

CS 683 Emerging Technologies
Fall Semester, 2008
Doc 18 Android Intents
Nov 19 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>

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

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

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

DroidDraw

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

A graphical GUI builder for Android

Creates xml layout files

Application is slightly out of date

- Generated wrong property name

- Could not reproduce later

Widget guide

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

Warning

Android 1.0 is new

Changes have been made from earlier versions

Some blog examples may not work with 1.0

Some Android documentation has errors

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

First Intent Example - Dial Phone

Activity with button

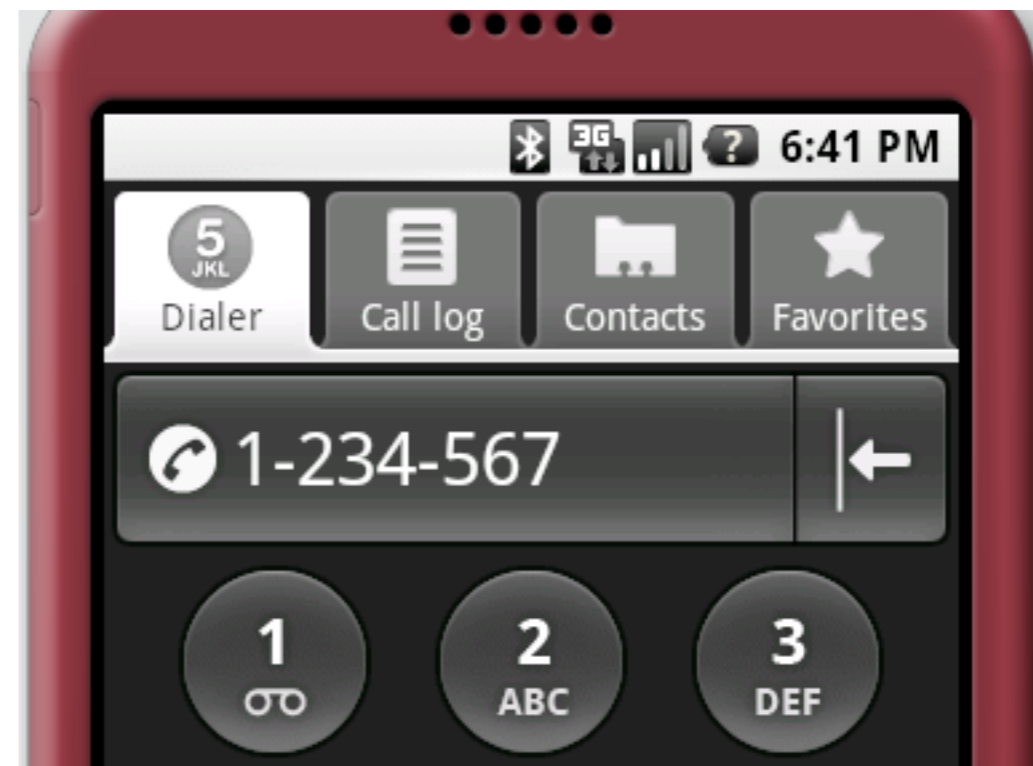
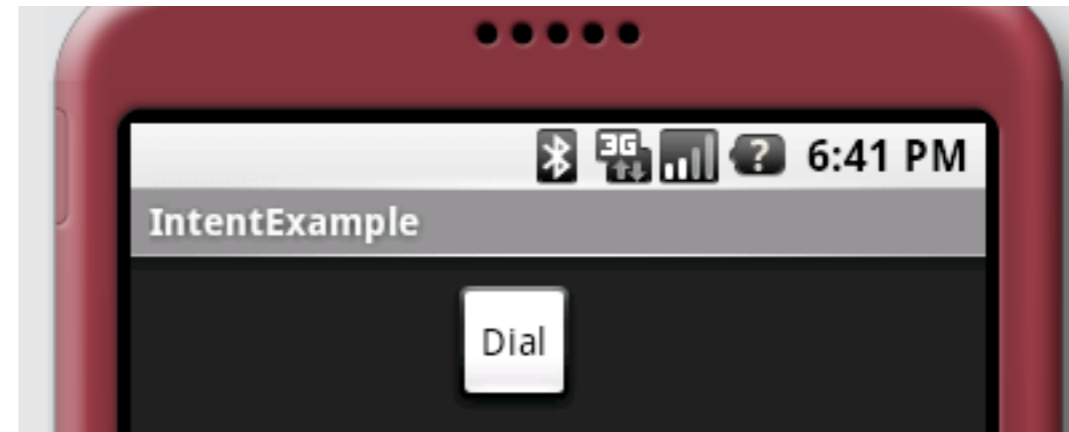
When button is pressed

- 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 final static int ACTIVITY_INVOKE = 0;

    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 dial = new Intent();
        dial.setAction( android.content.Intent.ACTION_DIAL );
        dial.setData(Uri.parse("tel:1234567"));
        startActivity(i);
    }
}
```

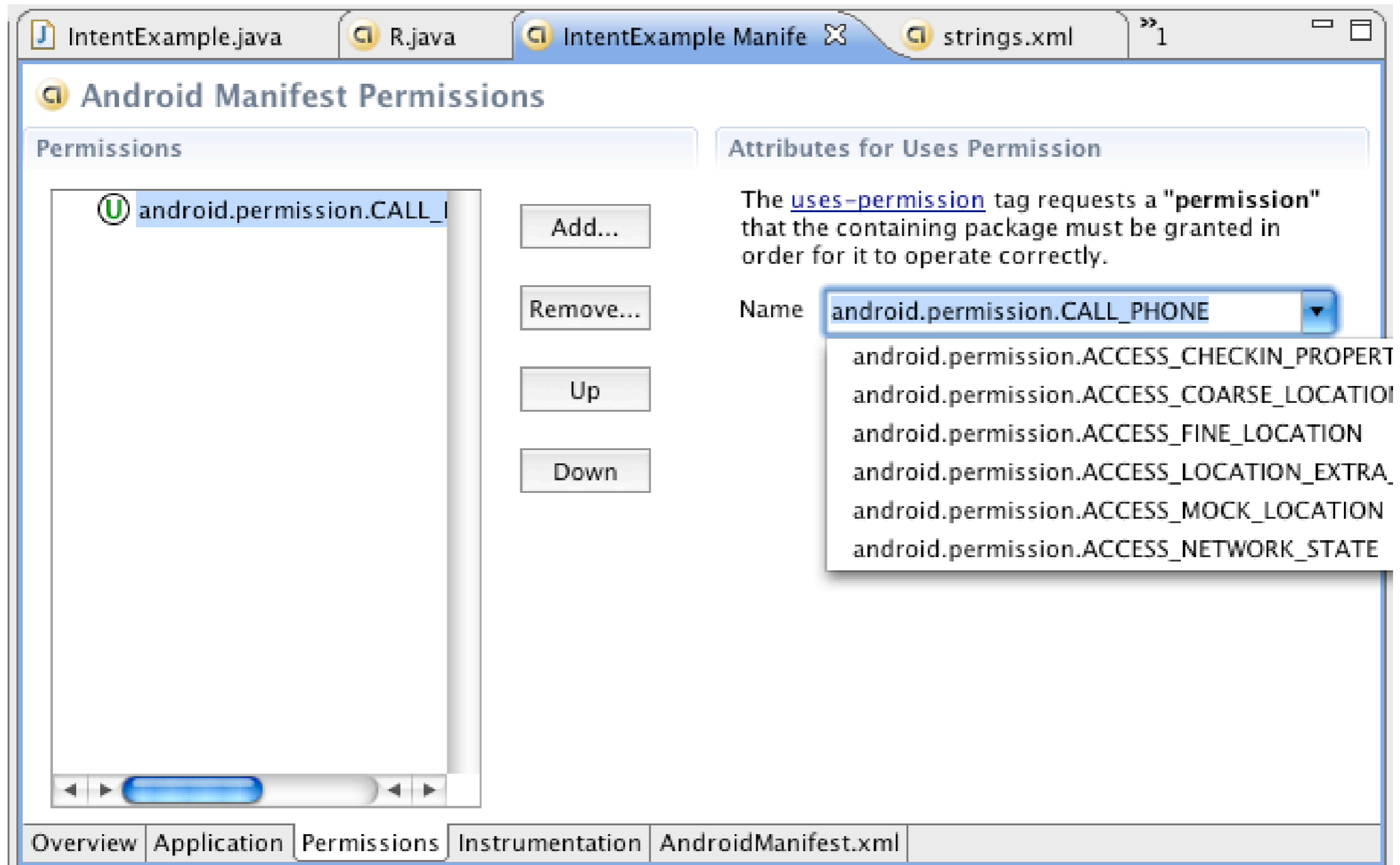
Other Ways to create the Intent

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

IntentExample.Manifest

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="edu.sdsu.cs683.example" android:versionCode="1"
    android:versionName="1.0.0">
    <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-permission android:name="android.permission.CALL_PHONE">
</uses-permission>
</manifest>
```

Adding the Permission



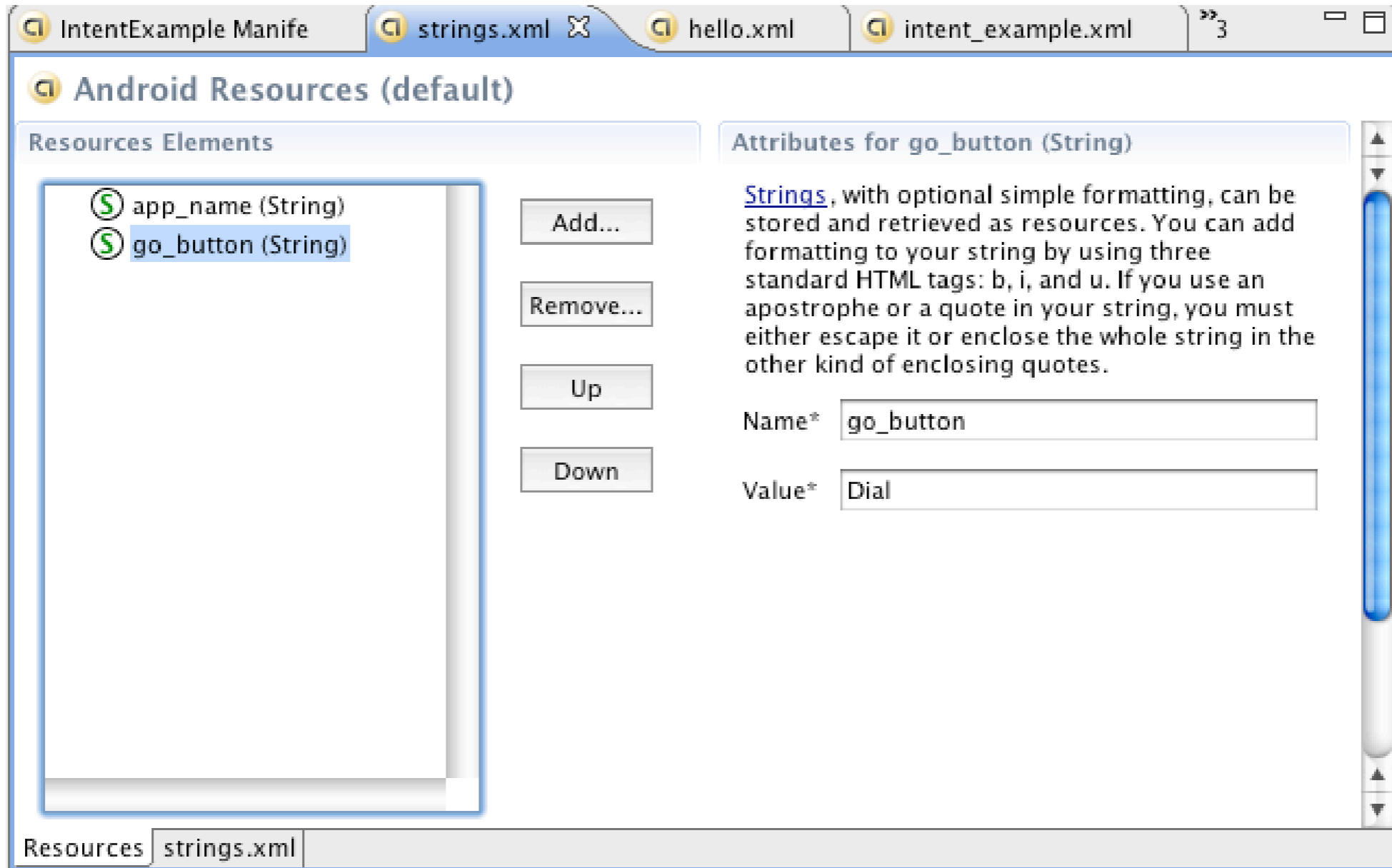
intent_example.xml

```
<?xml version="1.0" encoding="utf-8"?>
<AbsoluteLayout
  android:id="@+id/layout"
  android:layout_width="fill_parent"
  android:layout_height="fill_parent"
  xmlns:android="http://schemas.android.com/apk/res/android"
  >
  <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

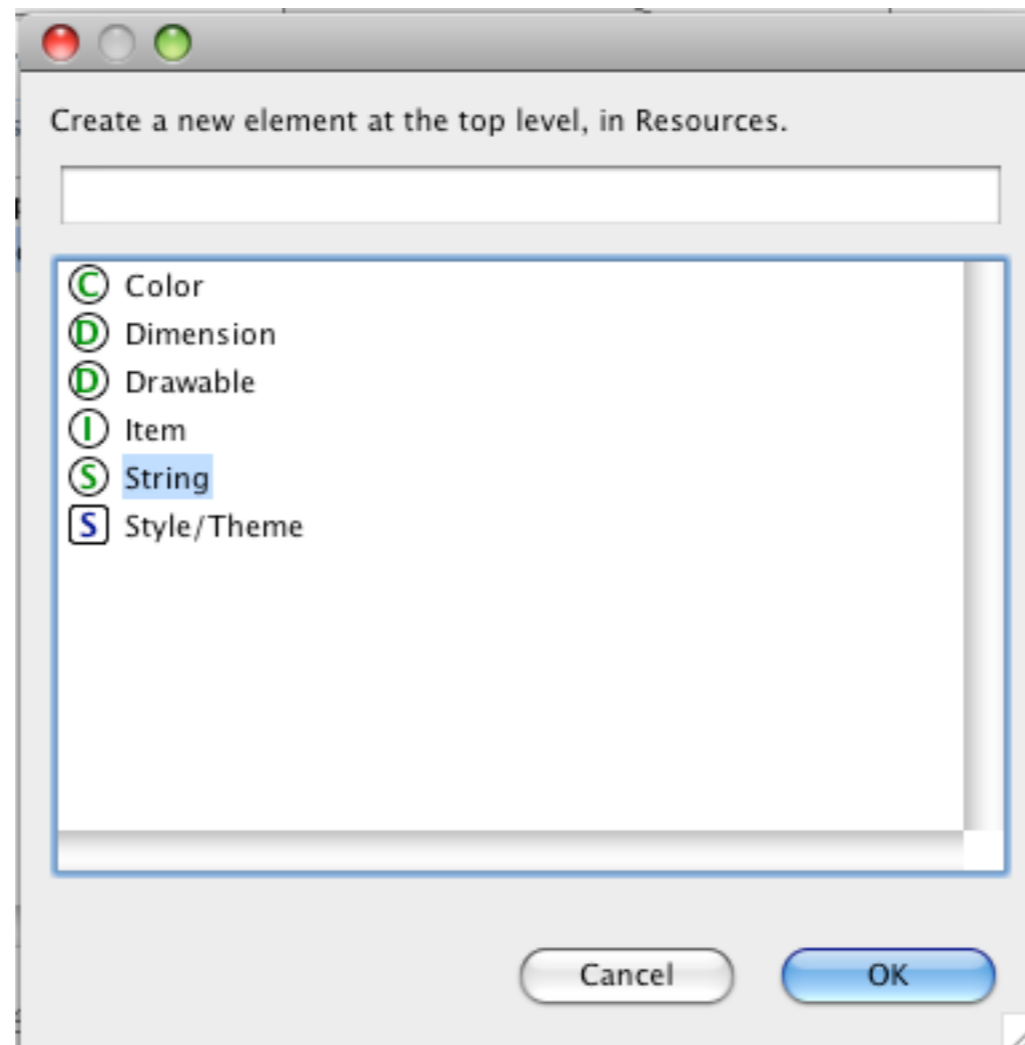
```
<?xml version="1.0" encoding="utf-8"?>  
<resources>  
    <string name="app_name">IntentExample</string>  
<string name="go_button">Dial</string>  
</resources>
```

Editing Resources



Add a new String

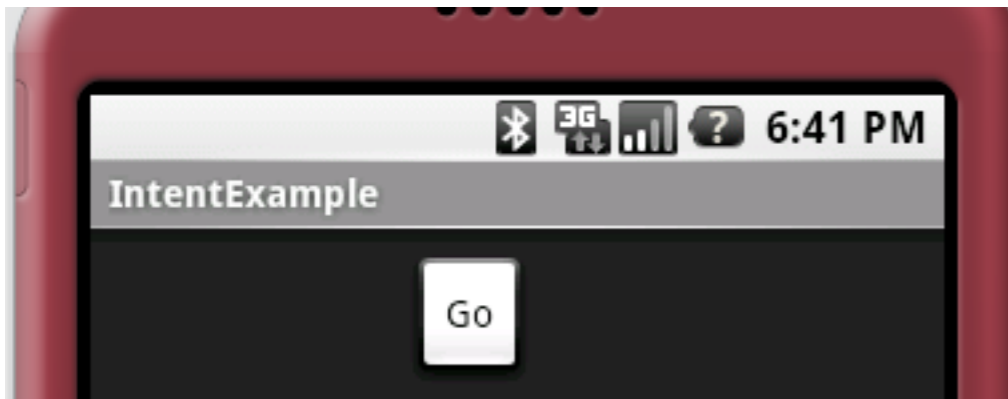
Click on the "Add" button in the Resource view of Strings.xml
(See previous slide)



Explicit Intent Example

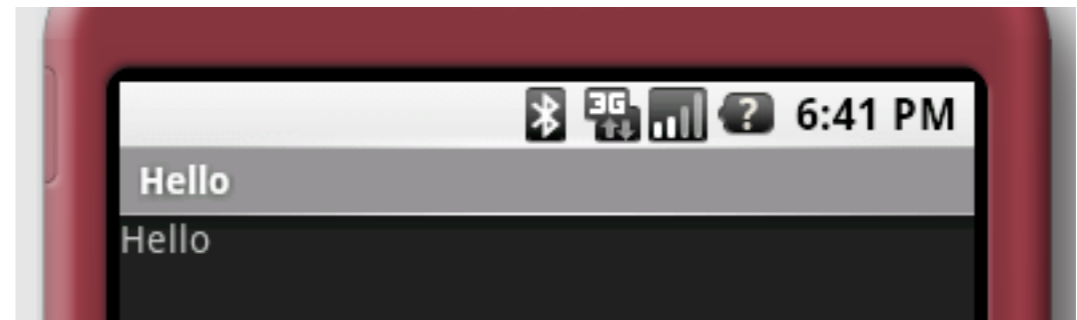
Button click will call another component

IntentExample.java



Clicking the "Go" button will launch the Hello activity

Hello.java



IntentExample.java

```
package edu.sdsu.cs683.example;
```

```
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;  
        go = new Intent();  
        go.setClassName("edu.sdsu.cs683.example",  
"edu.sdsu.cs683.example.Hello");  
        startActivity(go);  
    }
```

//Full class name required

```
}
```

Hello.java

```
package edu.sdsu.cs683.example;

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);
    }
}
```

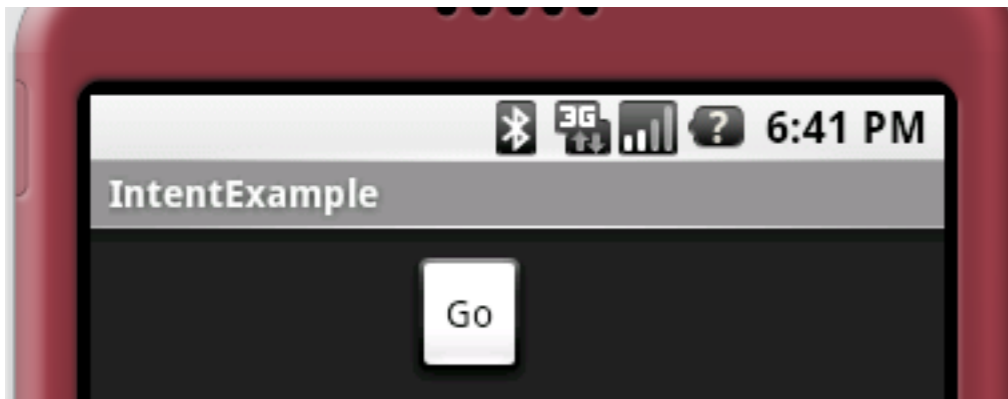
AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="edu.sdsu.cs683.example" android:versionCode="1"
    android:versionName="1.0.0">
    <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 - Sender & Receiver

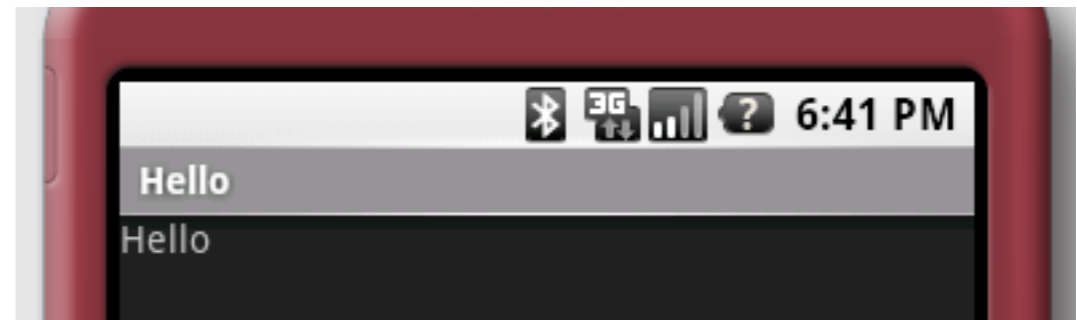
Button click will call another component

IntentExample.java



Clicking the "Go" button will launch the Hello activity

Hello.java



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("cs683.sender.add");
        startActivity(go);
    }
}
```

Hello.java

```
package edu.sdsu.cs683.example;

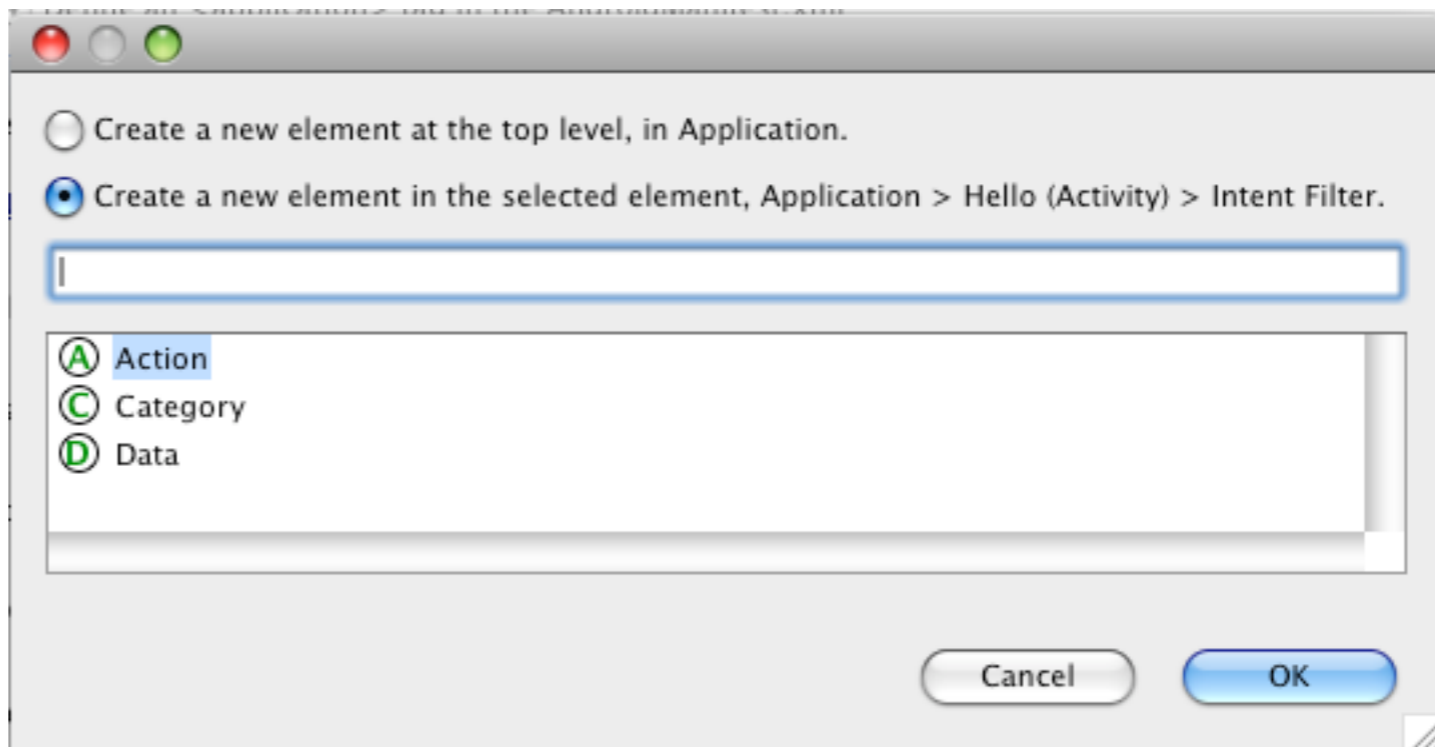
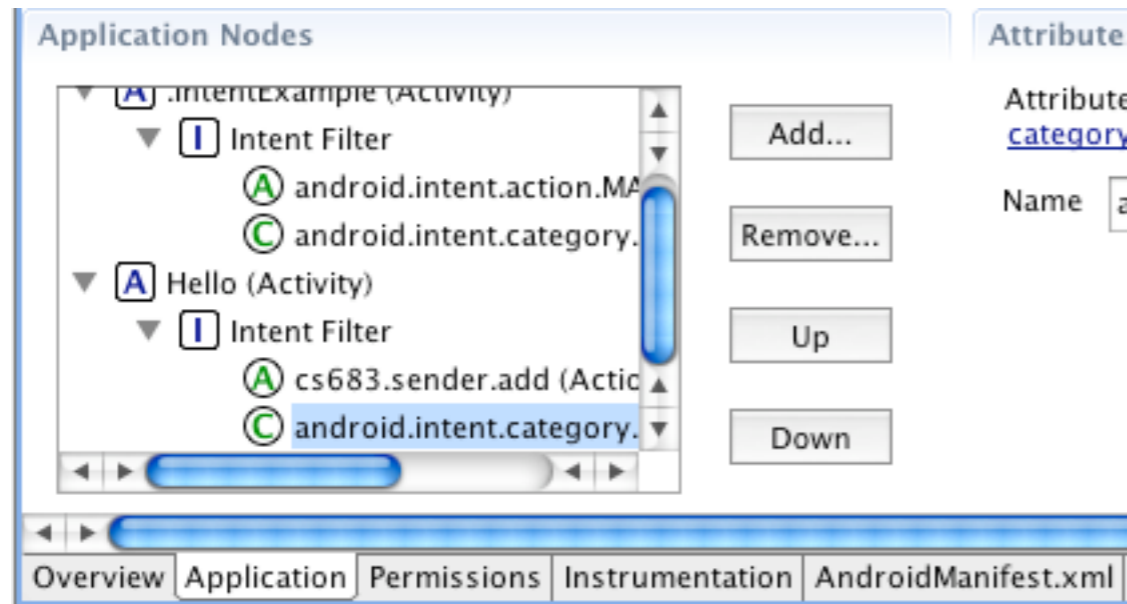
import android.app.Activity;
import android.os.Bundle;

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.cs683.example" android:versionCode="1"
    android:versionName="1.0.0">
    <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="cs683.sender.add"></action>
                <category android:name="android.intent.category.DEFAULT">
            </category>
            </intent-filter>
        </activity>
    </application>
</manifest>
```

Adding Categories etc

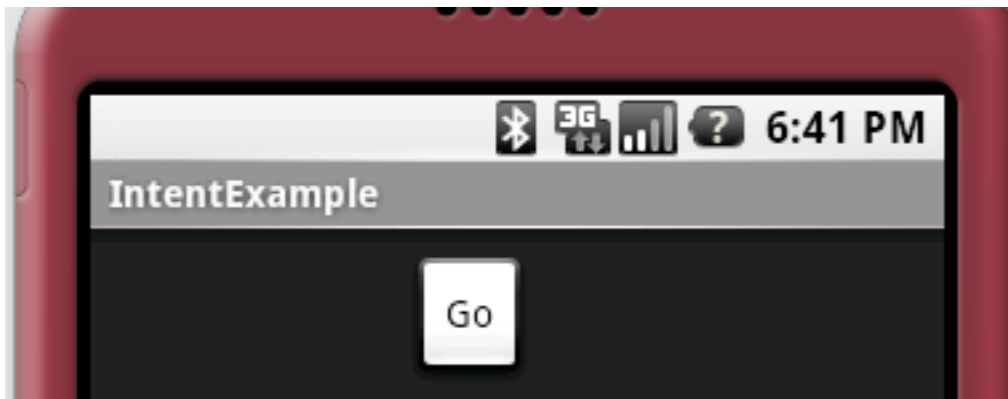


This GUI interface to editing the AndroidManifest.xml file does not always work. A few times it would not allow me to add the name of a category. It added the category, but would not show the category in the graphical display. I had to edit the xml file by hand.

Implicit Example with Categories

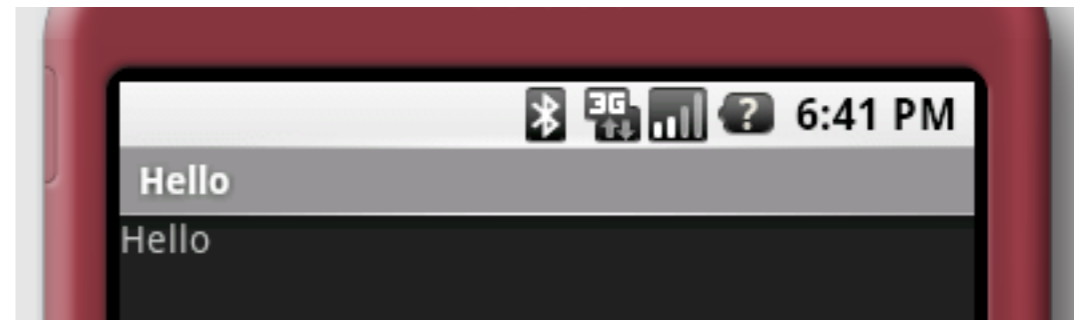
Button click will call another component

IntentExample.java



Clicking the "Go" button will launch the Hello activity

Hello.java



Category Example

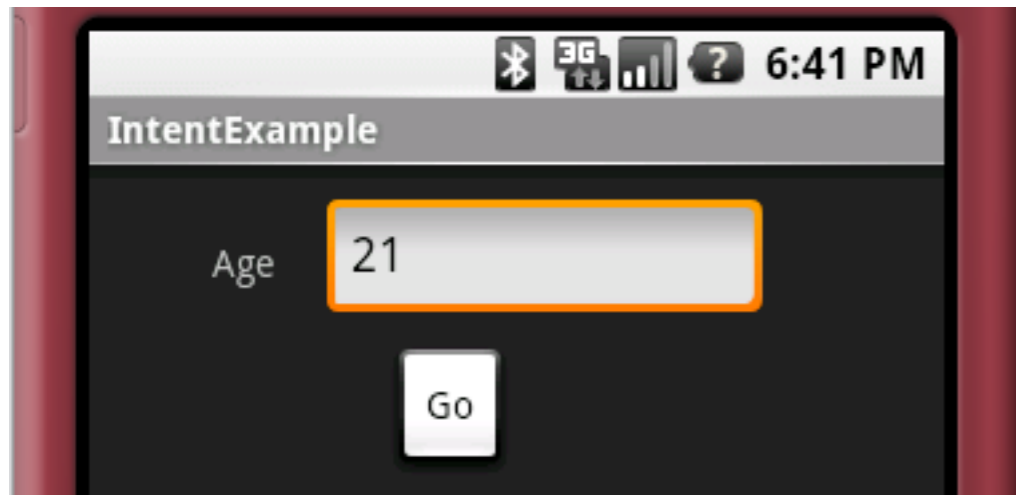
```
public class IntentExample extends Activity implements View.OnClickListener {  
  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.intent_example);  
        Button ok = (Button) findViewById(R.id.go);  
        ok.setOnClickListener(this);  
    }  
  
    public void onClick(View v) {  
        Intent go = new Intent();  
        go.setAction("cs683.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.cs683.example" android:versionCode="1"
    android:versionName="1.0.0">
    <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="cs683.sender.add"></action>
                <category android:name="android.intent.category.DEFAULT"></category>
                <category android:name="foo"></category>
            </intent-filter>
        </activity>
    </application>
</manifest>
```

Intent - Passing Data

IntentExample



Displays/Edits age

Go button

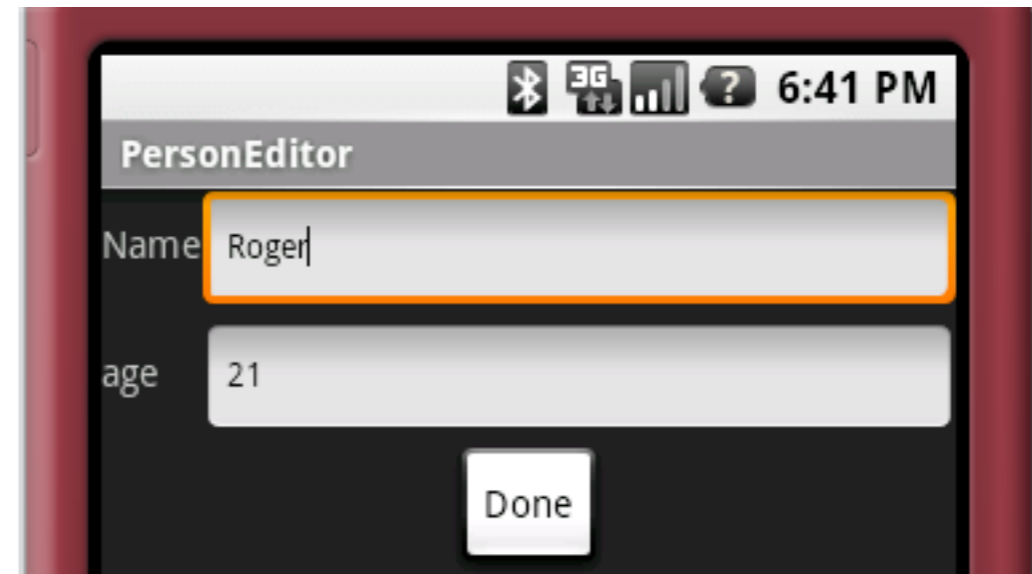
Calls PersonEditor

Passes data

Name

Age

PersonEditor



Displays/Edits Name and age

Done button

Returns edited data back

Age = 0 cancels edit


```
package edu.sdsu.cs683.example;
```

IntentExample.java

```
import android.app.Activity;
```

```
import android.content.Intent;
```

```
import android.os.Bundle;
```

```
import android.widget.Button;
```

```
import android.widget.EditText;
```

```
import android.view.View;
```

```
public class IntentExample extends Activity implements View.OnClickListener {
```

```
    private EditText numberText;
```

```
    private static final int INTENT_EXAMPLE_REQUEST = 123;
```

```
    public void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.intent_example);
```

```
        Button ok = (Button) findViewById(R.id.go);
```

```
        ok.setOnClickListener(this);
```

```
        numberText = (EditText) this.findViewById(R.id.number);
```

```
        numberText.setText("21");
```

```
    }
```

IntentExample.Java continued

Sending the data to PersonEditor

```
public void onClick(View v) {  
    Intent go;  
    go = new Intent();  
    go.setAction("android.intent.action.EDIT");  
    go.addCategory("person_editor");  
    String newAge = numberText.getText().toString();  
    go.putExtra("age", newAge);  
    go.putExtra("name", "Roger");  
    startActivityForResult(go, INTENT_EXAMPLE_REQUEST);  
}
```

IntentExample.Java continued

Getting the Results back

```
protected void onActivityResult(int requestCode, int resultCode, Intent data) {  
    if (requestCode != INTENT_EXAMPLE_REQUEST) {  
        numberText.setText("Not from me");  
        return;  
    }  
    switch (resultCode) {  
    case RESULT_OK:  
        String editedAge = data.getStringExtra("age");  
        numberText.setText(editedAge);  
        break;  
    case RESULT_CANCELED:  
        numberText.setText("Cancelled");  
        break;  
    }  
    }  
}
```

PersonEditor.java

```
public class PersonEditor extends Activity implements View.OnClickListener {
    private EditText ageText;
    private EditText nameText;

    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.person_editor);
        Button done = (Button) findViewById(R.id.edit_done);
        done.setOnClickListener(this);
        ageText = (EditText) this.findViewById(R.id.edit_age);
        nameText = (EditText) this.findViewById(R.id.edit_name);
        Bundle personData = getIntent().getExtras();
        String age = personData.getString("age");
        String name = personData.getString("name");
        if ((age != null) && (name != null)) {
            ageText.setText(age);
            nameText.setText(name);
        }
    }
}
```

PersonEditor.java

Returning the data

```
public void onClick(View v) {  
    String newAge = ageText.getText().toString();  
    Intent result = getIntent();  
    result.putExtra("age", newAge);  
    if (newAge.equals("0"))  
        setResult(RESULT_CANCELED, result);  
    else  
        setResult(RESULT_OK, result);  
    finish();  
}
```

PersonEditor.java - imports

```
package edu.sdsu.cs683.example;
```

```
import android.app.Activity;
```

```
import android.content.Intent;
```

```
import android.os.Bundle;
```

```
import android.view.View;
```

```
import android.widget.Button;
```

```
import android.widget.EditText;
```

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="edu.sdsu.cs683.example" android:versionCode="1"
    android:versionName="1.0.0">
    <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="PersonEditor" android:name="PersonEditor">
            <intent-filter>
                <action android:name="android.intent.action.EDIT"></action>
                <category android:name="person_editor"></category>
                <category android:name="android.intent.category.DEFAULT">
                    </category>
            </intent-filter>
        </activity>
    </application>
</manifest>
```

intent_example.xml

```
<?xml version="1.0" encoding="utf-8"?>
<AbsoluteLayout
  android:id="@+id/layout"
  android:layout_width="fill_parent"
  android:layout_height="fill_parent"
  xmlns:android="http://schemas.android.com/apk/res/android"
  >
  <Button
    android:id="@+id/go"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/go_button"
    android:layout_x="113px"
    android:layout_y="68px"
  >
  </Button>
```

```
<TextView
  android:id="@+id/widget53"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:text="Age"
  android:layout_x="46px"
  android:layout_y="28px"
  >
</TextView>
<EditText
  android:id="@+id/number"
  android:layout_width="163px"
  android:layout_height="wrap_content"
  android:text="EditText"
  android:textSize="18sp"
  android:layout_x="89px"
  android:layout_y="12px"
  >
</EditText>
</AbsoluteLayout>
```


person_editor.xml

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<RelativeLayout
    android:id="@+id/widget60"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    xmlns:android="http://schemas.android.com/apk/res/android"
    >
    <TableLayout
        android:id="@+id/widget78"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:stretchColumns="1"
        android:layout_alignParentTop="true"
        android:layout_alignParentLeft="true"
        >
        <TableRow
            android:id="@+id/widget79"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal"
            >
            <TextView
                android:id="@+id/widget81"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Name"
            >
            </TextView>
            <EditText
                android:id="@+id/edit_name"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:textSize="12sp"
            >
            </EditText>
        </TableRow>
```

```
    <TableRow
        android:id="@+id/widget80"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        >
        <TextView
            android:id="@+id/widget81"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="age"
        >
        </TextView>
        <EditText
            android:id="@+id/edit_age"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textSize="12sp"
        >
        </EditText>
    </TableRow>
</TableLayout>
<Button
    android:id="@+id/edit_done"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Done"
    android:layout_below="@+id/widget78"
    android:layout_centerHorizontal="true"
    >
</Button>
</RelativeLayout>
```