

CS 696 Mobile Phone Application Development
Fall Semester, 2009
Doc 3 Application flow, Layouts & Intents
Sept 9, 2009

Copyright ©, All rights reserved. 2009 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

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

Playing with Intents, Gabor Paller, <http://mylifewithandroid.blogspot.com/2007/12/playing-with-intents.html>

DroidDraw, <http://www.droiddraw.org/>

CS 683 Fall 2008 Lecture notes, <http://www.eli.sdsu.edu/courses/fall08/cs683/notes/index.html>

jQuery Touch Demo

<http://www.jqtouch.com/>

jQuery plugin for mobile web development

iPhone

Android

Palm Pre

Beta release

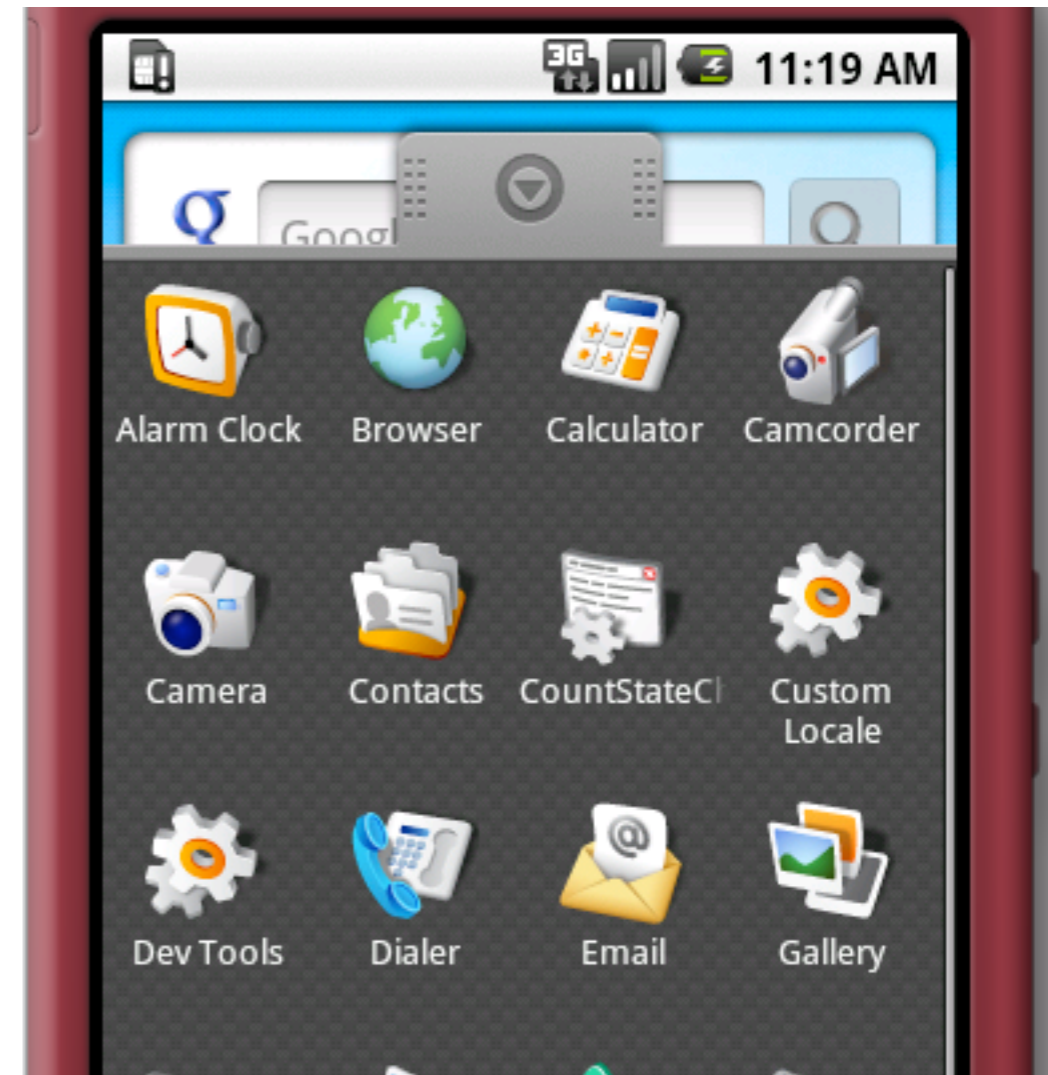
Application Flow

Android Application

Set of related activities

Combined into one application file (.apk)

Launchable from the home screen



Tasks

Sequence of activities the user follows to accomplish an objective

A user can

- Interrupt a task to start a new task

- Resume the first task where they left off

Interrupting a Task

User presses Home and starts an application

Notifications

Activity Stack

Back Stack

History of activities used by user

May include activities of different applications

Back button

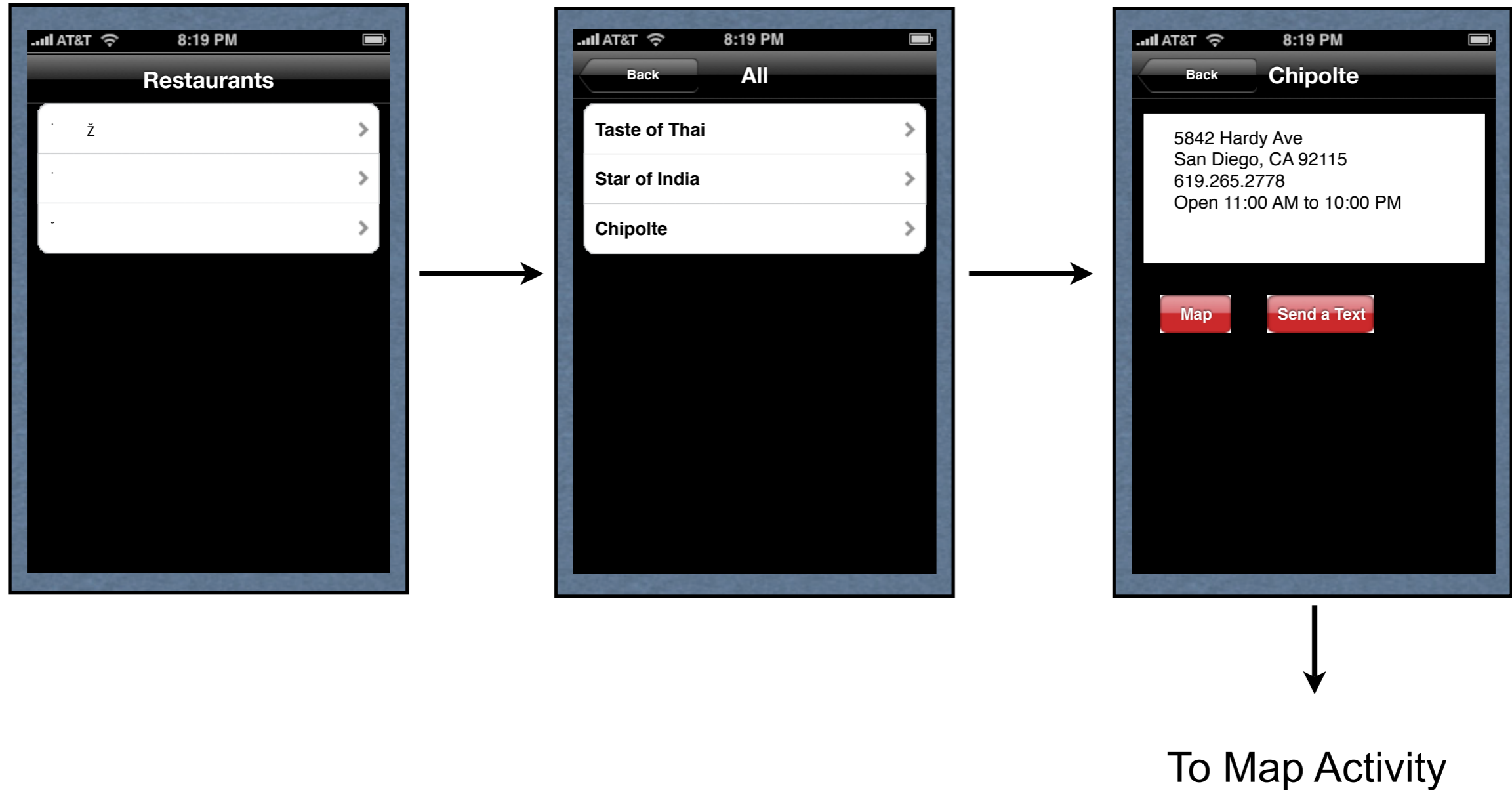
- Removes top of activity stack

- Makes next activity active

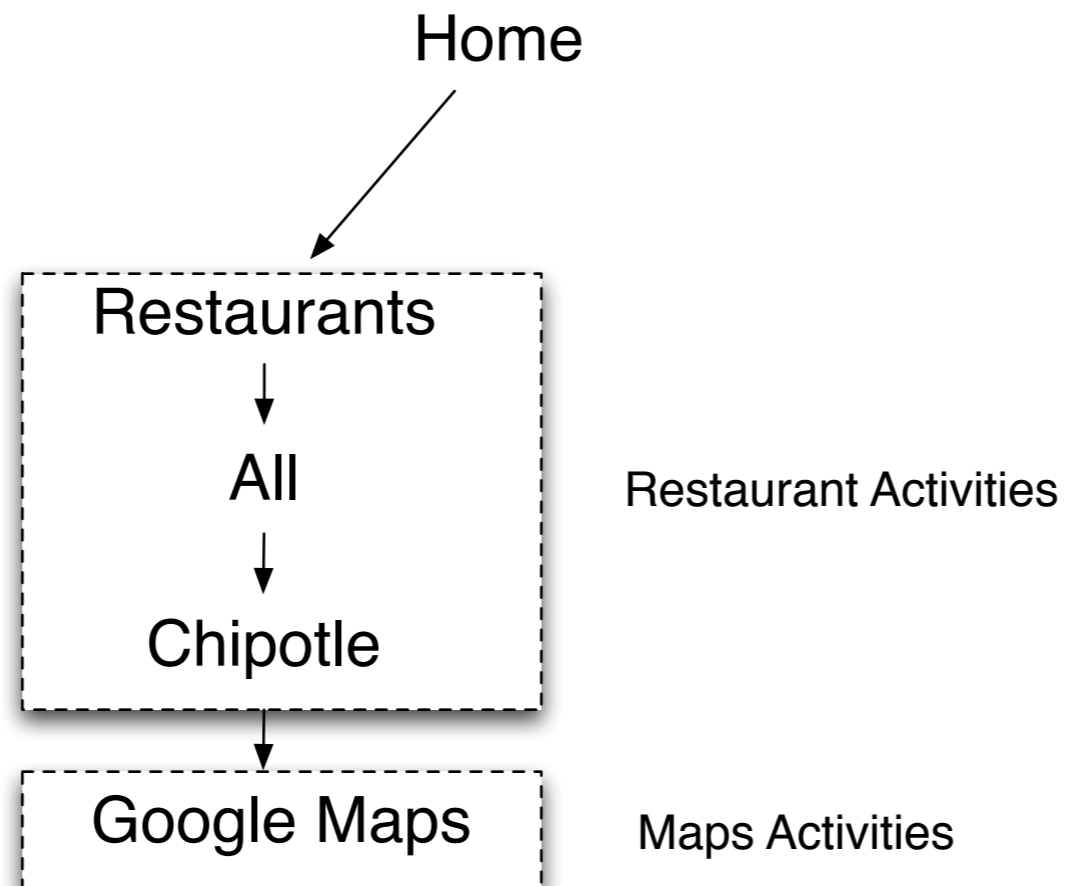
Stack only goes back to the start of the application at Home



Sample User Flow

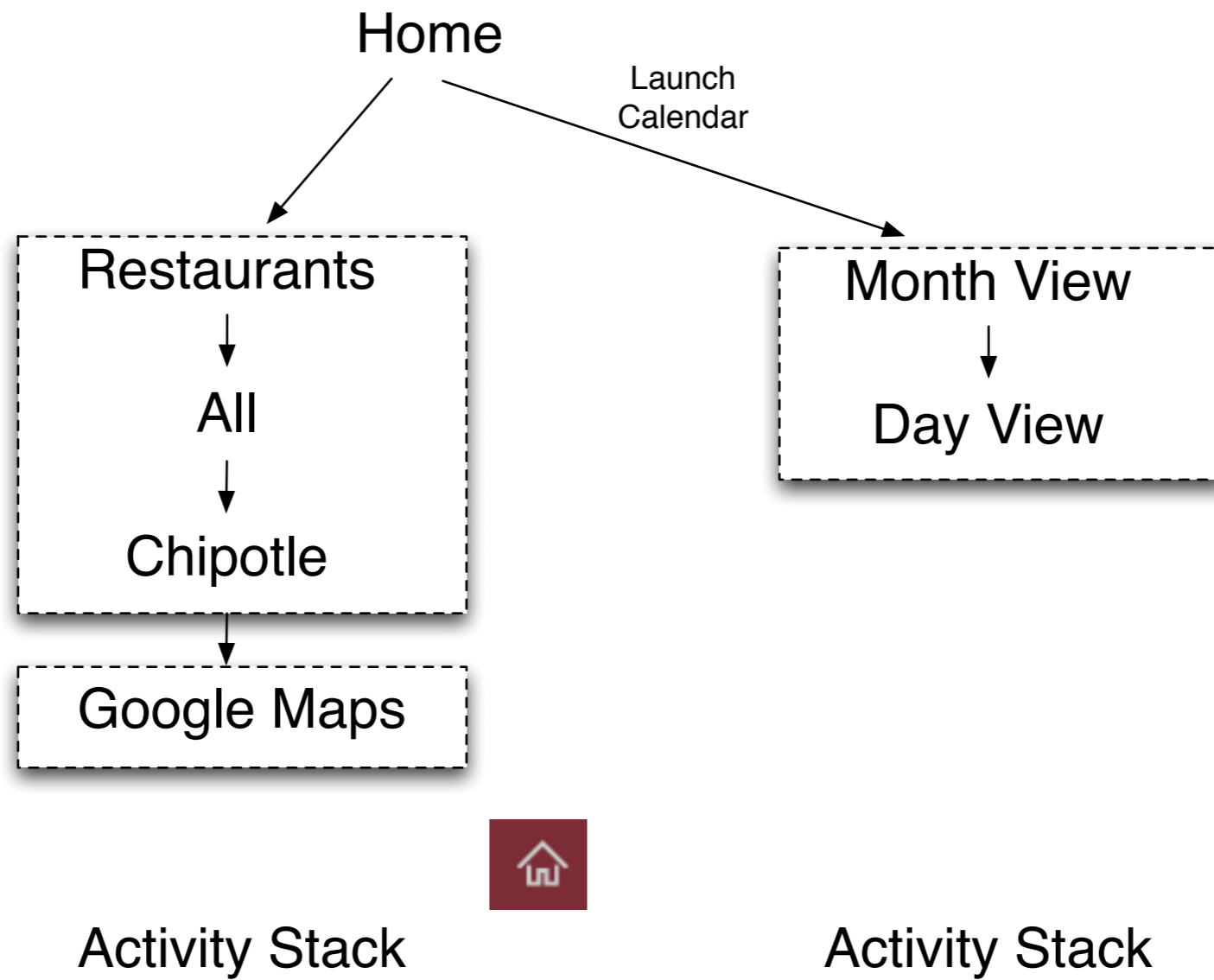


Activity Stack



Activity Stack

Multiple Activity Stacks



Applications & Activity Stacks

Launching a non-running application

- Create new activity stack

- Put application's beginning activity on stack

Launching a running application

- Show activity on top of applications activity stack

- That activity may be from another application

Exceptions

- Some background activities return to their initial screen

 - Contacts & Gallery

- Some activities continue to run while in the background

 - Music player

Layouts

Layouts

FrameLayout

Displays every child in same location

LinearLayout

One child displayed after another

Vertical has one child per row

Respects child's

margins, gravity, weight

TableLayout

Has rows and columns

Each cell can contain a child

AbsoluteLayout

Each child is given absolute x, y location

RelativeLayout

Each child position is given relative to parent or other children

DroidDraw

<http://www.droiddraw.org/>

A graphical GUI builder for Android
Creates xml layout files

Widget guide

<http://www.droiddraw.org/widgetguide.html>

Intents

Intent

Abstract description of an operation to be performed

Methods in android.content.Context	Action
startActivity(Intent) startActivityForResult(Intent, int)	Launch an Activity
sendBroadcast(Intent)	send it to any interested BroadcastReceiver components
startService(Intent) bindService(Intent, ServiceConnection, int)	

Intent Data

Primary Attributes

action

Action to be taken

data

Data to operation on as Uri

Secondary Attributes

category

additional information about the action to execute

type

Mime type of intent data

component

Explicit class to run

extras

Data for other component

<http://code.google.com/android/reference/android/content/Intent.html>

Intent Resolution

Explicit Intents

Specify the component (class) an intent is to run

Implicit Intents

System determines which component to run
Information about the request is given

action, type, scheme, categories

```
<intent-filter>  
  <action android:name="android.intent.action.MAIN" />  
  <category android:name="android.intent.category.LAUNCHER" />  
</intent-filter>
```

Implicit Intents

action

If given, must be listed by the component as one it handles.

String, which we can create

type

Retrieved from the Intent's data, if not already supplied in the Intent.

If given, must be listed by the component as one it handles

data that is not a content: URI and where no explicit type,

The scheme of the intent data (such as http: or mailto:) is considered

If given, must be listed by the component as one it handles

Categories

If given, all must be listed by the component as ones it handles

Intents Handled By Google Android Apps

Scheme	Action	Description
http://web_address https://web_address	VIEW	Open a browser window to the URL specified.
"" (empty string) http://web_address https://web_address	WEB_SEARCH	Opens the file at the location on the device in the browser.
tel: phone_number	CALL	Calls the entered phone number.
tel:phone_number voicemail:	DIAL	Dials but does not actually initiate the call the number given
geo:latitude,longitude geo:latitude,longitude?z=zoom geo:0,0?q=my+street+address geo:0,0?q=business+near+city	VIEW	Opens the Maps application to the given location

Intent Examples

First Intent Example - Dial Phone

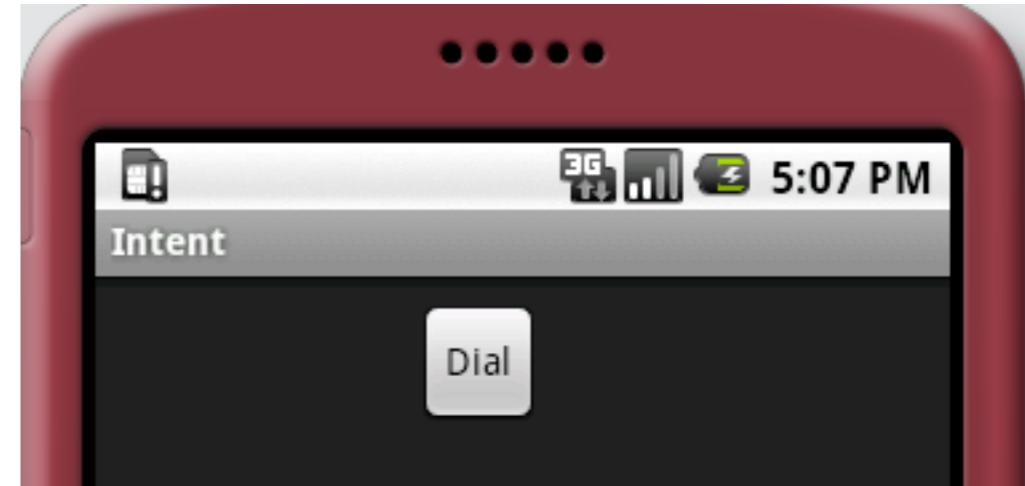
Activity with button

When button is pressed

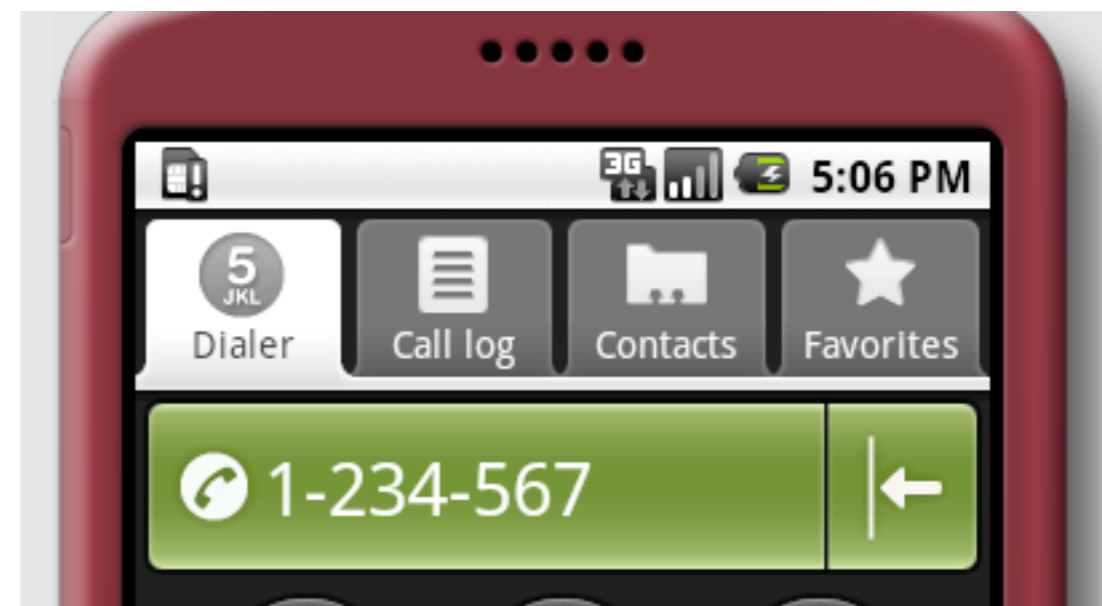
- Phone activity is run

- Phone number is entered

- Phone number is hard coded



Implicit Intent to another application



IntentExample.java

```
public class IntentExample extends Activity implements View.OnClickListener {  
  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.intent);  
        Button ok = (Button) findViewById(R.id.go);  
        ok.setOnClickListener(this);  
    }  
  
    public void onClick(View v) {  
        Intent dial = new Intent();  
        dial.setAction(android.content.Intent.ACTION_DIAL);  
        dial.setData(Uri.parse("tel:1234567"));  
        startActivity(dial);  
    }  
}
```

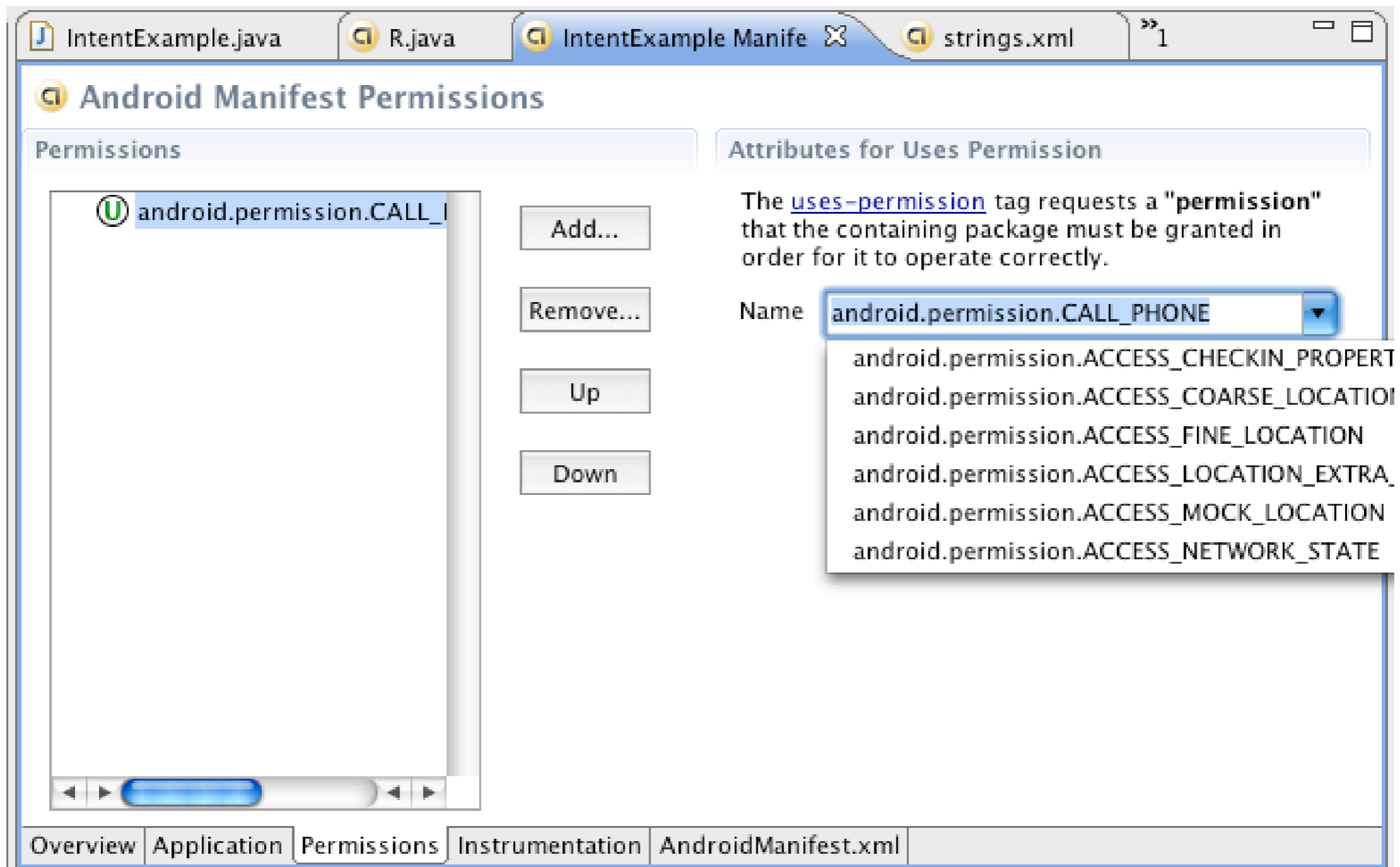

Other Ways to create the Intent

```
public void onClick(View v) {  
    Intent dial = new Intent(Intent.ACTION_DIAL, Uri.parse("tel:1234567"));  
    startActivity(dial);  
}
```

IntentExamples Manifest

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="edu.sdsu.cs696"
    android:versionCode="1"
    android:versionName="1.0">
    <application android:icon="@drawable/icon" android:label="@string/app_name">
        <activity android:name=".IntentExample"
            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>
    <uses-sdk android:minSdkVersion="2" />
    <uses-permission android:name="android.permission.CALL_PHONE"></uses-
permission>
</manifest>
```

Adding the Permission



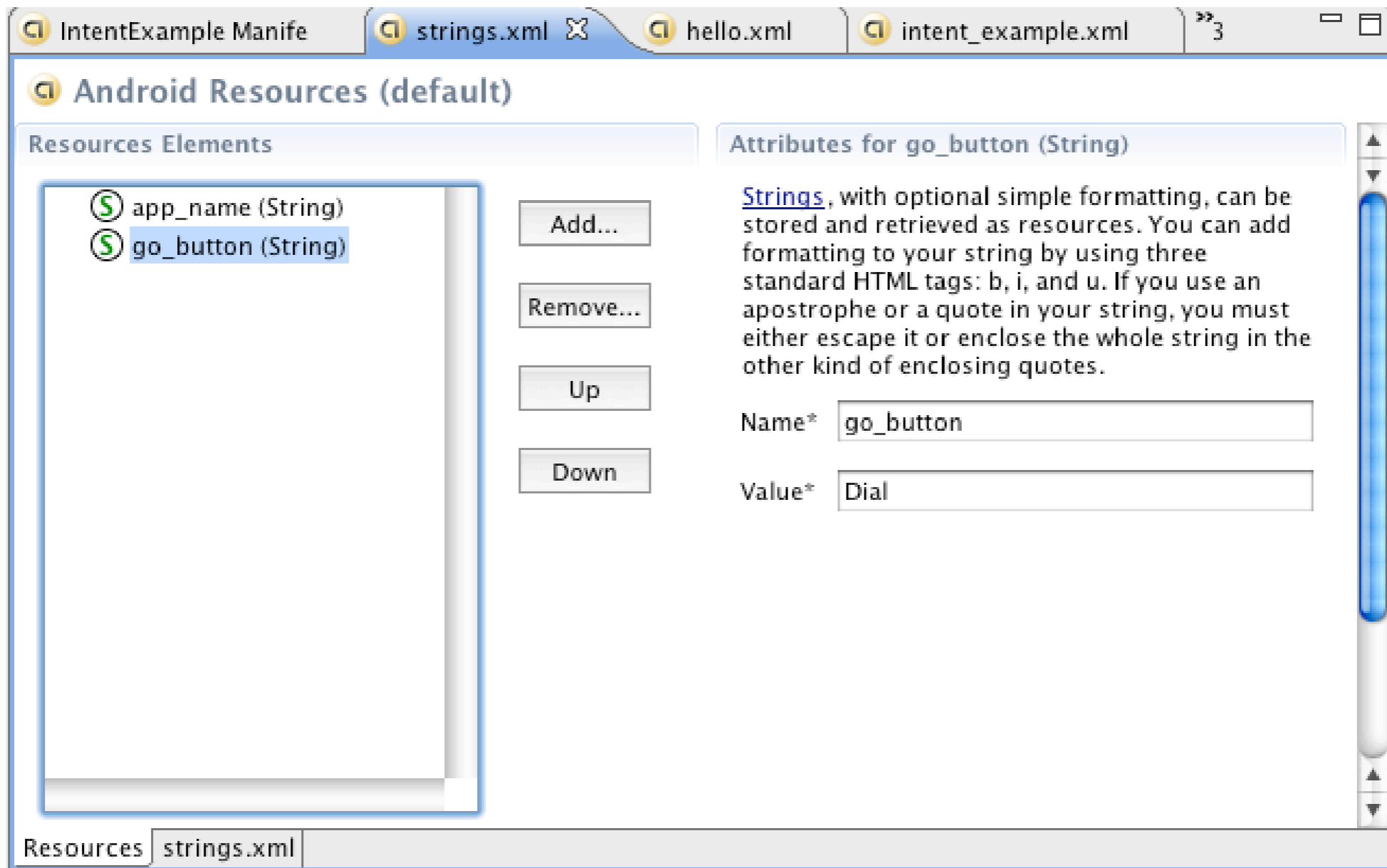
intent.xml

```
<?xml version="1.0" encoding="utf-8"?>
<AbsoluteLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:id="@+id/layout"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
<Button
    android:id="@+id/go"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/go_button"
    android:gravity="center"
    android:layout_x="120px"
    android:layout_y="10px"
    >
</Button>
</AbsoluteLayout>
```

strings.xml

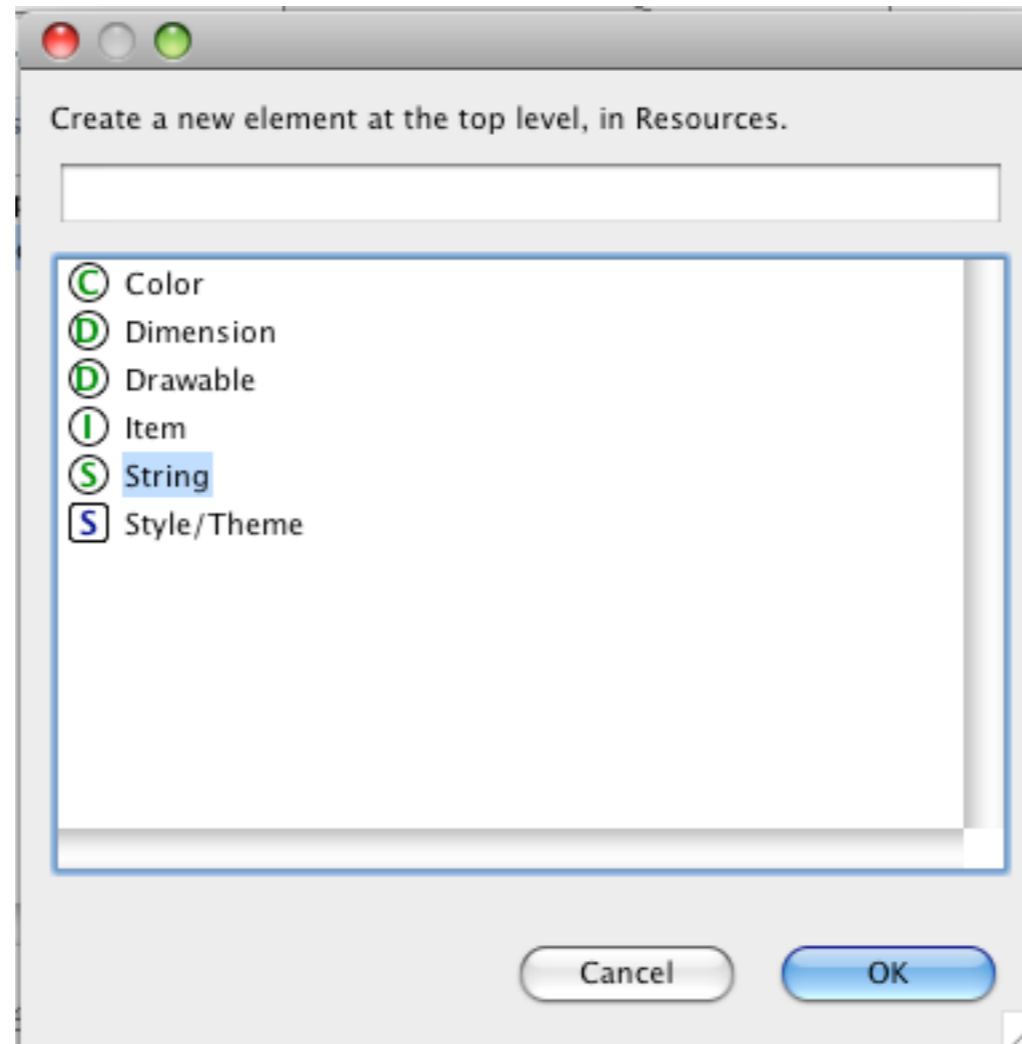
```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <string name="hello">Hello World, IntentExample!</string>
  <string name="app_name">Intent</string>
  <string name="go_button">Dial</string>
</resources>
```

Editing Resources



Add a new String

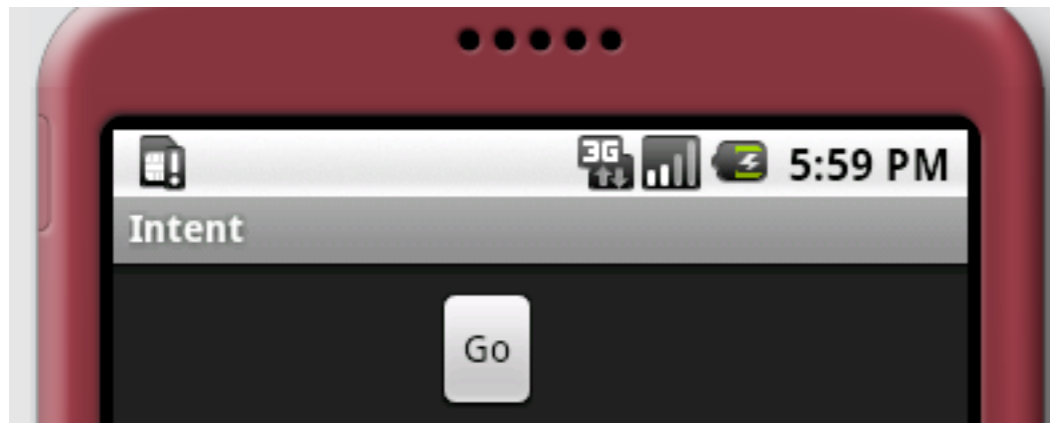
Click on the "Add" button in the Resource view of Strings.xml
(See previous slide)



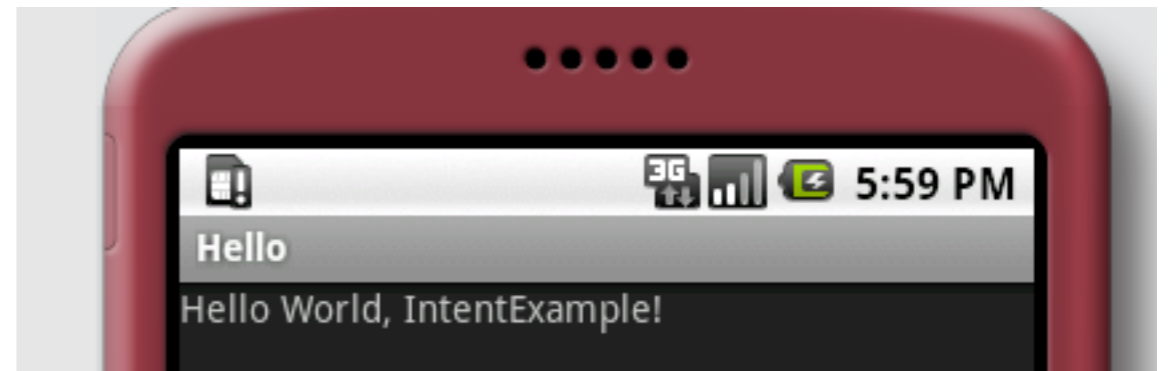
Class Name Intent Example

Button click will call another activity in same application
Will call the class directly
Useful when activity is private to application

IntentExample.java



Hello.java



Clicking the "Go" button will launch the Hello activity

IntentExample.java

```
package edu.sdsu.cs696;

public class IntentExample extends Activity implements View.OnClickListener {

    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.intent);
        Button ok = (Button) findViewById(R.id.go);
        ok.setOnClickListener(this);
    }

    public void onClick(View v) {
        Intent go = new Intent();
        go.setClassName("edu.sdsu.cs696", "edu.sdsu.cs696.Hello");
        startActivity(go);
    }
} //Full class name required
```

Hello.java

```
package edu.sdsu.cs696;

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

public class Hello extends Activity {
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.hello);
    }
}
```

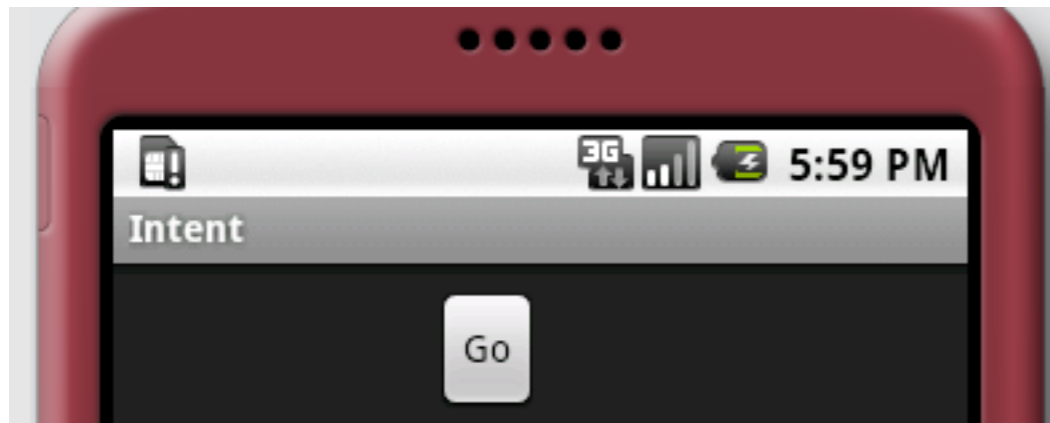
AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="edu.sdsu.cs696"
    android:versionCode="1"
    android:versionName="1.0">
<application android:icon="@drawable/icon" android:label="@string/app_name">
    <activity android:name=".IntentExample"
        android:label="@string/app_name">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
    <activity android:label="Hello" android:name="Hello">
        </activity>
</application>
</manifest>
```

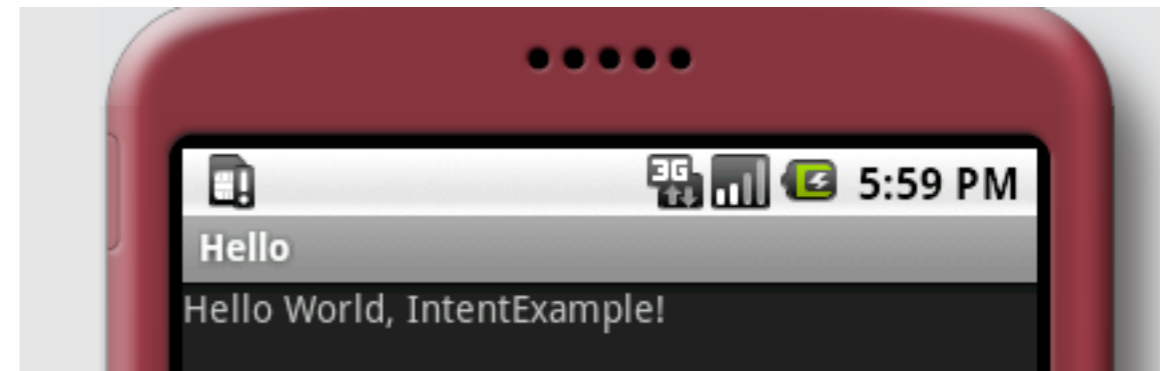
Implicit Example - Intent Filter

Button click will call another component using intent filters
Indirect access of activity

IntentExample.java



Hello.java



Clicking the "Go" button will
launch the Hello activity

IntentExample

```
public class IntentExample extends Activity implements View.OnClickListener {  
  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.intent_example);  
        Button ok = (Button) findViewById(R.id.go);  
        ok.setOnClickListener(this);  
    }  
  
    public void onClick(View v) {  
        Intent go = new Intent();  
        go.setAction("cs696.sender.add");  
        startActivity(go);  
    }  
}
```

Hello.java

```
package edu.sdsu.cs696;

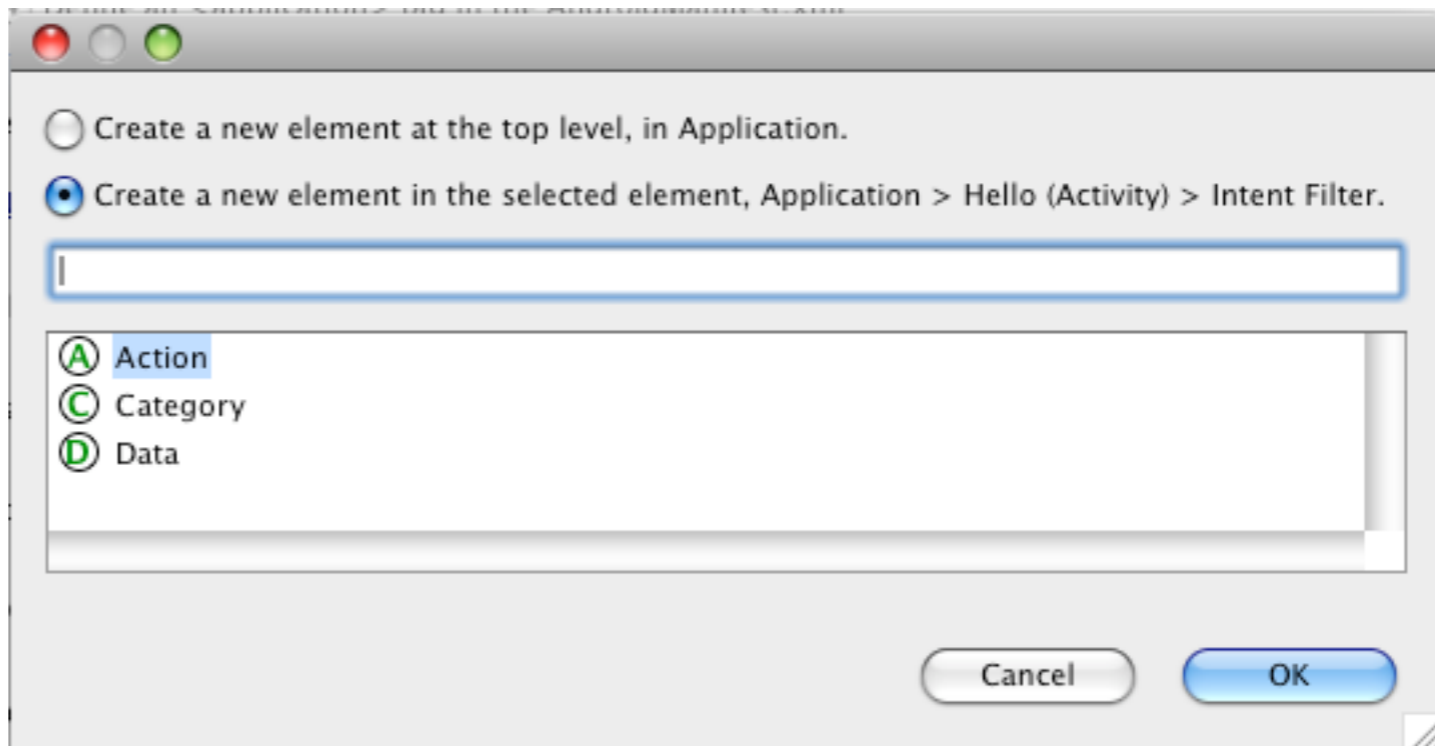
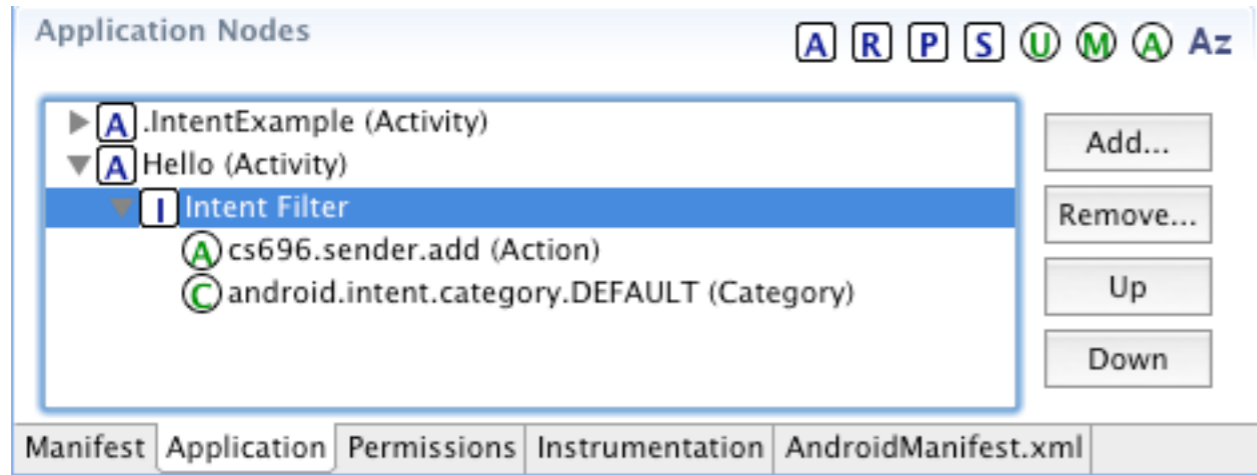
import android.app.Activity;
import android.os.Bundle;

public class Hello extends Activity {
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.hello);
    }
}
```

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="edu.sdsu.cs696"
    android:versionCode="1"
    android:versionName="1.0">
<application android:icon="@drawable/icon" android:label="@string/app_name">
    <activity android:name=".IntentExample"
        android:label="@string/app_name">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
    <activity android:label="Hello" android:name="Hello">
    <intent-filter>
        <action android:name="cs696.sender.add"></action>
        <category android:name="android.intent.category.DEFAULT">
        </category>
    </intent-filter>
    </activity>
</application>
</manifest>
```

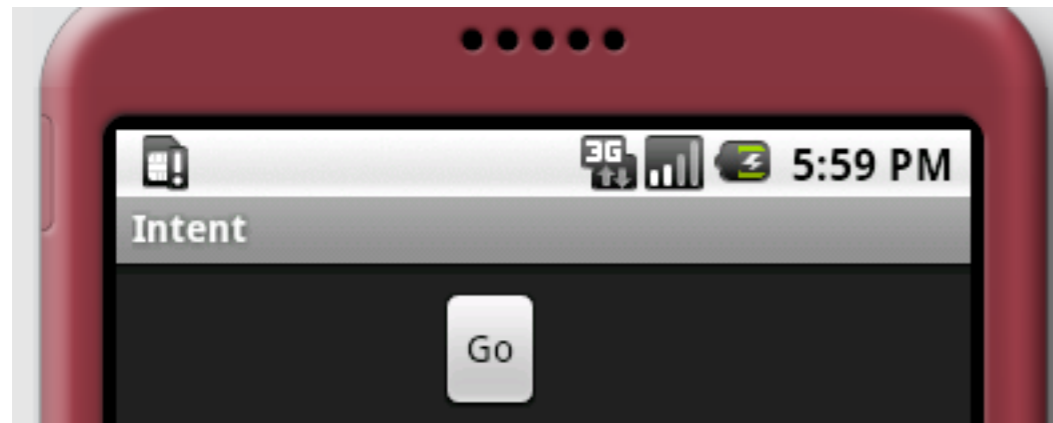
Adding Categories etc



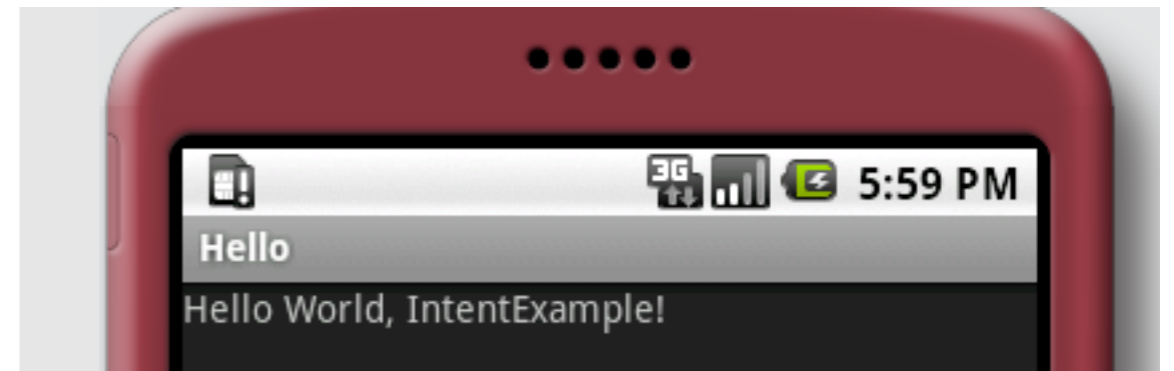
Implicit Example - Filter with Categories

Button click will call another component

IntentExample.java



Hello.java



Clicking the "Go" button will launch the Hello activity

Category Example

```
public class IntentExample extends Activity implements View.OnClickListener {  
  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.intent);  
        Button ok = (Button) findViewById(R.id.go);  
        ok.setOnClickListener(this);  
    }  
  
    public void onClick(View v) {  
        Intent go = new Intent();  
        go.setAction("cs696.sender.add");  
        go.addCategory("foo");  
        startActivity(go);  
    }  
}
```

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="edu.sdsu.cs696"
    android:versionCode="1"
    android:versionName="1.0">
    <application android:icon="@drawable/icon" android:label="@string/app_name">
        <activity android:name=".IntentExample"
            android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:label="Hello" android:name="Hello">
            <intent-filter>
                <action android:name="cs696.sender.add"></action>
                <category android:name="android.intent.category.DEFAULT">
                </category>
                <category android:name="foo"></category>
            </intent-filter>
        </activity>
    </application>
</manifest>
```

Some Application Settings

<u>Name*</u>	<input type="text" value=".IntentExample"/>	<input type="button" value="Browse..."/>
Theme	<input type="text"/>	<input type="button" value="Browse..."/>
Label	<input type="text" value="@string/app_name"/>	<input type="button" value="Browse..."/>
Icon	<input type="text"/>	<input type="button" value="Browse..."/>
Launch mode	<input type="text"/>	<input type="button" value="▼"/>
Screen orientation	<input type="text"/>	<input type="button" value="▼"/>
Config changes	<input type="text"/>	<input type="button" value="Select..."/>
Permission	<input type="text"/>	<input type="button" value="▼"/>
Multiprocess	<input type="text"/>	<input type="button" value="▼"/>
Process	<input type="text"/>	<input type="button" value="Browse..."/>
Task affinity	<input type="text"/>	<input type="button" value="Browse..."/>
Allow task reparenting	<input type="text"/>	<input type="button" value="▼"/>
Finish on task launch	<input type="text"/>	<input type="button" value="▼"/>
Clear task on launch	<input type="text"/>	<input type="button" value="▼"/>
Always retain task state	<input type="text"/>	<input type="button" value="▼"/>
State not needed	<input type="text"/>	<input type="button" value="▼"/>
Exclude from recents	<input type="text"/>	<input type="button" value="▼"/>
Enabled	<input type="text"/>	<input type="button" value="▼"/>