Android

development environment

2010.03.12

Database Laboratory
Hanyang Univ.

System Requirements

OS

- Windows XP (32-bit) or Vista (32- or 64-bit)
- Mac OS X 10.4.8 or later (x86 only)
- Linux (tested on Linux Ubuntu Hardy Heron)

Supported Development Environments

- Eclipse 3.4 (Ganymede) or 3.5 (Galileo)
  - http://www.eclipse.org/downloads/
  - Eclipse JDT plugin (included in most Eclipse IDE packages)
- JDK 5 or JDK 6 (JRE alone is not sufficient)
- Android Development Tools plugin

Installing Eclipse

- http://www.eclipse.org/downloads/
- Recommended Eclipse IDE packages: Eclipse IDE for Java EE Developers, Eclipse IDE for Java Developers, Eclipse for RCP/Plug-in Developers, or Eclipse Classic (3.5.1+)

Install Java

Install Java

- Install Java to run Eclipse (In Ubuntu)

```
joo@ubuntu:~$ sudo apt-get install sun-java6-bin
[sudo] password for joo:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  linux-headers-2.6.31-14 linux-headers-2.6.31-14-generic
Use 'apt-get autoremove' to remove them.
The following extra packages will be installed:
  gsfonts-x11 java-common odbcinst1debian1 sun-java6-jre unixodbc
Suggested packages:
  equivs sun-java6-plugin ia32-sun-java6-plugin sun-java6-fonts
  ttf-kochi-gothic ttf-sazanami-gothic ttf-kochi-mincho ttf-sazanami-mincho
  ttf-arphic-uming libmyodbc odbc-postgresql libct1
The following NEW packages will be installed:
  gsfonts-x11 java-common odbcinst1debian1 sun-java6-bin sun-java6-jre
  unixodbc
```

- Install Java to run Eclipse (In Windows)
  - http://java.sun.com/javase/downloads/widget/jdk.jsp
  - JDK 5 or JDK 6 is recommended
  - To specify a JVM using configuration file, insert below contents to a text file named eclipse.ini in the same folder as eclipse.exe

    “–vm

    C:\Program Files\Java\jdk1.6.0_18\bin\javaw.exe”

Android SDK(1/2)

❖ Download and Install Android SDK

- Select a starter package
  - http://developer.android.com/sdk/index.html

Download the Android SDK

December 2009

The Android SDK has changed! If you've worked with the Android SDK before, you will notice several important differences:
show more

If you are new to the Android SDK, please read the Quick Start, below, for an overview of how to install and set up the SDK.

Platform	Package	Size	MD5 Checksum
Windows	android-sdk_r04-windows.zip	23069119 bytes	c48b407de852ba483869f17337e90997
Mac OS X (intel)	android-sdk_r04-mac_86.zip	19657927 bytes	b08512765aa9b0369bb9b8fecdf763e3
Linux (i386)	android-sdk_r04-linux_86.tgz	15984887 bytes	ef84b08fd9da84f4c4ae77564fe4eaee

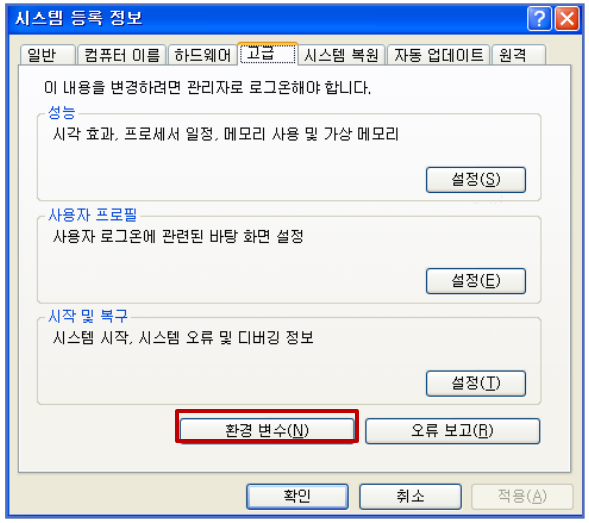
Android SDK(2/2)

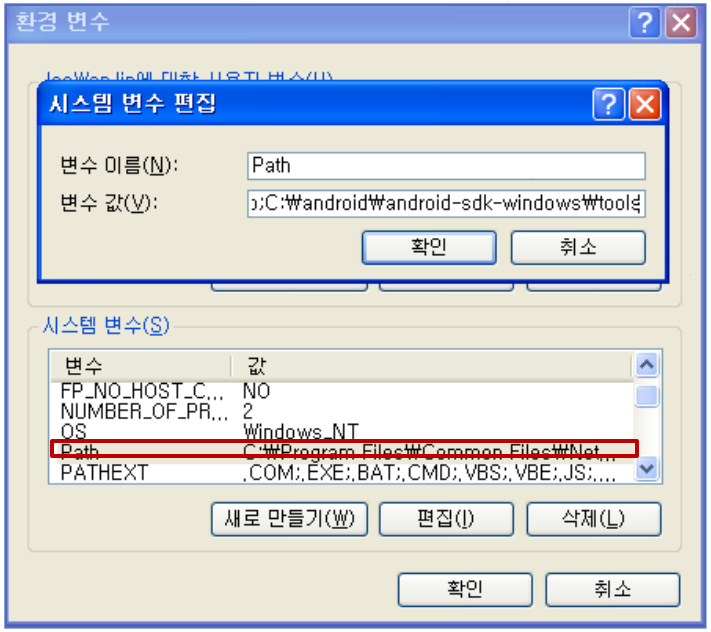
- To install the SDK, simply unpack the starter package to a safe location and then add the location to your PATH
  - In Ubuntu

    ```
    PATH=/android/android-sdk/tools:$PATH
    ```

  - In Windows

    [제어판] – [시스템]





Install the ADT(1/2)

❖ Install the ADT Plugin for Eclipse

- For developing in Eclipse, set up a remote update site and install the Android Development Tools(ADT) Plugin

  http://developer.android.com/sdk/eclipse-adt.html

❖ Installing the ADT Plugin

Eclipse 3.5 (Galileo)

1. Start Eclipse, then select **Help** > **Install New Software**.
2. In the Available Software dialog, click **Add...**.
3. In the Add Site dialog that appears, enter a name for the remote site (for example, "Android Plugin") in the "Name" field.

   In the "Location" field, enter this URL:

   ```
   https://dl-ssl.google.com/android/eclipse/
   ```

   Note: If you have trouble acquiring the plugin, you can try using "http" in the URL, instead of "https" (https is preferred for security reasons).

   Click **OK**.
4. Back in the Available Software view, you should now see "Developer Tools" added to the list. Select the checkbox next to Developer Tools, which will automatically select the nested tools Android DDMS and Android Development Tools. Click **Next**.
5. In the resulting Install Details dialog, the Android DDMS and Android Development Tools features are listed. Click **Next** to read and accept the license agreement and install any dependencies, then click **Finish**.
6. Restart Eclipse.

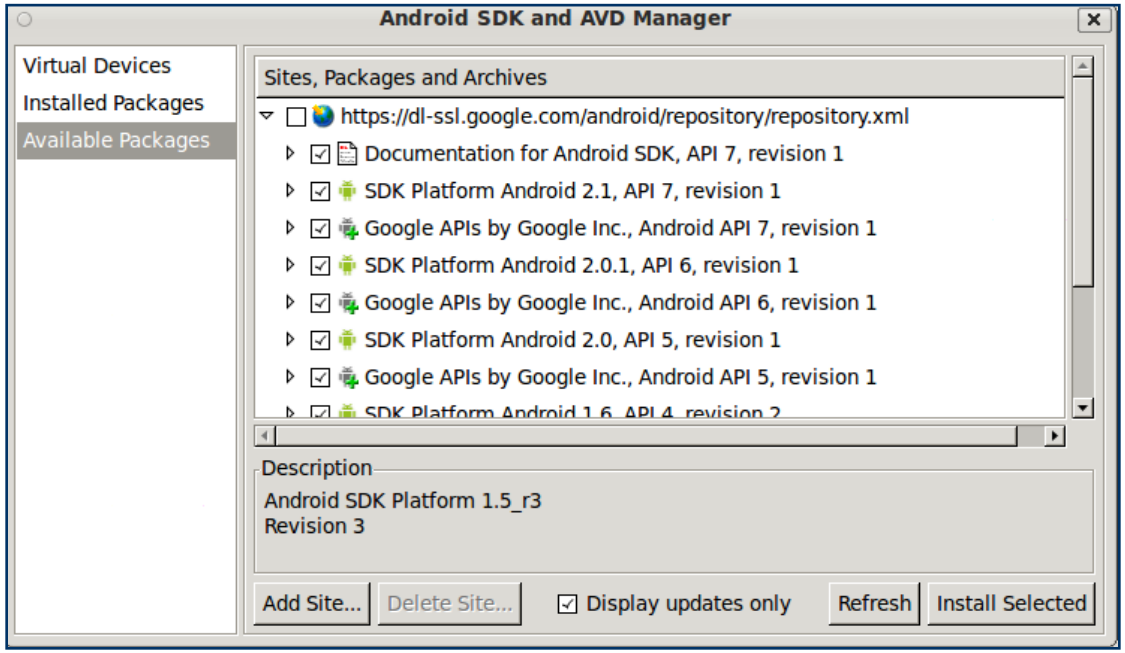
Install the ADT(2/2)

- **Modify Eclipse preferences to point to the Android SDK directory**
  - Select **Window** > **Preferences...** to open the Preferences panel (Mac OS X: **Eclipse** > **Preferences**)
  - Select **Android** from the left panel.
  - For the *SDK Location* in the main panel, click **Browse...** and locate your downloaded SDK directory.
  - Click **Apply**, then **OK**.

- **If you have encountered any Problems, then**
  - Reference to http://developer.android.com/sdk/eclipse-adt.html
  - Call teaching assistant

Adding SDK Components

- **Installing SDK Components**
  - you can select **Window >Android SDK and AVD Manager**.
- **you can quickly install set of development tools or documentation**
  1. Select **Available Components** in the left panel. This will reveal all components currently available for download.
  2. Select the components you'd like to install and click **Install Selected**.
  3. Verify and accept the components you want and click **Install Accepted**.



Get the latest Documentation

Development documentation

- If you develop while offline
  - Use the Android SDK and AVD Manager to download the latest documentation package.
  - The documentation covers all versions of the API and lets you filter out those versions that your application won't support
  - Once installed, the documentation is also available to you directly from the Eclipse IDE
- When you are online
  - you can always access the latest documentation at the Android Developers site.

Download other SDK components

- You can use the Android SDK and AVD Manager to download other SDK components (ex. SDK Add-on)
- An SDK add-on provides a development environment for an Android external library or a customized Android system image
- For example, the Google APIs Add-On lets you develop an application that takes advantage of the Google Maps external library.

Get started with an application project

❖ Application project

- Once you've set up your SDK, the next step is to start a new application project or move existing applications into the new SDK
- you can use the Hello World tutorial to get started quickly

❖ "Welcome to Android World"

Hello World Tutorial(1/7)

Create an AVD

- Before you can launch the emulator, you must create an Android Virtual Device (AVD)
- An AVD defines the system image and device settings used by the emulator.

In Linux

- To create an AVD, use the "android" tool provided in the Android SDK

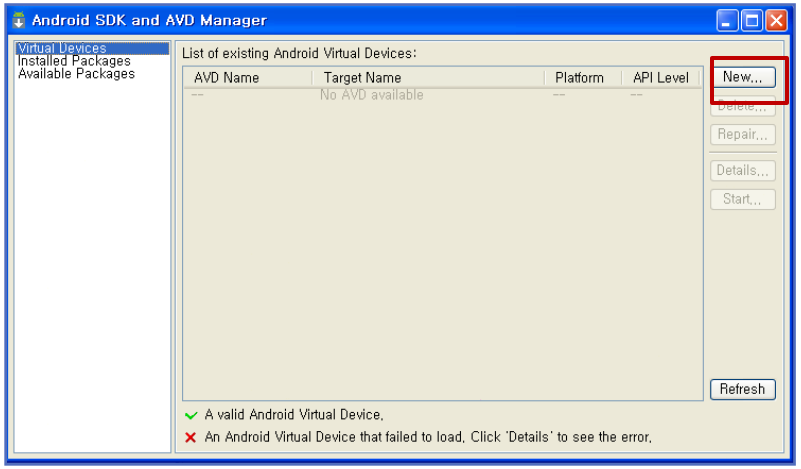
```
joo@ubuntu:~$ android create avd --name my_avd --target 7
```

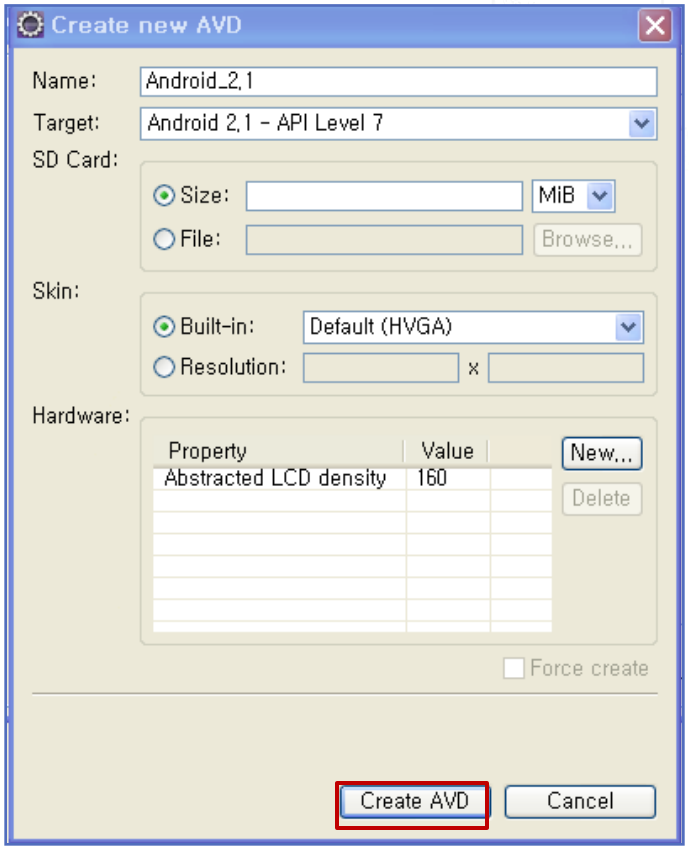
- The tool now asks if you would like to create a custom hardware profile. For the time being, press Return to skip it ("no" is the default response)
- In the above command, the --target option is required and specifies the deployment target to run on the emulator
- The --name option is also required and defines the name for the new AVD

Hello World Tutorial(2/7)

❖ Create an AVD(In Windows)

- Select **Windows** > **Android SDK and AVD Manager**



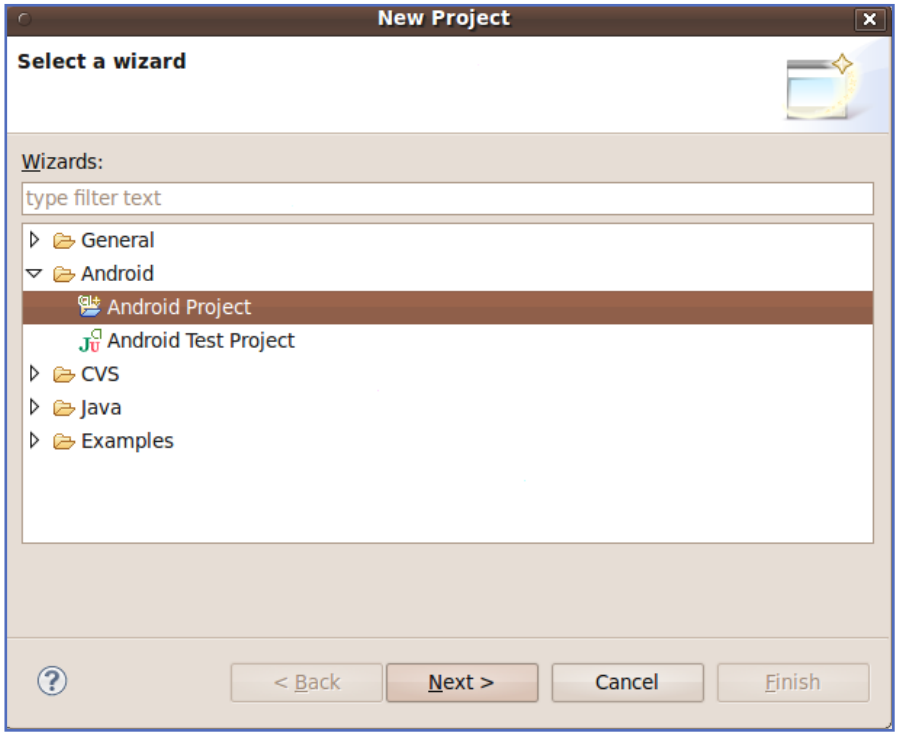


Hello World Tutorial(3/7)

❖ Create a New Android Project

- After you've created an AVD, the next step is to start a new Android project in Eclipse.

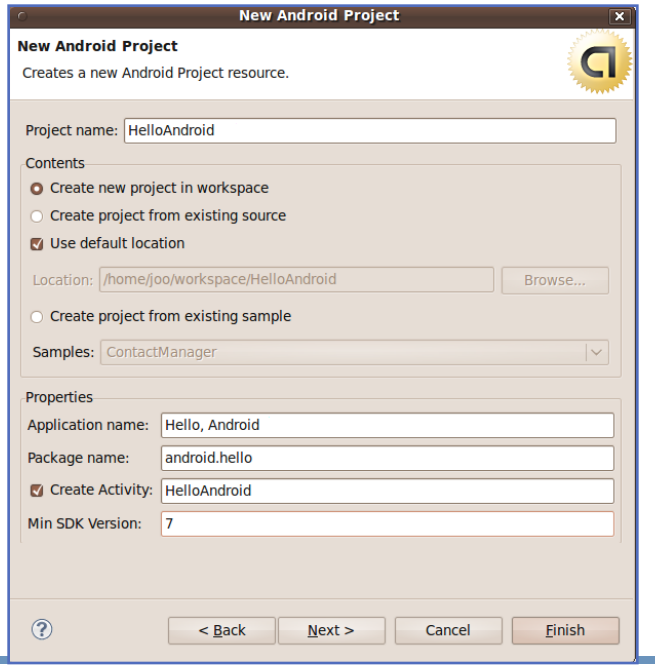
1. From Eclipse, select **File > New > Project**.
2. Select "Android Project" and click **Next**.



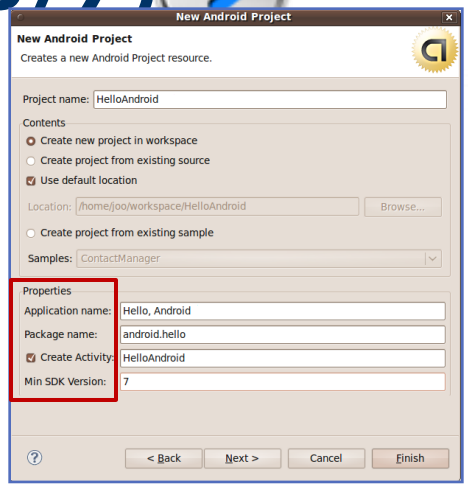
Hello World Tutorial(4/7)

❖ Create a New Android Project

3. Fill in the project details with the following values:
   - Project name: HelloAndroid
   - Application name: Hello, Android
   - Package name: com.example.helloandroid (or your own private namespace)
   - Create Activity: HelloAndroid
   - Min SDK Version: 2
4. Click **Finish**.



Hello World Tutorial(5/7)



❖ Description of each field

- Project Name
  - the name of the directory that will contain the project files
- Application Name
  - the name that will appear on the Android device
- Package Name
  - This is the package namespace that you want all your source code to reside under
- Create Activity
  - This is the name for the class stub that will be generated by the plugin
  - This will be a subclass of Android's Activity class
- Min SDK Version
  - This value specifies the minimum API Level required by your application

Hello World Tutorial(6/7)

❖ Android Project is now ready

- Open the HelloAndroid.java file, then type like this:

*HelloAndroid.java

```
package android.hello;

import android.app.Activity;
import android.os.Bundle;
import android.widget.TextView;

public class HelloAndroid extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        TextView tv = new TextView(this);
        tv.setText("Hello, Android!");
        setContentView(tv);
    }
}
```

Hello World Tutorial(7/7)

❖ Run the Application

1. Select **Run > Run**.
2. Select "Android Application".

- The Eclipse ADT will automatically create a new run configuration for your project and the Android Emulator will automatically launch.
- You will need some patience, while the emulator is booted up