

CS 683 Emerging Technologies  
Fall Semester, 2008  
Doc 17 Android 2  
Nov 18 2008

Copyright ©, All rights reserved. 2008 SDSU & Roger Whitney, 5500  
Campanile Drive, San Diego, CA 92182-7700 USA. OpenContent ([http://  
www.opencontent.org/openpub/](http://www.opencontent.org/openpub/)) license defines the copyright on this  
document.

## References

Google Android Documentation, <http://code.google.com/android/documentation.html>

Other references given per slide

# Building Blocks Of Android Application

AndroidManifest.xml

Activities

Views

Intents

Services

Notifications

Content Providers

# 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>
```

# Activities

Code that does some work

Has a life cycle

Paused, Killed, etc.

Views have an associated activity

Activities can be viewless

# View

An object that knows how to draw itself to the screen

Set of existing views or widgets

Can create your own view

Games

New widgets

# Intent

Used to move from screen to screen

Contains

- Data

- Action

  - What you want done

  - MAIN, VIEW, PICK, EDIT, etc

Intent filter

- What intents an activity can handle

To move to a new screen register an intent  
`startActivity(anIntent)`.

# Service

Code that runs in the background

Can be long lived

No UI

Example

music player



# Notifications

Icon that appears in the status bar

Used to notify user

SMS

Voicemail

# Content Holder

Set of methods to let applications access content provider data

Used to allow multiple applications access data

Address book

# XML Views Example

```
package com.android.hello2;

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

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

# src/com.android.hello2/R.java

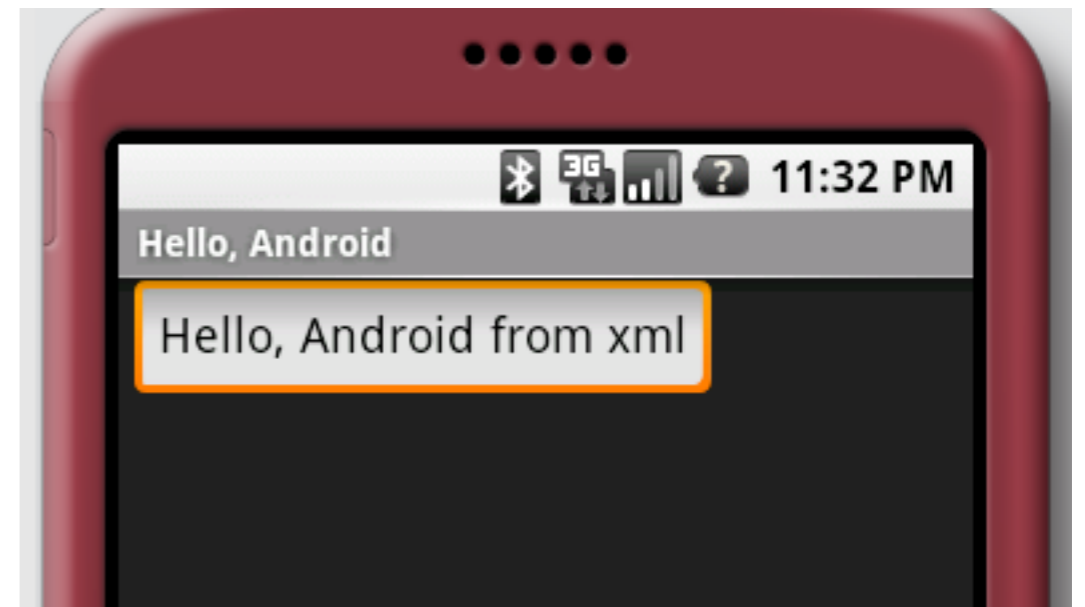
```
package com.android.hello2;
```

```
//Autogenerated
```

```
public final class R {  
    public static final class attr {  
    }  
    public static final class drawable {  
        public static final int icon=0x7f020000;  
    }  
    public static final class layout {  
        public static final int main=0x7f030000;  
    }  
    public static final class string {  
        public static final int Foo=0x7f040002;  
        public static final int app_name=0x7f040001;  
        public static final int hello=0x7f040000;  
    }  
}
```

# res/layout/main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:paddingLeft="6dip"
    android:paddingRight="6dip"
    android:paddingBottom="3dip">
<EditText
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:text="Hello, Android from xml"/>
</LinearLayout>
```



# Measurement Units

px	pixels
dip	device independent pixels
sp	scaled pixels — best for text size
pt	points
in	inches
mm	millimeters

# XML Attributes

android.R.styleable

Defines all xml attributes

<http://code.google.com/android/reference/android/R.styleable.html>

Widget class docs contain links to classes attributes

# Logging

```
package com.android.hello2;

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

public class HelloAndroid extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        Log.i("test", "me");
    }
}
```



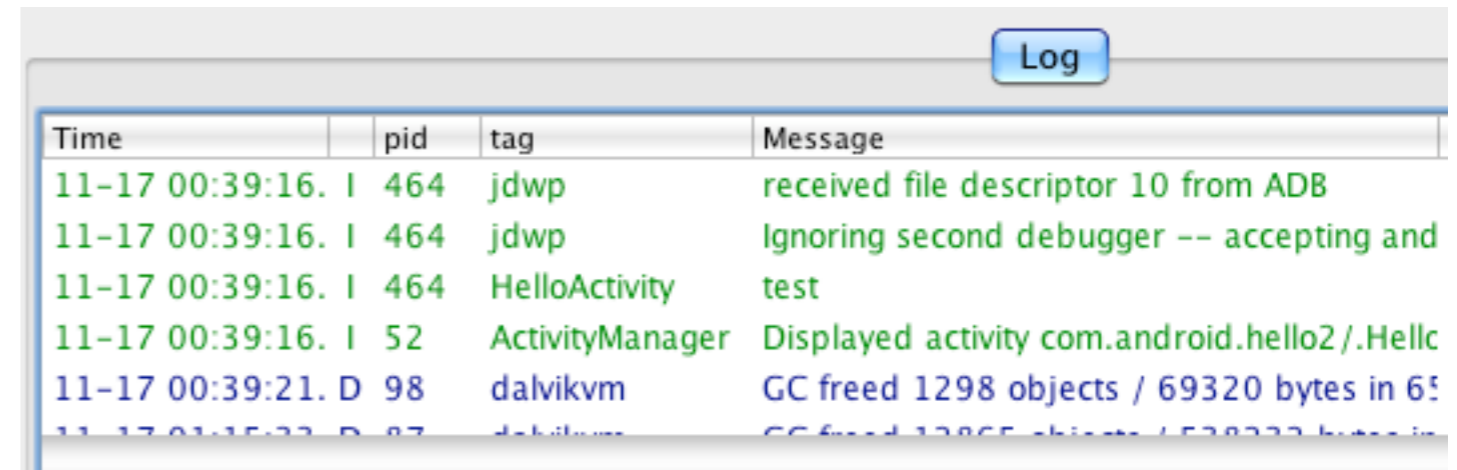
# Viewing the Log

Start the emulator

Start ddms

AI pro 12->ddms

Run your program



The screenshot shows the Logcat window in Android Studio. The window title is "Log". It displays a list of log entries with columns for Time, pid, tag, and Message. The entries are as follows:

Time	pid	tag	Message
11-17 00:39:16. I	464	jdwp	received file descriptor 10 from ADB
11-17 00:39:16. I	464	jdwp	Ignoring second debugger -- accepting and
11-17 00:39:16. I	464	HelloActivity	test
11-17 00:39:16. I	52	ActivityManager	Displayed activity com.android.hello2/.Hello
11-17 00:39:21. D	98	dalvikvm	GC freed 1298 objects / 69320 bytes in 65
11-17 01:15:22. D	87	dalvikvm	GC freed 12865 objects / 528222 bytes in

# Log Methods

## Error

In order of verbosity

e(String tag, String message, Throwable throw)

e(String tag, String message)

## WARN

w(String tag, String message, Throwable throw)

w(String tag, String message)

## INFO

i(String tag, String message, Throwable throw)

i(String tag, String message)

## DEBUG

d(String tag, String message, Throwable throw)

d(String tag, String message)

## VERBOSE (only for development)

v(String tag, String message, Throwable throw)

v(String tag, String message)

# Development Tools

Android Emulator	
Hierarchy ViewerLayout tool	View layouts
Draw 9-Patch	image editor
Dalvik Debug Monitor Service (ddms)	Debug emulator processes
Android Debug Bridge	Used by eclipse
Android Asset Packaging Tool	Used by eclipse
Android Interface Description Language	Passing data between processes
sqlite3	Database
Traceview	
mksdcard	add SD cards to emulator
dx	Converts java to android byte code
UI/Application Exerciser Monkey	Tests applications
activitycreator	Not needed if use eclipse

# Accessing Widgets

```
package com.android.hello2;

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;

public class HelloAndroid extends Activity {
    private EditText messageText;

    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        messageText = (EditText) this.findViewById(R.id.message);
        messageText.setText("From onCreate");
    }
}
```

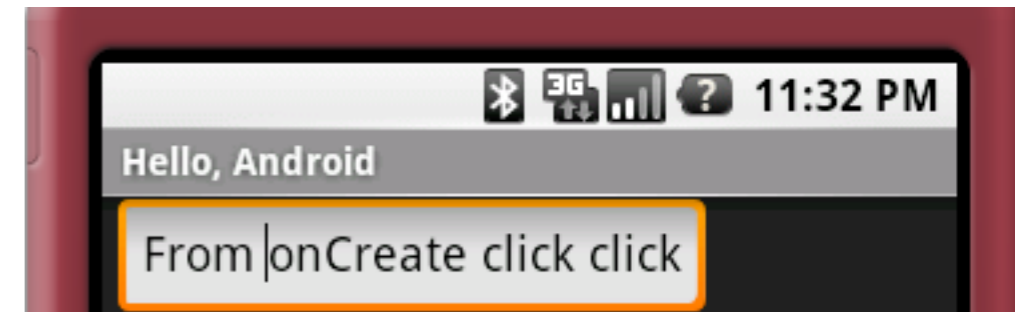
## main.xml with id

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:paddingLeft="6dip"
    android:paddingRight="6dip"
    android:paddingBottom="3dip">
<EditText android:id="@+id/message"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:text="Hello, Android from xml2"/>
</LinearLayout>
```

# See id.message exists now

```
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 message=0x7f050000;  
    }  
    public static final class layout {  
        public static final int main=0x7f030000;  
    }  
    public static final class string {  
        public static final int Foo=0x7f040002;  
        public static final int app_name=0x7f040001;  
        public static final int hello=0x7f040000;  
    }  
}
```

# Click Listener



```
public class HelloAndroid extends Activity implements View.OnClickListener{
    private EditText messageText;

    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        messageText = (EditText) this.findViewById(R.id.message);
        messageText.setText("From onCreate");
        messageText.setOnClickListener(this);
    }

    public void onClick(View v) {

        messageText.setText(messageText.getText() + " click");
    }
}
```

# Listeners

<code>GestureDetector.OnGestureListener</code>	Notify when gestures occur
<code>MenuItem.OnMenuItemClickListener</code>	a menu item is clicked.
<code>View.OnClickListener</code>	a view is clicked.
<code>View.OnCreateContextMenuListener</code>	the context menu for this view is being built.
<code>View.OnFocusChangeListener</code>	the focus state of a view changed.
<code>View.OnKeyListener</code>	a key event is dispatched to this view.
<code>View.OnLongClickListener</code>	a view has been clicked and held.
<code>View.OnTouchListener</code>	a touch event is dispatched to this view.
<code>ViewGroup.OnHierarchyChangeListener</code>	the hierarchy within this view changed.
<code>ViewStub.OnInflateListener</code>	ViewStub has successfully inflated its layout resource.
<code>ViewTreeObserver.OnGlobalFocusChangeListener</code>	the focus state within the view tree changes.
<code>ViewTreeObserver.OnGlobalLayoutListener</code>	the global layout state or the visibility of views within the view tree changes.
<code>ViewTreeObserver.OnPreDrawListener</code>	the view tree is about to be drawn.
<code>ViewTreeObserver.OnTouchModeChangeListener</code>	the touch mode changes.



# Button

```
import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

public class HelloAndroid extends Activity implements View.OnClickListener{

    private EditText messageText;

    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        messageText = (EditText) this.findViewById(R.id.message);
        Button ok = (Button) findViewById(R.id.ok);
        ok.setOnClickListener(this);
    }

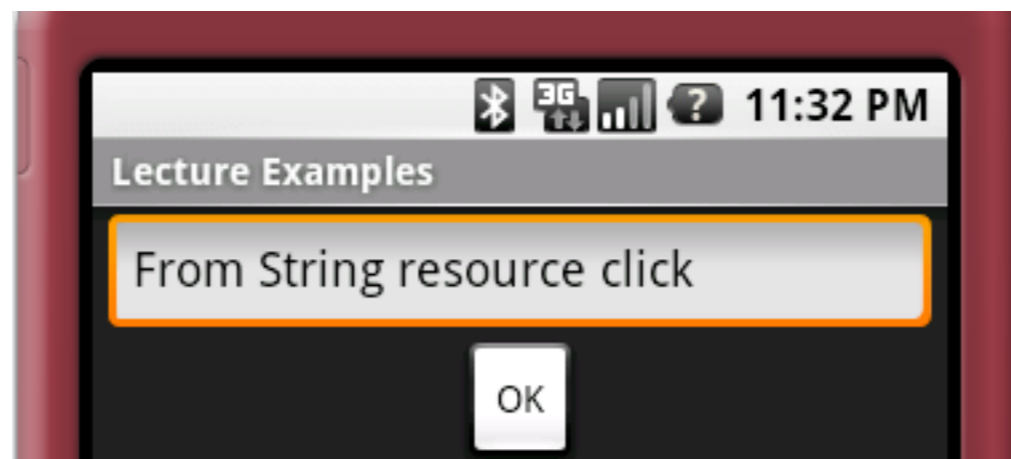
    public void onClick(View v) {
        messageText.setText(messageText.getText() + " click");
    }
}
```

# res/layout/main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:paddingLeft="6dip"
    android:paddingRight="6dip"
    android:paddingBottom="3dip">
    <EditText android:id="@+id/message"
        android:layout_marginTop="2dip"
        android:layout_width="wrap_content"
        android:ems="25"
        android:layout_height="wrap_content"
        android:autoText="true"
        android:capitalize="sentences"
        android:scrollHorizontally="true"
        android:text="@string/hello"/>
    <Button android:id="@+id/ok"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:text="@string/button_ok" />
</LinearLayout>
```

# res/values/strings.xml

```
<?xml version="1.0" encoding="utf-8"?>  
<resources>  
  <string name="hello">From String resource</string>  
  <string name="app_name">Lecture Examples</string>  
  <string name="button_ok">OK</string>  
</resources>
```



# Menus

```
import android.app.Activity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.EditText;

public class HelloAndroid extends Activity {

    private EditText messageText;
    private static final int CLICK_ID = Menu.FIRST;
    private static final int MOM_ID = Menu.FIRST + 1;

    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        messageText = (EditText) this.findViewById(R.id.message);
    }
}
```

## Menus Cont.

```
public boolean onCreateOptionsMenu(Menu menu) {
    super.onCreateOptionsMenu(menu);

    menu.add(0, CLICK_ID, 0, R.string.menu_click).setShortcut('0', 'c');
    menu.add(0, MOM_ID, 0, R.string.menu_mom);
    return true;
}

public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case CLICK_ID:
            messageText.setText(messageText.getText() + " click");
            return true;
        case MOM_ID:
            messageText.setText(messageText.getText() + " Mom");
            return true;
    }
    return super.onOptionsItemSelected(item);
}
```

# Menus Cont.

```
public boolean onPrepareOptionsMenu(Menu menu) {  
    super.onPrepareOptionsMenu(menu);  
}  
}
```

# res/values/strings.xml

```
<?xml version="1.0" encoding="utf-8"?>  
<resources>  
  <string name="hello">From String resource</string>  
  <string name="app_name">Lecture Examples</string>  
  <string name="button_ok">OK</string>  
  
  <string name="menu_click">Click</string>  
  <string name="menu_mom">Hi Mom</string>  
  
</resources>
```

# Display

