



# 16

## Android External Resources

Victor Matos  
Cleveland State University

Notes are based on:  
The Busy Coder's Guide to Android Development  
by Mark L. Murphy  
Copyright © 2008-2009 CommonsWare, LLC.  
ISBN: 978-0-9816780-0-9  
&  
Android Developers  
<http://developer.android.com/index.html>



### 16. Android – Resources

## Android Resources

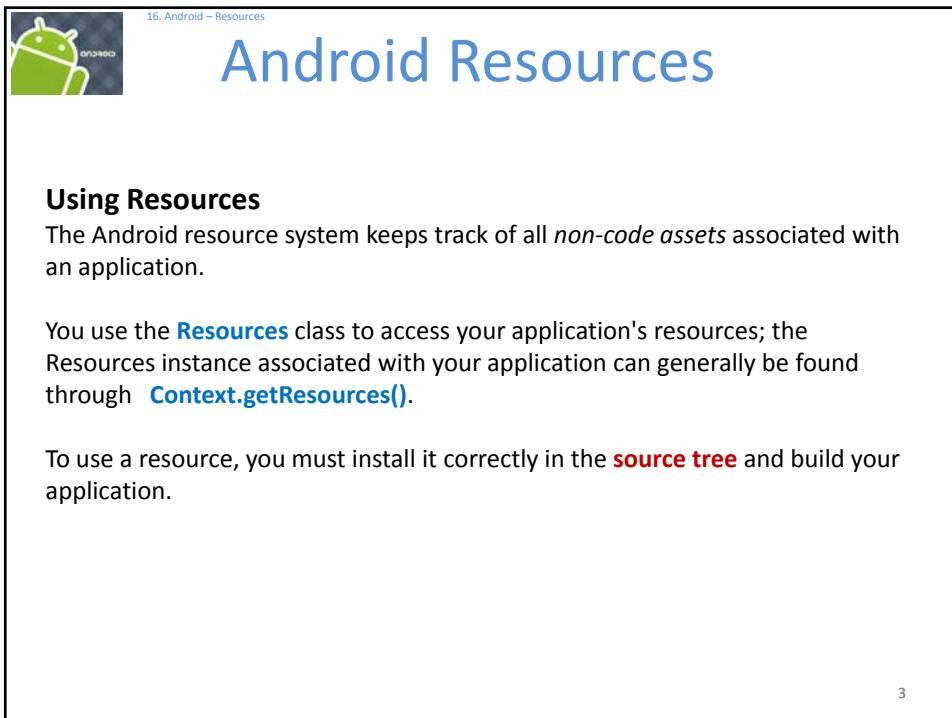
### Resources and Internationalization

Resources are external files (that is, non-code files) that are used by your code and compiled into your application at build time.

Android supports a number of different kinds of resource files, including XML, PNG, and JPEG files.

Resources are externalized from source code, and XML files are compiled into a binary, fast loading format for efficiency reasons. Strings, likewise, are compressed into a more efficient storage form.

<http://developer.android.com/guide/topics/resources/resources-i18n.html>



**16. Android – Resources**

# Android Resources

## Using Resources

The Android resource system keeps track of all *non-code assets* associated with an application.

You use the **Resources** class to access your application's resources; the Resources instance associated with your application can generally be found through **Context.getResources()**.

To use a resource, you must install it correctly in the **source tree** and build your application.

3



**16. Android – Resources**

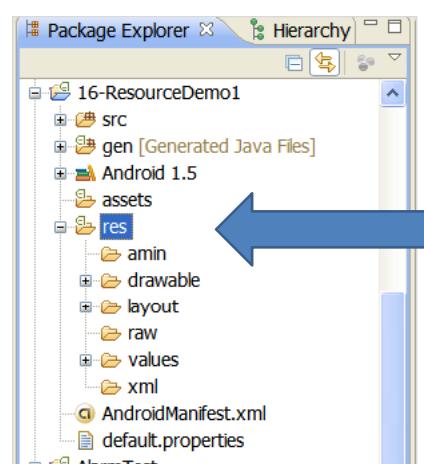
# Android Resources

## Copy/Paste Resources

You will create and store your resource files under the appropriate subdirectory under the **res/** directory in your project.

**Resources are compiled into the final APK file.**

Android creates a wrapper class, called **R**, that you can use to refer to these resources in your code. R contains subclasses named according to the path and file name of the source file



4

16. Android – Resources

## Android Resources

Directory	Resource Types
<code>res/anim/</code>	XML files that are compiled into <a href="#">frame by frame animation</a> or <a href="#">tweened animation</a> objects
<code>res/drawable/</code>	.png, .9.png, .jpg files. To get a resource of this type, use <code>mContext.getResources().getDrawable(R.drawable.imageId)</code>
<code>res/layout/</code>	XML files that are compiled into screen layouts (or part of a screen).
<code>res/values/</code>	XML files that can be compiled into many kinds of resource.  <code>arrays.xml</code> to define arrays <code>colors.xml</code> to define <a href="#">color drawables</a> and <a href="#">color string values</a> . Use <code>Resources.getDrawable()</code> and <code>Resources.getColor()</code> , respectively, to get these resources. <code>dimens.xml</code> to define <a href="#">dimension value</a> . Use <code>Resources.getDimension()</code> to get these resources. <code>strings.xml</code> to define <a href="#">string</a> values (use either <code>Resources.getString</code> or preferably <code>Resources.getText()</code> to get these resources. <code>getText()</code> will retain any rich text styling which is usually desirable for UI strings. <code>styles.xml</code> to define <a href="#">style</a> objects.
<code>res/xml/</code>	Arbitrary XML files that are compiled and can be read at run time by calling <a href="#">Resources.getXML()</a> .
<code>res/raw/</code>	Arbitrary files to copy directly to the device. They are added uncompiled to the compressed file that your application build produces. To use these resources in your application, call <a href="#">Resources.openRawResource()</a> with the resource ID, which is <code>R.raw.somefilename</code>

5



 16. Android – Resources

# Resources

**Examples.**

To see a number of samples you should explore the folder:  
**c:\Android\platforms\android-1.5\data\res\**

Name	Ext	Size	Date
[dir]			
[anim]	[values-fr-rF]	[values-mcc230-zh-rCN]	[values-mcc260-ay]
[color]	[values-fr-rBL]	[values-mcc223]	[values-mcc260-zh-rCN]
[drawable]	[values-fr-rCA]	[values-mcc230-zh-rCN]	[values-mcc260-zh-rCN]
[drawable-land]	[values-fr-rCH]	[values-mcc232-de]	[values-mcc262-ay]
[layout]	[values-hu-rL]	[values-mcc232-de]	[values-mcc262-eS]
[layout-port]	[values-hu-rR]	[values-mcc232-eO]	[values-mcc262-dA]
[raw]	[values-hu-rH]	[values-mcc232-eS]	[values-mcc262-dU]
[raw-arr]	[values-hu-rH]	[values-mcc232-eT]	[values-mcc262-eG]
[raw-cs]	[values-id-rD]	[values-mcc232-iD]	[values-mcc262-eS-iS]
[raw-dm]	[values-mcc232-iL]	[values-mcc232-kO]	[values-mcc262-iG]
[raw-dk]	[values-mcc232-kH]	[values-mcc232-kO]	[values-mcc262-iG]
[raw-en]	[values-it-rT]	[values-mcc232-rE]	[values-mcc262-kH]
[raw-eS]	[values-ja-rP]	[values-mcc232-oI]	[values-mcc262-kO]
[raw-IT]	[values-ja-rP]	[values-mcc232-oP]	[values-mcc262-kO]
[raw-it]	[values-ko-rK]	[values-mcc232-oT]	[values-mcc262-pT]
[raw-md]	[values-it-rT]	[values-mcc232-nD]	[values-mcc262-pR]
[raw-nD]	[values-it-rT]	[values-mcc232-nV]	[values-mcc262-pT]
[raw-nL]	[values-mcc204-eI]	[values-mcc232-th-rC]	[values-mcc262-pU]
[raw-nR]	[values-mcc204-eI]	[values-mcc232-th-rC]	[values-mcc262-pU]
[raw-nG]	[values-mcc204-eS]	[values-mcc232-zh-rT]	[values-mcc262-tr]
[raw-nL]	[values-mcc204-dB]	[values-mcc234]	[values-mcc262-zh-rCN]
[raw-nR]	[values-mcc204-dB]	[values-mcc234]	[values-mcc262-zh-rCN]
[raw-nG]	[values-mcc204-dB]	[values-mcc234-dB]	[values-mcc262-dU]
[raw-nL]	[values-mcc204-dB]	[values-mcc234-de]	[values-mcc262-pT]
[raw-nR]	[values-mcc204-eS]	[values-mcc234-eI]	[values-rf]
[raw-nG]	[values-mcc204-eS]	[values-mcc234-eS]	[values-rf-iB]
[raw-nL]	[values-mcc204-dC]	[values-mcc234-eT]	[values-rf-iL]
[raw-nR]	[values-mcc204-dC]	[values-mcc234-eT]	[values-rf-iL]
[raw-nG]	[values-mcc204-zh-rCN]	[values-mcc234-eV]	[values-sl-SJ]
[values]	[values-mcc204-eI]	[values-mcc234-eT]	[values-sl-pL]
[values-af-rEG]	[values-mcc204-eI]	[values-mcc234-iA]	[values-pd]
[values-af-rE]	[values-mcc204-eI]	[values-mcc234-iA]	[values-pd]
[values-af-rE]	[values-mcc204-pI]	[values-mcc234-iA]	[values-pd]
[values-af-rE]	[values-mcc204-pI]	[values-mcc234-pI]	[values-pd-pT]
[values-ca-rEQ]	[values-mcc204-pI]	[values-mcc234-pI]	[values-pd-pT]
[values-ca-rS]	[values-mcc204-pI]	[values-mcc234-pI]	[values-rf]
[values-ca-rCZ]	[values-mcc204-pI]	[values-mcc234-pt-pT]	[values-rf]
[values-de-rD]	[values-mcc204-pI]	[values-mcc234-pt-pT]	[values-rf-iR]
[values-de-rDK]	[values-mcc204-zh-rCN]	[values-mcc234-eV]	[values-sl-SJ]
[values-de-rD]	[values-mcc204-zh-rCN]	[values-mcc234-eV]	[values-sl-SJ]
[values-de-rT]	[values-mcc204-zh-rCN]	[values-mcc234-iR]	[values-sl-pL]
[values-de-rA]	[values-mcc204-zh-rCN]	[values-mcc234-iR]	[values-sl-pL]
[values-de-rC]	[values-mcc230-eC]	[values-mcc234-zh-rT]	[values-sl-pL]
[values-de-rDE]	[values-mcc230-eD]	[values-mcc260]	[values-th-rT]
[values-de-rL]	[values-mcc230-eD]	[values-mcc260-cS]	[values-il-pH]
[values-de-rC]	[values-mcc230-eD]	[values-mcc260-de]	[values-tr-rT]
[values-de-rR]	[values-mcc230-eD]	[values-mcc260-de]	[values-tr-rT]
[values-en-rU]	[values-mcc230-eU]	[values-mcc260-eI]	[values-uk-rU]
[values-en-rCA]	[values-mcc230-f]	[values-mcc260-eS]	[values-vi-VN]
[values-en-rGB]	[values-mcc230-jA]	[values-mcc260-eS-iS]	[values-zh-rCN]
[values-en-rD]	[values-mcc230-jA]	[values-mcc260-eT]	[values-zh-rCN]
[values-en-rH]	[values-mcc230-kO]	[values-mcc260-iD]	[xml]
[values-en-rH]	[values-mcc230-mI]	[values-mcc260-jS]	[xml]
[values-en-rS]	[values-mcc230-pI]	[values-mcc260-kO]	[values-mcc260-kO]
[values-en-rS]	[values-mcc230-pI]	[values-mcc260-kO]	[values-mcc260-kO]
[values-en-rZ]	[values-mcc230-pT]	[values-mcc260-pI]	[values-mcc260-pI]
[values-en-rS]	[values-mcc230-pT]	[values-mcc260-pI]	[values-mcc260-pI]
[values-en-rU]	[values-mcc230-pT]	[values-mcc260-pI]	[values-mcc260-pI]

7

 16. Android – Resources

# Resources

**More Examples.**

Try to install the **ApiDemos** application. Explore its resource folder. Find the source code in the folder:  
**c:\Android\platforms\android-1.6\samples\**

**How to install the App:**  
File -> New -> Project -> **Android Project** -> Next  
Select "Create project from existing source"  
Select the **ApiDemos** folder (all remaining fields will be self adjusted)



The screenshot shows the Eclipse IDE's Package Explorer view. The project structure is as follows:

- ApiDemos
  - src
  - gen [Generated Java Files]
  - Android 1.6
  - assets
  - res
    - anim
    - drawable
    - drawable-hdpi
    - drawable-ldpi
    - drawable-nodpi
    - layout
    - menu
    - raw
    - values
    - values-large
    - values-large-long
    - values-large-notlong
    - values-long
    - values-normal
    - values-normal-long
    - values-normal-notlong
    - values-notlong
    - values-small
    - values-small-long
    - values-small-notlong
    - xml
  - tests
  - AndroidManifest.xml

8

**16. Android – Resources**

# Android Resources

## Java Statements for Using Resources

Displaying a screen layout:

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

```
setContentView(R.layout.screen2);
```

9

**16. Android – Resources**

# Android Resources

## Java Statements for Using Resources

Retrieving **String** Resources from: **res/values/...**

```
/res/values/strings.xml
```

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
<string name="hello">Hola Mundo!, ResourceDemo1!</string>
<string name="app_name">ResourceDemo1</string>
<string name="good_bye">Hasta luego</string>
<string name="color_caption">Color:</string>
<string name="color_prompt">Seleccione un Color</string>
<string name="planet_caption"><b>Planeta</b> Planeta <i>Planeta</i> Planeta:</string>
<string name="planet_prompt">Seleccione un Planeta</string>
</resources>
```

```
String msg =
    this.getString(R.string.color_prompt);
```

10

 16. Android – Resources

# Android Resources

## Java Statements for Using Resources

Enhancing externalized **String** resources from: **res/values/...**

**/res/values/strings.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
<string name="hello">Hola Mundo!, ResourceDemo1!</string>
<string name="app_name">ResourceDemo1</string>
<string name="good_bye">Hasta luego</string>
<string name="color_caption">Color:</string>
<string name="color_prompt">Seleccione un Color</string>
<string name="planet_caption">
    <b>Planeta</b> Planeta <i>Planeta</i> <u>Planeta: </u></string>
<string name="planet_prompt">Seleccione un Planeta</string>
</resources>
```



As in HTML a string using **<b>**, **<i>**, **<u>** modifiers will be rendered in: bold, italics, and, underlined modes. In our example:

**Planeta Planeta Planeta Planeta**

11

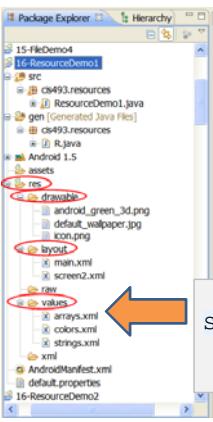
 16. Android – Resources

# Android Resources

## Java Statements for Using Resources

Retrieving **Array** Resources from: **res/values/...**

**/res/values/arrays.xml**



```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string-array name="planets">
        <item>Mercury</item>
        <item>Venus</item>
        <item>Earth</item>
        <item>Mars</item>
        <item>Jupiter</item>
        <item>Saturn</item>
        <item>Uranus</item>
        <item>Neptune</item>
        <item>Pluto</item>
    </string-array>
    <string-array name="colors"> ←
        <item>red</item>
        <item>orange</item>
        <item>yellow</item>
        <item>green</item>
        <item>blue</item>
        <item>violet</item>
    </string-array> ←
</resources>
```

**String myColors[] =  
    this.getResources().getStringArray(R.array.colors);**

12

 16. Android – Resources

# Android Resources

**Java Statements for Using Resources**

Retrieving a **drawable** image from: **res/drawable/...**



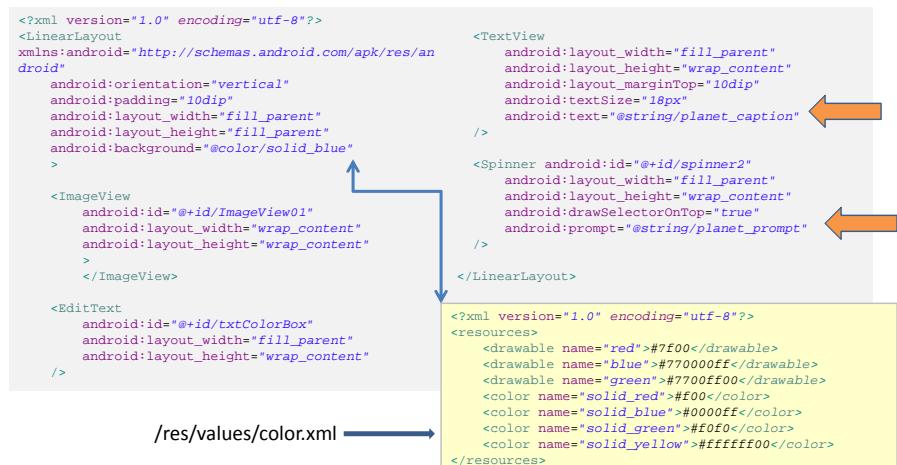
```
// same as xml layout attribute  
// android:src="@drawable/android_green_3d"  
  
imageView1.setImageResource(  
    R.drawable.android_green_3d);
```

13

 16. Android – Resources

# Android Resources

**Example1.** Using Embedded Resources (drawable, string, array).



```
<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout  
    xmlns:android="http://schemas.android.com/apk/res/android"  
        android:orientation="vertical"  
        android:padding="10dp"  
        android:layout_width="fill_parent"  
        android:layout_height="fill_parent"  
        android:background="@color/solid_blue"  
    >  
  
    <ImageView  
        android:id="@+id/ImageView01"  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
    </ImageView>  
  
    <EditText  
        android:id="@+id/txtColorBox"  
        android:layout_width="fill_parent"  
        android:layout_height="wrap_content"  
    />
```

**/res/values/color.xml**

```
<?xml version="1.0" encoding="utf-8"?>  
<resources>  
    <drawable name="red">#ff0000</drawable>  
    <drawable name="blue">#700000ff</drawable>  
    <drawable name="green">#7700ff00</drawable>  
    <color name="solid_red">#ff0000</color>  
    <color name="solid_blue">#0000ff</color>  
    <color name="solid_green">#00ff00</color>  
    <color name="solid_yellow">#ffff00</color>  
</resources>
```

16. Android – Resources

# Android Resources

**Example1.** Using Embedded Resources (drawable, string, array).

A blue arrow points from the left screenshot to the right one, indicating a transition or relationship between the two examples.

15

16. Android – Resources

# Android Resources

**Example1.** Using Embedded Resources (drawable, string, array).

```
// using Resources (adapted from Android - ApiDemos)
package cis493.resources;

import android.app.Activity;
import android.os.Bundle;
import android.widget.ArrayAdapter;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.Spinner;

public class ResourceDemo1 extends Activity {

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        //TRY: setContentView(R.layout.screen2);

        ImageView imageView1 = (ImageView)findViewById(R.id.ImageView01);
        //similar to xml layout android:src="@drawable/android_green_3d"
        imageView1.setImageResource(R.drawable.android_green_3d);
    }
}
```

16



16. Android – Resources

## Android Resources

**Example1.** Using Embedded Resources (drawable, string, array).

```

EditText txtColorBox = (EditText)findViewById(R.id.txtColorBox);

String msg = this.getString(R.string.color_caption);

String myColors[] = this.getResources().getStringArray(R.array.colors);

for ( int i=0; i<myColors.length; i++) {
    msg += "\n\t" + myColors[i];
}
txtColorBox.setText(msg);

Spinner s2 = (Spinner) findViewById(R.id.spinner2);
ArrayAdapter<CharSequence> adapter = ArrayAdapter.createFromResource(
    this,
    R.array.planets,
    android.R.layout.simple_spinner_item);

adapter.setDropDownViewResource(
    android.R.layout.simple_spinner_dropdown_item);
s2.setAdapter(adapter);
}
}

```

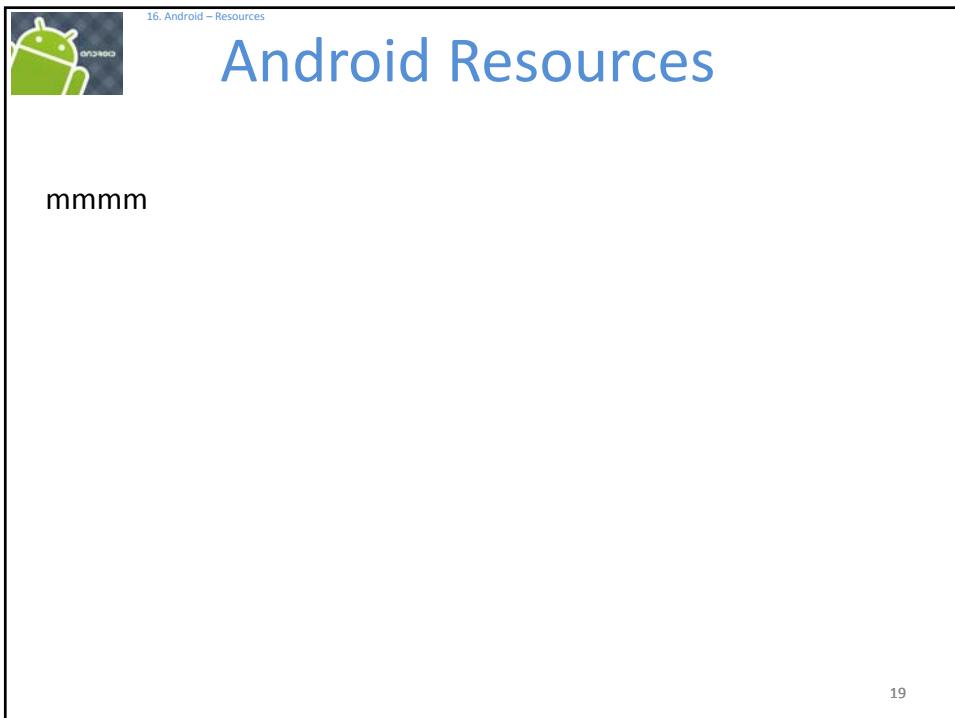


16. Android – Resources

## Android Resources

mmmm

18



16. Android – Resources

# Android Resources

mmmm

19

This slide is titled "Android Resources". It features the Android robot icon in the top left corner. The title "Android Resources" is centered in large blue font. Below the title is the placeholder text "mmmm". In the bottom right corner, there is a small number "19".

16. Android – Resources

# Android Resources

mmmm

20

This slide is titled "Android Resources". It features the Android robot icon in the top left corner. The title "Android Resources" is centered in large blue font. Below the title is the placeholder text "mmmm". In the bottom right corner, there is a small number "20".

