



1부

안드로이드 앱 개발 기초

안드로이드 개발 환경은 다른 자바 개발 환경과 같이 공급자가 서로 다른 여러 개발킷들을 조합하여 구축하므로 각 개발킷들의 버전이 서로 일치해야 정상적으로 작동하는 특성이 있습니다. 일반적으로 각 개발 도구를 최신버전으로 다운받아 설치할 것을 권유하지만, 서로 JAVA 버전이 일치하지 않아 오작동하는 문제가 발생할 수 있습니다.



[1장]

안드로이드 개발 환경 구축하기

안드로이드 개발 환경을 구축하려면, 다음과 같이 자바 기반의 개발 프로그램들이 필요합니다.

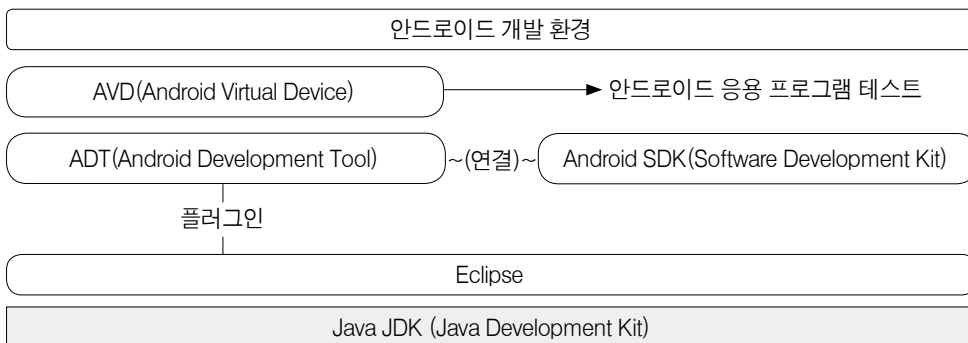
- ① Java JDK (Java Development Kit) : 자바 개발킷
- ② Eclipse : 자바 개발을 손쉽게 할 수 있도록 지원하는 자바 개발 도구
- ③ ADT (Android Development Tool) Eclipse Plugin : 이클립스용 플러그인으로, 구글에서 제공하는 안드로이드 개발툴
- ④ Android SDK (Software Development Kit) : 안드로이드 소프트웨어 개발킷
- ⑤ AVD (Android Virtual Device) : 안드로이드 가상 실행기

주의 개발 프로그램들 간에 호환성 있는 자바 버전 관리

안드로이드 개발 환경은 다른 자바 개발 환경과 같이 공급자가 서로 다른 여러 개발킷들을 조합하여 구축하므로 각 개발킷들의 버전이 서로 일치해야 정상적으로 작동하는 특성이 있습니다. 일반적으로 각 개발 도구를 최신 버전으로 다운받아 설치할 것을 권유하지만, Java 버전이 일치하지 않아 오작동하는 문제가 발생할 수 있습니다.

안드로이드 개발 도구들 간의 관계는 다음과 같이 도식화할 수 있습니다.

안드로이드는 자바 기반으로 개발된 솔루션이므로, 전반적인 개발 기반을 "Java JDK"로 구현하고, 자바 개발 도구로 "Eclipse"를 사용하며, "Eclipse"는 안드로이드의 솔루션 집합체라 할 수 있는 "Android SDK"와 연동하기 위해 이클립스용 플러그인인 "ADT"를 사용합니다. 끝으로 "AVD"를 통해 개발한 안드로이드 응용 프로그램을 테스트할 수 있습니다.



1 Java JDK 설치

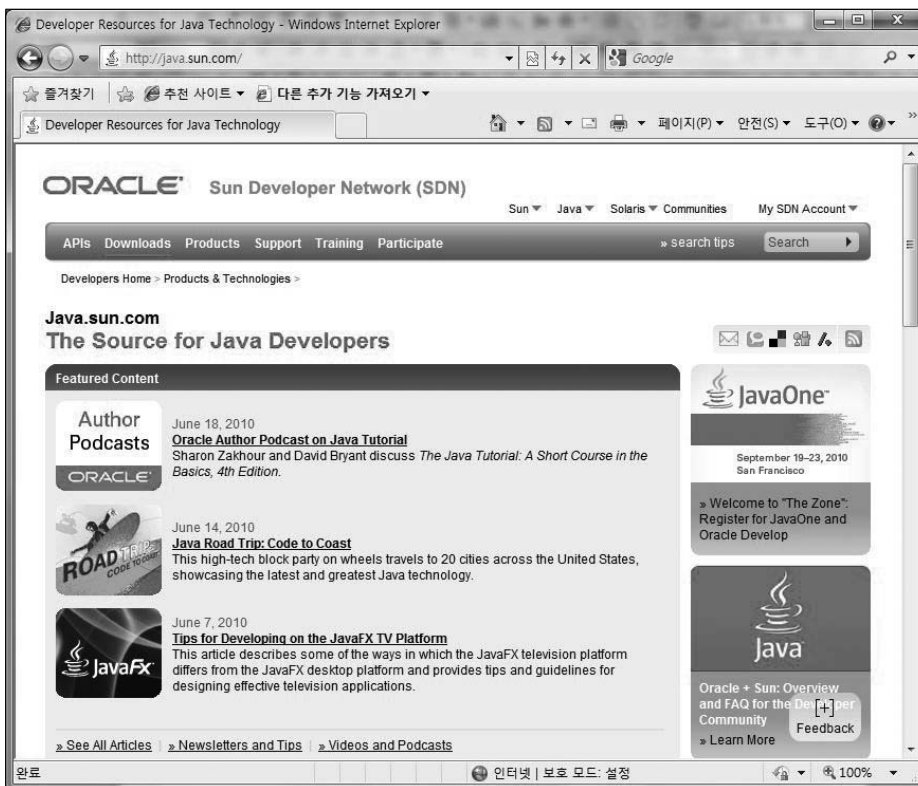
Java JDK는 다음과 같이 자바 공식 사이트에서 다운받아 설치할 수 있습니다. 안드로이드 개발 환경에 알맞는 JDK는 Android SDK 버전에 따라 차이가 있을 수 있으나, 5버전 또는 6버전 이상이어야 합니다. 본 장에서는 6버전을 설치하는 스텝을 보여주고 있습니다.

- 설치본 : jdk-6u20-windows-i586.exe
- 배포 사이트 : java.sun.com

1.1 Java JDK 다운로드

스텝 1

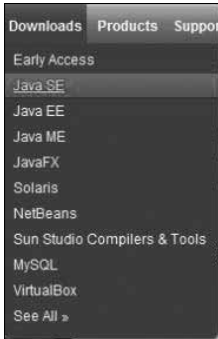
웹브라우저로 <http://java.sun.com/> 사이트에 접속합니다.



스텝 2

상단 메뉴에서 [Downloads - Java SE]를 클릭합니다.

Java JDK에는 여러가지가 있습니다. 안드로이드 개발 환경은 이 중에서 Java SE (Standard Edition) 버전이 적합합니다.



스텝 3

"Java JDK 6 Update 20 JDK"를 다운로드합니다.

JDK 6 버전 이상 다운받아 설치할 것을 권장합니다. JDK는 JRE를 포함하고 있으므로 JDK를 설치하면 JRE를 별도로 설치할 필요는 없습니다.



스텝 4

개발 컴퓨터 OS에 맞는 플랫폼을 선택합니다. 플랫폼 목록 중 Windows 버전을 선택한 후, 라이선스 동의 항목을 체크하고 다운로드 버튼을 클릭합니다.

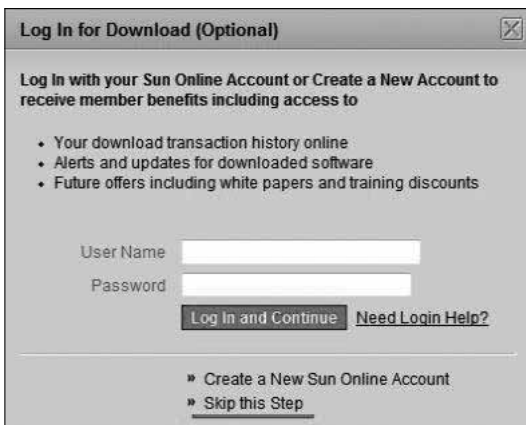
주의 개발 컴퓨터의 OS가 Windows 일지라도 32비트인지 64비트인지 확인하여 다운로드받아 설치해야 합니다.



스텝 5

다운로드하기 위한 사이트 인증 화면이 나타납니다.

다운로드 버튼을 클릭하면 로그인 팝업창이 나타나는데, 굳이 회원 가입할 필요가 없다면, 그림과 같이 "Skip this Step" 버튼을 클릭하면 로그인하지 않고 다운로드받을 수 있습니다.

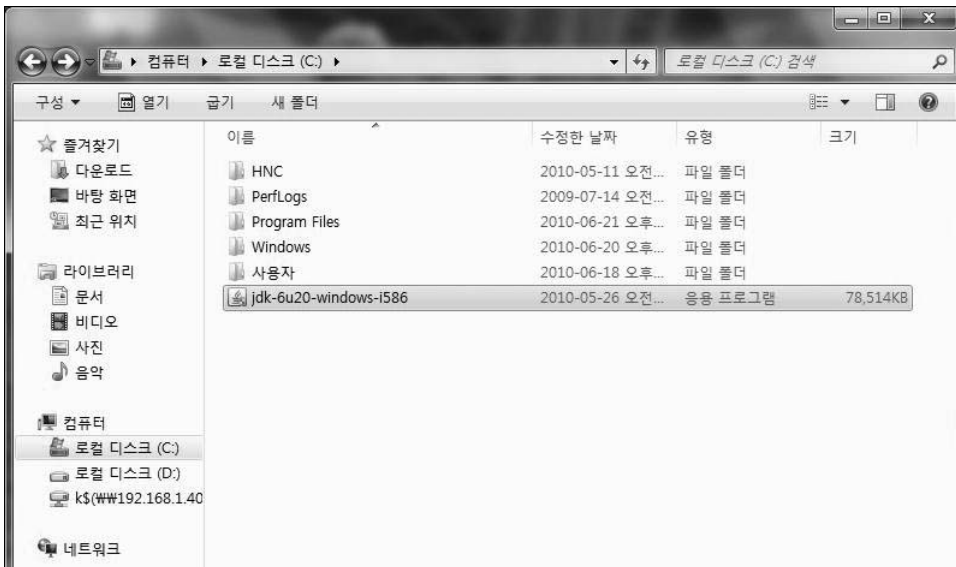


1.2 Java JDK 설치

스텝 1

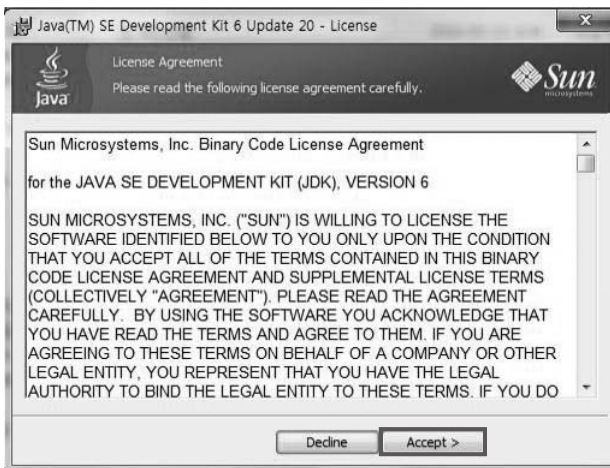
공식 사이트에서 다운로드한 파일을 설치해 보겠습니다.

그림과 같이 다운로드 받은 파일을 더블클릭하여 실행합니다.



스텝 2

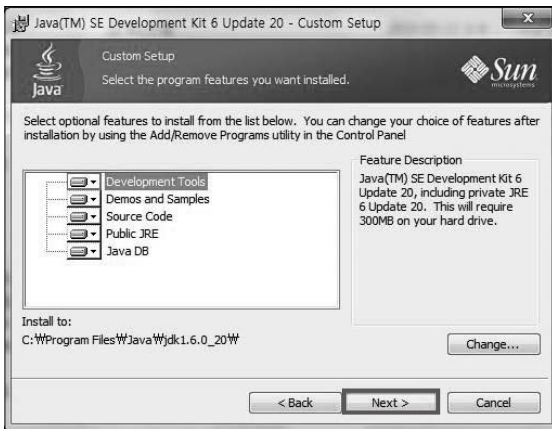
라이선스 창이 나타나면, "Accept" 버튼을 클릭합니다.



스텝 3

설치할 패키지와 JDK 설치 위치를 지정한 후 "Next" 버튼을 클릭합니다.

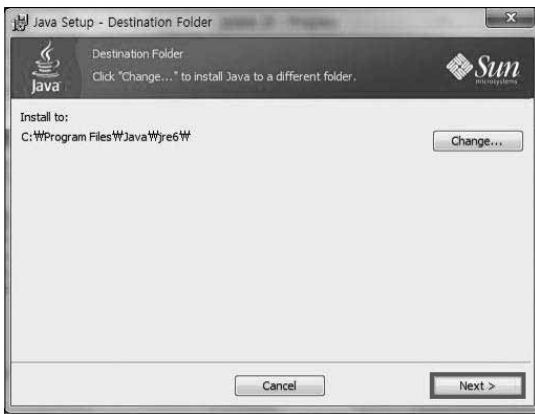
가급적이면 기본 설정으로 설치하고, 필요에 따라 "Change..." 버튼을 이용하여 개인적인 개발 환경에 맞춰 설치 위치를 변경할 수 있습니다. 개인적인 개발 환경에 따라 다를 수 있으나 그림과 같이 모든 패키지를 선택하여 설치할 것을 권장합니다.



스텝 4

JRE 설치 위치를 정의한 후, "Next" 버튼을 클릭합니다.

이전 화면에서 "Public JRE" 패키지를 선택했다면, 다음과 같이 JRE 설치 경로를 지정하는 화면이 나타납니다.



스텝 5

그림과 같이 설치 스텝이 완료되면, "Finish" 버튼을 클릭하여 Java JDK 설치를 완료합니다.



1.3 Java JDK 환경 설정

Java에 대한 시스템 환경 변수를 설정합니다.

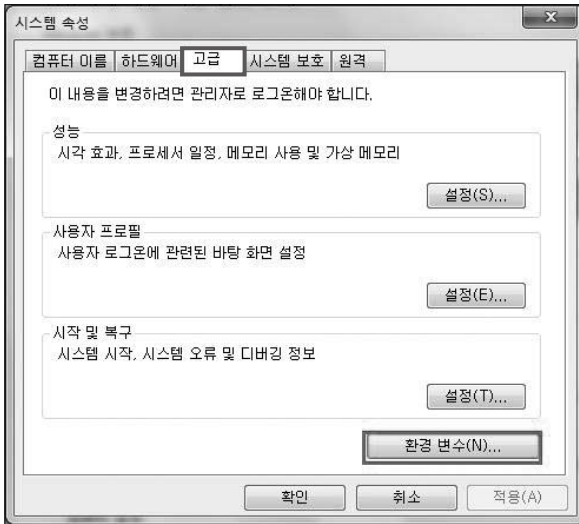
자바의 실행 명령은 용도에 따라 여러가지가 있는데, 그 중 핵심적인 명령은 자바 응용 프로그램을 실행하는 "java.exe"와 자바 소스를 컴파일하는 "javac.exe"입니다. 이 자바 명령어를 시스템에서 인식하도록 시스템 환경 변수에 설치한 자바의 경로를 지정합니다.

주의 Java를 새로 설치했다고 해서 시스템에서 새로 설치한 Java 버전을 바로 인식하는 것은 아닙니다. 만약, 이전에 다른 버전의 Java가 설치되어 있었다면 시스템은 이전에 설치한 Java 버전을 인식하고 있어 안드로이드가 제대로 작동하지 않을 수 있습니다.

스텝 1

다음과 같이 "시스템 속성" 화면의 "고급" 탭에서 "환경 변수" 버튼을 클릭합니다.

- Windows 7 : [제어판>시스템>고급 시스템 설정>고급]
- Windows XP : [제어판>시스템>고급]



스텝 2

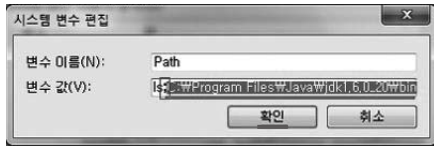
"환경 변수" 화면이 나타나면, "Path" 항목을 선택하고, "편집" 버튼을 클릭합니다.



스텝 3

"Path" 환경 변수에 자바 실행 경로를 지정합니다.

"Path"에 대한 "시스템 변수 편집" 창이 나타나면, 그림과 같이 변수값에 "세미콜론(:)"을 입력하고, 그 뒤에 "사용할 자바 실행 경로"를 입력한 후, "확인" 버튼을 클릭합니다.



주의 사용할 자바 실행 경로를 "Path" 환경 변수에 등록을 할 때, 각 실행 경로를 구분하는 "세미콜론(:)"을 누락하지 않도록 주의해야 합니다.

참고 만일 개발 컴퓨터에 자바가 여러 버전이 설치되어 있고, "Path" 환경에도 중복하여 여러 버전의 자바 실행 경로가 지정되어 있다면, 맨 앞에 있는 자바 실행 경로가 우선적으로 인식됩니다.

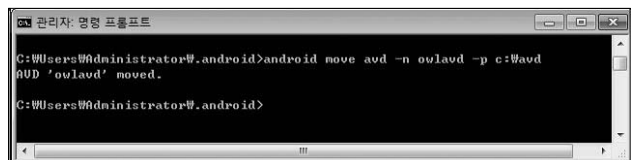
1.4 Java JDK 버전 확인

설치한 자바 버전을 시스템에서 잘 인식하고 있는지를 확인합니다.

스텝 1

다음과 같이 명령 프롬프트를 실행합니다.

[윈도우 시작 메뉴 > 모든 프로그램 > 보조 프로그램 > 명령 프롬프트]

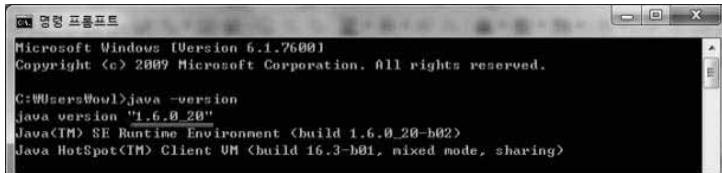


스텝 2

다음과 같이 "자바 실행 명령어"와 "자바 컴파일 명령어"를 실행하여 시스템에서 인식하고 있는 자바 버전을 확인합니다.

그림과 같이 원하는 자바 버전이 출력된다면 시스템이 자바 버전을 올바르게 인식하고 있다는 것을 의미합니다.

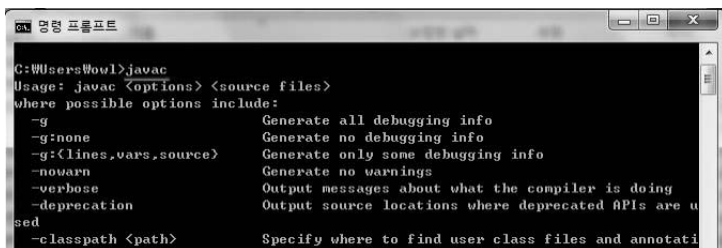
- 자바 실행 명령어 : `java -version`



```
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Wool>java -version
java version "1.6.0_20"
Java(TM) SE Runtime Environment (build 1.6.0_20-b02)
Java HotSpot(TM) Client VM (build 16.3-b01, mixed mode, sharing)
```

- 자바 컴파일 명령어 : `javac`



```
C:\Users\Wool>javac
Usage: javac <options> <source files>
where possible options include:
  -g               Generate all debugging info
  -g:none          Generate no debugging info
  -g:{lines,vars,source}  Generate only some debugging info
  -nowarn          Generate no warnings
  -verbose         Output messages about what the compiler is doing
  -deprecation     Output source locations where deprecated APIs are u
sed
  -classpath <path> Specify where to find user class files and annotati
```

2 이클립스 설치

이클립스는 자바 개발 도구입니다. 이클립스가 없어도 자바 개발을 원시적으로 할 수는 있지만, 자바 개발이 갈수록 많은 자바 응용 프로그램과 연동하여 개발해야 하기 때문에 이와 같은 자바 개발 도구가 없다면 자바 개발에 많은 노력이 필요합니다. 자바 소스를 체계적으로 관리하고 컴파일하는 과정에서 번거로움을 덜어 주지만, 개발 컴퓨터의 사양을 많이 요구하는 단점도 있습니다. 여기서는 이클립스를 기반으로 개발하는 스텝을 제시하겠습니다.

- 설치본 : eclipse-jee-galileo-SR2-win32.zip
- 배포 사이트 : www.eclipse.org

2.1 이클립스 설치

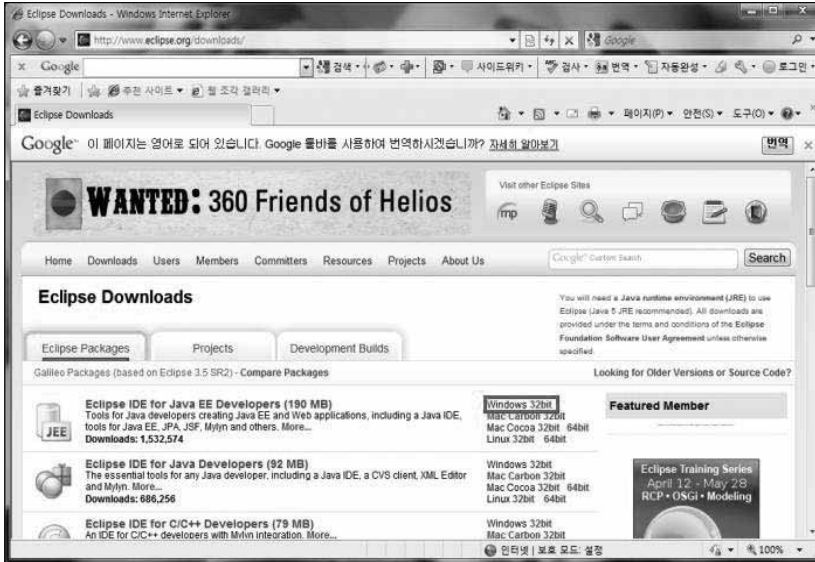
스텝 1

이클립스 사이트에서 그림과 같이 "Downloads" 메뉴를 클릭하여 다운로드 페이지로 이동합니다.



스텝 2

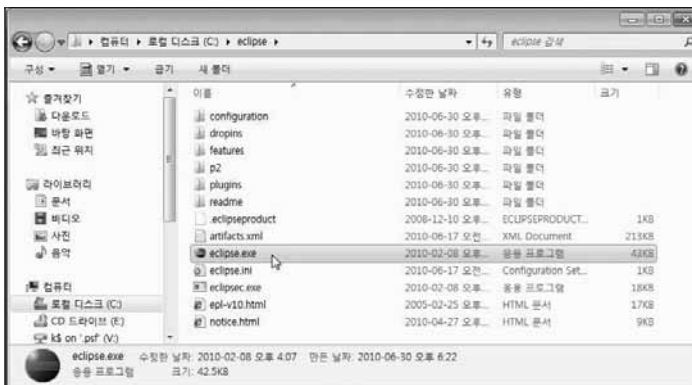
"Eclipse Packages"에서 "Eclipse IDE..." 중 적당한 플랫폼의 설치본을 다운로드합니다. 여기서는 "Eclipse IDE..." 중 "Windows 32bit"를 클릭하여 다운로드했습니다.



스텝 3

적당한 위치에 다운받은 파일의 압축을 풉니다.

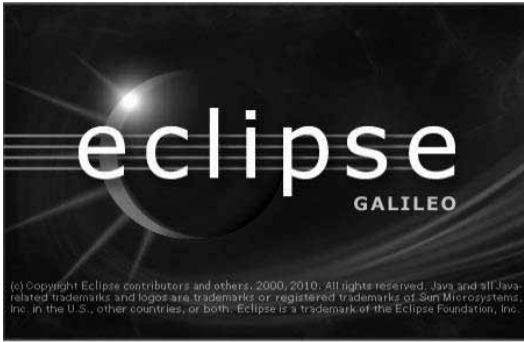
다운받은 이클립스는 별도 설치 스텝 없이 압축만 풀어서 사용합니다. 이클립스를 실행하기 전, 적당한 위치에 압축을 풀고, 압축을 푼 이클립스 폴더를 원하는 위치로 이동하여도 무관합니다. 가급적이면, 이클립스가 경로를 인식하는데 문제가 발생하지 않는 "C:\Program Files" 또는 "C:\"에 위치할 것을 권장합니다.



2.2 이클립스 실행 및 작업 경로 정의

스텝 1

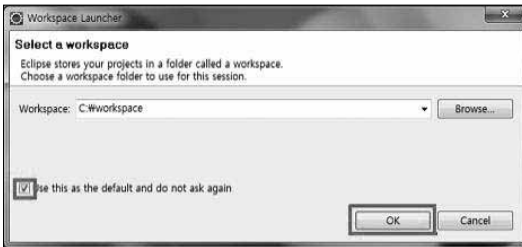
"eclipse.exe"를 실행하면, 이클립스가 시작됩니다.



스텝 2

이클립스 작업 경로를 설정합니다.

설치한 이클립스를 처음 실행하면, "Workspace Launcher" 창이 나타납니다. 이 창은 프로젝트를 저장하는 경로를 지정하는 창입니다. 여기서는 프로젝트 저장하는 곳을 "C:\workspace"로 지정하였습니다.



그림과 같이 "Use this as the default and do not ask again"을 체크하면, 다음에 이클립스를 실행할 때, 워크스페이스 경로를 다시 묻지 않고, 현재 지정한 경로를 기본 워크스페이스로 사용합니다.

참고 워크스페이스(workspace) 경로는 필요에 따라 다음 명령을 사용하여 나중에 변경할 수 있습니다.

- 워크스페이스 경로 변경 명령 : File>Switch Workspace>Other...



스텝 3

"Welcome" 화면이 나타나면, "Workbench" 버튼을 클릭하여 이클립스 작업 화면으로 이동합니다.
"Welcome" 화면도 이클립스를 처음 실행했을 때 나타납니다. 이후에 이클립스를 실행하면,
"Welcome" 화면을 거치지 않고 이클립스 작업 화면이 나타납니다.

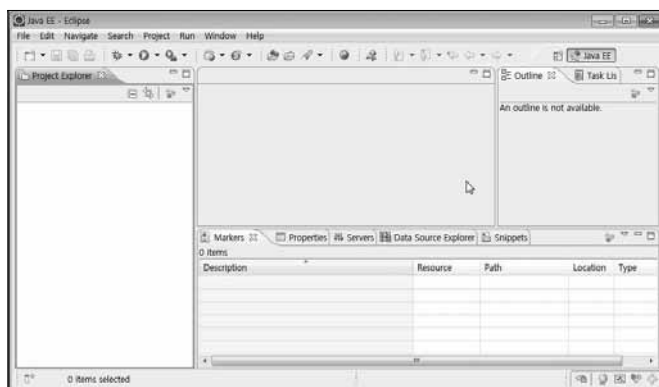
참고 나중에 "Welcome" 화면을 다시 보고 싶으면, 다음과 같이 이클립스 메뉴를 실행하면 됩니다.

- 이클립스 "Welcome" 화면 메뉴 : Help > Welcome

▼ 이클립스 "Welcome" 화면



▼ 이클립스 작업 화면

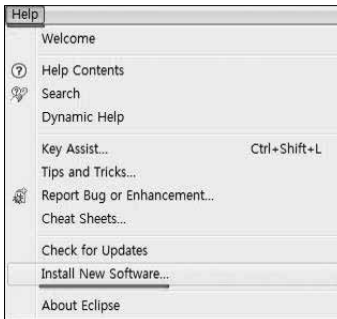


3 ADT 플러그인 설치

이클립스에서 안드로이드를 개발하려면, 이클립스 플러그인으로 ADT(Android Development Tool)를 설치해야 합니다.

스텝 1

이클립스 상단 메뉴에서 [Help > Install New Software]를 선택합니다.



스텝 2

"install" 화면에서 "Add" 버튼을 클릭합니다.



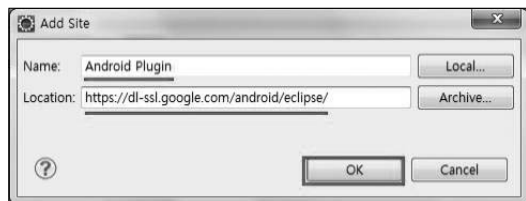
스텝 3

"Add Site" 창에서 ADT를 배포하는 사이트 주소를 정의합니다.

"Name"에 "Android Plugin"이라고 이름을 입력하고(Name은 임의로 주셔도 상관없습니다.),

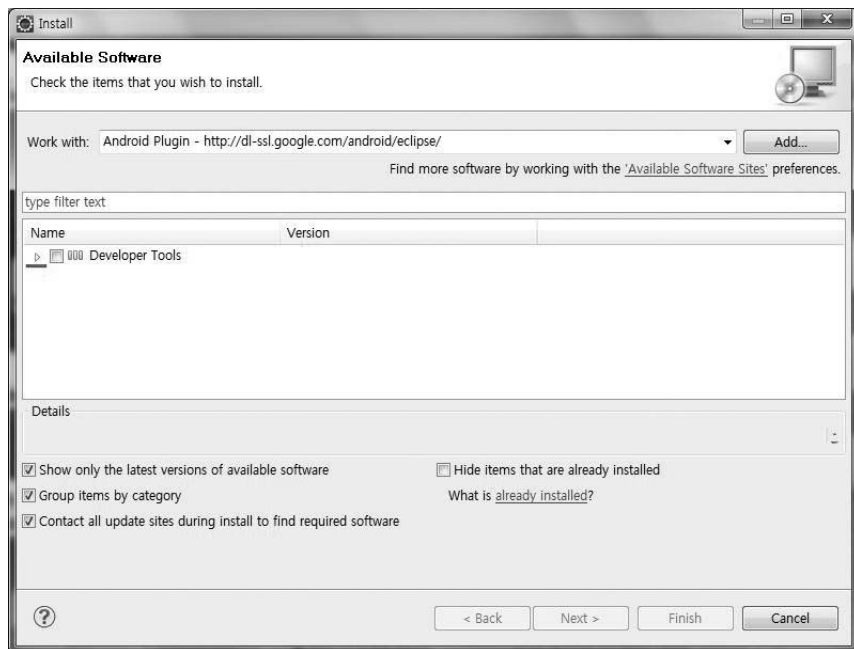
"Location"에는 ADT 사이트 주소를 "https://dl-ssl.google.com/android/eclipse/"로 입력한 후, "OK" 버튼을 클릭합니다.

참고 네트워크 보안 문제나 윈도우 방화벽으로 인해 올바르게 작동하지 않는 경우가 있는데 이 경우, ADT 사이트 주소에 기재한 "https"를 "http"로 변경해 시도해 보기 바랍니다.



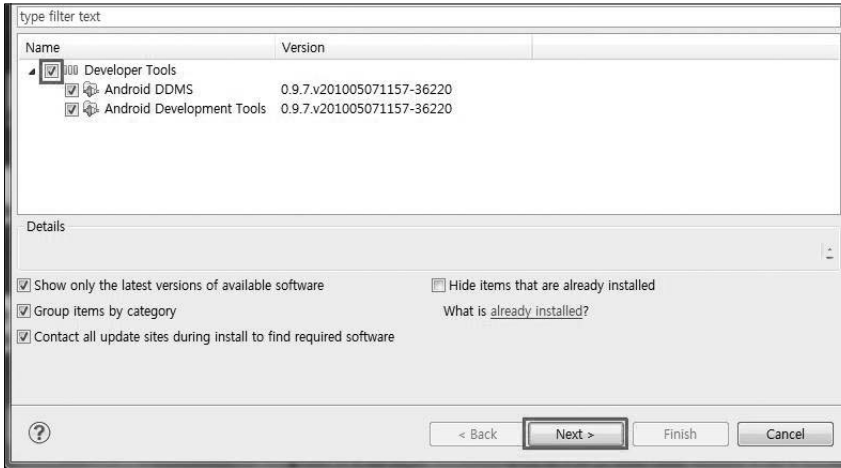
스텝 4

인터넷을 통해 ADT 사이트에 접속이 성공하면, 그림과 같이 설치할 패키지 목록에 "Developer Tools"라고 나타납니다.



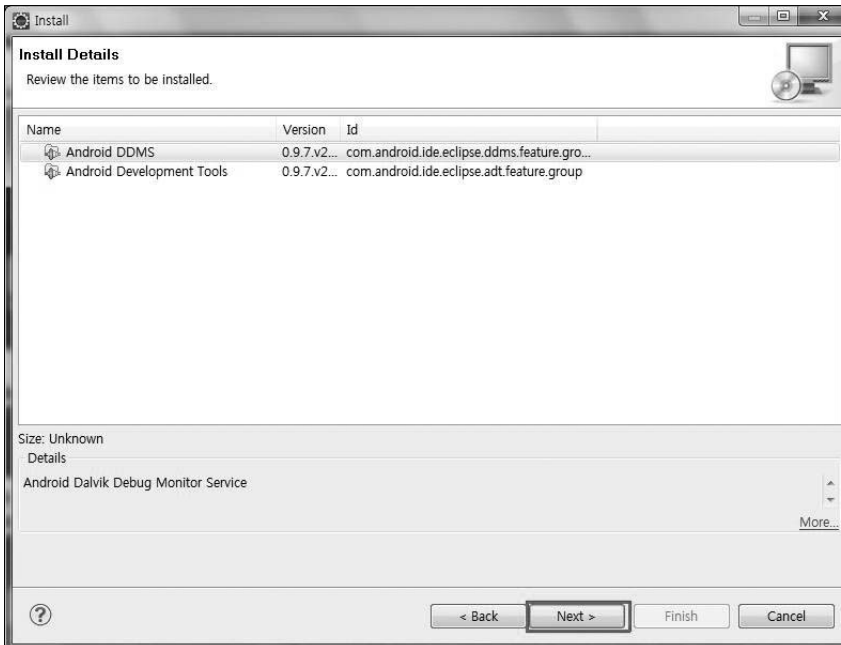
스텝 5

"Developer Tools"를 체크하고 "Next" 버튼을 눌러 다음 단계로 넘어갑니다. "Developer Tools" 항목을 확장하면, "Android DDMS", "Android Developer Tools"가 목록에 나타납니다.



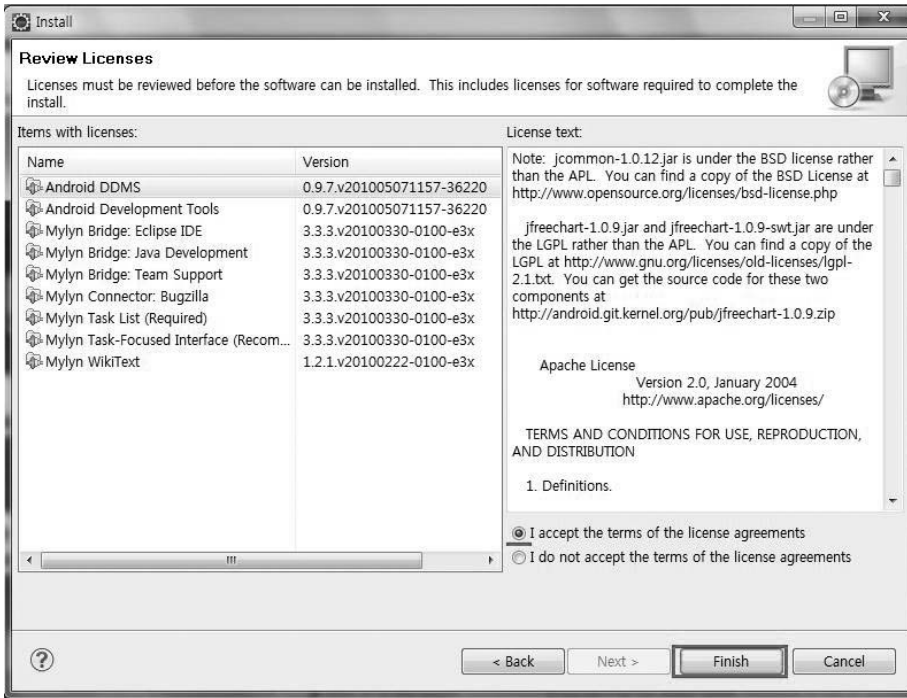
스텝 6

설치할 목록을 확인하고, "Next" 버튼을 클릭합니다.



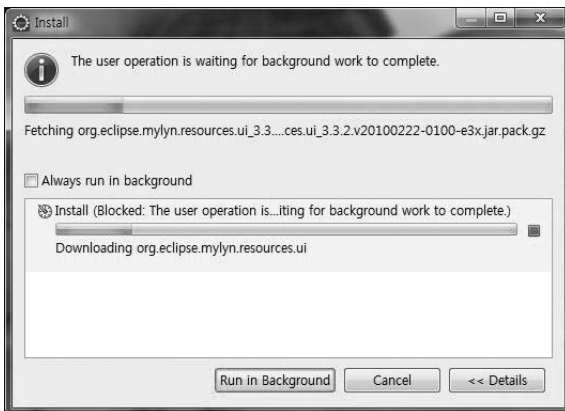
스텝 7

라이선스 창이 나타나면, 동의하는 체크박스를 선택하고 "Finish" 버튼을 클릭합니다.



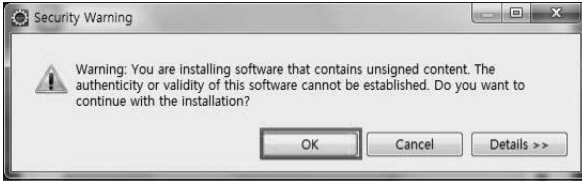
스텝 8

설치 스텝이 진행되면, 그림과 같이 나타납니다.



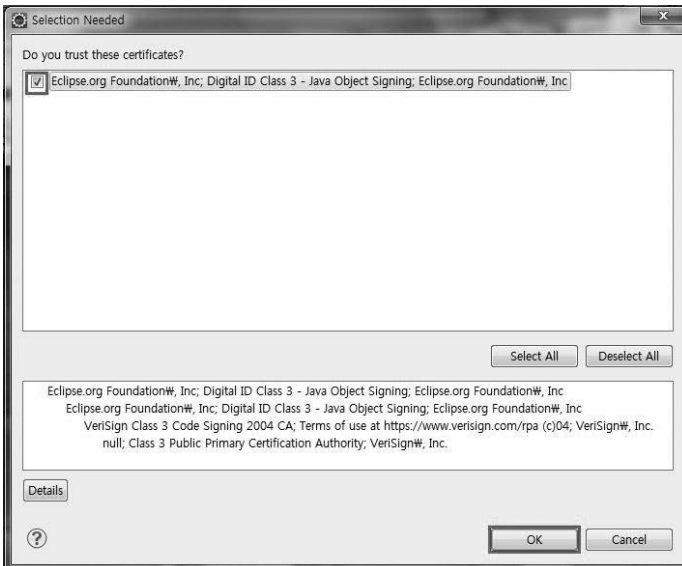
스텝 9

설치 중에 서명 누락된 소프트웨어로 인해 중단되어 경고 메시지 창이 뜨는 경우가 있습니다. 이 때, "Ok" 버튼을 클릭하여 설치를 계속 진행합니다.



스텝 10

"Selection Needed" 화면이 나타나면, 그림과 같이 체크하고 "OK" 버튼을 클릭합니다.

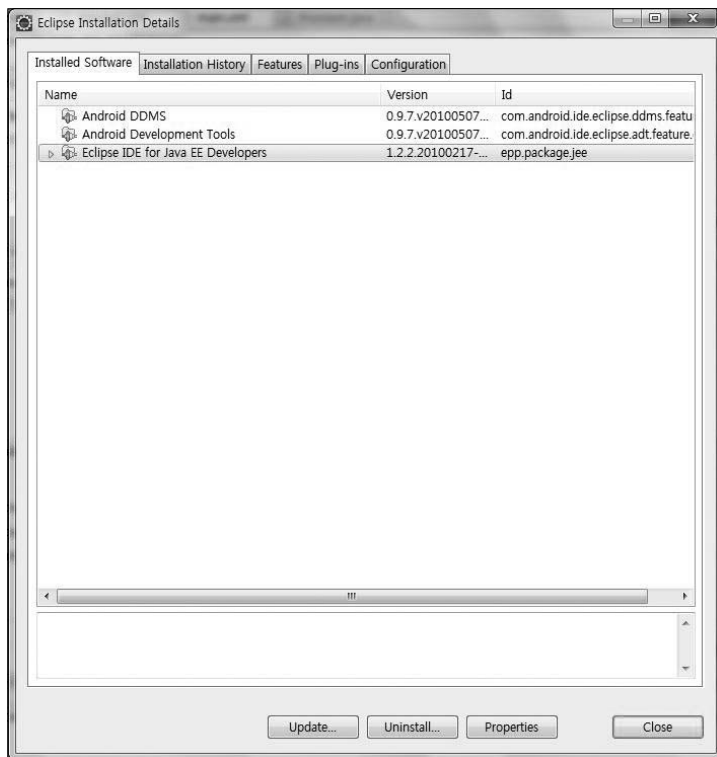
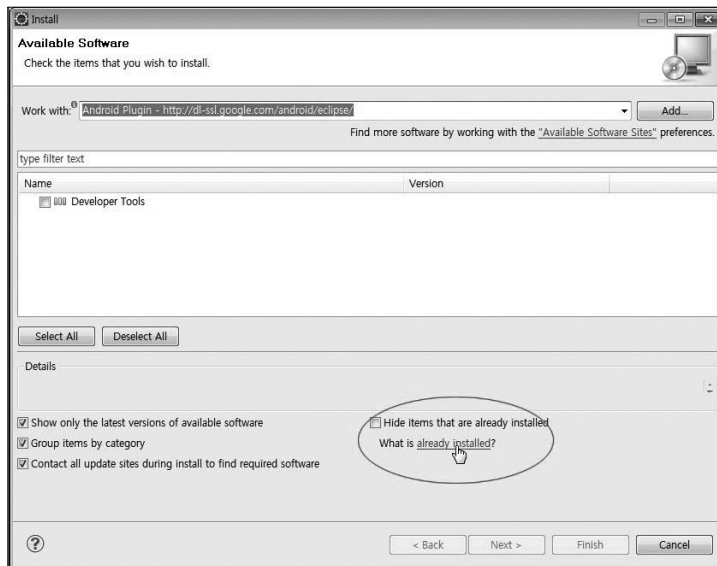


스텝 11

설치가 완료 후, 재시작을 묻는 메시지가 나타납니다. "Yes"를 클릭하여 이클립스를 재실행시킵니다.



참고 ADT가 설치되어 있는지를 확인하려면, [Help > Install New Software] 메뉴를 실행하고, "already Installed" 링크 버튼을 클릭하면, 설치된 ADT 패키지를 확인할 수 있으며, 필요에 따라 "Update", "Uninstall" 등의 패키지 관리를 할 수 있습니다.



4 Android SDK 설치

Android SDK (Software Development Kit)는 Android가 구동하는데 필요한 각종 기능들을 모두 담고 있는 "Android Software의 집합체"라고 할 수 있습니다. 좀 더 쉽게 이해하자면, "안드로이드의 핵심 엔진은 Android SDK이다"라고 표현할 수 있습니다.

참고 많은 자바 배포본이 그러하듯이 "Android SDK"도 본서를 집필 당시에는 압축본(zip)으로 배포되고 있습니다. 나중에 설치본으로 바뀌 배포할 수도 있으나 자바 개발자의 환경에 따라 SDK의 연동 방식이 다양하므로 설치정보보다는 압축형으로 배포하는 것이 범용적입니다. 안드로이드 개발자 사이트에서 SDK를 배포할 때, 배포 버전에 따라 설치 방법과 주의 사항을 잘 설명하고 있으니 이를 참조하면, 최신 버전이 다른 형태로 배포된다고 할지라도 SDK를 설치하고 연동하는데 큰 어려움이 없을 것입니다.

- 설치본 : android-sdk_r06-windows.zip
- 배포 사이트 : developer.android.com

4.1 Android SDK 다운로드 및 압축 해제

스텝 1

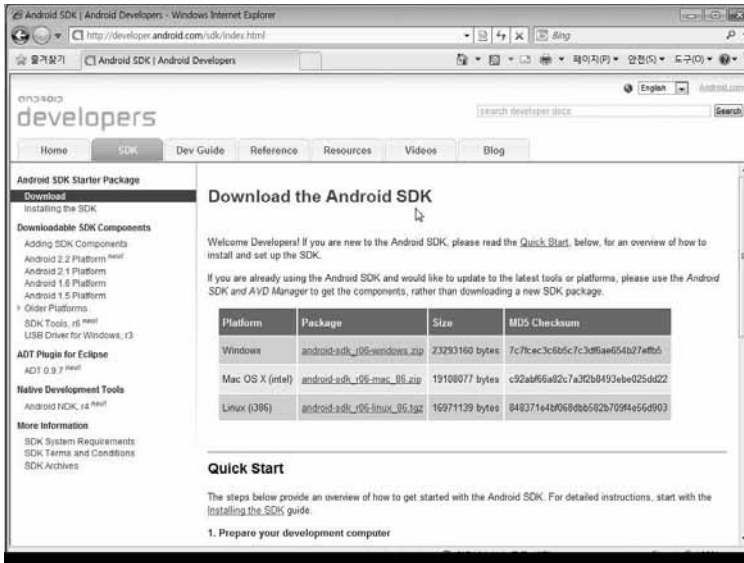
안드로이드 개발자 사이트에 접속하고, 상단 메뉴의 "SDK" 메뉴를 선택합니다.



스텝 2

개발자 환경에 맞는 "SDK"를 선택하여 다운로드합니다.

그림과 같이 SDK 페이지에서 여러 플랫폼에 따른 SDK 배포본을 확인할 수 있을 것입니다. 본서에서는 Windows용 패키지를 다운로드합니다.



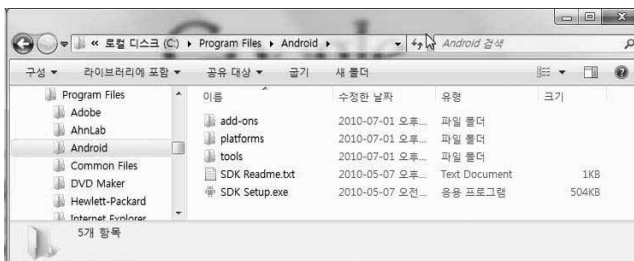
Platform	Package	Size	MD5 Checksum
Windows	android-sdk_r06-windows.zip	23293160 bytes	7c7fcec3c6b5c7c3d8a6654b27efb5
Mac OS X (intel)	android-sdk_r06-mac_osx.zip	19108077 bytes	c92abf66a82c7a3f2b8493ebe025d422
Linux (i386)	android-sdk_r06-linux_386.tar.gz	16971139 bytes	848371e4b0609db6b52b709f4e56d903

스텝 3

다운로드한 SDK를 압축 해제하고, 적당한 위치에 SDK 폴더를 이동합니다.

본서에서는 다음과 같이 "Android SDK" 위치를 정의했습니다.

- Android SDK 경로 : C:\Program Files\Android



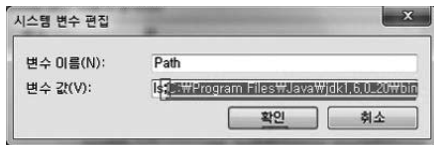
4.2 Android SDK 환경 설정

스텝 1

"시스템 환경 변수"에 "Android SDK"의 "tools" 경로를 지정합니다.

앞서 "Java SDK"에 대한 환경 설정에서 설명한 바와 같이, 시스템이 안드로이드 SDK 도구들을 인식할 수 있도록 "시스템 환경 변수" 중 "Path"에 다음과 같이 "안드로이드툴" 경로를 설정합니다.

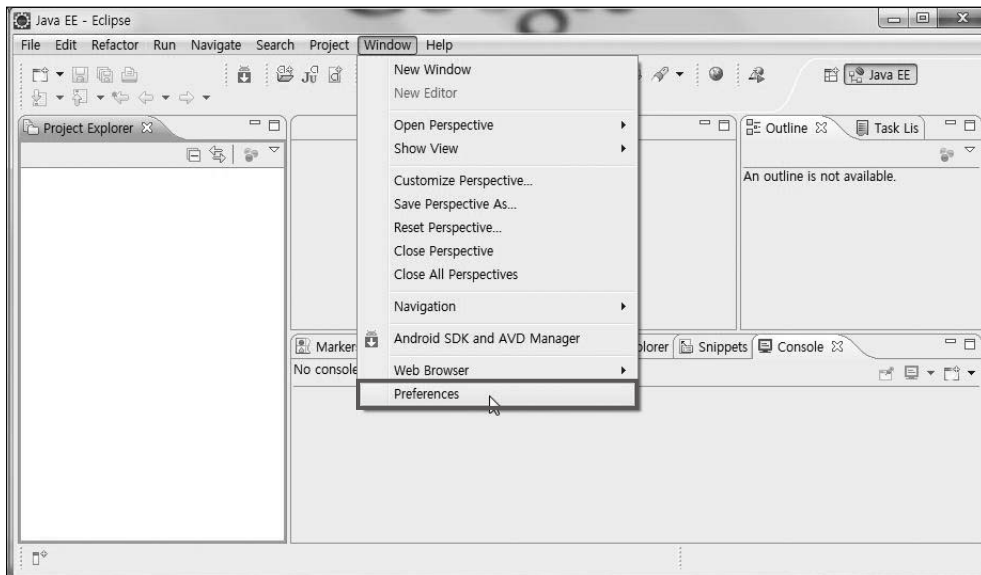
- Windows 7 : [제어판]>시스템>고급 시스템 설정>고급>환경 변수]
- Windows XP : [제어판]>시스템>고급>환경 변수]
- 추가할 Path : C:\Program Files\Android\tools



스텝 2

이클립스를 실행하고, 이클립스 상단 메뉴에서 [Window > Preferences]를 선택합니다.

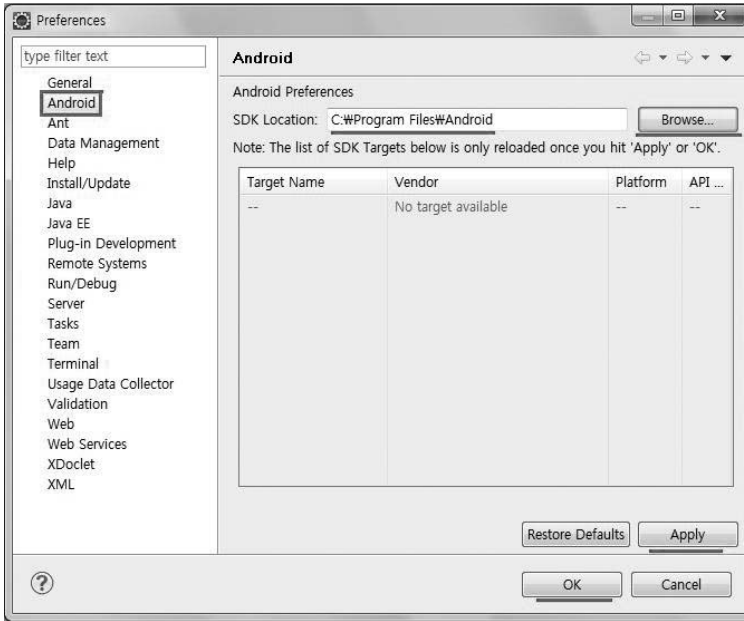
이클립스에서 Android SDK를 인식할 수 있도록 환경 설정을 합니다.



스텝 3

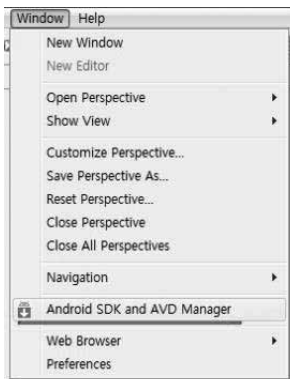
"Preferences" 창이 나타나면, 그림과 같이 "Android" 메뉴를 선택하고, "SDK Location" 입력란에 다음과 같이 "Browse.." 버튼을 이용하여 "Android SDK" 경로를 선택한 후, "Apply" 버튼을 클릭하고, "Ok" 버튼을 클릭합니다.

- "Android SDK" 경로 : C:\Program Files\Android



스텝 4

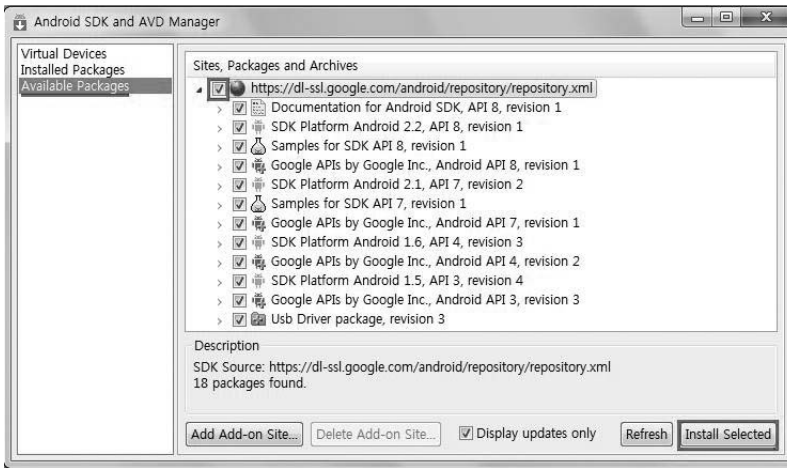
이클립스 상단 메뉴에서 [Window > Android SDK and AVD Manager]를 실행합니다.



스텝 5

"Android SDK and AVD Manager" 창에서 "Available Packages"를 선택하면, 설치가 가능한 목록이 나타납니다. 최상위에 있는 "안드로이드 패키지 배포 서버 주소"를 체크하고, "Install Selected" 버튼을 클릭하면, 안드로이드 추가 패키지들에 대한 설치가 진행됩니다.

참고 "Android SDK and AVD Manager"는 "구글 안드로이드 서버"를 통해 안드로이드에 관련된 패키지들을 설치하거나 업데이트하는 등의 관리를 할 수 있게 해주는 "안드로이드 패키지 관리자"라고 할 수 있습니다.



스텝 6

"Choose Packages to Install" 화면이 나타나면, 패키지 라이선스에 대한 동의 처리로 "Accept All"을 선택하고 "Install" 버튼을 클릭합니다.



5 AVD(Android Virtual Device) 생성

안드로이드 응용 프로그램을 만들어 실행해 보려면, 안드로이드 단말기가 필요로 합니다. 개발자를 위해 제공하는 안드로이드 가상기기(Virtual Device)가 바로 "AVD"입니다. "AVD"를 구동하기 위한 소프트웨어는 이미 앞의 설치 스텝에 포함되어 있으므로 이클립스에서 가상기기를 생성하는 작업이 필요합니다.

5.1 AVD 생성하기

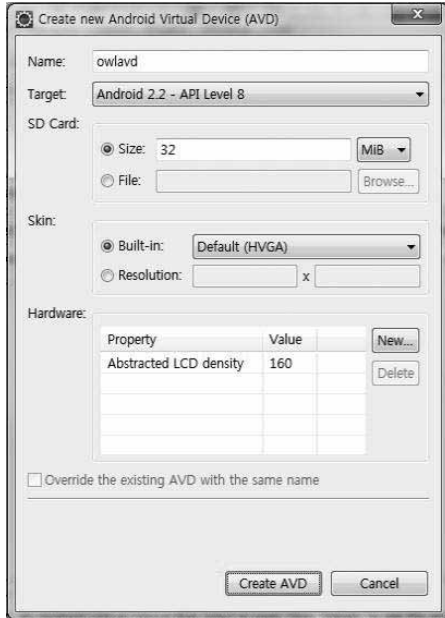
스텝 1

이클립스 상단 메뉴에서 [Window > Android SDK and AVD Manager]를 선택하고, "Android SDK and AVD Manager" 창에서 "Virtual Device"를 선택합니다. 처음에는 그림과 같이 "No AVD available"이 목록에 나타납니다. "New" 버튼을 클릭하여 AVD 생성을 시작합니다.



스텝 2

"Create new Android Virtual Device (AVD)" 창에서 가상기기의 이름(Name)과 유형(Target), 메모리(SD Card), 스킨(Skin), 해상도(Resolution), 기기속성(Hardware) 등을 정의할 수 있습니다.



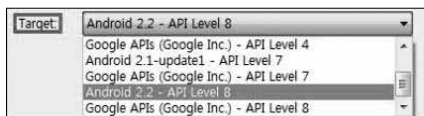
스텝 3

AVD "Name"은 가상기기의 이름입니다. 여기서는 "owlavd"로 작성합니다.
("Name"란은 임의로 정해도 무관합니다.)



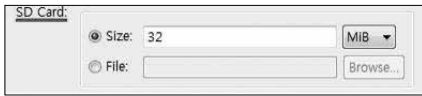
스텝 4

"Target"은 개발 대상 단말기에서 지원하는 안드로이드의 버전을 선택하는 곳입니다.
여기서는 "Android SDK"에서 설치한 버전 중 최신 버전인 "Android 2.2" 버전을 선택합니다.



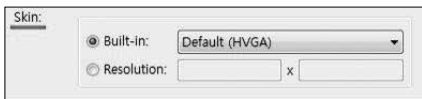
스텝 5

"SD Card"는 가상기기가 사용할 메모리 크기를 설정해 주는 곳입니다.
여기서는 32MB로 설정합니다.



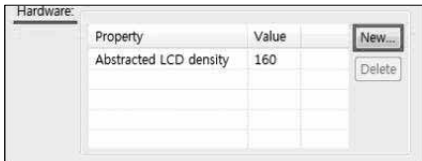
스텝 6

"Skin"은 가상기기 화면의 해상도를 선택하는 곳입니다.
여기서는 안드로이드의 디폴트 값인 HVGA를 그대로 사용합니다.



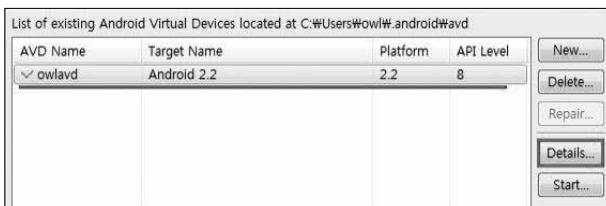
스텝 7

"Hardware"는 가상기기에 대한 장치 옵션을 선택하는 곳입니다.
여기서는 기본적인 하드웨어 장치 옵션 설정을 그대로 사용합니다.



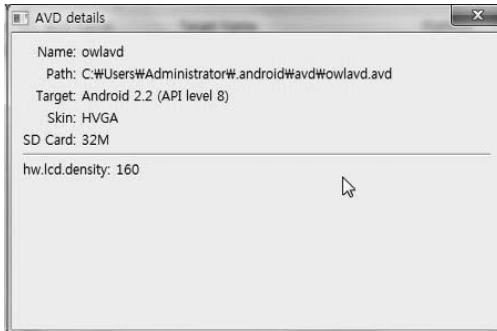
스텝 8

생성할 "AVD" 속성을 점검하였으면 "Create AVD" 버튼을 클릭하여 "AVD"를 생성합니다.
생성이 성공적으로 끝났으면, 그림과 같이 "Android SDK and AVD Manager" 목록에 생성한 "AVD"가 나타날 것입니다.



스텝 9

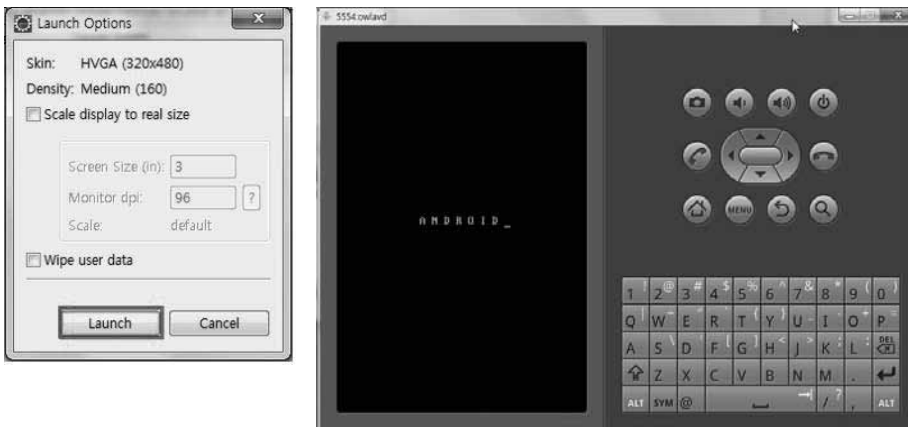
"Android SDK and AVD Manager" 창에서 생성한 "AVD"를 선택하고, "Details" 버튼을 클릭하면, 생성된 "AVD"의 상세 정보를 확인할 수 있습니다.



5.2 AVD 실행하기

스텝 1

"Android SDK and AVD Manager" 창에서 원하는 AVD를 선택하고, "Start" 버튼을 클릭하면, AVD를 실행하기 위한 "Launch Option" 창이 나타납니다. 이제 그냥 "Launch"를 클릭하면 안드로이드 가상기가 실행됩니다.



참고 Launch Option

- Scale display to real size : 실행할 가상기기의 화면 크기 조절
- Wipe user data : 가상기기를 실행할 때 기존에 사용했던 사용자 데이터를 삭제하는 기능

스텝 2

가상기기가 부팅하는데 약간의 시간이 소요됩니다. 부팅이 완료되면, 그림과 같이 안드로이드 잠금 화면이 나타납니다.



스텝 3

안드로이드 가상기기 우측의 키패드에서 "menu" 버튼을 클릭하거나 열쇠 모양의 "Unlock" 버튼을 우측으로 드래그하면, 그림과 같이 "홈화면"으로 이동합니다.



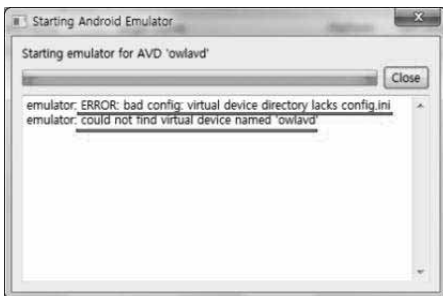


참고 안드로이드 가상기기에서는 대부분의 안드로이드 응용 프로그램이 작동되지만 실제 안드로이드 폰처럼 통화하거나 문자 메시지를 보내는 등의 기능은 당연히 제한되어 있습니다.

5.3 AVD 오작동 사례

위의 스텝대로 실행했는데도 불구하고 안드로이드 가상기기가 제대로 실행이 되지 않고 오류가 발생하는 경우가 있을 수 있습니다. 여러가지 원인이 있을 수 있지만 가장 많이 오류를 범하여 원인을 찾기 난해한 경우가 다음과 같은 사례입니다.

그림과 같이 가상기기를 실행하여 오류가 발생하는 경우는 가상기기가 구동하기 위해 필요한 파일들의 경로를 제대로 인식하지 못해서 발생하는 문제입니다.



이는 경로명에 있는 공백 또는 한글, 특수문자 등을 자바에서 해독하지 못해 발생합니다. Windows 사용자 계정을 한글로 사용하는 경우에도 발생할 수 있습니다. 나중에 안드로이드가 업그레이드되면서 이러한 문제가 자연스럽게 해결될 수 있기를 바랍니다.

앞서 설명한 바와 같이 오류가 발생하는 "AVD"에 대한 "Details" 정보를 확인해 보면, 그림과 같이 경로 인식에 문제가 있을 만한 정보를 확인할 수 있습니다.



5.4 AVD 홈폴더 변경

AVD 홈폴더를 변경하려면, 다음과 같이 합니다.

스텝 1

명령 프롬프트를 실행합니다.

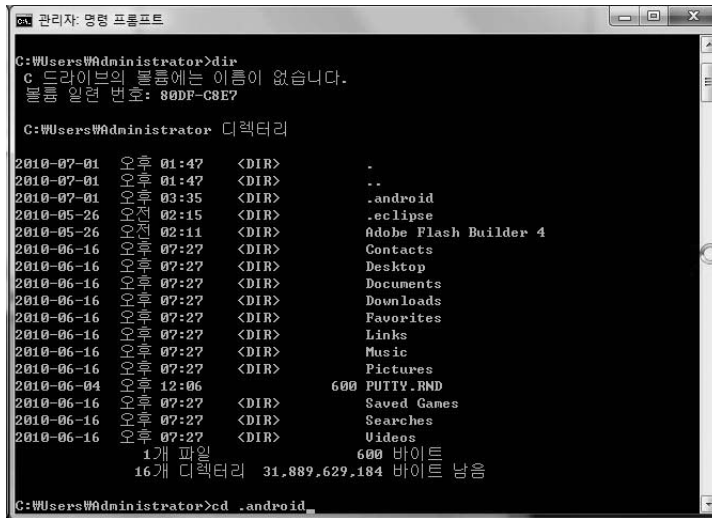
[윈도우 시작 메뉴 > 모든 프로그램 > 보조 프로그램 > 명령 프롬프트]



스텝 2

"명령 프롬프트"에서 ".android" 폴더로 이동합니다.

명령 프롬프트를 실행하면, 기본 위치가 윈도우에서 로그인한 "사용자 계정 폴더"로 잡혀 있습니다. "사용자 계정 폴더"에서 "dir" 명령을 실행하면, 그림과 같이 ".android" 폴더를 확인할 수 있을 것입니다. 이 폴더에 안드로이드에 대한 환경 파일들이 존재합니다. "cd .android" 명령을 실행하여 ".android" 폴더로 이동합니다.



```
C:\Users\Administrator>dir
C 드라이브의 볼륨에는 이름이 없습니다.
볼륨 일련 번호: 80DF-C8E7

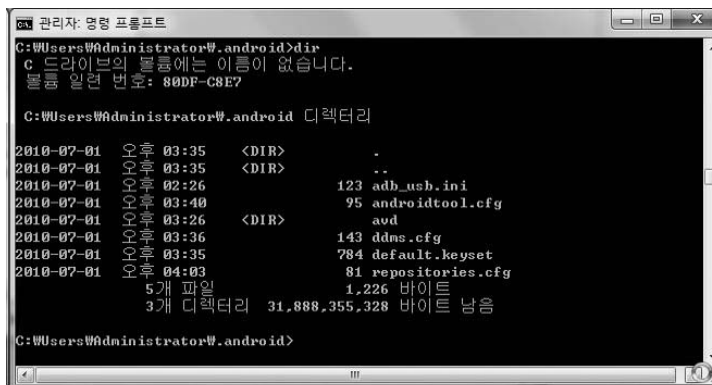
C:\Users\Administrator 디렉터리

2010-07-01 오후 01:47 <DIR>      .
2010-07-01 오후 01:47 <DIR>      ..
2010-07-01 오후 03:35 <DIR>      .android
2010-05-26 오전 02:15 <DIR>      .eclipse
2010-05-26 오전 02:11 <DIR>      Adobe Flash Builder 4
2010-06-16 오후 07:27 <DIR>      Contacts
2010-06-16 오후 07:27 <DIR>      Desktop
2010-06-16 오후 07:27 <DIR>      Documents
2010-06-16 오후 07:27 <DIR>      Downloads
2010-06-16 오후 07:27 <DIR>      Favorites
2010-06-16 오후 07:27 <DIR>      Links
2010-06-16 오후 07:27 <DIR>      Music
2010-06-16 오후 07:27 <DIR>      Pictures
2010-06-04 오후 12:06          600 PUTTY.RND
2010-06-16 오후 07:27 <DIR>      Saved Games
2010-06-16 오후 07:27 <DIR>      Searches
2010-06-16 오후 07:27 <DIR>      Videos
                1개 파일          600 바이트
                16개 디렉터리  31,889,629,184 바이트 남음

C:\Users\Administrator>cd .android
```

스텝 3

"dir" 명령으로 ".android" 폴더의 내용을 확인해보면, 그림과 같이 "avd" 폴더가 나타날 것입니다. 이 폴더의 위치를 AVD가 인식하지 못하면 문제가 발생합니다.



```
C:\Users\Administrator\.android>dir
C 드라이브의 볼륨에는 이름이 없습니다.
볼륨 일련 번호: 80DF-C8E7

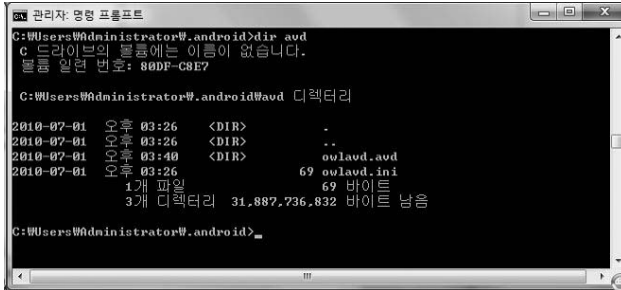
C:\Users\Administrator\.android 디렉터리

2010-07-01 오후 03:35 <DIR>      .
2010-07-01 오후 03:35 <DIR>      ..
2010-07-01 오후 02:26          123 adb_usb.ini
2010-07-01 오후 03:40          95 androidtool.cfg
2010-07-01 오후 03:26 <DIR>      avd
2010-07-01 오후 03:36          143 ddms.cfg
2010-07-01 오후 03:35          784 default.keyset
2010-07-01 오후 04:03          81 repositories.cfg
                5개 파일          1,226 바이트
                3개 디렉터리  31,888,355,328 바이트 남음

C:\Users\Administrator\.android>
```

스텝 4

"dir avd" 명령으로 "avd" 폴더 안에 있는 내용을 확인해보면, AVD에 관련된 환경 파일들을 확인할 수 있습니다.



```
C:\Users\Administrator\W.android>dir avd
c 드라이브의 볼륨에는 이름이 없습니다.
볼륨 일련 번호: 80DF-C8E7

C:\Users\Administrator\W.android\avd 디렉터리

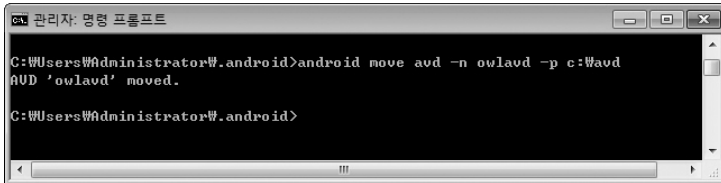
2010-07-01 오후 03:26 <DIR> .
2010-07-01 오후 03:26 <DIR> ..
2010-07-01 오후 03:40 <DIR> owlavd.avd
2010-07-01 오후 03:26        69 owlavd.ini
                1개 파일            69 바이트
                3개 디렉터리  31,887,736,832 바이트 남음

C:\Users\Administrator\W.android>
```

스텝 5

"avd" 폴더를 AVD가 인식할 수 있도록 다음과 같이 이동시킵니다.

- "avd" 폴더 이동 명령 : `android move avd -n 가상기기 이름 -p 이동할 경로`



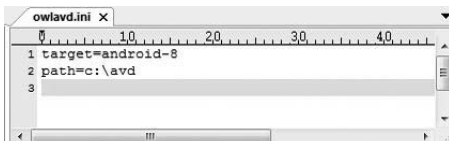
```
C:\Users\Administrator\W.android>android move avd -n owlavd -p c:\w\avd
AVD 'owlavd' moved.

C:\Users\Administrator\W.android>
```

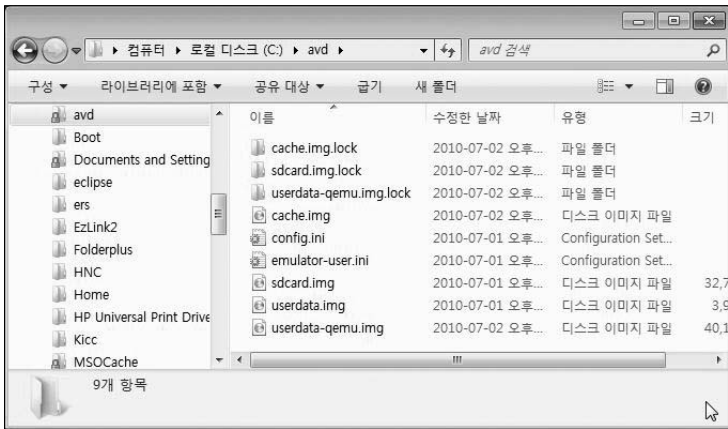
[참고: android.bat] "android" 명령은 "[안드로이드 SDK 설치경로]\tools\android.bat" 명령을 의미합니다. 앞에서 "안드로이드 툴 경로"를 이미 시스템 환경 변수 "Path"에 인식시켜 놓았기 때문에 어느 경로에서든 지 "android.bat" 명령을 전체 경로를 입력하지 않아도 실행할 수 있는 것입니다.

참고 .android\avd 폴더의 역할

- "~.android\avd\가상기기명.ini" 파일 : 안드로이드 버전과 "가상기기 홈 폴더"의 위치를 지정하는 환경 파일입니다.



- "~.android\avd\가상기기명.avd" 폴더 : 가상기기가 구동하는데 필요한 환경 파일이나 디스크 이미지 파일 등이 있으며, 필요에 따라 이 폴더에 임시 파일을 생성하는 등 "AVD(가상기기) 홈폴더" 역할을 합니다.



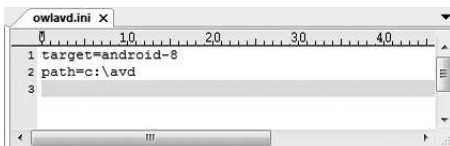
위와 같은 일련의 작업들은 "AVD 홈폴더"인 "~.android\avd\가상기기명.avd"를 AVD가 인식하는데 문제가 없는 위치로 이동하는데 목적이 있습니다. 따라서, 다음과 같이 "AVD 홈폴더"를 이동시킬 수도 있습니다.

스텝 1

윈도우 탐색기에서 "~.android\avd" 폴더의 "가상기기명.avd" 폴더를 드래그하여 다른 위치(사례: "c:\")로 이동합니다. 필요에 따라 "가상기기명.avd" 폴더명을 "avd"라고 변경해도 됩니다. 이렇게 하면 위의 방법과 같이 "AVD 홈폴더"는 "c:\avd"로 변경됩니다.

스텝 2

AVD가 구동할 때 변경한 "AVD 홈폴더"를 인식할 수 있도록 "~.android\avd\가상기기명.ini" 파일의 내용을 다음과 같이 수정한 후 저장합니다.



6 Mac OS X용 안드로이드 개발 환경 구축

안드로이드는 자바 기반의 프로그램이므로 자바를 지원하는 모든 OS에서 개발 환경을 구축할 수 있습니다. 애플사의 "iPhone/iPad"용 앱은 "Mac OS X" 기반이기 때문에 "Mac OS X"를 사용해야 합니다. 개발자의 입장에서는 안드로이드와 애플 앱으로 모두 개발해야 하는 경우가 점점 늘어날 것입니다. 다행히 안드로이드는 다음에서 설명하는 바와 같이 "Mac OS X"에 대한 개발 환경을 잘 지원해주고 있습니다.

애플 앱이 선구자 역할을 했음에도 불구하고, 이를 칭송하고 더욱 발전시키는 길을 찾아가지 않고, 마치 안드로이드가 애플과 큰 전쟁이라도 해야 하는 것처럼 오도하는 혼탁한 분위기가 안타까울 때가 있습니다. 어떤 존재이든 세상에 꼭 하나만 존재할 필요는 없습니다. 서로 선의의 경쟁을 하며 한때는 앞서기도 하고, 뒤지기도 하면서 상대의 허점을 험담하기보다는 타산지석으로 삼아 발전하는 것이 올바른다는 것은 누구나 잘 아는 사실입니다. 장인 정신을 공고히 해야 기술이 탄탄해지듯이, 개발자들이 이러한 근시안적인 분위기에 휩쓸리지 않는다면, 한때의 혼란함은 조만간에 조화롭게 자리 잡을 수 있을 것입니다.

애플 앱과 안드로이드 앱은 기술의 바탕이 다르기 때문에 서로 장단점이 있고, 상보 관계에 있을 수 있습니다. 애플 앱은 자바 기반의 안드로이드보다 운영체제의 기반이 매우 탄탄합니다. 필자는 "Mac OS X"가 매우 오랜 시간을 투자하고 많은 고뇌를 극복한 후에 탄생한 운영체제임을 목격해왔고, 그 성능 또한 지금까지 봐왔던 운영체제와는 부러울 정도로 차원이 달랐습니다. 이러한 기반을 가진 애플 앱은 그만큼 탁월한 성능을 낼 수 있고, 훌륭한 화면으로 그 결과물을 보여 주고 있습니다.

반면, 안드로이드는 과거 애플과 마이크로소프트가 그래왔듯이, 꼭 그와 양상이 같지는 않지만, 접근하기 쉽고 대중적인 배경을 가지고 있습니다. 통계적으로도 애플 앱보다는 그 잠재 시장이 매우 넓습니다. 자바를 기술 기반으로 하고 있기 때문에, 인터넷 기반의 수많은 개발자들이 손쉽게 접근할 수 있어 개발 붐을 일으키고 있습니다.

일부 리더들이 과격한 방식으로 애플 앱과 안드로이드 앱의 음지의 대립각을 내세우고 있지만, 많은 대중들은 "이들이 왜 그런 것인지 이해할 수 없다"는 반응이거나 무관심합니다. 오히려, 관계자 이외의 대부분의 대중들은 이들보다 더 넓은 시야에서 앱 시장을 보고 있습니다.

근본이 공고한 몇 마디의 말이나 설정된 분위기는 세상을 움직일 수도 있지만, 존재의 근원을 벗어난 말이나 설정된 분위기는 오래 가지는 못합니다. 아무리 무관심한 대중을 대상으로 한다고 반박해도, 이러한 사실은 유사 이래의 역사가 잘 예시하고 있습니다. 주가가 오르내리며 반복하는 것과 같이

"프로그램의 세계"에서도 "반복적인 역사의 흐름"이 있습니다. 한편으로는, 역사가 똑같이 반복하고 있는 것 같이 보이지만, 공존하려는 꾸준한 노력이 있기에 상황은 더욱 발전해가는 것입니다. "프로그램의 세계"에서도 의식을 깨우려는 꾸준한 노력이 있어 더욱 발전해 갈 것입니다.

필자는 이와 같은 이유로 "Mac OS X"에 안드로이드 개발 환경을 구축하는 과정을 보이고, "상호 존재의 인정"과 "조그마한 공존의 유익함"을 보여 주려합니다. "공존의 생각"이 널리 퍼지는 것은 결국, 이 조그마한 프로그램 세상에서도 나를 포함한 사용자에게 편리함과 유익함을 나누어 주게 됩니다.

6.1 JDK (Java Development Kit) 설치 확인

"Mac OS X"에 안드로이드 개발 환경을 구축하는 과정은 "MS Windows"에서와 크게 다르지 않습니다. "MS Windows" 운영체제에서는 안드로이드 개발 기반인 "JDK"를 별도로 설치해야 합니다. 하지만 "Mac OS X" 운영체제에서는 버전에 따라 차이가 있을 수 있기는 하지만, 대부분 "Mac OS X"가 설치될 때, 이미 "JDK"가 설치되어 있습니다.

"JDK"가 설치되어 있는지 확인하려면, 다음과 같이 터미널을 이용하여 확인할 수 있습니다.

스텝 1

다음과 같이 "터미널" 응용 프로그램을 찾아 실행합니다.

- 터미널 위치 : 응용 프로그램 > 유틸리티 > 터미널

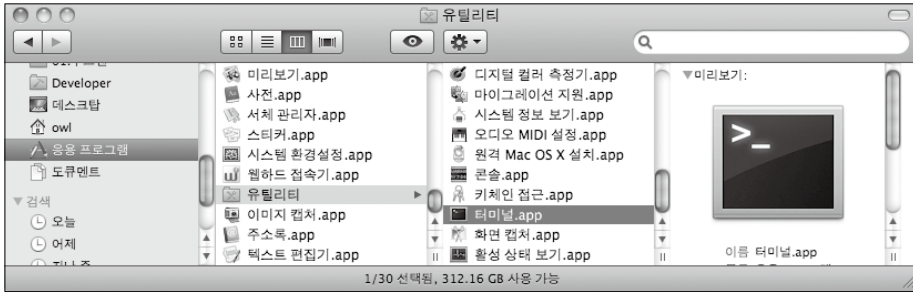
"Mac OS X"의 응용 프로그램은 그림과 같이 "Finder"를 통해 찾을 수도 있고, "독(Dock)"을 통해서도 찾을 수 있습니다.

- "Finder"를 통해 "터미널"을 실행할 경우

"독(Dock)"에서 "Finder"를 클릭하여 "Finder"를 실행합니다.



"Finder"가 나타나면, 그림과 같이 [응용 프로그램 > 유틸리티 > 터미널]을 더블클릭하여 실행합니다.



- "독(Dock)"을 통해 "터미널"을 실행할 경우

"독(Dock)"에서 "응용 프로그램" 폴더를 클릭하여, "독 메뉴(Dock Menu)"를 엽니다.



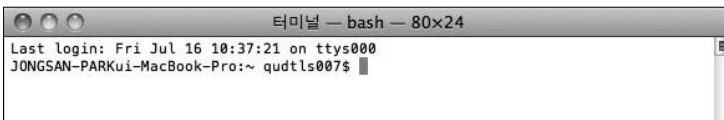
"응용 프로그램 독 메뉴"에서 "유틸리티" 폴더를 클릭합니다.



"유틸리티 독 메뉴"에서 "터미널" 응용 프로그램 파일을 클릭하여 "터미널"을 클릭합니다.

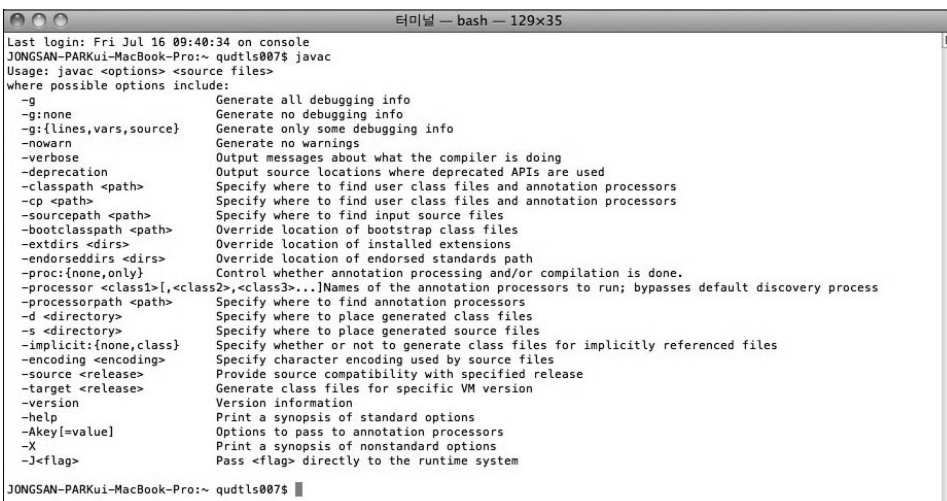


"터미널"은 그림과 같이 "MS Windows"의 "명령 프롬프트(도스 화면)"이라고 생각하면 이해가 쉬울 것입니다. 정확하게 말하자면, "Mac OS X"는 유닉스를 기반으로 한 운영체제이므로 유닉스 명령어를 사용하는 "유닉스 터미널"이 나타난 것입니다.



스텝 2

자바 컴파일러가 설치되어 있는지 확인하기 위해 그림과 같이 "javac" 명령을 실행해봅니다.



스텝 3

설치되어 있는 자바 버전을 확인하기 위해 "java -version" 명령을 실행해봅니다.

```
터미널 — bash — 93x18
Last login: Fri Jul 16 09:42:38 on ttys000
JONGSAN-PARKui-MacBook-Pro:~ qudtls007$ java -version
java version "1.6.0_20"
Java(TM) SE Runtime Environment (build 1.6.0_20-b02-279-10M3065)
Java HotSpot(TM) 64-Bit Server VM (build 16.3-b01-279, mixed mode)
JONGSAN-PARKui-MacBook-Pro:~ qudtls007$
```

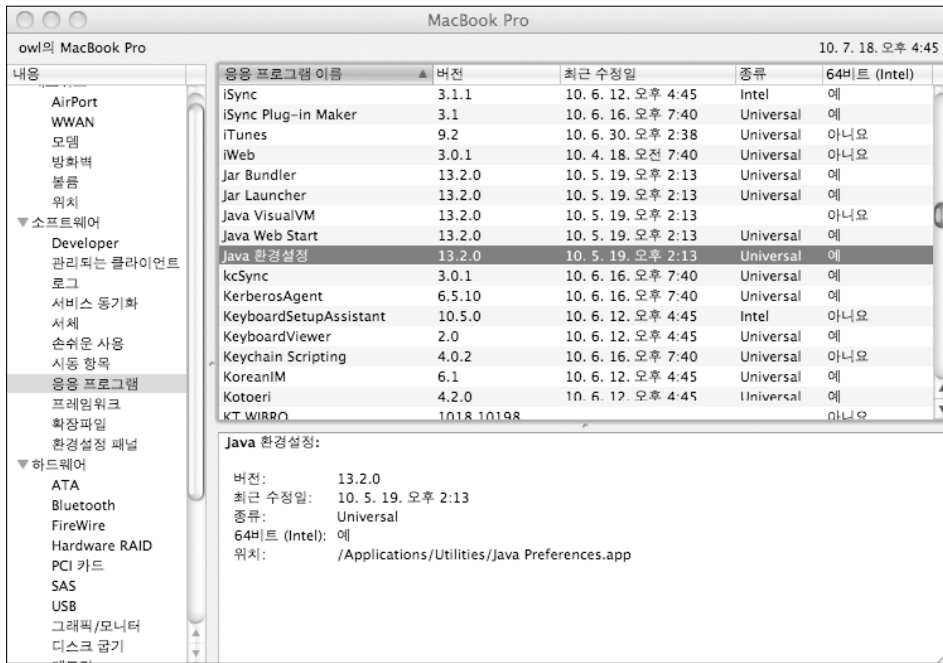
필자의 맥에는 "Mac OS X Snow Leopard"가 설치되어 있어 자바가 "1.6.0_20" 버전으로 설치되어 있습니다. "Mac OS X"에 포함된 패키지들은 주기적으로 "Mac OS X"가 자동 업데이트될 때, 같이 업데이트합니다.

참고 매킨토시 정보를 통한 자바 버전 확인

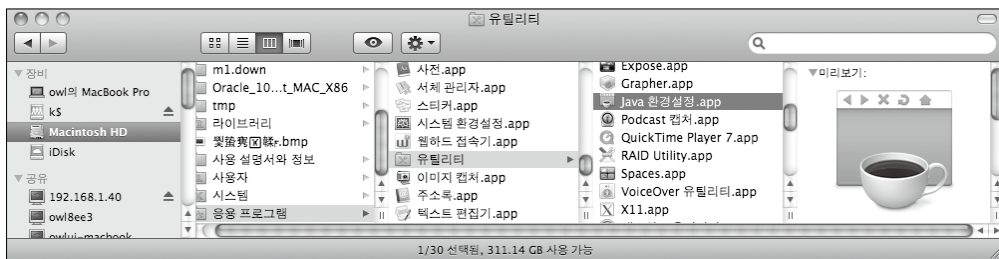
"Mac OS X"에 포함되어 있는 소프트웨어 패키지 정보를 확인하는 또 다른 방법이 있습니다. [애플 > 이 매킨토시에 관하여] 메뉴를 실행하고, "이 매킨토시에 관하여" 대화상자가 나타나면 "자세한 정보.." 버튼을 클릭합니다.



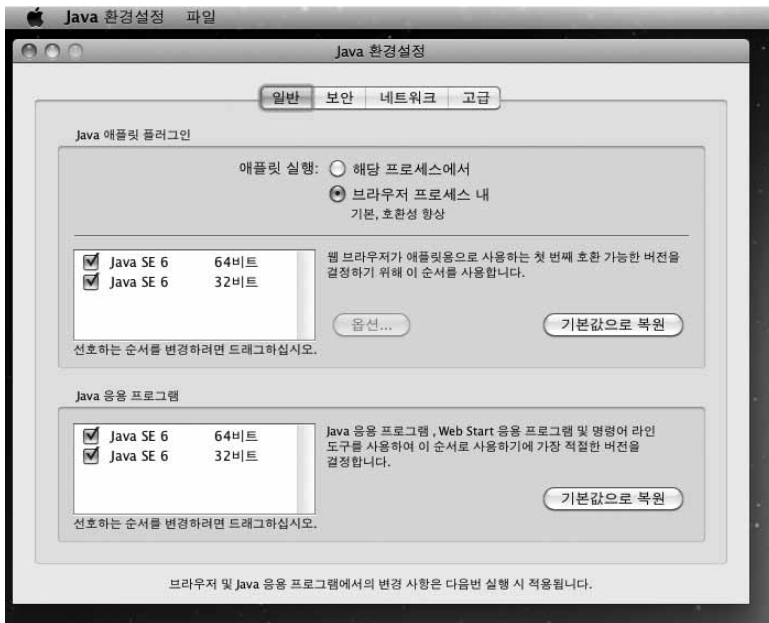
자신의 매킨토시에 관한 자세한 정보를 살펴보면, 그림과 같이 "java 환경 설정"과 자바 관련 응용 프로그램이 설치되어 있는 것을 확인할 수 있습니다. "Mac OS X"에서 패키지로 제공하는 자바는 "Java 환경 설정"에 의해 제어됩니다. 만일 여러 가지의 자바 버전이 설치되어 있다면, "Mac OS X"에서 기본 버전으로 사용할 자바 버전을 "java 환경 설정"에서 정의합니다.



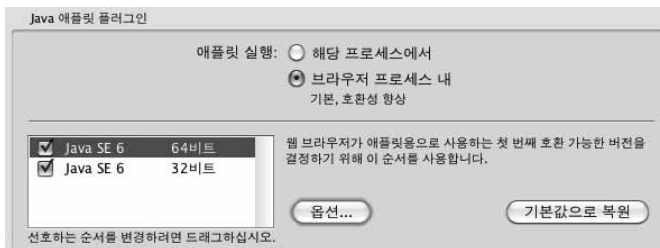
그림과 같이 [응용 프로그램 > 유틸리티 > Java 환경 설정]을 실행합니다.



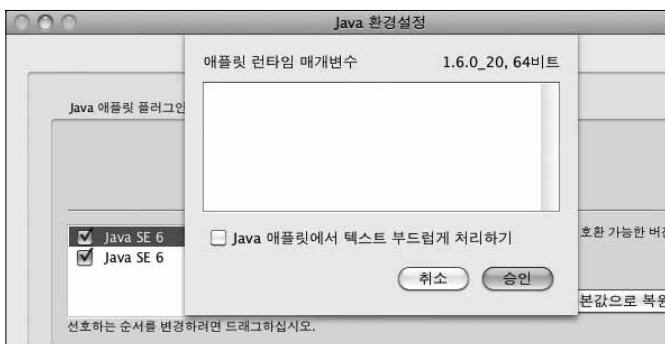
"Java 환경 설정" 대화상자를 보면, 자신의 시스템에서 사용하고 있는 "JDK"가 어떤 버전인가를 확인할 수 있습니다. 그림에서는 "Java SE 6" 즉, "Java Standard Edition" 중 1.6.x 버전을 사용하고 있습니다.



좀 더 상세한 자바 버전을 확인하려면, 그림과 같이 목록에서 "Java SE 6"을 선택하고, "옵션..." 버튼을 클릭합니다.



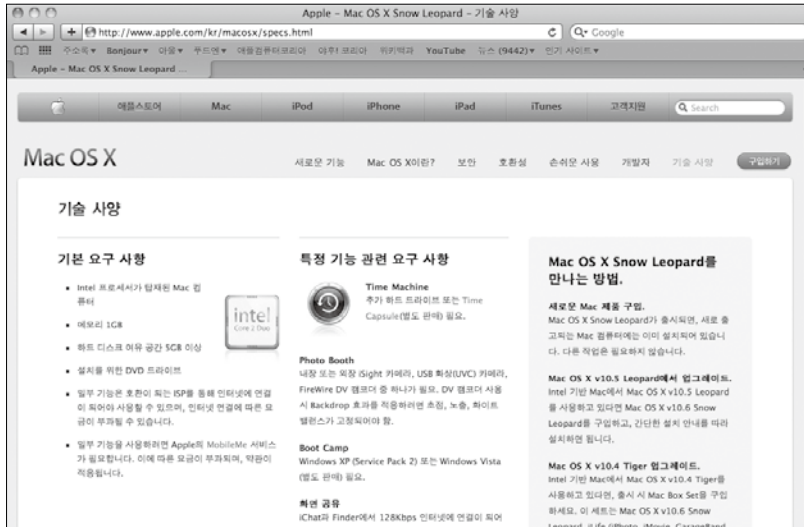
옵션창을 통해 "Java SE 6"에 대한 상세한 자바 버전을 확인할 수 있습니다.



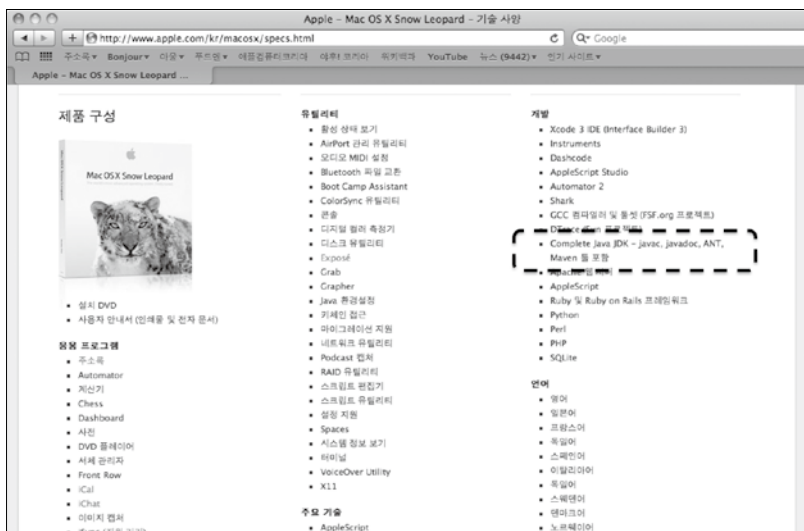
참고 "Mac OS X" 스펙 정보를 통한 자바 패키지 확인

다음과 같이 애플 사이트를 통해서도 확인할 수 있습니다.

- "Mac OS X"의 스펙 정보 주소 : "<http://www.apple.com/kr/macosex/specs.html>"



이 페이지에서는 "Mac OS X"에 설치되어 있는 상세 스펙에 대해서 열거를 하고 있는 것을 볼 수 있습니다. 개발 항목을 보면 "Complete Java JDK - javac, javadoc, ANT, Maven 툴 포함"이라고 기재가 되어있는 것으로 보아 JDK가 설치되어 있다는 것을 알 수 있습니다.



6.2 Mac OS X용 이클립스 설치

- 설치본 : eclipse-jee-helios-macosx-cocoa-x86_64.tar
- 배포 사이트 : <http://www.eclipse.org/>

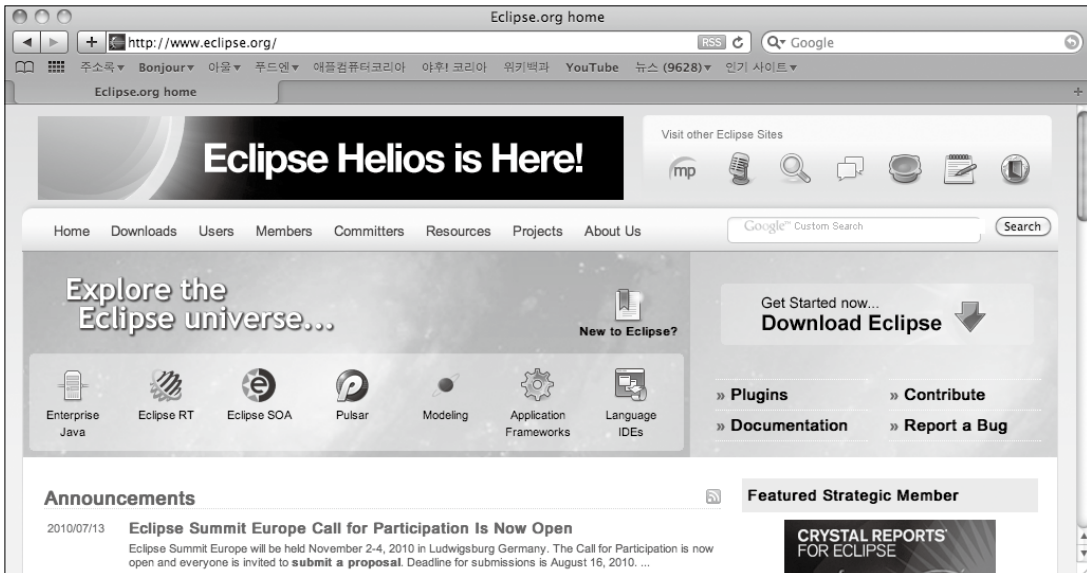
스텝 1

독(Dock)에서 "Safari"를 실행합니다.



스텝 2

사파리(Safari) 브라우저를 통해 "<http://www.eclipse.org/>" 이클립스 사이트에 접속합니다.



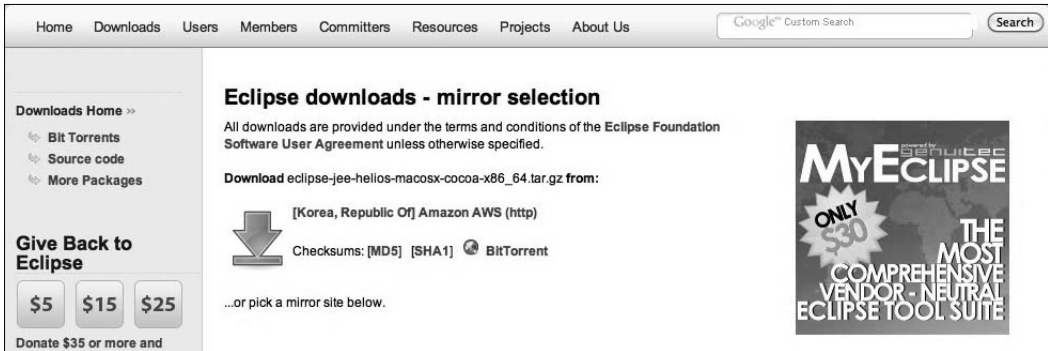
스텝 3

사이트의 상단 메뉴에서 "Downloads" 버튼을 클릭하여 다운로드 페이지로 이동합니다.



스텝 4

이클립스 사이트는 친절하게도 사용자의 운영체제를 자동으로 인식해서 "Mac OS X"용 JDK를 바로 다운로드할 수 있도록 안내합니다. 이 페이지에서 "Eclipse IDE for Java EE Developers (204MB)"의 "Mac OS X 64 bit"를 클릭하여 다운로드 페이지로 이동합니다.



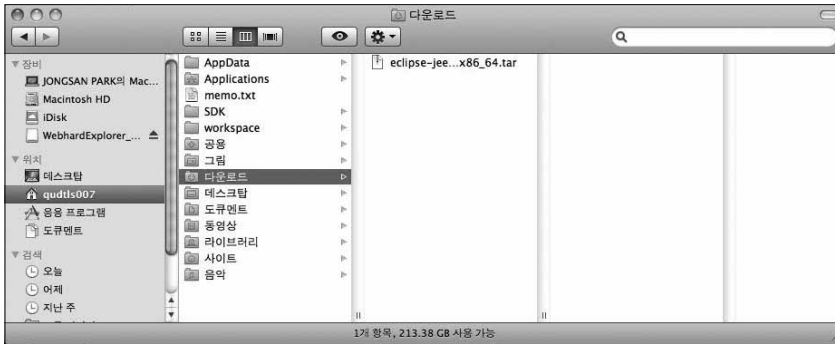
스텝 5

다운로드 페이지에서 초록색 화살표를 클릭하면, 그림과 같이 다운로드합니다.



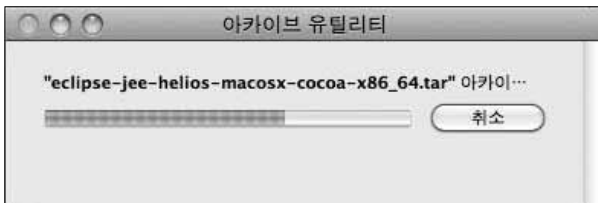
스텝 6

다운로드 받은 폴더로 이동하여, 이클립스 설치 파일을 더블클릭합니다.



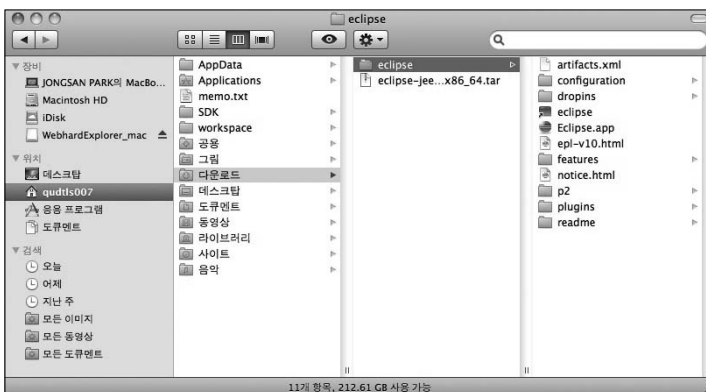
스텝 7

"아카이브 유틸리티"가 자동으로 이클립스 설치 파일의 압축을 해제합니다.



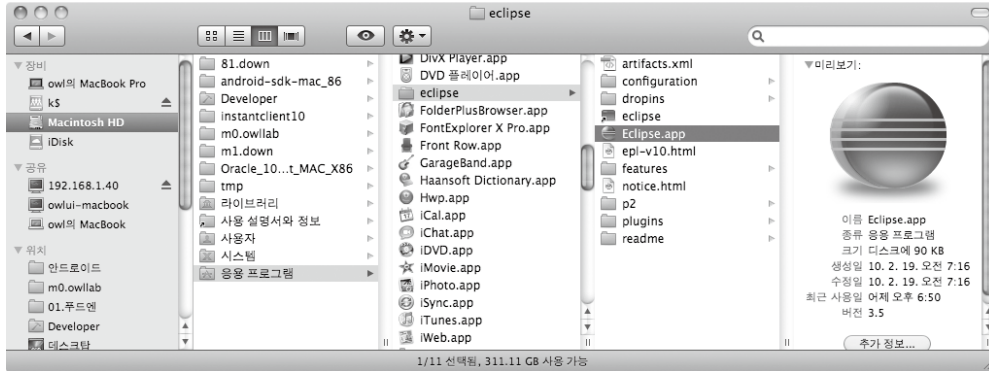
스텝 8

압축이 해제되면, 동일한 경로에 "eclipse" 폴더가 생성이 되었습니다. "eclipse" 폴더를 확인해보면 그림과 같이 "Eclipse.app" 실행 파일이 있는 것을 확인할 수 있습니다.



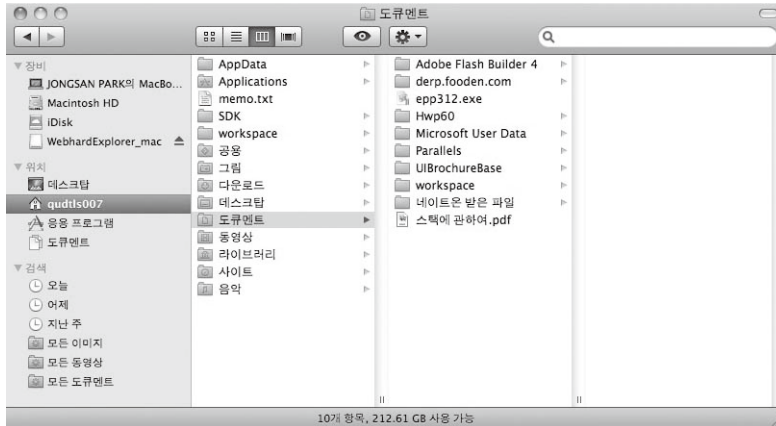
스텝 9

앞서 설명한 바와 같이 이클립스는 압축을 해제하고, 수동으로 "eclipse" 폴더를 적절한 위치로 이동합니다. 본서에서는 그림과 같이 "응용 프로그램" 폴더로 이동하였습니다.



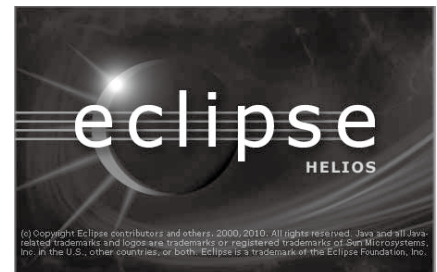
스텝 10

그림과 같이 "Finder"를 이용하여 적절한 위치에 "workspace" 폴더를 생성합니다.



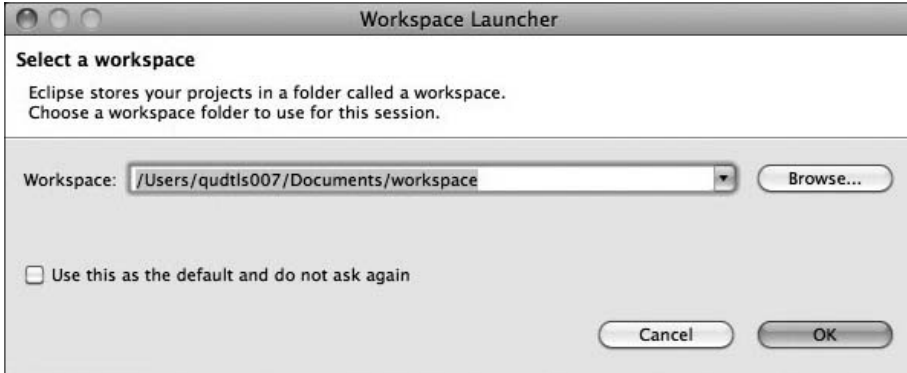
스텝 11

"Finder"에 앞서 설치한 이클립스 폴더에 있는 "Eclipse.app" 실행 파일을 더블클릭하면, 이클립스 로딩 화면과 함께 이클립스가 시작됩니다.

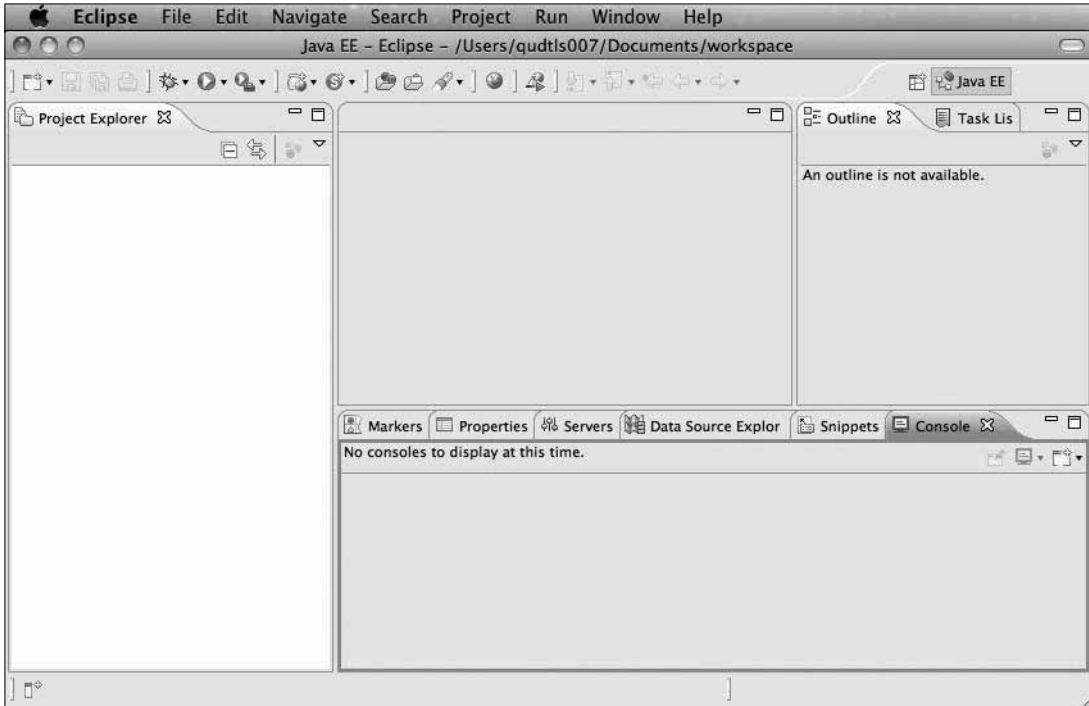


스텝 12

몇 초가 지난 후에는 "Workspace Launcher" 창이 나오는데, 앞에서 생성한 "workspace" 폴더를 지정합니다. "Use this as the default and do not ask again"을 체크하면, 다음에 이클립스를 실행할 때, workspace 경로를 다시 묻지 않고, 현재 지정한 경로를 기본 workspace로 사용합니다.

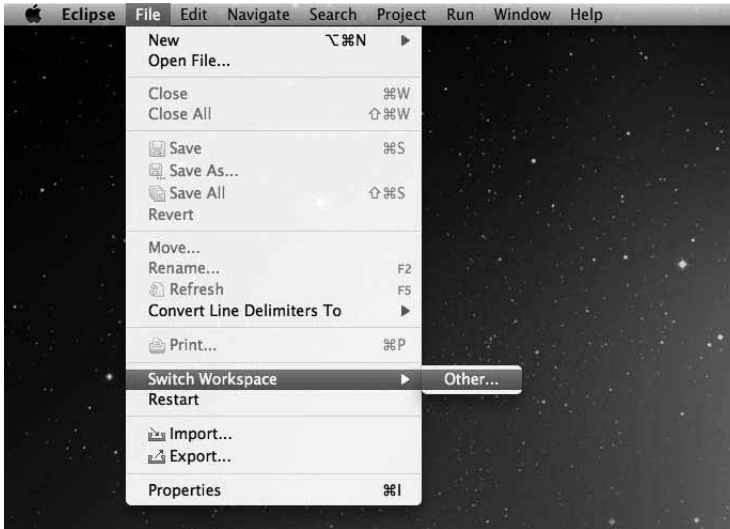


이클립스가 성공적으로 실행되면 그림과 같이 나타납니다. 이클립스는 자바 기반의 응용 프로그램이므로 "MS Windows"에서의 이클립스와 다를 바 없습니다.



참고 workspace 변경

프로젝트를 진행하다가 "workspace"를 변경하는 경우가 발생할 수도 있습니다. 이때 [File > Switch Workspace > Other...] 메뉴를 실행하여 워크스페이스 경로를 변경할 수 있습니다.



참고 "Mac OS X" 독(Dock)에 응용 프로그램 아이콘 유지하기

이클립스를 실행하기 위해 매번 "Finder"로 이클립스 폴더에 들어가서 "Eclipse.app"를 클릭하는 것은 번거롭습니다. "Eclipse.app"를 클릭하여 실행하면 하단의 독(Dock)에 이클립스 아이콘이 자리하고 있는 것을 볼 수 있습니다. 독에 있는 아이콘을 "마우스 오른쪽" 버튼으로 눌러 독의 컨텍스트 메뉴(Context Menu)에서 "옵션" 메뉴에 마우스를 올리면 그림과 같이 "Dock에 유지" 메뉴를 볼 수 있습니다.

자주 쓰는 응용 프로그램은 "Dock에 유지" 메뉴를 눌러 지정한 응용 프로그램이 종료되더라도 독에서 응용 프로그램의 아이콘이 사라지지 않고 유지되게 합니다.

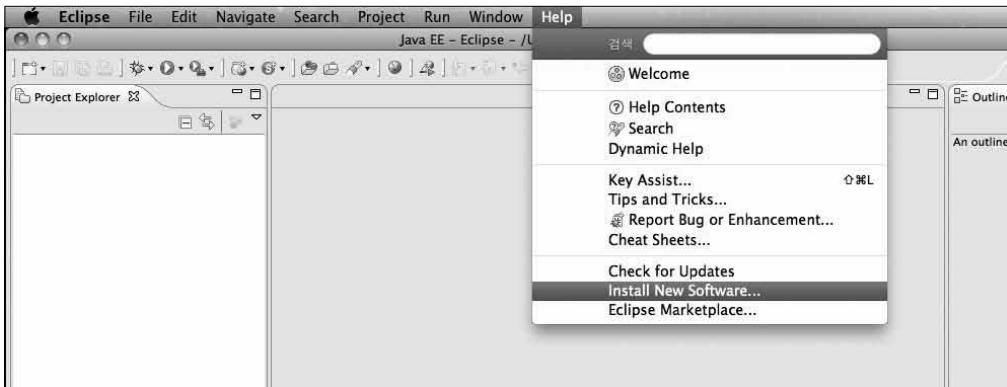


6.3 Mac OS X용 ADT 플러그인 설치

이클립스에서 안드로이드를 개발하려면, 이클립스 플러그인으로 ADT(Android Development Tool)를 설치해야 합니다.

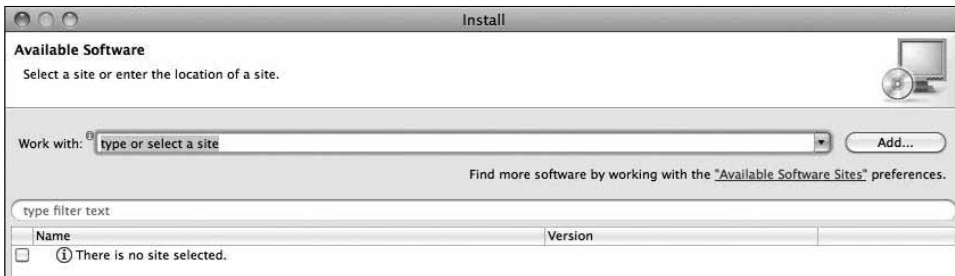
스텝 1

이클립스 상단 메뉴에서 [Help > Install New Software]를 선택합니다.



스텝 2

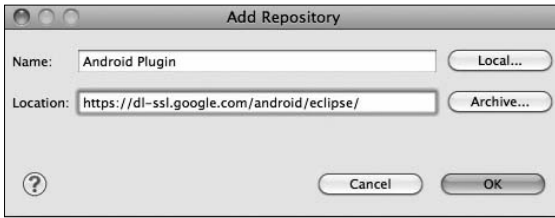
"Install" 화면에서 "Add..." 버튼을 클릭합니다.



스텝 3

"Add Repository" 창이 나타나면, ADT를 배포하는 사이트 주소를 정의합니다.

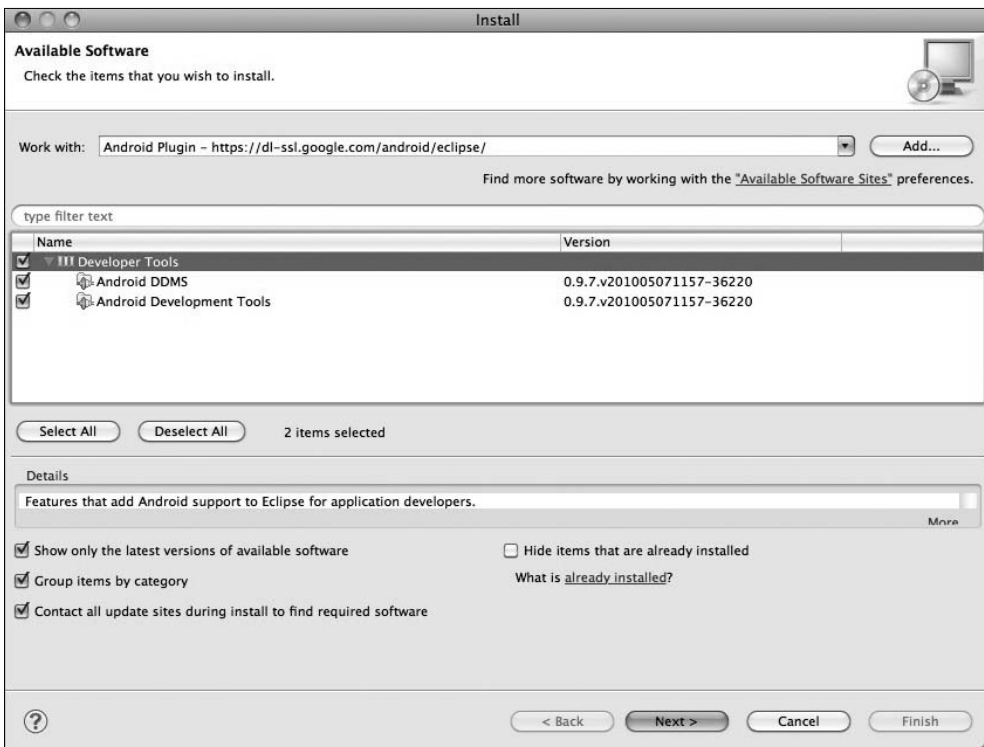
"Name"에 "Android Plugin"이라고 입력하고(Name은 임의로 주셔도 상관은 없습니다.), "Location"에는 ADT 사이트 주소로 "https://dl-ssl.google.com/android/eclipse/"를 입력한 후, "OK" 버튼을 클릭합니다.



참고 네트워크 보안 문제나 윈도우 방화벽으로 인해 올바르게 작동하지 않을 경우가 있는데 이 경우, ADT 사이트 주소에 기재한 "https"를 "http"로 변경해 시도해 보기 바랍니다.

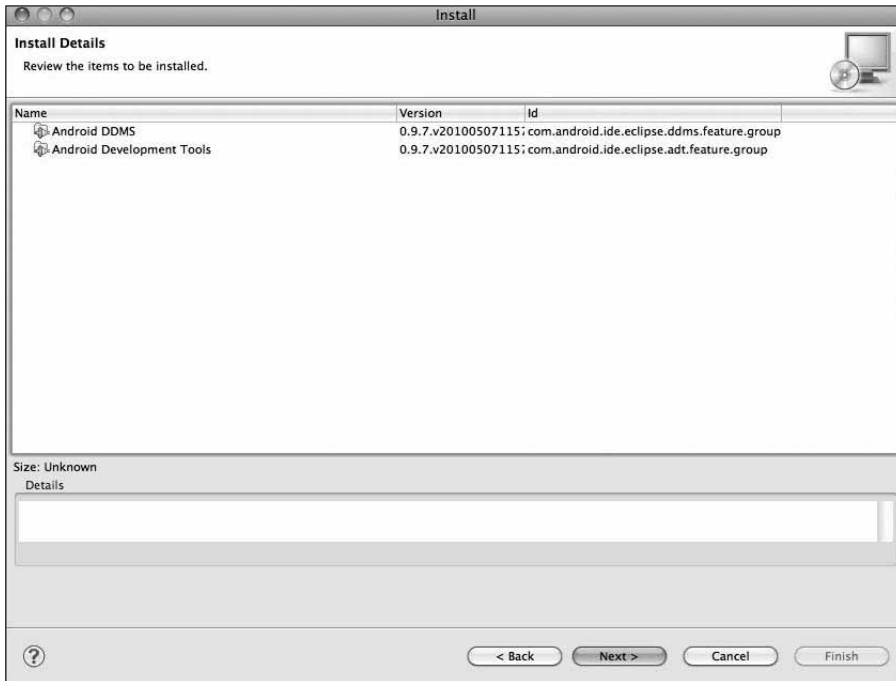
스텝 4

인터넷을 통해 ADT 사이트에 접속이 성공하면, 그림과 같이 설치할 패키지 목록에 "Developer Tools"라고 나타납니다. "Select All"을 선택하고 "Next" 버튼을 클릭합니다.



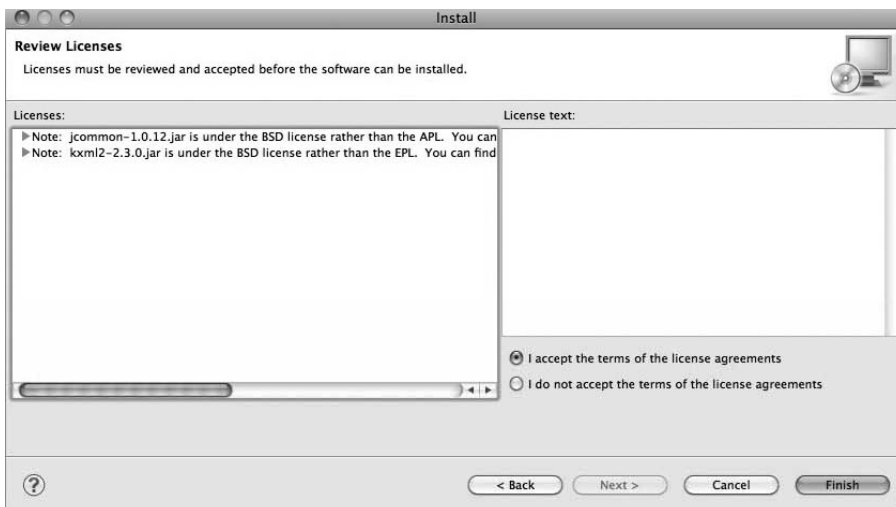
스텝 5

설치할 목록을 확인하고, "Next" 버튼을 클릭합니다.



스텝 6

라이선스 약관에 대한 동의 화면이 나오면, "I accept..."를 선택하고, "Finish" 버튼을 클릭합니다.



스텝 7

설치 과정이 진행되면, 그림과 같이 나타납니다.



스텝 8

설치 과정이 정상적으로 진행되는 중에 선택하는 창이 나옵니다. 설치가 진행 중에 이런 경우가 발생 되면 오류/경고의 판단이 들기 마련이지만, 이 경고 메시지는 설치 중에 서명 누락된 소프트웨어로 인해 중단되었다는 내용인데 "OK" 버튼을 클릭하여 설치를 계속 진행합니다.



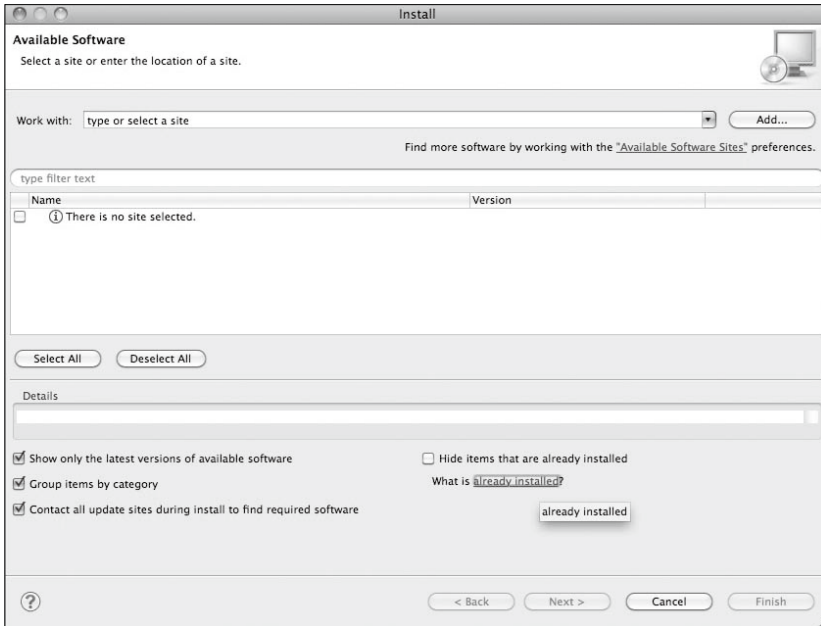
스텝 9

설치가 완료되면 "Software Updates" 창에 "이클립스의 변경 사항이 적용되려면 이클립스를 재시작 해야 된다"는 메시지와 함께 재시작을 권장합니다. "Restart Now" 버튼을 클릭하여 이클립스를 재시작합니다.



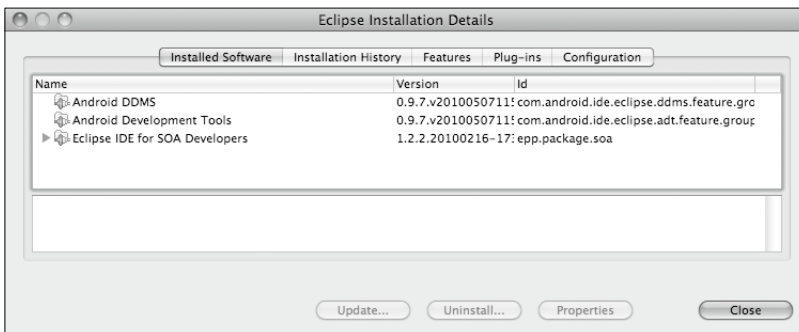
스텝 10

ADT 플러그인 설치가 잘 되었는지 확인하고, 필요에 따라 "Update", "Uninstall"할 수 있는 방법이 있습니다. 상단 메뉴에서 [Help > Install New Software...]를 다시 실행해서 "Install" 화면이 나타나면, "already installed"를 클릭합니다.



스텝 11

"Eclipse Installation Details" 창에서 이클립스에 설치된 상세 항목들을 볼 수 있습니다. 설치된 목록을 선택하면, 하단에 비활성화되어 있는 "Update" 버튼과 "Uninstall" 버튼이 활성화 상태로 변경됩니다. 이 버튼을 이용하여 필요에 따라 패키지를 업데이트하거나 설치 해제하고 다시 설치할 수 있습니다. "Properties" 버튼은 설치된 항목의 상세 속성들을 확인할 수 있습니다.



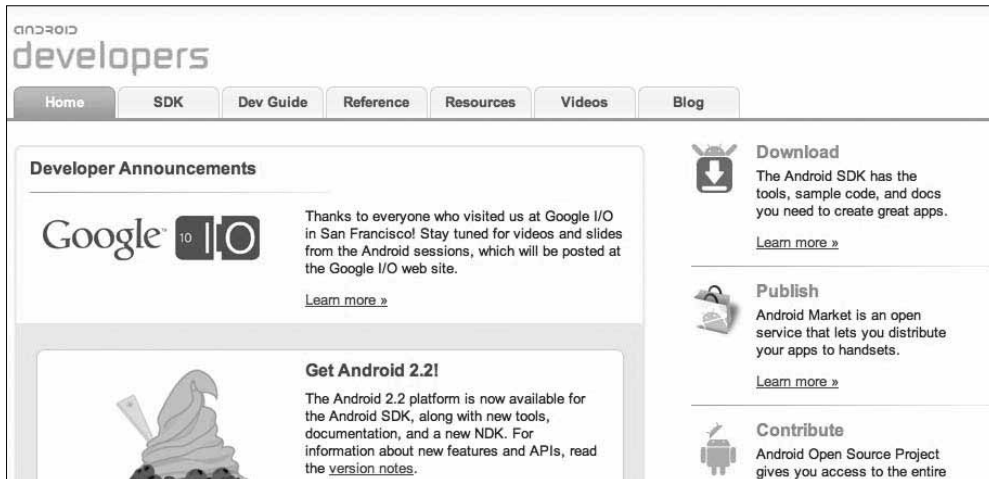
6.4 Mac OS X용 Android SDK 설치

다음과 같이 안드로이드 개발자 사이트에 접속하여 "Android SDK (Software Development Kit)"을 다운받습니다.

- 설치본 : android-sdk_r06-mac_86.zip
- 배포 사이트 : <http://developer.android.com>

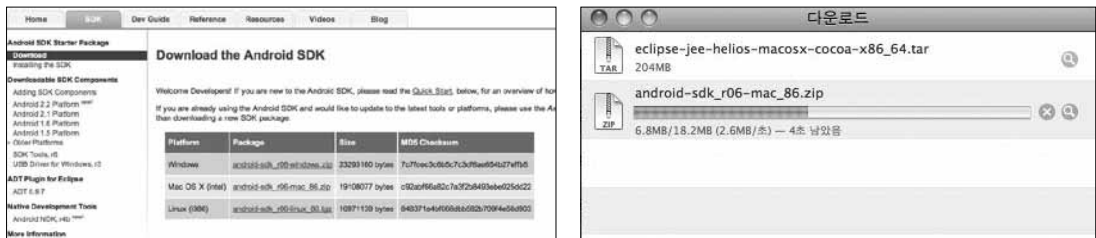
스텝 1

<http://developer.android.com> 사이트 메인 화면의 모습입니다. 상단 탭 메뉴의 "SDK" 버튼을 클릭하거나 본문 오른쪽의 "Download"를 클릭합니다.



스텝 2

다운로드 페이지에서 "Mac OS X(intel)"용 패키지인 "android-sdk_r06-mac_86.zip"를 클릭하여 다운로드 합니다.



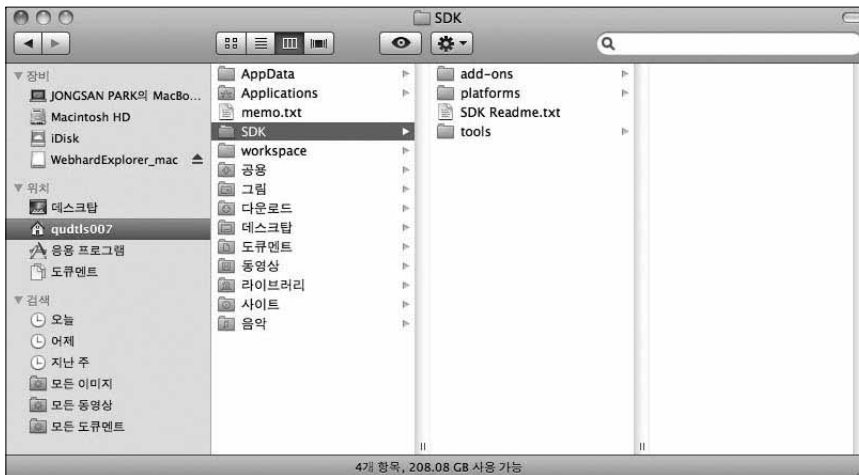
스텝 3

"Mac OS X"의 환경에 따라 조금 다르지만, 기본 설정에 의하면, ".zip" 파일을 다운받으면 자동으로 압축이 해제되어 그림과 같이 "Android SDK" 폴더가 보일 것입니다.



스텝 4

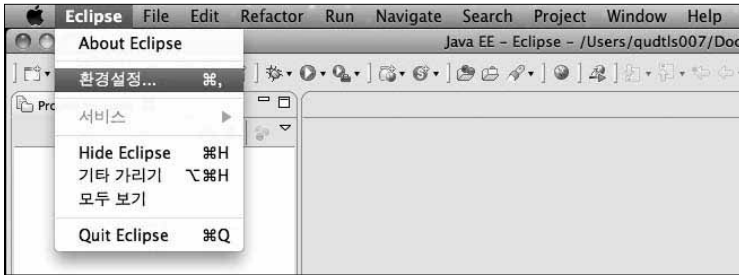
다운받은 "Android SDK" 폴더를 적절한 위치로 이동하고 "Android SDK" 경로를 쉽게 기억할 수 있도록 간단한 이름으로 변경합니다.



6.5 Mac OS X용 Android SDK 환경 설정

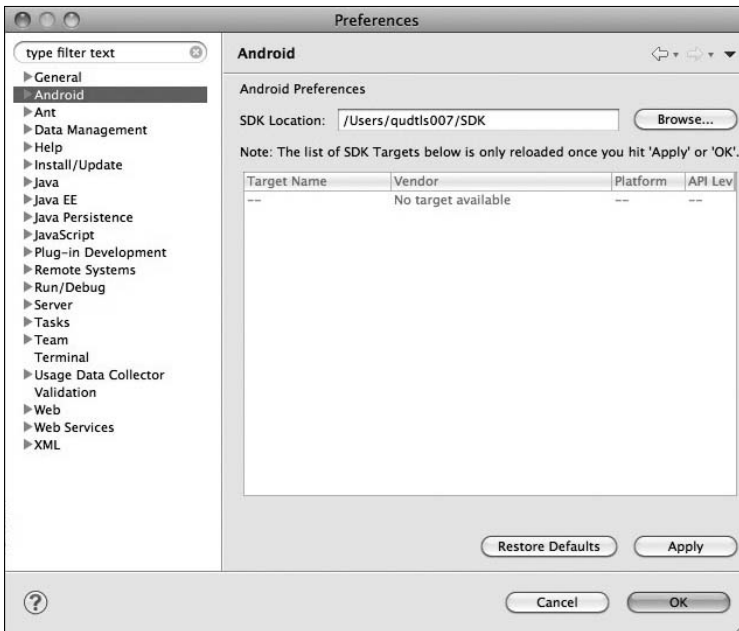
스텝 1

이클립스에서 [Eclipse > 환경 설정...] 메뉴를 실행합니다.



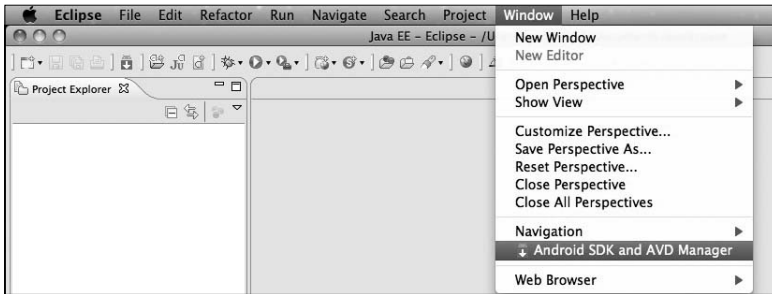
스텝 2

"Preferences" 창에서 "Android" 탭을 클릭합니다. "Android Location" 항목에서 "Browse.."를 클릭하여 "Android SDK"의 경로를 찾아 설정하고 "OK" 버튼을 클릭합니다.



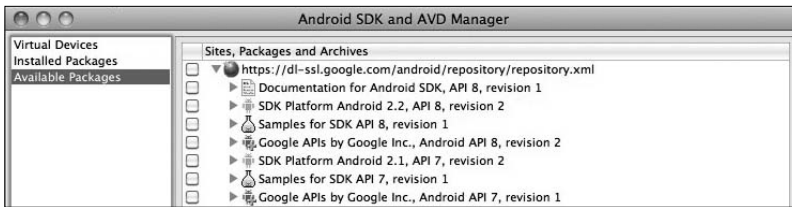
스텝 3

[Window > Android SDK & SDK Manager] 메뉴를 실행합니다.



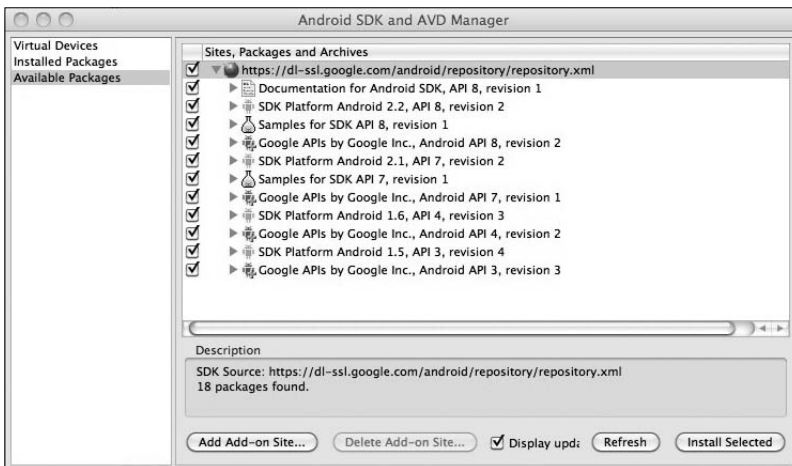
스텝 4

"Android SDK & SDK Manager" 창에서 "Available Packages"를 선택하면 오른쪽에 설치 가능한 목록이 나타납니다.



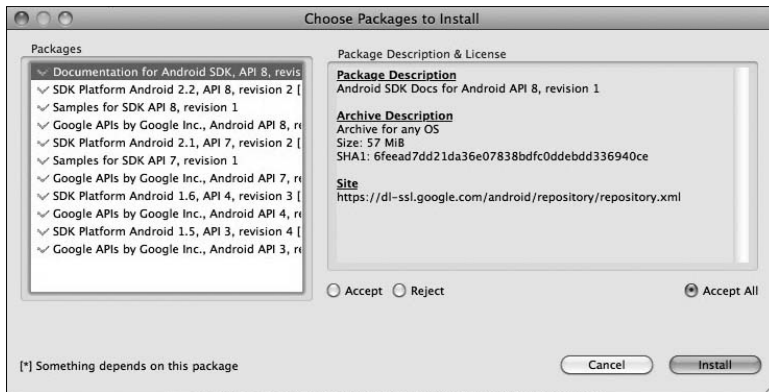
스텝 5

그림과 같이 체크하고, "Install Selected"를 클릭합니다.



스텝 6

"Accept All"을 체크하고 "Install" 버튼을 클릭합니다.



스텝 7

설치 과정이 진행되면, "Installing Archives" 창이 나타납니다.



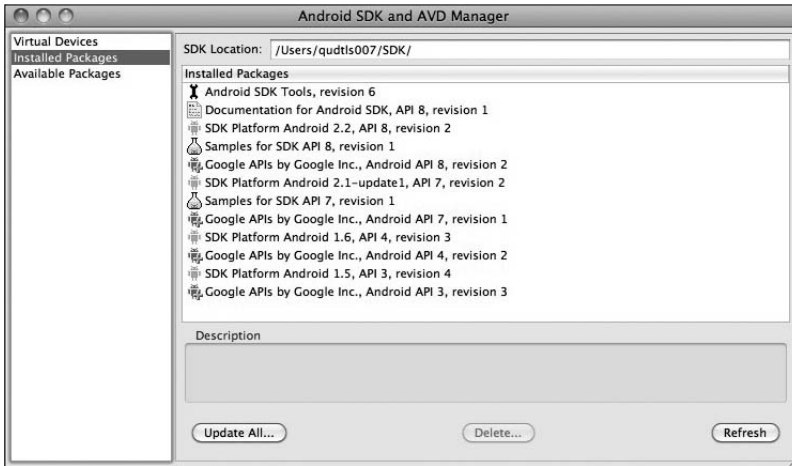
스텝 8

설치가 완료되면, "ADB Restart" 창이 나타나고, "ADB(Android Debug Bridge)가 업데이트되어 ADB를 재시작할 것을 권한다는 메시지가 나타납니다. "Yes" 버튼을 클릭하여 재시작을 합니다.



스텝 9

재시작을 하고 난 후, "Andorid SDK and AVD Manager" 창의 "Installed Packages"를 클릭하면 설치된 패키지들의 목록을 확인할 수 있습니다.



6.6 Mac OS X용 AVD(Android Virtual Device) 생성과 실행

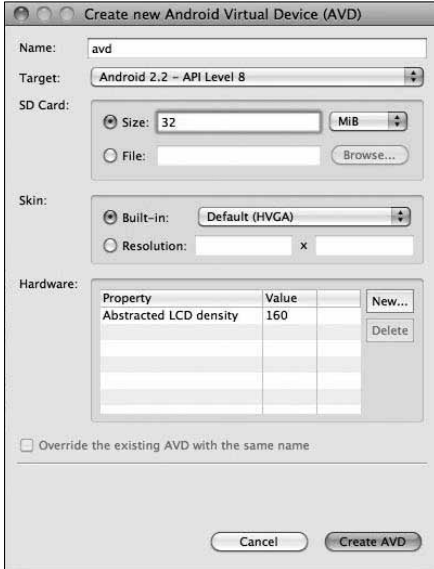
스텝 1

[Window > Android SDK and AVD Manager] 메뉴를 실행하여 "Android SDK and AVD Manager" 창을 엽니다.



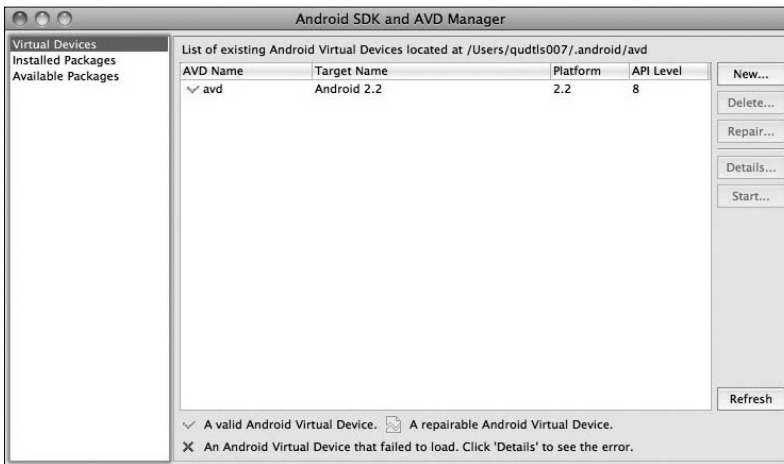
스텝 2

"New..." 버튼을 클릭하여 "Create new Android Virtual Device (AVD)" 창을 엽니다.
그림과 같이 "Name", "Target", "SD Card Size" 등을 정의하고, "Create AVD" 버튼을 클릭합니다.



스텝 3

"AVD"가 성공적으로 생성되었으면, "Virtual Devices" 목록에 생성된 "AVD"를 확인할 수 있습니다.



스텝 4

목록에서 "AVD"를 선택하고, "Details..."를 클릭하면 그림과 같이 "AVD Details(가상기기 상세 정보)"를 확인할 수 있습니다.



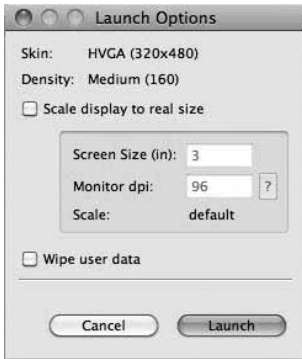
스텝 5

생성한 가상기기를 실행하려면, 목록에서 가상기기를 선택하고, "Start..."를 클릭합니다



스텝 6

"Launch Options" 창이 나타나면, "Launch" 버튼을 클릭합니다.



스텝 7

안드로이드가 부팅하는데 약간의 시간이 소요될 수도 있습니다.



스텝 8

부팅이 완료되면, 그림과 같이 안드로이드의 잠금 화면이 나타납니다.

