# 15장 안드로이드 마켓



## 시작하면서



#### ◎ 목차

- 15.1 APK 파일
- 15.2 안드로이드 마켓

## 15.1 apk 파일

#### ⊙ 필요한 소프트웨어

- JDK 5.0 이상
- 안드로이드 SDK
- 이클립스

#### ◉ 다운로드할 폴더 생성

- D:\android 폴더 생성
- D:\android\software 폴더 생성



### 15.1 apk 파일



4

- Apk 파일
  - /bin에 프로그램의 액티비티 이름으로 생성

| 😂 bin  |                  |   |                   | - DX                                |
|--|------------------|---|-------------------|-------------------------------------|
| 파일( <u>F</u> ) 편집( <u>E</u> ) 보기( <u>V</u> ) 즐겨찾기  | ( <u>A</u> ) 도구( | <u>T</u> ) 도움말( <u>H</u> )  |                   | At .                                |
| 🌀 뒤로 • 🕥 · 🏂 🔎 검색  | 🕞 폴더             | ≩ ≫ X Ю   | <u>.</u> .        |                                     |
| 주소(D) 🛅 D:₩android₩workspace₩H   | elloAndroid      | ₩bin  |                   | 🗸 🏹 이동                              |
| 폴더   | ×                | 이름 🔺  | 37                | 종류                                  |
| <ul> <li>android</li> <li>android-sdk</li> <li>eclipse</li> <li>ndk</li> <li>software</li> <li>workspace</li> <li>metadata</li> <li>android</li> <li>HelloAndroid</li> <li>,settings</li> <li>assets</li> <li>bin</li> <li>com</li> <li>e com</li> <li>e metadata</li> </ul> |                  | classes, dex<br>classes, dex<br>HelloAndroid, apk<br>resources, ap_ | 3KB<br>8KB<br>5KB | 파일 폴더<br>DEX 파일<br>APK 파일<br>AP_ 파일 |
| <  |                  | <   |                   | >                                   |

[그림 15-1] Helloandroid.apk 생성

### 15.1 apk 파일



- ⊙ 압축해제
  - Unzip 사용

| 🖆 temp                                       |           | _               |                                |            |            |        |
|--|-----------|-----------------|--------------------------------|------------|------------|--------|
| 파일( <u>F</u> ) 편집( <u>E</u> ) 보기( <u>V</u> ) | 즐겨찾기      | ( <u>A</u> ) 도구 | <sup>1</sup> ( <u>T</u> ) 도움말( | <u>H</u> ) |            | R.     |
| ③ 뒤로 - ③ - 🏠                                 | ,○ 검색     | (6) ¥C          |                                | ×ŋ         | · · ·      |        |
| 주소(D) 🛅 D:₩android₩wo                        | rkspace₩⊦ | lelloAndro      | id₩bin₩temp                    |            |            | 🗸 🏹 이동 |
| 이름 🔺   | 12        | 크기              | 종류                             | 수정         | 성한 날짜      |        |
|  |           |                 | 파일 폴더                          | 2010       | )-06-20 오후 |        |
| 🛅 res  |           | 10-20-20-20     | 파일 폴더                          | 2010       | )-06-20 오후 |        |
| 📄 🕋 AndroidManifest, xml                     |           | 2KB             | XML 문서                         | 2010       | )-05-21 오후 |        |
| 🖬 classes, dex                               |           | 3KB             | DEX 파일                         | 2010       | )-05-21 오후 |        |
| 🖬 resources, arsc                            |           | 2KB             | ARSC 파일                        | 2010       | )-05-21 오후 |        |
|  |           |                 |                                |            |            |        |
|  |           |                 |                                |            |            |        |
|  |           |                 |                                |            |            |        |

[그림 15-2] apk 파일의 압축 해제

#### ● 마켓 판매 및 관리 과정



#### [그림 15-3] 안드로이드 마켓 및 관리 과정

- ◎ 프로그램 작성
- ◉ 프로그램 작성 후 빌드
  - 디버그(debug) 모드
  - 배포모드 (release mode)

#### ◉ 빌드 도구

• ant





- ◉ 애플리케이션 개발자로 등록하기
  - 안드로이드 마켓 사이트 방문 (http://www.android.com/market/)
  - 등록비 지불
  - Android Market Developer Distribution Agreement에 동의



안드로이드 마켓을 방문하여 사용자 프로필, 비용 지불, 사용자 동의 등을 거쳐 등 록을 해본다.



#### ● [실습 15-2] apk 파일용 비밀키 생성하기

(1) 이클립스 /workspace/(project name)/bin 디렉터리의 애플리케이션을 안 드로이드 마켓에 업로드해본다. 어떤 오류 메시지가 나오는지 확인해본다. 키를 생성하고 apk를 서명하는 표준 도구인'keytool.exe'와'jarsigner.exe'를 사용한다. 특히 keytool은 개인키와 인증서를 관리한다.

>keytool -genkey -keystore helloworld.keystore -alias helloworld keyalg RSA -validity 10000



10

### ◉ [실습 15-2] apk 파일용 비밀키 생성하기

(2) mkdir d:₩android₩workspace₩Keytools

애플리케이션 인증 작업을 하기 위해 미리 필요한 디렉터리를 만들어 둔 다. 기존에 있던'SDK₩android₩workspace'디렉터리 아래에 작업 디렉터리 를 만든다.

(3) cd d:₩android₩workspace

성된 것 을 확인한다.

[그림 15-5]

작업 디렉터리 생성

[그림 15-5]처럼 D:<SDK>₩android₩workspace₩Keytools 디렉터리가 생

| 파일(E) 편집(E) 보기(Y) 즐겨찾기(A) 도구(T) 도움말(H)<br>중 뒤로 · · · · · · · · · · · · · · · · · ·  | eytools   |                               |               |               |
|---|---|-------------------------------|---------------|---------------|
| STR · · · · · · · · · · · · · · · · · · ·   | 』(E) 편집(E) 보기(⊻) 즐겨찾기(  | <u>A</u> ) 도구( <u>T</u> ) 도움물 | 발( <u>H</u> ) | AL            |
| 주소(D) D:\#android\#workspace\#Keytools  | ) 뒤로 🔹 🕥 🕤 🏂 🎾 검색 🏾   | 🚱 हत 🕼 🕉                      | ▷ 🗙 🍤 📖 -     |               |
| 폴더     이름     크기 종류     수정한 날짜            android<br>android-sdk<br>configuration<br>dropins<br>features<br>p 2<br>b        plugins<br>readme<br>m dk<br>software<br>workspace     이름     크기 종류     수정한 날짜  | (D) 🛅 D:\#android\#workspace\Ke   | eytools                       |               | 💌 🔁 이동        |
| ■ android A P P P P P P P P P P P P P P P P P P   |   | × 이름 🔺                        | 크기 종류         | 수정한 날짜        |
| <ul> <li>metadata</li> <li>mandroid</li> <li>ma</li></ul> | <ul> <li>android</li> <li>android-sdk</li> <li>configuration</li> <li>dropins</li> <li>features</li> <li>p2</li> <li>plugins</li> <li>readme</li> <li>ndk</li> <li>software</li> <li>workspace</li> <li>.metadata</li> <li>android</li> <li>Hello-Android</li> <li>Hello-Jni</li> <li>Keytools</li> <li>keys</li> </ul> | Keys                          | 마님 즐너         | 2UIU-U6-27 포후 |



11

### ● [실습 15-2] apk 파일용 비밀키 생성하기

(4) Keytools> keytool -genkey -keystore helloAndroid.keystore -alias helloAndroid -keyalg RSA -validity 10000

| 🔤 Command Shell - keytool -genkey -keystore helloAndroid.keystore -alias hell 💶                                     | □×       |
|---|----------|
| D:\android\workspace\Keytools>keytool -genkey -keystore helloAndroid.keystore                                       | _a       |
| keystore 암호를 입력하십시오:<br>새 암호를 다시 입력하십시오:<br>의치하지 않습니다. 다시 시도하십시오  |          |
| keystore 암호를 입력하십시오:<br>Keystore 암호를 입력하십시오:<br>Keystore 암호갈 너무 짧습니다. 여섯 글자 이상이어야 합니다.                              |          |
| keystore 임오늘 입력이십시오:<br>새 암호를 다시 입력하십시오:<br>이름과 성을 입력하십시오.  |          |
| [Unknown]: Sugoog Shon<br>조직 단위 이름을 입력하십시오.<br>[Unknown]: University of Suyon                                       |          |
| 조직 이름을 입력하십시오.  |          |
| [Unknown]: Department of ITE<br>구/군/시 이름을 입력하십시오?   |          |
| [Unknown]: Hwasung city<br>시/도 이름을 입력하십시오.  |          |
| [Unknown]: Wauri Bongdam-eup<br>이 조직의 두 자리 국가 코드를 입력하십시오.<br>[Unknown]: 82  |          |
| CN=Sugoog Shon, OU=University of Suwon, O=Department of ITE, L=Hwasung city, S<br>Wauri Bongdam-eup, C=82이(가) 맞습니까? | T =      |
| [ [아니오]: y_   | <b>_</b> |

[그림 15-6] keytool 사용 예



### ◉ [실습 15-2] apk 파일용 비밀키 생성하기

(5) Keytools>dir

[그림 15-7]에서는 ₩keytools 디렉터리에 자체 서명된 인증서 'helloAndroid .keystore' 파일 생성을 보여준다.

| 👞 Comman                       | d Shell                                  |                            |  | - 🗆 × |
|--------------------------------|--|----------------------------|--|-------|
| D:₩android₩<br>D 드라이브<br>볼륨 일련 | workspace₩Key<br>의 볼륨에는 (<br>번호: 449B-72 | tools>dir<br>))름이 없;<br>4C | 슬니다.                                   |       |
| D:₩android                     | ₩workspace#Ke                            | ytools []                  | 렉터리                                    |       |
| 2010-06-27                     | 오후 01:53                                 | <dir></dir>                | -                                      |       |
| 2010-06-27<br>2010-06-27       | 오후 01:53<br>오후 01:53                     | <dir></dir>                | 1,446 helloAndroid.keystore            |       |
| 2010-06-27                     | 오후 01:43<br>1개 파일<br>3개 디렉터              | <dir><br/>러리 9,6</dir>     | keys<br>1,446 바이트<br>60,178,432 바이트 남음 |       |
| D∶₩android₩                    | workspace#Key                            | tools>_                    |  |       |

[그림 15-7] HelloAndroid.keystore 파일 생성 확인



#### ◎ [실습 15-3] 애플리케이션 서명하기

 (1) 디버그 키로 자기서명하기 확인 : 이클립스에서 window - Preferences -Android - Build에 가보면 Default debug keystore가 지정되어 있는 것을 알수 있다.

에뮬레이터에서는 디버그 모드로 애플리케이션이 동작하며, SDK 도구에 의해 디버그키로 자동으로 서명이 이루어진다. 빌드 작업을 할 때 안드로이드 빌드 도구가 디버<u>그키로 apk 파일을 자동으로 서명하도록 도와준다.</u>

| = Preferences  | Y  |  |
|--|--|--|
| type filter text   | Build  |  |
| General     Android     DDMS     DDMS     Launch     LogCat     Usage Stats     Ant     Help     Install/Update     Java     Run/Debug     Tasks     Team     Usage Data Collector | Build Settings:<br>✓ Automatically refresh Resources and Assets folder on buil<br>✓ Force error when external jars contain native libraries<br>Build output<br>③ Silent<br>○ Normal<br>○ Verbose<br>Default debug keystore:<br>C:\Documents and Settings\Adr<br>Custom debug keystore: | ild<br>ministrator₩,android₩debug,keysto<br> |
|  |  | (Restore <u>D</u> efaults) ( <u>A</u> pply   |
| ?  |  | OK Cance                                     |

[그림 15-8] Default debug keystore 확인하기



14

◎ [실습 15-3] 애플리케이션 서명하기

(1) 계속

디버그 모드에서 서명 작업을 수행할 때 SDK 도구는 미리 정의된 이름과 비밀번호를 갖는 디버그 keystore/key를 생성한다.

- Keystore 이름:" debug.keystore"
- Keystore 비밀번호:" android"
- Key 별칭:" androiddebugkey"
- Key 비밀번호:" android"
- CN:" CN=Android Debug,O=Android,C=US"



### ◉ [실습 15-4] 미서명 애플리케이션 만들기

(1) 미서명 애플리케이션 만들기: Android Tools → Export Unsigned Application Package

이클립스에서 [그림 15-9]처럼 프로젝트 이름(android)에 마우스를 올려 놓고, 마우스 우측 버튼을 클릭하면 팝업 메뉴가 나타난다. 'Android Tools → Export Unsigned Application Package'를 클릭하여 미서명 애플리케이 션을 만든다.

[그림 15-9] Android Tools → Export Unsigned Application Package

| ੴ. Navigator                                 | 8 -0   | Console 🛛                 |                                  |   |              | ▋・▆・□□゛ |
|--|--|---------------------------|----------------------------------|---|--------------|---------|
|  | @ 🕒 🍄 🎽  | Android                   |                                  |   |              | ~ .     |
| androi<br>□-22 , se<br>□-22 as:<br>□-25 bin  | Ne <u>w</u><br>Go <u>I</u> nto<br>Open in <u>N</u> ew Wir  | ndow                      |                                  |   |              | 6       |
| in ⊕ gen<br>in ⊕ gen<br>in ⊕ gen<br>in ⊕ sro | E Copy   |                           |                                  |   |              | >       |
| , cli<br>, cli                               | Move.  |                           | ld,java                          | 🕽 hellojni,java                         | 🕽 🕽 Led,java |         |
| An   | Rena <u>m</u> e  |                           | droid;                           |   |              | ~       |
| i det<br>i ⊡ 12 Hello-<br>i ⊡12 Led          | f <mark>≥ I</mark> mport<br>⊿ Exp <u>o</u> rt              |                           | tivity                           | 2                                       |              |         |
|  |  | Projects                  | <pre>extends Activity (</pre>    |   |              |         |
|  | Validate<br><u>R</u> un As<br><u>D</u> ebug As             |                           | te (save<br>• <sup>:w(R.la</sup> | dInstanceState)<br>yout. <i>main</i> ); | 2            |         |
|  | T <u>e</u> am<br>Comp <u>a</u> re With<br>Restore from Loc | al History                | •                                |   |              |         |
|  | Android Tools  |                           | J <sup>C</sup> <sub>U</sub> New  | Test Project                            | 1            |         |
|  | Source   |                           | Mew I                            | Resource File                           |              |         |
|  | P <u>r</u> operties  |                           | Expor                            | t Signed Application I                  | Package      | ~       |
| <  | 🕭 Remove from Co   | ntext Ctrl+Alt+Shift+Down | Expor                            | t Unsigned Applicatio                   | in Package   | 2       |
|  | 🗒 🔍 android  |                           | Fix Pr                           | oject Properties                        |              |         |



### ◉ [실습 15-4] 미서명 애플리케이션 만들기

(1) 미서명 애플리케이션 만들기: Android Tools → Export Unsigned
 Application Package (계속)



#### [그림 15-10] 미서명 APK 파일 저장 시 메시지



17

#### ◉ [실습 15-4] 미서명 애플리케이션 만들기

#### (2) 미서명된 애플리케이션 확인 윈도우 탐색기로 파일이 저장되었는지 확인하면 [그림 15-11]과 같이 나타 난다.

| 🖿 Keytools  |     |   |             |                                |  |
|---|-----|---|-------------|--------------------------------|--|
| 파일(E) 편집(E) 보기(⊻) 즐겨찾기( <u>A</u> )  | 도구( | <u>T</u> ) 도움말( <u>H</u> )                  |             |                                | Ar   |
| 🌀 뒤로 🔹 🅥 - 🎓 🎾 검색 🕞   | 폴더  | ₿ <b>```X '</b> 9                           |             |                                |  |
| 주소(D) 🗁 D:₩android₩workspace₩Keytool  | ls  |   |             |                                | ♥ → 미동   |
| 폴더  | ×   | 이름 🔺  | 크기          | 종류                             | 수정한 날짜   |
| <ul> <li>workspace</li> <li>metadata</li> <li>android</li> <li>HelloAndroid</li> <li>HelloJni</li> <li>Hello-Jni</li> <li>Keytools</li> <li>keys</li> <li>LedJni</li> </ul> |     | raikeys<br>■ helloAndroid<br>■ android, apk | 2KB<br>12KB | 파일 폴더<br>KEYSTORE 파일<br>APK 파일 | 2010-06-27 오후 1:43<br>2010-06-27 오후 1:53<br>2010-06-27 오후 2:47 |

#### [그림 15-11] 미서명된 애플리케이션



18

#### ◉ [실습 15-4] 미서명 애플리케이션 만들기

(3) DOS 명령창을 열고; adb devices; emulator -avd myAVD

미서명 애플리케이션을 실행하기 위한 타깃 장치가 필요하다. 하이버스 장 치도 좋으나 본 실습에서는 에뮬레이터를 타깃 장치로 사용하고자 한다. DOS 창을 열고'adb devices'로 현재 동작 중인 장치를 확인한 결과 [그림 15-12]처럼 아무 장치도 연결되어 있지 않다. 그래서'emulator -avd myAV D'로 에뮬레이터를 구동시킨다.

Command Shell

 C:WProgram FilesWWindows Resource KitsWTools>adb install D:WandroidWworkspaceWKe ytoolsWandroid.apk

 145 KB/s (0 bytes in 11628.000s)

 pkg: /data/local/tmp/android.apk

 Failure [INSTALL\_PARSE\_FAILED\_NO\_CERTIFICATES]

 C:WProgram FilesWWindows Resource KitsWTools>adb devices

 List of devices attached

 emulator-5554

 device



19

### ◉ [실습 15-4] 미서명 애플리케이션 만들기

(4) 새로운 DOS 명령창을 열고; adb devices; adb install d:₩android₩ workspace₩Keytools₩android.apk

미서명된 애플리케이션 apk 파일을 에뮬레이터에 설치하여 실행시키기 위 하여 새로운 DOS 창을 연다. 'adb devices'명령으로 emulator-5554가 동 작 중인 것을 확인할 수 있다. 이어서 'adb install d:₩android₩workspace ₩Keytools₩android.apk'를 사용하여 서명이 되어 있지 않은 애플리케이션 을 실행해본다. 에뮬레이터에서 디버그키로 서명되었을 때는 오류 없이 실 행되던 애플리케이션이 미서명 상태로 [그림 15-13] 오류를 발생



#### ◉ [실습 15-5] 배포용 서명하기

(1) jarsigner -verbose -keystore helloAndroid.keystore -signedjar helloAndroid-signed.apk android.apk helloAndroid.keystore

keytool을 사용하여 입력하였던 암호를 입력하면 [그림 15-14]처럼 서명 작업이 진행되는 것을 관찰.

그리고 작업이 완료되면 [그림 15-15]처럼 d:₩android₩workspace₩Keyto ols 디렉터리에 helloAndroid-signed.apk 파일이 생성되는 것을 확인.

| D:WandroidW<br>-signed.jar | workspace#Keytools>jarsigner -verbose -keystore helloAndroid.keystor<br>helloAndroid-signed.apk helloAndroid.apk helloAndroid.keystore |
|----------------------------|--|
| Enter Passp                | hrase for keystore:  |
| jarsigner:                 | unable to open jar file: helloAndroid.apk  |
| D∶₩android₩                | workspace#Keytools>jarsigner -verbose -keystore helloAndroid.keystor   |
| -signedjar                 | helloAndroid-signed.apk android.apk helloAndroid.keystore  |
| Enter Passp                | hrase for keystore:  |
| adding:                    | META-INF/MANIFEST.MF   |
| adding:                    | META-INF/HELLOAND.SF   |
| adding:                    | META-INF/HELLOAND.RSA  |
| signing:                   | res/layout/main.xml  |
| signing:                   | AndroidManifest.xml  |
| signing:                   | resources.arsc   |
| signing:                   | res/drawable-hdpi/icon.png   |
| signing:                   | res/drawable-ldpi/icon.png   |
| signing:                   | res/drawable-mdpi/icon.png   |
| signing:                   | classes.dex  |

[그림 15-14] jarsigner로 서명히



[실습 15-5] 배포용 서명하기
 (1) 계속

| <i>≣</i> <sup>µ</sup>  |
|--|
|  |
| 🗸 🕞 이동   |
| 크기 종류 수정한 날짜   |
| 12KB APK 파일 2010-06-27 오후 2:47<br>3KB KEYSTORE 파일 2010-06-27 오후 5:23<br>14KB APK 파일 2010-06-27 오후 5:28 |
|  |

[그림 15-15] 서명 파일 생성 확인



#### ◉ [실습 15-5] 배포용 서명하기

(2) jarsigner -verify -verbose helloAndroid-signed.apk

#### 명령창에서 helloAndroid-signed.apk가 서명되었는지 검사한다. 실습 결과 가 [그림 15-161에 이다.

- 🗆 × D:WandroidWworkspaceWKeytools>jarsigner -verify -verbose helloAndroid-signed.apk 583 Sun Jun 27 17:28:34 KST 2010 META-INF/MANIFEST.MF 704 Sun Jun 27 17:28:34 KST 2010 META-INF/HELLOAND.SF 946 Sun Jun 27 17:28:34 KST 2010 META-INF/HELLOAND.RSA 640 Sun Jun 27 14:47:18 KST 2010 res/layout/main.xml SM 1416 Sun Jun 27 14:47:18 KST 2010 AndroidManifest.xml SM 1296 Fri Jun 18 20:24:08 KST 2010 resources.arsc SM 3966 Fri Jun 18 20:24:08 KST 2010 res/drawable-hdpi/icon.png SM 1537 Fri Jun 18 20:24:08 KST 2010 res/drawable-ldpi/icon.png SM 2200 Fri Jun 18 20:24:08 KST 2010 res/drawable-mdpi/icon.png SM 1904 Sun Jun 27 14:47:18 KST 2010 classes.dex SM signature was verified entry is listed in manifest k = at least one certificate was found in keystore i = at least one certificate was found in identity scope jar verified. D:WandroidWworkspaceWKeytools>

#### ◉ [실습 15-5] 배포용 서명하기

(3) adb install helloAndroid-signed.apk

생성된 apk는 마켓에 배포를 할 수 있고, 다른 사용자들의 의견을 받을 수 있다.

서명된 애플리케이션은 다른 모바일 장치에 설치되어 테스트될 수도 있다. apk 파일의 설치를 위해 adb를 [그림 15-17]처럼 사용한다.



[그림 15-17] helloAndroid-signed.apk 타깃 설치 및 실행

### ◉ [실습 15-5] 배포용 서명하기

(3) adb install helloAndroid-signed.apk (계속)

[그림 15-18]은 에뮬레이터에서 helloAndroid-signed.apk가 설치된 모습을 보여준다. ● 5554myAVD

[그림 15-18] helloAndroid-signed.apk가 설치된 모습



#### Opload and publishing

- 업로드
- 서명을 완료하면 안드로이드 마켓에 업로드할 수 있다.
- 업로드할 때는 애플리케이션의 제목, 개요, 형태, 카테고리, 가격, 배포 지 역 및 복사에 대한 보호를 입력
- 마켓에 올리기 전에 개발 관련 기본 조항을 유의하여야 한다.
- 가격, 지불 조건, 환불, 라이선스 등 개발자가 알아야 한다.
- 퍼블리싱

#### ◎ [실습 15-6] 업로드 및 퍼블리싱

- (1) 실제 업로드를 실습해본다.
- (2) 실제 퍼블리싱을 실습한다.
- (3) 테스트용으로 퍼블리싱한 것을 취소시킨다.



- 아플리케이션 버전 및 패키지 관리
- ⊙ 아이콘
- ◎ 라벨
- 버전
- SDK 버전



28

#### ● [실습 15-7] 버전 코드 1, 버전명 1.0.0이 되도록 Manifest 파일 내용 설정

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://kepricon.tistory.com/script/power</pre>
Editor/pages/"
     package="com.tistory.kepricon"
     android:versionCode="1"
                                                      // 버전 코드
     android:versionName="1.0.0">
                                                      // 버전명
 <uses-sdk android:minSdkVersion="1" />
    <application android:icon="@drawable/icon" android:label="@string/app name">
                                                      // 아이콘 정의
        <activity android:name=".Lotto"</pre>
                 android:label="@string/app name"> // 라벨 정의
           <intent-filter>
                <action android:name="android.intent.action.MAIN" />
   <category android:name="android.intent.category.LAUNCHER" />
           </intent-filter>
        </activity>
```

#### <표 15-3> 안드로이드 라이브러리와 라이선스

| 라이브러리 및<br>주요 프로그램 이름 | 라이선스                      | 크기<br>(byte) | 개요   |
|-----------------------|---------------------------|--------------|--|
| linux kernel          | GPL                       |              |  |
| DBUS                  | GPL2                      |              |  |
| emulator              | GPL                       |              |  |
| adb                   | Apache 2.0                |              |  |
| eclipse               | Eclipse Public<br>license |              |  |
| penBinder(core)       | GPL2                      |              |  |
| libc,so               | BSD                       | 226K         | CLibrary, system calls & other basic facilities such as open, malloc, printf, exit |



| YAFFS2           | GPL                    |      |   |
|------------------|------------------------|------|---|
| WebCore          | LGPL                   |      |   |
| liborypto,so     | BSD                    | 833K | cryptographic algorithms  |
| libctest,so      | BSD                    | 3,5K | SystemCal Test Library(ex: tread, twrite, tork 5)                 |
| libcufils,so     | BSD                    | 53K  | SystemCal Utilities(ex: qsort, select, free 등)                    |
| libdbus,so       | GPL2                   | 301K | Message Bus System Library  |
| lbd\so           | BSD                    | 1,5K | Runtime Dynmaic Loading Library(ex dopen, didose 등)               |
| Webkit           | BSD                    |      | including LGPL  |
| SDL              | LGPL                   |      |   |
| SGL              | Google(Skla)           |      |   |
| OpenGL           | SGI penGL<br>(BSD/MPL) |      |   |
| libd vm,so       | Apache                 | 405K | Dalvik Virtual Machine Library                                    |
| libembunit,so    | GPL2                   | 5,2K | Embedded Unit Testing Framework                                   |
| libevent,so      | BSD(3)                 | 18K  | Event Notification Library  |
| libexpat,so      | MIT                    | 120K | XML Parser Library  |
| libhardware;so   | GPL                    | 28K  | Functions to access into about system<br>hardware(PCI, USB, SCSI) |
| lbopengles_cm,so | OpenGL                 | 87K  | OpenGL for Embedded Systems                                       |
| libsgl,so        | Skla                   | 845K | Skla Graphic library : 2D Graphic Engine                          |
| libsonivax,so    | OHA                    | 94K  | Sonivox's audioNSIDE* Tech Library                                |
| libsq lite, so   | public doamin          | 422K | sqite, GPL2   |
| libss ,so        | GPL2                   | 150K | SSL library   |
| libstdc++,so     | LGPL                   | 3,7K | ISO 14882 Standard C++  |
| libthread_db,so  | LGPL                   | 3,0K | Thread Debuging Library   |

\* OHA : Open Handset Alliance



