

## Android Intents

Victor Matos  
Cleveland State University

Notes are based on:

Android Developers

<http://developer.android.com/index.html>





# Intents

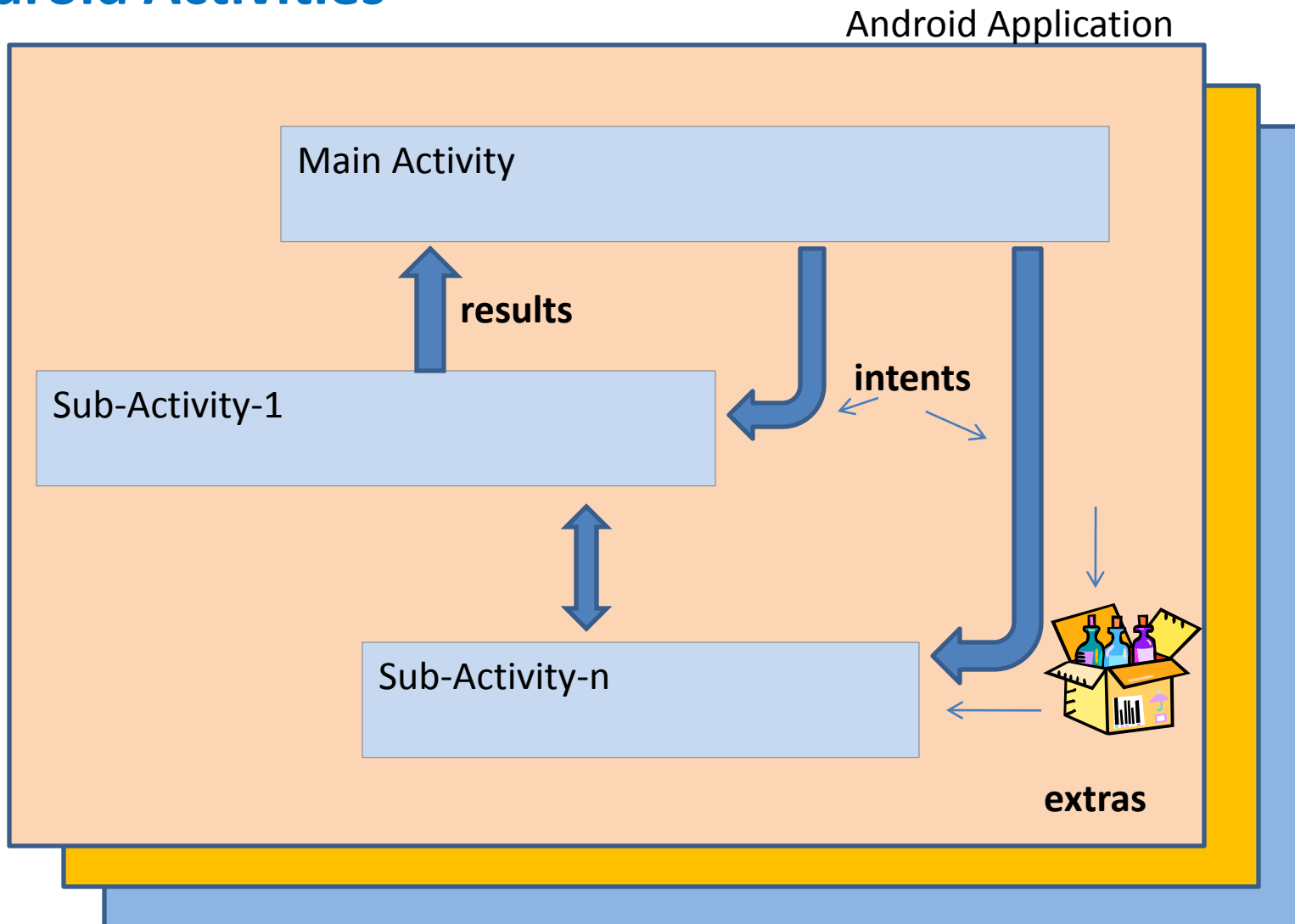
## Android Activities

An Android application could include any number of activities.

- An *activity* uses the *setContentView(...)* method to expose (usually) a single UI from which a number of actions could be performed.
- Activities are independent of each other; however they usually cooperate exchanging data and actions.
- Typically, one of the activities is designated as the first one (*main*) that should be presented to the user when the application is launched.
- Moving from one activity to another is accomplished by asking the current activity to execute an *intent*.
- Activities interact with each other in an **asynchronous** mode.

# Intents

## Android Activities





# Intents

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

**Intents** are invoked using the following options

<b><i>startActivity (intent)</i></b>	launches an <i>Activity</i>
<b><i>sendBroadcast (intent)</i></b>	sends an intent to any interested <i>BroadcastReceiver</i> components
<b><i>startService(intent)</i> or <i>bindService(intent, ...)</i></b>	communicate with a background Service.

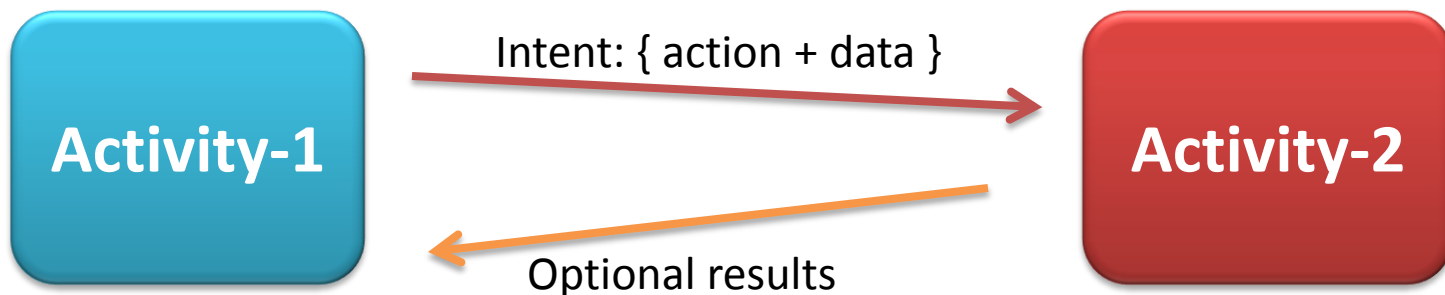


# Intents

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

The main arguments of an Intent are:

- 1. Action** The built-in action to be performed, such as **ACTION\_VIEW**, **ACTION\_EDIT**, **ACTION\_MAIN**, ... or *user-created-activity*
- 2. Data** The primary data to operate on, such as a phone number to be called (expressed as a **Uri**).



# Intents

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

Typically an intent is called as follows:

```
Intent myActivity = new Intent (action, data);  
startActivity (myActivity);
```

Built-in or  
user-created  
activity

Primary data (as an URI)  
tel://  
http://  
sendto://



# Intents

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

Examples of **action/data** pairs are:

**ACTION\_DIAL**      *tel:123*

Display the phone dialer with the given number filled in.

**ACTION\_VIEW**      *http://www.google.com*

Show Google page in a browser view. Note how the VIEW action does what is considered the most reasonable thing for a particular URI.

**ACTION\_EDIT**      *content://contacts/people/2*

Edit information about the person whose identifier is "2".

**ACTION\_VIEW**      *content://contacts/people/2*

Used to start an activity to display 2-nd person.

**ACTION\_VIEW**      *content://contacts/people/*

Display a list of people, which the user can browse through. Selecting a particular person to view would result in a new intent



# Intents

## Built-in Standard Actions

List of standard actions that Intents can use for launching activities (usually through *startActivity(Intent)*).

**ACTION\_MAIN**

ACTION\_VIEW

ACTION\_ATTACH\_DATA

**ACTION\_EDIT**

ACTION\_PICK

ACTION\_CHOOSER

ACTION\_GET\_CONTENT

ACTION\_DIAL

**ACTION\_CALL**

ACTION\_SEND

**ACTION\_SENDTO**

ACTION\_ANSWER

ACTION\_INSERT

ACTION\_DELETE

ACTION\_RUN

ACTION\_SYNC

ACTION\_PICK\_ACTIVITY

**ACTION\_SEARCH**

**ACTION\_WEB\_SEARCH**

ACTION\_FACTORY\_TEST



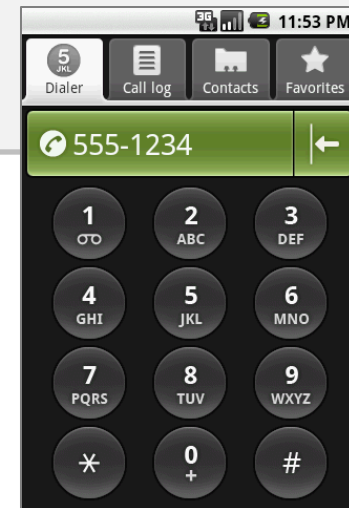
# Intents

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

## Example

Display the phone dialer with the given number filled in.

```
Intent myActivity2 = new Intent (Intent.ACTION_DIAL,  
                                Uri.parse( "tel:555-1234" ) );  
startActivity(myActivity2);
```



# Intents

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

## Intents - Secondary Attributes

In addition to the primary *action/data* attributes, there are a number of **secondary attributes** that you can also include with an intent, such as:

1. Category
2. Components
3. Type
4. Extras

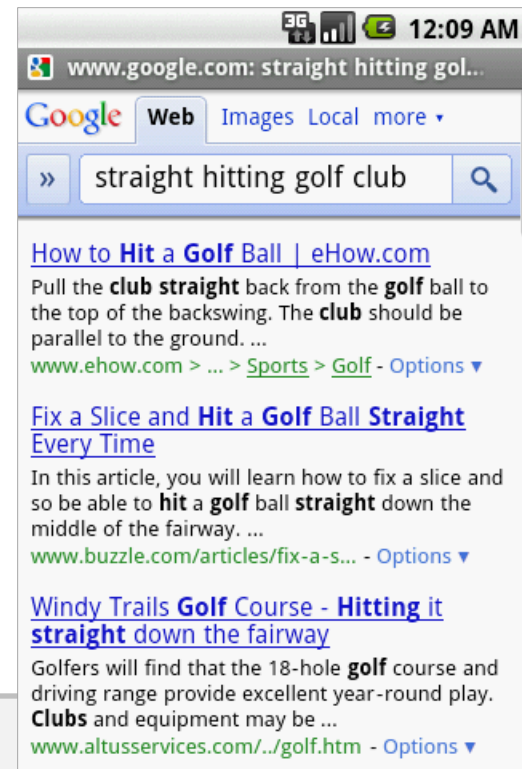
**Example:** Doing a Google search looking for golf clubs

```
Intent intent = new Intent (Intent.ACTION_WEB_SEARCH );

intent.putExtra (SearchManager.QUERY,
                "straight hitting golf clubs");

startActivity (intent);
```

Secondary data





# Intents

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

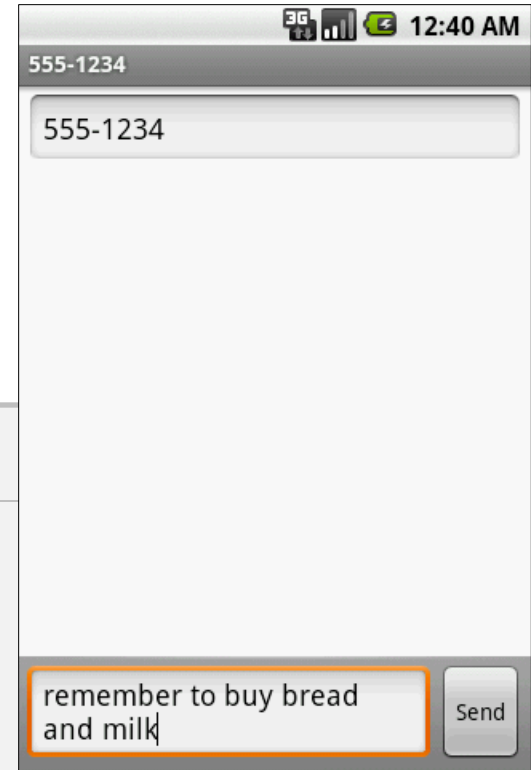
## Intents - Secondary Attributes

**Example:** Sending a text message (using extra attributes)

```
Intent intent = new Intent( Intent.ACTION_SENDTO,
                           Uri.parse("sms://"));

intent.putExtra("address", "555-1234");
intent.putExtra("sms_body", "remember to buy bread and milk");

startActivity(intent);
```





# Intents

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

## Intents - Secondary Attributes

**Example:** Showing Pictures (using extra attributes)

```
Intent myIntent = new Intent();
```

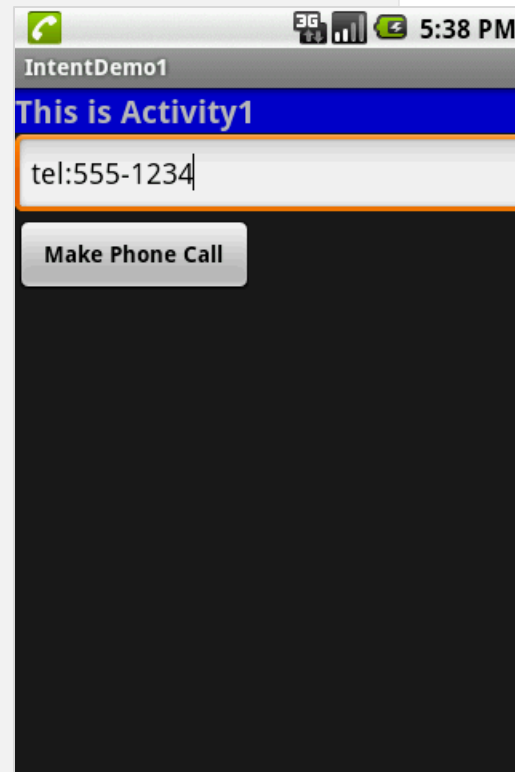
```
myIntent.setType("image/pictures/*");  
myIntent.setAction(Intent.ACTION_GET_CONTENT);  
  
startActivity(myIntent);
```



# Intents

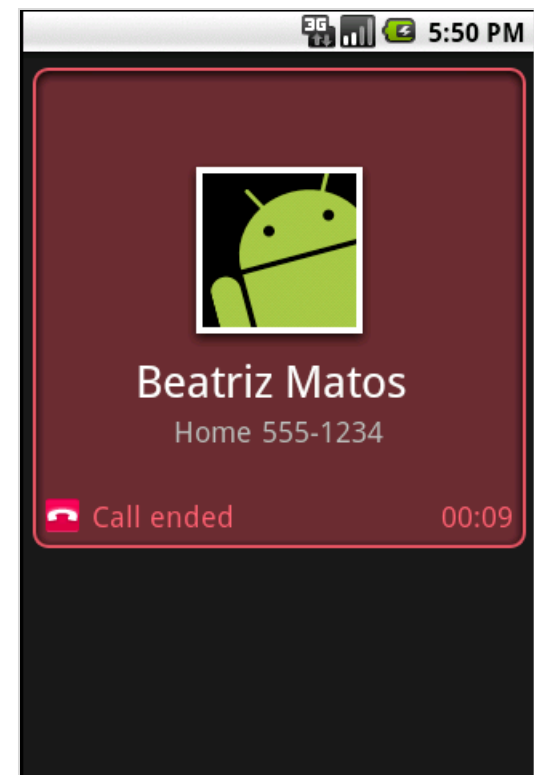
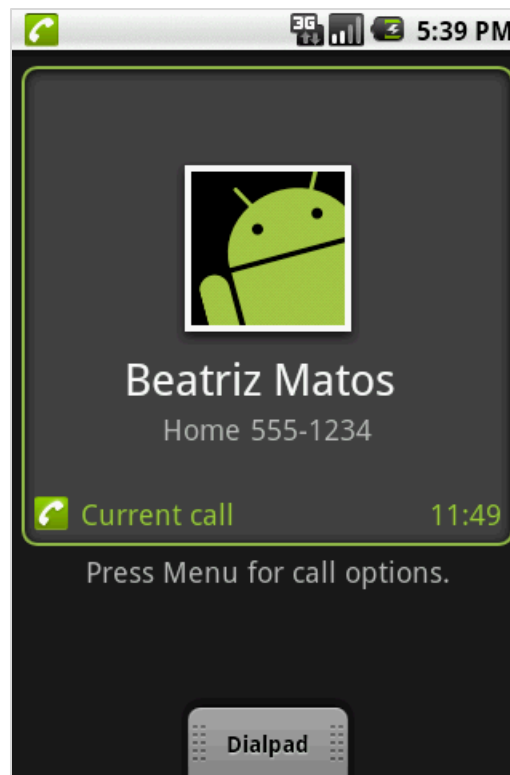
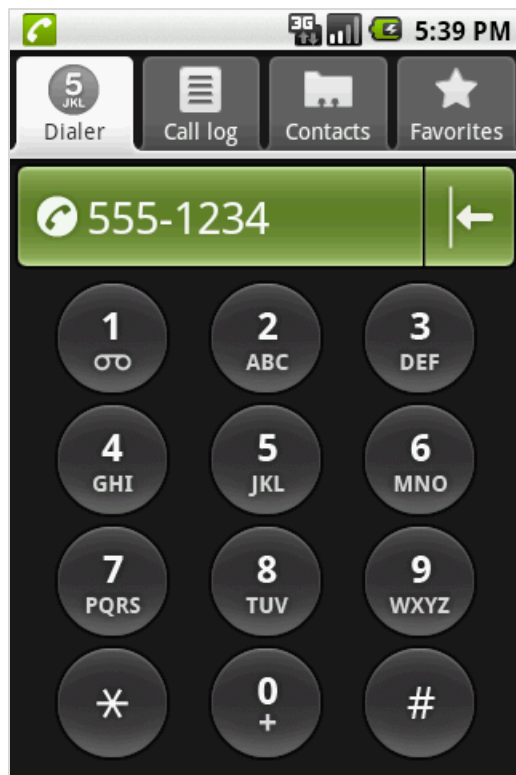
**1. A Complete Example:** Activity1 displays an interface to accept a phone number and requests (built-in) Activity2 to make the call.

```
<?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:id="@+id/label1"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:background="#ff0000cc"
    android:text="This is Activity1"
    android:textStyle="bold"
    android:textSize="20sp" />
<EditText
    android:id="@+id/text1"
    android:layout_width="fill_parent"
    android:layout_height="54px"
    android:text="tel:555-1234"
    android:textSize="18sp" />
<Button
    android:id="@+id/btnCallActivity2"
    android:layout_width="149px"
    android:layout_height="wrap_content"
    android:text="Make Phone Call"
    android:textStyle="bold" />
</LinearLayout>
```



# Intents

**1. A Complete Example:** Activity1 displays an interface to accept a phone number and requests (built-in) Activity2 to make the call.





# Intents

**1. A Complete Example:** Activity1 displays an interface to accept a phone number and requests (built-in) Activity2 to make the call.

```
//IntentDemo1_Intent: making a phone call
package cis493.intents;
import android.app.Activity;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.*;

public class IntentDemo1 extends Activity {
    TextView label1;
    EditText text1;
    Button    btnCallActivity2;
```



# Intents

**1. A Complete Example:** Activity1 displays an interface to accept a phone number and requests (built-in) Activity2 to make the call.

```
@Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        try {

            setContentView(R.layout.main);
            label1 = (TextView)findViewById(R.id.label1);
            text1 = (EditText)findViewById(R.id.text1);

            btnCallActivity2 = (Button)findViewById(R.id.btnCallActivity2);
            btnCallActivity2.setOnClickListener(new ClickHandler());
        }
        catch (Exception e) {
            Toast.makeText(getApplicationContext(), e.getMessage(),
                                Toast.LENGTH_LONG).show();
        }
    }
} //onCreate
```



# Intents

**1. A Complete Example:** Activity1 displays an interface to accept a phone number and requests (built-in) Activity2 to make the call.

```
private class ClickHandler implements OnClickListener {
    @Override
    public void onClick(View v) {
        try {
            // myActivity2 places a phone call
            // for ACTION_CALL or ACTION_DIAL
            // use 'tel:' formatted data: "tel:555-1234"
            // for ACTION_VIEW use data: "http://www.youtube.com"
            // (you also need INTERNET permission - see Manifest)

            String myData = text1.getText().toString();
            Intent myActivity2 = new Intent(Intent.ACTION_DIAL,
                                           Uri.parse(myData));
            startActivity(myActivity2);
        }
        catch (Exception e) {
            Toast.makeText(getApplicationContext(), e.getMessage(),
                           Toast.LENGTH_LONG).show();
        }
    } //onClick
} //ClickHandler
} //IntentDemo1
```



# Intents

**1. A Complete Example:** Activity1 displays an interface to accept a phone number and requests (built-in) Activity2 to make the call.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="cis493.intents"
    android:versionCode="1"
    android:versionName="1.0">
    <application android:icon="@drawable/icon"
        android:label="@string/app_name">
        <activity android:name=".IntentDemo1"
            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="3" />
</manifest>
```

← Action/category



# Intents

## Built-in Standard Broadcast Actions

List of standard actions that Intents can use for receiving broadcasts (usually through *registerReceiver(BroadcastReceiver, IntentFilter)* or a *<receiver>* tag in a manifest).

ACTION_TIME_TICK ACTION_TIME_CHANGED ACTION_TIMEZONE_CHANGED ACTION_BOOT_COMPLETED ACTION_PACKAGE_ADDED ACTION_PACKAGE_CHANGED ACTION_PACKAGE_REMOVED ACTION_UID_REMOVED ACTION_BATTERY_CHANGED	
---	--

# Intents

## More Examples: Using Standard Actions

### Call Immediately

Modify the *complete* example1 replacing the method 'ClickHandler' with the following code

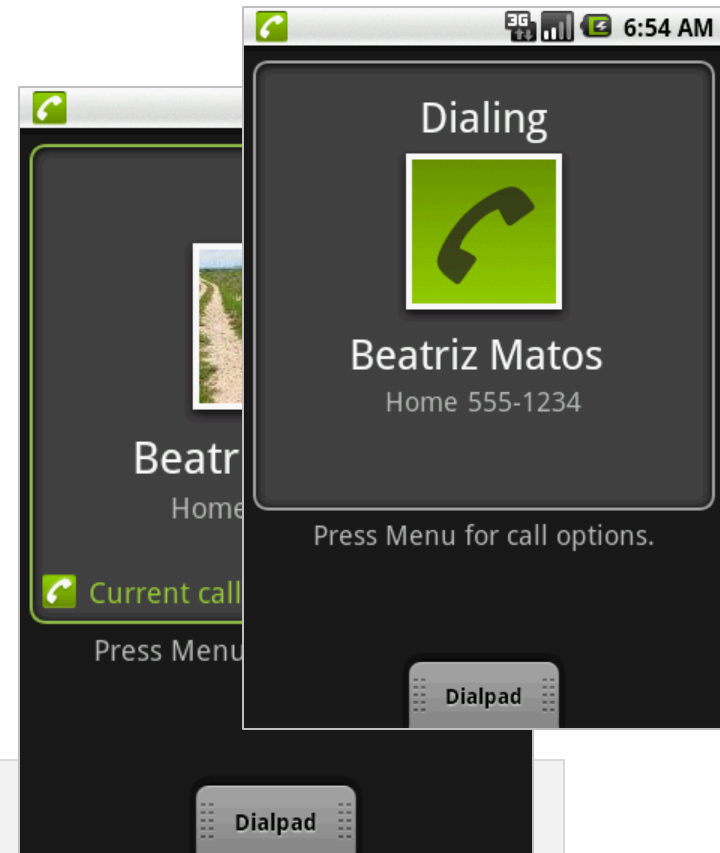
```
String myData = "tel:555-1234";

Intent myActivity2 = new Intent(Intent.ACTION_CALL,
                                Uri.parse(myData));

startActivity(myActivity2);
```

Needs Permission:

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





# Intents

## More Examples: Using Standard Actions

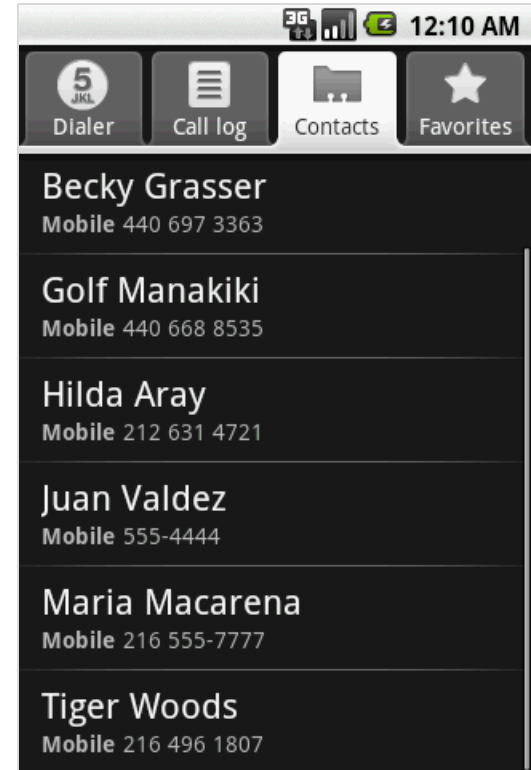
### Show all your Contacts

Modify the *complete* example1 replacing the method 'ClickHandler' with the following code

```
String myData = "content://contacts/people/";

Intent myActivity2 = new Intent(Intent.ACTION_VIEW,
                                Uri.parse(myData));

startActivity(myActivity2);
```

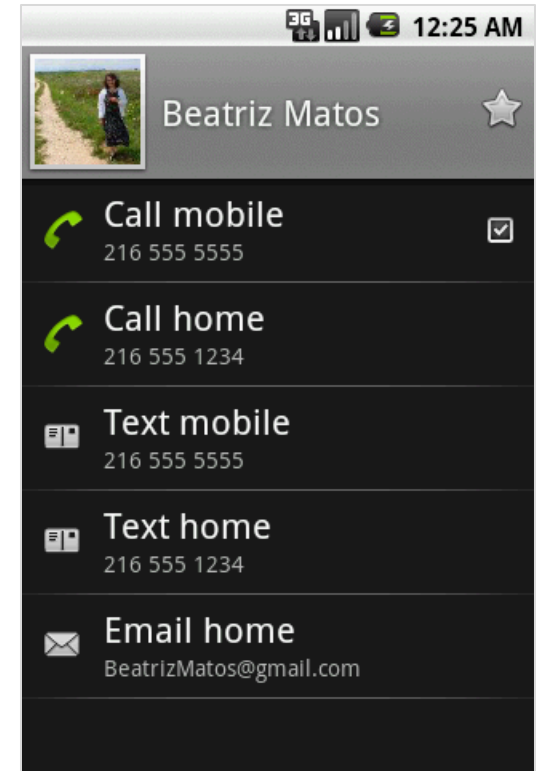


# Intents

## More Examples: Using Standard Actions

### Show a Particular Contact (ID = 2)

Modify the *complete* example1 replacing the method 'ClickHandler' with the following code



```
String myData = "content://contacts/people/2";

Intent myActivity2 = new Intent(Intent.ACTION_VIEW,
                                Uri.parse(myData));

startActivity(myActivity2);
```

# Intents

## More Examples: Using Standard Actions

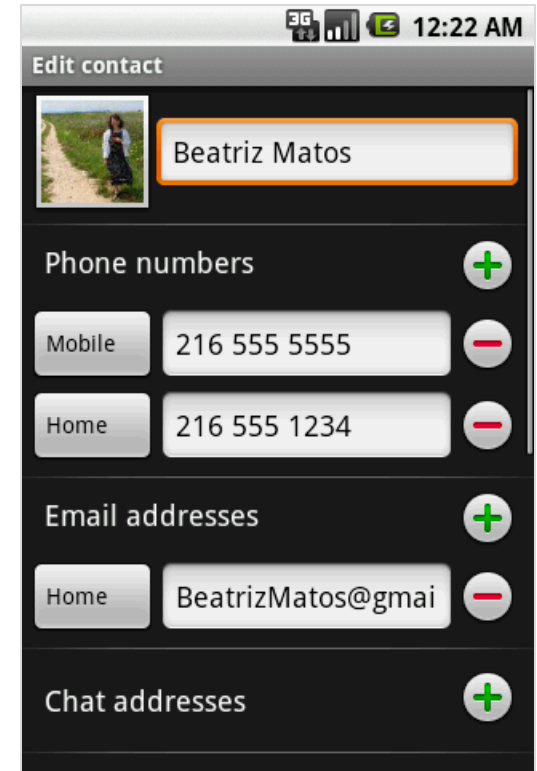
### Edit a Particular Contact (ID = 2)

Modify the *complete* example1 replacing the method 'ClickHandler' with the following code

```
String myData = "content://contacts/people/2";

Intent myActivity2 = new Intent(Intent.ACTION_EDIT,
                                Uri.parse(myData));

startActivity(myActivity2);
```



# Intents

## More Examples: Using Standard Actions

### View a Webpage

Modify the *complete* example1 replacing the method 'ClickHandler' with the following code

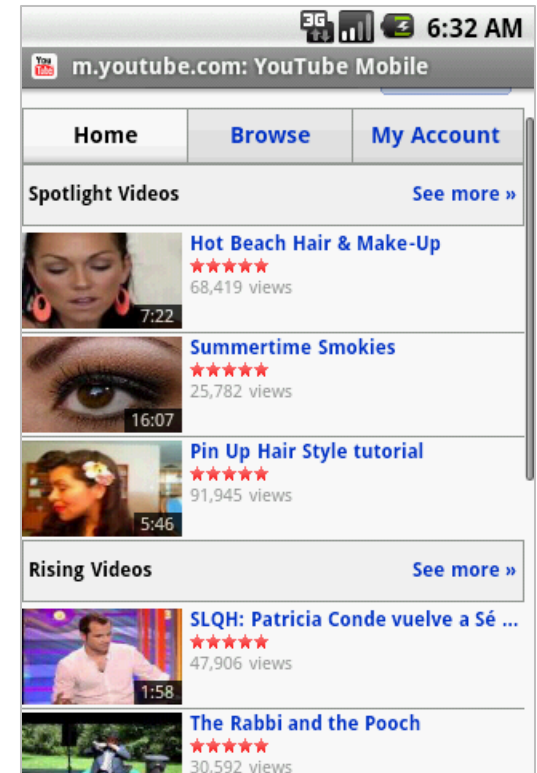
```
String myData = "http://www.youtube.com";

Intent myActivity2 = new Intent(Intent.ACTION_VIEW,
                                Uri.parse(myData));

startActivity(myActivity2);
```

**Caution.** Add to the Manifest a request to use the Internet:

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





## More Examples: Using Standard Actions

## Geo Mapping an Address

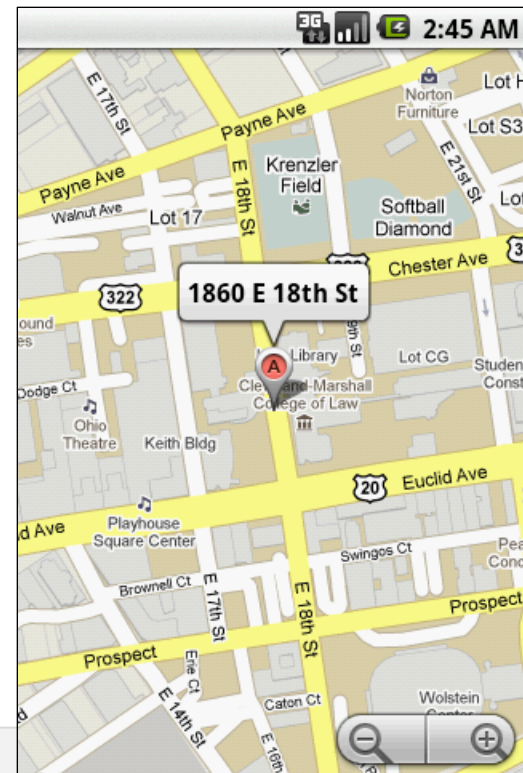
Provide a geoCode expression holding a street address (or place, such as 'golden gate ca' )

## Replace spaces with '+'.

```
String geoCode =  
    "geo:0,0?q=1860+east+18th+street+cleveland+oh";  
Intent intent = new Intent(Intent.ACTION_VIEW,  
    Uri.parse(geoCode));  
startActivity(intent);
```

Modify the Manifest adding the following requests:

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



# Intents

## More Examples: Using Standard Actions

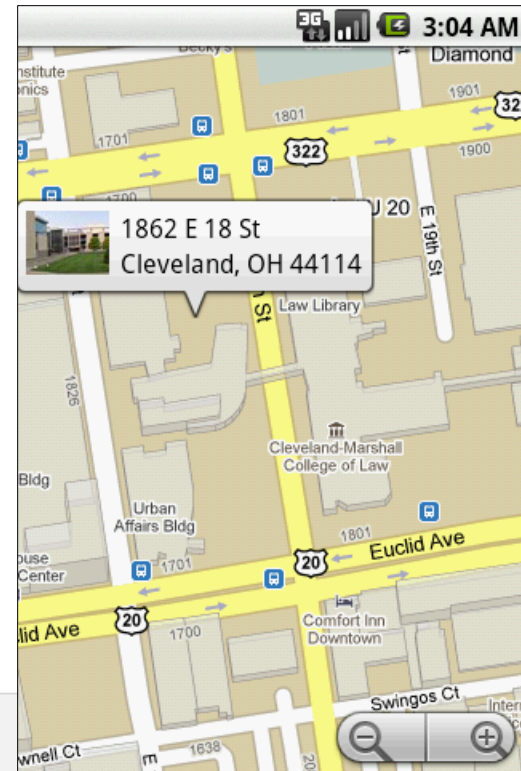
### Geo Mapping Coordinates (latitude, longitude)

Provide a geoCode holding latitude and longitude (also an additional zoom '**?z=xx**' with xx in range 1..23)

```
String geoCode =
    "geo:41.5020952,-81.6789717";
Intent intent = new Intent(Intent.ACTION_VIEW,
                           Uri.parse(geoCode));
startActivity(intent);
```

Modify the Manifest adding the following requests:

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



# Intents

## More Examples: Using Standard Actions

### Geo Mapping - Google StreetView

geoCode Uri structure:

`google.streetview:cbll=lat,lng&cbp=1,  
yaw,,pitch,zoom&mz=mapZoom`

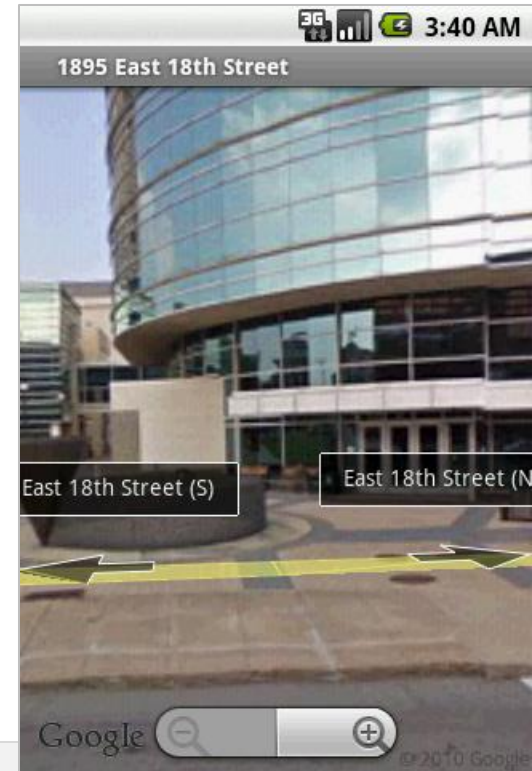
Reference: <http://developer.android.com/guide/appendix/g-app-intents.html>

```
String geoCode =
    "google.streetview:cbll=41.5020952,-81.6789717&cbp=1,270,,45,1&mz=1";

Intent intent = new Intent(Intent.ACTION_VIEW,
                           Uri.parse(geoCode));
startActivity(intent);
```

Modify the Manifest adding the following requests:

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





# Intents

## More Examples: Using Standard Actions

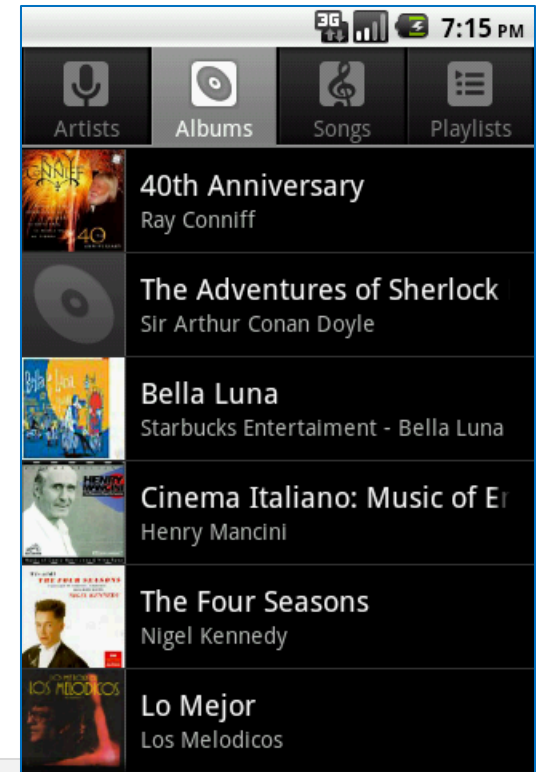
### Launching the Music Player

Reference: <http://developer.android.com/guide/appendix/g-app-intents.html>

```
//launch music player
```

```
Intent myActivity2 =
    new Intent("android.intent.action.MUSIC_PLAYER");

startActivity(myActivity2);
```



# Intents

## More Examples: Using Standard Actions

### Playing a song stored in the SD card

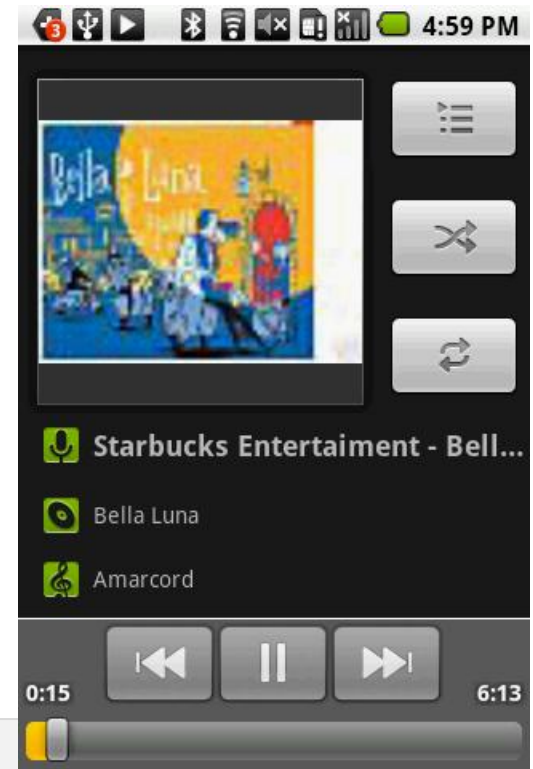
Reference: <http://developer.android.com/guide/appendix/g-app-intents.html>

```
// play song "amarcord.mp3" saved in the SD
Intent myActivity2 =
    new Intent(android.content.Intent.ACTION_VIEW);

Uri data = Uri.parse("file:///sdcard/amarcord.mp3");
String type = "audio/mp3";

myActivity2.setDataAndType(data, type);

startActivity(myActivity2);
```





# Intents

## More Examples: Using Standard Actions

### Sending MMS

Add picture #1 from SD to MMS

Reference: <http://developer.android.com/guide/appendix/g-app-intents.html>

```
//send mms attach picture #1 to it
```

```
Uri uri = Uri.parse("content://media/external/images/media/1");
```

```
myActivity2 = new Intent(Intent.ACTION_SEND);
```

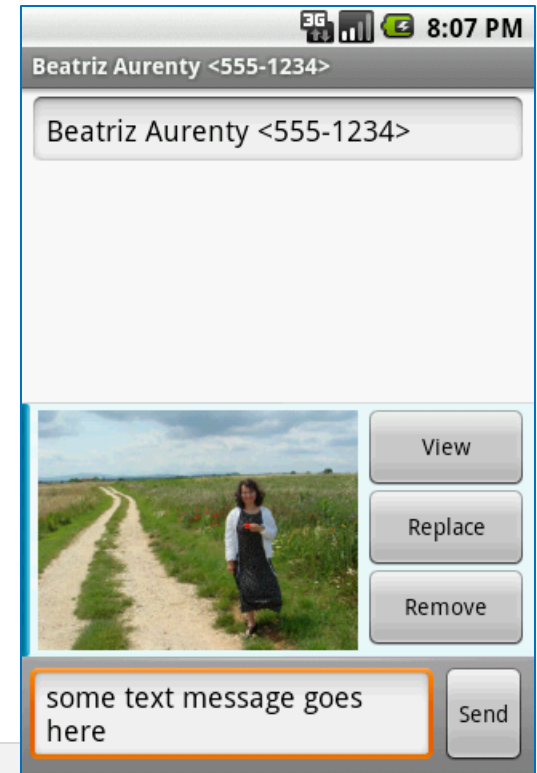
```
myActivity2.putExtra("address", "555-1234");
```

```
myActivity2.putExtra("sms_body", "some text message goes here");
```

```
myActivity2.putExtra(Intent.EXTRA_STREAM, uri);
```

```
myActivity2.setType("image/png");
```

```
startActivity(myActivity2);
```



# Intents

## More Examples: Using Standard Actions

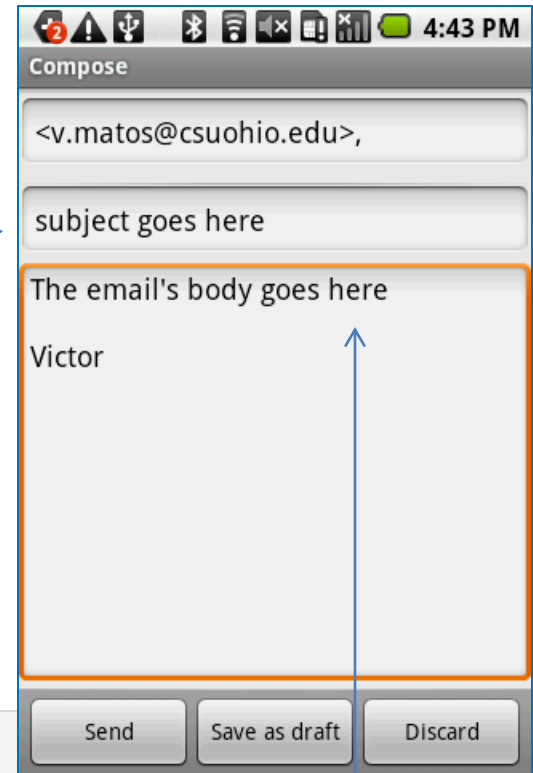
### Sending Email

Reference: <http://developer.android.com/guide/appendix/g-app-intents.html>

```
// send email
Uri uri = Uri.parse("mailto:v.matos@csuohio.edu");
Intent myActivity2 = new Intent(Intent.ACTION_SENDTO, uri);

// you may skip the next two pieces [subject/text]
myActivity2.putExtra(Intent.EXTRA_SUBJECT,
    "subject goes here");
myActivity2.putExtra(Intent.EXTRA_TEXT,
    "The email's body goes here");

startActivity(myActivity2);
```

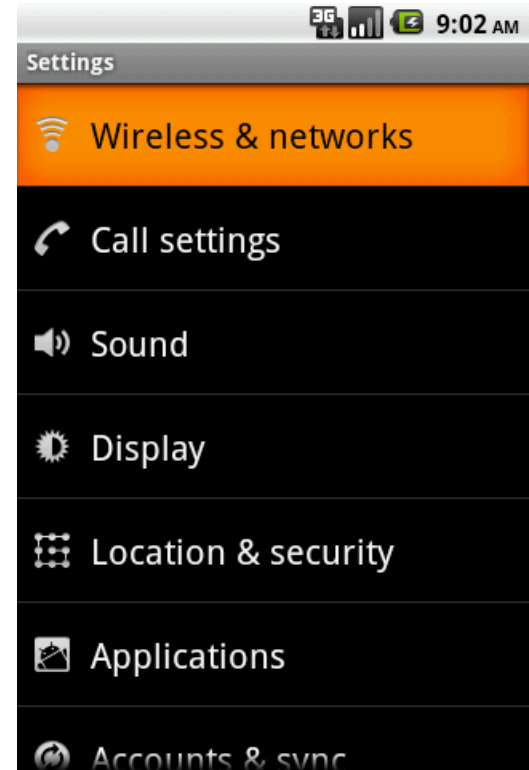


# Intents

## More Examples: Using Standard Actions

### Setting System

Reference: <http://developer.android.com/reference/android/provider/Settings.html>



```
Intent intent = new Intent(  
    android.provider.Settings.ACTION_SETTINGS);  
  
startActivity(intent);
```





# Intents

## More Examples: Using Standard Actions

### Setting System Locale: Language & Keyboard

Reference: <http://developer.android.com/reference/android/provider/Settings.html>

```
Intent intent = new Intent(
    android.provider.Settings.ACTION_LOCALE_SETTINGS);
startActivity(intent);
```







# Intents

## Starting Activities and Getting Results

The **startActivity(Intent)** method is used to start a new activity, which will be placed at the top of the activity stack.

Sometimes you want to get a result back from the called sub-activity when it ends.



For example, you may start an activity that let the user pick a person from a list of contacts; when it ends, it returns the person that was selected.



# Intents

## Starting Activities and Getting Results

In order to get results back from the called activity we use the method

**`startActivityForResult ( Intent, requestCodeID )`**



Where the second (*requestCodeID*) parameter identifies the call.

The result sent by the sub-activity could be picked up through the asynchronous method

**`onActivityResult ( requestCodeID, resultCode, Intent )`**





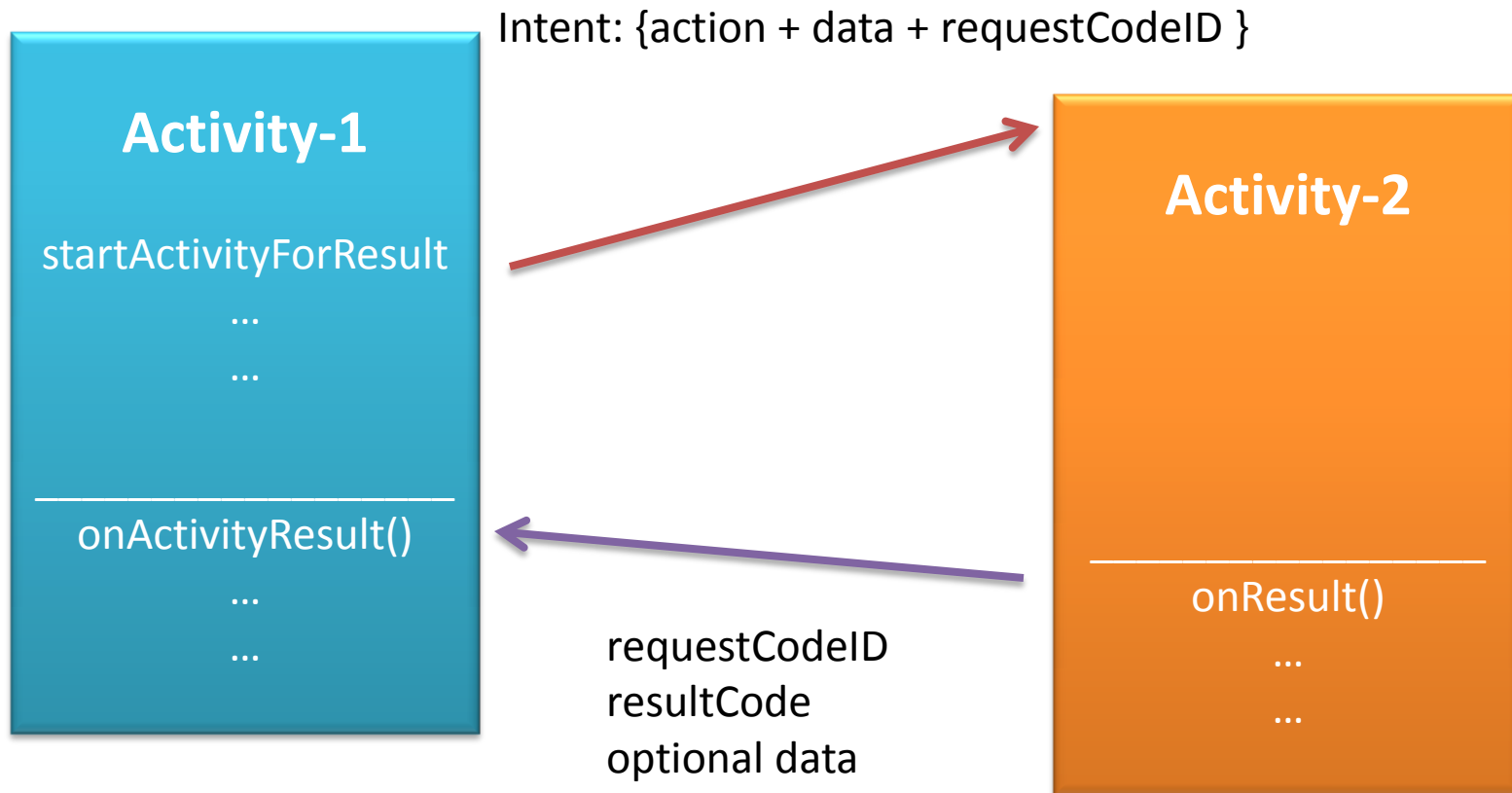
# Intents

## Starting Activities and Getting Results

- Before an activity exits, it can call **setResult (resultCode)** to return a termination signal back to its parent.
- Always supply a result code, which can be the standard results **Activity.RESULT\_CANCELED, Activity.RESULT\_OK**, or any custom values.
- All of this information can be capture back on the parent's **onActivityResult (int requestCodeID, int resultCode, Intent data)** along with the integer identifier it originally supplied.
- If a child activity fails for any reason (such as crashing), the parent activity will receive a result with the code **RESULT\_CANCELED**.

# Intents

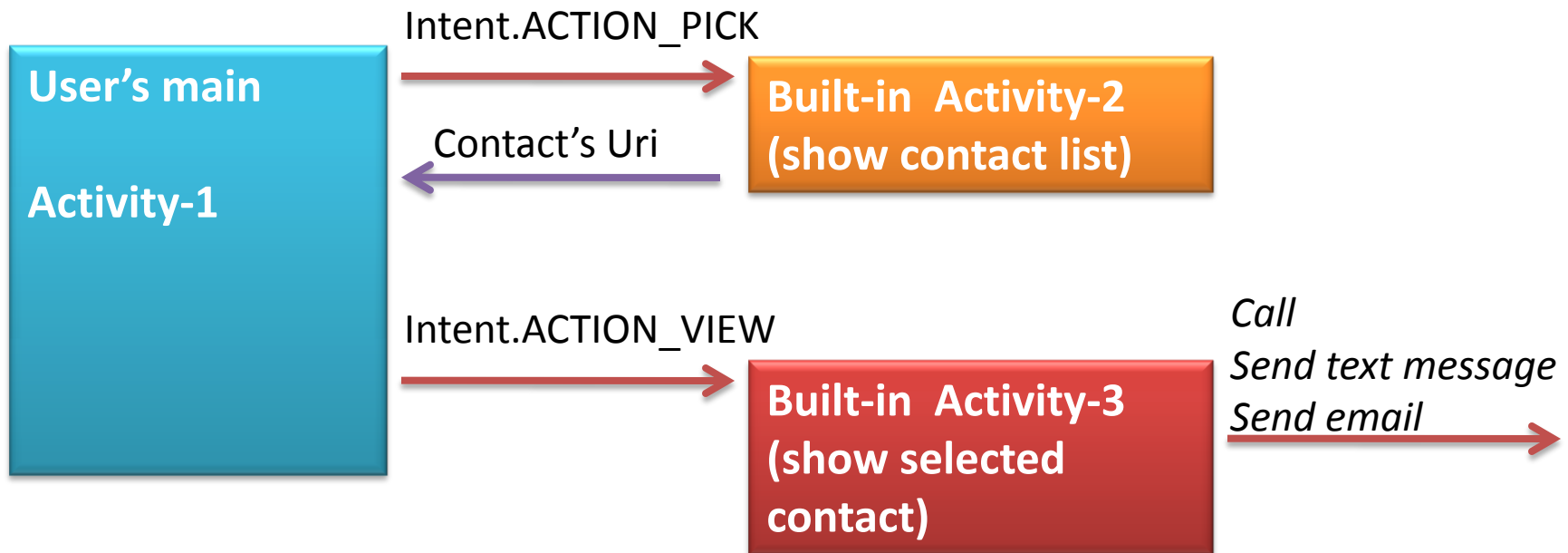
## Starting Activities and Getting Results



# Intents

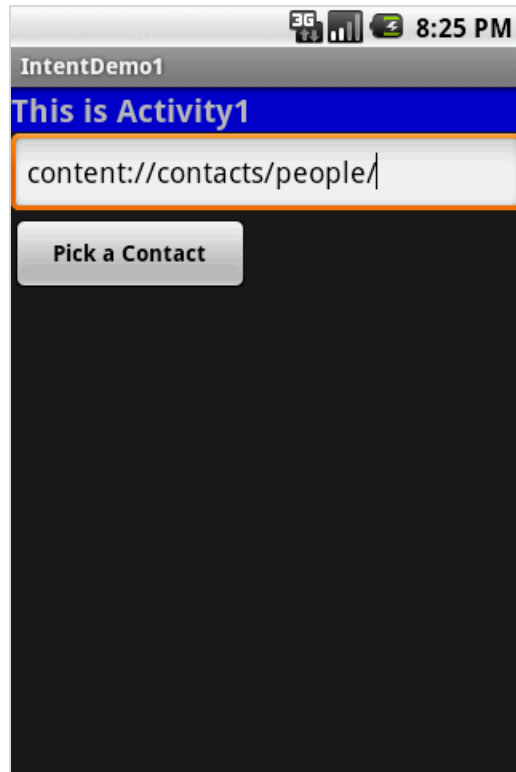
**Example2.** Let's play golf - Call for a tee-time.

1. Show all contacts and pick a particular one (*Intent.ACTION\_PICK*).
2. For a successful interaction the main-activity accepts the returned URI identifying the person we want to call (*content://contacts/people/n*).
3. 'Nicely' show the selected contact's entry allowing calling, texting, emailing actions (*Intent.ACTION\_VIEW*).

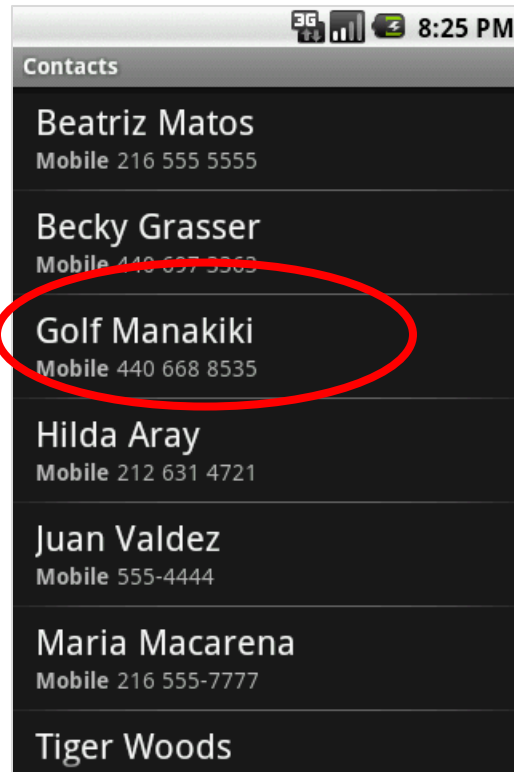


# Intents

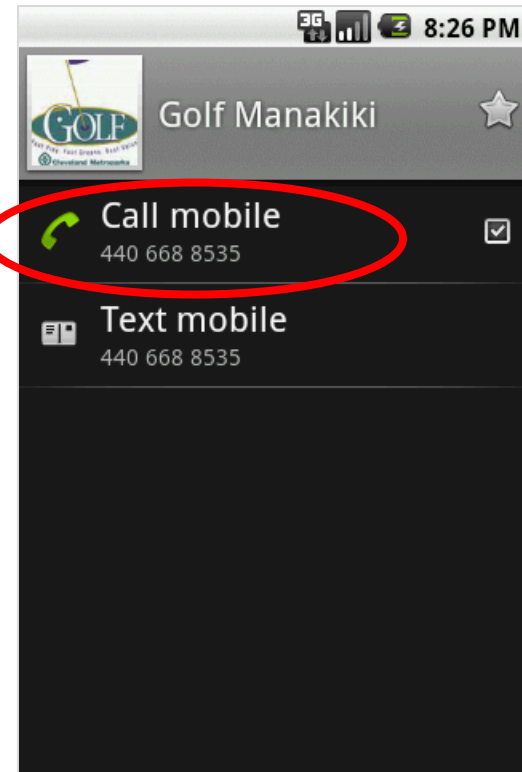
**Example2.** Let's play golf - *Call for a tee-time.*



Main Activity



Intent.ACTION\_PICK



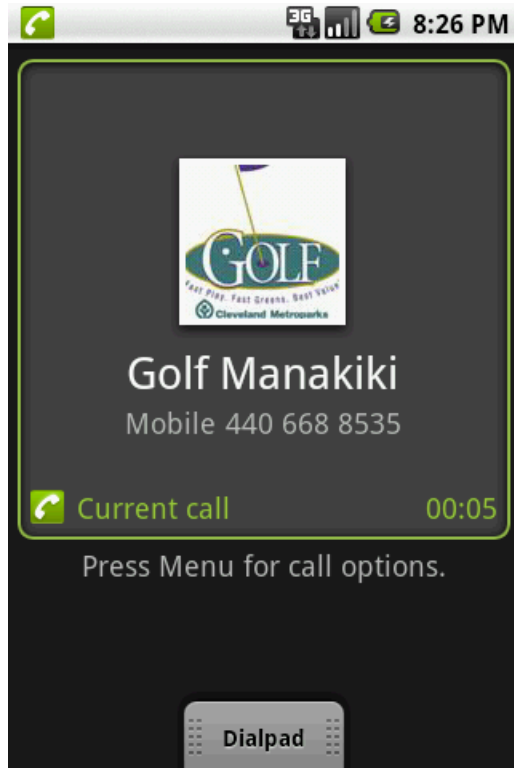
Intent.ACTION\_VIEW

Cont.

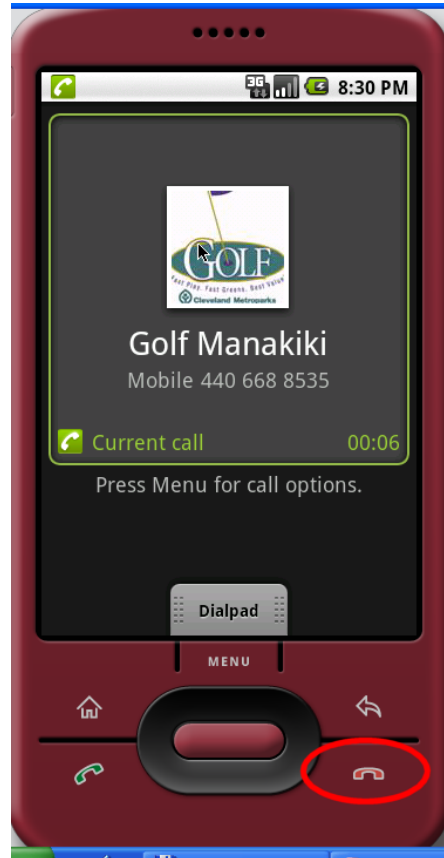


# Intents

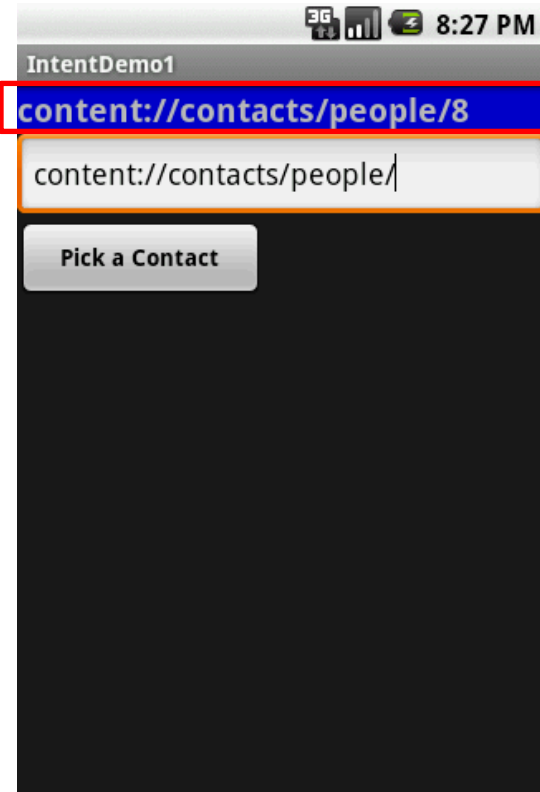
**Example2 (cont.)** Let's play golf - *Call for a tee-time*



Place the call



Terminate the call



Selected contact's URI



# Intents

**Example2.** *Calling a sub-activity, receiving results.*

```
//IntentDemo2_Intent: making a phone call
//receiving results from a sub-activity
package cis493.intents;
import android.app.Activity;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.*;

public class IntentDemo2 extends Activity {
    TextView label1;
    EditText text1;
    Button    btnCallActivity2;
```



# Intents

## Example2. *Calling a sub-activity, receiving results.*

```
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    try {
        setContentView(R.layout.main);
        label1 = (TextView)findViewById(R.id.label1);
        text1 = (EditText)findViewById(R.id.text1);

        btnCallActivity2 = (Button)findViewById(R.id.btnPickContact);
        btnCallActivity2.setOnClickListener(new ClickHandler());
    }
    catch (Exception e) {
        Toast.makeText(getBaseContext(),
            e.getMessage(), Toast.LENGTH_LONG).show();
    }
} //onCreate
```

# Intents

## Example2. *Calling a sub-activity, receiving results.*

```
private class ClickHandler implements OnClickListener {
    @Override
    public void onClick(View v) {
        try {
            // myData refer to: content://contacts/people/
            String myData = text1.getText().toString();

            //you may also try ACTION_VIEW instead
            Intent myActivity2 = new Intent(Intent.ACTION_PICK,
                                           Uri.parse(myData));

            // start myActivity2.
            // Tell it that our requestCodeID (or nickname) is 222
            startActivityForResult(myActivity2, 222);

            // Toast.makeText(getApplicationContext(),
            //                      "I can't wait for you", 1).show();

        }
        catch (Exception e) {
            label1.setText(e.getMessage());
        }
    } //onClick
} //ClickHandler
```



# Intents

## Example2. *Calling a sub-activity, receiving results.*

```

@Override
protected void onActivityResult(int requestCode,
                                int resultCode,
                                Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    try {
        // use requestCode to find out who is talking back to us
        switch (requestCode) {
            case (222): {
                // 222 is our friendly contact-picker activity
                if (resultCode == Activity.RESULT_OK) {
                    String selectedContact = data.getDataString();
                    // it will return an URI that looks like:
                    // content://contacts/people/n
                    // where n is the selected contacts' ID
                    label1.setText(selectedContact.toString());

                    //show a 'nice' screen with the selected contact
                    Intent myAct3 = new Intent (Intent.ACTION_VIEW,
                                                Uri.parse(selectedContact));
                    startActivity(myAct3);
                }
            }
        }
    }
}

```

Diagram annotations:

- A blue arrow points from the `requestCode` parameter to the `switch (requestCode)` statement.
- A blue arrow points from the `resultCode` parameter to the `if (resultCode == Activity.RESULT_OK)` condition.
- A blue arrow points from the `Intent data` parameter to the `data.getDataString()` call.
- A blue box labeled "Listener" points to the `onActivityResult` method signature.

# Intents

**Example2.** *Calling a sub-activity, receiving results.*

```

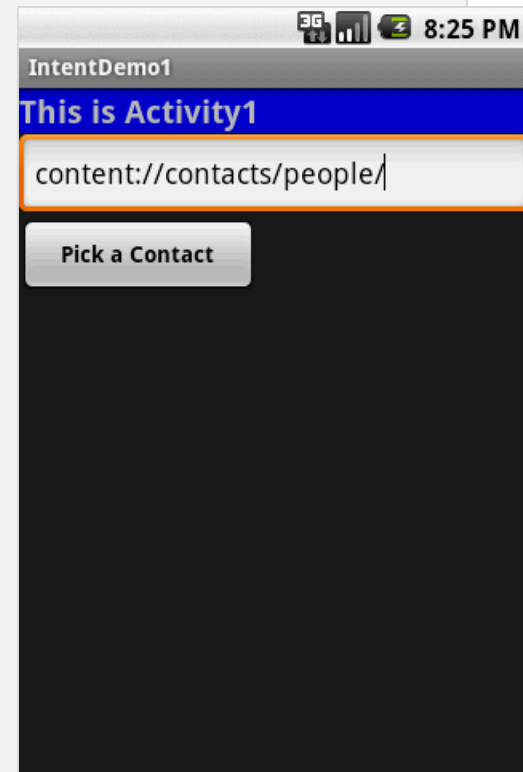
        else {
            //user pressed the BACK button
            label1.setText("Selection CANCELLED "
                           + requestCode + " " + resultCode);
        }
        break;
    }
} //switch
}
catch (Exception e) {
    Toast.makeText(getApplicationContext(), e.getMessage(),
                   Toast.LENGTH_LONG).show();
}
} // onActivityResult
} // IntentDemo2

```

# Intents

## Example2. *Calling a sub-activity, receiving results.*

```
<?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:id="@+id/label1"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:background="#ff0000cc"
        android:text="This is Activity1"
        android:textStyle="bold"
        android:textSize="20sp"/>
    <EditText
        android:id="@+id/text1"
        android:layout_width="fill_parent"
        android:layout_height="54px"
        android:text="content://contacts/people/"
        android:textSize="18sp" />
    <Button
        android:id="@+id/btnPickContact"
        android:layout_width="149px"
        android:layout_height="wrap_content"
        android:text="Pick a Contact"
        android:textStyle="bold" />
</LinearLayout>
```



# Intents

**Example3.** Showing Pictures and Video - Calling a sub-activity, receiving results.

```
private void showSoundTracks() {

    Intent myIntent = new Intent();
    myIntent.setType("video/*, images/*");
    myIntent.setAction(Intent.ACTION_GET_CONTENT);
    startActivityResult(myIntent, 0);

} //showSoundTracks

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent intent) {
    super.onActivityResult(requestCode, resultCode, intent);

    if ((requestCode == 0) && (resultCode == Activity.RESULT_OK)) {

        String selectedImage = intent.getDataString();

        Toast.makeText(this, selectedImage, 1).show();

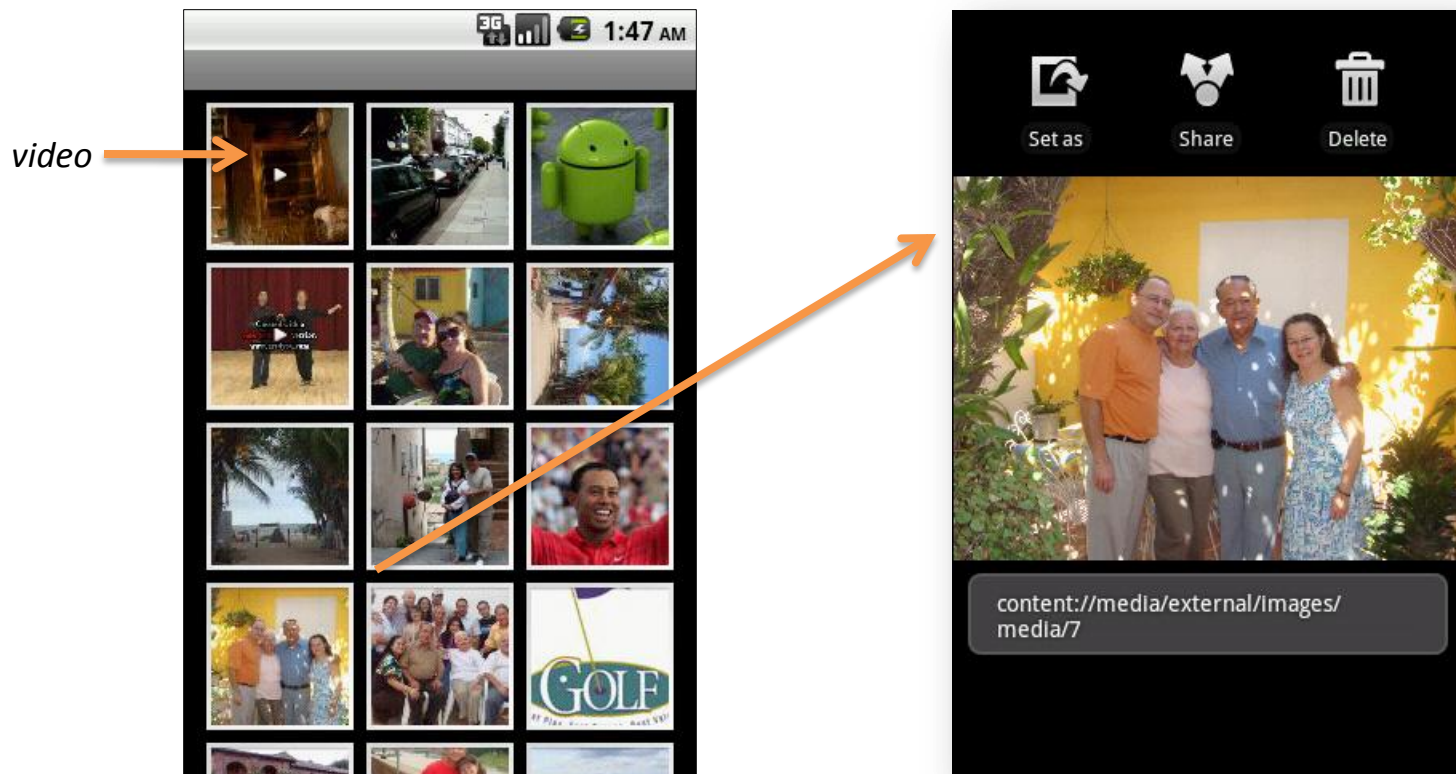
        // show a 'nice' screen with the selected image
        Intent myAct3 = new Intent(Intent.ACTION_VIEW, Uri.parse(selectedImage));
        startActivity(myAct3);
    }
} //onActivityResult
```

*All videos and all still images*



# Intents

**Example3.** Showing Pictures and Video - Calling a sub-activity, receiving results.

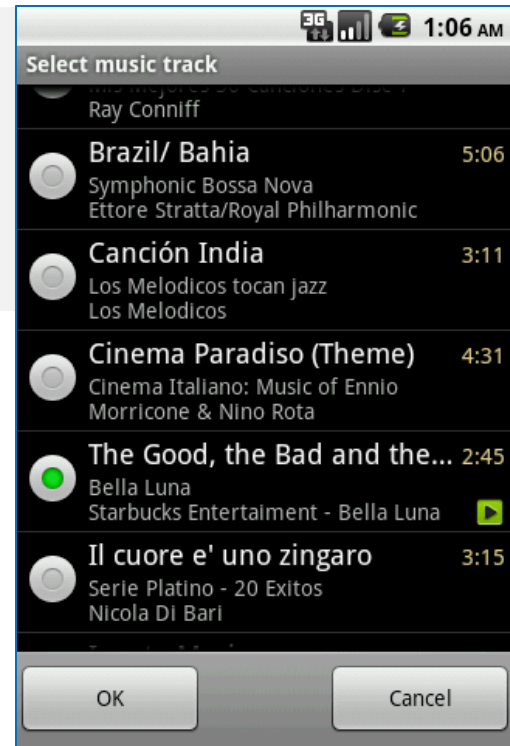


# Intents

## Example4. Showing/Playing Sound Tracks - Calling a sub-activity, receiving results.

```
private void showSoundTracks() {
    Intent myIntent = new Intent();
    myIntent.setType("audio/mp3");
    myIntent.setAction(Intent.ACTION_GET_CONTENT);
    startActivityForResult(myIntent, 0);
} //showSoundTracks
```

The returned string value is similar to the following  
 “content://media/external/audio/media/14”  
 ACTION\_VIEW on that Uri would produce a result  
 similar to the image on the right





# Intents

## Questions ?



# Intents

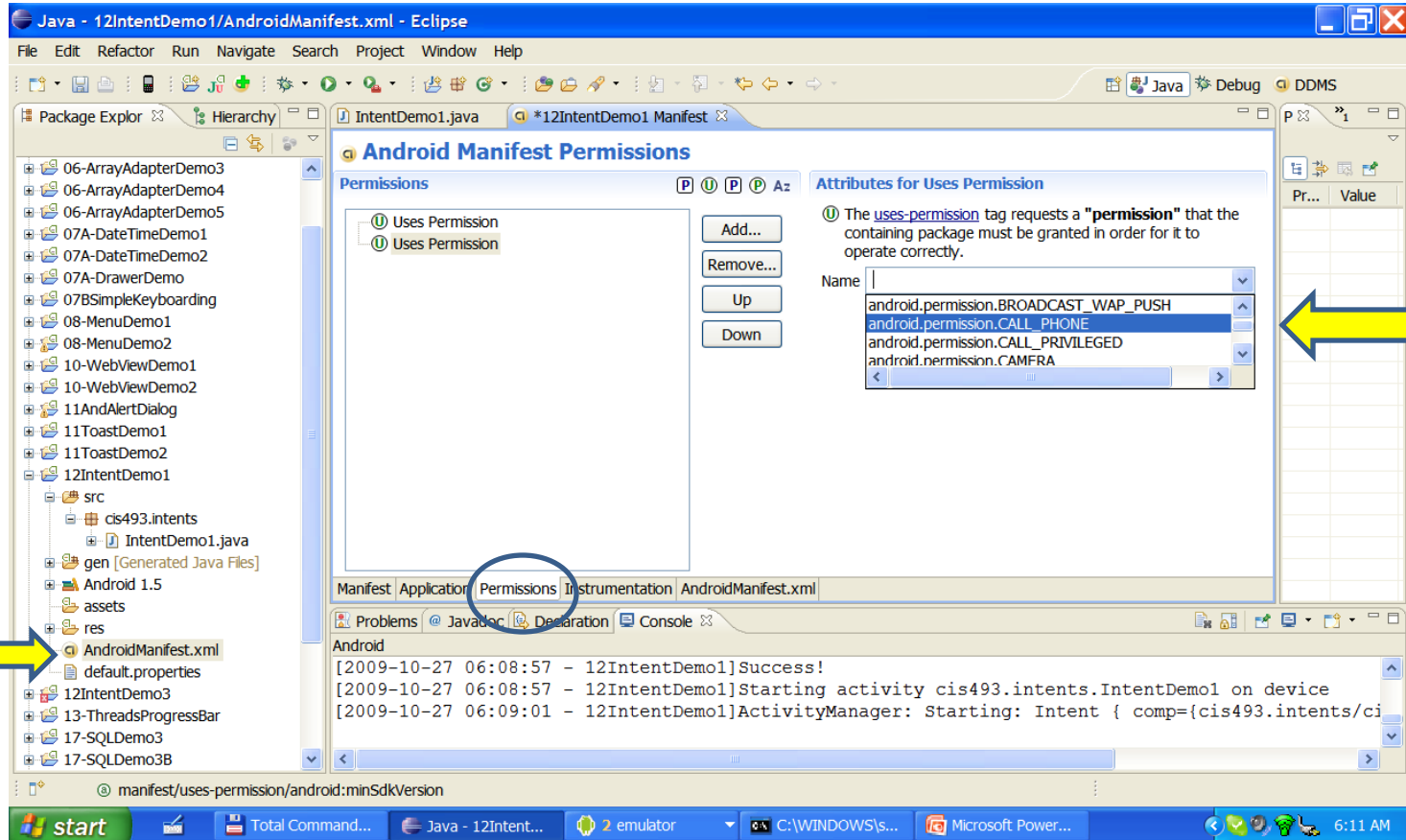
## Built-in Standard Broadcast Actions

List of standard actions that Intents can use for receiving broadcasts (usually through *registerReceiver(BroadcastReceiver, IntentFilter)* or a *<receiver>* tag in a manifest).

ACTION_TIME_TICK ACTION_TIME_CHANGED ACTION_TIMEZONE_CHANGED ACTION_BOOT_COMPLETED ACTION_PACKAGE_ADDED ACTION_PACKAGE_CHANGED ACTION_PACKAGE_REMOVED ACTION_UID_REMOVED ACTION_BATTERY_CHANGED	
---	--

# Intents

## Appendix: Getting Permissions



Becomes:

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