



nVIDIA®

AN INTRODUCTION TO ANDROID DEVELOPMENT
CS231M | Alejandro Troccoli

Outline

- Overview of the Android Operating System
- Development tools
- Deploying application packages
- Step-by-step application development

The Android ecosystem

- An open source platform for mobile, embedded and wearable devices
- Google is the principle maintainer
- Other companies contribute to the system.
- Each device manufacturer can customize Android to suite their needs

Android architecture

User applications
Use Java framework and, optionally, native code.

Android framework
Java classes under com.android

Native framework layer
User mode C, C++ code - compiled to native platform or 32bit
compatibility mode on 64 bits.

Linux Kernel (GPL license)
C code - compiled to native platform (x86, arm, mips)

Android versioning

- Platform version
 - 5.0 Lollipop
 - 4.4 KitKat
- Framework API level
 - SDK compatibility
 - Each platform version has an API level
- NDK API level
 - API level for native headers
- Distribution
 - <http://developer.android.com/about/dashboards/index.html>



Browsing the Android Source

- Source at:
 - <https://android.googlesource.com/>
- Porting instructions (for system developers)
 - <https://source.android.com/devices/index.html>

Reference to the framework APIs

- com.android classes
 - <http://developer.android.com/reference/packages.html>

Outline

- Overview of the Android Operating System
- **Development tools**
- Deploying application packages
- Step-by-step application development

Tools for application development

- Android SDK
 - Provides the Java framework classes
 - Compiles to java bytecode
 - Class framework is updated with every OS release
- Android NDK
 - C/C++ toolchain for compiling to machine code
- Android platform tools
 - adb (android debug bridge) : runs and debugs apps from your dev machine
- Android developer tools
 - Eclipse plug-in for Android
 - Android studio (doesn't yet fully support all NDK features)

Setup options

- Tegra Android Development pack
- Install components manually
- Use a Linux virtual machine with the tools pre-installed.

Tegra Android Development Pack

The screenshot shows the NVIDIA GameWorks website's download center. The browser address bar displays <https://developer.nvidia.com/gameworksdownload>. The navigation menu includes links for About, Platforms, Devices, Documentation, Support, Downloads, and Login. Below the navigation, there are categories like VisualFX, PhysX, Core SDK, OptiX, Samples, and Tools. The main heading is "DOWNLOAD CENTER". The breadcrumb trail is "Home > GameWorks > Download Center". A search filter box is present with the text "Filter...". The text "Showing 55 downloads." is displayed above a table. The table has three columns: "Title", "Version", and "Release Date". The "Tegra Android Development Pack" is highlighted in yellow in the original image.

Title	Version	Release Date
> PhysX: Core PhysX SDK	3.3.0	2014/02/21
> PhysX: APEX SDK	1.3.0	2014/02/21
> Tegra Android Development Pack	2.0r8	2013/03/18
> PhysX: 3D Studio MAX DCC plug-in	3.0.0	2014/02/21
> PhysX: Maya DCC plug-in	3.0.0	2014/02/21
> PhysX Lab	1.3.0	2014/02/21

Tegra Android Development Pack

- Register for an account at:
 - <https://developer.nvidia.com/user/register>
- Sign-up for Gameworks Registered Developer Program

NVIDIA GameWorks Registered Developer Program Application:

What platforms are you interested in? (Select all that apply): *

- GeForce
- Quadro
- Tegra
- GRID
- Other...

What products families are you interested in? (Select all that apply): *

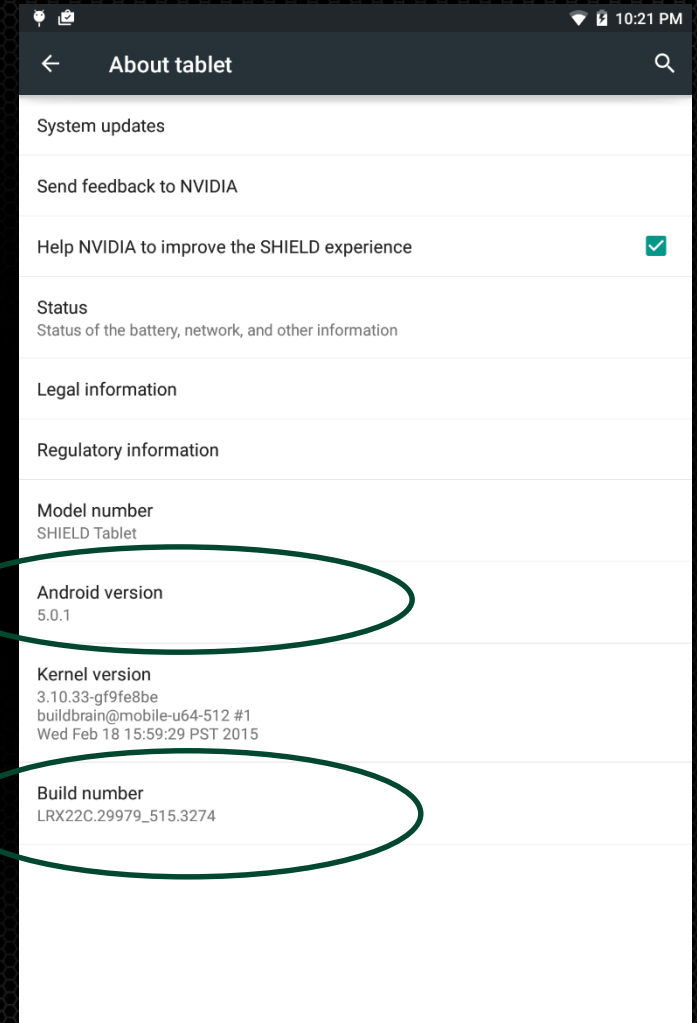
- Core SDK
- Developer Tools
- Graphics and Compute Samples
- OptiX
- PhysX
- VisualFX
- Other...

Select the specific topics you are interested in (Select all that apply): *

- NVIDIA Nsight Visual Studio Edition
- Tegra Android Development Pack
- NVAPI
- DirectX
- OpenGL
- OpenGL ES
- NVIDIA SHIELD
- Mobile Development
- Android Development
- Other...

[Complete Registration](#)

NVIDIA Shield Tablet



Click 7 times to enable developer options 

Outline

- Overview of the Android Operating System
- Development tools
- **Deploying application packages**
- Step-by-step application development

Application packages

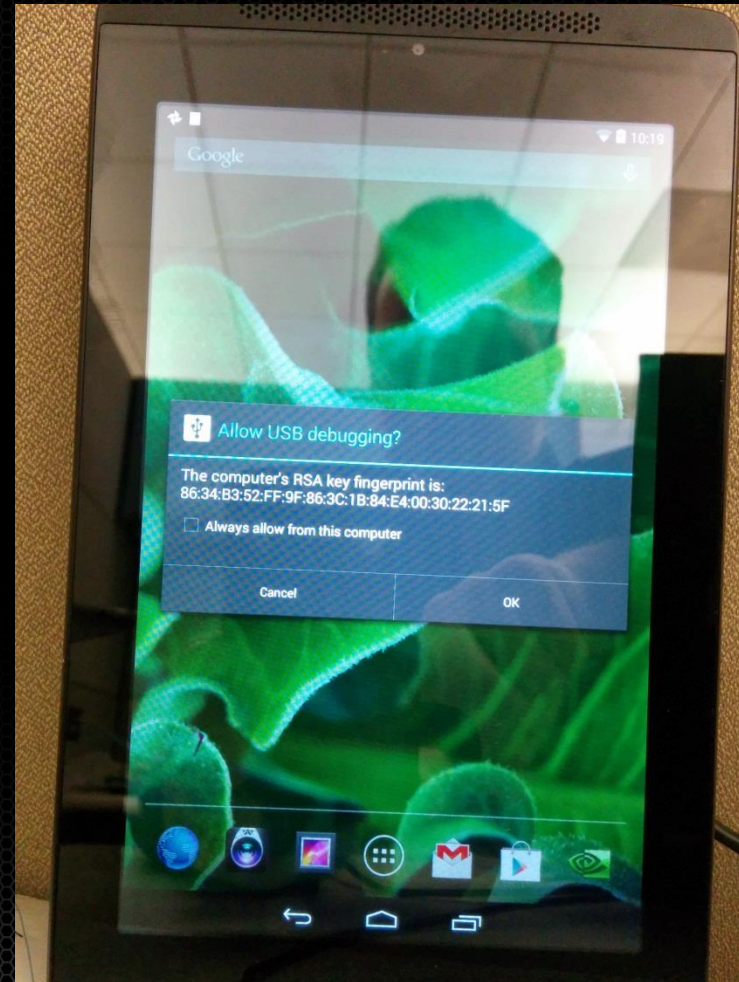
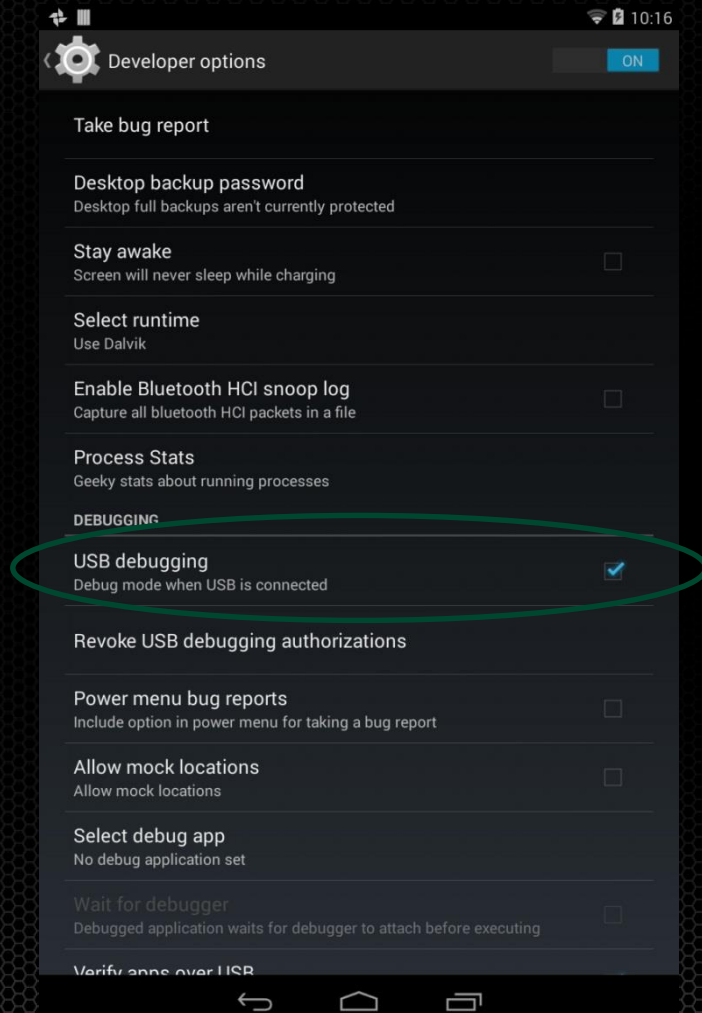
- .apk files: compressed files
 - class byte code
 - resources(icons, sounds, etc).
 - Binary native files
- All .apks are signed
 - Default development key is created by SDK.
 - When updating an application, signature are checked.

Installing an application

- From application distribution markets
 - Google Play
 - Amazon AppStore
- From your local computer using adb

```
C:\work\tadp\2.0r8\TDK_Samples\tegra_android_native_samples_v10p14\prebuilt>adb install native_globe.apk
7015 KB/s (5826921 bytes in 0.811s)
  pkg: /data/local/tmp/native_globe.apk
Success
C:\work\tadp\2.0r8\TDK_Samples\tegra_android_native_samples_v10p14\prebuilt>
```


Enabling android debug bridge (adb)



Useful adb commands

```
C:\>adb devices
List of devices attached
0524513118124000E614    device
```

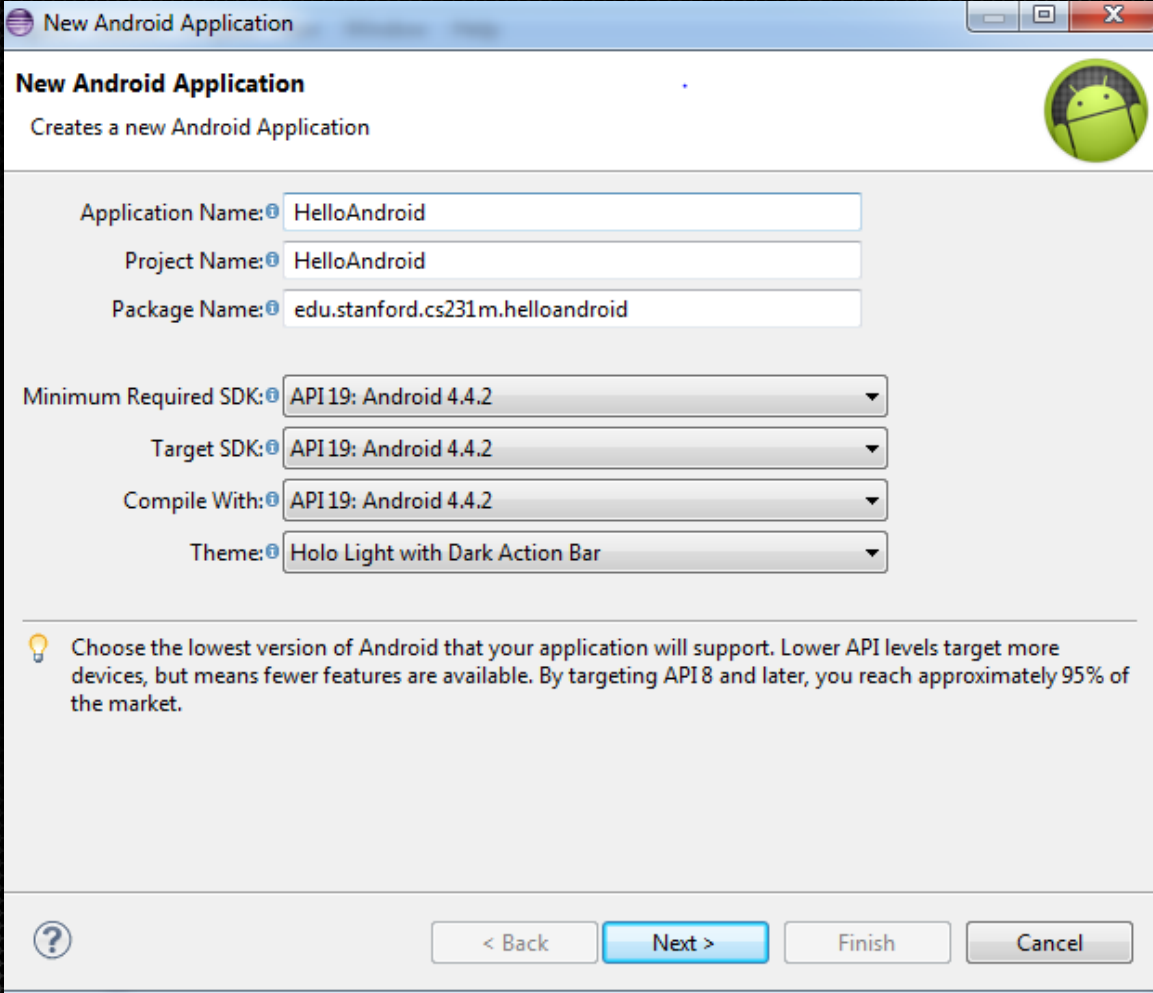
```
C:\>adb logcat
----- beginning of /dev/log/system
I/Vold      ( 217): Vold 2.1 (the revenge) firing up
D/Vold      ( 217): Volume sdcard1 state changing -1 (Initializing) -> 0 (No-Media)
D/Vold      ( 217): Volume usbdrive state changing -1 (Initializing) -> 0 (No-Media)
I/SystemServer( 997): Entered the Android system server!
I/SystemServer( 997): Waiting for installd to be ready.
I/Installer( 997): connecting...
I/SystemServer( 997): Power Manager
^C
```

Outline

- Overview of the Android Operating System
- Development tools
- Deploying application packages
- **Step-by-step application development**

Hello Android!

- In Eclipse
 - File -> New -> Android Application Project




The screenshot shows the 'New Android Application' dialog box in the Eclipse IDE. The dialog has a title bar with the text 'New Android Application' and a close button. Below the title bar, the text 'New Android Application' is displayed in bold, followed by the subtitle 'Creates a new Android Application'. On the right side, there is an Android robot icon. The main area contains several input fields and dropdown menus:

- Application Name: HelloAndroid
- Project Name: HelloAndroid
- Package Name: edu.stanford.cs231m.helloandroid
- Minimum Required SDK: API 19: Android 4.4.2
- Target SDK: API 19: Android 4.4.2
- Compile With: API 19: Android 4.4.2
- Theme: Holo Light with Dark Action Bar

At the bottom, there is a lightbulb icon followed by a tip: 'Choose the lowest version of Android that your application will support. Lower API levels target more devices, but means fewer features are available. By targeting API 8 and later, you reach approximately 95% of the market.' Below the tip, there are four buttons: '< Back', 'Next >', 'Finish', and 'Cancel'.

Hello Android!

New Android Application 

Configure Project

Create custom launcher icon

Create activity

Mark this project as a library

Create Project in Workspace

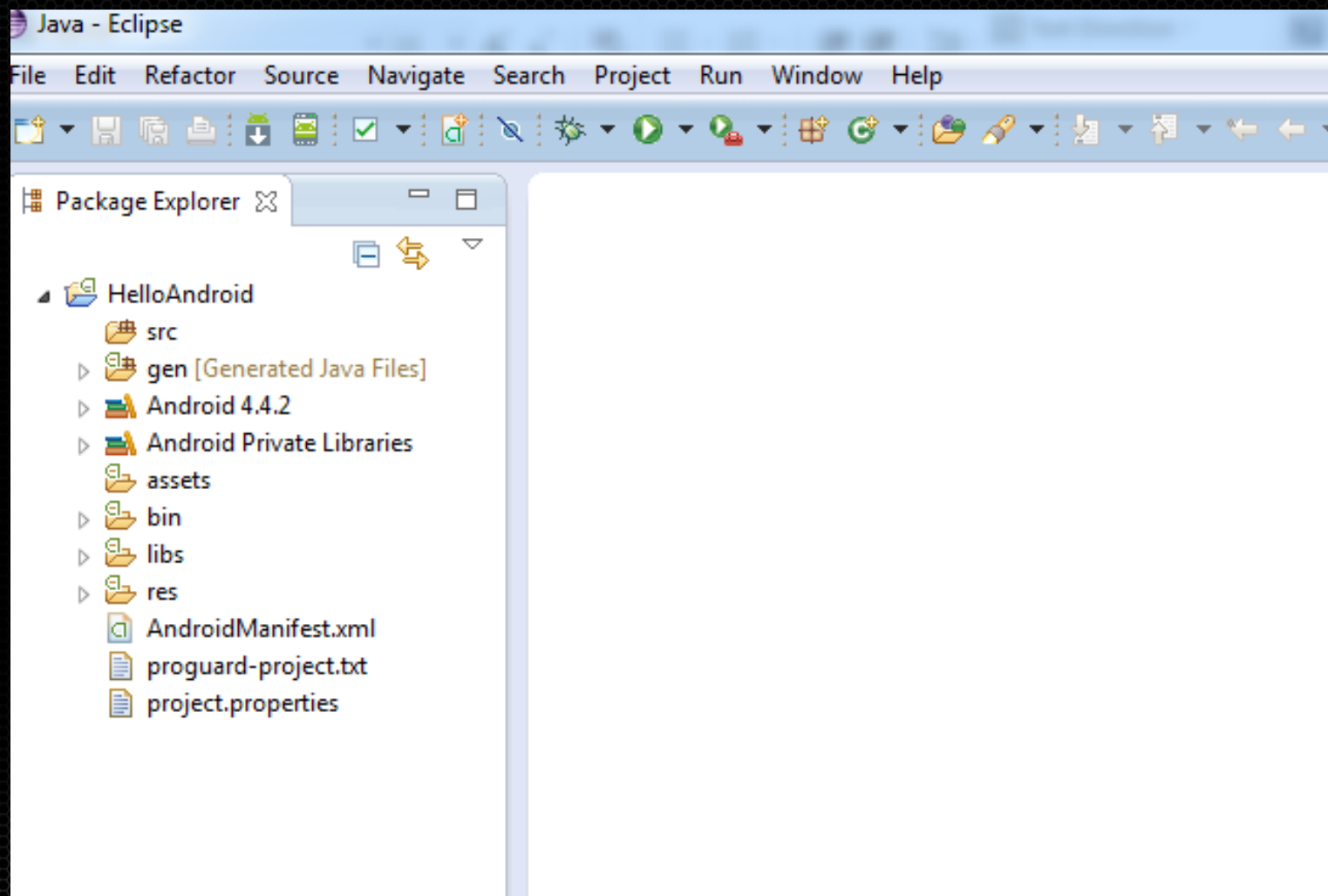
Location:

Working sets

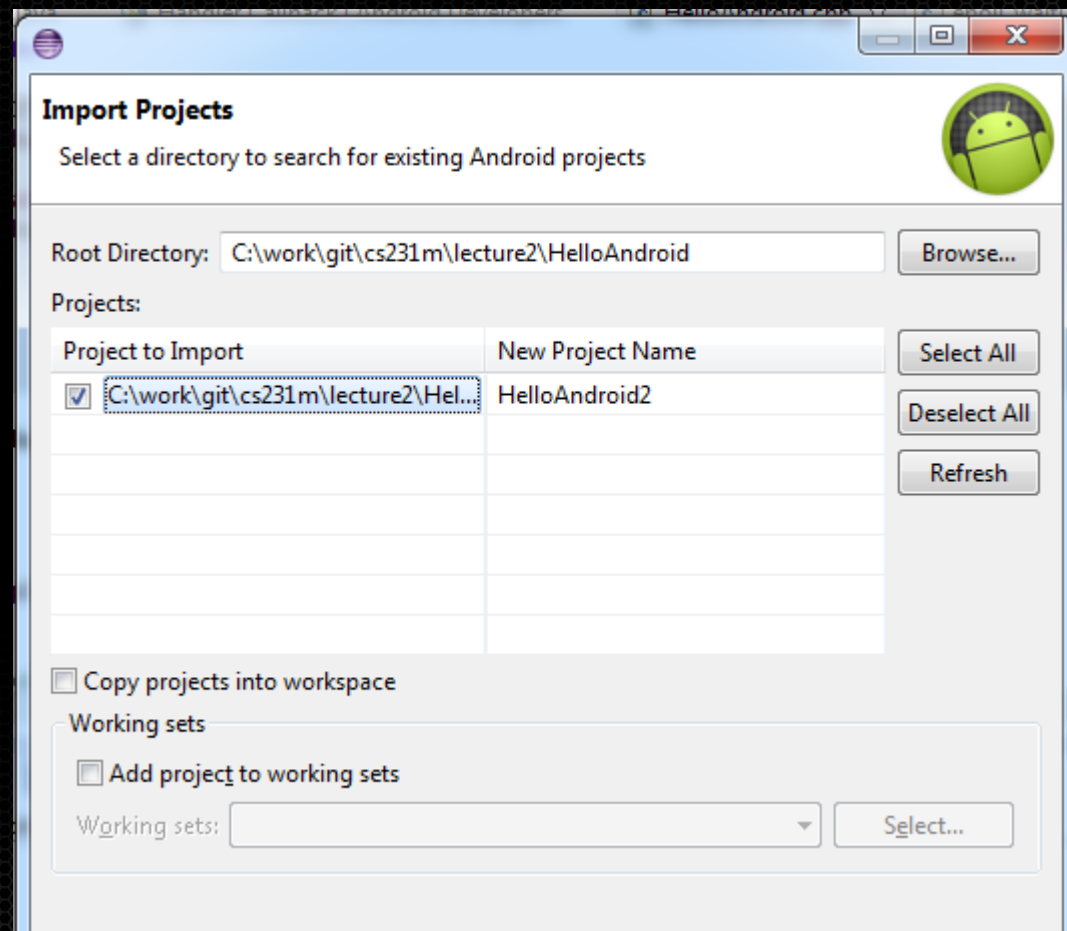
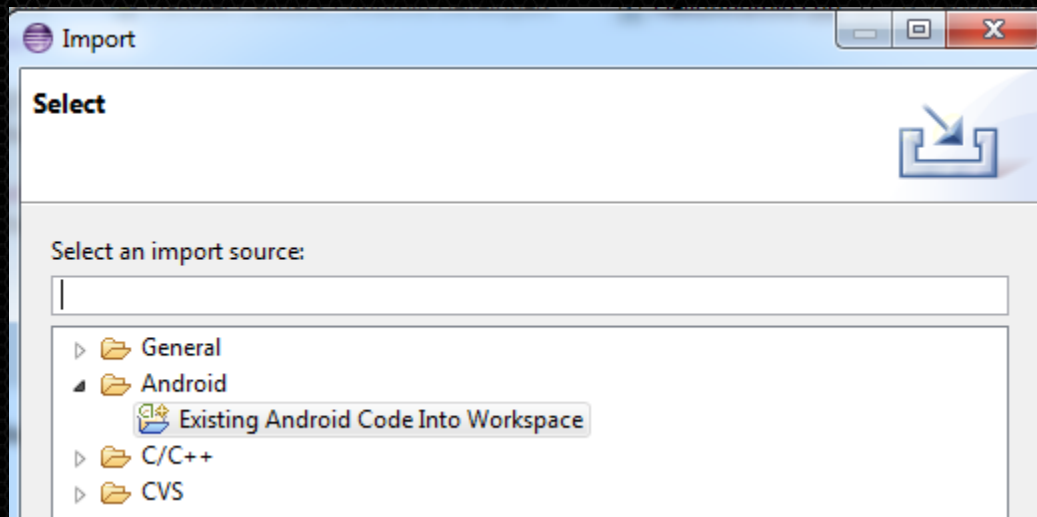
Add project to working sets

Working sets:

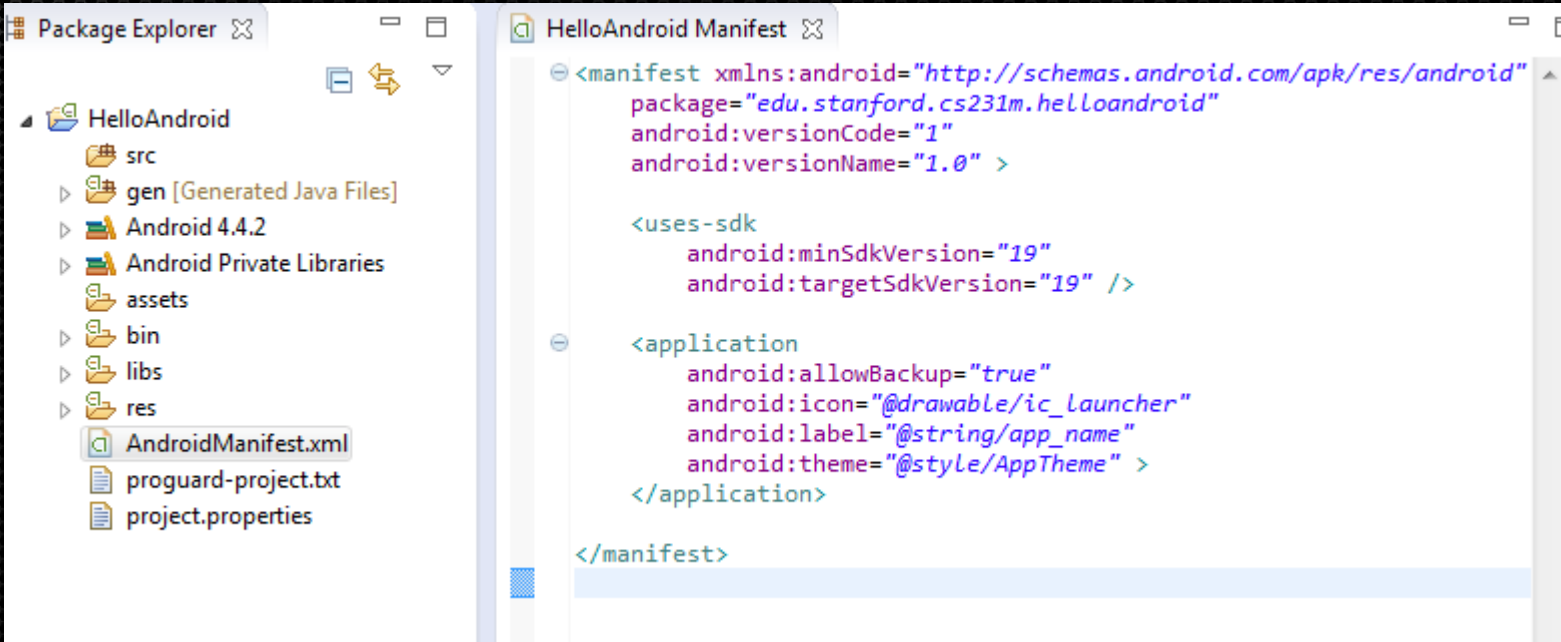
Hello Android!



Import an project from existing code



AndroidManifest.xml



The screenshot shows an IDE window with the Package Explorer on the left and the HelloAndroid Manifest editor on the right. The Package Explorer shows the project structure: HelloAndroid, src, gen [Generated Java Files], Android 4.4.2, Android Private Libraries, assets, bin, libs, res, AndroidManifest.xml, proguard-project.txt, and project.properties. The HelloAndroid Manifest editor displays the following XML code:

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="edu.stanford.cs231m.helloandroid"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="19"
        android:targetSdkVersion="19" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
    </application>

</manifest>
```

- Package Name / version
- Required SDK and target SDK
- Application/Activities
- Permissions

Android Activity

- Provides user interaction
 - <http://developer.android.com/reference/android/app/Activity.html>
- Callbacks for life-cycle management
 - onCreate()
 - onResume()
 - onPause()
- An application can have multiple activities..
 - Needs one launcher activity...

HelloAndroidActivity

New Java Class

Java Class
Create a new Java class.

Source folder: HelloAndroid/src

Package: edu.stanford.cs231m.helloandroid

Enclosing type:

Name: HelloAndroidActivity

Modifiers: public default private protected
 abstract final static

Superclass: android.app.Activity

Interfaces:

Which method stubs would you like to create?

public static void main(String[] args)
 Constructors from superclass
 Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))
 Generate comments

Package Explorer

- HelloAndroid
 - src
 - edu.stanford.cs231m.helloandroid
 - HelloAndroidActivity.java
 - gen [Generated Java Files]
 - Android 4.4.2
 - Android Private Libraries
 - assets
 - bin
 - libs
 - res
 - AndroidManifest.xml
 - proguard-project.txt
 - project.properties

HelloAndroid Manifest HelloAndroidActivity.java

```
package edu.stanford.cs231m.helloandroid;  
  
import android.app.Activity;  
  
public class HelloAndroidActivity extends Activity {  
  
}
```

HelloAndroidActivity

- Use onCreate() to create UI.

```
public class HelloAndroidActivity extends Activity {  
  
    @Override  
    public void onCreate(Bundle settings) {  
  
        TextView txtView = new TextView(this);  
        txtView.setText("Hello Android!");  
        setContentView(txtView);  
    }  
}
```

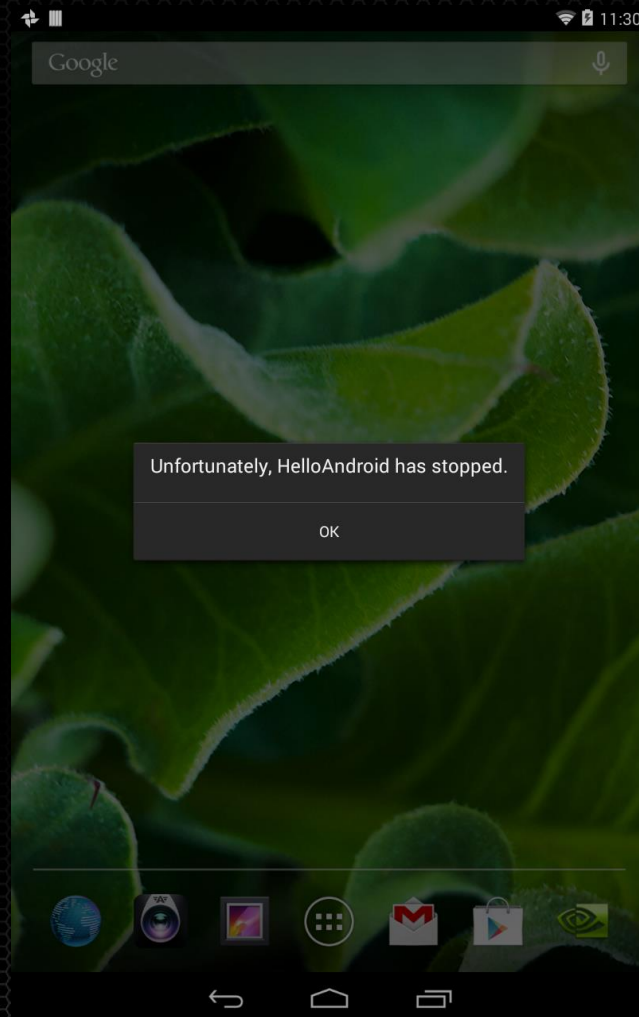
Launch!

```
Unable to write output: already prepared
[2014-03-26 11:08:37 - HelloAndroid] -----
[2014-03-26 11:08:37 - HelloAndroid] Android Launch!
[2014-03-26 11:08:37 - HelloAndroid] adb is running normally.
[2014-03-26 11:08:37 - HelloAndroid] No Launcher activity found!
[2014-03-26 11:08:37 - HelloAndroid] The launch will only sync the application package on the device!
[2014-03-26 11:08:37 - HelloAndroid] Performing sync
[2014-03-26 11:08:37 - HelloAndroid] Automatic Target Mode: using device '0524513118124000E614'
[2014-03-26 11:08:37 - HelloAndroid] Uploading HelloAndroid.apk onto device '0524513118124000E614'
[2014-03-26 11:08:37 - HelloAndroid] Installing HelloAndroid.apk...
[2014-03-26 11:08:39 - HelloAndroid] Success!
[2014-03-26 11:08:39 - HelloAndroid] \HelloAndroid\bin\HelloAndroid.apk installed on device
[2014-03-26 11:08:39 - HelloAndroid] Done!
```

Add Activity to AndroidManifest.xml:

```
<application
  android:allowBackup="true"
  android:icon="@drawable/ic_launcher"
  android:label="@string/app_name"
  android:theme="@style/AppTheme" >
  <activity android:name="HelloAndroidActivity" android:label="@string/app_name">
    <intent-filter>
      <category android:name="android.intent.category.LAUNCHER"/>
      <action android:name="android.intent.action.MAIN"/>
    </intent-filter>
  </activity>
</application>
```

Launch! (take 2)



Launch with debugger

Debug - Source not found. - Eclipse

File Edit Navigate Search Project Run Window Help

Debug Package Explorer

- HelloAndroid [Android Application]
 - DalvikVM [localhost:8601]
 - Thread [<1> main] (Suspended (exception SuperNotCalledException))
 - <VM does not provide monitor information>
 - ActivityThread.performLaunchActivity(ActivityThread\$ActivityClientRecord, Intent) line: 21
 - ActivityThread.handleLaunchActivity(ActivityThread\$ActivityClientRecord, Intent) line: 224
 - ActivityThread.access\$800(ActivityThread, ActivityThread\$ActivityClientRecord, Intent) line: 1196
 - ActivityThread\$H.handleMessage(Message) line: 102
 - ActivityThread\$H(Handler).dispatchMessage(Message) line: 136
 - Looper.loop() line: 5017
 - ActivityThread.main(String[]) line: 601
 - Method.invokeNative(Object, Object[], Class, Class[], Class, int, boolean) line: not available
 - Method.invoke(Object, Object...) line: 785
 - ZygoteInit\$MethodAndArgsCaller.run() line: 601
 - ZygoteInit.main(String[]) line: not available [native method]

Source not found.

Edit Source Lookup Path...

LogCat

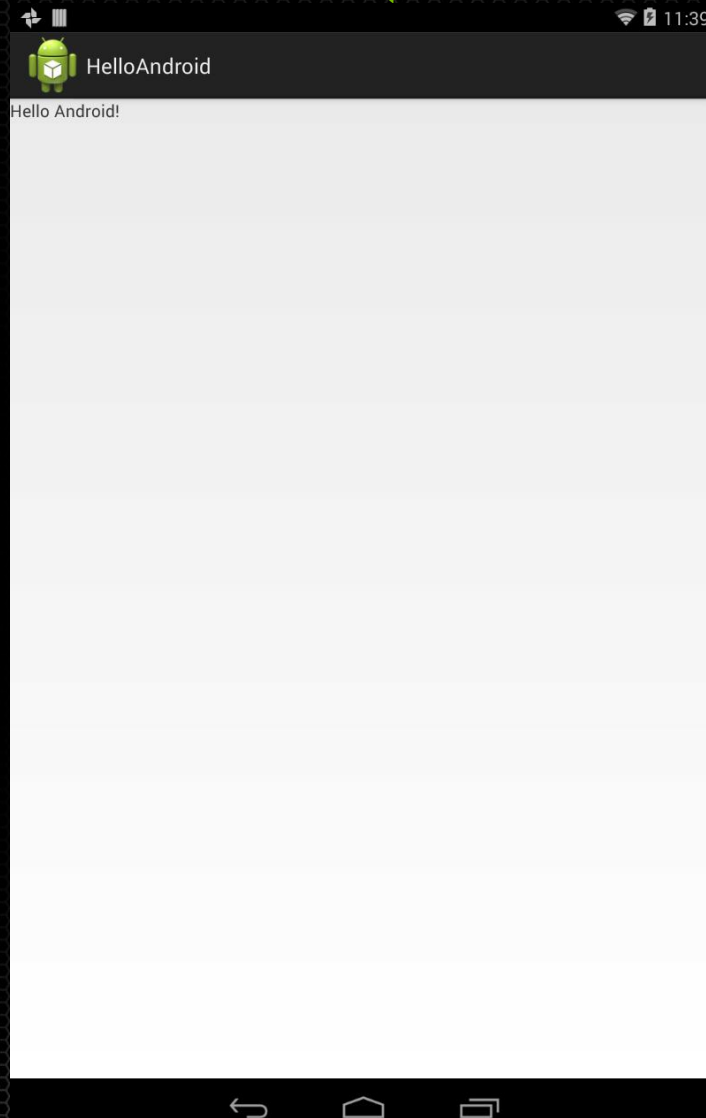
Search for messages. Accepts Java regexes. Prefix with pi verbose

L...	Time	PID	TID	Application	Tag
D	03-26 11:34:1...	11747	11747	edu.stanford.cs...	dalvikvm

Variables Breakpoints

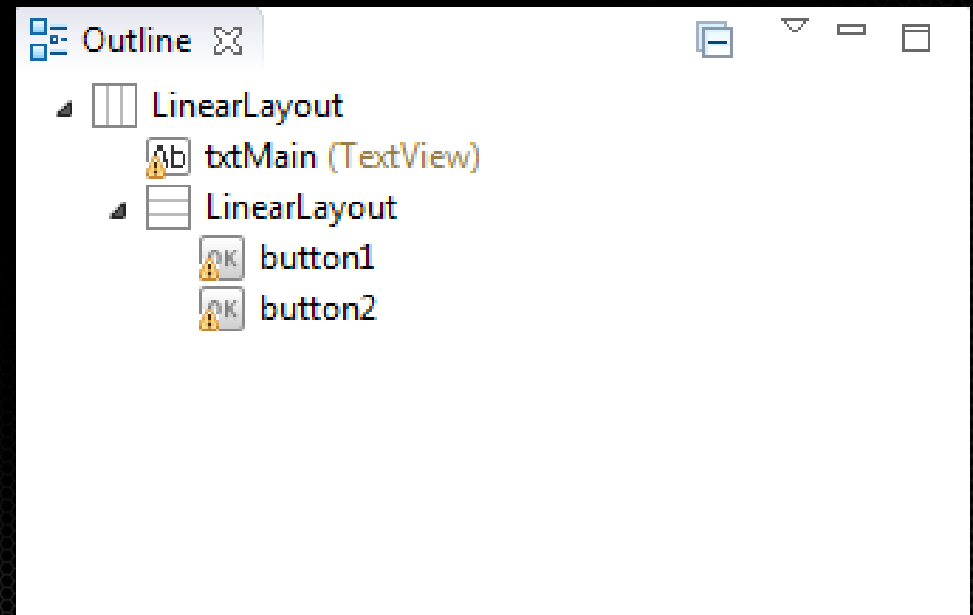
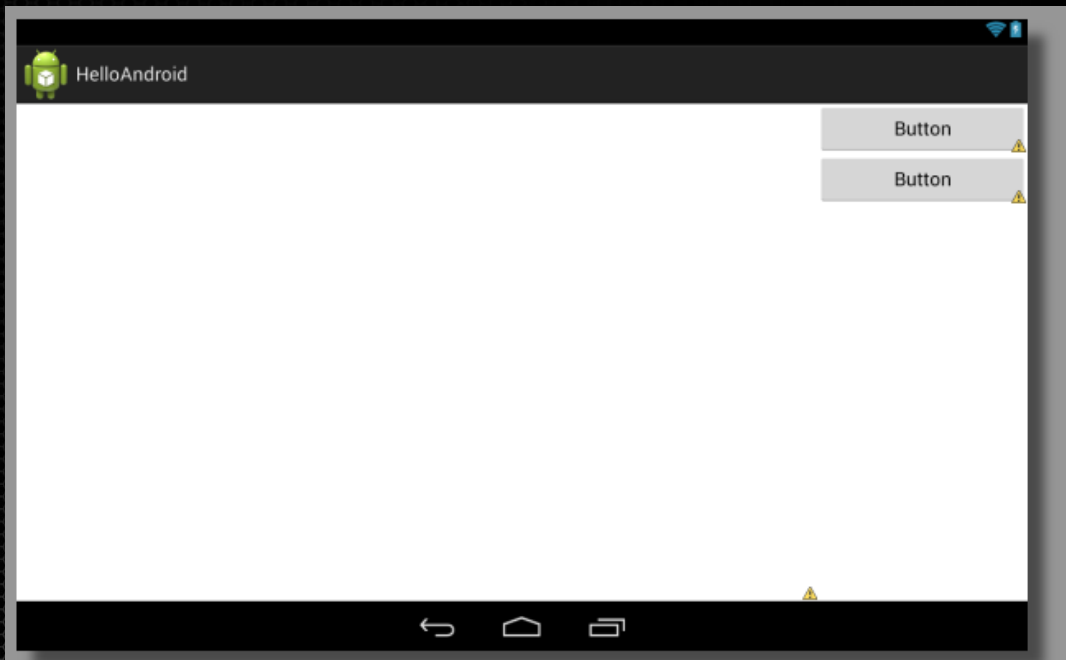
Name	Value
this	ActivityThread (id=830038276528)
e	SuperNotCalledException (id=830038427472)
cause	SuperNotCalledException (id=830038427472)
cause	SuperNotCalledException (id=830038427472)
detailMessage	"Activity {edu.stanford.cs231m.helloandroid/edu.stanford.cs231m.helloandroid/edu.stanford.cs231m.helloandroid.HelloAndroidActivity} did not call through to super.onCreate()"
stackState	(id=830038429488)
stackTrace	StackTraceElement[0] (id=830035424512)
suppressedExceptions	Collections\$EmptyList (id=830035422392)
detailMessage	"Activity {edu.stanford.cs231m.helloandroid/edu.stanford.cs231m.helloandroid/edu.stanford.cs231m.helloandroid.HelloAndroidActivity} did not call through to super.onCreate()"

Launch (take 3!)

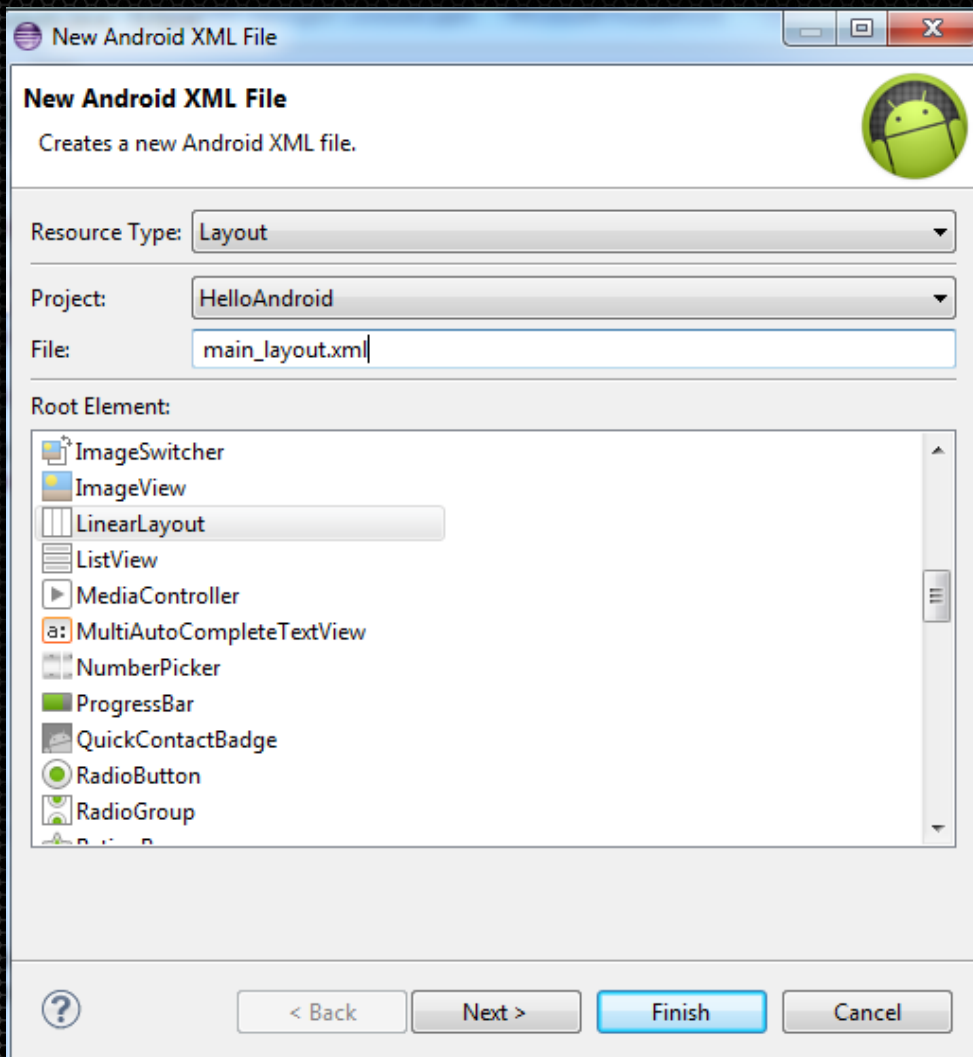


Views

- Can be composed in a tree hierarchy.
- The root View is the argument to setContentView



Creating a layout



```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal" >

    <TextView
        android:id="@+id/txtMain"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:layout_weight="1" />

    <LinearLayout
        android:layout_width="200dp"
        android:layout_height="wrap_content"
        android:orientation="vertical" >

        <Button
            android:id="@+id/button1"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="Button" />

        <Button
            android:id="@+id/button2"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="Button" />

    </LinearLayout>

</LinearLayout>
```

Accessing layout elements from Activity

```
public class HelloAndroidActivity extends Activity {  
  
    private TextView mMainText;  
    private Button mButton1;  
    private Button mButton2;  
  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
  
        // Setup the layout hierarchy  
        setContentView(R.layout.main_layout);  
  
        // Find the layout elements  
        mMainText = (TextView) findViewById(R.id.txtMain);  
        mButton1 = (Button) findViewById(R.id.button1);  
        mButton2 = (Button) findViewById(R.id.button2);  
  
        mMainText.setText("HelloAndroid!");  
  
        super.onCreate(savedInstanceState);  
    }  
}
```

● Use findViewById

R.id.name corresponds to the name given in the xml file

Event listeners (and logging, too)

```
// Button1 action on click
mButton1.setOnClickListener( new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        mMainText.setText("Button 1 was pressed!");
        Log.i(TAG, "Button 1 was pressed!");
    }
});

// Button2 action on click
mButton2.setOnClickListener( new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        mMainText.setText("Button 2 was pressed!");
        Log.i(TAG, "Button 2 was pressed!");
    }
});
```

Logcat

- Window -> Show View -> Other -> Android -> Logcat

Search for messages. Accepts Java regexes. Prefix with pid;, app;, tag; or text: to limit scope. verbose

L...	Time	PID	TID	Application	Tag	Text
						uquiet/active": Invalid argument
I	03-27 13:38:2...	304	304		nvusd	Automatically enabled
I	03-27 13:38:2...	304	304		nvusd:tbc	TBC probe cpu limit=89.00
I	03-27 13:38:2...	304	304		nvusd:tbc	TBC probe ext original=43000, backof
I	03-27 13:38:2...	8347	8347	edu.stanford.cs231m.helloandroid	HelloAndro...	Button 1 was pressed!
I	03-27 13:38:2...	8347	8347	edu.stanford.cs231m.helloandroid	HelloAndro...	Button 2 was pressed!
D	03-27 13:38:2...	233	263		volt_cap	VC: Volt update change is -18024.351
I	03-27 13:38:2...	8347	8347	edu.stanford.cs231m.helloandroid	HelloAndro...	Button 2 was pressed!
D	03-27 13:38:2...	8347	8347	edu.stanford.cs231m.helloandroid	dalvikvm	GC_FOR_ALLOC freed 309K, 12% free 29 8ms
I	03-27 13:38:2...	8347	8347	edu.stanford.cs231m.helloandroid	HelloAndro...	Button 2 was pressed!
I	03-27 13:38:2...	8347	8347	edu.stanford.cs231m.helloandroid	HelloAndro...	Button 2 was pressed!
I	03-27 13:38:2...	8347	8347	edu.stanford.cs231m.helloandroid	HelloAndro...	Button 2 was pressed!
I	03-27 13:38:2...	8347	8347	edu.stanford.cs231m.helloandroid	HelloAndro...	Button 2 was pressed!
I	03-27 13:38:2...	304	304		nvusd	Automatically disabled
V	03-27 13:38:3...	2679	2679		NVSS	16 ---Desk---
V	03-27 13:38:3...	226	1114		nvaudio_hw	nvaudio_dev_set_parameters : EQ_MODE

Creating a log file

```
private BufferedWriter openLogFile()
{
    File appExternalDir = new File( Environment.getExternalStorageDirectory(),
        "HelloAndroid");

    if ( !appExternalDir.exists() )
    {
        if ( appExternalDir.mkdirs() )
        {
            Log.i(TAG, "External storage directory created: " + appExternalDir.toString() );
        }
        else
        {
            Log.e(TAG, "Failed to create directory " + appExternalDir.toString() );
            return null;
        }
    }

    File logFile = new File( appExternalDir, "log.txt");

    BufferedWriter writer = null;
    try {
        writer = new BufferedWriter( new FileWriter(logFile));
    } catch (IOException e) {
        Log.e(TAG, "Failed to create file " + logFile.toString() );
        return null;
    }

    return writer;
}
```

Writing to the log file

```
private void logMessage( String message )
{
    if ( mLogWriter != null )
    {
        try {
            mLogWriter.write(message);
            mLogWriter.newLine();
            mLogWriter.flush();
        } catch (IOException e) {
            Log.e(TAG, "Failed to write to log file");
        }
    }
}
```

```
> <manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="edu.stanford.cs231m.helloandroid"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="19"
        android:targetSdkVersion="19" />

    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
```

Long running task

- Long running tasks on the main thread can block the UI
- App looks unresponsive

```
private void longRunningTask( long taskDurationInMs )
{
    long startTime = System.currentTimeMillis();
    mMainText.append("Starting long running task at " + startTime + "\n" );

    long currentTime = startTime;
    do
    {
        try {
            Thread.sleep( taskDurationInMs );
        } catch (InterruptedException e) {
        }

        currentTime = System.currentTimeMillis();

    } while ( currentTime < startTime + taskDurationInMs );

    mMainText.append("Ended long running task at " + currentTime + "\n");
}
```

Use a separate Thread instead

```
private BufferedWriter mLogWriter = null;  
private Thread mWorkerThread = null;
```

```
// Button2 action on click  
mButton2.setOnClickListener( new View.OnClickListener() {  
  
    @Override  
    public void onClick(View v) {  
        mMainText.setText("Button 2 was pressed!\n");  
        Log.i(TAG, "Button 2 was pressed!");  
        logMessage("Button 2 was pressed!");  
  
        mWorkerThread = new Thread( new Runnable() {  
  
            @Override  
            public void run() {  
                longRunningTask(6000);  
            }  
        });  
  
        mWorkerThread.start();  
  
    }  
});
```


Use Handlers to update UI

```
private Handler mHandler = null;

private final static int MSG_ASYNC_TASK_STARTED = 0;
private final static int MSG_ASYNC_TASK_COMPLETED = 1;
```

```
mHandler = new Handler( mHandlerCallback );
```

```
private Handler.Callback mHandlerCallback = new Handler.Callback() {

    @Override
    public boolean handleMessage(Message msg) {

        long currentTime = System.currentTimeMillis();
        switch( msg.what )
        {
            case MSG_ASYNC_TASK_STARTED:
                mMainText.append("Async task started at " + currentTime + "\n");
                return true;
            case MSG_ASYNC_TASK_COMPLETED:
                mMainText.append("Async task ended at " + currentTime + "\n");
                return true;
            default:
                // The message was not handled, return false
                return false;
        }
    }
};
```

Add a Progress dialog

```
private Thread mWorkerThread = null;  
private Handler mHandler = null;  
private ProgressDialog mProgress = null;
```

```
mHandler = new Handler( mHandlerCallback );  
mProgress = new ProgressDialog(this);
```

```
switch( msg.what )  
{  
case MSG_ASYNC_TASK_STARTED:  
    mMainText.append("Async task started at " + currentTime + "\n");  
    mProgress.setTitle("Running async task");  
    mProgress.setMessage("Wait...");  
    mProgress.show();  
    return true;  
case MSG_ASYNC_TASK_COMPLETED:  
    mMainText.append("Async task ended at " + currentTime + "\n");  
    mProgress.dismiss();  
    return true;  
default:  
    // The message was not handled, return false  
    return false;  
}
```

Summary