

Android Development

Chad Maughan
January 28, 2012

Introduction

- Chad Maughan
- <http://chadmaughan.com>
- @chadmaughan (blue birdie)

Goals

- Familiarize yourself with the Android stack and development environment
- Provide sample code for now and later
- Have fun!

Books

- **Android in Action, Third Edition**
 - W. Frank Ableson; Robi Sen; Chris King; C. Enrique Ortiz
- **Pro Android 3**
 - Satya Komatineni; Dave MacLean; Sayed Y. Hashimi
- **Learning Android**
 - Marko Gargenta
- **Professional Android 2 Application Development**
 - Reto Meier

Other Resources

- <http://developer.android.com>
- <http://vkedco.blogspot.com>
- <http://code.chadmaughan.com>
- <http://github.com/chadmaughan>

System requirements

- <http://developer.android.com/sdk/requirements.html>
 - Windows XP/Vista/7, Mac OS X >= 10.4.8, Linux
 - JDK and Java Runtime Environment (JRE) >= 5
 - Eclipse >= 3.4, including the JDT and Web Tools Platform, which are included in the Eclipse installation package
 - ADT plug-in for Eclipse

JDK

- Check version with

```
$ java -version
```

```
java version "1.6.0_29"
```

```
Java(TM) SE Runtime Environment (build  
1.6.0_29-b11-402-11M3527)
```

```
Java HotSpot(TM) 64-Bit Server VM (build  
20.4-b02-402, mixed mode)
```

- Download:

- <http://www.oracle.com/technetwork/java/javase/downloads/index.html>

Git (optional)

- <http://git-scm.com/download>
- Other installation
 - Debian based Linux
 - apt-get install git
 - OS X (with Homebrew)
 - brew install git
- Windows Help
 - <http://help.github.com/win-set-up-git/>
- Tutorial
 - <http://rogerdudler.github.com/git-guide/>

Setup & Installation

- Following guide on Android Developer site
 - <http://developer.android.com/sdk/installing.html>
- Steps:
 1. Eclipse
 2. Android SDK (Starter Package)
 3. ADT Plugin
 4. Add platforms and components

1 - Eclipse

- Current version: Indigo 3.7.1
- Download:
 - <http://eclipse.org/downloads/>
- “Eclipse Classic” recommended version
- Recommended Plugins (Eclipse Marketplace)
 - EGit
 - Maven Integration for Eclipse
 - Spring Tools (STS)

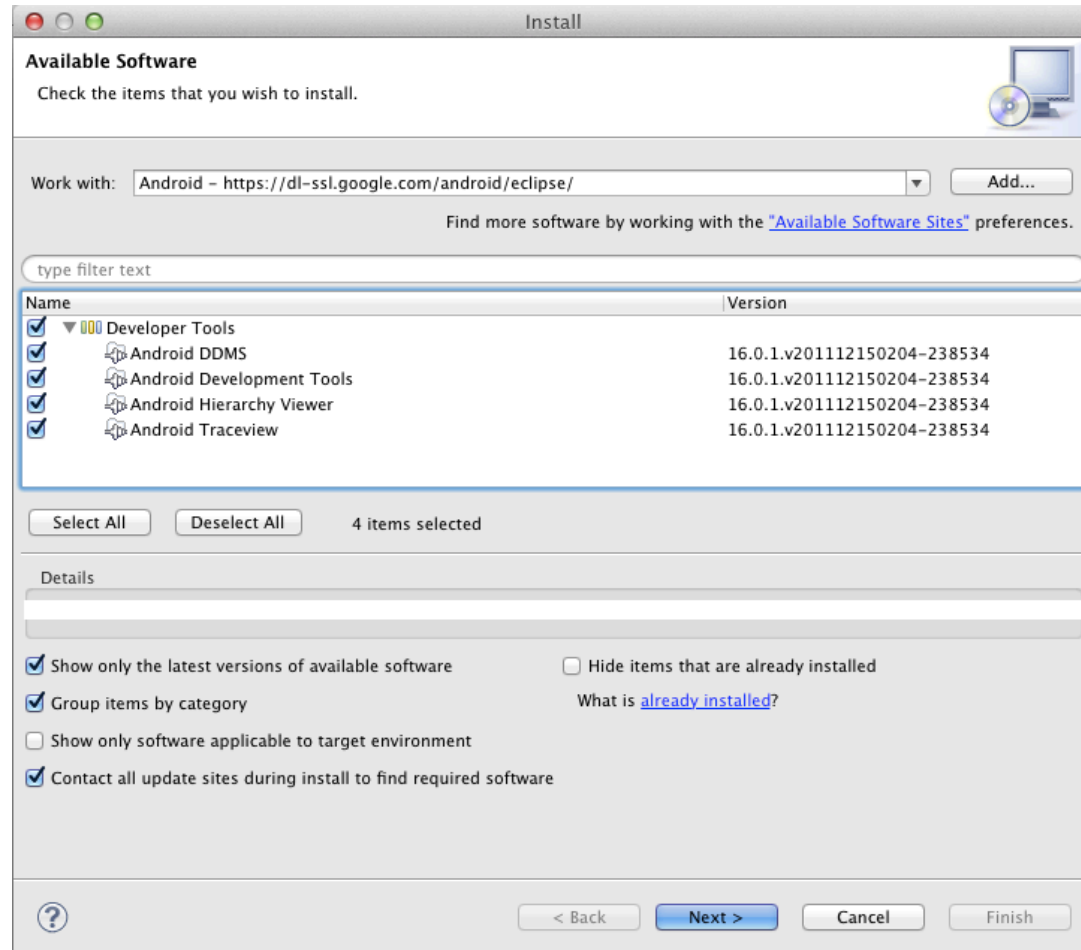
2 - Android SDK

- Download at
 - <http://developer.android.com/sdk/>
- Windows
 - Recommended installer checks for Java and installs if necessary
 - Note installation location (needed later)

3 - Eclipse ADT Plugin

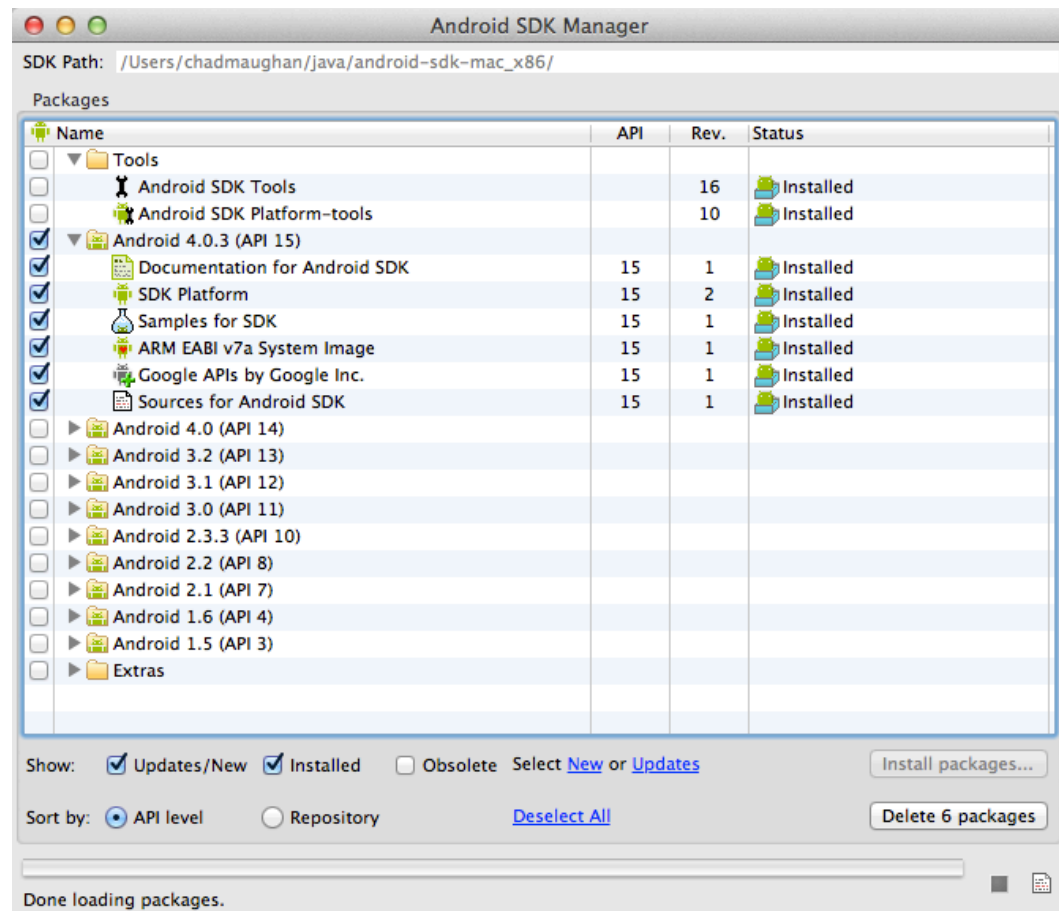
- Instructions
 - <http://developer.android.com/sdk/eclipse-adt.html#installing>
- **Help > Install New Software**
 - Name: Android Plugin (or whatever)
 - Site: <https://dl-ssl.google.com/android/eclipse/>
- Choose top “Developer Tools” plugin (so that all sub options are selected)

3 - Eclipse ADT Plugin



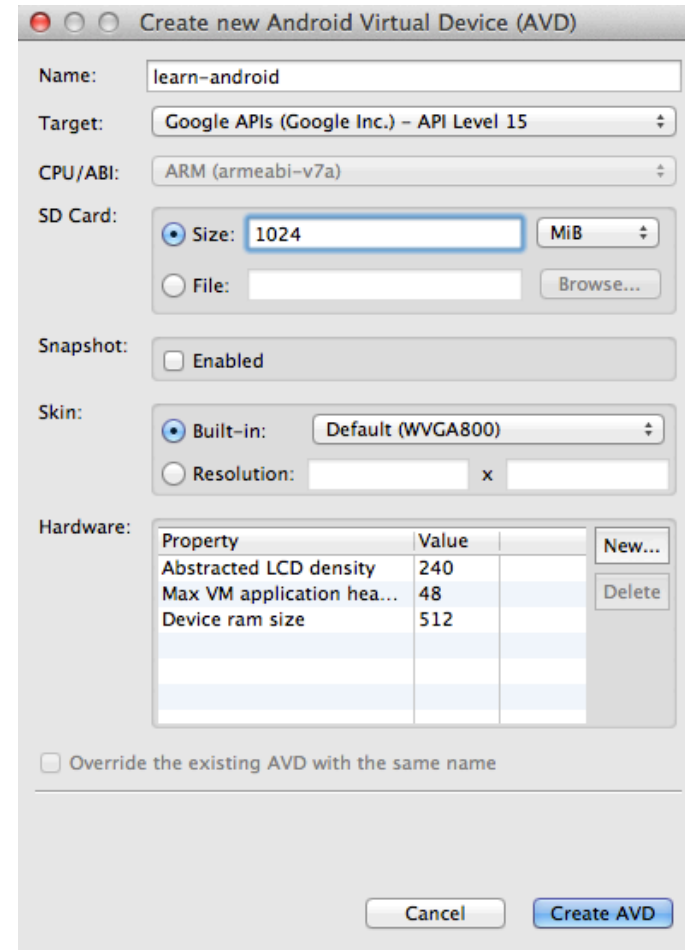
4 – Install Platform

- Versions, tools, add-ons separated into installable components



Create Virtual Device

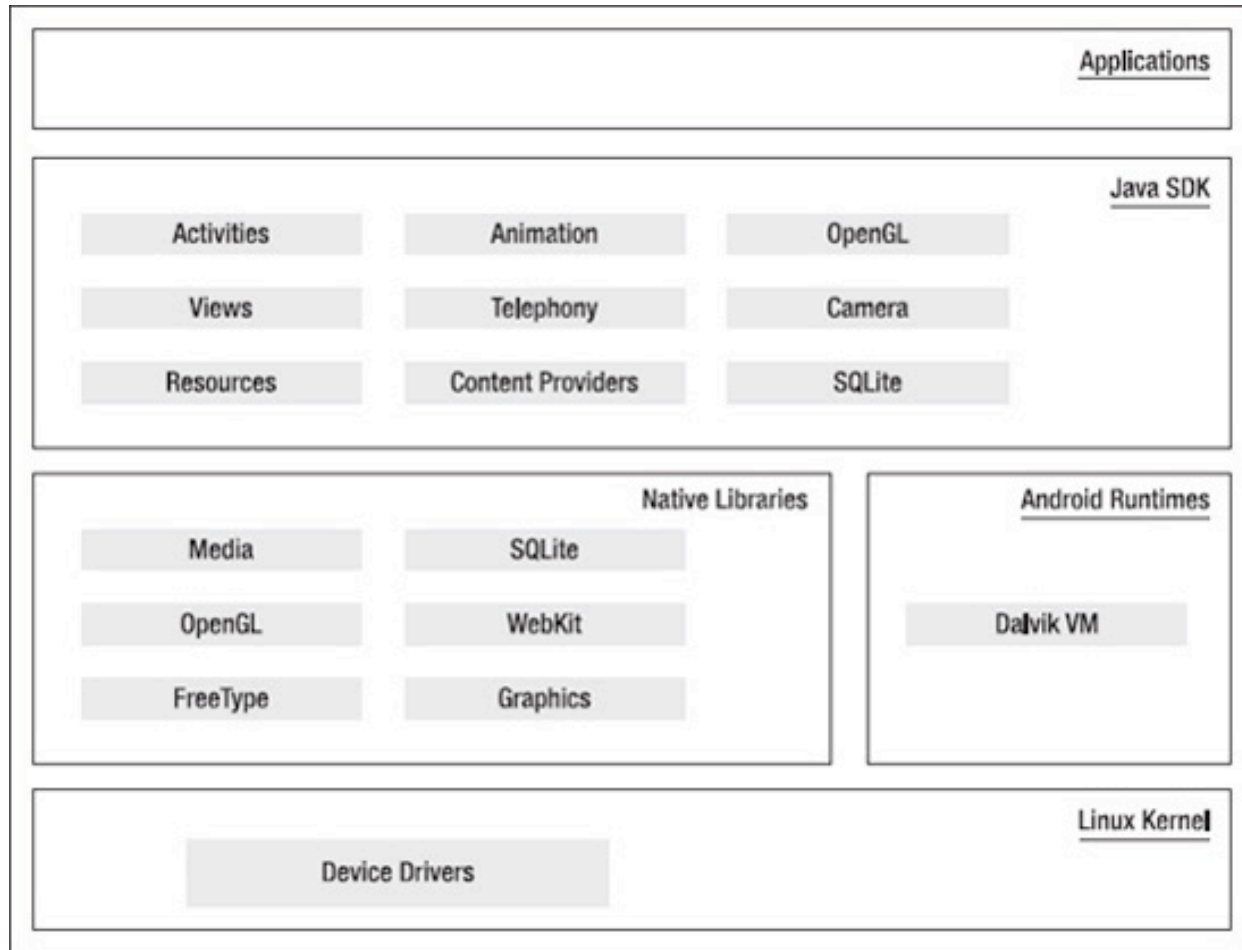
- Call the device whatever you'd like
- Choose the target of Google APIs Level 15
 - You can also choose Android 4.0.3 API Level 15 (doesn't include maps, etc)
- SD card required for external storage, etc



Android Platform

Android is a software environment built for mobile devices. It's not a hardware platform. Android includes a Linux kernel-based OS, a rich UI, end-user applications, code libraries, application frameworks, multimedia support, telephone functionality and more.

Android Stack



Source: Pro Android 3

Key Components

- View
- Activity
- Service
- BroadcastReceiver
- ContentProvider
- AndroidManifest.xml

View

- All UI controls, including the layout classes are derived from class *View*
- A *View* is assigned to an activity
- A widget is a view object for interaction
 - Comprehensive list at android.widget
- Can be declared **programmatically** (in code) or **declaratively** (in XML)
- A *ViewGroup* is a special view that can have children

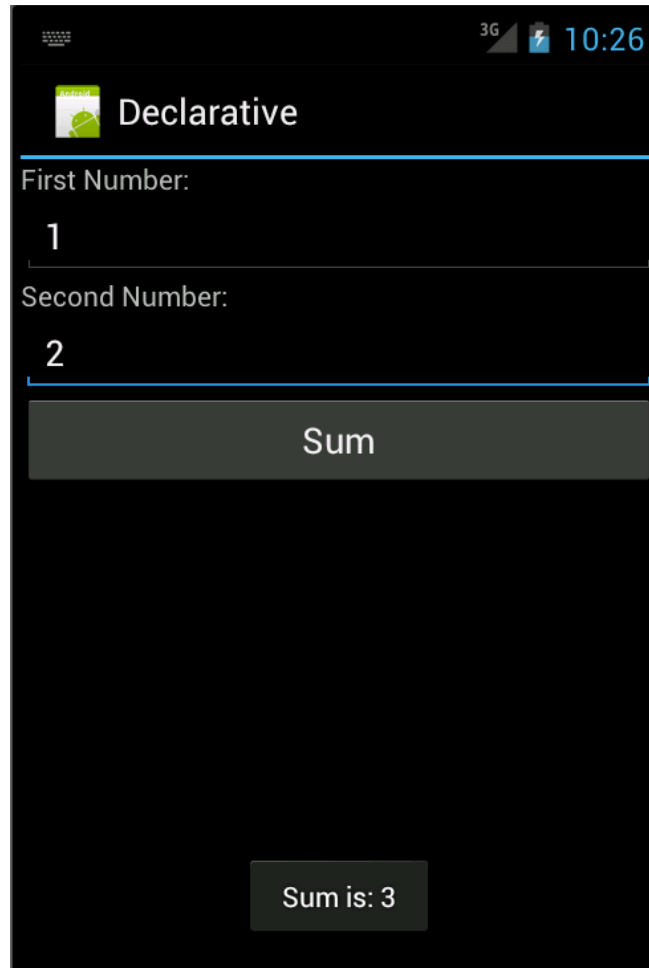
Layouts

- All layouts are subclasses of *ViewGroup*
- XML layouts are compiled into a View resource in */res/layout* directory
 - `R.layout.main_layout`
 - `setContentView(R.layout.main_layout);`
- On problems
 - Clean project
 - Right click, then Android Tools -> Fix Project Properties
- Tutorial: [Hello, Views](#)

Example Layout XML

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/
android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
    <TextView android:id="@+id/text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello, I am a TextView" />
    <Button android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello, I am a Button" />
</LinearLayout>
```

Lab 1



The screenshot shows an Android application interface with a black background. At the top, the status bar displays '3G', a battery icon, and the time '10:26'. Below the status bar, the app's title bar shows a green Android icon and the text 'Declarative'. The main content area contains two text input fields. The first field is labeled 'First Number:' and contains the value '1'. The second field is labeled 'Second Number:' and contains the value '2'. Below these fields is a large, dark gray button with the text 'Sum'. At the bottom of the screen, there is a smaller, dark gray button with the text 'Sum is: 3'.

3G 10:26

Declarative

First Number:

1

Second Number:

2

Sum

Sum is: 3

Activities

- Activity = Screen
- Allows interaction (via interface)
- Application usually consists of multiple activities
- Two important methods to implement
 - onCreate(Bundle)
 - onPause()
- Developer Guide
 - <http://developer.android.com/guide/topics/fundamentals/activities.html>

Application Manifest

android:name only required attribute

```
<manifest ... >  
  <application ... >  
    <activity android:name=".MainActivity" />  
    ...  
  </application ... >  
  ...  
</manifest >
```


Intents

Used to start another activity

```
Intent intent = new  
Intent(Intent.ACTION_SEND);  
intent.putExtra(  
    Intent.EXTRA_EMAIL, recipients);  
startActivity(intent);
```

Start for a result

To receive a result from a started activity

```
Intent intent = new Intent(  
    Intent.ACTION_PICK, Contacts.CONTENT_URI);  
  
startActivityForResult(  
    Intent, PICK_CONTACT_REQUEST);
```

To receive a result from a started activity

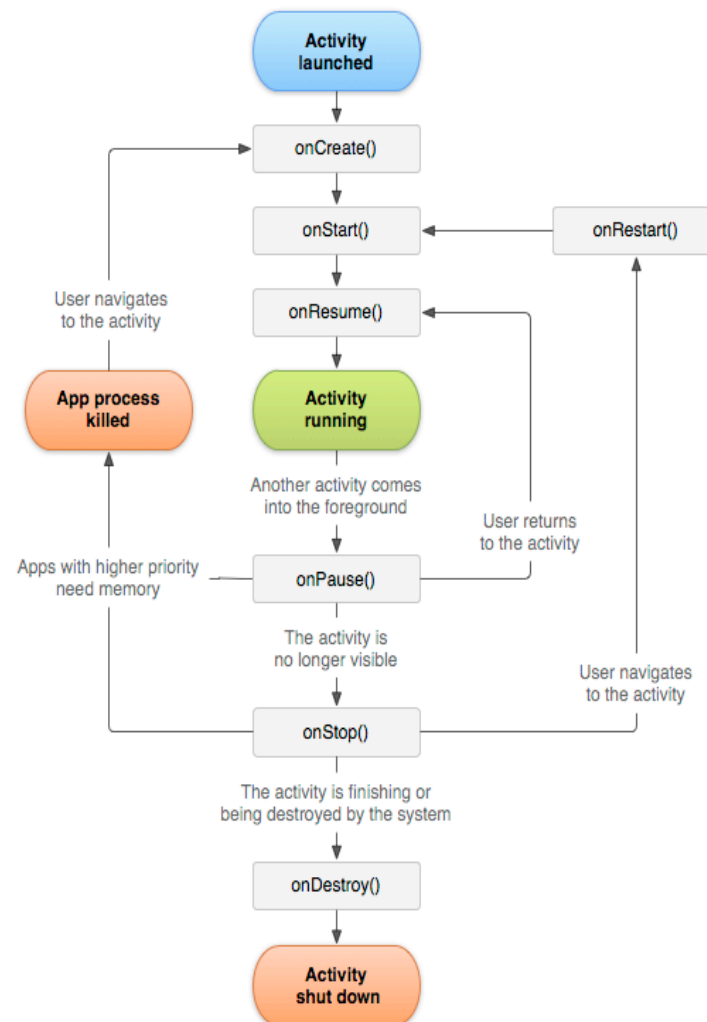
```
onActivityResult(  
    int requestCode, int resultCode, Intent data)
```

Intent Filter

- Allows other applications to activate your Activity
- Three “tests”
 - Action
 - Category
 - Data

```
<activity android:name=".MainActivity">  
  <intent-filter>  
    <action android:name="android.intent.action.MAIN" />  
    <category android:name="android.intent.category.LAUNCHER" />  
    <data android:mimeType="audio/mpeg" android:scheme="http" />  
  </intent-filter>  
</activity>
```

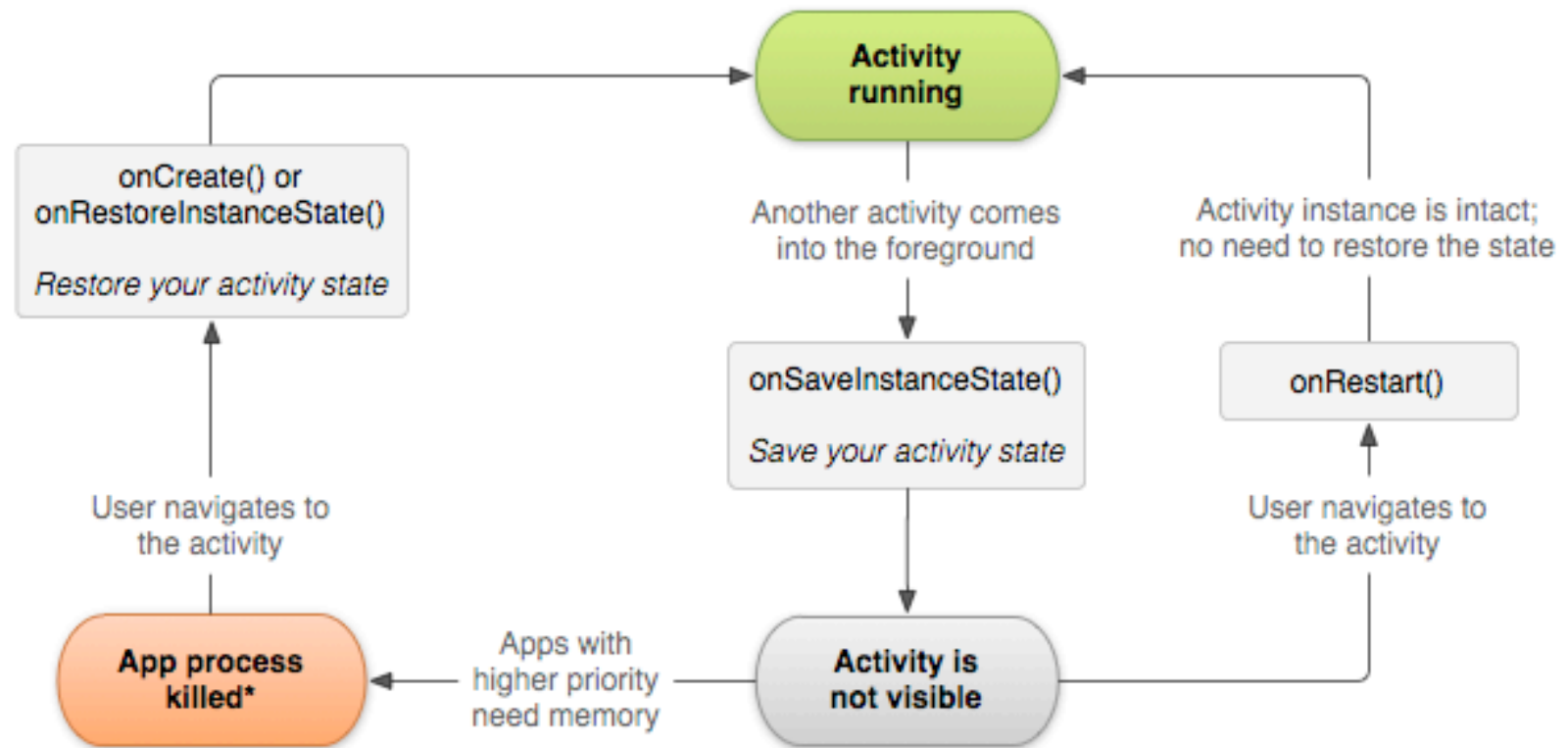
Activity Lifecycle



Activity States

- Essentially three states
 - Resumed (running)
 - Paused
 - Another activity has focus, but not obscured
 - Stopped
 - Totally obscured, in “background”
 - Can be “finish()-ed” or killed

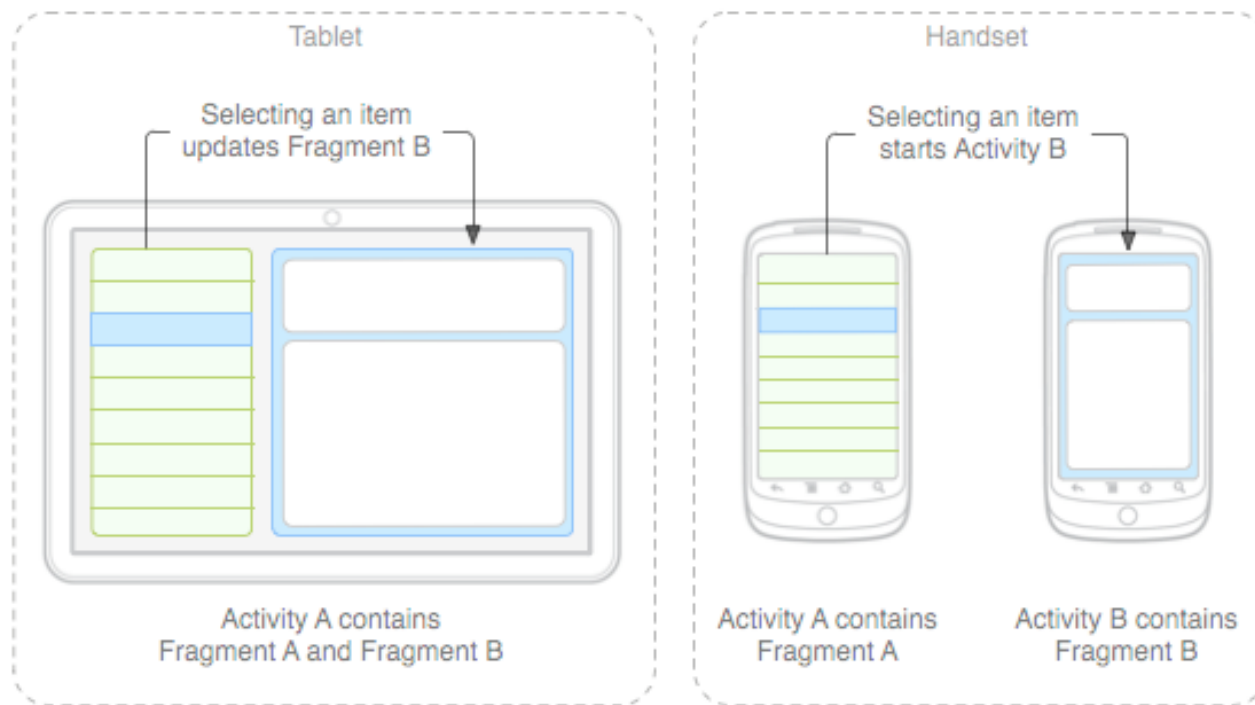
Saving Activity State



*Activity instance is destroyed, but the state from `onSaveInstanceState()` is saved

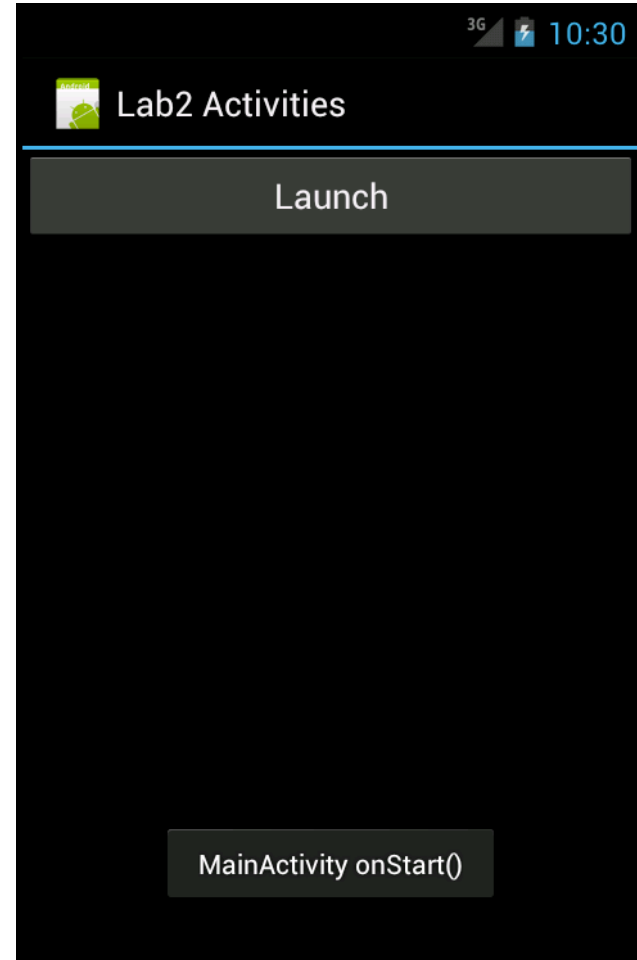
Fragments

- Introduced in Android 3.0
- More flexible UI for larger screens (tablets)



Lab 2

Activity Lifecycle,
Intents and passing
data between activities



Menus

- Three types of application menus
 - Options
 - Context
 - Submenu

Options

- Appear when you touch the menu button
- Android 3.0 (API 11) and later allow ActionBar



Context Menu

- Floating list of menu items when the user touches and **holds** a view that is registered to provide a context menu

Submenu

- Floating list displayed when Options menu has a nested menu

Creating a Menu

- Define in XML

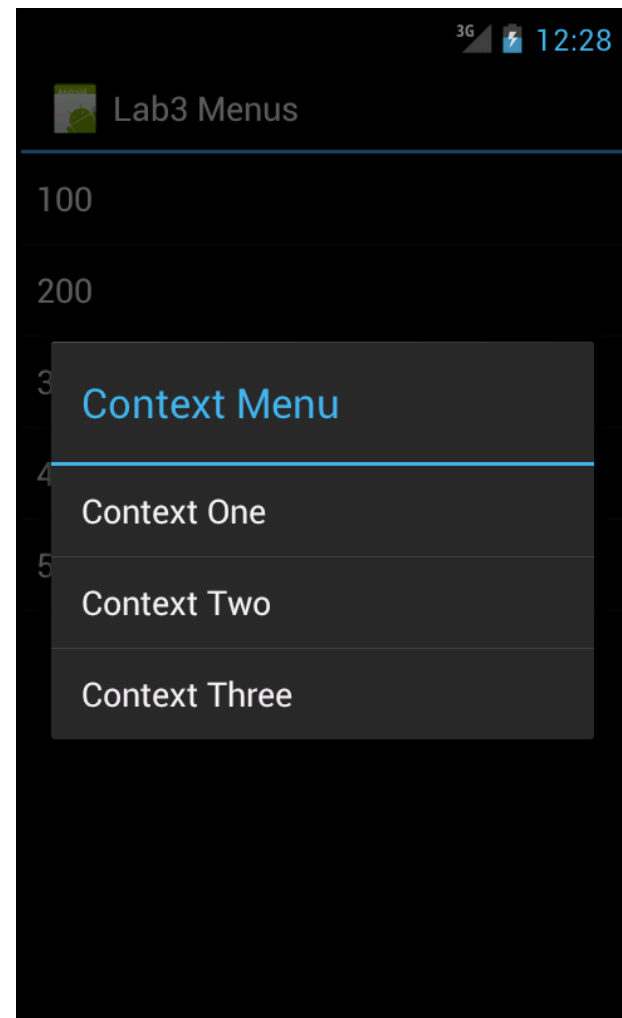
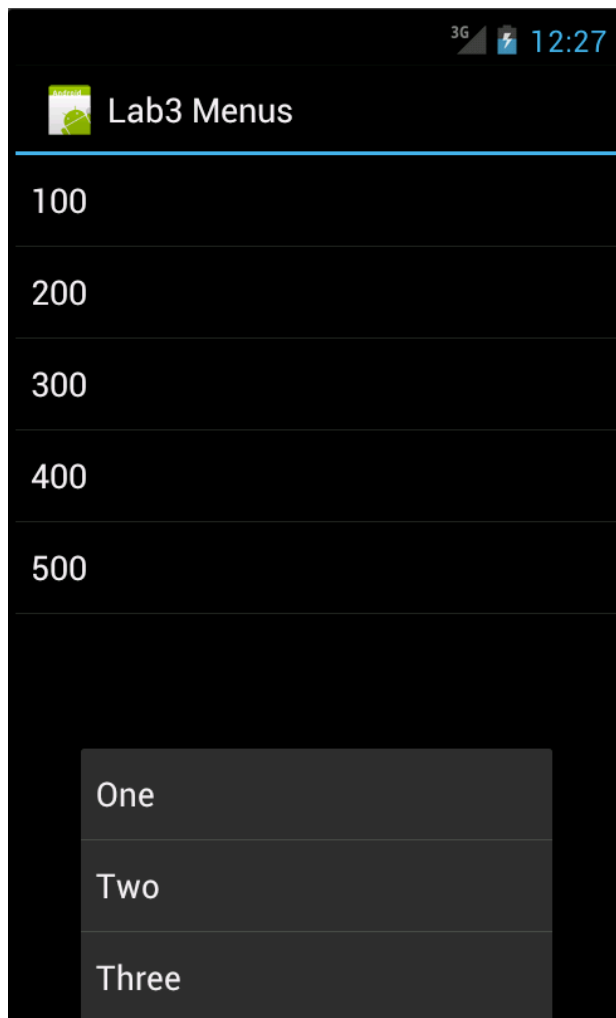
```
<menu xmlns:android="
    http://schemas.android.com/apk/res/android">
    <item android:id="@+id/new_game"
        android:icon="@drawable/ic_new_game"
        android:title="@string/new_game" />
    <item android:id="@+id/help"
        android:icon="@drawable/ic_help"
        android:title="@string/help" />
</menu>
```

Creating a Menu (cont.)

- Inflate in code

```
@Override
public boolean onCreateOptionsMenu(Menu menu){
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.game_menu, menu);
    return true;
}
```

Lab 3



Data Storage

- Android provides multiple storage options:
 - Shared Preferences
 - Internal Storage
 - External Storage
 - SQLite Database
 - Network Connection

Shared Preferences

- Save and retrieve persistent key-value pairs of primitive data types
- Data will persist across user sessions (even if your application is killed)
- Not to be used to store “user preferences”
- Multiple files: `getSharedPreferences()`
- Single file: `getPreferences()`
- Write with: `SharedPreferences.Editor`

Internal Storage

- Store files on device's internal storage
- Files saved are private by default
 - `MODE_PRIVATE`, `MODE_APPEND`,
`MODE_WORLD_READABLE`,
`MODE_WORLD_WRITEABLE`
- Cache files with `getCacheDir()`
 - Android deletes when space is low
- Static files in `res/raw` directory
 - Access w/ `openRawResource()`

External Storage

- `getExternalFilesDir()` while specifying “type”
 - `DIRECTORY_ALARM`, `DIRECTORY_DCIM`, `DIRECTORY_DOWNLOADS`,
`DIRECTORY_MOVIES`, `DIRECTORY_MUSIC`, `DIRECTORY_NOTIFICATIONS`,
`DIRECTORY_PICTURES`, `DIRECTORY_PODCASTS`, `DIRECTORY_RINGTONES`
- Allows media scanner to find files
 - Hide with `.nomedia` file in directory
- `getExternalCacheDir()`
 - deleted when uninstalled, we manage other times
- User can “disappear” your files being dumb

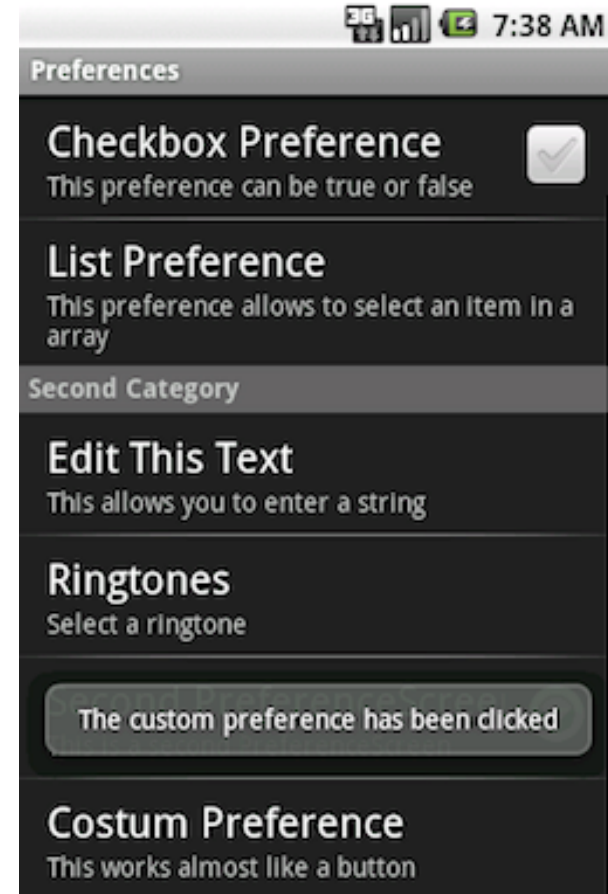
SQLite

- Created databases are private to application, but available to all classes
- SQLiteOpenHelper
 - Manage creation and version management
- SQLiteQueryBuilder()
- Cursor
- Connect from host computer

```
$ adb -s <emulator serial number> shell
#sqlite3 <dbname>
sqlite> .exit
```

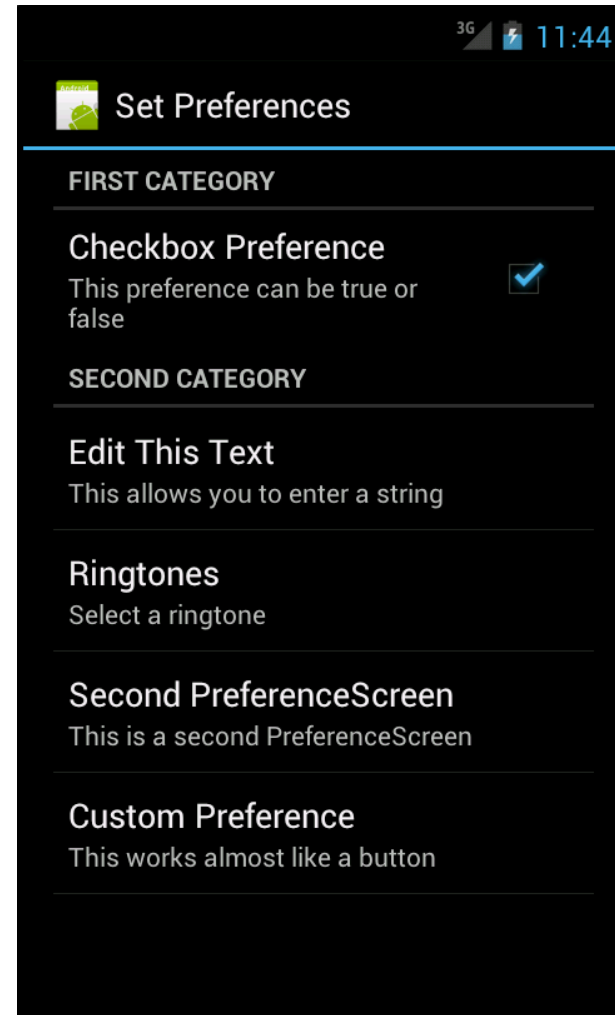
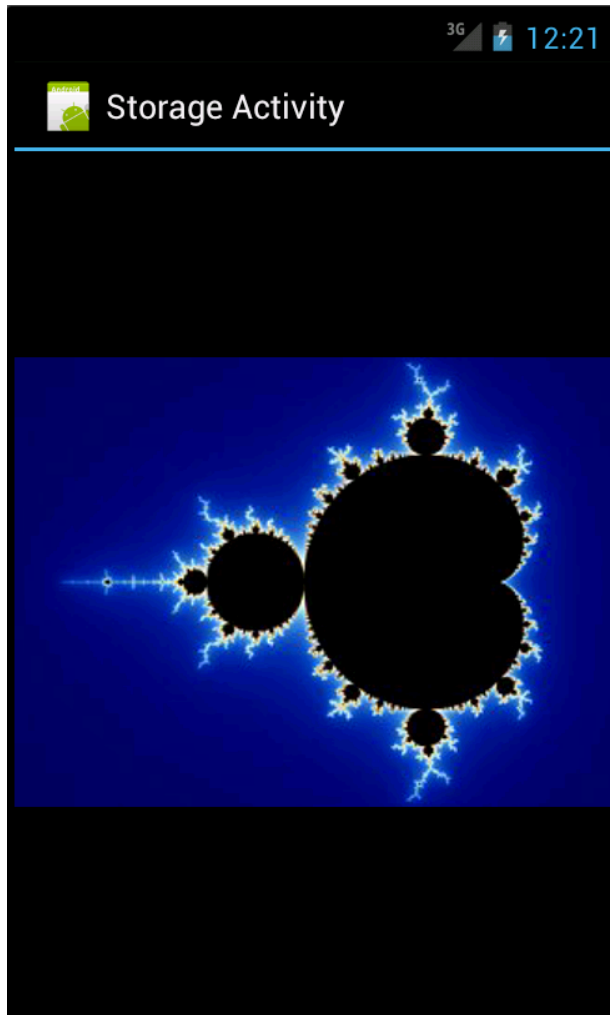
PreferenceActivity

- Makes storing user application data easy
- 5 different preference views
 - CheckBoxPreference
 - ListPreference
 - EditTextPreference
 - RingtonePreference
 - Preference (custom)



Source: <http://www.kaloer.com/android-preferences>

Lab 4



Location, Location, Location

- MapView
- MapActivity
- Geocoder

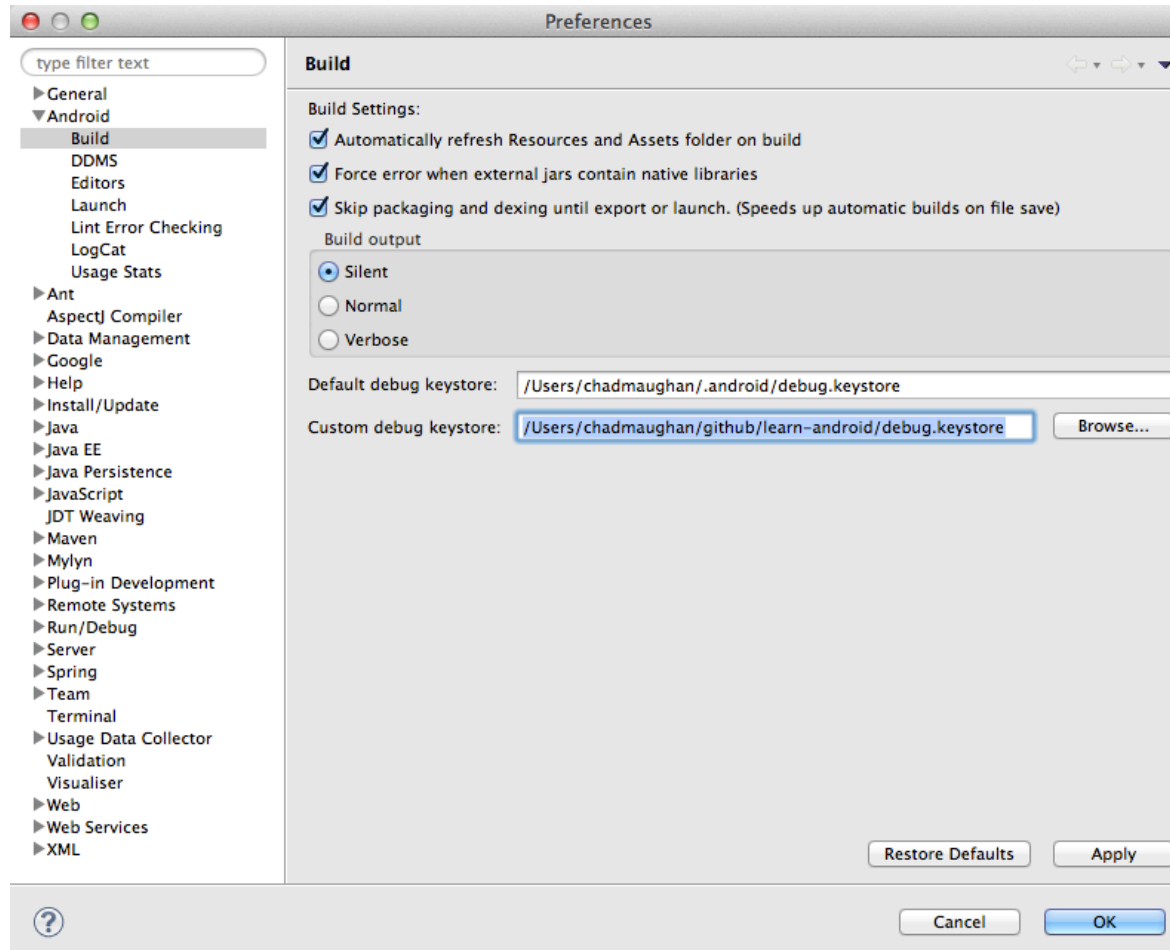
Get API Key

<http://code.google.com/android/maps-api-signup.html>

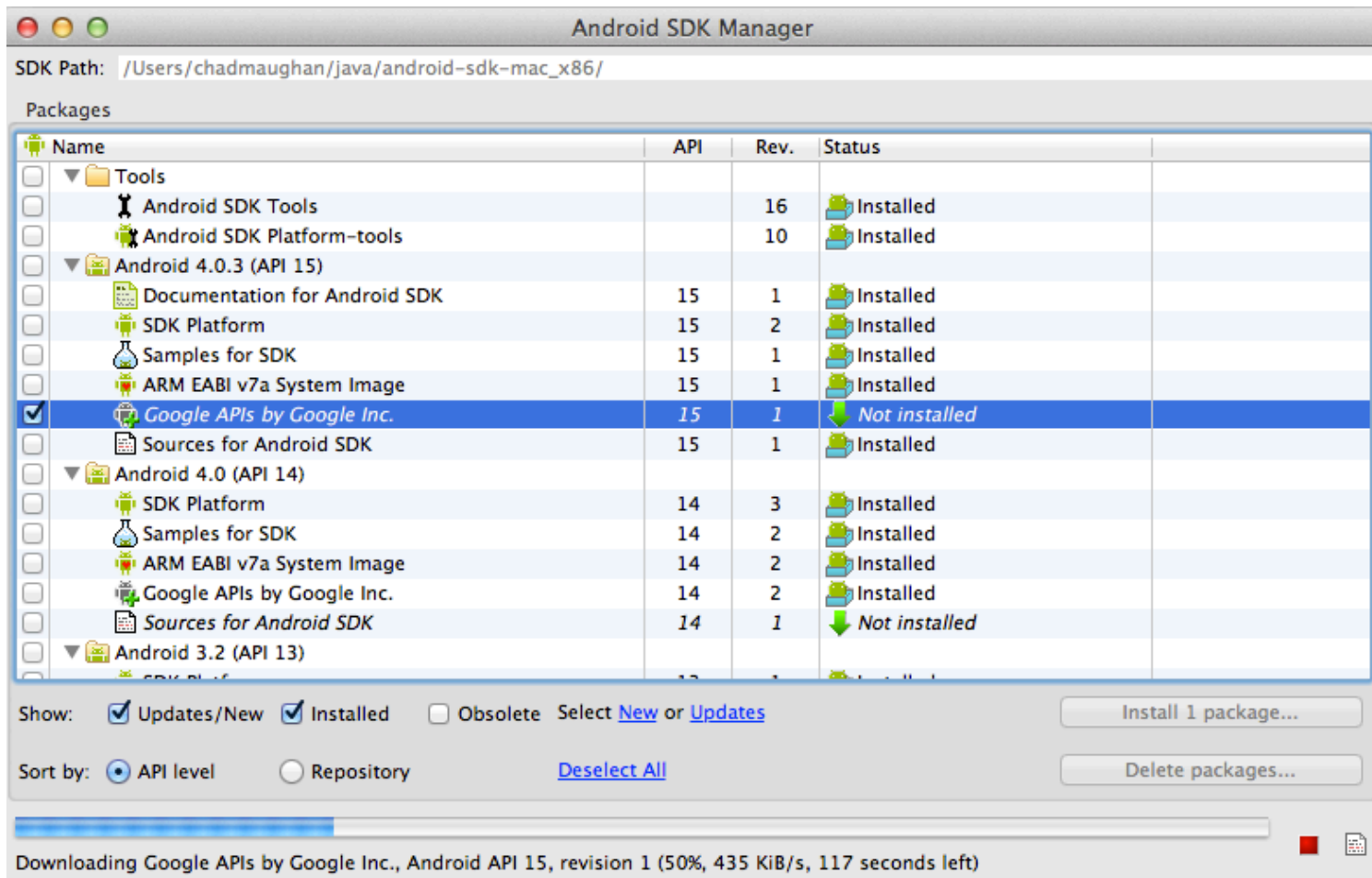
```
keytool -list -alias androiddebugkey -  
keystore "/full/path/learn-android/  
debug.keystore" -storepass android -keypass  
android
```

```
androiddebugkey, Sep 4, 2011,  
PrivateKeyEntry,  
Certificate fingerprint (MD5): 70:3C:  
3D:A9:52:2B:73:9E:E1:81:B4:DD:B9:5B:F6:F8
```

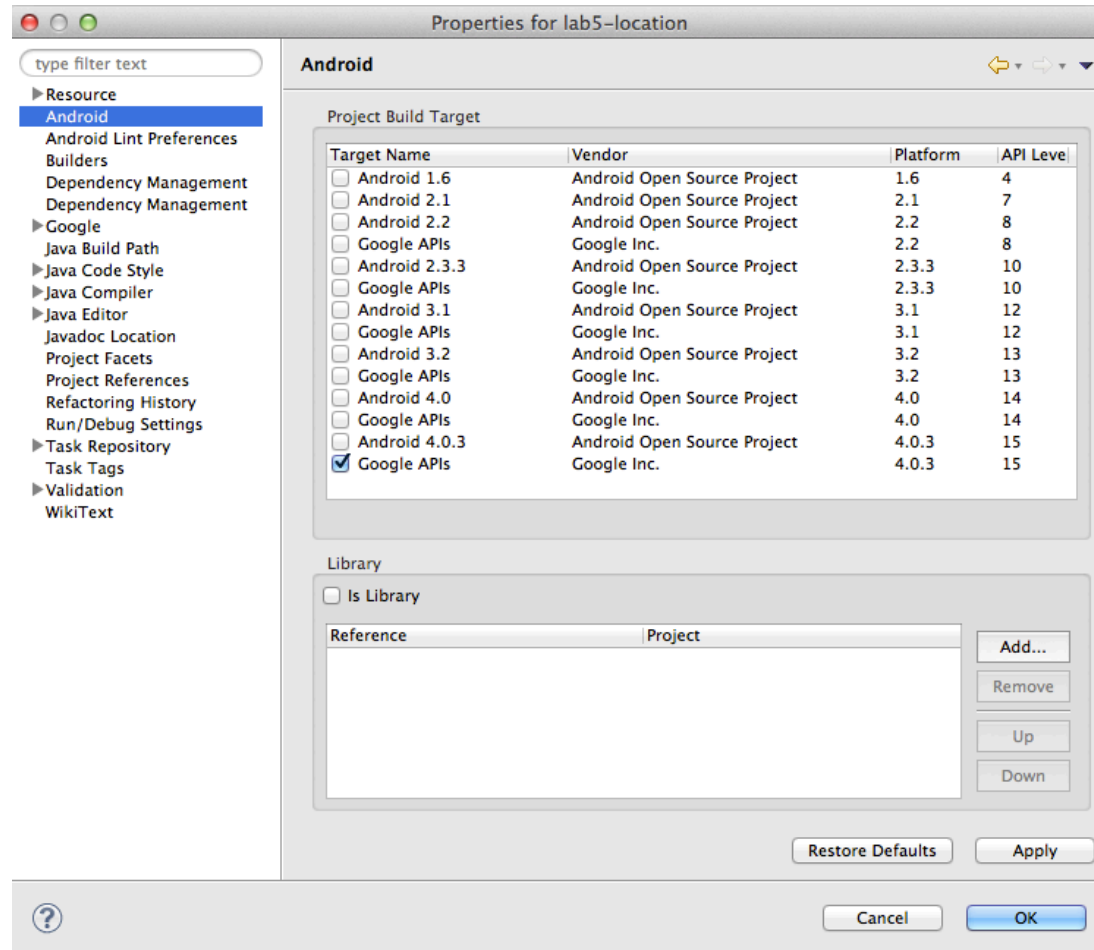

Use Debug Keystore



Install APIs



Change Build Target



Lab 5

