

Mobile and Wireless Systems Programming

Application Fundamentals



- Java programming language
- Android package : .apk
- Dalvik VM executes files in the Dalvik Executable (.dex) format
- 1 .apk = 1 application
- 1 application = 1 Linux process
- 1 application = 1 virtual machine

- A visual user interface
- Implemented as a subclass of `android.app.Activity`
- Many activities in the same project
- Each activity is independent of the others
- 1 Activity = 1 window

- No visual UI
- Runs in the background for an indefinite period of time
- Extends `android.app.Service`

- Receive and react to broadcast announcements
- No visual UI
- Extends `android.content.BroadcastReceiver`
- For example : announcements that the battery is low, that a picture has been taken, or that the user changed a language preference. . .

- Makes a specific set of the application's data available to other applications
- Data can be stored in the file system, in an SQLite database. . .
- Extends `android.content.ContentProvider`

Intent

- An abstract description of an operation to be performed
- Can be caused with :
 - `startActivity`
 - `broadcastIntent`
 - `startService` or `bindService`

Intent

- Intent()
- Intent(Intent o)
- Intent(String action)
- Intent(String action, Uri uri)
- Intent(Context packageContext, Class<?> cls)
- Intent(String action, Uri uri, Context packageContext, Class<?> cls)

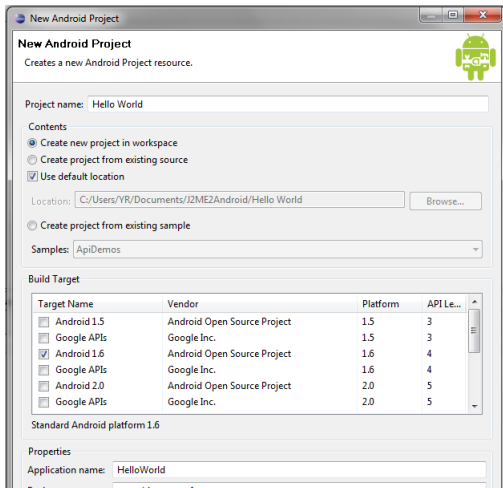
```
Intent myIntent = new Intent(CurrentActivity.this,  
NextActivity.class);  
CurrentActivity.this.startActivity(myIntent);
```

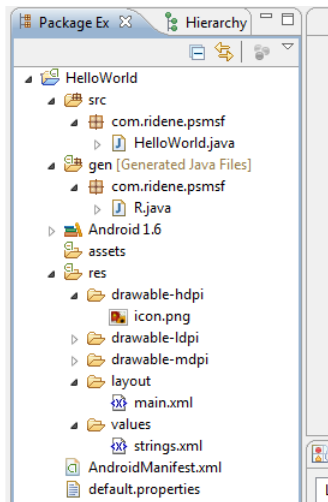

Intent

- Action :
 - ACTION_VIEW
 - ACTION_MAIN
 - ACTION_EDIT
 - ACTION_DIAL
 - ...
- Uri :
 - tel :123
 - http ://www.uni-pau.fr
 - content ://contacts/people/1
 - ...

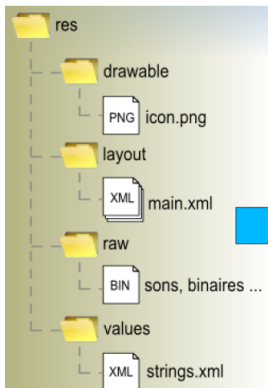
```
Intent viewIntent = new Intent("android.intent.action.VIEW",  
Uri.parse("http ://www.univ-pau.fr"));  
startActivity(viewIntent);
```

Project wizard





Resources



```
public final class R {  
  
    public static final class drawable {  
        public static final int icon=0x7f01000;  
    }  
  
    public static final class layout {  
        public static final int main=0x7f02000;  
    }  
  
    public static final class raw {  
        public static final int music=0x7f03000  
    }  
  
    public static final class values {  
        public static final int strings=0x7f04000;  
    }  
}
```

Resources

- asset (AssetManager)
- Instantiate layout elements at runtime

```
Button myButton = (Button) findViewById(R.id.my_button);  
TextView myTextView = (TextView)  
findViewById(R.id.my_textView);
```

- AndroidManifest.xml
- Contain all the application components (Activity, Permissions, SDK version...)

```
<activity>...</activity>  
<service>...</service>  
<receiver>...</receiver>  
<provider>...</provider>
```

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest ...>
  <application ...>
    <activity android:name="com.uppa.m2.FirstActivity"
              android:icon="@drawable/icon.png"
              android:label="@string/firstLabel"
              ...>
    </activity>
    ...
  </application>
</manifest>
```

Activity

- Android Activity is a equivalent to a MIDP MIDlet but !
- GUI and User inupts
- One Activity for each screen
- Lifecycle :
 - void onCreate(Bundle savedInstanceState)
 - void onStart()
 - void onRestart()
 - void onResume()
 - void onPause()
 - void onStop()
 - void onDestroy()

Activity

```
package com.uppa.helloandroid;  
  
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);  
    }  
}
```

Declare UI elements in XML (Wysiwyg editor)

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
    <TextView android:id="@+id/my_textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello , u l u a m u a u TextView" />
    <Button android:id="@+id/my_button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello , u l u a m u a u Button" />
</LinearLayout>
```

UI Builder

The screenshot displays the Android Studio UI Builder interface. The central canvas shows a dark-themed clock application with a white clock face and a button labeled "@+id/Button01". Below the button is a checkbox labeled "@+id/CheckBox01", which is highlighted with a red box. The left sidebar shows the project structure, including folders for resources and layout. The right sidebar shows the Outline view, listing the UI components: LinearLayout, AnalogClock01 (Analog), Button01 (Button), and CheckBox01 (CheckBox). The bottom panel shows the Properties view for the selected CheckBox component, with a table of its attributes.

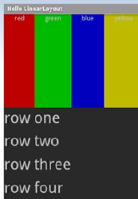
Property	Value
CheckBox	
Auto link	
Background	
Buffer type	
Button	
Checked	
Clickable	
Content description	
Cursor visible	

Components

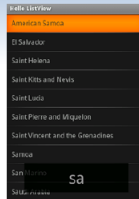
- Layouts
- Widgets
- Dialog
- Menus

Layout

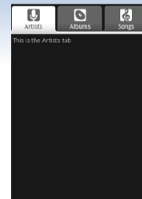
LinearLayout



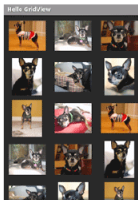
ListView



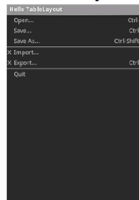
TabHost



GridView



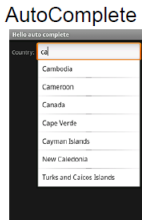
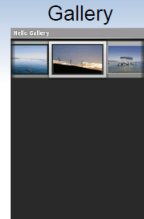
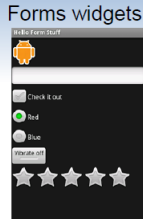
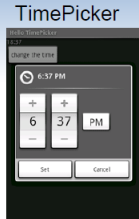
TableLayout



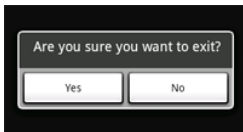
RelativeLayout



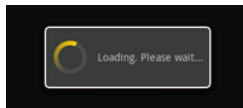
Widgets



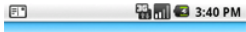
AlertDialog



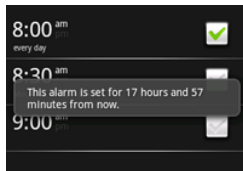
ProgressDialog



Notification



Toast



User interaction

- KeyPad
 - ClickListener (onClick(View))
 - KeyListener (onKeyUp(KeyEvent), onKeyDown(KeyEvent))
- TouchScreen
 - TouchListener (onTouchEvent(MotionEvent))