

# Android

Android의 소개 및 구조, 개발 환경, 개발 기초

# Android의 소개 및 구조

# Android는 언어가 아닌 플랫폼이다.

1. 동작대상 - Embedded Device (예) Mobile phone, Digital TV
2. Software Stack

OS  
Library  
Application Framework  
Applications



# Android Platform의 특징

## 1. Open Source 기반

- OSA 단체에서 Google이 주도적으로 확산시킴
- 소스로 변경하고 볼 수 있지만, Open Source에도 “License”가 있음
- **License** (Open Source 를 받았을 때 License를 확인해야 함)

구 분	GPL	LGPL	BSD	MPL
코드의 무료이용	O	O	O	O
코드의 자유배포	O	O	O	O
소스코드의 공개	O	O	O	O
소스코드의 수정	O	O	O	O
수정코드의 소스공개	O	O	X	O
상용소프트웨어와의 링크	X	O	O	O

## 2. Android Kernel – Linux Kernel OS

Original Linux Kernel을 사용하는 것이 아니라 Android용으로 patch해서 사용하는 것임

patch 내용 – 알람, 디버거, Low Memory Killer(전원 관리를 위해)의 사용, 크고 무거운 기능은 제거

## Android Platform의 특징

---

### 3. JAVA 언어 사용

생산성이 높고, 하드웨어 추상 층을 제공하여 전문 지식이 없어도 개발 가능 (성능, 섬세 함에서는 다소 불편함)

### 4. 검증된 많은 라이브러리를 대거 포함하여 웬만한 기능은 별도의 외부 라이브러리를 사용할 필요가 없음

### 5. BUILT-IN 프로그램과 사용자가 만든 프로그램이 동일한 API를 사용하여 모든 프로그램이 평등함. 사용자가 교체할 수 있음



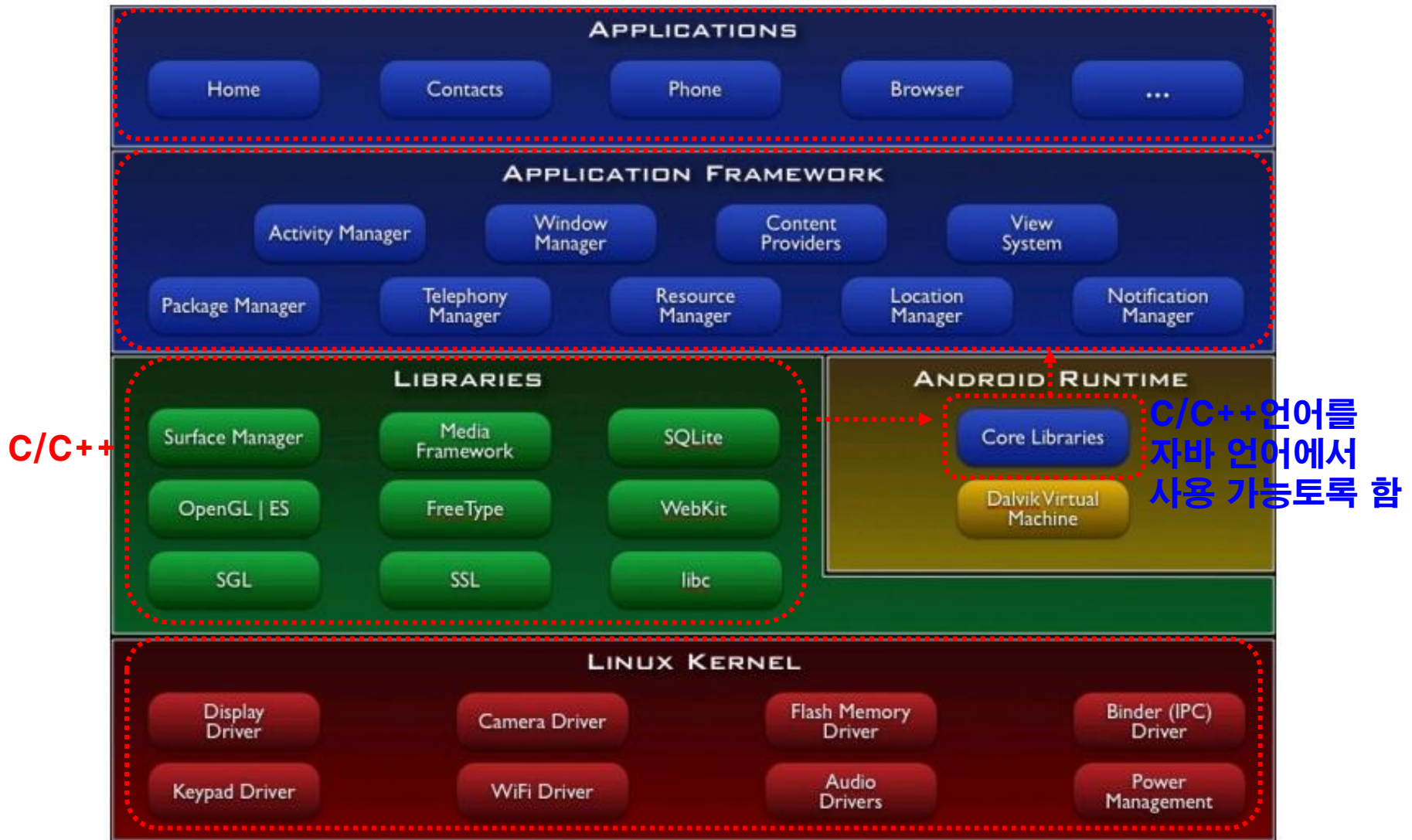
## Android의 Architecture

---

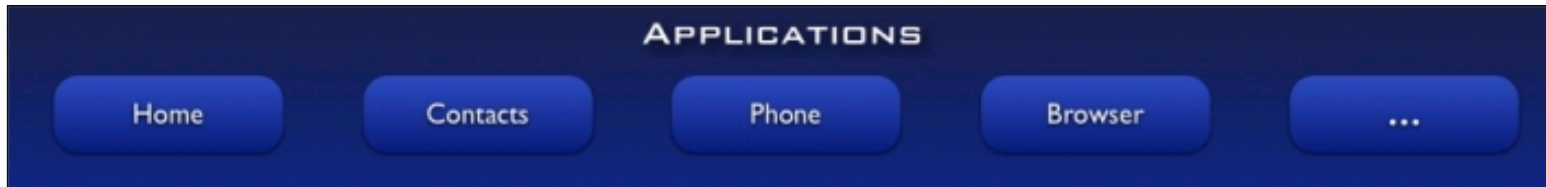
1. 구현에 사용하는 언어는 Java이며, Java의 특징은 Platform에 독립적이라는 것임
2. Platform에 독립적이라는 것은 Java Virtual Machine을 사용하기 때문인데, JVM은 License가 있어 비용을 지불해야하기 때문에 Android Run Time(실행환경구성)을 사용함
3. Android Run Time = Core Library + Dalvik VM
4. \*.class 파일 사용 하지 않고 Instance 방식(\*.dex)을 사용하며, Application 1개당 1개의 DVM이 생성됨
5. Android Core Java Library는 Java SE, Java ME에 똑같지는 않지만 중복되는 부분이 상당히 많이 있음
6. Java EE, Java SE, Java ME에서 ME는 라이선스가 있으며, SE는 라이선스가 없으므로 Android에서는 SE를 사용함



# Android의 Architecture



# 1 Applications



- Google에서 자체 제작한 application
- 안드로이드에서는 E-mail 클라이언트, SMS 프로그램, 달력, 지도, 브라우저, 전화번호부, 그리고 다른 것들을 포함하는 핵심 애플리케이션들이 탑재되어 있음
- 모든 애플리케이션은 JAVA 프로그래밍 언어로 작성됨
- Handle Application ; 플랫폼 내부에 Built-In 된 것으로 사용자가 삭제 할 수 없음, 제품 판매시 일반 user 권한으로 제공되기 때문에 삭제가 불가능함. (Google의 Built In도 있고, 제조사의 Built In도 있음)
- Download Application : 사용자가 Download 받은 것으로 사용자가 삭제 가능





## 2 Applications Framework

개발자들은 애플리케이션 Framework API에 완벽하게 접근할 수 있음



- Google에서 JAVA로 자체 제작
- 안드로이드 애플리케이션 프레임워크는 Java 기반의 Framework이며, 대부분이 JNI(Java Native Interface) 통해 C/C++ 코드로 작성되어 있음
- 핵심 시스템 서비스를 담당하는 Core 시스템 서비스 + 하드웨어와의 인터페이스를 담당하는 하드웨어 서비스들
  - Core system service: Activity Manager, Package Manager, Window Manager, Resource Manager, Content Providers, View system
  - Hardware service: Provides low-level access to hardware service, Location manager, Telephony manager, Bluetooth service, Wifi/USB/Sensor Service

### 3 Libraries



안드로이드 시스템의 다양한 컴포넌트에 의해 사용되는 **C/C++ 라이브러리 집합**.

이러한 라이브러리의 기능들은 안드로이드 애플리케이션 Framework를 통해서 개발자들에게 제공됨

#### ➔ 시스템 C 라이브러리 libc

임베디드 리눅스 기반의 디바이스를 위해서 구현된 표준 C 시스템 라이브러리의 구현물이며 **Bionic**이라 한다. Bionic은 임베디드에서 사용을 위해 **최적화된 직접 구현된 libc**이다. **기존 Linux libc를 사용하지 않고 새로 만든 이유는** user-space에서는 **GPL(General Public License : GNU 일반 공중 사용 허가서로 소스 수정 시 꼭 소스를 공개하여야 한다는 규약이 있다.)**을 사용하지 않기 위한 라이선스 이유와, **개별 프로세스마다 포함되어야 하는 영역이기 때문에 크기를 최소화하기 위해서, 제한적 CPU 파워 하에서도 빠를 필요가 있기 때문임**

#### ➔ Function Library

웹 브라우저를 위한 WebKit, PacketVideo의 OpenCore 플랫폼 기반의 미디어 프레임 워크, 가벼운 DB SQLite

## 4 안드로이드 런타임(RunTime)



Dalvik 가상 머신은 안드로이드에서 자체적으로 만든 것으로 clean-room(청정영역)을 제공하는 가상 머신. 이것은, 애플리케이션의 호환성과 실행 일관성을 제공하며, 최적화된 파일 포맷(.dex)과 Dalvik 바이트 코드로 실행함  
더불어 빌드 시점에서  
Java.class / .jar 파일들은 **.dex파일로 변환됨**

### Dalvik 가상 머신

- 임베디드 환경을 위해 디자인 됨
- 디바이스 별로 다양한 가상머신 프로세스들을 지원함
- 높은 수준으로 CPU에 최적화된 바이트코드 인터프리터에 기반하여 실행 시 메모리를 매우 효율적으로 사용함

### Core 라이브러리

- **Java 언어로 라이브러리들을 사용할 수 있도록 함**
- 강력하지만, 단순하고 익숙한 개발 플랫폼을 제공하는 Java 언어를 위한 Core API들을 포함하고 있음 (Data structure, Utility, File Access, Network Access, Graphic 등이 포함)

## 5 리눅스 커널



- 안드로이드는 리눅스 커널을 기반하고 있으나, 리눅스는 아님
- X-Window와 같은 내장 윈도우 시스템을 포함하지 않으며
- **glibc를 지원하지 않으며, 표준 리눅스 유틸리티 전체를 포함하지 않음**
- 안드로이드 리눅스 커널 버전 2.6.23, 2.6.24, 2.6.25, 2.6.27을 사용해 왔으며 안드로이드 지원을 위해 리눅스 커널 확장을 위한 패치를 포함함
- 안드로이드에서 리눅스를 사용하는 이유: 메모리 및 프로세스 관리, Permission 기반의 보안 모델, 검증된 드라이버 모델, 공유 라이브러리 지원, 오픈 소스 기반 등의 장점 때문
- **안드로이드 확장 리눅스 커널 영역:** Alarm, Ashmem, Binder, Power Management, Low Memory Killer, Kernel Debugger, Logger 임



# Android의 개발 환경

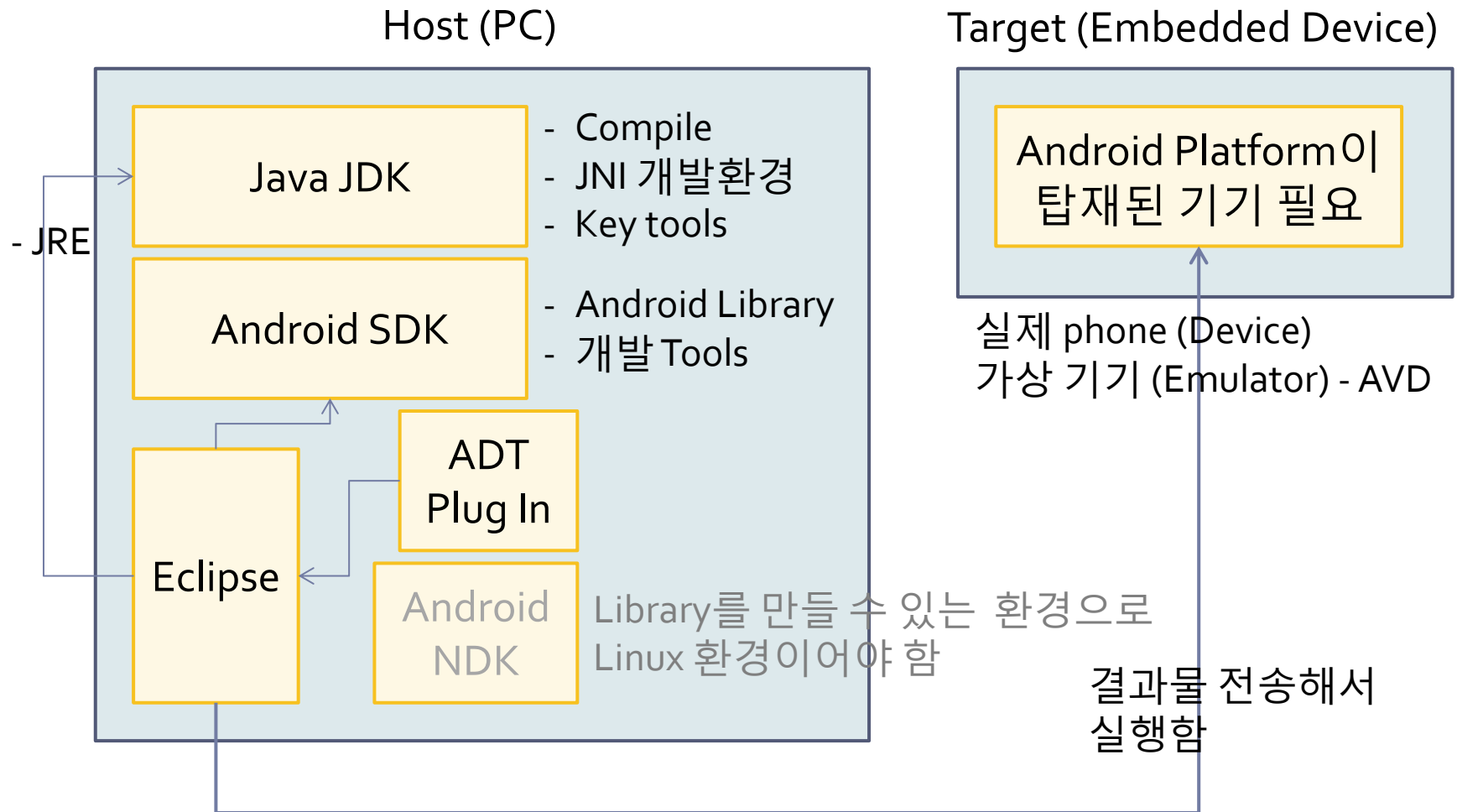
## Program 개발 유형별 개발 환경

---

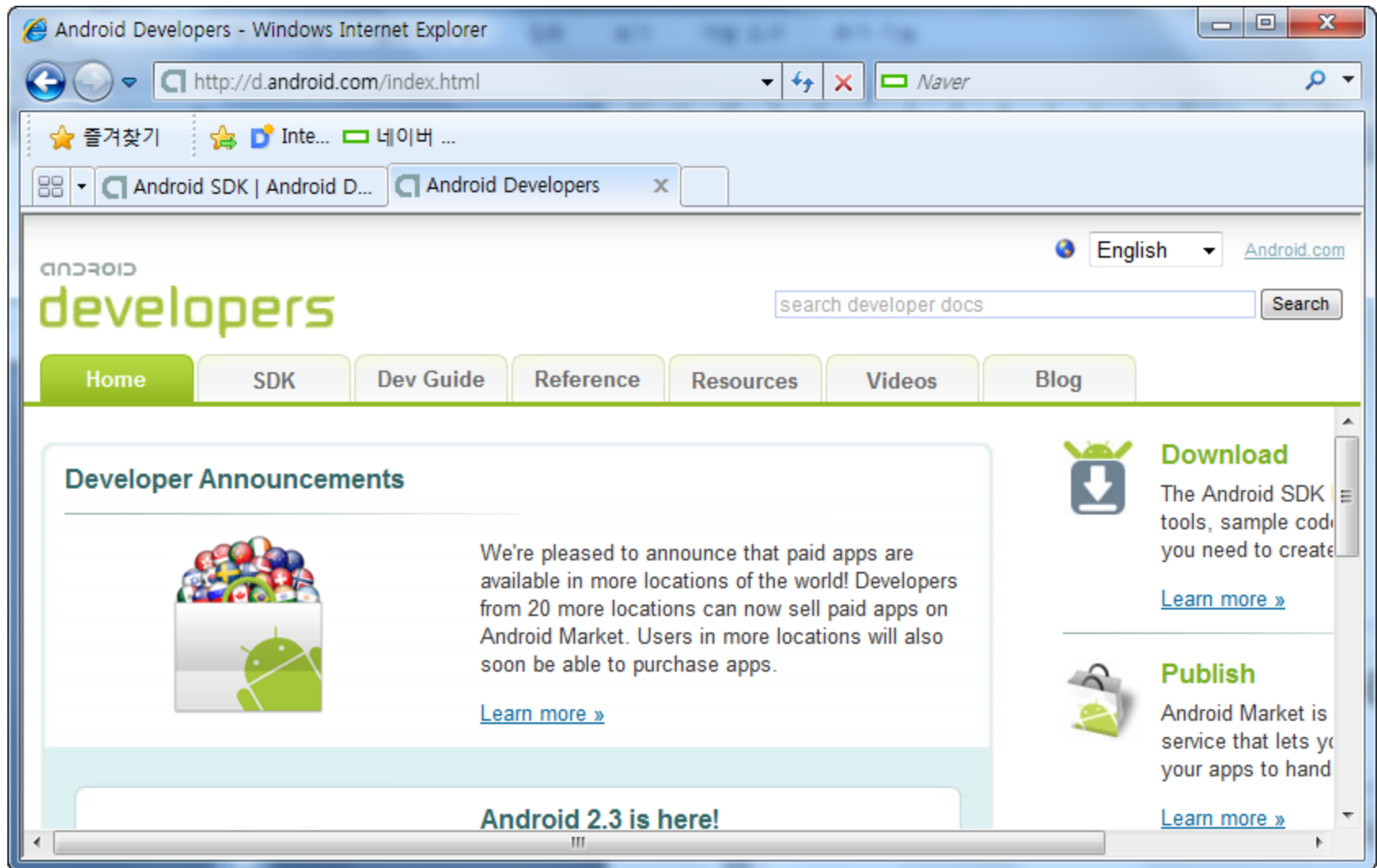
개발 유형	개발 환경
Android Application	Java 기반, Android Library 사용 (Java, Android 환경)
Android Application + C 기반 Library(Library 구축)	+ Library 개발 환경
Linux System Program	Linux Kernel 기반
Android Platform 포팅, Device 개발	Windows 환경이 아닌 Linux 환경이 필수



# Embedded 개발 환경



개발자 사이트 : <http://d.android.com>





## Resources - Platform Versions – Current Distribution

---

Platform	Codename	API Level	Distribution
<a href="#">Android 1.5</a>	Cupcake	3	1.1%
<a href="#">Android 1.6</a>	Donut	4	1.4%
<a href="#">Android 2.1</a>	Eclair	7	11.7%
<a href="#">Android 2.2</a>	Froyo	8	45.3%
<a href="#">Android 2.3 - Android 2.3.2</a>	Gingerbread	9	0.5%
<a href="#">Android 2.3.3 - Android 2.3.7</a>		10	38.2%
<a href="#">Android 3.0</a>	Honeycomb	11	0.2%
<a href="#">Android 3.1</a>		12	0.9%
<a href="#">Android 3.2</a>		13	0.7%

*Data collected during a 14-day period ending on **October 3, 2011***

---



# 개발 툴 설치

---

1. JDK 1.7
2. Eclipse (Indigo)
3. Android SDK (Android SDK version 13)
4. ADT Plug In 설치 (Eclipse의 Preference에서)
5. AVD 생성 및 옵션 조정 (Emulator 생성)



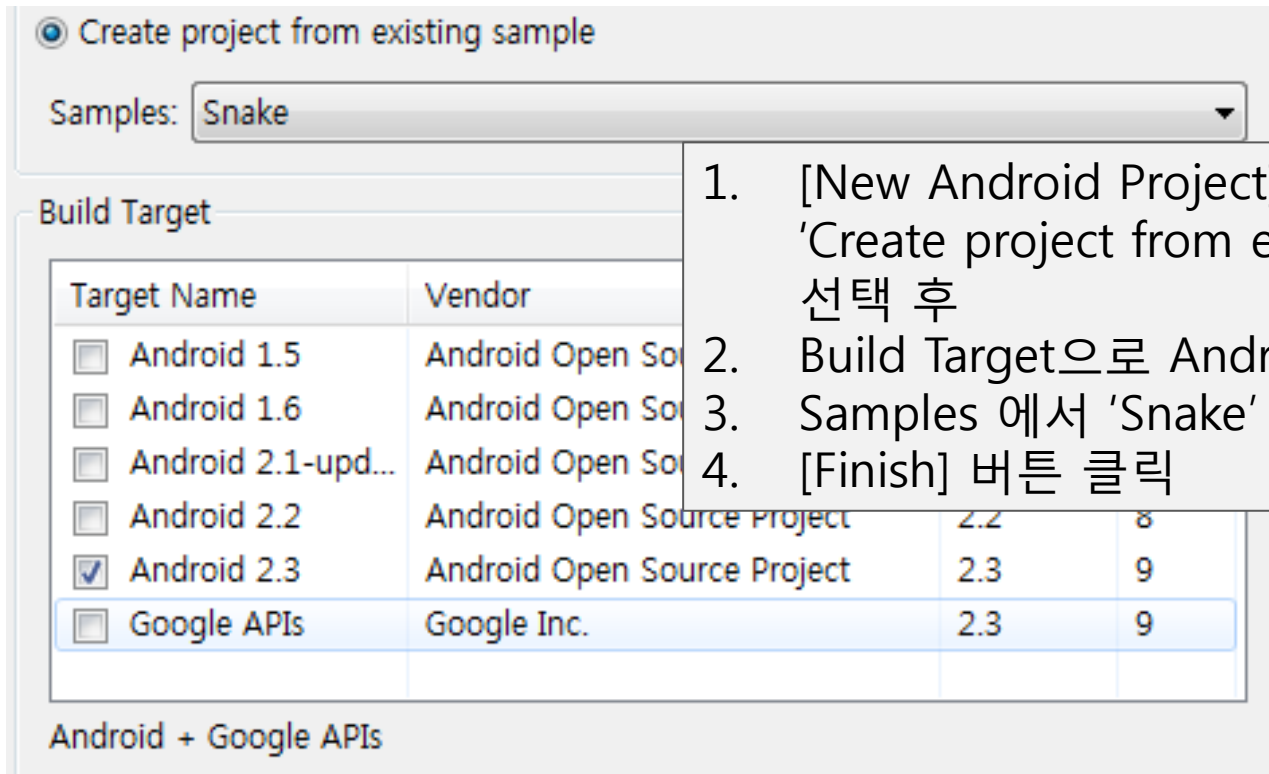


# 첫 안드로이드 응용 프로그램 Snake

## 개발 환경 점검

### Snack 응용프로그램을 Eclipse 작업공간의 새 프로젝트로 추가

1. Eclipse 실행 후, 메뉴에서 File – New – Project 선택
2. Android와 Android Project를 선택 후, [Next] 버튼 클릭

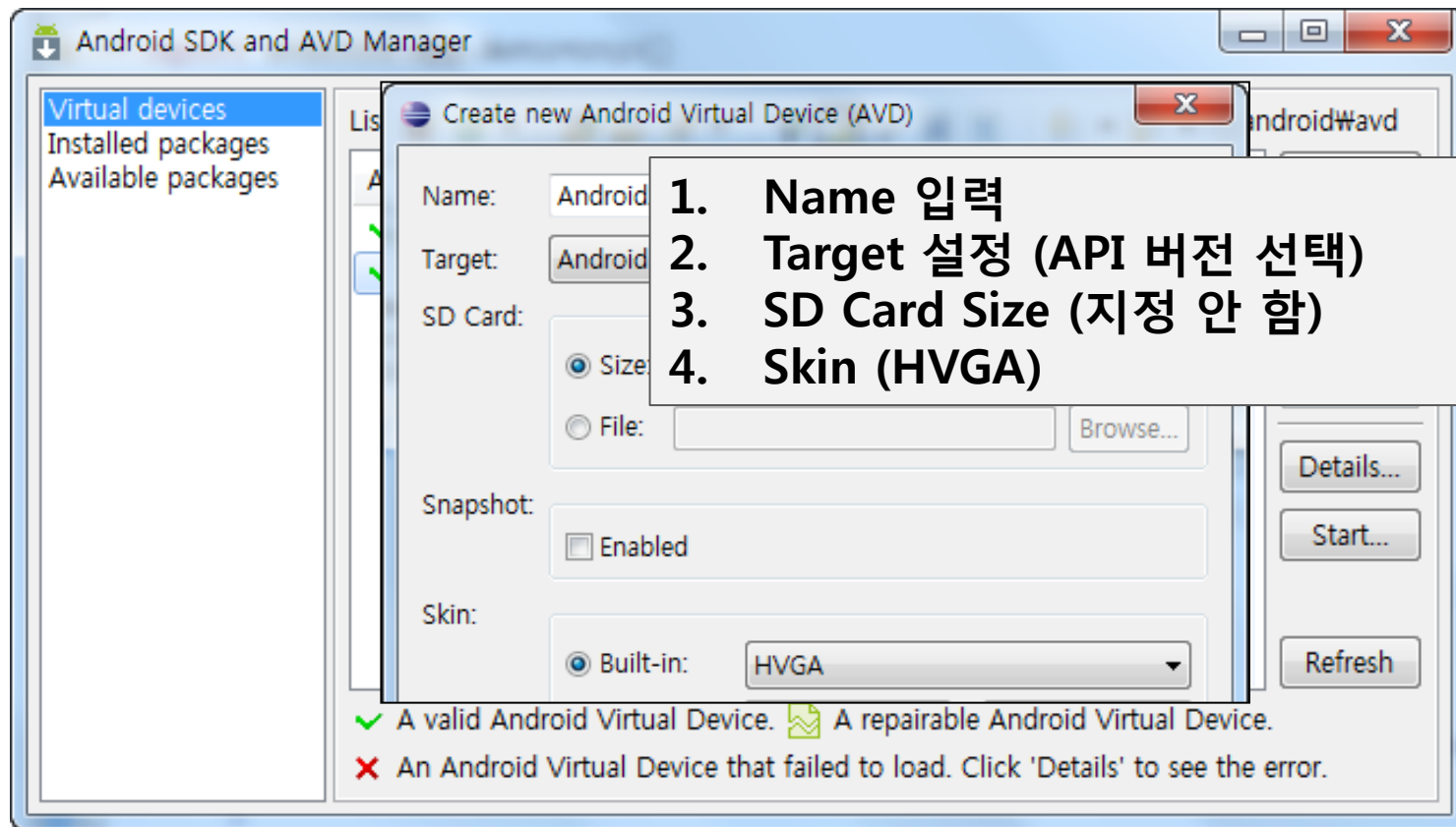


1. [New Android Project] 창에서 'Create project from existing sample' 선택 후
2. Build Target으로 Android 2.3.3 선택
3. Samples 에서 'Snake' 선택
4. [Finish] 버튼 클릭

## Snake Project를 위한 Android 가상 장치생성

### 안드로이드 가상 장치(Android Virtual Device, AVD) 만들기

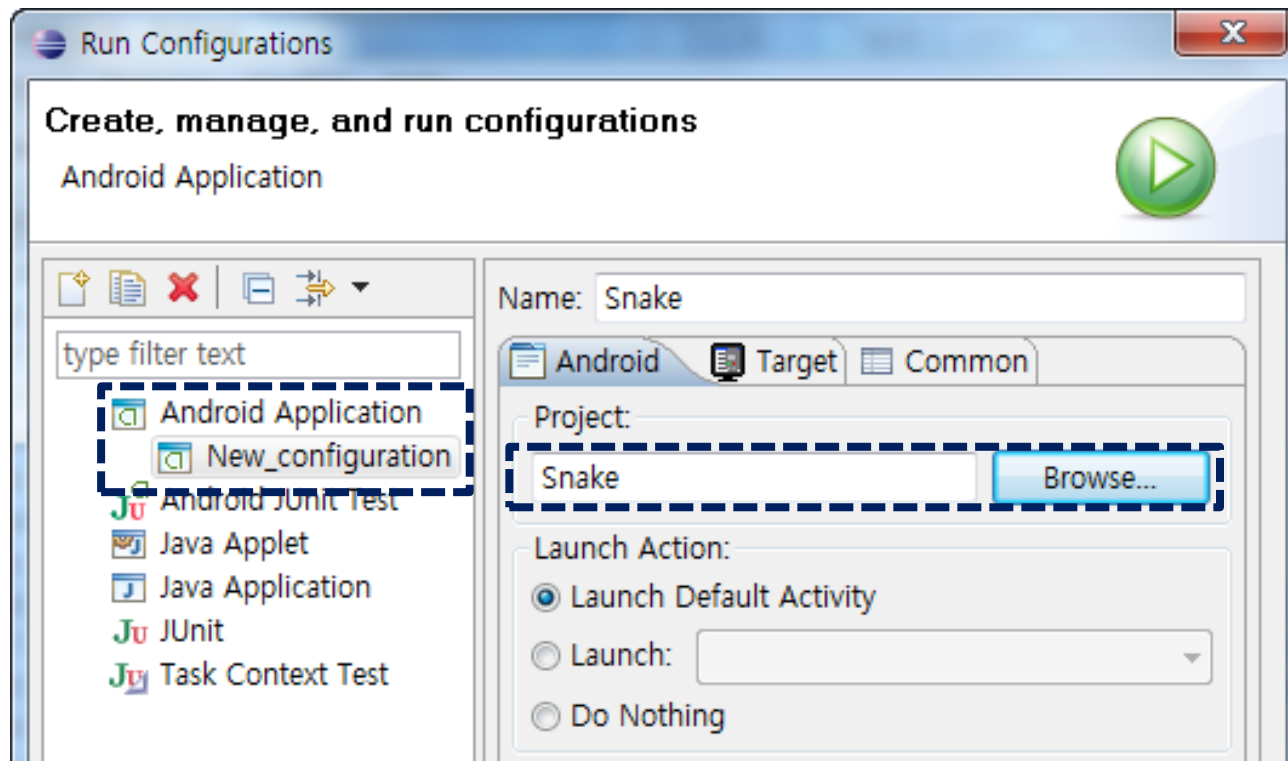
1. [Windows]-[Android SDK and AVD Manager] 메뉴 클릭
2. 'Virtual devices' 선택 후, [New] 버튼 클릭



## Snake 프로젝트의 시동 구성 설정

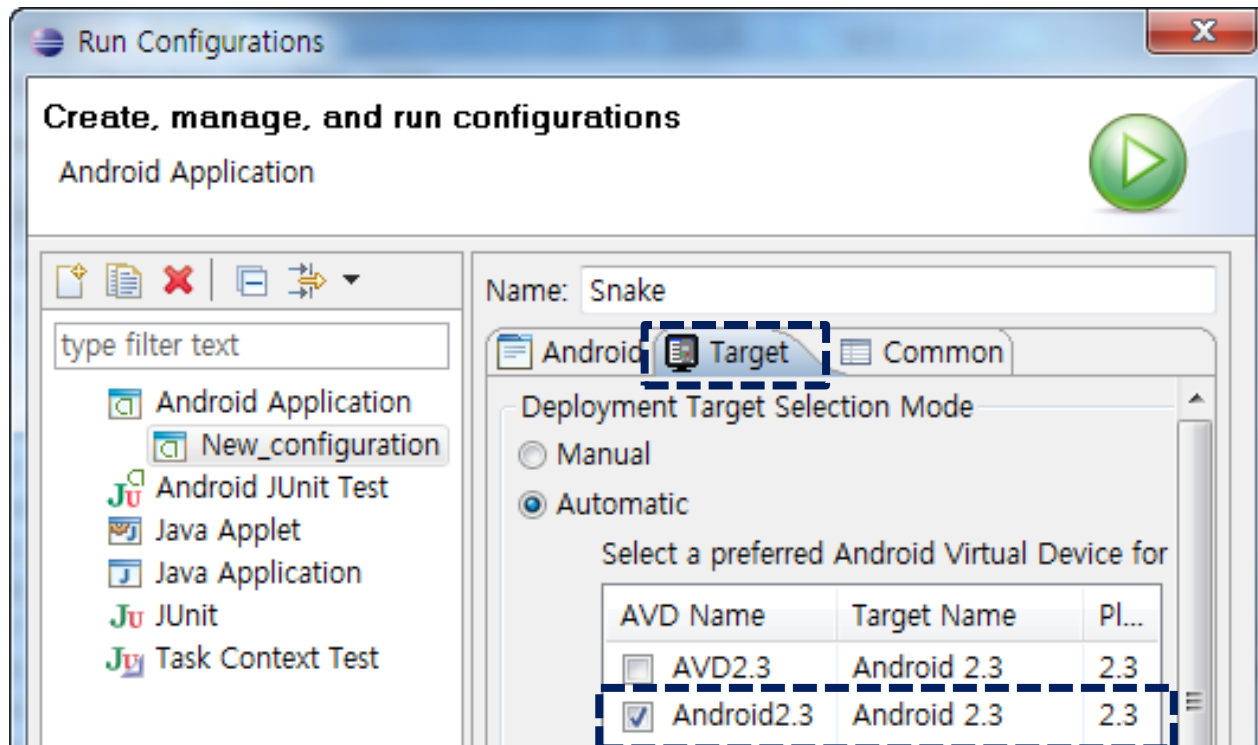
Snake 응용 프로그램이 어떤 환경에서 실행될 것인지를 Eclipse에게 알려주기 위해 Eclipse 안에서 시동 구성이라는 것을 새로 만들

### 1. 메뉴에서 [Run]-[Run Configurations] 선택



## Snake 프로젝트의 시동 구성 설정

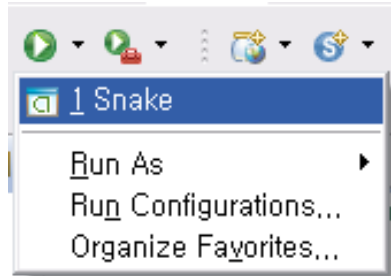
2. Target 탭으로 가서, AVD 목록에서 원하는 AVD를 선택하고 [Apply] 버튼 클릭 후, [Run] 버튼 클릭



## Snake Application을 Android Emulator에서 실행

---

1. 툴바에 있는 Run As의 드롭다운 메뉴를 펼침
2. Snake 선택

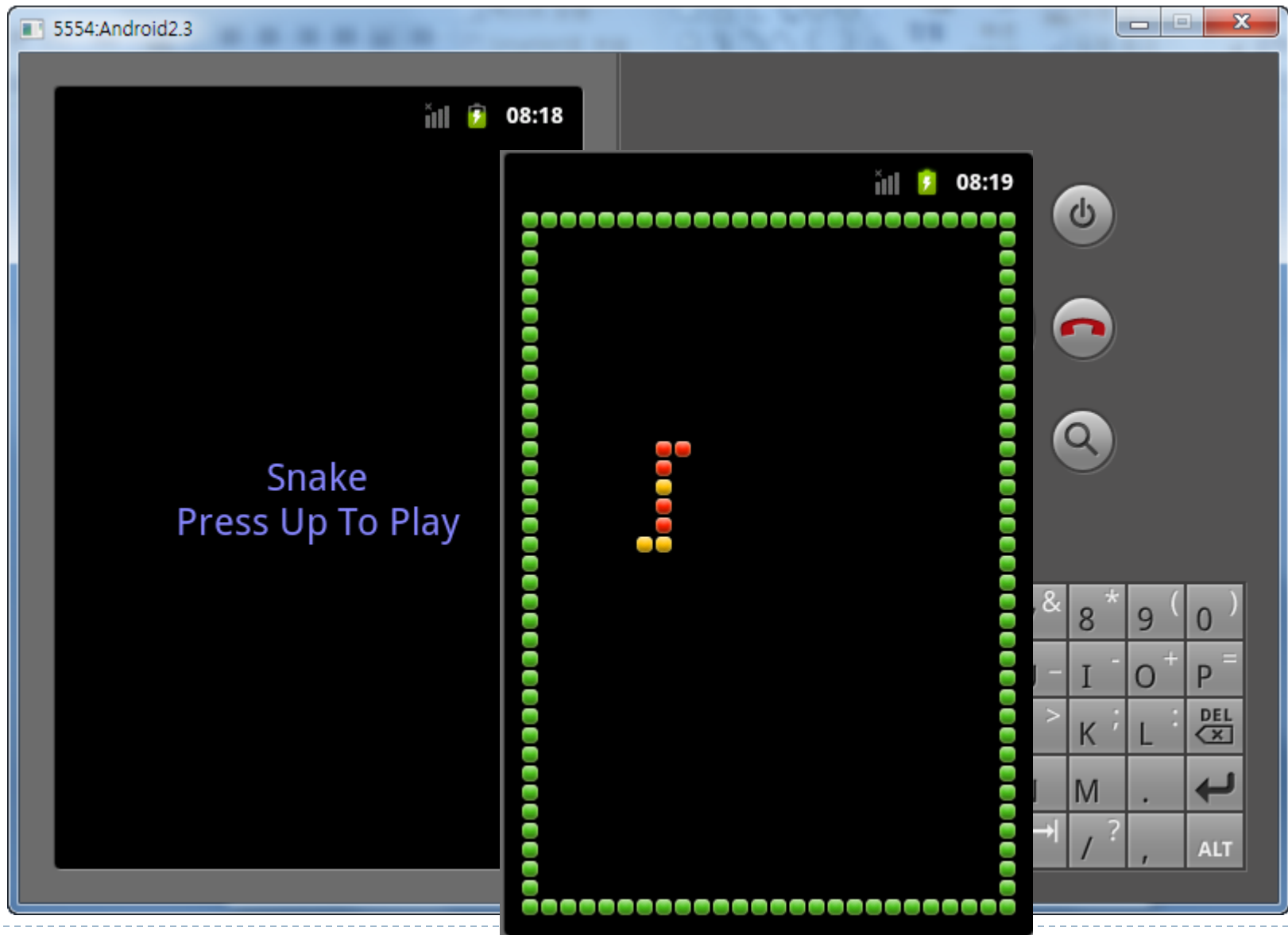


3. 안드로이드 에뮬레이터가 시동됨 (Snake가 실제 뜨기까지 시간이 걸릴 수 있음)

<사용자 계정 이름이 한글인 경우 오류가 발생할 수 있음~>



# 에뮬레이터~~!

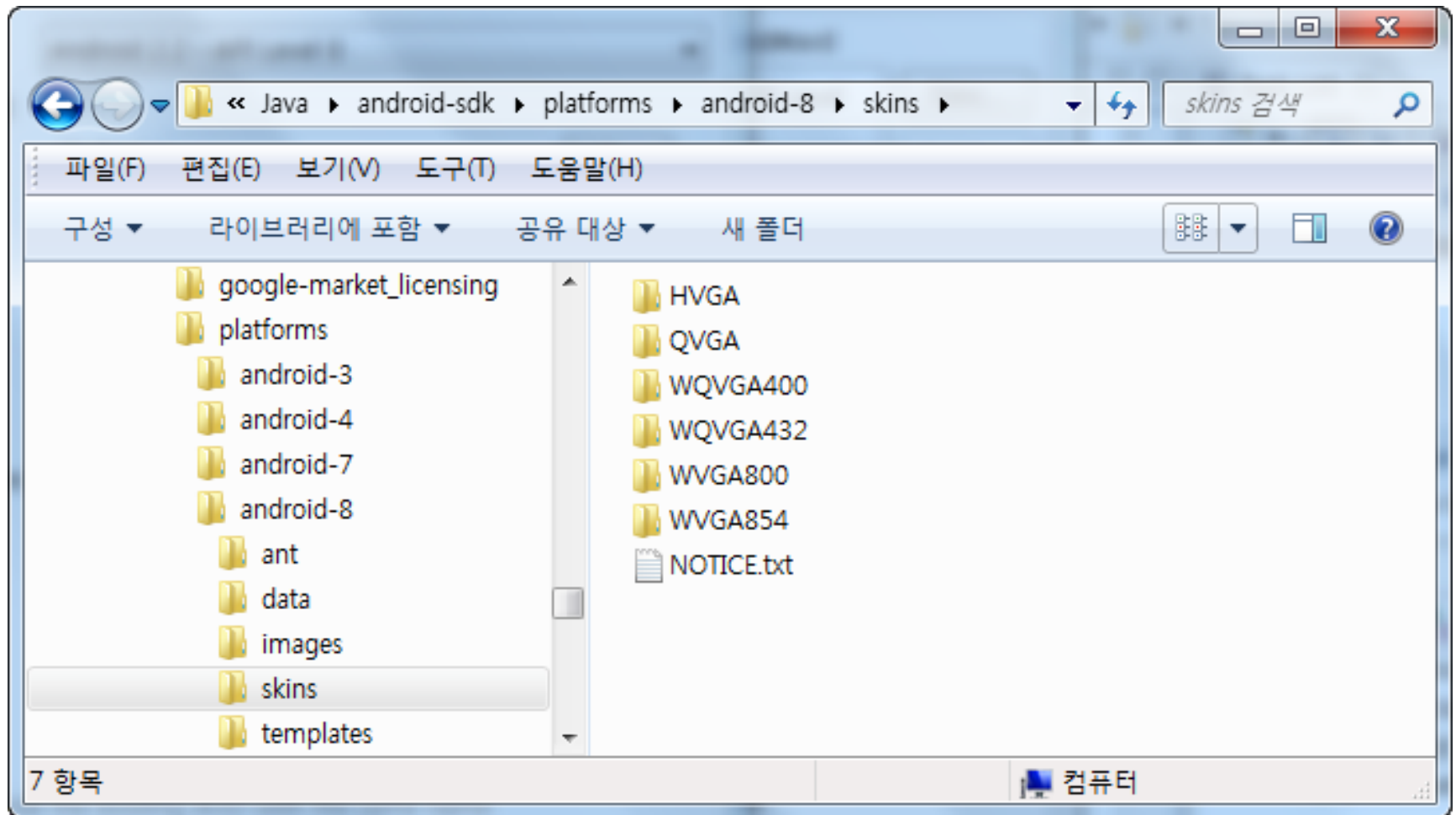




## AVD 2.3 Emulator



## AVD의 스킨 종류



## adb(Android Debug Bridge) 사용 (교재 311-314)

---

```
C:\Users\julie yoon>adb devices  
List of devices attached  
emulator-5554    device
```

### android-sdk/platform-tools/adb.exe

#### adb devices

adb 서버가 인식한 휴대폰과 에뮬레이터 목록을 보여줌  
연결된 devices의 TCP/IP 포트 번호(5555~5585)를 알아낼 때 도움이 됨

#### adb에 대한 안내

<http://d.android.com/guide/developing/tools/adb.html>



## adb(Android Debug Bridge) 사용

```
C:\#Users\julie yoon>
C:\#Users\julie yoon>adb -s emulator-5554 shell
# ls -l
ls -l
dr-x----- root      root      2011-01-24 03:50 config
drwxrwx--- system    cache    2011-01-24 03:50 cache
lrwxrwxrwx root      root      2011-01-24 03:50 sdcard -> /mnt/sdcard
drwxr-xr-x root      root      2011-01-24 03:50 acct
drwxrwxr-x root      system    2011-01-24 03:50 mnt
lrwxrwxrwx root      root      2011-01-24 03:50 d -> /sys/kernel/debug
lrwxrwxrwx root      root      2011-01-24 03:50 etc -> /system/etc
drwxr-xr-x root      root      2010-06-30 21:06 system
drwxr-xr-x root      root      1970-01-01 00:00 svcs
```

### adb shell

타겟 시스템의 쉘에 연결하고 # 프롬프트를 띄움. 쉘은 간소한 유닉스 쉘 같아서 명령으로 타겟 시스템을 탐색하고 수정할 수 있음

```
drwx
dr-x
-rwx
-rwx
-rwx
-rw-r--r-- root      root      118 1970-01-01 00:00 default.prop
drwxrwx--x system    system    2011-01-24 03:54 data
drwx----- root      root      2010-01-28 00:59 root
drwxr-xr-x root      root      2011-01-24 03:51 dev
..
```

## emulator의 폴더

---

data : Download Application

Application 별 작업(Application)

file, 환경 설정 정보, database, Library 등이 들어감

mnt : 외부장치 (SD Card)

system : Android system - supervisor 권한 획득 필요

Kernel

Library

Framework

Built-In Application



## shell의 사용 (system/app 폴더에 포함된 파일 목록 확인)

### Built- in Applications

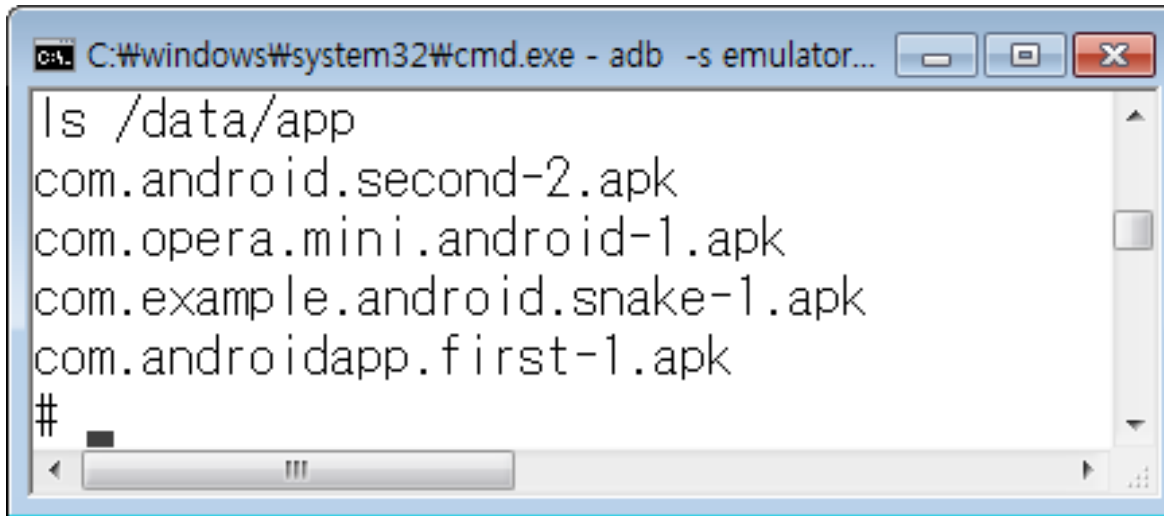
```
C:\windows\system32\cmd.exe - adb -s emulator-5554 shell
# ls -l /system/app
ls -l /system/app
-rw-r--r-- root    root      43500 2010-06-30 21:06 Term.apk
-rw-r--r-- root    root     941883 2010-06-30 21:06 Email.apk
-rw-r--r-- root    root      3966 2010-06-30 21:06 SdkSetup.apk
-rw-r--r-- root    root     15133 2010-06-30 21:06 PicoTts.apk
-rw-r--r-- root    root     30823 2010-06-30 21:06 TtsService.apk
-rw-r--r-- root    root     58024 2010-06-30 21:06 PackageInstalle
-rw-r--r-- root    root    5604964 2010-06-30 21:07 Launcher2.apk
-rw-r--r-- root    root     22897 2010-06-30 21:07 NetSpeed.apk
-rw-r--r-- root    root    2684364 2010-06-30 21:06 OpenWnn.apk
-rw-r--r-- root    root     529373 2010-06-30 21:07 Music.apk
-rw-r--r-- root    root     39584 2010-06-30 21:06 LiveWallpapersP
-rw-r--r-- root    root     49646 2010-06-30 21:06 CertInstaller.a
-rw-r--r-- root    root     560310 2010-06-30 21:07 Browser.apk
```



## shell의 사용 (data 폴더에 포함된 파일 목록 확인)

---

사용자가 install한 app /data/app/에 \*.apk 형태로 존재



```
C:\windows\system32\cmd.exe - adb -s emulator...  
ls /data/app  
com.android.second-2.apk  
com.opera.mini.android-1.apk  
com.example.android.snake-1.apk  
com.androidapp.first-1.apk  
#
```



http://www.freewarelover.com/android

The screenshot shows a Windows Internet Explorer browser window. The title bar reads 'Android Freeware: Best and Free Software for Android mobile platform - Windows Internet Explorer'. The address bar shows the URL 'http://www.freewarelovers.com/android'. The page content includes a header with the site name and a search bar. A white box highlights the text 'PAC MAN Opera Mini'. Below this is a section titled '« Android Freeware Lovers »'. The main content area is divided into three columns: 'Categories', 'Popular in January', and 'New Android Devices'. The 'Categories' column lists various app categories with their counts. The 'Popular in January' column lists popular apps with their download counts. The 'New Android Devices' column lists new Android devices.

Android Freeware: Best and Free Software for Android mobile platform - Windows Internet Explorer

http://www.freewarelovers.com/android

안드로이드 스킨

즐거찾기

Android SDK | Android D... Platform Versions | Andro... Android Freeware: Best...

♥ Android Freeware  
Best and Free Software for Android mobile platform

PAC MAN  
Opera Mini

Google™ Custom Se

« Android Freeware Lovers »

Categories	Popular in January	New Android Devices
<a href="#">Communications</a> (175)	<a href="#">Adobe Flash Player</a>	↓ 15723
<a href="#">Entertainment</a> (226)	<a href="#">Angry Birds</a>	↓ 10908
<a href="#">Finance</a> (57)	<a href="#">ASTRO File Manager</a>	↓ 8649
<a href="#">Games</a> (141)	<a href="#">Skype</a>	↓ 8041
<a href="#">Health</a> (77)	<a href="#">Puzzles</a>	↓ 7821
<a href="#">Multimedia</a> (94)	<a href="#">Apps Installer</a>	↓ 7589
<a href="#">News</a> (27)	<a href="#">PAC-MAN</a>	↓ 7373
<a href="#">Productivity</a> (185)	<a href="#">Angry Birds Seasons</a>	↓ 7292
<a href="#">Reference</a> (129)	<a href="#">Skype Lite Beta</a>	↓ 6077
	<a href="#">Documents To Go</a>	↓ 5158

• [Samsung Galaxy A](#)

• [Samsung Galaxy 580](#)

• [Nexus S](#)

• [Motorola MOTOROI](#)

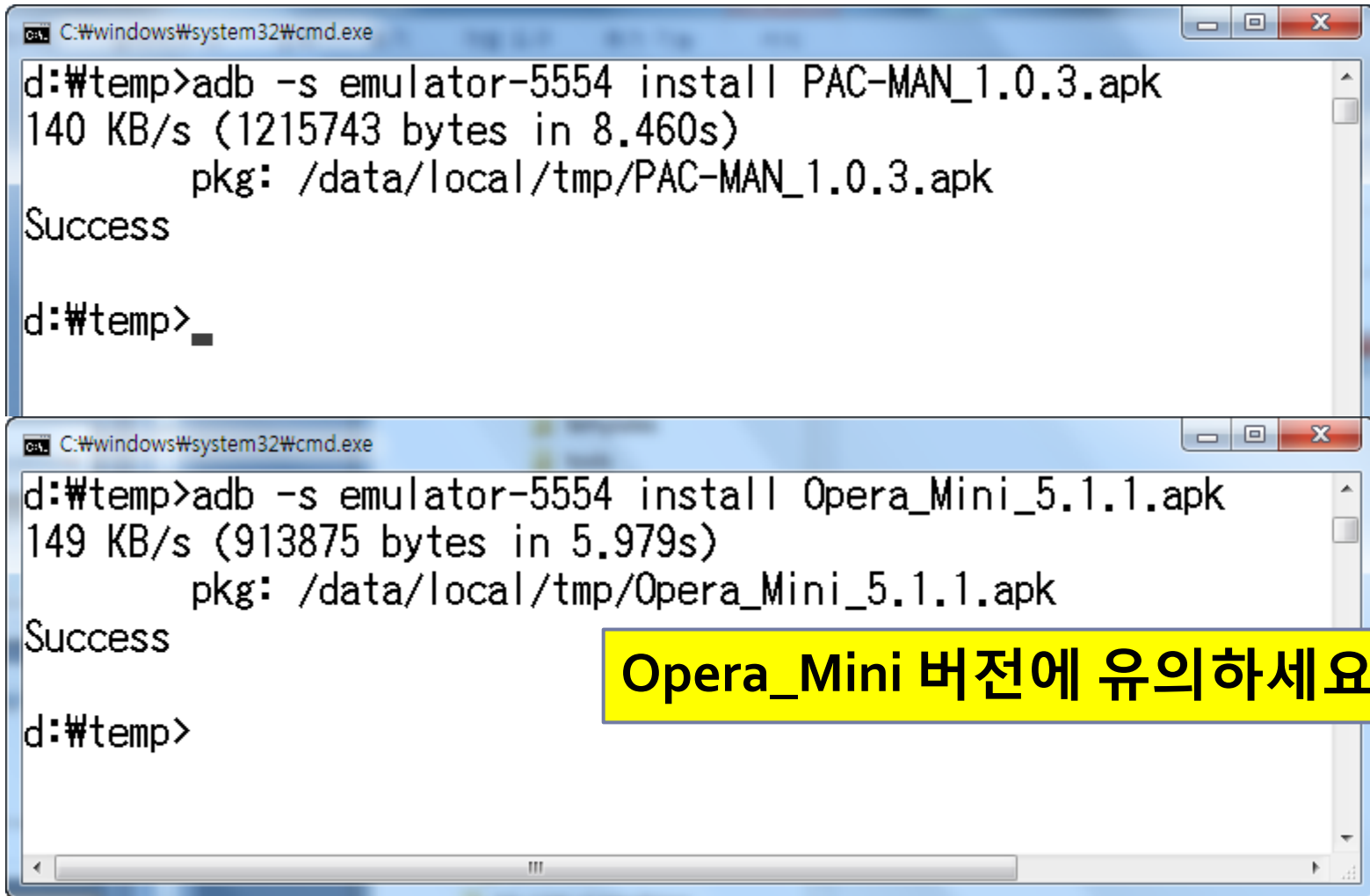
• [Motorola FLIPOUT](#)

• [Motorola DEXT](#)

• [Motorola BRAVO](#)

• [Motorola DROID Pro](#)

## 응용 프로그램 설치 및 확인



The image shows two sequential screenshots of a Windows command prompt window. The title bar of the window is 'C:\windows\system32\cmd.exe'. The first screenshot shows the command 'adb -s emulator-5554 install PAC-MAN\_1.0.3.apk' being executed. The output shows a transfer rate of 140 KB/s, a total of 1215743 bytes transferred in 8.460 seconds, the package path '/data/local/tmp/PAC-MAN\_1.0.3.apk', and a 'Success' status. The second screenshot shows the command 'adb -s emulator-5554 install Opera\_Mini\_5.1.1.apk'. The output shows a transfer rate of 149 KB/s, a total of 913875 bytes transferred in 5.979 seconds, the package path '/data/local/tmp/Opera\_Mini\_5.1.1.apk', and a 'Success' status. A yellow callout box with black text is overlaid on the right side of the second screenshot, stating 'Opera\_Mini 버전에 유의하세요.'

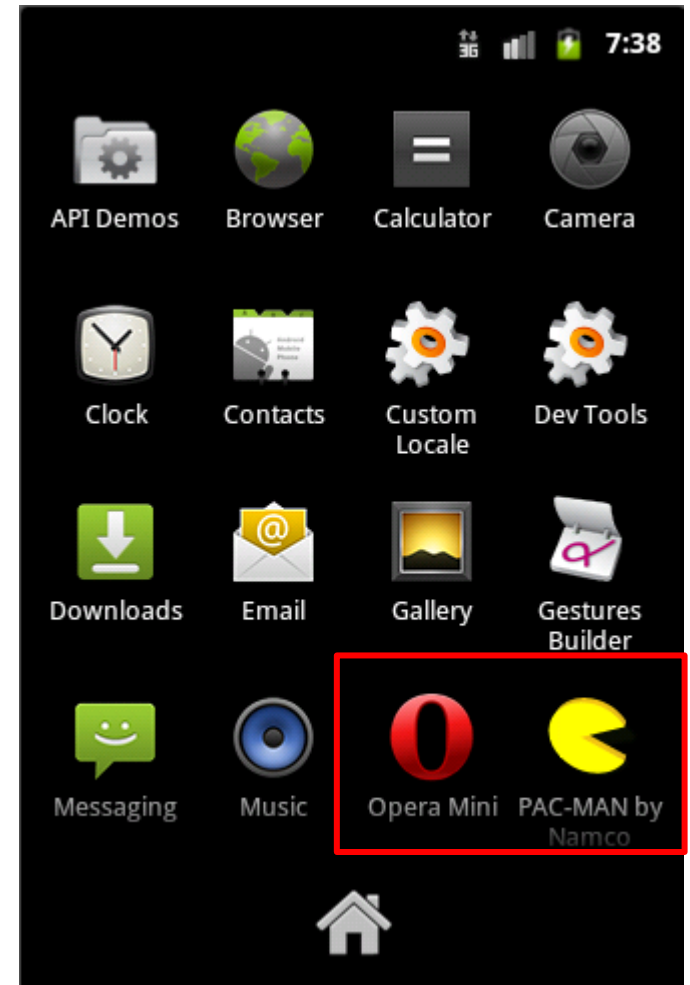
```
C:\windows\system32\cmd.exe
d:\#temp>adb -s emulator-5554 install PAC-MAN_1.0.3.apk
140 KB/s (1215743 bytes in 8.460s)
    pkg: /data/local/tmp/PAC-MAN_1.0.3.apk
Success
d:\#temp>
```

```
C:\windows\system32\cmd.exe
d:\#temp>adb -s emulator-5554 install Opera_Mini_5.1.1.apk
149 KB/s (913875 bytes in 5.979s)
    pkg: /data/local/tmp/Opera_Mini_5.1.1.apk
Success
d:\#temp>
```

**Opera\_Mini 버전에 유의하세요.**

## 응용 프로그램 설치 및 확인

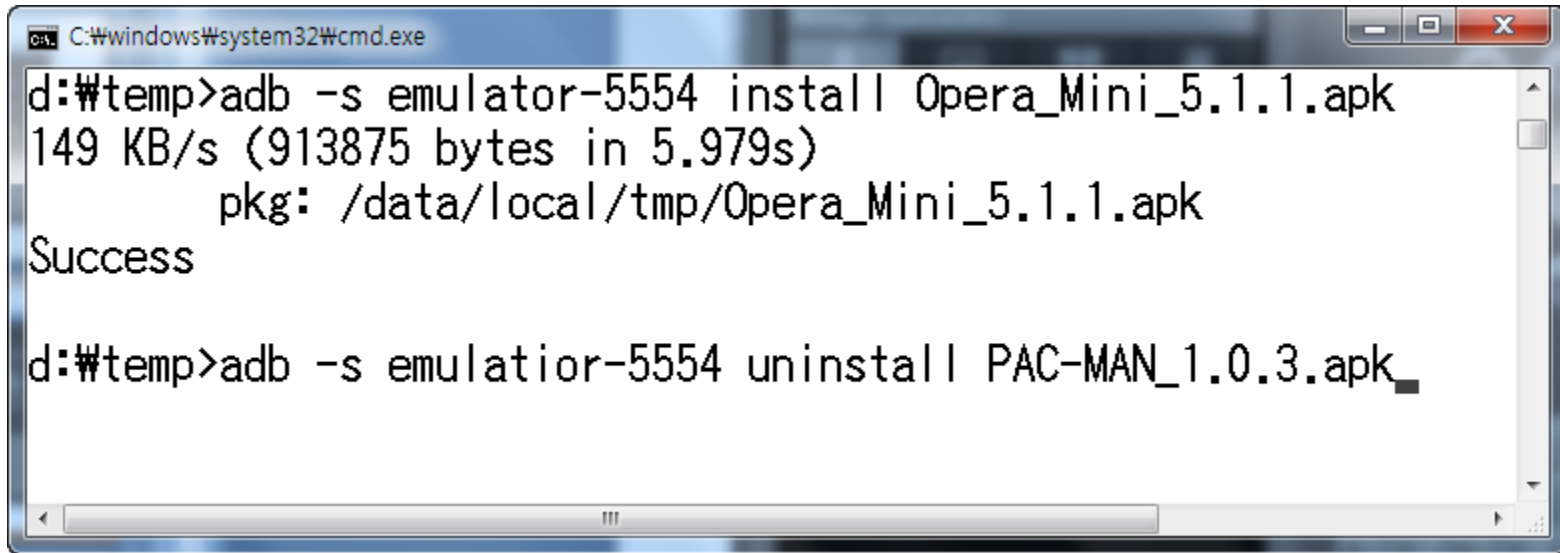
```
C:\windows\system32\cmd.exe - adb -s emulator-5554 s...
c:\windows>adb -s emulator-5554 shell
# ls /data/app
ls /data/app
com.NamcoNetworks.PacMan-1.apk
com.opera.mini.android-1.apk
com.example.android.snake-1.apk
test_type_traits_host
test_ios_pos_types_host
test_streambuf_host
```



# Settings – Manage applications/ Menu – Manage applications



## 응용 프로그램 제거



```
C:\windows\system32\cmd.exe
d:\temp>adb -s emulator-5554 install Opera_Mini_5.1.1.apk
149 KB/s (913875 bytes in 5.979s)
    pkg: /data/local/tmp/Opera_Mini_5.1.1.apk
Success

d:\temp>adb -s emulator-5554 uninstall PAC-MAN_1.0.3.apk_
```

만일 adb 에 문제가 발생하면, 서버를 죽였다가 다시 실행함  
(emulator를 자주 종료 시키고 문제가 발생하면...)

adb kill-server

adb start-server

(공백에 주의 할 것)



## DDMS (교재 304 페이지)

DDMS - AndroidFirst/src/exam/AndroidFirst/AndroidFirst.java - Eclipse

File Edit Run Source Refactor Navigate Search Project Window Help

Devices

Name	Online	AVD2.2 [2...
emulator-5554	Online	AVD2.2 [2...
system_process	69	8600
jp.co.omronsoft.openwnn	134	8601
com.android.phone	139	8602
com.android.launcher	176	8603
com.android.alarmclock	215	
com.android.quicksearchbox	247	

Emulator Control

Telephony Status

Voice: home Speed: Full

Data: home Latency: None

Telephony Actions

Incoming number:

LogCat

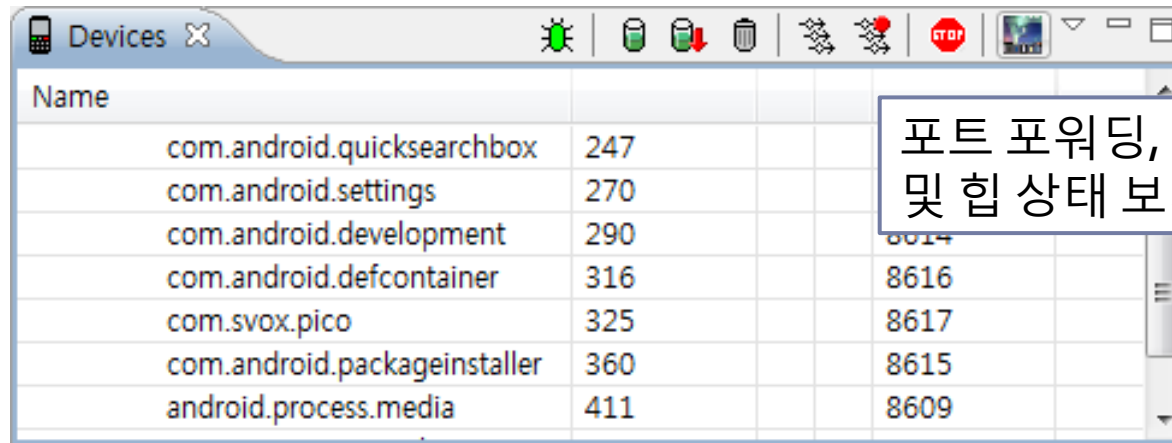
Log

Filter:

Android SDK Content Loader

<ADT와 함께 설치되는 플러그인>  
포트 포워딩, 화면 캡처, 스레드 및 힙 상태  
보기, 장비의 파일 관리, 디버그 로그 보기,  
통화 및 문자 메시지 에뮬레이션 등이 포함  
되어 있음

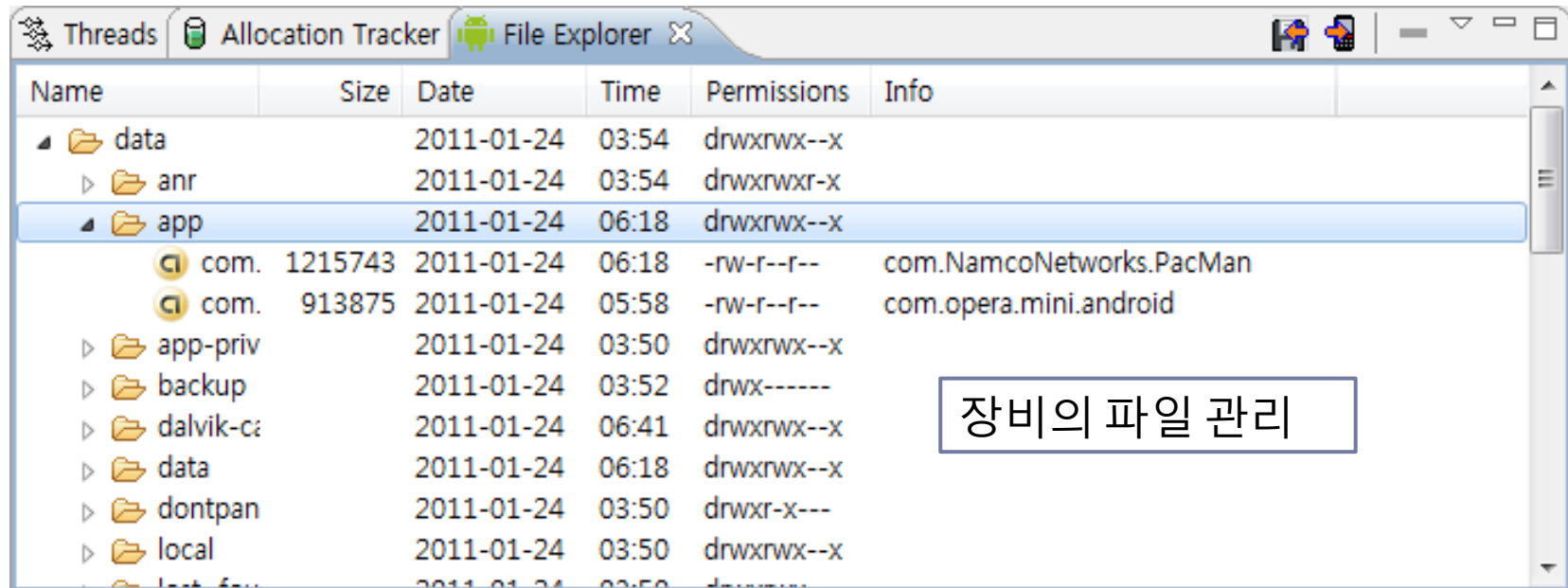
# DDMS



Devices

Name			
com.android.quicksearchbox	247		
com.android.settings	270		
com.android.development	290		
com.android.defcontainer	316	8616	
com.svox.pico	325	8617	
com.android.packageinstaller	360	8615	
android.process.media	411	8609	

포트 포워딩, 화면 캡처, 스레드  
및 힙 상태 보기



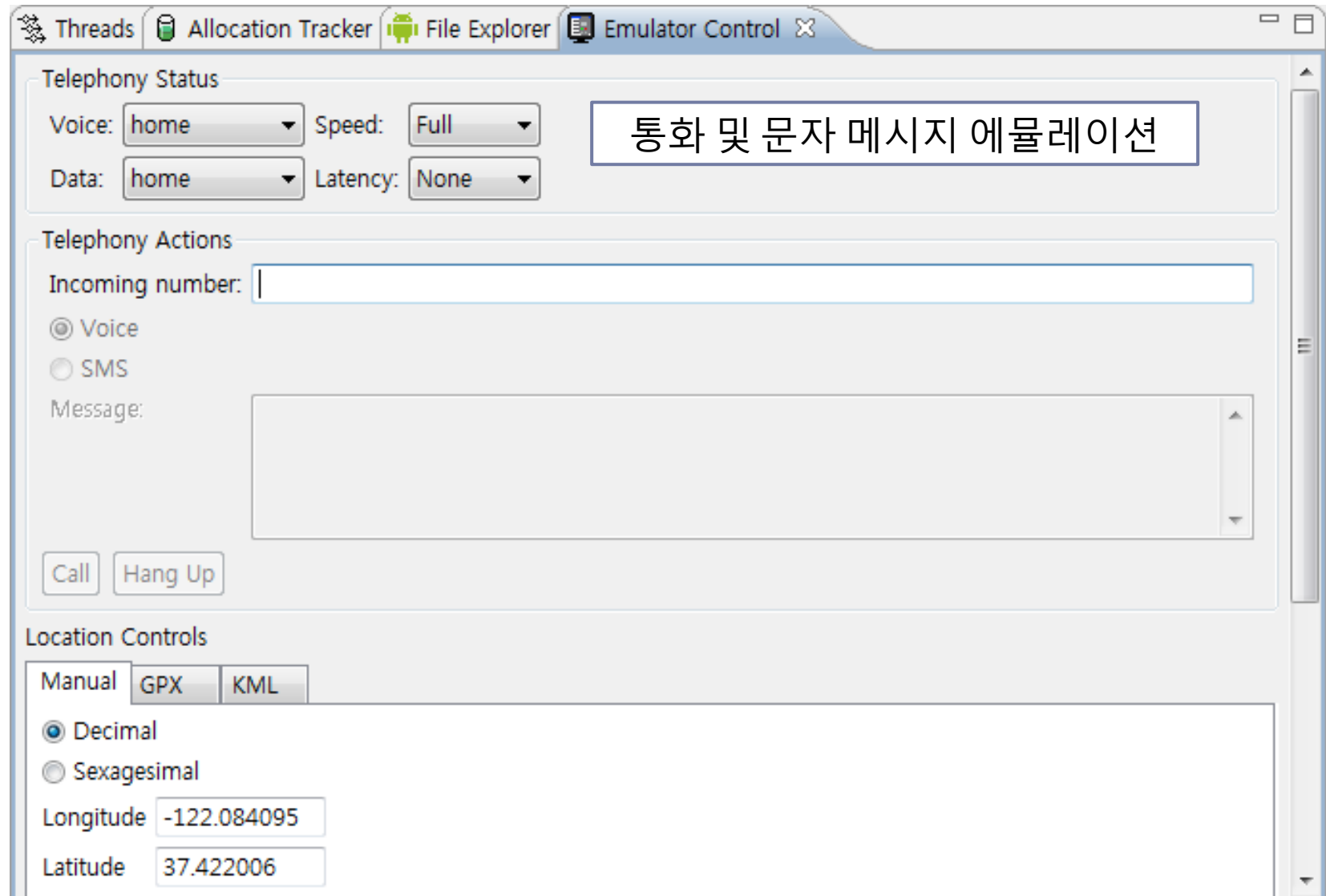
Threads Allocation Tracker File Explorer

Name	Size	Date	Time	Permissions	Info
data		2011-01-24	03:54	drwxrwx--x	
anr		2011-01-24	03:54	drwxrwxr-x	
app		2011-01-24	06:18	drwxrwx--x	
com.	1215743	2011-01-24	06:18	-rw-r--r--	com.NamcoNetworks.PacMan
com.	913875	2011-01-24	05:58	-rw-r--r--	com.opera.mini.android
app-priv		2011-01-24	03:50	drwxrwx--x	
backup		2011-01-24	03:52	drwx-----	
dalvik-cache		2011-01-24	06:41	drwxrwx--x	
data		2011-01-24	06:18	drwxrwx--x	
dontpan		2011-01-24	03:50	drwxr-x---	
local		2011-01-24	03:50	drwxrwx--x	
lost-found		2011-01-24	03:50	drwxrwx--x	

장비의 파일 관리



# DDMS



The screenshot shows the DDMS (Dalvik Debug Monitor Service) interface, specifically the Emulator Control tab. The interface is divided into several sections:

- Threads**: A tab at the top of the window.
- Allocation Tracker**: A tab at the top of the window.
- File Explorer**: A tab at the top of the window.
- Emulator Control**: The active tab, showing various controls for the emulator.

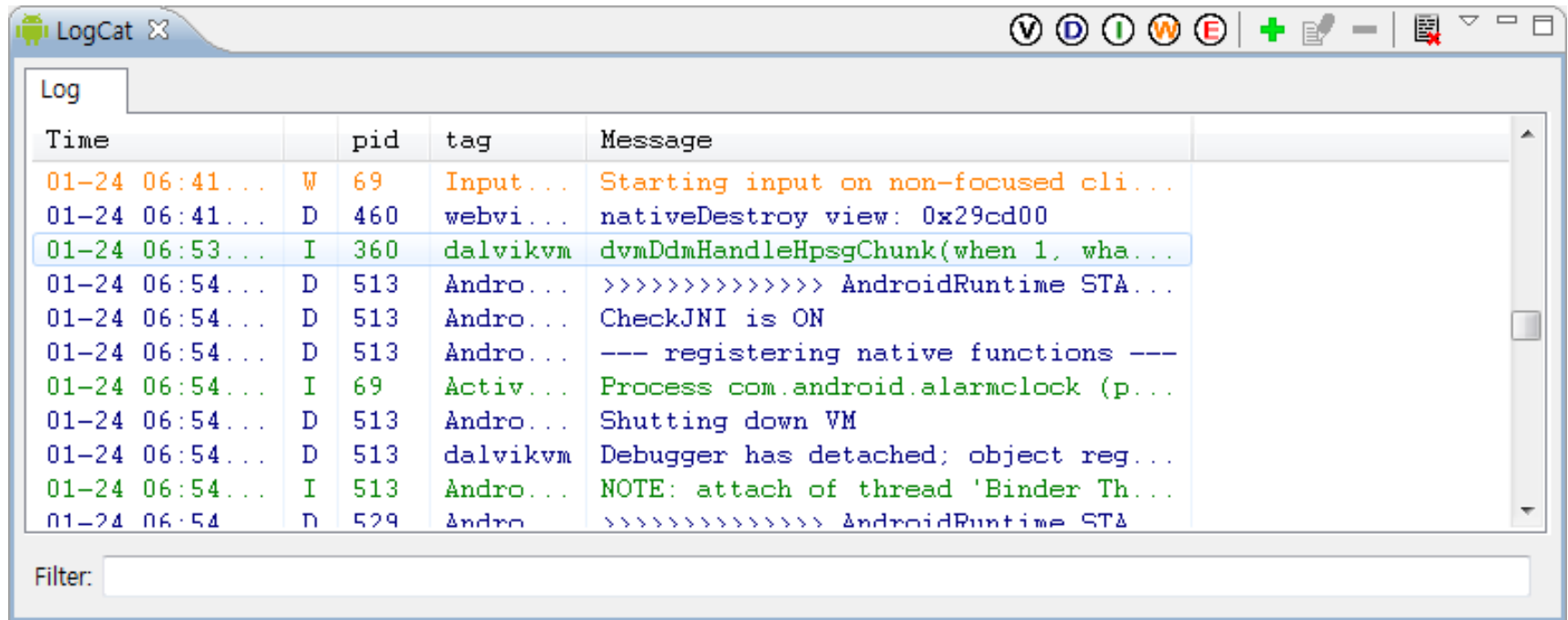
The Emulator Control tab is divided into three main sections:

- Telephony Status**: Contains dropdown menus for Voice (home), Speed (Full), Data (home), and Latency (None).
- Telephony Actions**: Contains a text input field for Incoming number, radio buttons for Voice and SMS, a text area for Message, and buttons for Call and Hang Up.
- Location Controls**: Contains tabs for Manual, GPX, and KML, radio buttons for Decimal and Sexagesimal, and text input fields for Longitude (-122.084095) and Latitude (37.422006).

A text box with the Korean text "통화 및 문자 메시지 에뮬레이션" (Call and SMS message emulation) is overlaid on the Telephony Status section.

# DDMS

## 디버그 로그 보기



The screenshot shows the LogCat window in Android Studio. The window title is "LogCat" with a close button. The toolbar includes icons for Verbose (V), Debug (D), Info (I), Warn (W), Error (E), a plus sign for more filters, a clipboard, and window management icons. The log list is displayed with columns for Time, pid, tag, and Message. The log entry "dvmDdmHandleHpsgChunk(when 1, wha..." is selected and highlighted in blue. Below the log list is a "Filter:" text box.

Time	pid	tag	Message
01-24 06:41...	W 69	Input...	Starting input on non-focused cli...
01-24 06:41...	D 460	webvi...	nativeDestroy view: 0x29cd00
01-24 06:53...	I 360	dalvikvm	dvmDdmHandleHpsgChunk(when 1, wha...
01-24 06:54...	D 513	Andro...	>>>>>>>>>>>> AndroidRuntime STA...
01-24 06:54...	D 513	Andro...	CheckJNI is ON
01-24 06:54...	D 513	Andro...	--- registering native functions ---
01-24 06:54...	I 69	Activ...	Process com.android.alarmclock (p...
01-24 06:54...	D 513	Andro...	Shutting down VM
01-24 06:54...	D 513	dalvikvm	Debugger has detached; object reg...
01-24 06:54...	I 513	Andro...	NOTE: attach of thread 'Binder Th...
01-24 06:54...	D 529	Andro...	>>>>>>>>>>>> AndroidRuntime STA...



# Android 개발 기초

## Android Project – 01\_HelloAndroid

Project name: **01\_HelloAndroid**

### Contents

- ☒ Create new project in workspace
- ☐ Create project from existing source

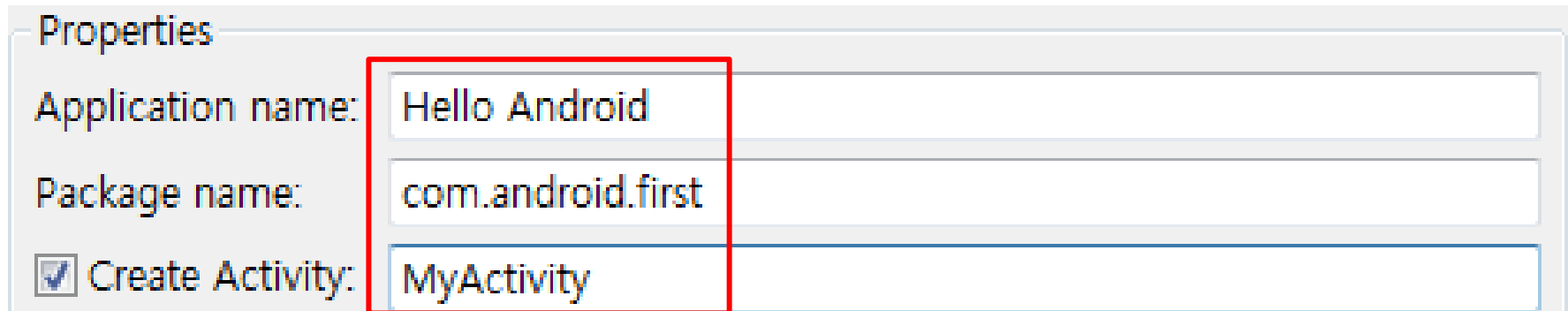
### Build Target

Target Name	Vendor	Platform	API ...
<input type="checkbox"/> Android 1.5	Android Open Source Project	1.5	3
<input type="checkbox"/> Android 1.6	Android Open Source Project	1.6	4
<input type="checkbox"/> Android 2.1-upd...	Android Open Source Project	2.1-upd...	7
<input type="checkbox"/> Android 2.2	Android Open Source Project	2.2	8
<input checked="" type="checkbox"/> <b>Android 2.3.3</b>	Android Open Source Project	2.3	<b>10</b>
<input type="checkbox"/> Google APIs	Google Inc.	2.3	9

Standard Android platform 2.3

## Android Project – SY\_HelloAndroid

---



Properties

Application name: Hello Android

Package name: com.android.first

☒ Create Activity: MyActivity

**Project name** : Eclipse에서 관리하는 프로젝트 이름

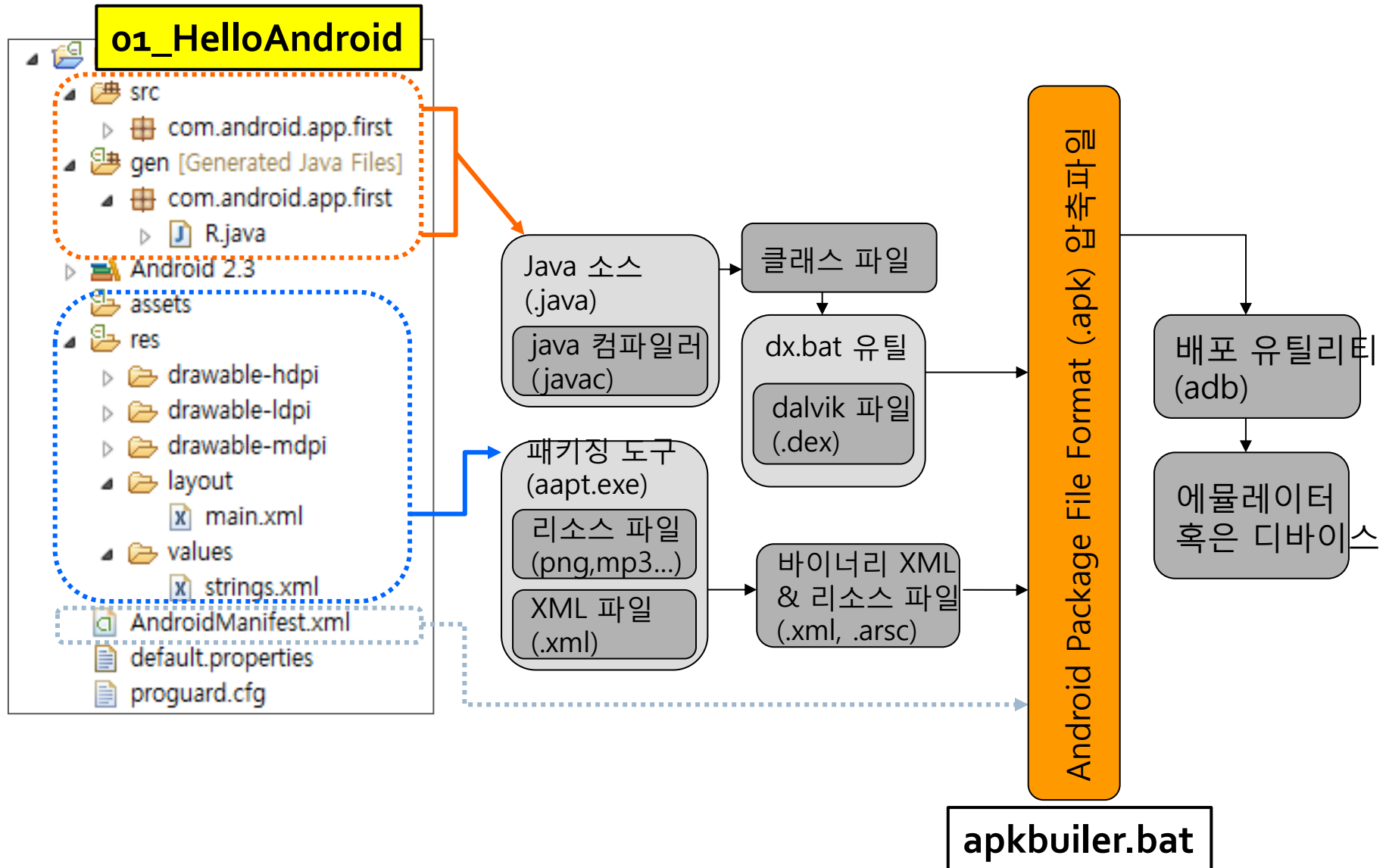
**Application name** : 설치될 때의 이름으로 공백이 있어도 됨

**Package name** : 디렉토리, data/app,  
data/data/package에 사용되는 이름

**Create Activity** : Activity를 생성하고자 할 때 사용



# Android Project - 01\_HelloAndroid



## Android Application 개발 시 project 구조

---

**Source** : 동적인 처리 (\*.java)

- src/ 사용자 작성 : class 단위
- gen/ ADT가 자동으로 작성 : R.java (Source와 Resource 연결 기능)

**Resource** : 정적인 요소 (\*.xml)

- res/ 값(문자열, 배열, 색상값, 크기...), 화면구성, Animation 정보, Image(\*.png)
- drawable, layout, value, animation, xml // Directory 단위 관리

**AndroidManifest.xml** : Application 정보가 들어 있음

- 실행 시 본 파일을 먼저 열어서 파일에 따라 실행하게 됨
- Application 전체 정보
- Application 구성 요소에 대한 정보
- 권한 정보
- Library 사용 정보 (Android SDK가 제공하지 않는 외부 라이브러리)

## Android Class Library

---



## Android .apk 생성 과정

---

1. Source Compile
  - Java Source 및 R.java를 Compile해서 \*.class를 생성
  - **dx.bat**을 이용해서 Android 에서 사용할 수 있는 \*.dex 형태의 파일로 변환함
  - **classes.dex** 가 결과물로 얻어짐
2. Resource Compile and Packing
  - Resource 중에는 Compile 할 수 있는 것(anim, layout, xml ...)과 할 수 없는(raw, assets) 것이 있음
  - compile 할 수 있는 것은 **aapt.exe**를 이용해서 **resources.arsc**로 만들고, compile 할 수 없는 것은 **package** 형태로 묶어 주어야 함
  - **resources.arsc** 가 결과물로 얻어짐
3. **AndroidManifest.xml** file
4. **Application** 에 대한 인증 정보

1~4의 파일을 apkbuilder.bat에 의해 묶어 **\*.apk** 파일이 생성됨





## netstat

---

- network에서 특정 port가 사용 중인지를 확인할 수 있음
- **netstat -nao** 를 수행하여 사용 중인 port와 해당 process의 pid 정보를 얻음
- [Windows 작업 관리자]에서 프로세스의 pid를 볼 수 있도록 한 뒤, netstat를 이용해서 얻는 pid 번호의 프로세스를 제거함
- [Windows 작업 관리자]의 메뉴에서 **[보기]-[열 선택]**을 이용하면 pid 정보가 표시되도록 할 수 있음



## string.xml // 수정 및 추가

---

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="hello">Hello World!! *^^* </string>
    <string name="app_name">Hello Android</string>
    <string name="btn_txt_finish">Finish Hello Android</string>
</resources>
```



## main.xml // 수정 및 추가

---

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://~" ... >
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:text="@string/hello"
    />
    <Button
        android:id="@+id/btn_finish"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/btn_txt_finish"
    />
</LinearLayout>
```



## aapt에 의한 R.java 생성

### 01\_HelloAndroid

#### src

##### com.android.first

MyActivity.java

#### gen [Generated Java Files]

##### com.android.first

R.java

#### Android 2.3.3

##### assets

#### res

##### drawable-hdpi

icon.png

##### drawable-ldpi

##### drawable-mdpi

##### layout

main.xml

##### values

strings.xml

AndroidManifest.xml

```
package com.android.first;
```

```
public final class R {  
    public static final class attr {  
    }  
    public static final class drawable {  
        public static final int icon=0x7f020000;  
    }  
    public static final class id {  
        public static final int btn_finish=0x7f050000;  
    }  
    public static final class layout {  
        public static final int main=0x7f030000;  
    }  
    public static final class string {  
        public static final int app_name=0x7f040001;  
        public static final int btn_txt_finish=0x7f040002;  
        public static final int hello=0x7f040000;  
    }  
}
```

## MyActivity.java //수정 및 추가

---

```
public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.main);  
  
    Button finish = (Button)findViewById(R.id.btn_finish);  
    finish.setOnClickListener(new View.OnClickListener() {  
        public void onClick(View v) { // 메서드  
            finish();  
        }  
    });  
}
```



## Button의 SetOnClickListener

public class Button extends TextView

java.lang.Object

↳ android.view.View

↳ android.widget.TextView

↳ android.widget.Button

method of View class

setOnClickListener(View.OnClickListener l)

anonymous inner class of  
View.OnClickListener interface

```
finish.setOnClickListener(new View.OnClickListener() {  
    public void onClick(View v) { // 메서드  
        finish();  
    }  
});
```

abstract void onClick(View v);

## 리소스 폴더

---

폴더	설명
drawable	이미지 파일, 도형을 정의하는 XML 파일 저장
layout	화면의 레이아웃 저장. 뷰 그룹, 파생 클래스의 배치 상태가 기록 됨
values	문자열, 색상, 배열, 크기, 단순 드로블, 스타일 등 여러 가지 값들을 정의하는 XML 파일 저장
menu	메뉴 구성 파일 저장
xml	실행 중 읽어서 사용할 임의의 XML 파일 저장
raw	임의의 이진 파일 저장
anim	애니메이션 방식을 정의하는 XML 파일 저장



# Run Configurations

Name:

Android Target Common

Project:

**01\_HelloAndroid**

Launch Action:

☒ Launch Default Activity

Name:

Android Target Common

Deployment Target Selection Mode

☐ Manual

☒ Automatic

Select a preferred Android Virtual Device for deployment:

AVD Name	Target Name	Platform
<input checked="" type="checkbox"/> AVD2.3	Android 2.3	2.3

Click [Start] Button

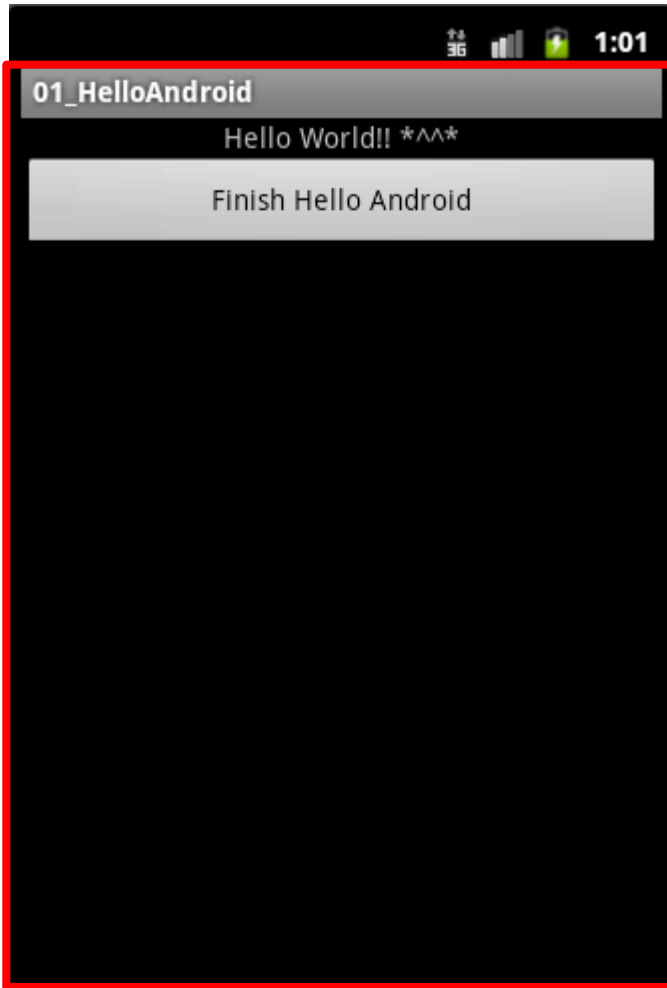
Uploading  
Installing  
Success  
Starting activity





# Hello Android 실행 화면

---



상태표시줄은 Activity가 관할하는 영역이  
아님

Activity 1개가 상태표시줄을 제외한 전체  
영역을 관할함



## adb를 이용한 확인 (DDMS의 File Explorer로도 확인 가능 - 그림)

---

```
adb -s emulator-5554 shell
```

```
# ls /data/app
```

```
com.android.first-1.apk
```

```
#ls /data/data/com.android.first
```

```
ls /data/data/com/android.first
```

```
lib
```

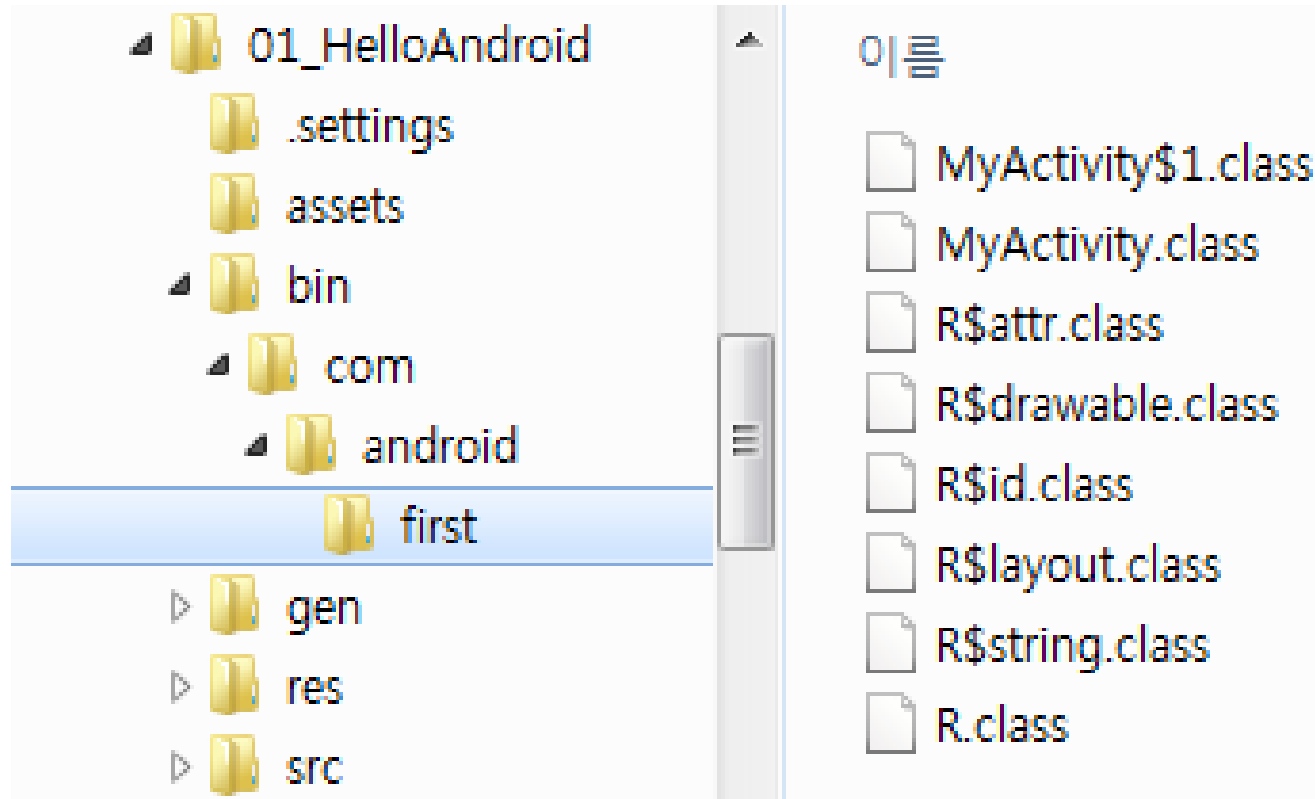
/data/app : Uploading

/data/data/com.android.first : Installing



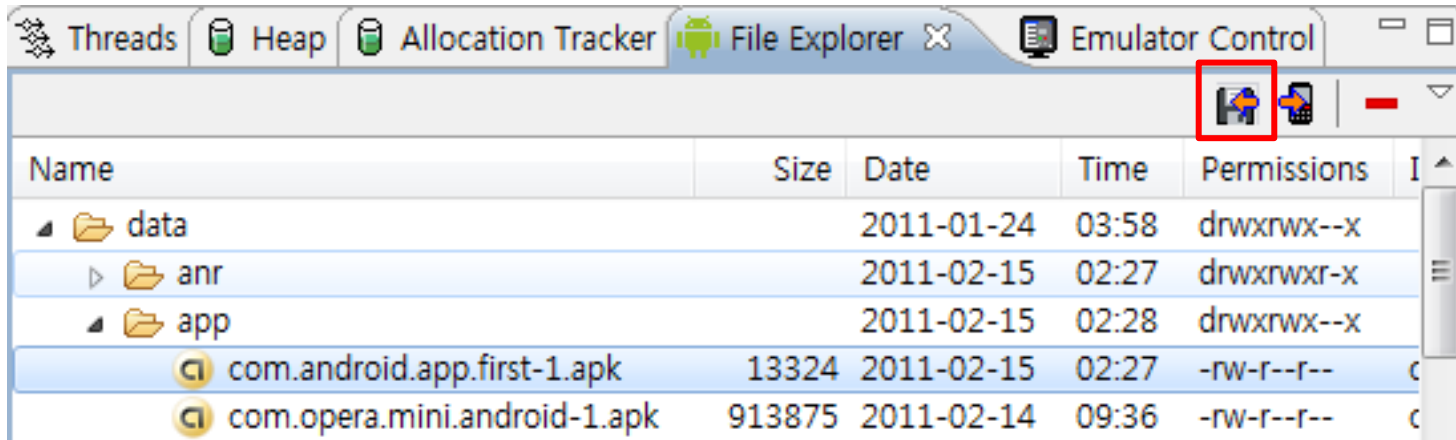
## compile된 class 파일들

/01\_HelloAndroid/bin/com/android/first








## apk 파일 다운로드 및 구성 확인

File Explorer에서 다운 또는 cmd 창에서 pull 사용



**adb pull /data/app/com.android.app.first-1.apk c:\temp\first.apk**

 META-INF	[META-INF] : 인증 정보
 res	[res] : packing된 resource
 AndroidManifest.xml	AndroidManifest.xml
 classes.dex	classes.dex : java로 구현된 application이라는 뜻
 resources.arsc	resources.arsc : Resource file

## apk (android package)

---

apk는 실행 파일이 아닌 **설치 대상 파일**임.

Package Manager, Resource Manager, Activity Manager를 통해 실행 됨

별도의 실행파일 없이 **apk 파일 내부에 실행 관련 정보**를 이용하여  
실행하며 system/app, data/app 에 Uploading 되며,  
동작을 위해 system/data, data/data/어플 package 이름으로 설치됨



## AndroidManifest.xml 구성정보

---

1. Application 정보 : package 명, version 정보  
<manifest> element, 한번만 나오며 전체를 감싸주어야 함
2. Application 구성 요소  
<application> element, 적어도 1개 나와야 하며, 화면 1개당 activity 1개로 표현. application에는 activity, service, BroadcastReceiver, contentProvider 등이 포함될 수 있음. 포함되지 않으면 실행되지 않음
3. 권한정보  
<permission> : 내가 외부 것을 사용할 때  
<user-library>: 내 것을 외부에서 사용할 때
4. 외부 Library
5. SDK version 정보 (최소, 최대) <uses-sdk>
6. <intent filter> - activity, service, broadcast Receiver 만 부여할 수 있는 것으로 외부에서 전달되는 Message 수신 가능 여부를 의미함  
이것을 지우면 실행, 설치가 되지 않음
7. <action>, <category> ...



The screenshot shows the d.android.com website with the 'Dev Guide' tab selected. The left sidebar contains a navigation menu with categories like 'Android Basics', 'Framework Topics', and 'The AndroidManifest.xml File'. The main content area is titled 'The AndroidManifest.xml File' and contains an introductory paragraph and a bulleted list of what the manifest does. A right-hand box titled 'In this document' lists links to various sections of the document.

ANDROID  
**developers**

English [Android.com](#)

search developer docs

Home SDK **Dev Guide** Reference Resources Videos Blog

**Android Basics**  
What Is Android?

**Framework Topics**  
Application Fundamentals  
▶ User Interface  
▶ Application Resources  
Intents and Intent Filters  
▶ Data Storage  
Content Providers  
Security and Permissions  
▼ **The AndroidManifest.xml File**  
    <action>  
    <activity>  
    <activity-alias>  
    <application>  
    <category>  
    <data>  
    <grant-uri-permission>  
    <instrumentation>  
    <intent-filter>  
    <manifest>

## The AndroidManifest.xml File

Every application must have an AndroidManifest.xml file (with precisely that name) in its root directory. The manifest presents essential information about the application to the Android system, information the system must have before it can run any of the application's code. Among other things, the manifest does the following:

- It names the Java package for the application. The package name serves as a unique identifier for the application.
- It describes the components of the application — the activities, services, broadcast receivers, and content providers that the application is composed of. It names the classes that implement each of the components and publishes their capabilities (for example, which [Intent](#) messages they can handle). These declarations let the Android system know what the components are and under what conditions they can be launched.
- It determines which processes will host application components.
- It declares which permissions the application must have in order to access protected parts of the API and interact with other applications.

**In this document**  
[Structure of the Manifest File](#)  
[File Conventions](#)  
[File Features](#)  
    [Intent Filters](#)  
    [Icons and Labels](#)  
    [Permissions](#)  
    [Libraries](#)