CS 696 Emerging Web and Mobile Technologies
Spring Semester, 2011
Doc 17 Android
Mar 17, 2011

Copyright ©, All rights reserved. 2011 SDSU & Roger Whitney, 5500 Campanile Drive, San Diego, CA 92182-7700 USA. OpenContent (http://www.opencontent.org/opl.shtml) license defines the copyright on this document.

References

Android Developer's Guide, http://developer.android.com/guide/index.html

Android

Googles mobile phone OS and SDK

Java only
- Special VM
- Nonstandard byte code

Eclipse is development IDE

Linux

Application framework
2D & 3D graphics
Audio, video and still image support
SQLite database
Embeddable web browser

Hardware dependent

GSM, CDMA
Bluetooth, EDGE, 3G, WIFI
Camera, GPS, compass
accelerometer, NFC

Android SDK

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

See Getting Started at Android Docs

Current version 2.3.3 & 3.0

Supported OS

- Windows XP, Vista
- Mac OS X 10.4.8 or later (intel processor only)
- Linux (Tested on Ubuntu Dapper Drake)

IDE

- Eclipse 3.3 or 3.4
- Java JDK 5 or JDK 6

Android 2.x verses 3.0

2.x for phones

Emulator - 1 minute to start

3.x for tablets

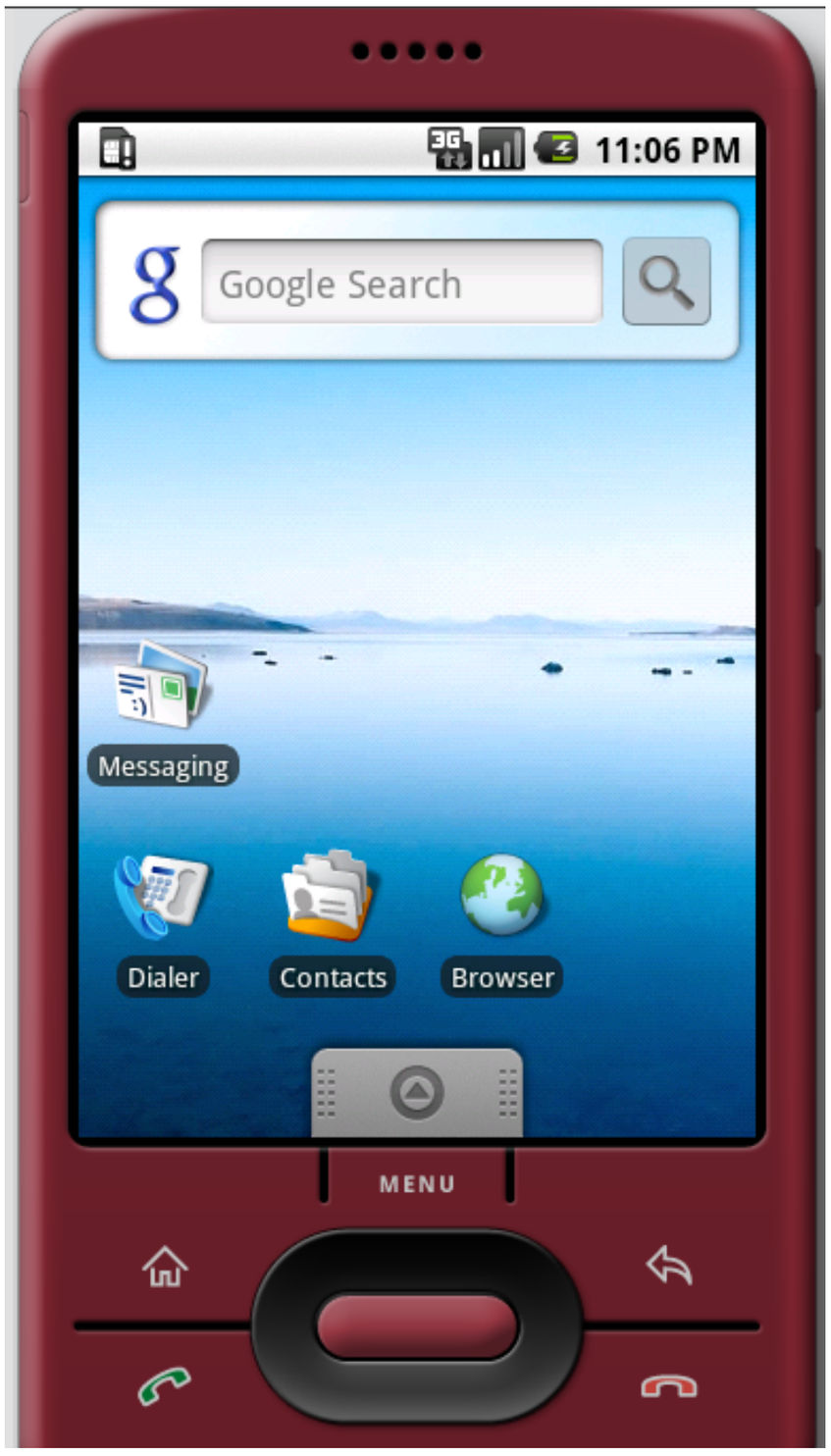
Emutalor - 4 mintues to start

Run on 3.0 devices

Need some care to look reasonable

Can provide several different layouts

Emulators



Very useful in developing applications

Not the same as running on real device
- Emulator has bugs
- Device has different bugs
- Device has restriction and limitations
- Device as resources not on your development machine

Eclipse starts emulator when run Android app
- Can recompile and run app without exiting and restarting emulator

Slow to start up

Hello World Example

Download Android

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

Install Android

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

Follow Hello World Tutorial

http://developer.android.com/resources/tutorials/hello-world.html

Hello World

Auto generated parts of application

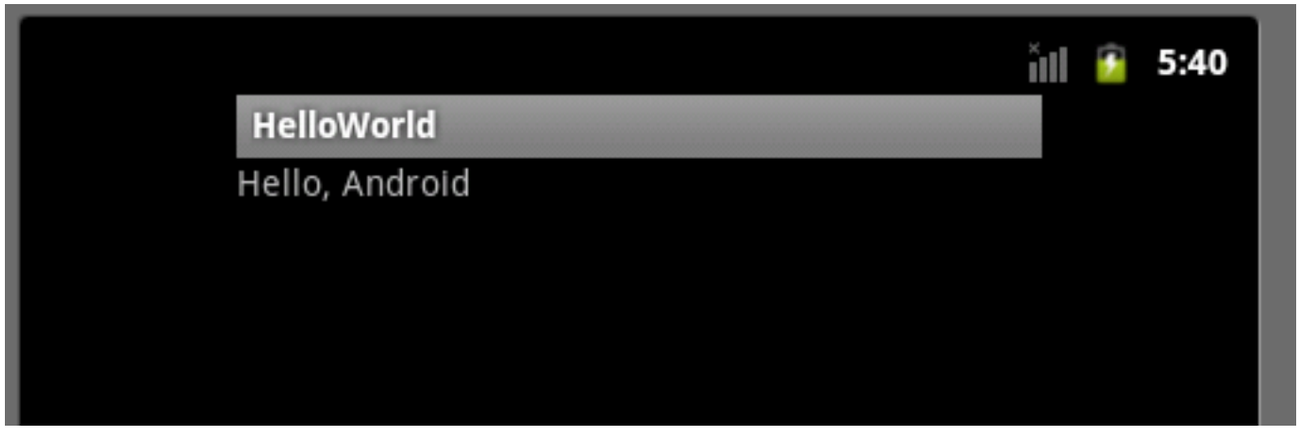
HelloAndroid.java
    Source code

R.java
    Provides access to resources

Resources
    icon.png (Application icon)
    main.xml (Optional Layout of application view)
    strings.xml (Allows separation of source code and display text)

AndroidManifest.xml
    Describes application contents

Hello.java




```
package sdsu.cs696;

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);
    }
}
```


Bold text indicates text added or modified from auto-generated code

PrintIn does not work

```
package sdsu.cs696;

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);
        System.out.println("Debug here");
    }
}
```

Use Log

```
package edu.sdsu.cs;

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

public class HelloWolrd extends Activity {
      public void onCreate(Bundle savedInstanceState) {
            super.onCreate(savedInstanceState);
            TextView tv = new TextView(this);
            tv.setText("Hello, Android");
            setContentView(tv);

            Log.i("Cat", "hello");
      }
}
```

Building Android GUIs

In code

- Instantiate GUI Widgets in code

In XML

- Use GUI builder in Eclipse
- Raw XML in layout.xml

Basic Android Application Parts

Activities
- UI building block
- Views & Activity subclasses

Content Providers
- Shares data between applications

Intents
- System messages

Services
- Long-running nonGUI code

AndroidManifest.xml

R.java

layouts

Fragments
- Sub-activity UI container
- Android 3.0
- Port pre-android 3 coming

Things your program can use

Data Storage
- SQL database

Network Access
- Raw sockets
- Embeddable Web browser

Multimedia
- Sound
- Video

GPS
- Location

Phone services

Basic Android Program Parts

AndroidManifest.xml

Contains information about the application needed by the phone to run it

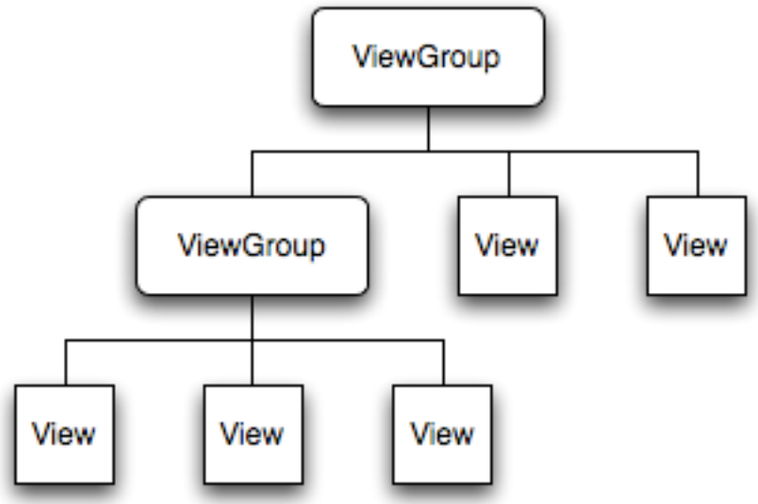
```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
      package="com.android.hello2"
      android:versionCode="1"
      android:versionName="1.0.0">
    <application android:icon="@drawable/icon" android:label="@string/app_name">
        <activity android:name=".HelloAndroid"
                  android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```


@string/app_name indicates that the string is to be found in the string resource file under the label app_name.

Views

View

- Displays content in rectangular area of screen
- Handles
  - Layout, focus, scrolling
  - Keyboard events
  - Gestures



ViewGroups

- Manages set of views and view groups
- Composite pattern

Some Views

AutoCompleteTextView
Button
CheckBox
CheckedTextView
Chronometer
DatePicker
DigitalClock
EditText
ExpandableListView
Gallery
GridView
ImageButton
ListView
MapView,
MultiAutoCompleteTextView
RadioButton
RatingBar
ScrollView
SeekBar
Spinner
TabHost
TabWidget
TableRow
TimePicker
ToggleButton
TwoLineListItem
VideoView
ViewAnimator
WebView
ZoomButton
ZoomControls

Activity

Code that does some work

Single, focused thing that a user can do

Usually each screen(View) has its own activity

An application may have multiple screens, hence multiple activities

An application runs in its own Linux process

Activities can be viewless

Activity Lifecycle States

Active (Resumed)

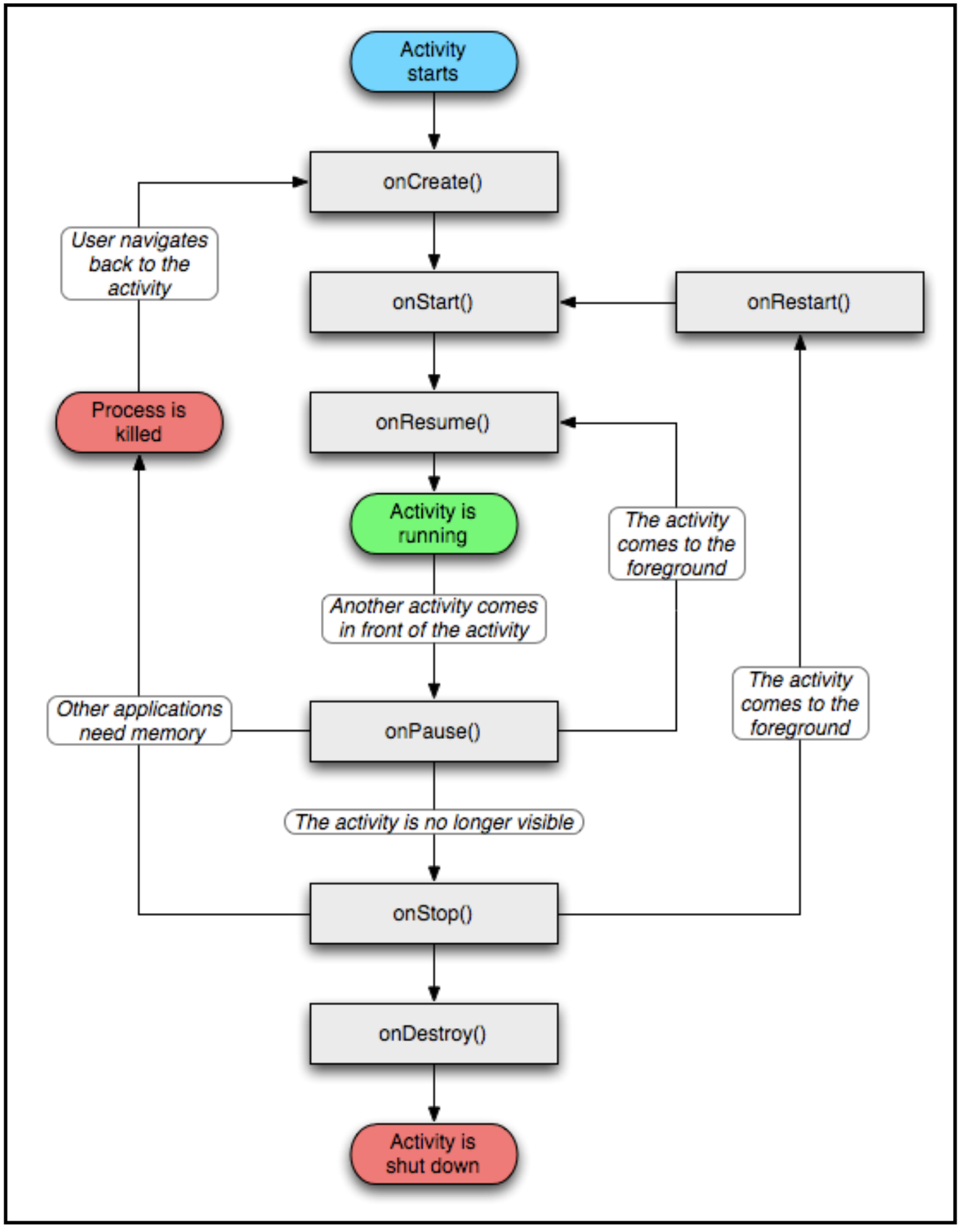
- Running activity in foreground of screen

Paused

- Lost focus, but still visible
- Retains all state information
- In extreme memory situations may be killed

Stopped

- Not visible
- Retains all state information
- Often will be killed


Activity starts
onCreate()
User navigates back to the activity
onStart()
onRestart()
Process is killed
onResume()
Activity is running
The activity comes to the foreground
Another activity comes in front of the activity
The activity comes to the foreground
Other applications need memory
onPause()
The activity is no longer visible
onStop()
onDestroy()
Activity is shut down

LifeCycle Methods

```
public class ExampleActivity extends Activity {
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }
    protected void onStart() {
        super.onStart();
    }
    protected void onResume() {
        super.onResume();
    }
    protected void onPause() {
        super.onPause();
    }
    protected void onStop() {
        super.onStop();
    }
    protected void onDestroy() {
        super.onDestroy();
    }
}
```

Examples

Activity Example

```
package edu.sdsu.cs683;

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

public class CountStates extends Activity {
        int paused = 0;
        int killed = 0;
        int stopped = 0;
        TextView text;
```

Activity Example

```
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    if (savedInstanceState != null) {
        paused = savedInstanceState.getInt("paused");
        killed = savedInstanceState.getInt("killed");
        stopped = savedInstanceState.getInt("stopped");
    }
    text = new TextView(this);
    text.setText("Paused: " + paused + " stopped: " + stopped + " killed "
            + killed);
    setContentView(text);
}
```

Activity Example

```
protected void onResume() {
        super.onResume();
        text.setText("Paused: " + paused + " stopped: " + stopped + " killed "
                + killed);
    }

protected void onStart() {
        super.onStart();
        text.setText("Paused: " + paused + " stopped: " + stopped + " killed "
                + killed);
    }

protected void onStop() {
        stopped++;
        super.onStop();
    }
```

Activity Example

```
protected void onPause() {
        paused++;
        super.onPause();
}

protected void onDestroy() {
        killed++;
        super.onDestroy();
}

protected void onSaveInstanceState(Bundle outState) {
        outState.putInt("paused", paused);
        outState.putInt("killed", killed);
        outState.putInt("stopped", stopped);
}

}
```

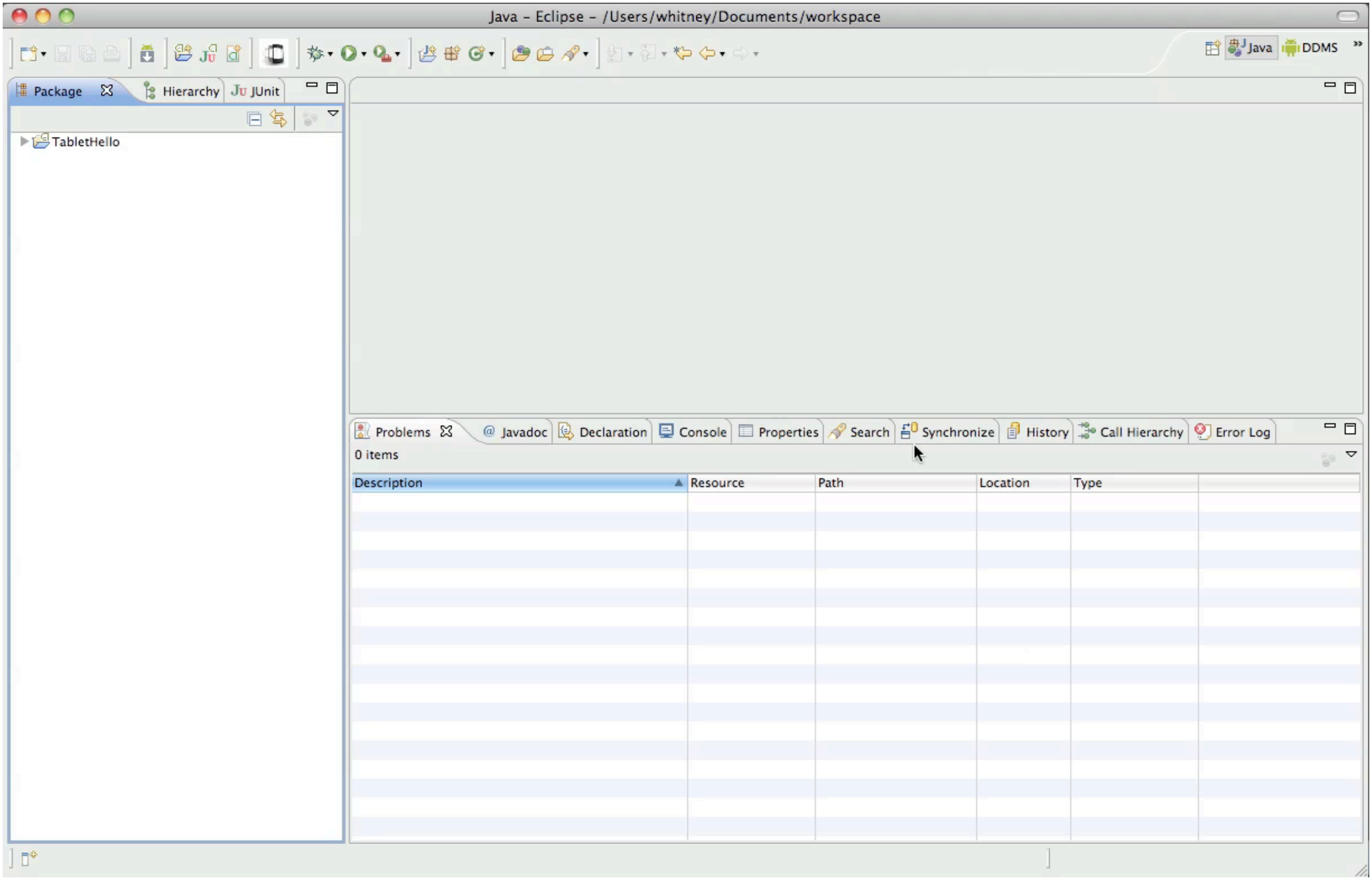
GUI Builder & Layouts

Can build GUIs

In Java code - instantiate Java widget objects

In GUI builder

GUI Demo


Java – Eclipse – /Users/whitney/Documents/workspace
Java
DDMS
Package
Hierarchy
JUnit
TabletHello
Problems
Javadoc
Declaration
Console
Properties
Search
Synchronize
History
Call Hierarchy
Error Log
0 items
Description
Resource
Path
Location
Type

Activity

```
package edu.sdsu.cs.whitney;

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

public class HelloWorld extends Activity {
     @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }
}
```

res/layout/main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="@string/hello"
    />
<AutoCompleteTextView android:text="AutoCompleteTextView"
      android:id="@+id/autoCompleteTextView1"
      android:layout_height="wrap_content"
      android:layout_width="match_parent" android:layout_marginLeft="10dip"
      android:layout_marginRight="10dip" android:layout_marginTop="10dip"></
      AutoCompleteTextView>
<AnalogClock android:layout_gravity="center" android:layout_width="wrap_content"
android:id="@+id/analogClock1" android:layout_height="wrap_content"></AnalogClock>
</LinearLayout>
```

res/values/strings.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="hello">Hello World, HelloWorld!</string>
    <string name="app_name">HelloWorld</string>
</resources>
```

gen/edu.sdsu.cs.whitney/R.java

```
/* AUTO-GENERATED FILE.  DO NOT MODIFY. */
package edu.sdsu.cs.whitney;

public final class R {
    public static final class attr { }
    public static final class drawable {
        public static final int icon=0x7f020000;
    }
    public static final class id {
        public static final int analogClock1=0x7f050001;
        public static final int autoCompleteTextView1=0x7f050000;
    }
    public static final class layout {
        public static final int main=0x7f030000;
    }
    public static final class string {
        public static final int app_name=0x7f040001;
        public static final int hello=0x7f040000;
    }
}
```