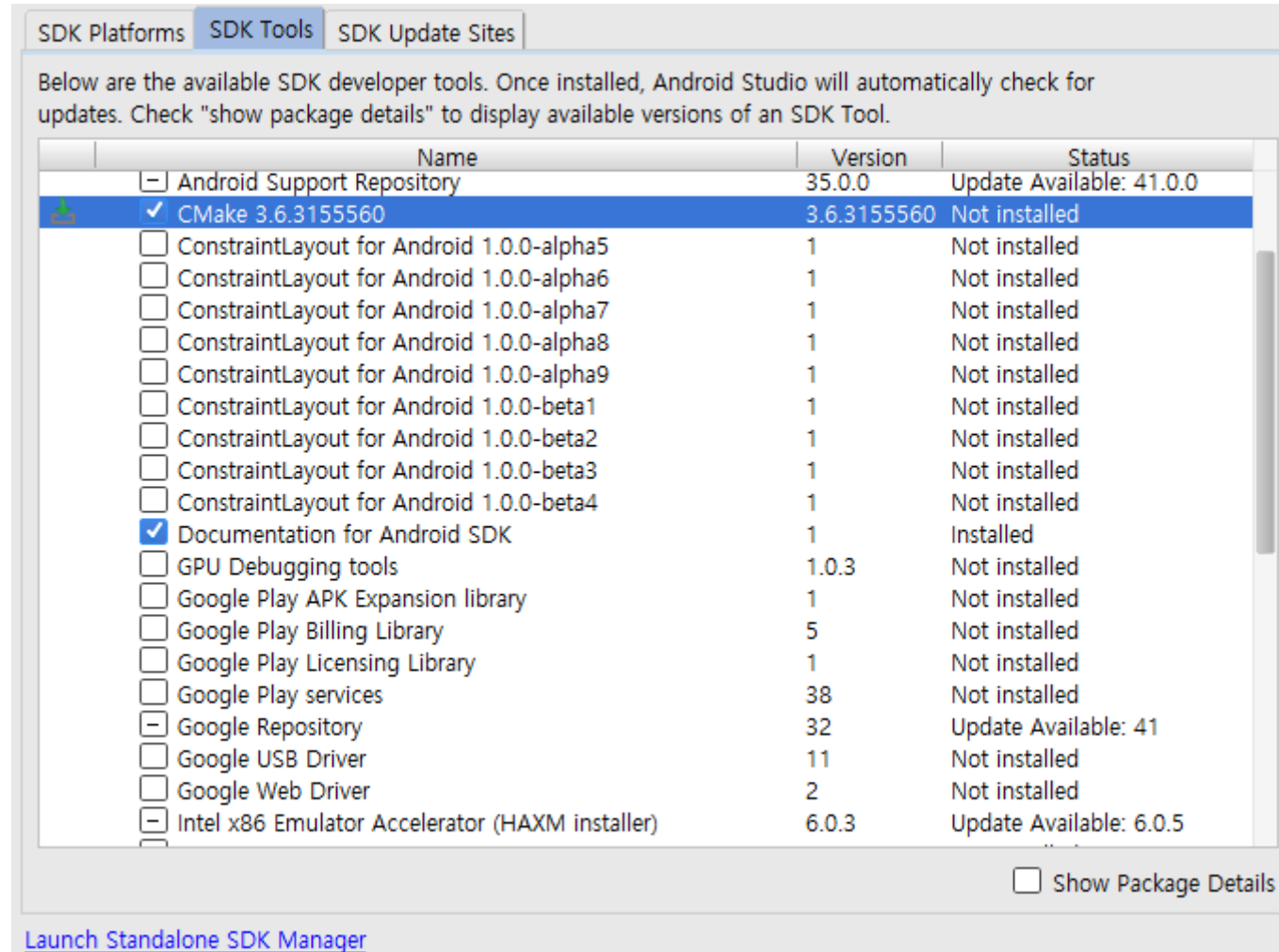
이제 최 상단의 Tools -> Android -> SDK Manager 를 누르도록 한다.



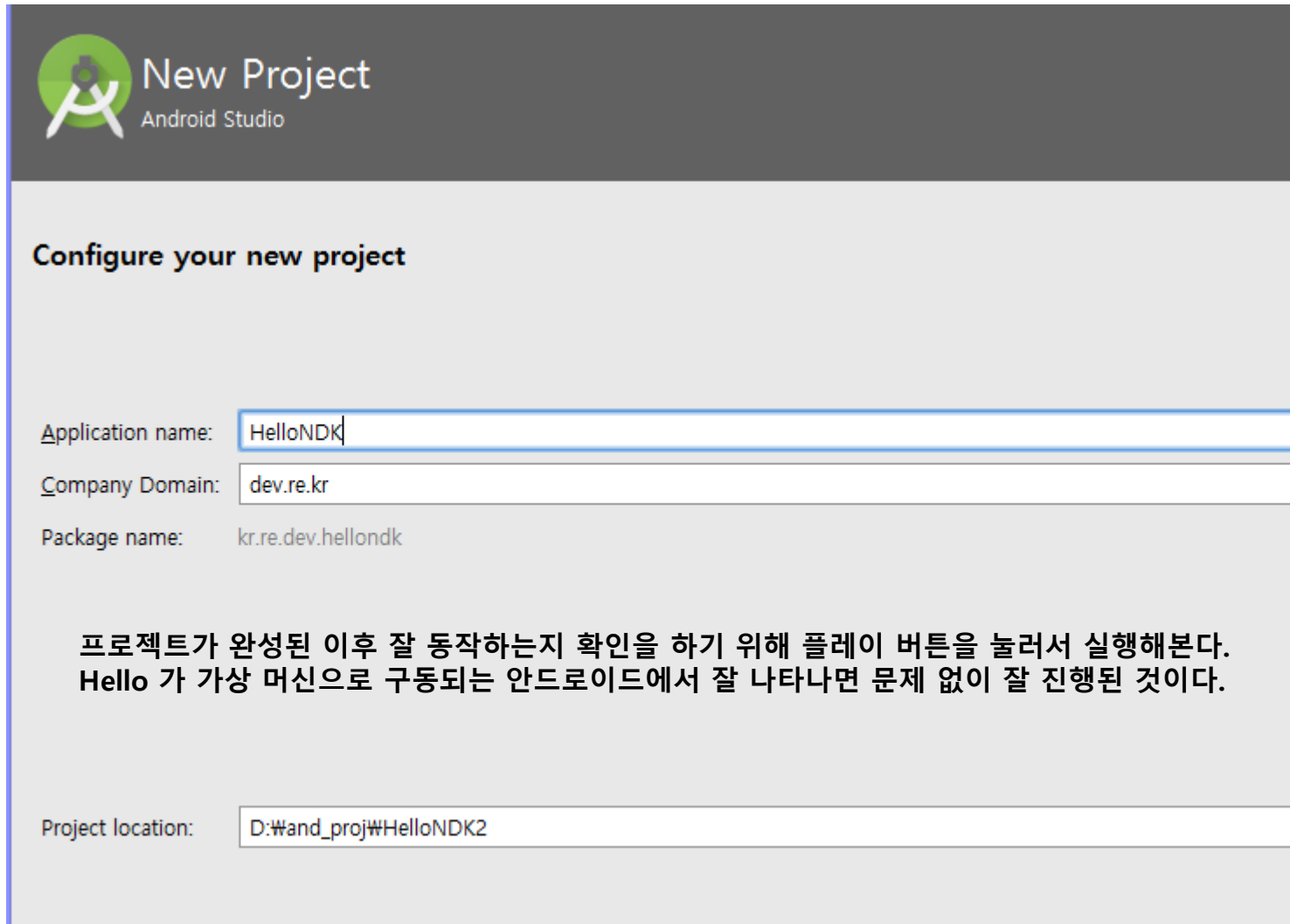
그러면 아래와 같은 화면이 나타날 것이다.




여기서 SDK Tools 로 가서 NDK, LLDB, CMake 를 선택해주도록 한다.
Apply 를 누르면 설치가 시작되는데 역시 반드시 Done 이 나올때까지 기다린 후 Finish 를 해줘야만 한다.



이제 기본적인 조건이 모두 갖춰졌으므로 NDK 프로젝트를 작성해보도록 하자 ~~!!~!
필자의 경우 Android Studio 의 버전이 낮아서 나오지는 않지만
현재 최신 버전을 설치한 경우에는 여기에 Include C++ Support 부분이 존재한다.
Include C++ Support 를 선택하고 Project Location 을 공백 문자, 한글 없이 작성한다.
이후는 그냥 전부다 Next Next Next 하면 된다.



 **New Project**
Android Studio

Configure your new project

Application name:

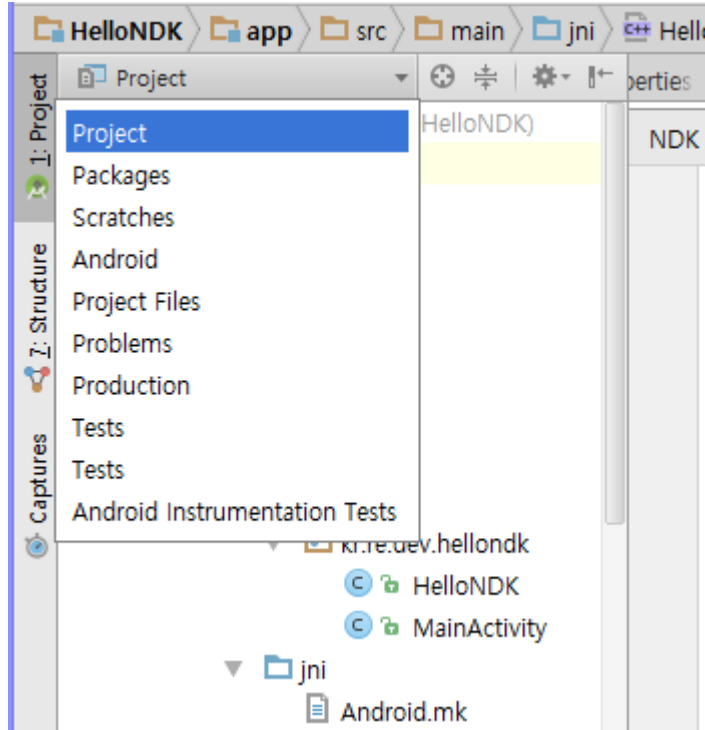
Company Domain:

Package name:

프로젝트가 완성된 이후 잘 동작하는지 확인을 하기 위해 플레이 버튼을 눌러서 실행해본다.
Hello 가 가상 머신으로 구동되는 안드로이드에서 잘 나타나면 문제 없이 잘 진행된 것이다.

Project location:

다음으로 Project View 를 선택하고 HelloNDK -> app -> src -> main -> java 를 우클릭하여 HelloNDK.java 를 만든다.



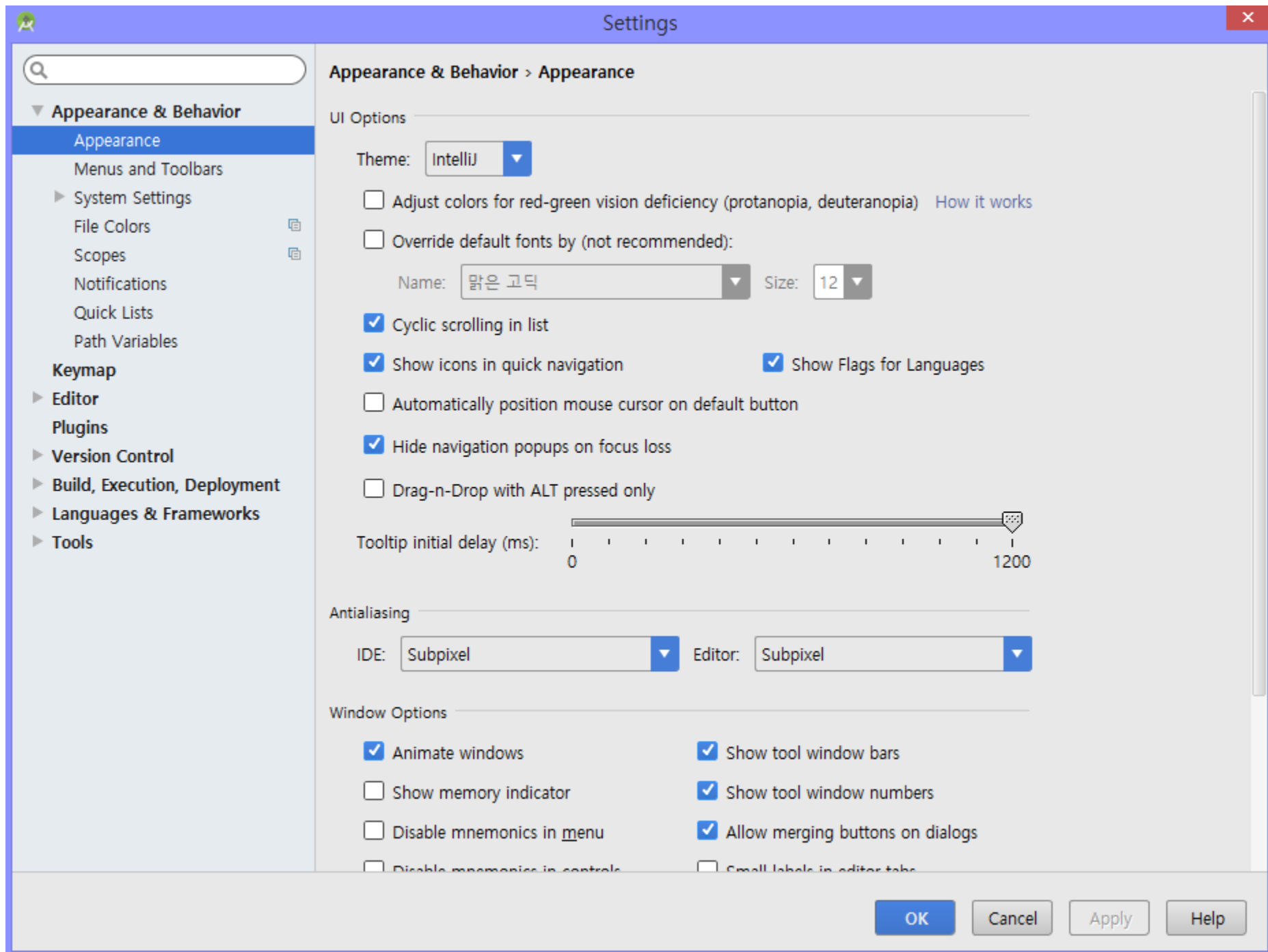
Project View 를 선택한다는 것은 저기서 Project 를 선택함을 의미한다.

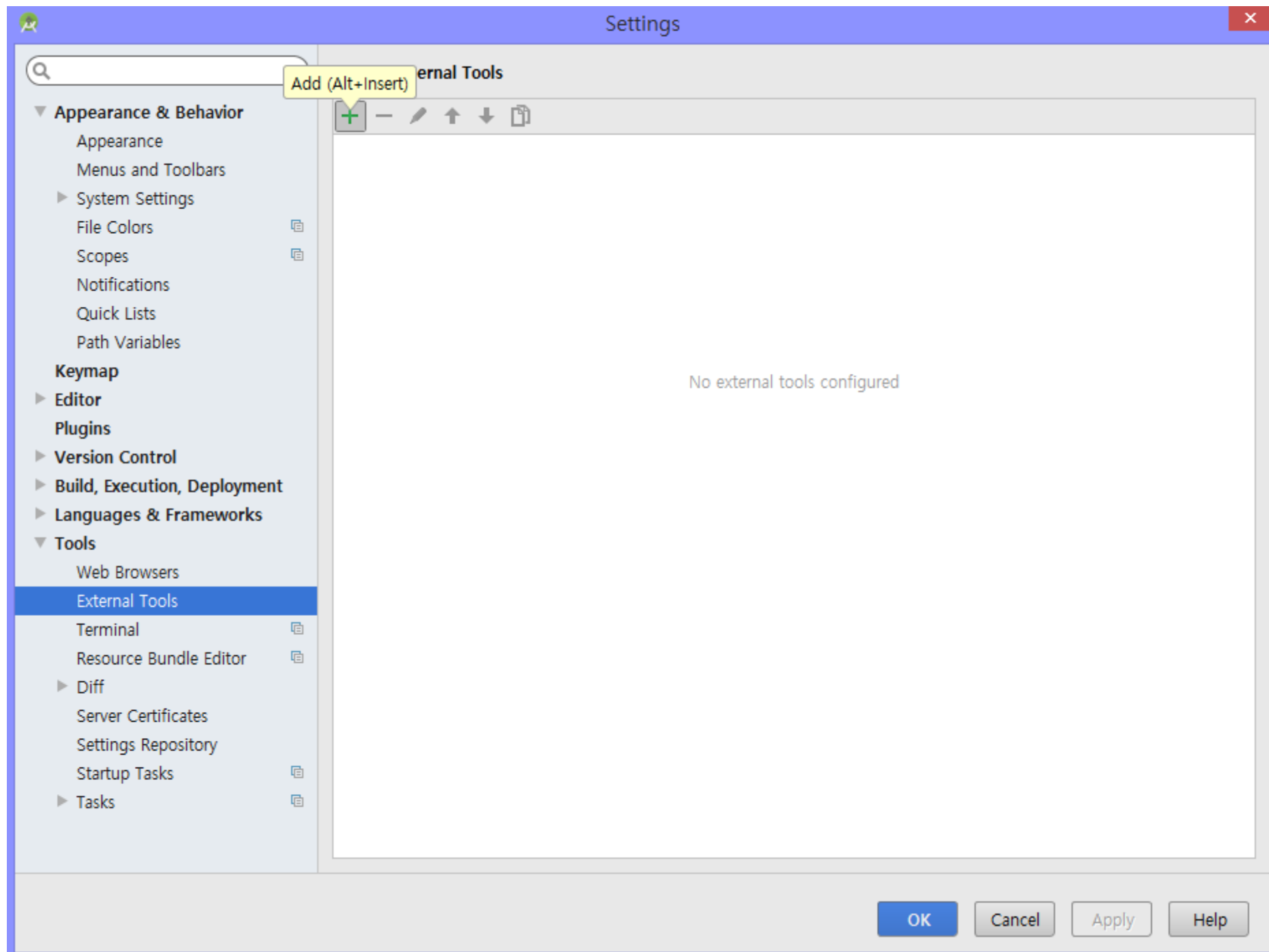
HelloNDK.java 코드는 아래와 같이 작성해준다.
코드를 모두 작성한 이후에 **Ctrl + F9** 를 눌러서 한 번 **Build** 해준다.
그려야지만 **javah** 가 어떤 녀석들을 참조해야하는지 정신을 차린다.



```
package kr.re.dev.hellondk;

/**
 * Created by keh on 2016-10-21.
 */

public class HelloNDK {
    static {
        System.loadLibrary("native-lib");
    }
    public native String stringFromJNI();
}
```





 Create Tool 

Name:

Group:

Description:

Options

☒ Synchronize files after execution

☒ Open console

☐ Show console when a message is printed to standard output stream

☐ Show console when a message is printed to standard error stream

Show in

☒ Main menu

☒ Editor menu

☒ Project views

☒ Search results

Tool settings

Program:

Parameters:

Working directory:

Create Tool

Name:

javah

Group:

NDK

Description:

Create NDK Header File

Options

☒ Synchronize files after execution

☒ Open console

Output Filters...

☐ Show console when a message is printed to standard output stream

☐ Show console when a message is printed to standard error stream

Show in

☒ Main menu

☒ Editor menu

☒ Project views

☒ Search results

Tool settings

Program:

...

Insert macro...

Parameters:

Insert macro...

Working directory:

...

Insert macro...

OK

Cancel

Help

여기에 Program 을 보면 필자는 아래와 같이 했지만
AndroidStudio로 한 경우에 앞서서 AndroidStudio 를 설치한 경로에 jre/bin/javah.exe 를 선택해주도록 한다.

Create Tool

Name: javah Group: NDK

Description: Create NDK Header File

Options

☒ Synchronize files after execution ☒ Open console ☐ Show console when a message is printed to standard output stream ☐ Show console when a message is printed to standard error stream

Show in

☒ Main menu ☒ Editor menu ☒ Project views ☒ Search results

Tool settings

Program: D:\Java\jdk1.8.0_101\bin\javah.exe

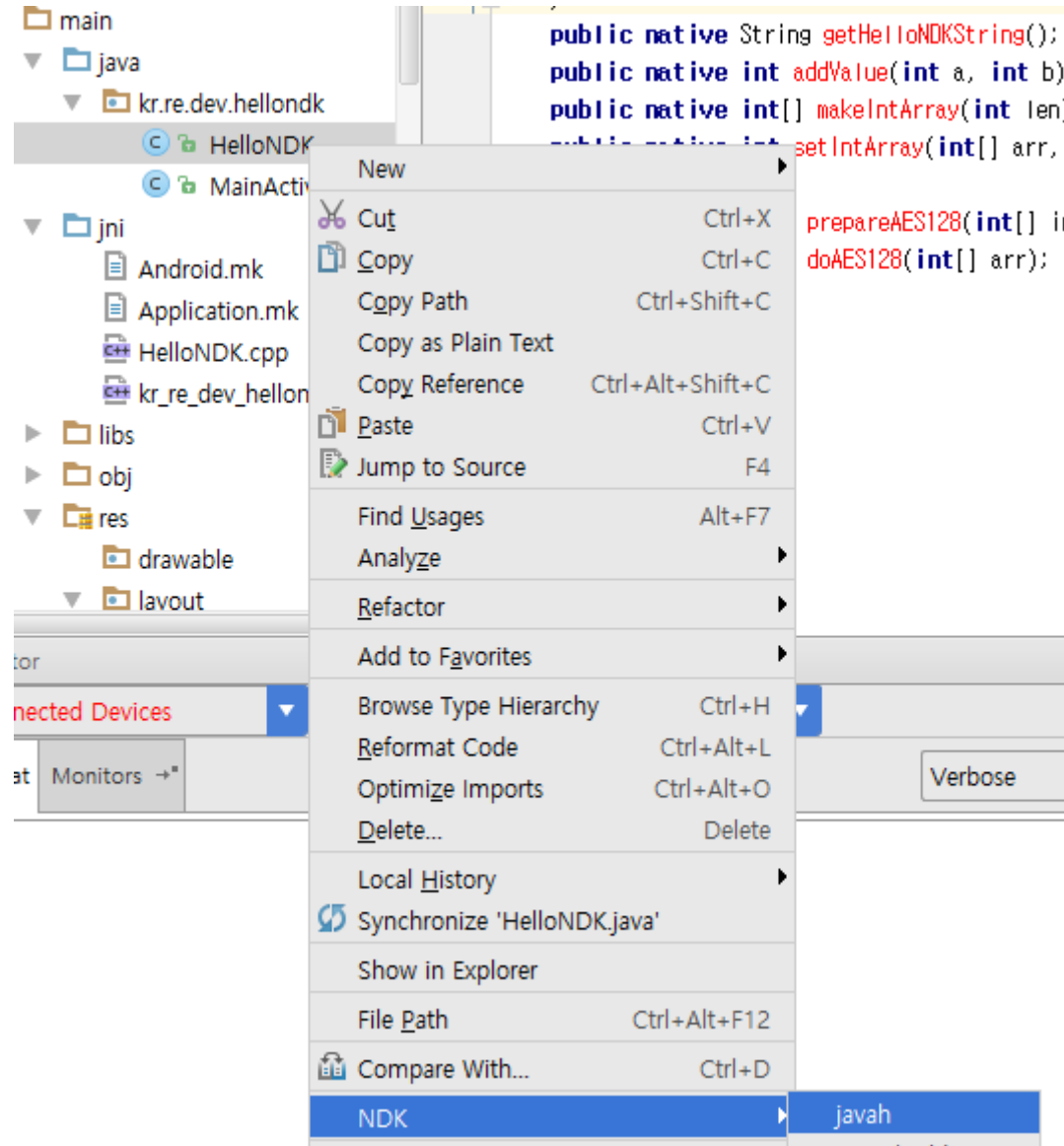
Parameters: -classpath \$Classpath\$ -v jni \$FileClass\$

Working directory: \$ProjectFileDir\$\app\src\main\jni

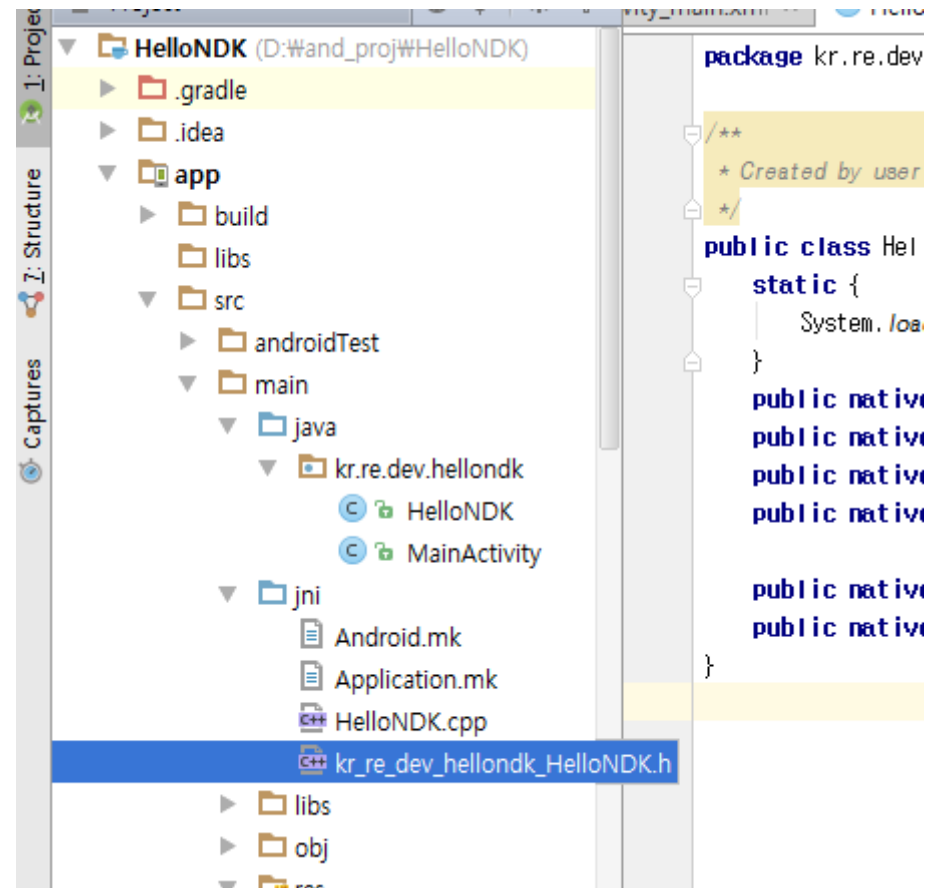
OK Cancel Help

또한 Parameters 도 `-classpath $Classpath$ -v -jni $FileClass$` 로 바꿔주도록 한다(jni 에 '-' 가 빠져있음)
Working directory 는 `$ProjectFileDir$\app\src\main\cpp` 로 바꿔주도록 한다.

이제 아래와 같이 HelloNDK.java 를 우클릭하여 javah 로 C 코드와 연동할 준비를 한다.



구동이 정상적으로 잘 진행되면 아래와 같이 C 코드 형태의 헤더 파일이 만들어질 것이다.



다음으로 필자는 Android Studio 버전이 낮아서 안보이지만
당사자들의 경우 cpp 쪽에 native-lib.cpp 라는 코드가 존재할 것이다.
이 부분을 아래와 같이 변경하도록 한다.

```
#include <jni.h>
#include <string>

#include "kr_re_dev_hellondk_HelloNDK.h"

JNIEXPORT jstring JNICALL Java_kr_re_dev_hellondk_HelloNDK_stringFromJNI(
    JNIEnv *env,
    jobject /* this */) {
    std::string hello = "Hello from C++";
    return env->NewStringUTF(hello.c_str());
}
```

다음으로 MainActivity.java 코드를 아래와 같이 작성한다.

```
public class MainActivity extends AppCompatActivity {

    // Used to load the 'native-lib' library on application startup.
    /*
    static {
        System.loadLibrary("native-lib");
    }
    */

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // Example of a call to a native method
        TextView tv = (TextView) findViewById(R.id.sample_text);
        HelloNDK hello = new HelloNDK();
        tv.setText(hello.stringFromJNI());
    }

    /**
     * A native method that is implemented by the 'native-lib' native library,
     * which is packaged with this application.
     */
    //public native String stringFromJNI();
}
```

NDK 프로젝트를 처음 만들었을때와 같은 결과가 나올것이다.
이로서 Android Java 와 C 코드를 연동하기 위한 모든 기본적인 작업이 완료되었다고할 수 있다.

이후 더 깊은 내용을 들어가고자 한다면
리눅스 시스템 프로그래밍과 리눅스 커널쪽을 공부해서 디바이스 드라이버를 공부하면됨.

바로 가기를 만들고자 한다면 아래와 같이 만들 수 있다.

```
kwrg@kwrg-Samsung-DeskTop-System: /usr/local/android-studio/bin
kwrg@kwrg-Samsung-DeskTop-System: /usr/local/android-studio/bin$ vim ~/.local/share/applications/android-studio.desktop
```

설정 파일을 아래와 같이 만들어준다.

```
[Desktop Entry]
Name=Android Studio
Comment=Android Developement Tool.
Exec=/usr/local/android-studio/bin/studio.sh
Icon=/usr/local/android-studio/bin/studio.png
Terminal=false
Type=Application
StartupNotify=true
Categories=TextEditor;Development;
```

그리고 아래와 같이 실제 데스크톱에 나타나게 만든다.

```
kwrg@kwrg-Samsung-DeskTop-System:/usr/local/android-studio/bin$ mv ~/.local/  
share/applications/android-studio.desktop ~/Desktop/  
kwrg@kwrg-Samsung-DeskTop-System:/usr/local/android-studio/bin$ vi ~/Desktop  
/android-studio.desktop  
kwrg@kwrg-Samsung-DeskTop-System:/usr/local/android-studio/bin$ chmod +x ~/D  
esktop/android-studio.desktop  
kwrg@kwrg-Samsung-DeskTop-System:/usr/local/android-studio/bin$
```

이후에 아이콘을 옆에 탭에 끌당하면 우측에 탭에 들어가게 된다.

Android NDK

Dev Env Setting for Windows

Android Development Environment Configuration

과거의 복잡함

1. <http://developer.android.com/sdk/>
2. <http://www.eclipse.org/downloads/>
3. <http://developer.android.com/sdk/installing/installing-adt.html>
4. <http://developer.android.com/sdk/index.html>

위 링크의 모든 것을 다운받고 설정을 맞춰줘야 하는 복잡함이 존재했음

현재

1. <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

여기 들어가서 라이선스에 동의하고 운영체제 버전에 맞는 jdk 를 다운받으면 됨

2. <https://developer.android.com/studio/index.html>

위 사이트 들어가서 ANDROID STUDIO 다운로드를 누르면 끝남
물론 추가적으로 Java 환경 변수 설정등이 필요하긴한데 우선 저것을 다운받도록 한다.

다음 페이지에 그림과 함께 설명이 있음

Java SE Development Kit 8 Downloads

Thank you for downloading this release of the Java™ Platform, Standard Edition Development Kit (JDK™). The JDK is a development environment for building applications, applets, and components using the Java programming language.

The JDK includes tools useful for developing and testing programs written in the Java programming language and running on the Java platform.

See also:

- [Java Developer Newsletter](#): From your Oracle account, select **Subscriptions**, expand **Technology**, and subscribe to **Java**.
- [Java Developer Day](#) hands-on workshops (free) and other events
- [Java Magazine](#)

JDK 8u101 Checksum

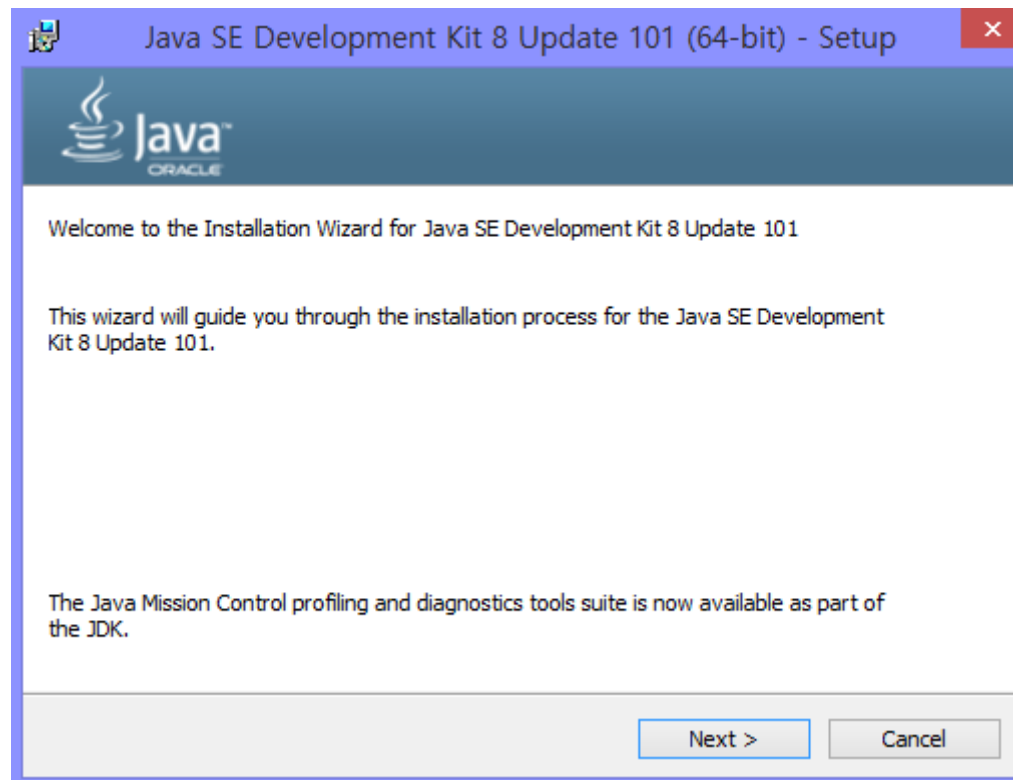
JDK 8u102 Checksum

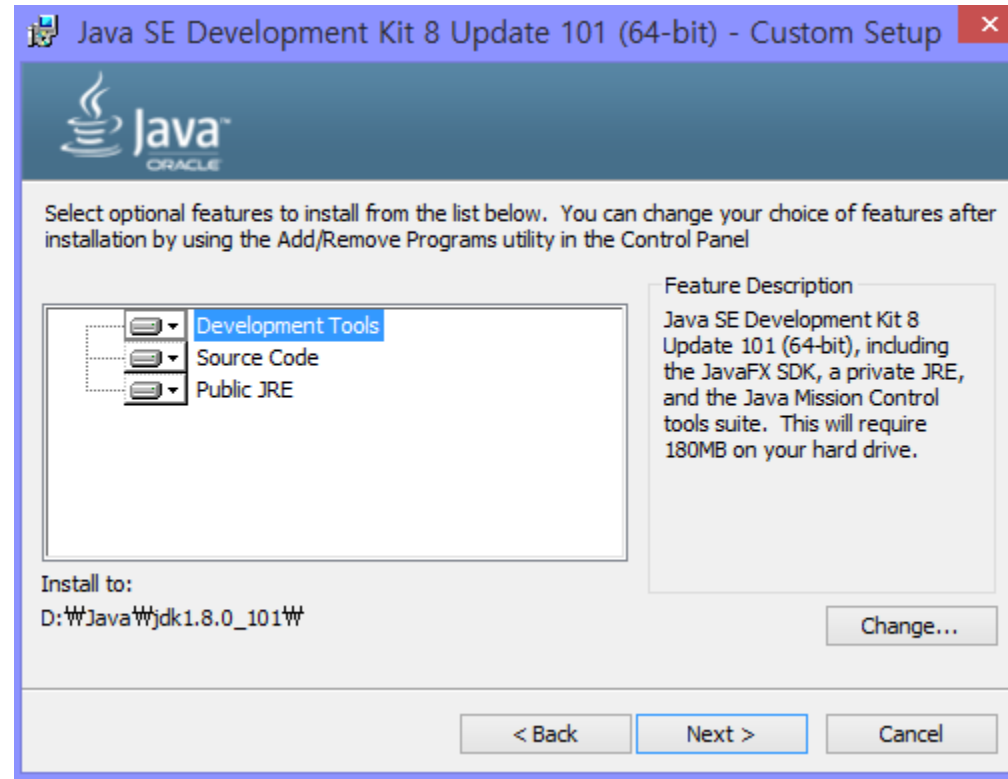
Java SE Development Kit 8u101

You must accept the [Oracle Binary Code License Agreement for Java SE](#) to download this software.

☐ Accept License Agreement ☒ Decline License Agreement

Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	77.77 MB	jdk-8u101-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	74.72 MB	jdk-8u101-linux-arm64-vfp-hflt.tar.gz
Linux x86	160.28 MB	jdk-8u101-linux-i586.rpm
Linux x86	174.96 MB	jdk-8u101-linux-i586.tar.gz
Linux x64	158.27 MB	jdk-8u101-linux-x64.rpm
Linux x64	172.95 MB	jdk-8u101-linux-x64.tar.gz
Mac OS X	227.36 MB	jdk-8u101-macosx-x64.dmg
Solaris SPARC 64-bit	139.66 MB	jdk-8u101-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	98.96 MB	jdk-8u101-solaris-sparcv9.tar.gz
Solaris x64	140.33 MB	jdk-8u101-solaris-x64.tar.Z
Solaris x64	96.78 MB	jdk-8u101-solaris-x64.tar.gz
Windows x86	188.32 MB	jdk-8u101-windows-i586.exe
Windows x64	193.68 MB	jdk-8u101-windows-x64.exe





위와 같이 경로에 띄어쓰기 공백이 없게 만들어주도록 한다.
또 중요한 한 가지는 한국에 대한 배려가 없기 때문에 되도록이면 경로명에 한글은 자제하도록 한다.

Java 설치 - 대상 폴더



대상 폴더

Java를 다른 폴더에 설치하려면 "변경"을 누르십시오.

설치 위치:

D:\Java\jre1.8.0_101

변경(C)...

< 뒤로(B)

다음(N) >

Java 설치 - 진행률



상태: Java 설치



ATMs, Smartcards, POS Terminals, Blu-ray Players, PCs
Set Top Boxes, Mobile Phones, Servers, Switches
Routers, Smart Meters, Medical Devices
Automobiles, Airplane Ticketing, Lottery
Systems, Access Control Systems, Building Controls
Programs, etc.

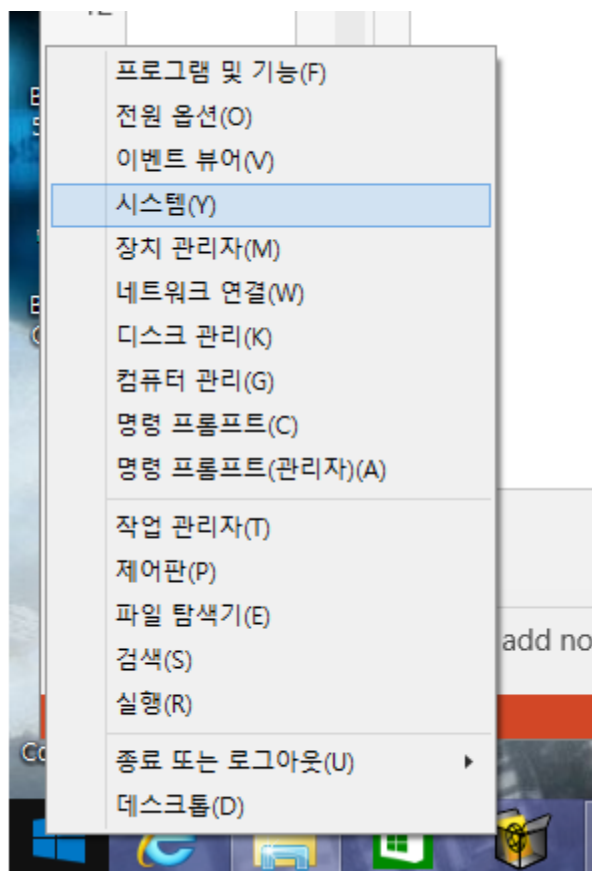
3 Billion Devices Run Java

 Java™ | #1 Development Platform

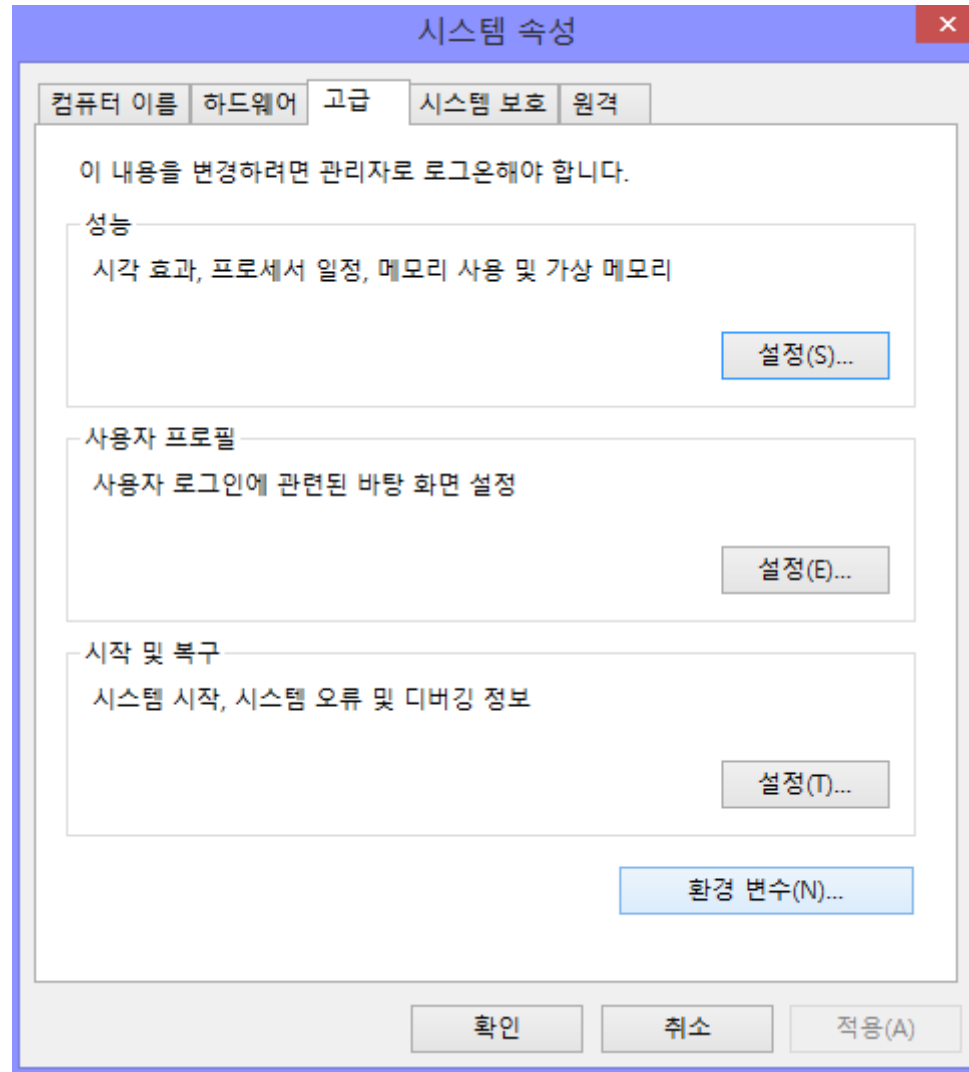
ORACLE™



설치가 잘 완료되면 아래와 같이 시스템 -> 고급 시스템 설정으로 이동한다.



그리고 환경 변수를 누른다.

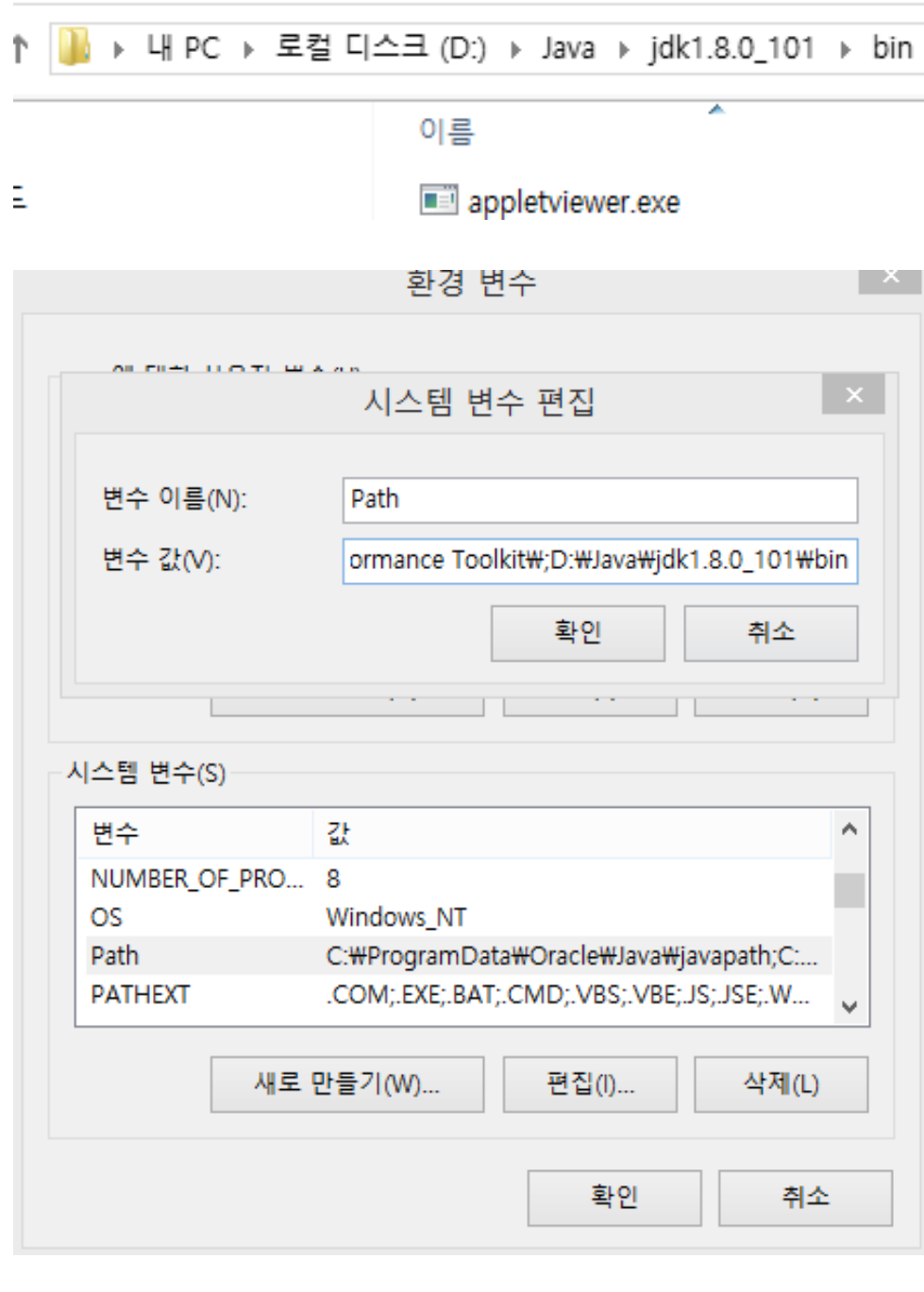


시스템 변수에서 Path 를 찾는다.
기존에 Path 에 있던 내용이 덮어써질 경우 문제가 발생함
그러므로 실수를 대비하여 기존 내용을 백업해둔다.
(메모장 같은데 복사 붙여넣기 해두라는 뜻)

이것이 완료되었다면 Path 의 맨 뒤쪽으로 이동한다.

‘;’ (세미콜론) 이 존재한다면
바로 우리가 Java 를 설치한 위치에 있는
bin 의 경로를 복사 붙여넣기 해주도록 한다.

그렇지 않다면
‘;’ (세미콜론) 을 붙여주고
위의 작업을 동일하게 수행하면 된다.



Environment Setting(Android Studio Installation)

Android Programming 을 수행하기 위한 기본적인 환경 설정을 해보자!

<https://developer.android.com/studio/index.html>

위의 링크로 이동하여 Android Studio 를 다운로드 하자!

Android Studio

Android용 공식 IDE

Android Studio는 모든 유형의 Android 기기에서 앱을 빌드하기 위한 가장 빠른 도구를 제공합니다.

최고 수준의 코드 편집, 디버깅, 성능 도구, 유연한 빌드 시스템 및 인스턴트 빌드/배포 시스템을 사용하여 뛰어난 품질의 앱을 빌드하는 데 집중할 수 있습니다.

ANDROID STUDIO 다운로드
2.1.3.0 FOR WINDOWS (1262 MB)





Please wait while Setup is loading...

verifying installer: 39%



사용자 계정 컨트롤



다음 프로그램이 이 컴퓨터를 변경할 수 있도록 허용
하시겠습니까?



프로그램 이름: android-studio-bundle-143.3101438-windo...

확인된 게시자: **Google Inc**

파일 원본: 이 컴퓨터의 하드 드라이브



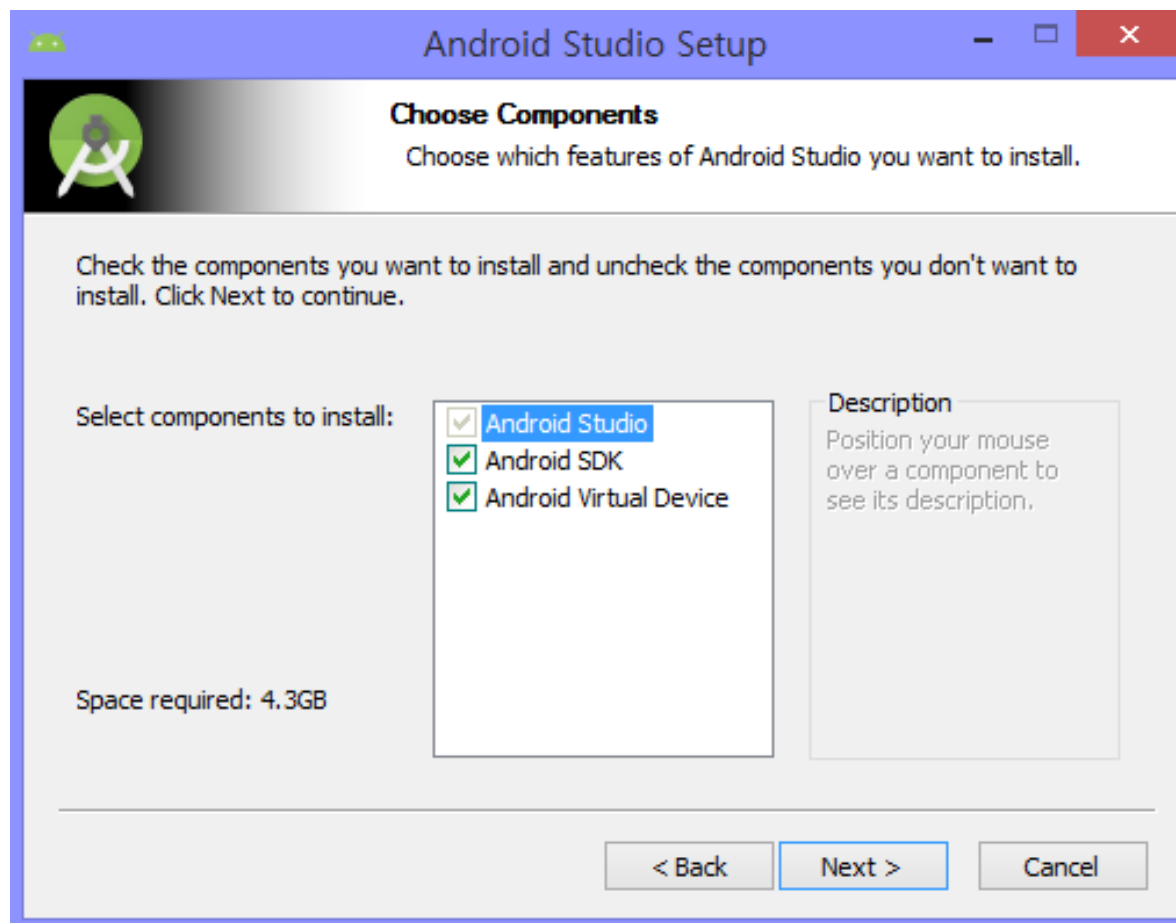
자세한 정보 표시(D)

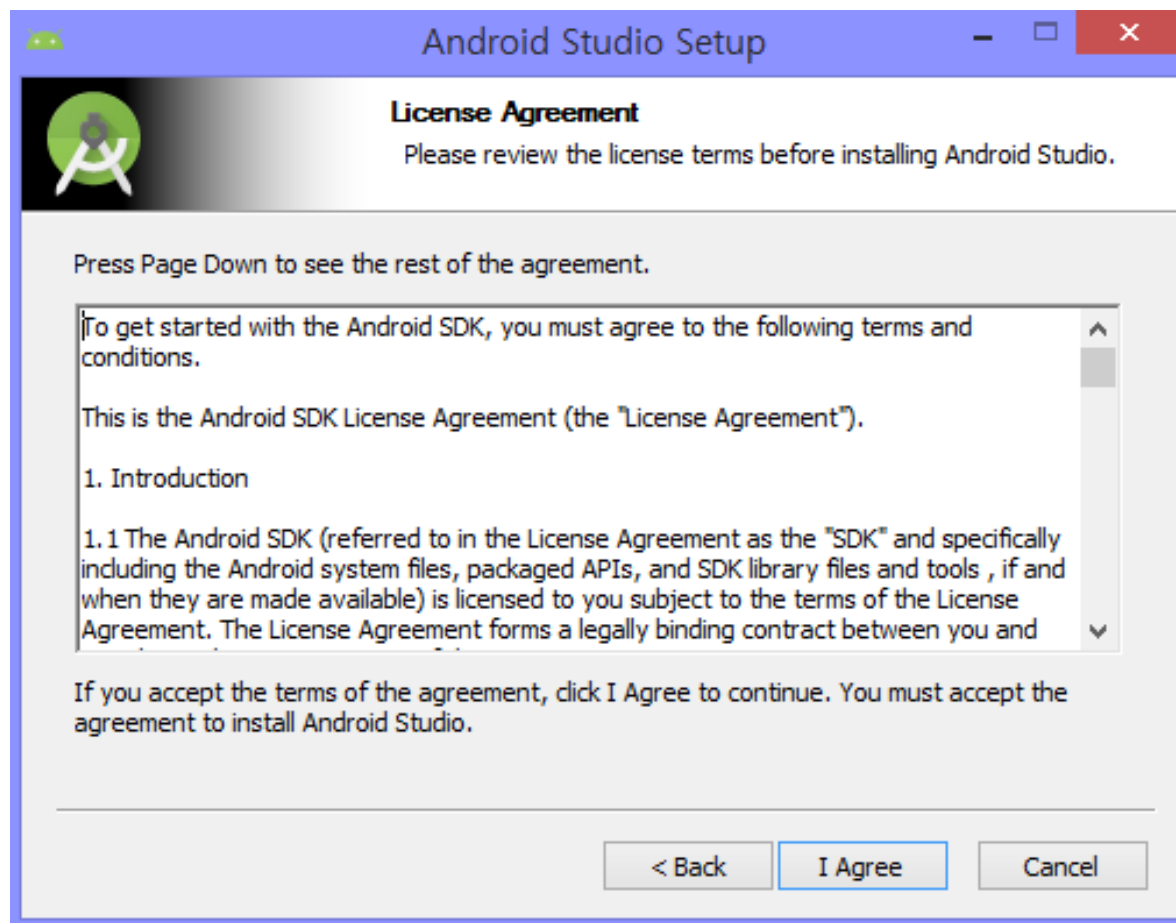
예(Y)

아니요(N)

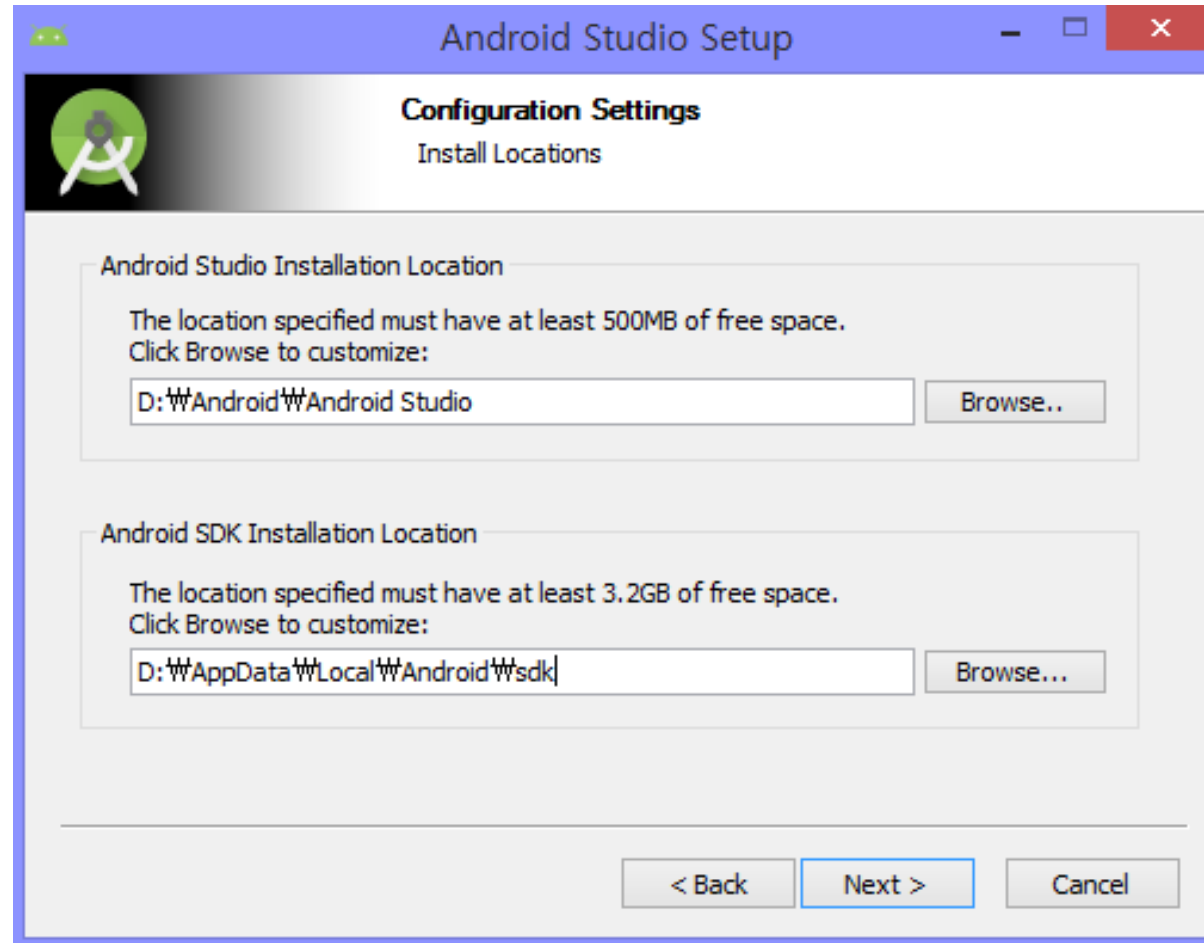
[알림이 표시될 때 변경](#)



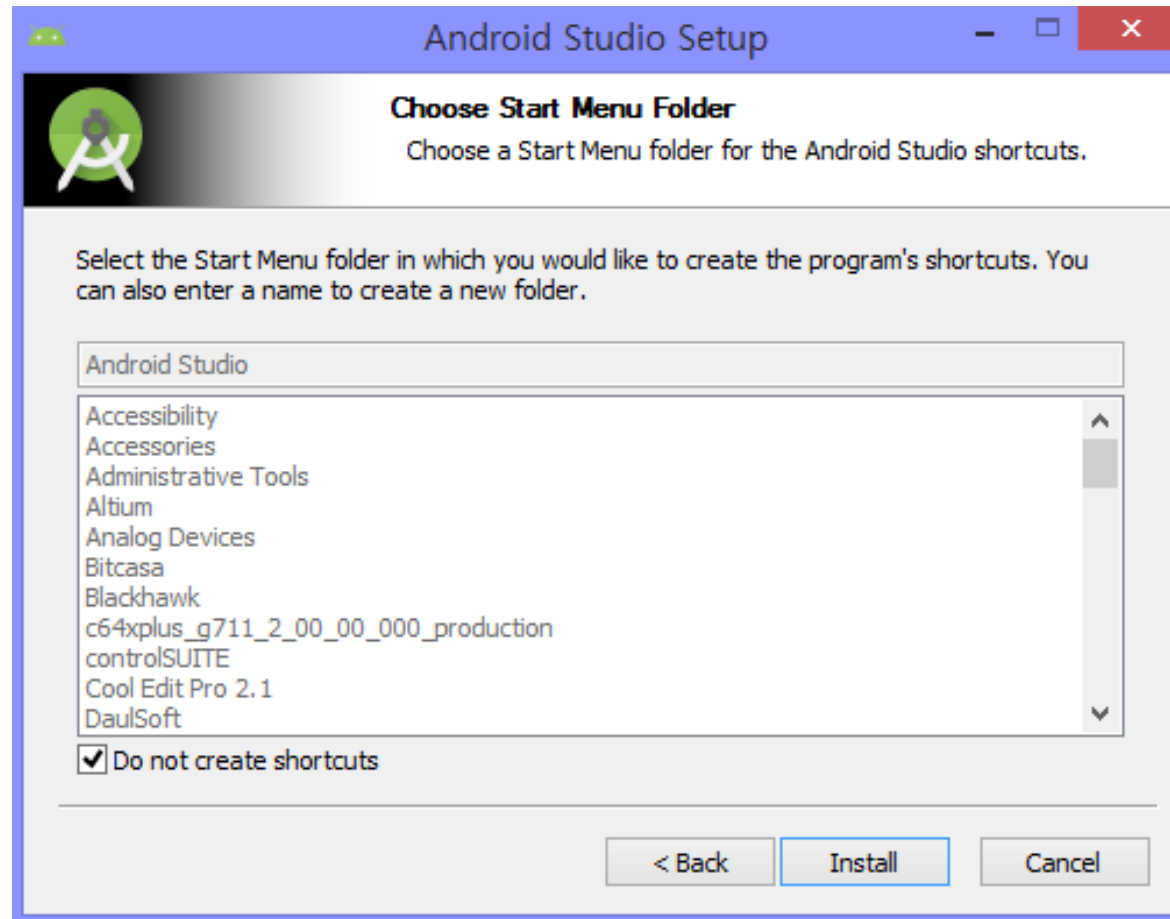


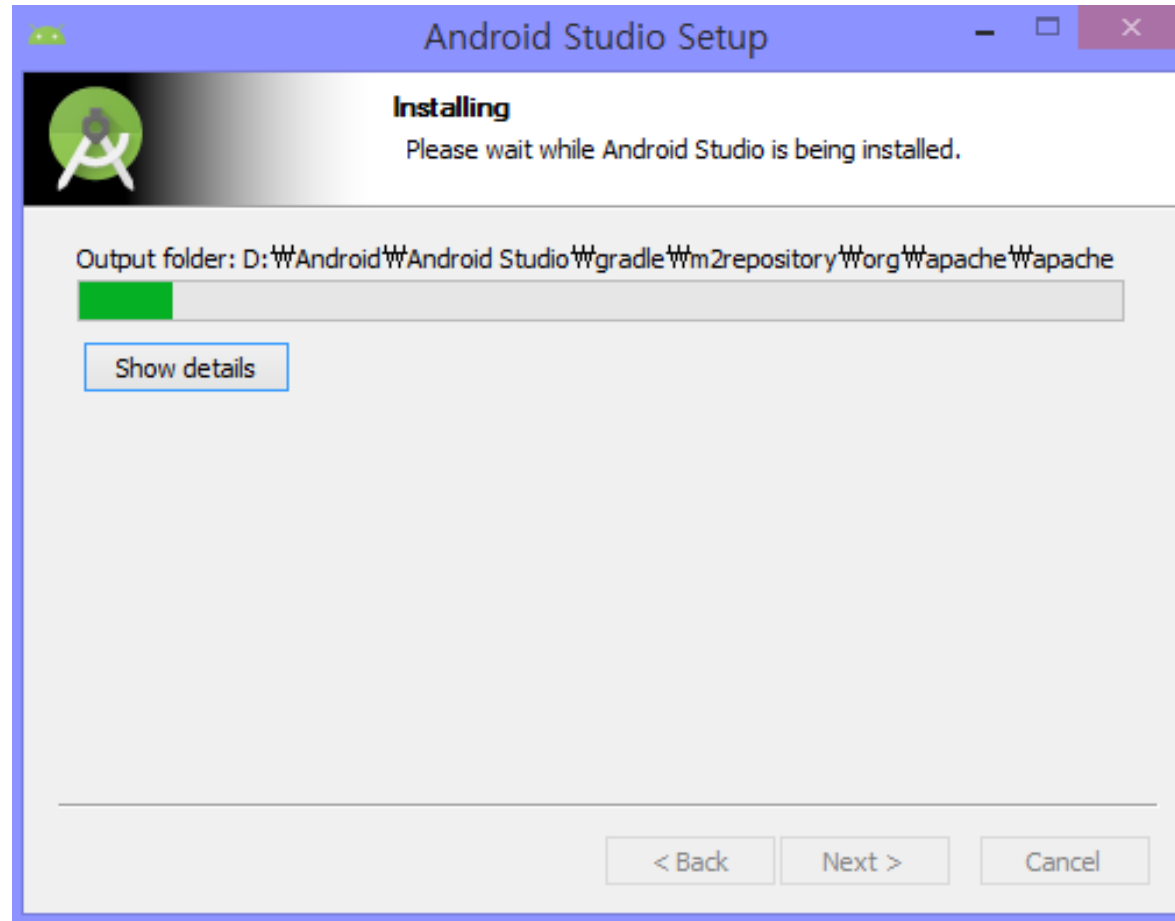


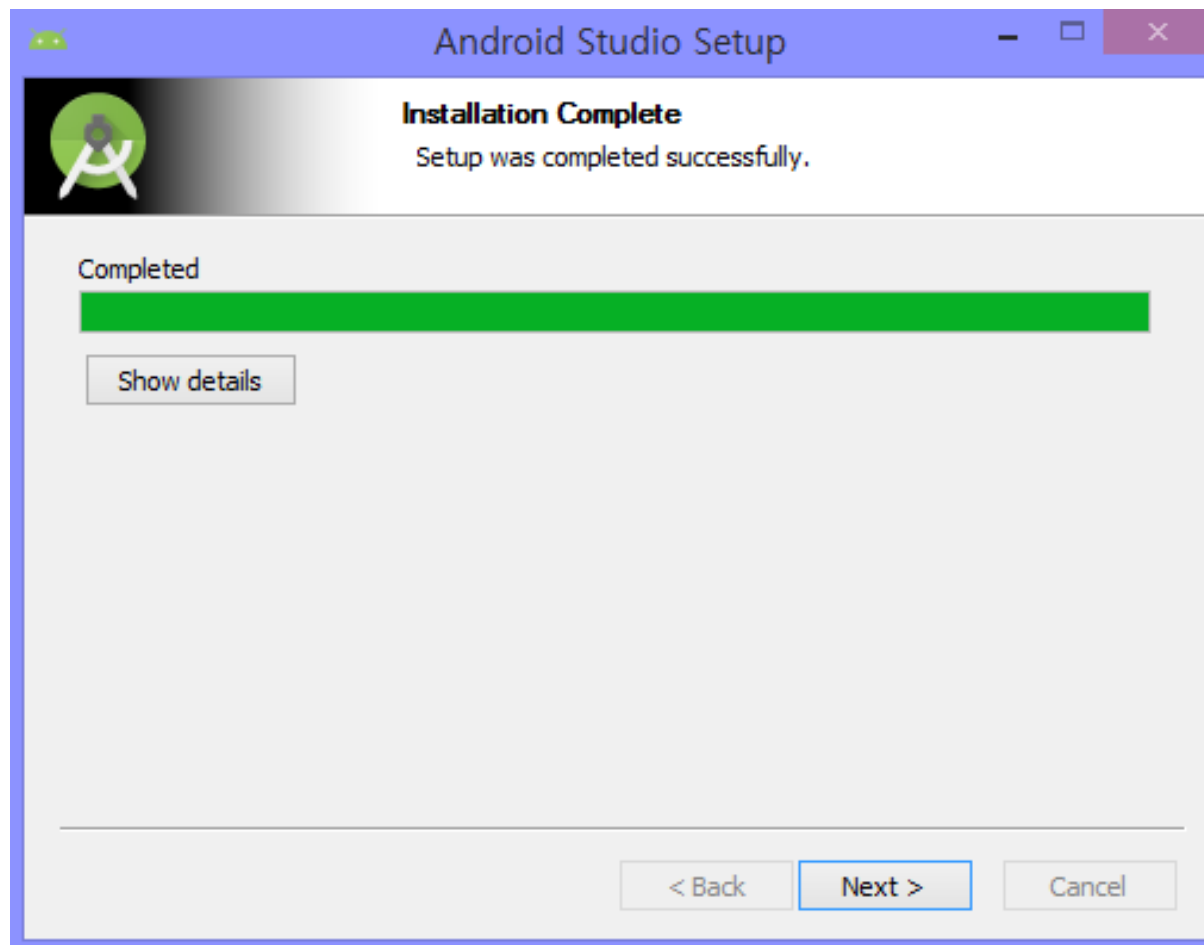
여기서도 반드시 주의할 것이 있는데 지금 D:\Android\Android Studio 라고 하면서 띄어쓰기가 들어가 있다. 한글에 대한 배려가 없고 띄어쓰기가 있을 경우 버그가 존재하여 띄어쓰기를 없애주도록 한다.

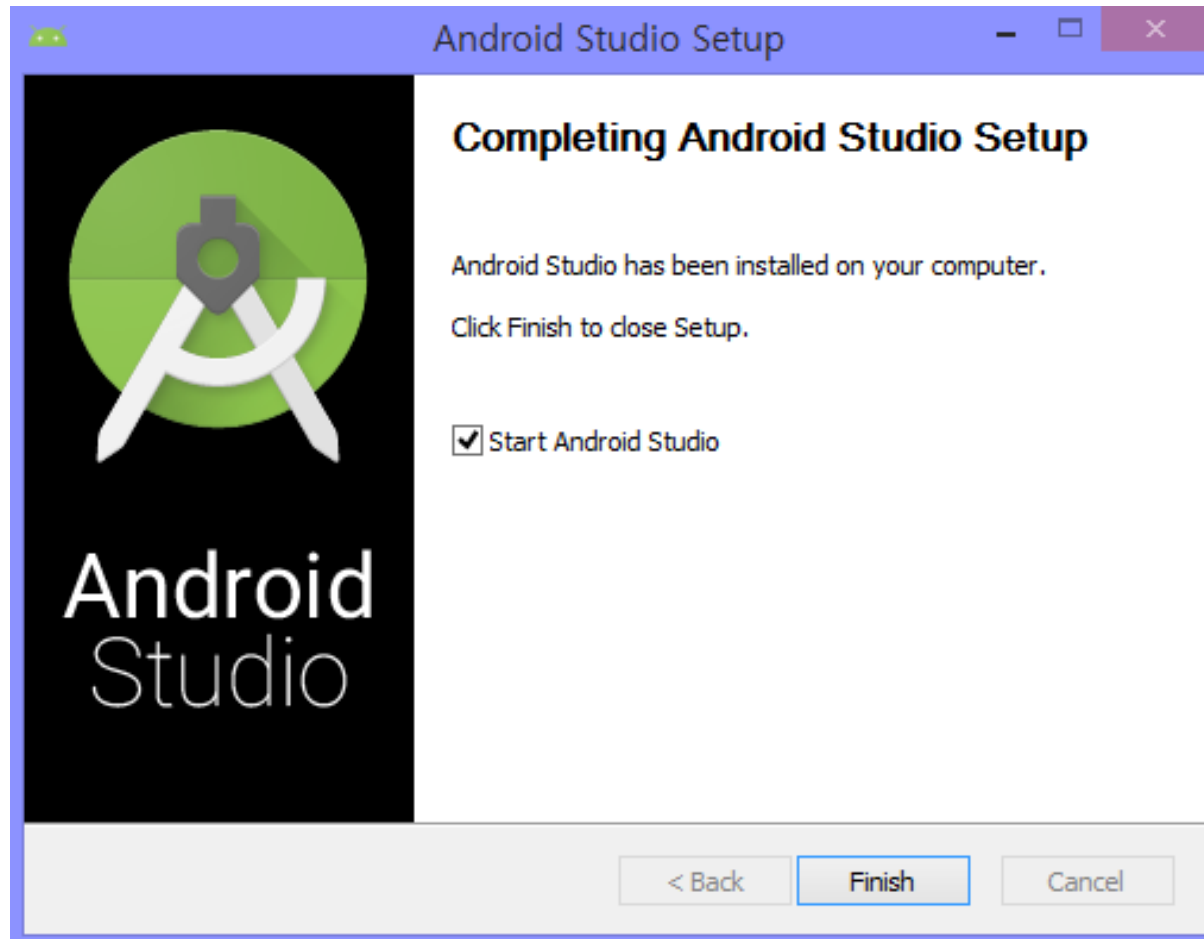


띄어쓰기가 존재한다면 역시 없앤 버전으로 연결하도록 하길 바란다.














 Complete Installation 

You can import your settings from a previous version of Studio.

☐ I want to import my settings from a custom location

Specify config folder or installation home of the previous version of Studio:



☒ I do not have a previous version of Studio or I do not want to import my settings



Android Studio

Powered by the IntelliJ Platform



Welcome

Android Studio

Welcome! This wizard will set up your development environment for Android Studio.
Additionally, the wizard will help port existing Android apps into Android Studio
or create a new Android application project.



Previous

Next

Cancel

Finish



Install Type

Choose the type of setup you want for Android Studio:

☒ Standard

Android Studio will be installed with the most common settings and options.
Recommended for most users.

☐ Custom

You can customize installation settings and components installed.

Previous

Next

Cancel

Finish



Verify Settings

If you want to review or change any of your installation settings, click Previous.

Current Settings:

Total Download Size:

832 MB

SDK Components to Download:

Android SDK Build-Tools 24.0.2	47.2 MB
Android SDK Platform 24	78.8 MB
Android SDK Platform-Tools 24.0.3	2.86 MB
Android SDK Tools 25.2.2	288 MB
Android Support Repository	276 MB
Google Repository	109 MB
Intel x86 Emulator Accelerator (HAXM installer)	1.73 MB
Sources for Android 24	28.9 MB

Previous

Next

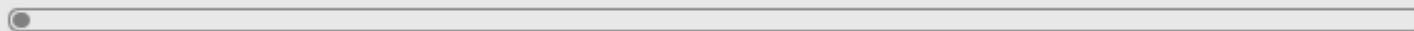
Cancel

Finish



Downloading Components

Downloading android-sdk_r22.6.2-windows.zip (7.32 MB/s)



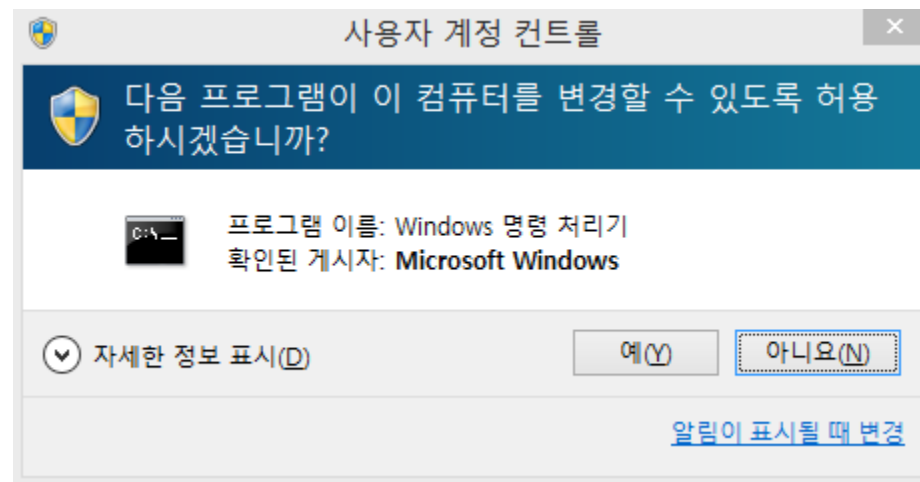
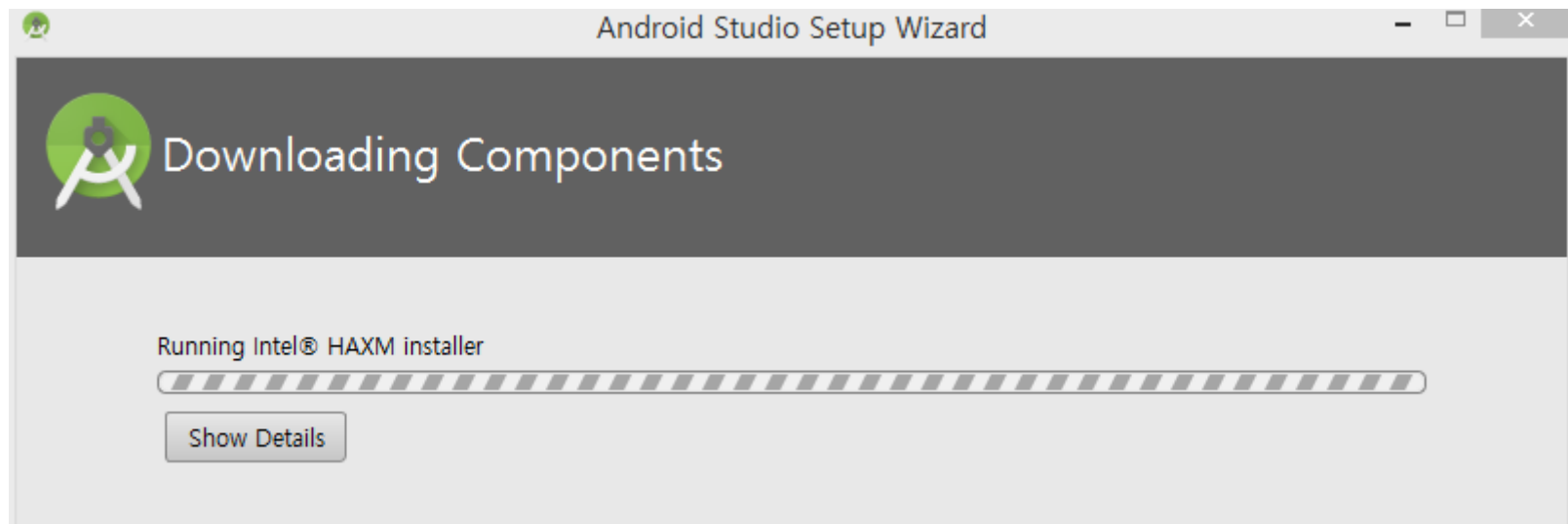
Show Details

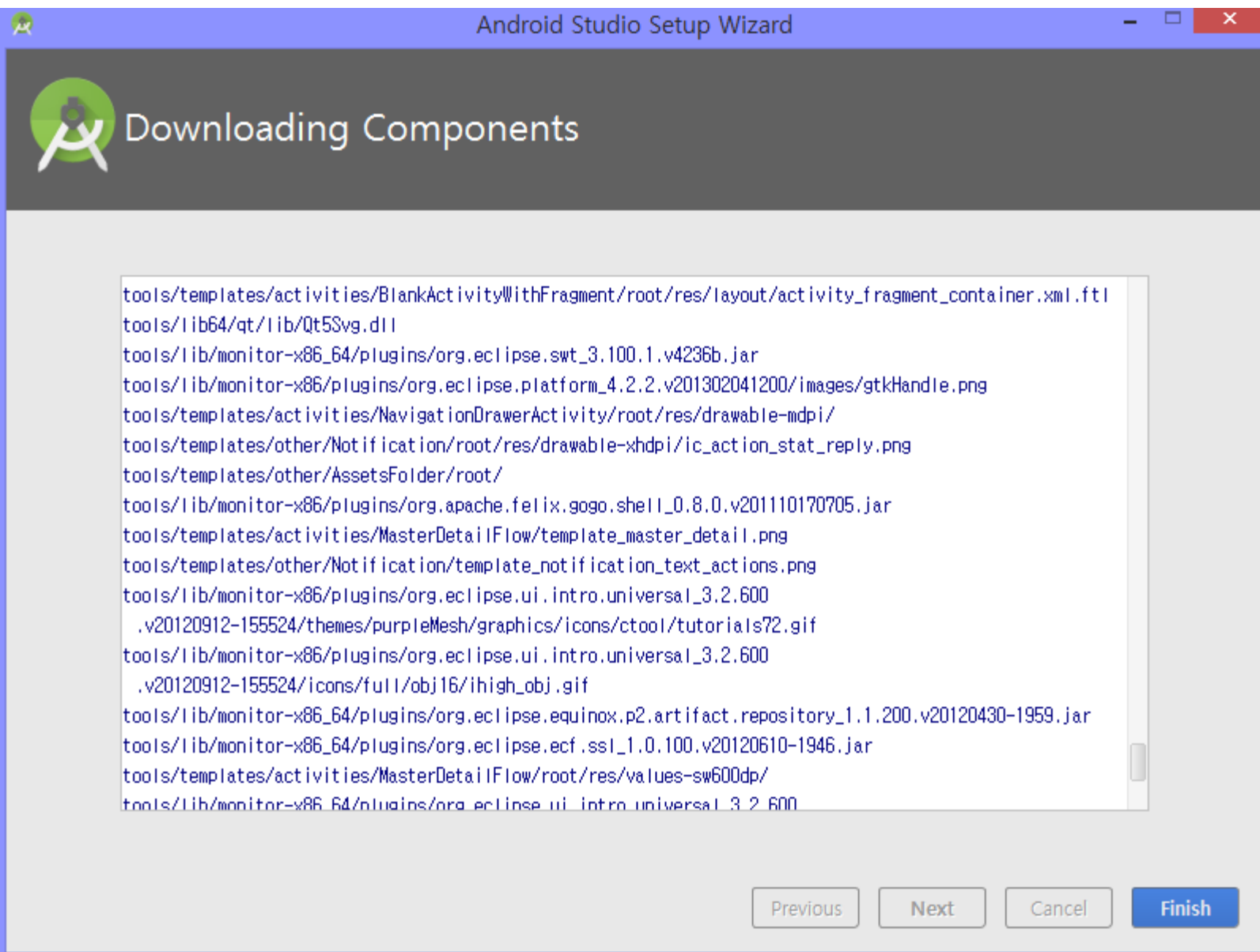
Previous

Next

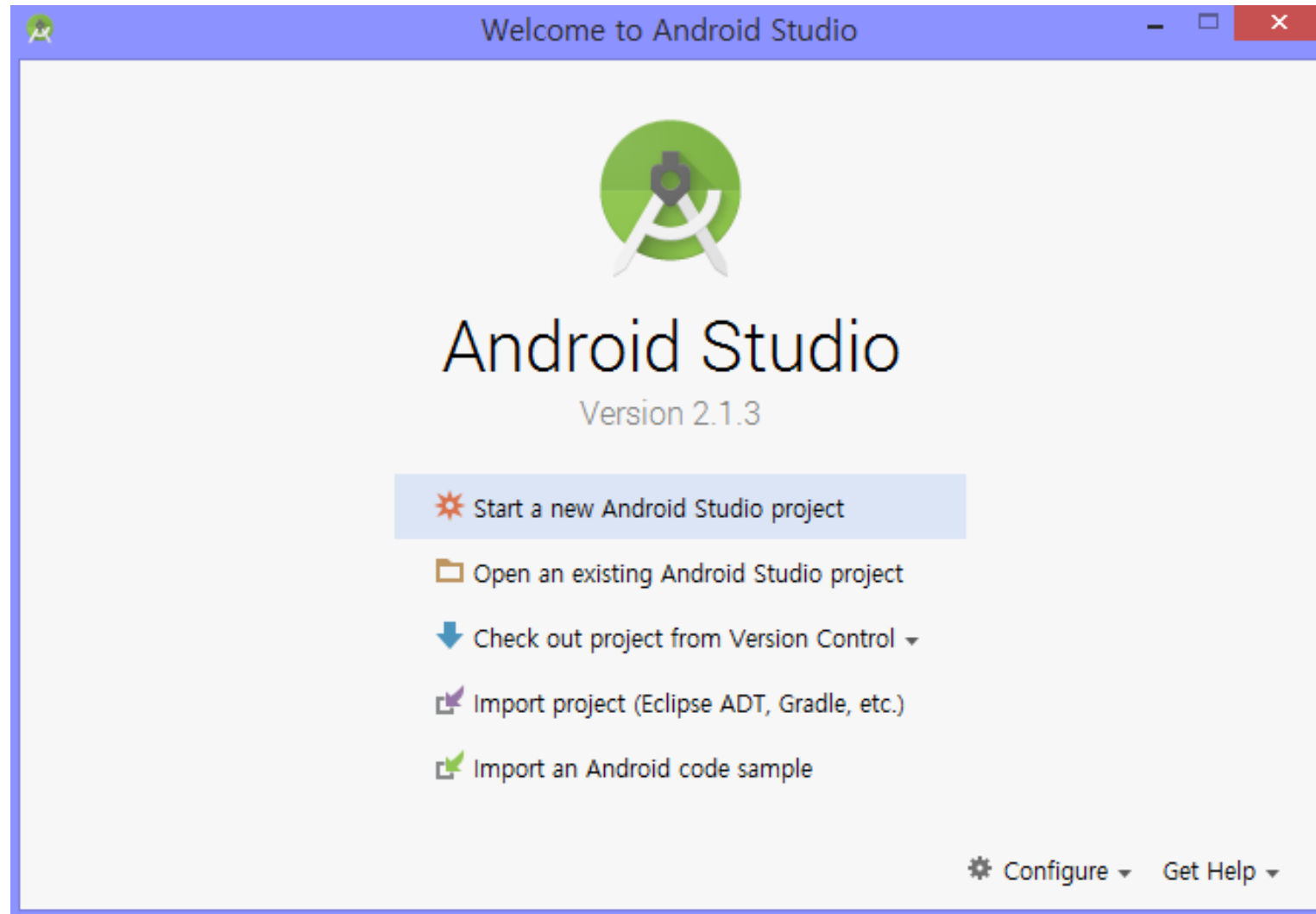
Cancel

Finish

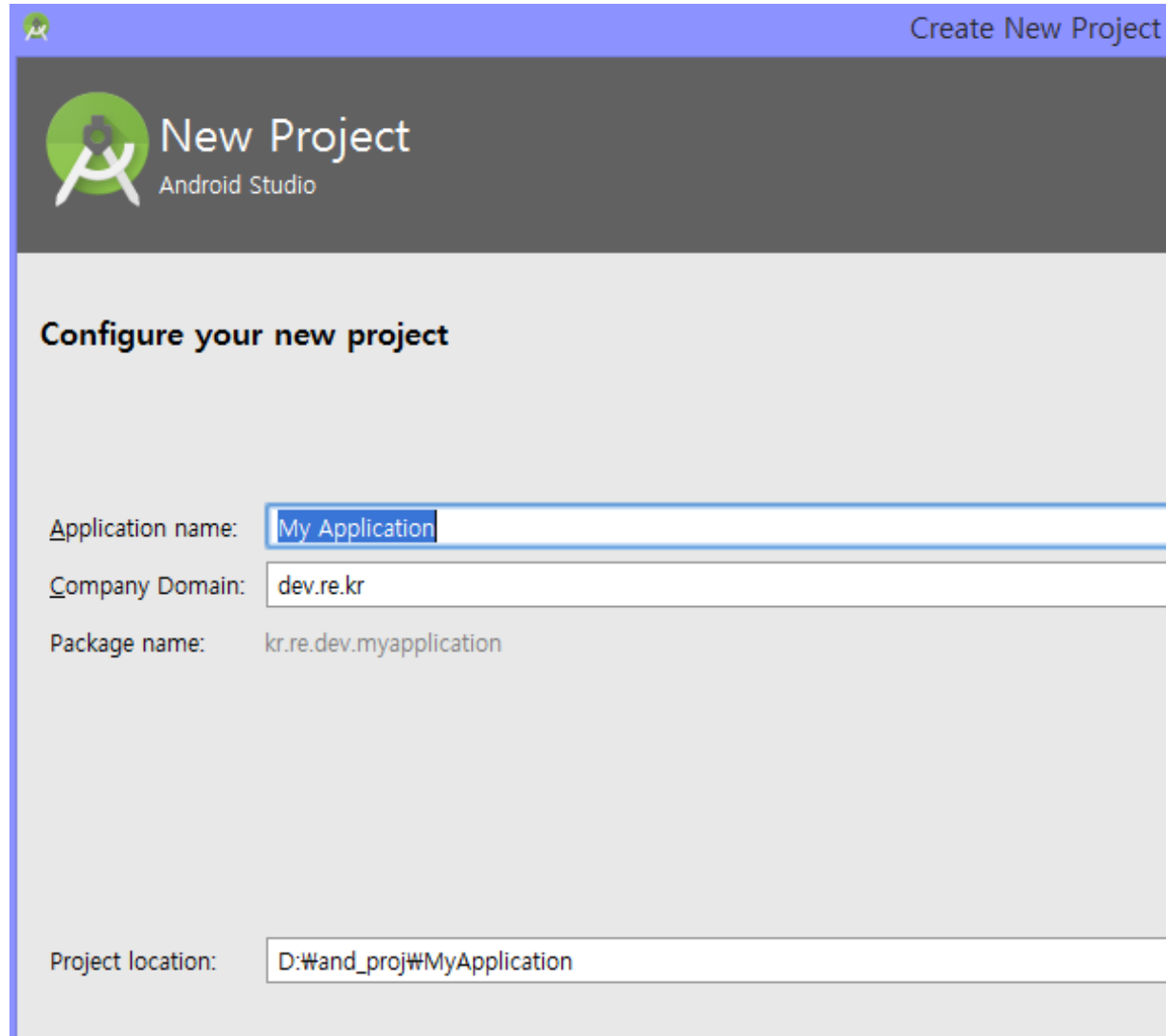




이제 Start a new Android Studio project 를 누르도록 한다.



우선 기본적으로 Android SDK 자체가 잘 구동되는지 확인할 필요가 있으므로 디폴트 값으로 전부 ok ok 하도록 한다.



The screenshot shows the 'Create New Project' dialog in Android Studio. The dialog has a blue header bar with the text 'Create New Project' on the right. Below the header, there is a dark gray section with the Android Studio logo and the text 'New Project' and 'Android Studio'. The main area is light gray and contains the text 'Configure your new project'. Below this, there are four input fields: 'Application name' with the value 'My Application', 'Company Domain' with the value 'dev.re.kr', 'Package name' with the value 'kr.re.dev.myapplication', and 'Project location' with the value 'D:\wand_proj\MyApplication'.

Create New Project

New Project
Android Studio

Configure your new project

Application name: My Application


Company Domain: dev.re.kr

Package name: kr.re.dev.myapplication

Project location: D:\wand_proj\MyApplication

역시 디폴트 값으로 잡고 넘어가도록 한다.

Create New Project

 Target Android Devices

Select the form factors your app will run on

Different platforms may require separate SDKs

☒ Phone and Tablet

Minimum SDK

Lower API levels target more devices, but have fewer features available.
By targeting API 16 and later, your app will run on approximately **95.2%** of the devices that are active on the Google Play Store.
[Help me choose](#)

☐ Wear

Minimum SDK

☐ TV

Minimum SDK

☐ Android Auto

☐ Glass

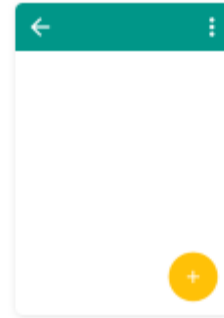
Minimum SDK



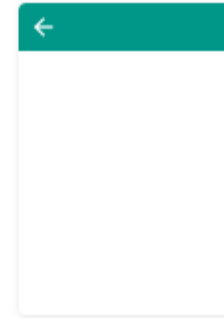
Add an Activity to Mobile



Add No Activity



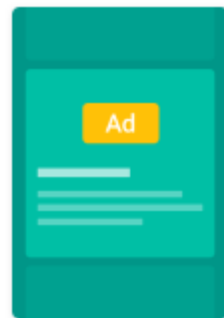
Basic Activity



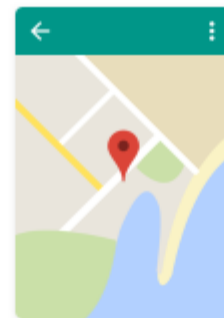
Empty Activity



Fullscreen Activity



Google AdMob Ads Activity



Google Maps Activity



Login Activity



Master/Detail Flow

Previous

Next

Cancel

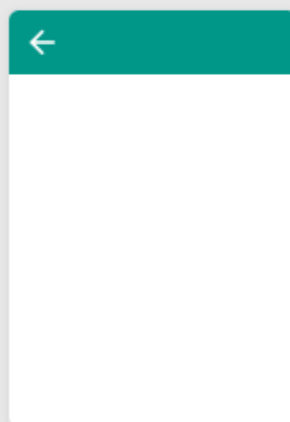
Finish



Customize the Activity



Creates a new empty activity



Empty Activity

Activity Name: MainActivity

☒ Generate Layout File

Layout Name: activity_main

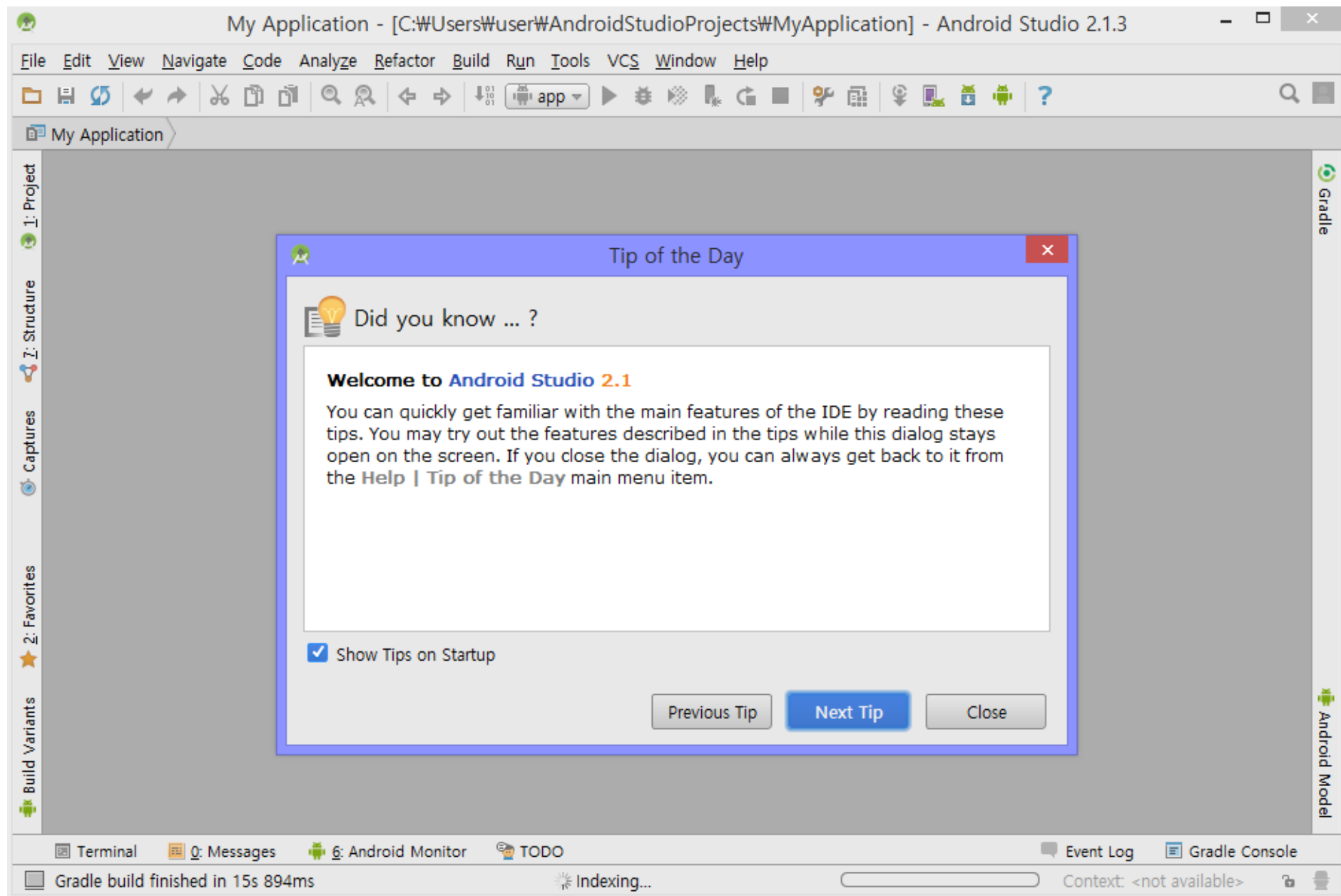
The name of the activity class to create

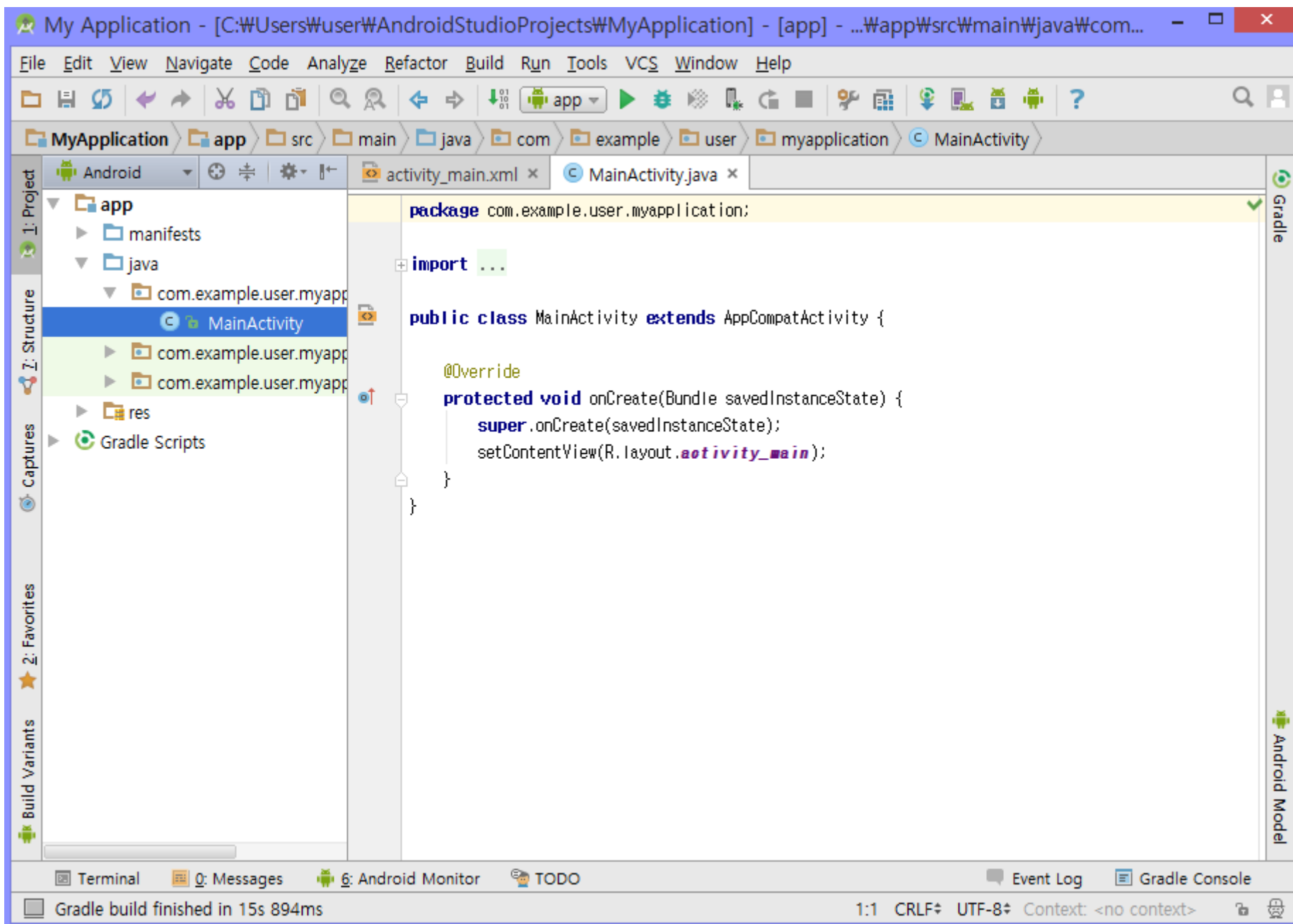
Previous

Next

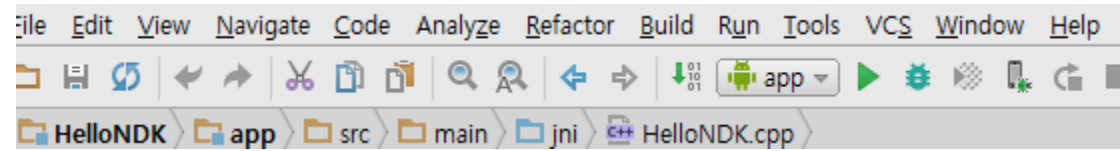
Cancel

Finish

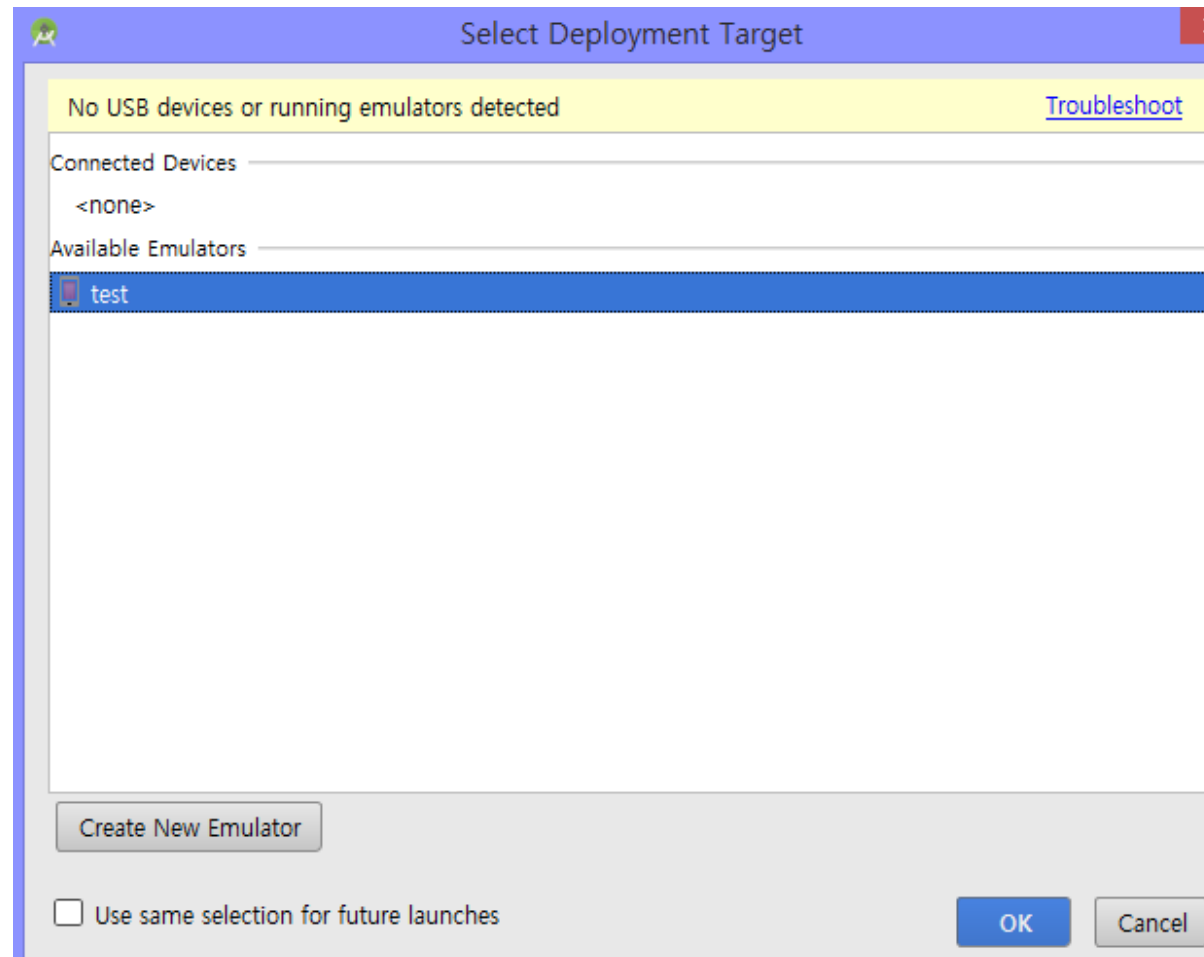




우선 초기 형태를 가지고 바로 위쪽에 플레이 버튼이 보일텐데 이것을 눌러서 구동을 확인해보도록 한다.



그러면 아래와 같은 창이 나타날텐데 Create New Emulator 를 클릭하도록 한다.







그러면 화면 크기등이 적당하게 나오게 되는데 적절한 크기를 선택해주면된다.
 (본인 컴퓨터에 무리가 가지 않을 적당한 선에서 선택하는 것이 좋으며 컴퓨터가 좋다면 무관하다)

Phone				
Tablet	Nexus 5X	5.2"	1080x1920	420dpi
	Nexus 5	4.95"	1080x1920	xxhdpi
	Nexus 4	4.7"	768x1280	xhdpi
	Galaxy Nexus	4.65"	720x1280	xhdpi
	5.4" FWVGA	5.4"	480x854	mdpi
	5.1" WVGA	5.1"	480x800	mdpi
	4.7" WXGA	4.7"	720x1280	xhdpi
	4.65" 720p (Galaxy Nexus)	4.65"	720x1280	xhdpi
	4" WVGA (Nexus S)	4.0"	480x800	hdpi
	3.7" WVGA (Nexus One)	3.4"	480x800	hdpi
	3.7" FWVGA slider	3.7"	480x854	hdpi
	3.4" WQVGA	3.4"	240x432	ldpi
	3.3" WQVGA	3.3"	240x400	ldpi
	3.2" QVGA (ADP2)	3.2"	320x480	mdpi
	3.2" HVGA slider (ADP1)	3.2"	320x480	mdpi





다음으로 OS 버전이라고 할 수 있는 부분을 설정하면 되는데
 본인 휴대폰의 버전에 맞게 해도 되고 최신 버전으로 테스트해봐도 된다.
 무난하게 Jelly Bean(하위 호환) 혹은 Nougat 이나 Marshmallow 등(최신)을 선택하는 것이 좋다.

Recommended x86 Images Other Images			
Release Name	API Level ▼	ABI	Target
<i>null</i> Download	25	<i>x86_64</i>	<i>Android null (with Google APIs)</i>
<i>null</i> Download	25	<i>x86</i>	<i>Android null (with Google APIs)</i>
<i>Nougat</i> Download	24	<i>x86_64</i>	<i>Android 7.0 (with Google APIs)</i>
<i>Nougat</i> Download	24	<i>x86</i>	<i>Android 7.0 (with Google APIs)</i>
Marshmallow	23	x86	Android 6.0 (with Google APIs)
<i>Marshmallow</i> Download	23	<i>x86_64</i>	<i>Android 6.0 (with Google APIs)</i>
<i>Lollipop</i> Download	22	<i>x86</i>	<i>Android 5.1 (with Google APIs)</i>
<i>Lollipop</i> Download	22	<i>x86_64</i>	<i>Android 5.1 (with Google APIs)</i>
<i>KitKat</i> Download	19	<i>x86</i>	<i>Android 4.4 (with Google APIs)</i>
<i>Jelly Bean</i> Download	18	<i>x86</i>	<i>Android 4.3 (with Google APIs)</i>
<i>Jelly Bean</i> Download	17	<i>x86</i>	<i>Android 4.2 (with Google APIs)</i>
Jelly Bean	16	x86	Android 4.1 (with Google APIs)
<i>Gingerbread</i> Download	10	<i>x86</i>	<i>Android 2.3.3 (with Google APIs)</i>

아래와 같이 적당한 이름을 지어주고 Finish 하면 가상 머신이 만들어진다.

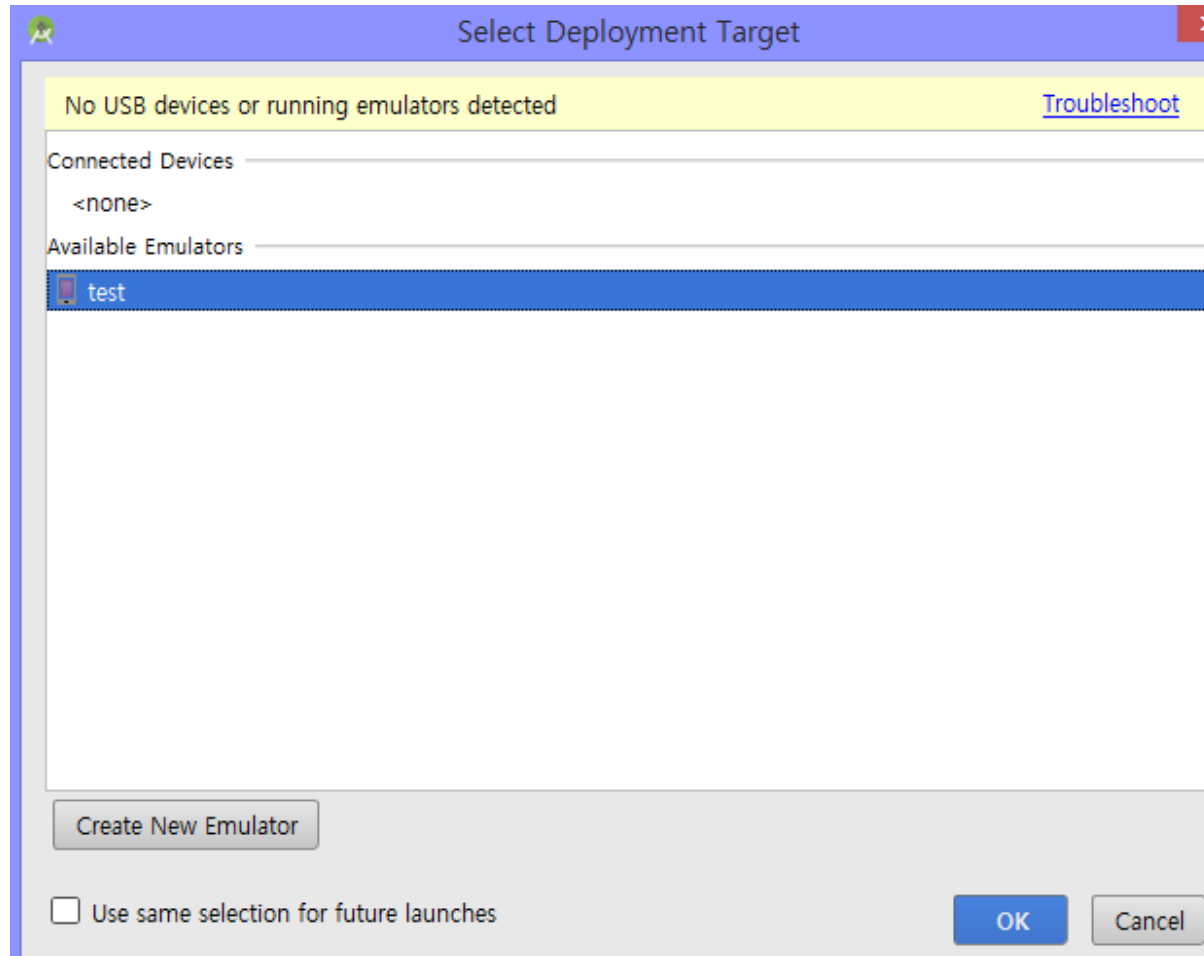
AVD Name	<input type="text" value="QVGA (ADP2) API 23"/>		
 3.2" QVGA (ADP2)	3.2" 320x480 mdpi	<button>Change...</button>	
 Marshmallow	Android 6.0 x86	<button>Change...</button>	
Startup size and orientation	Scale:	<input type="text" value="Auto"/> ▼	
	Orientation:	<div><div> Portrait</div><div> Landscape</div></div>	
Emulated Performance	Graphics:	<input type="text" value="Auto"/> ▼	
Device Frame	<input type="checkbox"/> Enable Device Frame		

아래와 같이 적당한 이름을 지어주고 Finish 하면 가상 머신이 만들어진다.
이 이름을 test 로 만들면 test 가 나오게 되며
실제 휴대폰을 연결할 경우 Connected Devices 에 당사자의 휴대폰이 잡히게 된다.
(USB 를 활용하여 휴대폰과 PC 를 연결하면 됨)

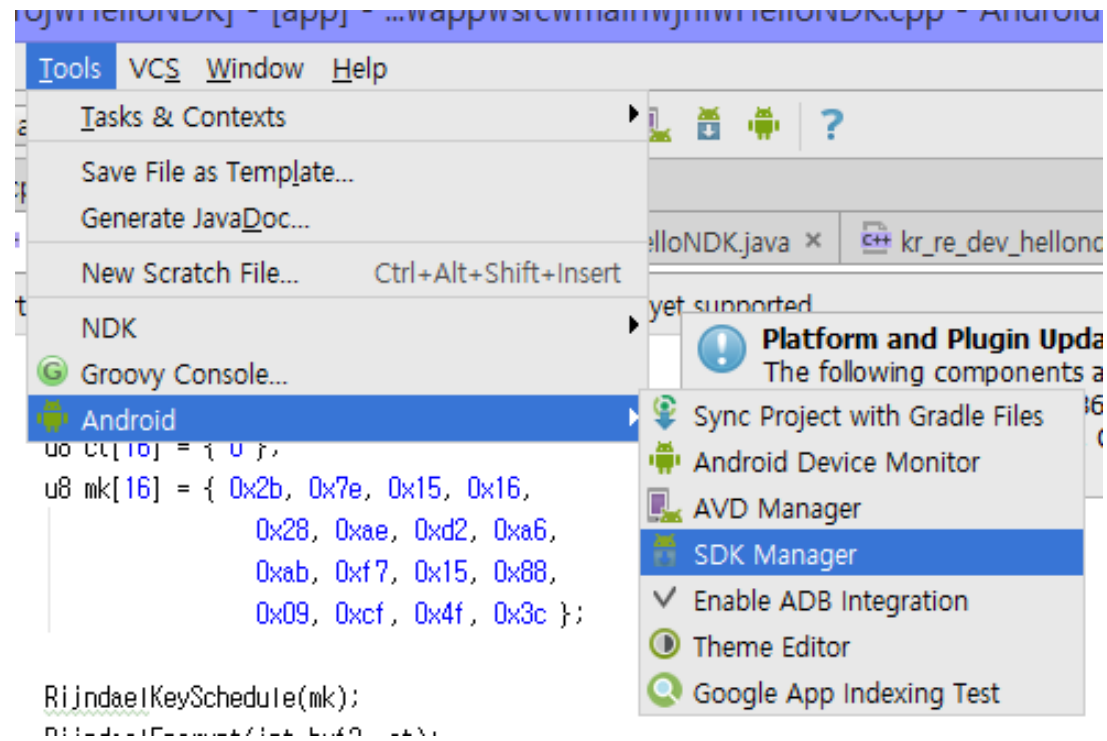
AVD Name	<input type="text" value="QVGA (ADP2) API 23"/>		
 3.2" QVGA (ADP2)	3.2" 320x480 mdpi	<button>Change...</button>	
 Marshmallow	Android 6.0 x86	<button>Change...</button>	
Startup size and orientation	Scale:	<input type="text" value="Auto"/> ▼	
	Orientation:	<div><div> Portrait</div><div> Landscape</div></div>	
Emulated Performance	Graphics:	<input type="text" value="Auto"/> ▼	
Device Frame	<input type="checkbox"/> Enable Device Frame		

아무튼 다시 플레이 버튼을 눌러보면 아래와 같은 창이 나타날 것이다.
맨 처음에는 Available Emulators 가 아무것도 없었겠지만
현재는 우리가 추가했기 때문에 나타날 것이다.
이를 선택하고 OK 누르면 해당 가상 머신에서 안드로이드가 구동하는 것을 볼 수 있을 것이다.

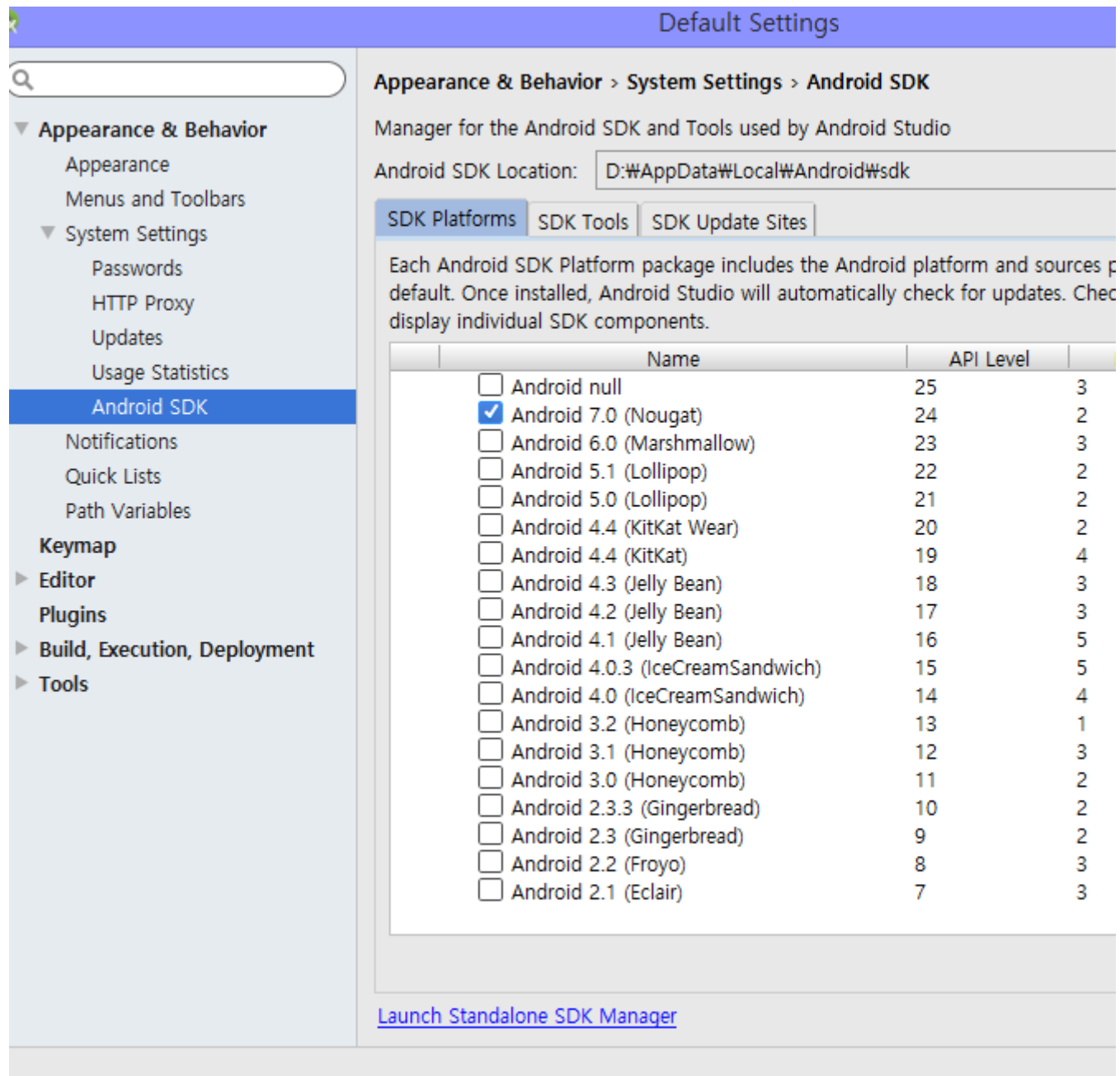
이때 주의할 것은 Instant Run 을 설치하라고 나오는데 Install and Continue 를 누른다.
그리고 반드시 Done 이 나오기 전까지 기다렸다가 나오면 Finish 를 누르고 실행하도록 한다.
정상적으로 잘 수행되었다면 이제 다음으로 NDK 를 설정하면 되겠다.



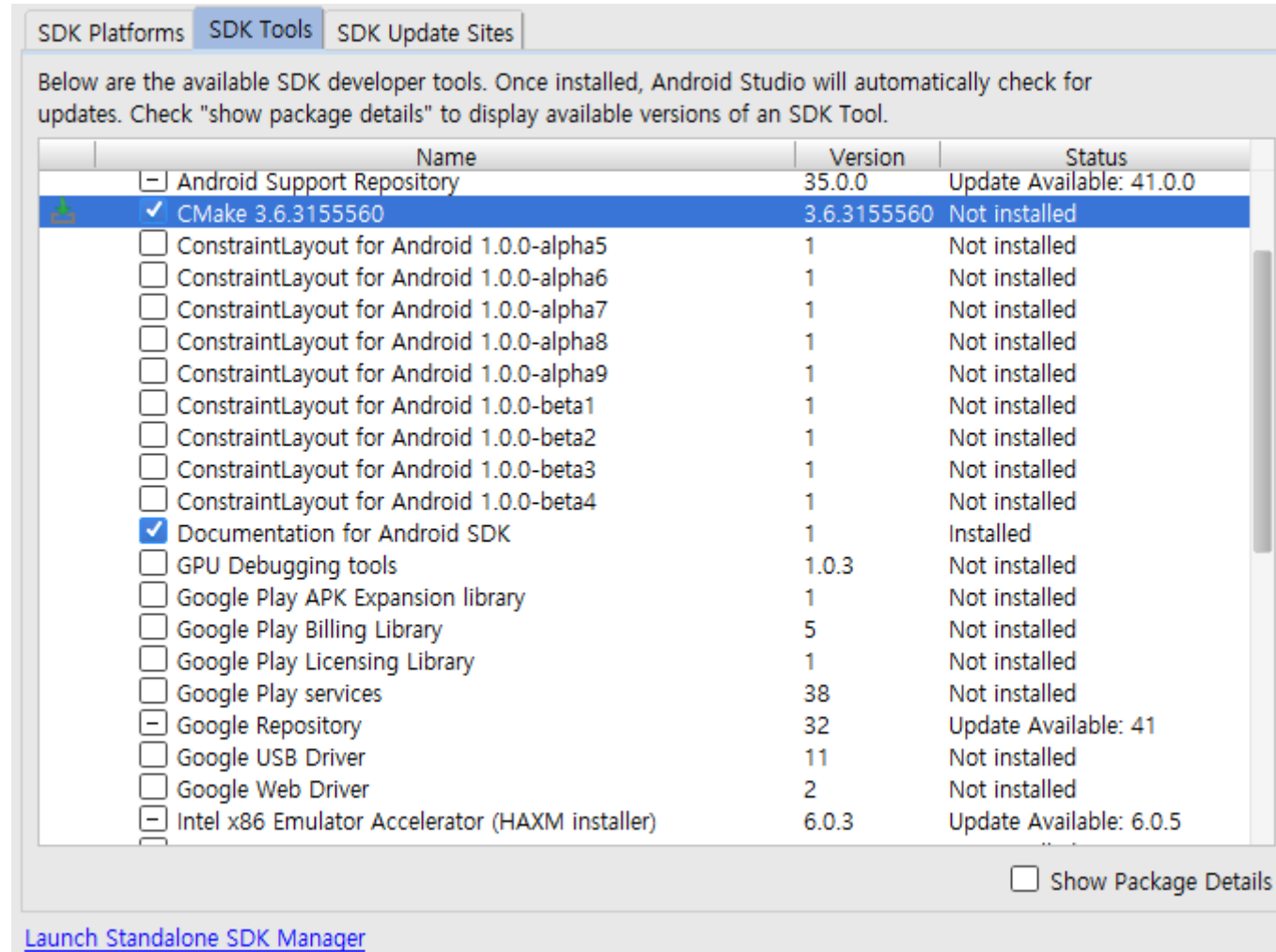
이제 최 상단의 Tools -> Android -> SDK Manager 를 누르도록 한다.



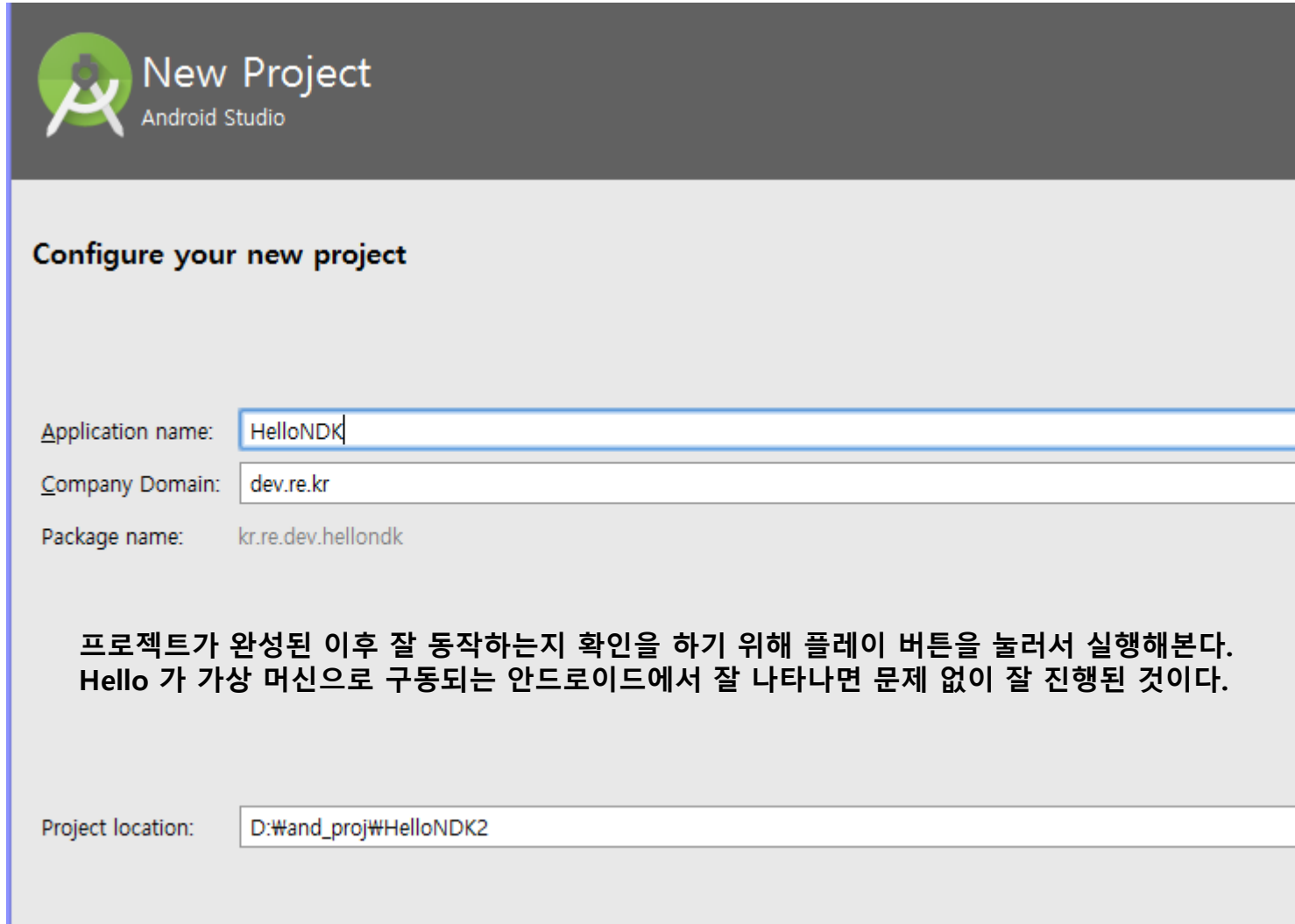
그러면 아래와 같은 화면이 나타날 것이다.




여기서 SDK Tools 로 가서 NDK, LLDB, CMake 를 선택해주도록 한다.
Apply 를 누르면 설치가 시작되는데 역시 반드시 Done 이 나올때까지 기다린 후 Finish 를 해줘야만 한다.



이제 기본적인 조건이 모두 갖춰졌으므로 NDK 프로젝트를 작성해보도록 하자 ~~!!~!
필자의 경우 Android Studio 의 버전이 낮아서 나오지는 않지만
현재 최신 버전을 설치한 경우에는 여기에 Include C++ Support 부분이 존재한다.
Include C++ Support 를 선택하고 Project Location 을 공백 문자, 한글 없이 작성한다.
이후는 그냥 전부다 Next Next Next 하면 된다.



 **New Project**
Android Studio

Configure your new project

Application name:

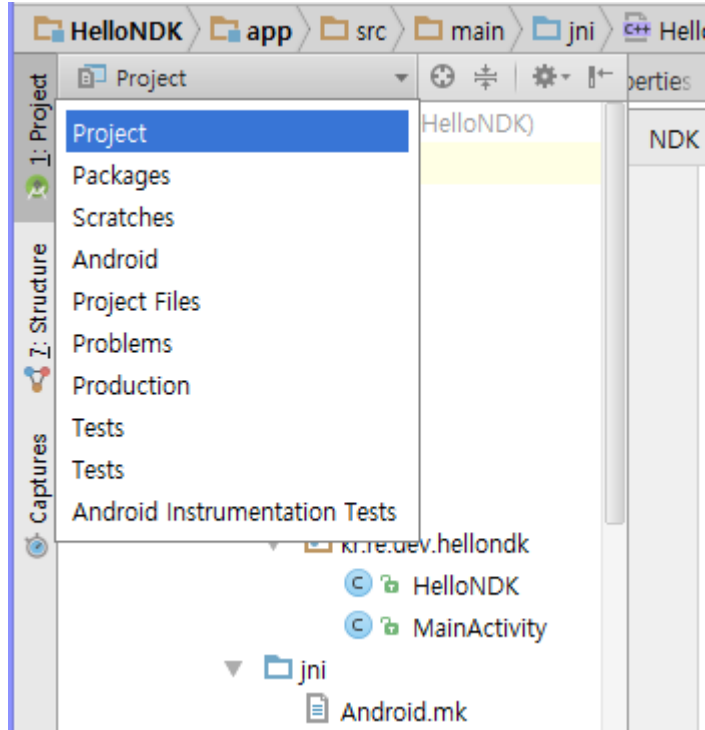
Company Domain:

Package name:

프로젝트가 완성된 이후 잘 동작하는지 확인을 하기 위해 플레이 버튼을 눌러서 실행해본다.
Hello 가 가상 머신으로 구동되는 안드로이드에서 잘 나타나면 문제 없이 잘 진행된 것이다.

Project location:

다음으로 Project View 를 선택하고 HelloNDK -> app -> src -> main -> java 를 우클릭하여 HelloNDK.java 를 만든다.



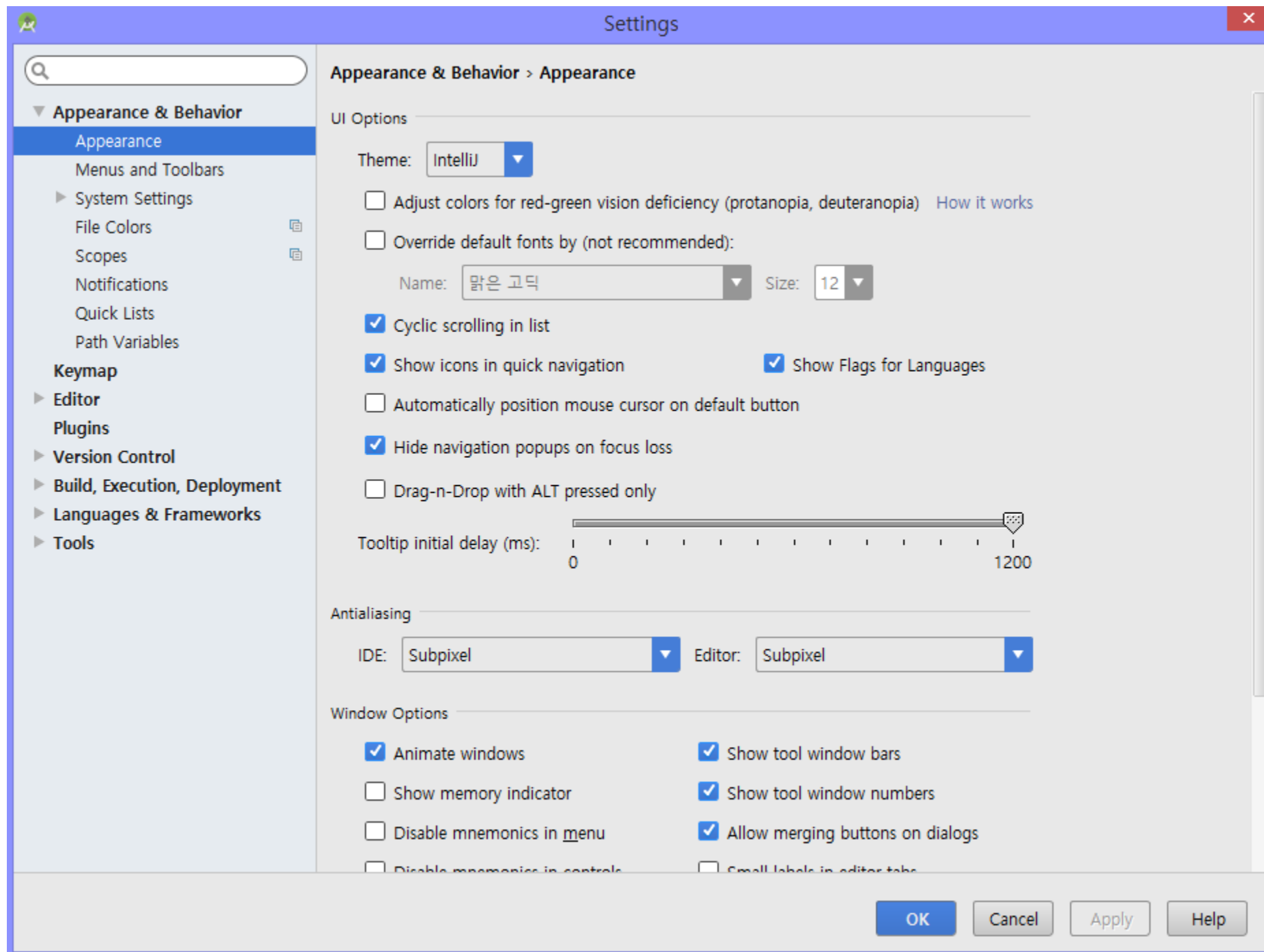
Project View 를 선택한다는 것은 저기서 Project 를 선택함을 의미한다.

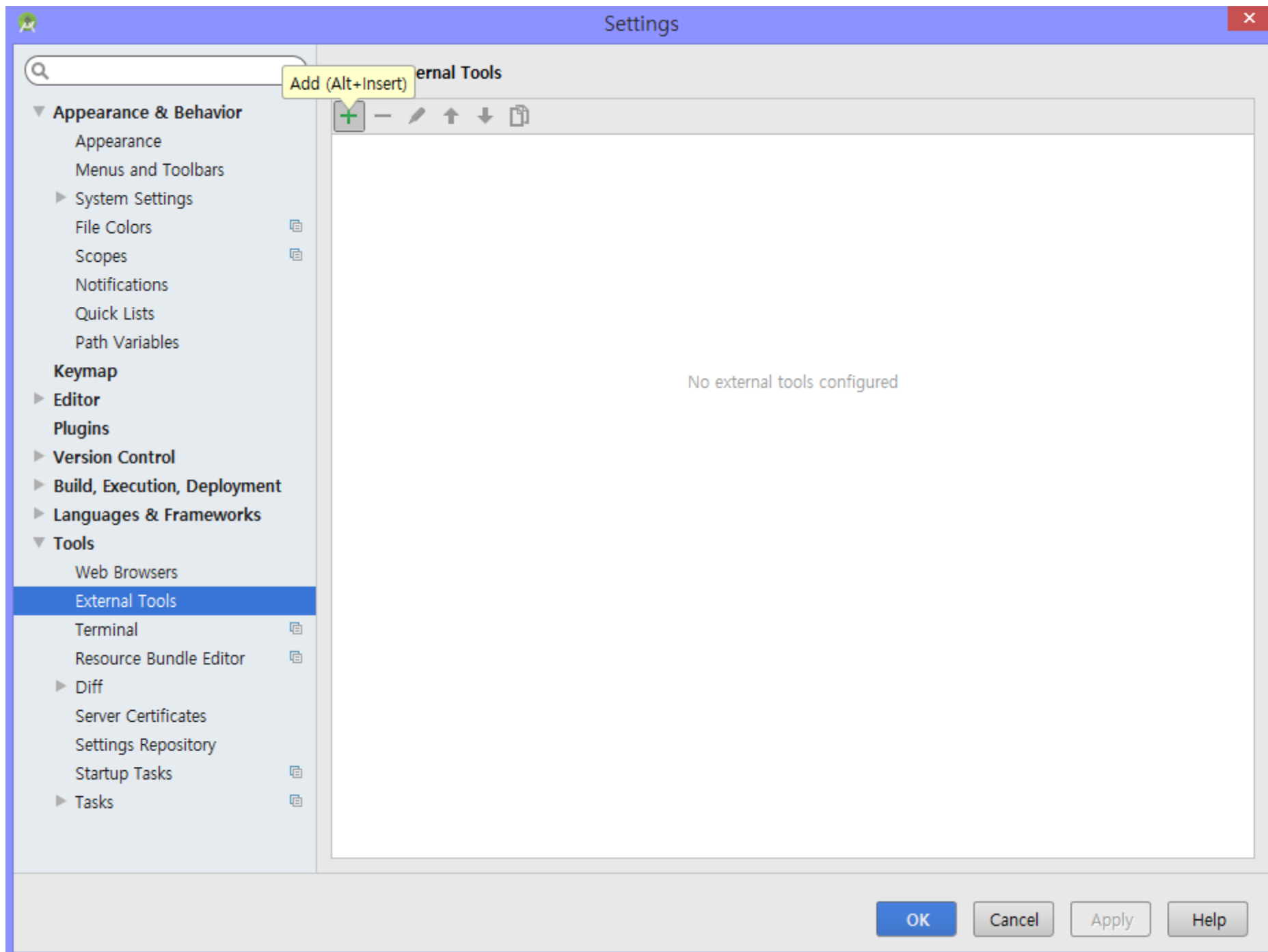
HelloNDK.java 코드는 아래와 같이 작성해준다.
코드를 모두 작성한 이후에 Ctrl + F9 를 눌러서 한 번 Build 해준다.
그려야지만 javah 가 어떤 녀석들을 참조해야하는지 정신을 차린다.



```
package kr.re.dev.hellondk;

/**
 * Created by keh on 2016-10-21.
 */

public class HelloNDK {
    static {
        System.loadLibrary("native-lib");
    }
    public native String stringFromJNI();
}
```





 Create Tool 

Name:

Group:

Description:

Options

☒ Synchronize files after execution

☒ Open console

☐ Show console when a message is printed to standard output stream

☐ Show console when a message is printed to standard error stream

Show in

☒ Main menu

☒ Editor menu

☒ Project views

☒ Search results

Tool settings

Program:

Parameters:

Working directory:

Create Tool

Name:

javah

Group:

NDK

Description:

Create NDK Header File

Options

☒ Synchronize files after execution

☒ Open console

Output Filters...

☐ Show console when a message is printed to standard output stream

☐ Show console when a message is printed to standard error stream

Show in

☒ Main menu

☒ Editor menu

☒ Project views

☒ Search results

Tool settings

Program:

...

Insert macro...

Parameters:

Insert macro...

Working directory:

...

Insert macro...

OK

Cancel

Help

여기에 Program 을 보면 필자는 아래와 같이 했지만
AndroidStudio로 한 경우에 앞서서 AndroidStudio 를 설치한 경로에 jre/bin/javah.exe 를 선택해주도록 한다.

Create Tool

Name: javah Group: NDK

Description: Create NDK Header File

Options

☒ Synchronize files after execution ☒ Open console ☐ Show console when a message is printed to standard output stream ☐ Show console when a message is printed to standard error stream

Show in

☒ Main menu ☒ Editor menu ☒ Project views ☒ Search results

Tool settings

Program: D:\Java\jdk1.8.0_101\bin\javah.exe

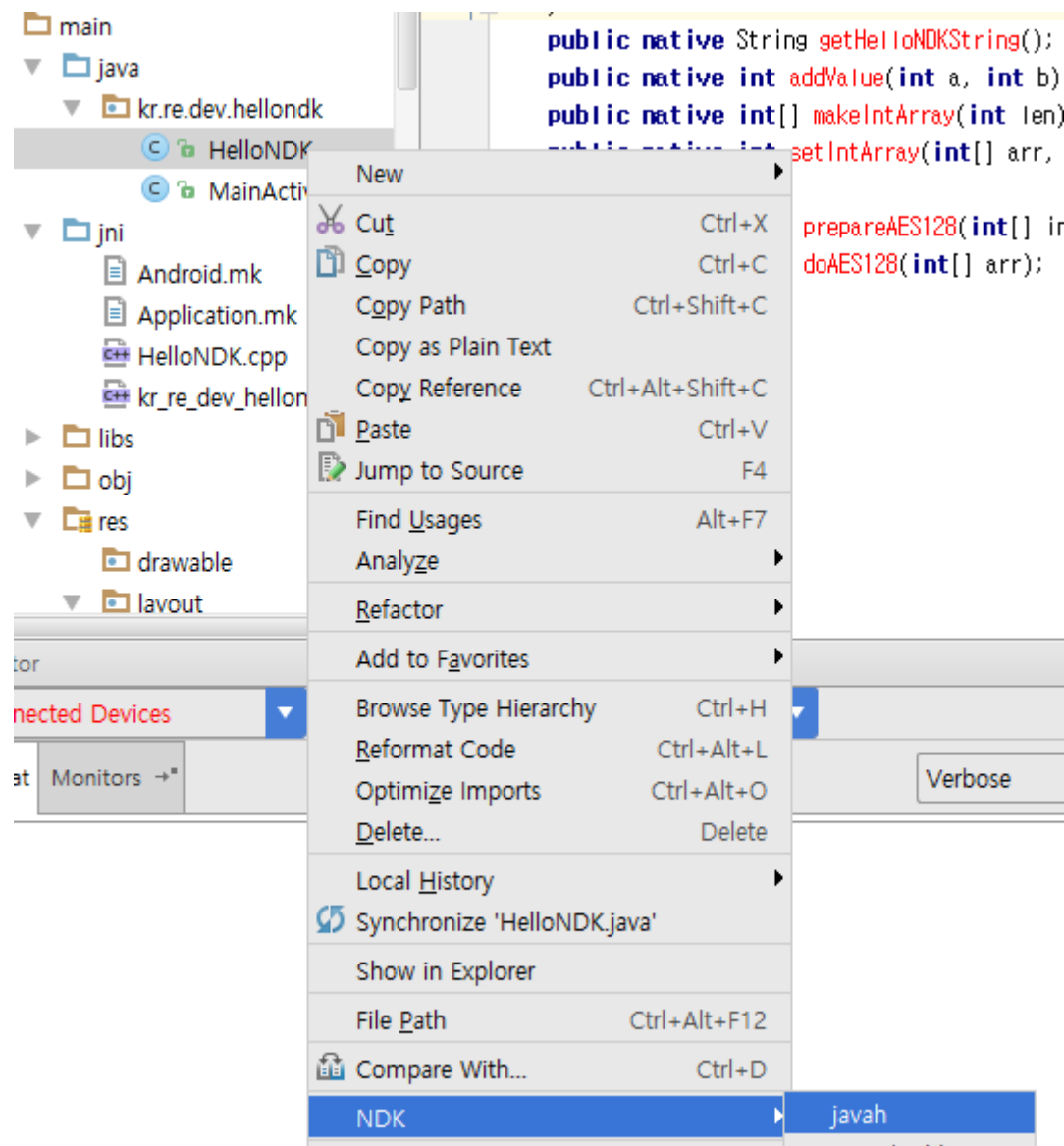
Parameters: -classpath \$Classpath\$ -v jni \$FileClass\$

Working directory: \$ProjectFileDir\$\app\src\main\jni

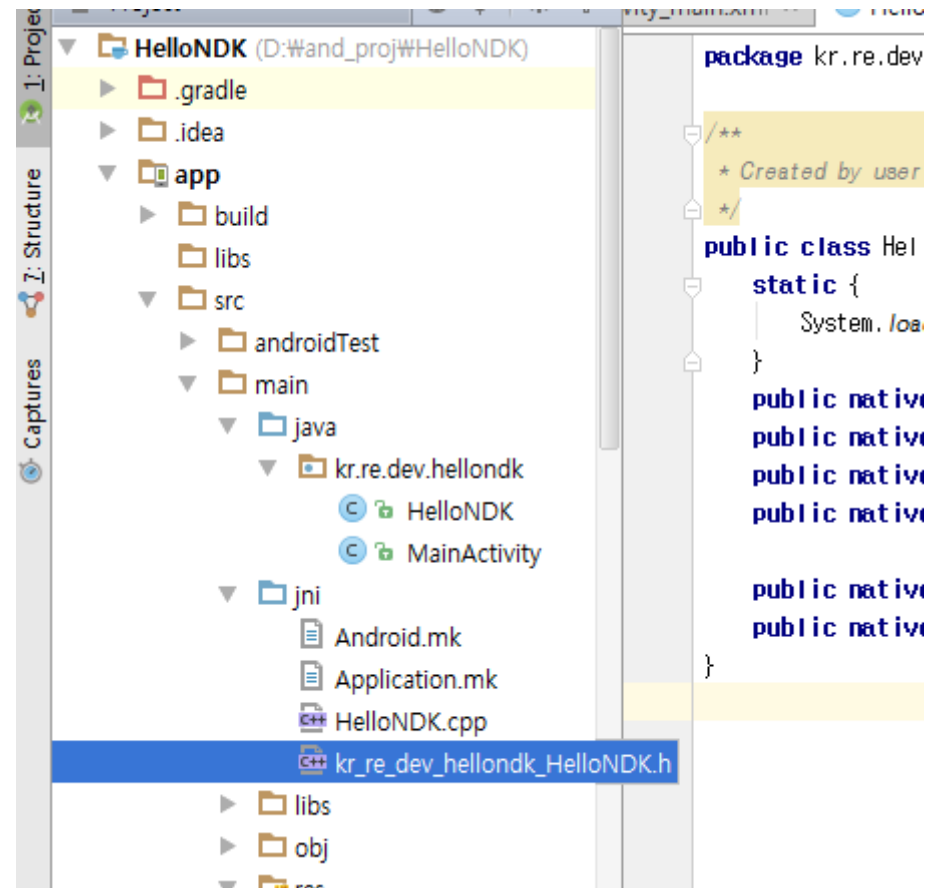
OK Cancel Help

또한 Parameters 도 `-classpath $Classpath$ -v -jni $FileClass$` 로 바꿔주도록 한다(jni 에 '-' 가 빠져있음)
Working directory 는 `$ProjectFileDir$\app\src\main\cpp` 로 바꿔주도록 한다.

이제 아래와 같이 HelloNDK.java 를 우클릭하여 javah 로 C 코드와 연동할 준비를 한다.



구동이 정상적으로 잘 진행되면 아래와 같이 C 코드 형태의 헤더 파일이 만들어질 것이다.



다음으로 필자는 Android Studio 버전이 낮아서 안보이지만
당사자들의 경우 cpp 쪽에 native-lib.cpp 라는 코드가 존재할 것이다.
이 부분을 아래와 같이 변경하도록 한다.

```
#include <jni.h>
#include <string>

#include "kr_re_dev_hellondk_HelloNDK.h"

JNIEXPORT jstring JNICALL Java_kr_re_dev_hellondk_HelloNDK_stringFromJNI(
    JNIEnv *env,
    jobject /* this */) {
    std::string hello = "Hello from C++";
    return env->NewStringUTF(hello.c_str());
}
```

다음으로 MainActivity.java 코드를 아래와 같이 작성한다.

```
public class MainActivity extends AppCompatActivity {

    // Used to load the 'native-lib' library on application startup.
    /*
    static {
        System.loadLibrary("native-lib");
    }
    */

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // Example of a call to a native method
        TextView tv = (TextView) findViewById(R.id.sample_text);
        HelloNDK hello = new HelloNDK();
        tv.setText(hello.stringFromJNI());
    }

    /**
     * A native method that is implemented by the 'native-lib' native library,
     * which is packaged with this application.
     */
    //public native String stringFromJNI();
}
```

NDK 프로젝트를 처음 만들었을때와 같은 결과가 나올것이다.
이로서 Android Java 와 C 코드를 연동하기 위한 모든 기본적인 작업이 완료되었다고할 수 있다.

이후 더 깊은 내용을 들어가고자 한다면
리눅스 시스템 프로그래밍과 리눅스 커널쪽을 공부해서 디바이스 드라이버를 공부하면됨.