Android Installation Guide (2)

© Frank Mueller & Seokyong Hong (TA)

North Carolina State University

Center for Efficient, Secure and Reliable Computing

Outline


Install JDK
Install Android SDK
Install Eclipse
Install Eclipse Plug-ins for Android
Sample Project: Hello, Android

Installation - JDK

- JDK is required to run Eclipse and develop Android applications.

- Installation Process
  - ✓ Download the latest version of JDK from

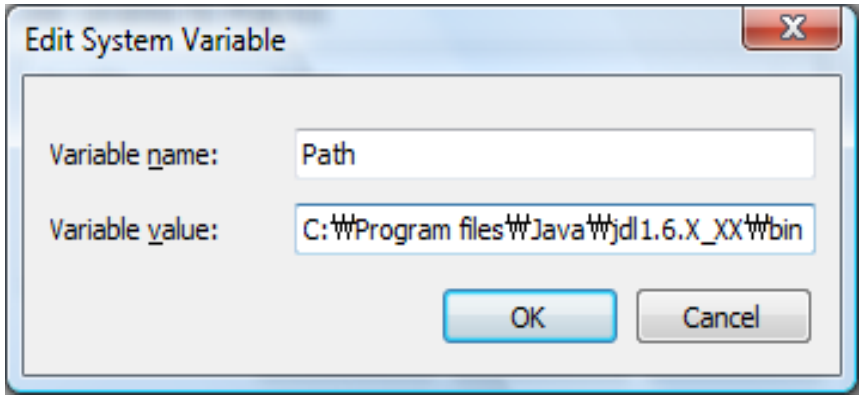
    http://java.sun.com

  - ✓ Install the downloaded JDK
  - ✓ Add the path of the JDK's '*bin*' directory to the system
    - ❖ For Linux, add it to *~/.bash_profile*:

```
# vi ~/.bash_profile
PATH=$PATH:/usr/java/jdk1.6.X_XX/bin

# source ~/.bash_profile
```

Installation - JDK (Cont.)

- Installation Process (Cont.)
  - For Windows, add it to Environment Varaibles:



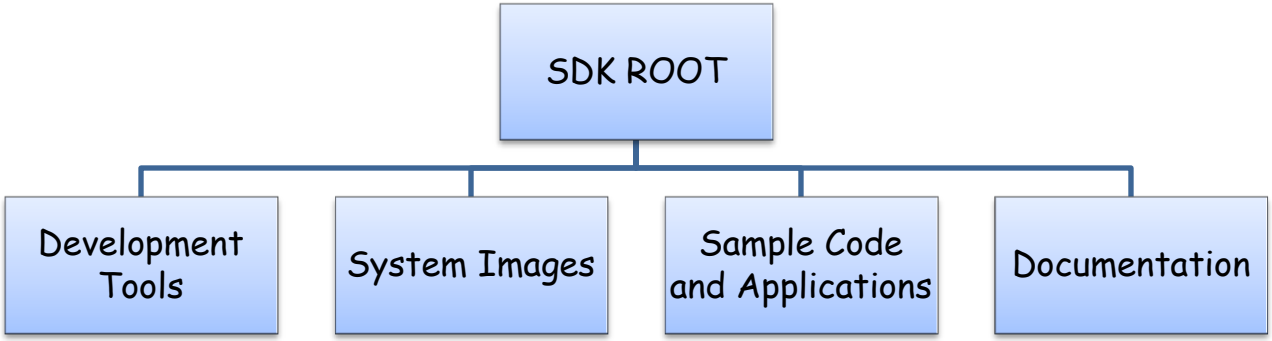
Installation - JDK (Cont.)

- Test
  - ✓ If you have successfully installed JDK, you should be able to see following messages in your command prompt window or terminal:

```
# javac -version
javac 1.6.X_XX
```

Installation - Android SDK

- Android SDK Source Tree



Directory	Description
Development Tools	A variety of tools for developing and debugging application code and designing an application UI
System Images	Android 1.1 system images
Sample Code and Applications	A variety of tutorials and samples
Documentation	A full set of local documentation

Installation - Android SDK (Cont)

- Development Tools
  - ✓ Two most important tools
    - ➢ The Android emulator
    - ➢ The Android development tools plug-in for Eclipse

Installation - Android SDK (Cont)

- Development Tools (Cont)
  - ✓ Hierarchy Viewer
    - ➢ A tool that allows developers to debug and optimize user interface
    - ➢ Providing a visual representation of layout's hierarchy of Views and a magnified inspector of the current display with a pixel grid
  - ✓ Android Debug Bridge (adb)
    - ➢ A tool that enables:
      - installation of Android Package Files (.apk) on an emulator ot device from a command line
      - linking a standard debugger to applications code running on an emulator or device

Installation - Android SDK (Cont)

- Development Tools (Cont)
  - ✓ Draw 9-patch
    - ➢ A WYSIWYG graphic editor that offers a handy way to create NinePatch images
    - ➢ NinePatch image
      - A strechable bitmap image which Android will automatically resize to accomodate the contents of the View where it is placed as the background
      - A standard PNG image that includes an extra 1-pixel-wide border
  - ✓ Android Asset Packaging Tool (aapt)
    - ➢ A tool that lets developers create Android Package Files (.apk) containing the binaries and resources

Installation - Android SDK (Cont)

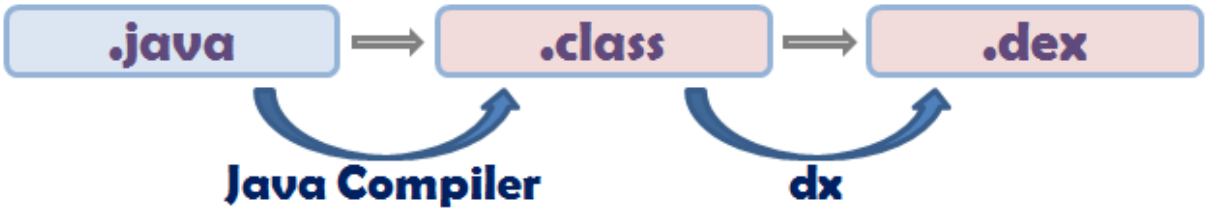
- Development Tools (Cont)
  - ✓ Dalvik Debug Monitor Service (ddms)
    - ➢ A tool that lets developers manage processes on an emulator or device and assists in debugging
    - ➢ Functions
      - Killing processes
      - Selecting a specific process to debug
      - Generating trace data
      - Viewing heap and thread information
      - Taking screenshots of the emulator or device

Installation - Android SDK (Cont)

- Development Tools (Cont)
  - ✓ Android Interface Description Language (aidl)
    - ➢ A language that lets developers generate code for an inter-process interface, such as what a service might use
  - ✓ sqlite3
    - ➢ A tool that lets you access the SQLite data files created and used by Android applications
  - ✓ Traceview
    - ➢ A tool that produces graphical analysis views of trace log data that developers can generate from Android application

Installation - Android SDK (Cont)

- Development Tools (Cont)
  - ✓ mksdcard
    - ➢ A tool that helps developers create a disk image that can be used with the emulator, to simulate the presence of an external storage card (such as an SD card)
  - ✓ dx
    - ➢ A tool that rewrites .class bytecode into Android bytecode (.dex)



Installation - Android SDK (Cont)

- Development Tools (Cont)
  - ✓ UI/Application Exerciser Monkey
    - ➢ A program that runs on emulator or device with generating pseudo-random streams of user events (clicks, touches, or gestures), as well as a number of system-level events
    - ➢ Can be used to stress-test applications being developed
  - ✓ activitycreator
    - ➢ A script that generates And build files to compile Android applications

More detail information is available with the following link:

http://developer.android.com/guide/developing/tools/index.html

Installation - Android SDK (Cont)

- Installation Process
  - ✓ Download the Android SDK from:

    http://developer.android.com/sdk/

  - ✓ Unpack the compressed file into a location you prefer.
  - ✓ Add the path to the SDK *tools* directory "<SDK ROOT>/tools" to ~/.bash_profile (Linux) or Environment Variables (Windows)

Installation - Android SDK (Cont)

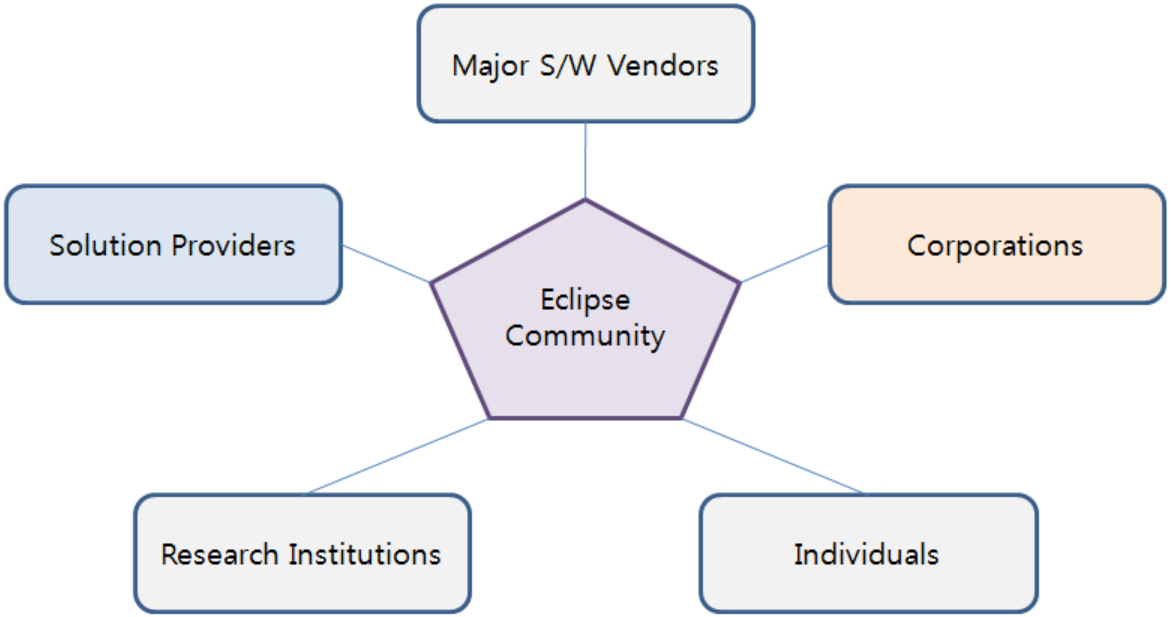
- Test
  - ✓ If you have successfully installed the Android SDK, you should be able to see following messages in your command prompt window or terminal:

 # dw --version

dx version 1.1

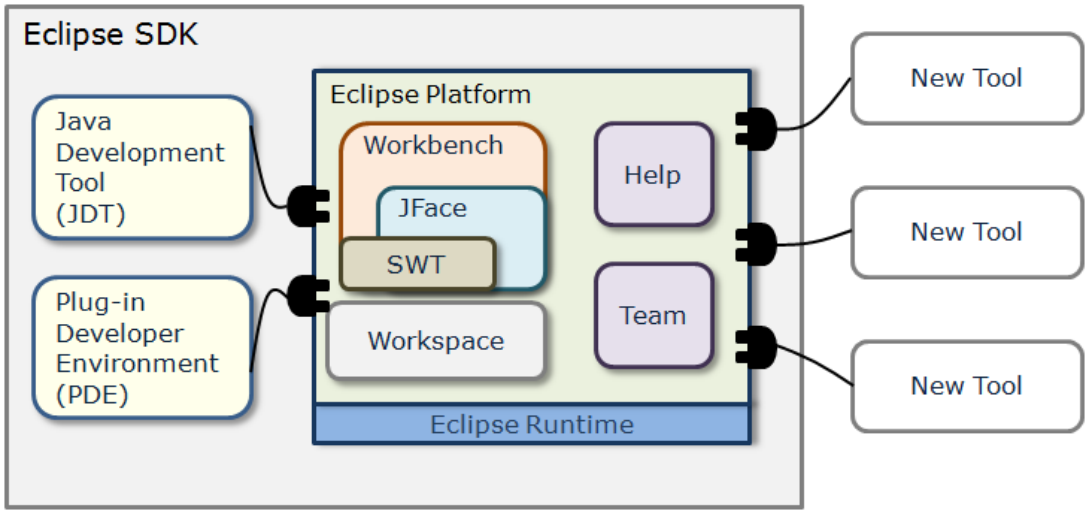
Installation - Eclipse

- Eclipse
  - ✓ An Integrated Development Toolkit licensed under Eclipse Public License
  - ✓ An open source project developed by the Eclipse community:



Installation - Eclipse (Cont)

- Eclipse Platform



- ✓ The Eclipse platform defines an open architecture to allow plug-in developers to add a variety of function to the basic tooling platform.

Installation - Eclipse (Cont)

- The Eclipse community provides various plug-ins:

Plug-in	Function
JDT	provides the capability to create, edit, navigate, build, and debug projects that use Java as a programming language
CDT	provides the capability to create, edit, navigate, build, and debug projects that use C and/or C++ as a programming language
UML2	provides the capability to create UML models
…	…

Installation - Eclipse (Cont)

- Installation Process
  - ✓ Download Eclipse from

    http://www.eclipse.org/downloads/

    Currently, a Java or RCP version of Eclipse 3.3 (Europa) or Eclipse 3.4 (Ganymede) are recommended.
  - ✓ Unpack the downloaded file into a path you prefer. For example,

    /home/user1/eclipse (Linux)

    or

    C:\Program Files\eclipse (Windows)
  - ✓ Add the path to *.bash_profile* (Linux) or *Environment Variables* (Windows)

Installation - Eclipse (Cont)

- Test
  - ✓ If you have successfully installed the Android SDK, you should be able to see following application:

 # eclipse

Installation - Android Development Tools

- ADT (Android Development Tools) provides a powerful, integrated enviornment to build Android applications.
- Installation Process (*based on Eclipse 3.4 Ganymede*)
  - ✓ Start Eclipse, then select **Help** > **Software Updates...**.
  - ✓ In the dialog that appears, click the **Available Software** tab.
  - ✓ Click **Add Site...** and enter following location:

http://dl-ssl.google.com/android/eclipse

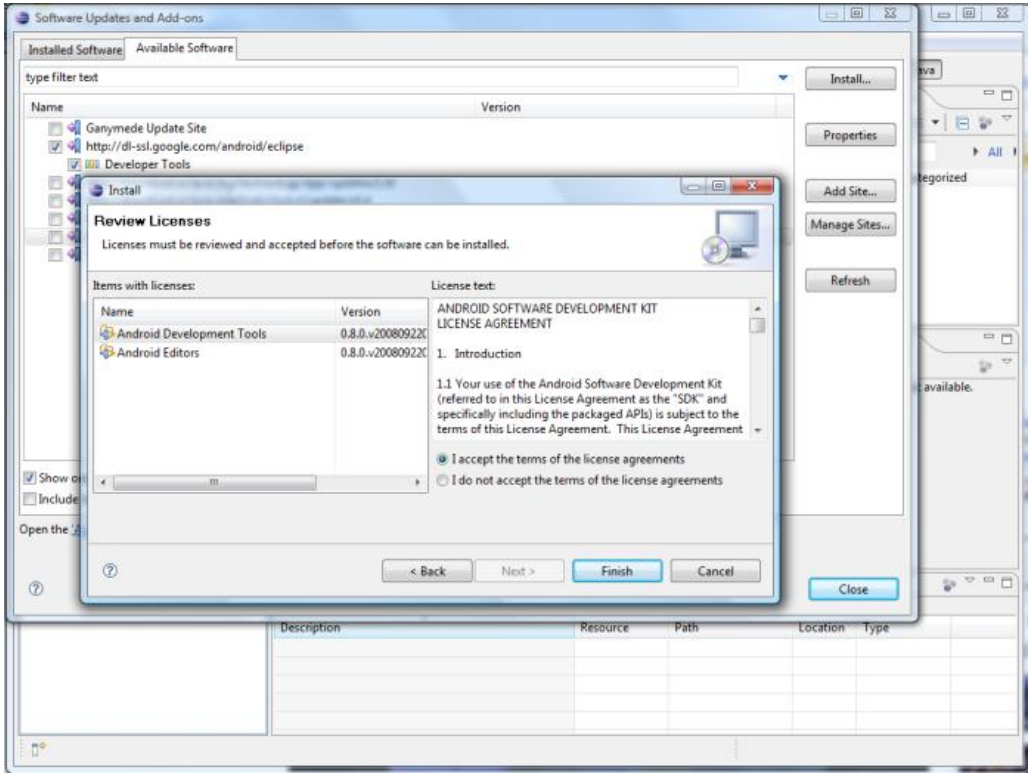
or

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

  - ✓ Back in the **Available Software** view, select the checkbox next to **Developer Tools** and click **Install...**.

Installation - ADT (Cont)

- Installation Process (Cont)
  - ✓ On the subsequent install window, check both "Android Developer Tools" and "Android Editors".



Installation - ADT (Cont)

- Installation Process (Cont)
  - ✓ Accept the license agreement and click "**Finish**".
  - ✓ Restart Eclipse.
  - ✓ Modify Eclipse preferences to point to the Android SDK directory:
    - ➢ Select **Window** > **Preferences...** to open the Preferences panel.
    - ➢ Select **Android** from the left panel.
    - ➢ For the SDK Location in the main panel, click **Browse...** and locate the SDK directory.
    - ➢ Click **Apply** and then **OK**.

Sample Project - "Hello, Android"

- Create "*HelloAndroid*" project.
  ① Select the **File** > **New** > **Project** menu item and select "Android Project" and click **Next**.

  ② Fill out the project details:

Field	Value	Meaning
Project Name	HelloAndroid	The name of the directory or folder you want to contain the project
Package Name	com.example.hello	The package namespace (following the same rules as for packages in the Java language) that all source code will reside under
Activity Name	HelloAndroid	The name for the class stub that will be generated by the plug-in
Application Name	Hello, Android	The human-readable title for application

Sample Project - "Hello, Android" (Cont)

- Take a look at *HelloAndroid.java* (*HelloAndroid* > *src* > *com.android.hello*) and modify the code like:

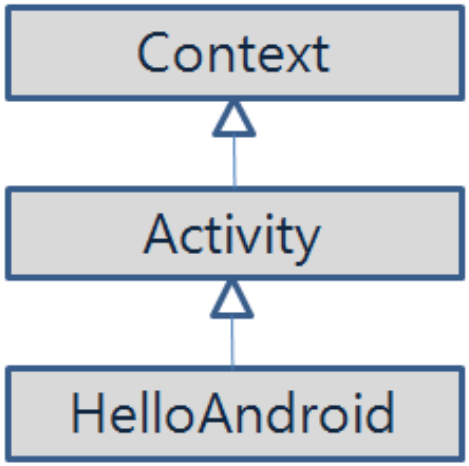
```
package com.android.hello;
import android.app.Activity;
import android.os.Bundle;
import android.widget.TextView;

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);
    }
}
```

Sample Project - "Hello, Android" (Cont)

- Source Description
  - ✓ Creating a **TextView** object

TextView tv = new TextView(this);

- A **View** (a drawable object) subclass that handles text
- The argument to TextView's constructor: an Android **Context** instance.



Sample Project - "Hello, Android" (Cont)

- Source Description (Cont)
  - ✓ Tell the TextView object to display a String:

```
tv.setText("Hello, Android");
```

  - ✓ Connect the **TextView** object with on-screen display:

```
setContentView(tv);
```

Sample Project - "Hello, Android" (Cont)

- Test
  1. Select the **Run** > **Run Configuration** menu entry.
  2. Highlight the "**Android Application**" entry, and then click the icons in the top left corner (the one depicting a sheet of paper with a plus sign in the corner) or simply double-click the highlighted entry.
  3. Change the name "*New_configuration*" to something like "*Hello Android*".
  4. Pick the *HelloAndroid* project by clicking the "**Browse**" button.
  5. Click "**Apply**" and then "**Run**".