



河北师范大学软件学院
Software College of Hebei Normal University



Android 基础开发

第六章 资源的使用



Java与移动智能设备开发



教学目标



- 掌握Android中常见资源的使用方法



目录

- 1 XML资源概述
- 2 使用res目录下资源
- 3 使用assets目录下资源
- 4 本地化Android应用



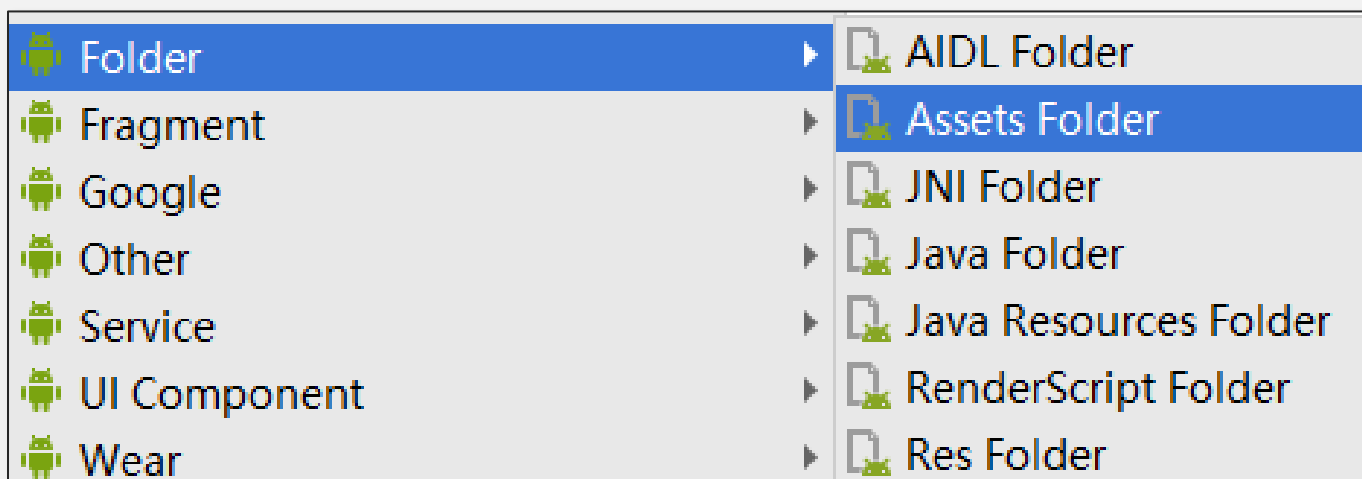
XML资源概述

- 为了实现Android应用MVC分离的目的，把应用中的各种资源单独进行分离，以实现资源的重复利用和代码维护上的高效。
- Android中的资源，一般分为两类：
 - **系统内置资源**：Android SDK中所提供的已经定义好的资源，用户可以直接拿来使用。
 - **用户自定义资源**：用户自己定义或引入的，只适用于当前应用的资源。



XML资源概述

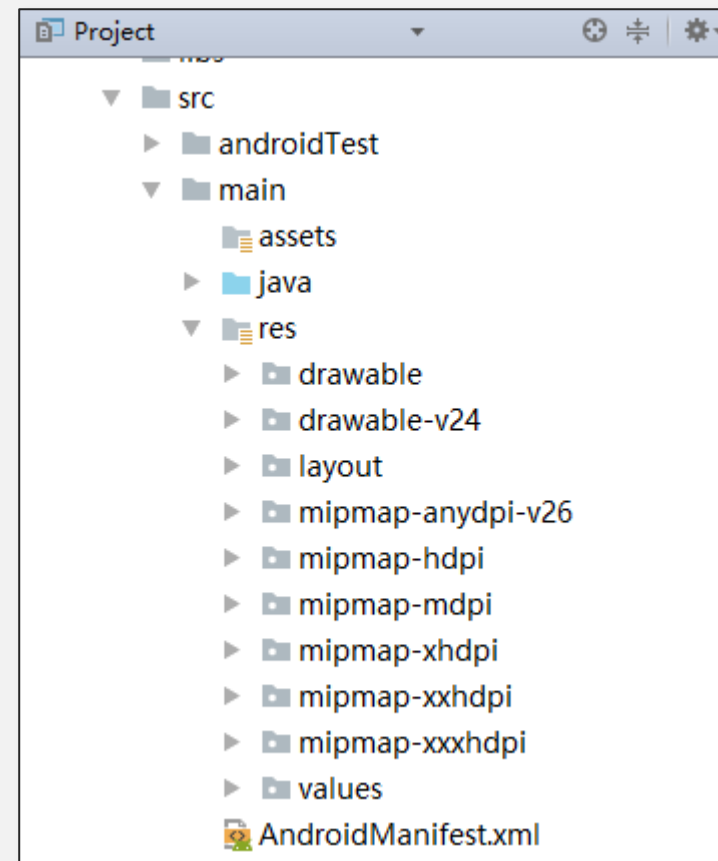
- Android应用中用到的各种用户自定义资源（包括字符串、颜色值、布局等）都集中放到**res目录（或assets）**下，以实现Android应用MVC分离的目的，而应用程序可以直接使用这些资源。
- Android Studio创建的项目中，默认不包含assets目录，需要手动创建。





XML资源概述

- **res目录**：可以使用R类访问的资源，放到该目录下。
- **assets目录**：无法直接访问的原生资源（只能通过AssetManager来处理）。





目录

- 1 XML资源概述
- 2 使用res目录下资源
- 3 使用assets目录下资源
- 4 本地化Android应用



res/目录下资源概述

- res目录可使用的资源有：

res子目录	可以存放的资源
res/anim	定义补间动画的XML文件
res/color	定义不同状态下颜色列表的XML文件
res/drawable	各种位图文件（png、jpg、gif、9-Patch） 可以编译成各种drawable对象的XML文件
res/mipmap	应用程序Launcher图标
res/layout	用户界面布局文件
res/menu	菜单资源布局文件（选择菜单、子菜单、上下文菜单）
res/raw	任意类型的原生资源
res/values	各种简单值的XML文件（包括字符串、整数、数组、尺寸等）
res/xml	其它任意的XML文件（可能没有特殊意义的XML文件）



res/目录下资源的使用

- 在Android中使用res目录下资源或系统内置资源：
 - 在其它资源文件中使用资源：
 - 使用res目录下资源：
`@[<pack_name>:]<res_type>/<res_name>`
 - 使用系统内置资源：`@android:<res_type>/<res_name>`
 - 在Java代码中使用资源：（使用资源的标识符）
 - 使用res目录下资源：`[<pack_name>.]R.res_type.res_name`
 - 使用系统内置资源：`android.R.res_type.res_name`



res/目录下资源的使用

- 在Android中使用res目录下或系统内置资源：
 - 在Java代码（Activity）中使用资源：
 - 使用R.res_type.res_name只是表示资源的标识符，并没有真正获得资源对象。
 - 真正获得资源对象的基本流程是：

```
Resources res = getResources();  
String appName = res.getString(R.string.app_name);  
Drawable drawable  
    = res.getDrawable(R.drawable.ic_launcher);
```



res/values/String.xml资源

- String资源是最简单的一类资源，用于存储程序中用到的字符串相关内容；String资源位于项目目录中的 **res/values/strings.xml** 文件中。
- 基本结构：

```
<resources>  
    <string name="app_name">HelloWorld</string>  
    <string name="user_name">张三</string>  
</resources>
```

- 根节点必须是<resources>元素；一个子元素<string>对应一个字符串表示。



颜色资源和尺寸资源

- Color资源基本结构：**res/values/colors.xml**

```
<resources>
    <color name="colorPrimary">#3F51B5</color>
</resources>
```

- 尺寸资源基本结构：**res/values/dimens.xml**

```
<resources>
    <dimen name="view_length">100dp</dimen>
</resources>
```



res/values/array.xml资源

- Array资源：把数组类信息存储到XML文件，方便统一管理这类资源。
- Array资源可以分为三类：
 - string-array：字符串组成的数组形式。
 - integer-array：int型数据组成的数组形式。
 - typed-array：一般数组形式（数据元素类型不限）。
- Array资源的使用：
 - 在XML文件中的定义：[res/values/arrays.xml](#)。
 - 在Activity文件中的使用。



res/values/array.xml资源

- Array资源的使用：
 - 在XML文件中的定义：
res/values/arrays.xml

```
<resources>
    <string-array name="user_name">
        <item>张三</item>
        <item>李四</item>
    </string-array>
    <integer-array name="age">
        <item>18</item>
        <item>20</item>
    </integer-array>
    <array name="pic">
        <item>@drawable/user1</item>
        <item>@drawable/user2</item>
    </array>
</resources>
```



res/values/array.xml资源

- Array资源的使用：
 - 在Activity文件中使用数组：

```
Resources res = getResources();  
String[] userName  
    = res.getStringArray(R.array.user_name);  
int[] age = res.getIntArray(R.array.age);  
TypedArray userPic = res.obtainTypedArray(R.array.pic);  
Drawable firstPic = userPic.getDrawable(0);
```

- 对于TypedArray类型，需要先获得其TypedArray对象，然后再从该对象中依次获得指定下标的元素。



res/layout/资源和res/menu/资源



- Layout资源是最常用的一类资源，其作用即是设定Activity的视图界面布局，该资源位于res/layout目录下。
- Menu资源是用来设置菜单布局的一类资源，该类资源位于res/menu目录下。



res/values/style资源

- Style资源可以设置layout视图组件中某一个视图元素的样式；该资源文件位于res/values目录下，文件名自定义。
 - 在res/values/目录下定义styles.xml文件（文件名任意）。

```
<resources>
  <style name="CustomText" >
    <item name="android:textSize">20sp</item>
    <item name="android:textColor">#FF4081</item>
  </style>
</resources>
```

- <style>元素中parent属性表明该样式继承的父级元素样式。



res/values/style资源

- 在res/layout/目录下的布局文件中使用当前资源。

```
<TextView  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Hello World!"  
    style="@style/CustomText" />
```

- 详细信息，请参考：

<http://www.android-doc.com/guide/topics/resources/style-resource.html>



res/values/theme资源

- Theme资源可以设置Activity窗口的样式；该资源文件位于res/values/目录下，文件名自定义。
 - Theme资源对某个Activity或整个Application起作用，而不是单独的视图组件。
 - Theme主题主要用来设置应用窗口的特征信息（如窗口标题、窗口背景、窗口边框等）。
- Theme资源的使用需要从两个角度考虑：
 - 在res/values/目录中定义主题资源文件。
 - 在AndroidManifest.xml文件或Activity中为整个Activity或整个Application应用主题。



res/values/theme资源

- 在res/values/目录下，定义主题资源文件（文件名任意，一般就命名为theme.xml）。

```
<resources >
    <style name="CustomTheme" parent="AppTheme">
        <item name="windowNoTitle">true</item>
        <item name="android:colorBackground">
            @color/custom_theme_color</item>
    </style>
</resources>
```

- <style>元素的parent属性，表明当前主题继承的父主题。
- 详细信息，请参考：

<http://www.android-doc.com/guide/topics/ui/themes.html>



res/values/theme资源

- 为应用程序应用主题：
 - 在AndroidManifest.xml文件静态绑定：
 - 可以为<application>元素添加 **android:theme**属性，属性值即为用户刚才所设计的theme资源（@style/CustomTheme）。
 - 可以为<activity>元素添加 **android:theme**属性，属性值即为用户刚才所设计的theme资源（@style/CustomTheme）。
 - 在Activity文件中动态绑定主题：
 - 使用Activity类的**setTheme(R.style.CustomTheme)**方法。
 - 该方法必须在setContentView()方法或getLayoutInflater().inflate()方法之前被调用。



res/mipmap资源

- 在res/目录下默认情况有六个mipmap子目录。
 - res/mipmap-anydpi-v26 : API26及以上使用自适应图标
 - res/mipmap-mdpi : 中分辨率图标
 - res/mipmap-hdpi、mipmap-xhdpi、mipmap-xxhdpi、mipmap-xxxhdpi : 高分辨率图标
 - 这六个子目录中的**图片名可以完全相同**，Android应用会根据用户设备分辨率的不同而自行选择。



res/drawable/资源

- Drawable资源泛指广义上的图片资源；在Android中Drawable资源的应用十分广泛。

类型	描述
Bitmap File	图片文件资源
9-patch File	基于适应内容而自动伸缩的图片（扩展名是 .9.png）
Layer List	一个XML文件，管理一系列图片资源的特殊资源
State List	一个XML文件，针对视图控件不同状态而设置的特殊drawable类型
Level List	一个XML文件，根据视图控件不同等级而现实不同的drawable资源
Shape Drawable	一个XML文件，定义包含颜色和渐变的几何图形
Scale Drawable	一个XML文件，根据当前对准值作相应的平铺处理
Transition Drawable	一个XML文件，用两张图片形成一个渐变效果的drawable资源



res/drawable/bitmap资源

- Bitmap资源，即原始图片（*.png、*.jpg、*.gif），在res/目录下默认情况有两个drawable子目录。
 - res/drawable：通用图片
 - res/drawable-v24：API24以上系统使用图片
 - 这两个子目录中的**图片名可以完全相同**，Android应用会根据用户设备API版本的不同而自行选择。



res/drawable/9patch资源

- 9-patch资源，一个扩展名是.9.png的图片，可以根据内容的大小，自动伸缩图片。
 - 使用示例：Button控件的背景，必须伸缩以适应Button的文字或图片。



res/drawable/LayerList资源

- LayerList资源，一个图层，用来管理一系列其它Drawable资源。
 - LayerList是res/drawable/目录下的一个XML文件。
 - XML文件基本语法定义为：





res/drawable/LayerList资源

```
<?xml version="1.0" encoding="utf-8"?>
<layer-list xmlns:android="http://schemas.android.com/apk/res/android">
    <item
        android:drawable="@drawable/pic1"
        android:id="@+id/resource_name"
        android:top="10dp"
        android:bottom="10dp"
        android:left="10dp"/>
    <item
        android:drawable="@drawable/pic2"
        android:id="@+id/resource_name1"
        android:top="20dp"
        android:bottom="20dp"
        android:left="20dp"/>
</layer-list>
```



res/drawable/StateList资源

- StateList：是一个定义在XML中的可绘制对象，使用几个不同的图像，根据对象的状态来呈现同一个图形。
 - StateList是res/drawable/目录下的一个XML文件。
 - XML文件基本语法定义为：

```
<selector
xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:drawable="@drawable/bg_pressed"
        android:state_pressed="true"/>
    <item android:drawable="@drawable/bg_normal"/>
</selector>
```



res/drawable/资源

- Drawable资源功能十分丰富，在此不再一一列举，具体请大家自行查阅Android开发者文档。

– 详细信息，参考：

<http://www.android-doc.com/guide/topics/resources/drawable-resource.html>



res/color/ColorStateList资源

- ColorStateList：在XML中定义并且作一个color来使用的对象，它会根据所使用它的View对象的状态来改变颜色。
 - 文件目录：**res/color/**目录下，文件名任意。
 - 文件基本结构：

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android" >
    <item
        android:color="hex_color"
        android:state_pressed=["true" | "false"]
        android:state_focused=["true" | "false"]
        android:state_selected=["true" | "false"]
        android:state_checkable=["true" | "false"]
        android:state_checked=["true" | "false"]
        android:state_enabled=["true" | "false"]
        android:state_window_focused=["true" | "false"] />
</selector>
```



res/color/ColorStateList资源

- ColorStateList :
 - 示例 :

```
<selector xmlns:android="http://schemas.android.com/apk/res/android">  
  <item android:state_pressed="true"  
    android:color="#ffff0000"/> <!-- pressed -->  
  <item android:state_focused="true"  
    android:color="#ff0000ff"/> <!-- focused -->  
  <item android:color="#ff000000"/> <!-- default -->  
</selector>
```

- 详细信息，请参考：

<http://www.android-doc.com/guide/topics/resources/color-list-resource.html>



res/xml/目录下资源

- 在res/xml/目录下的XML文件为任意规范的XML文档，这些文档没有其它附加意义，仅仅是存储数据或展示数据。
 - 这些XML文档，不能在其它XML资源中引用。
 - 在Activity中，可以使用Resources对象的getXML()方法，加载到内置的XML解析器中，以方便处理。
 - 该类资源的一个典型应用是：存储应用中使用到的配置信息。



res/raw/目录下资源

- res/raw/目录存储任意原始格式的文件（可能为.txt、.mp3、.flv等等）。
 - 该目录下的文件，一般不能使用在其它XML资源中。
 - 该目录中的文件，同样会被R类所索引；因此在Activity中可以使用R.raw.***方式引用资源；通过Resources对象的openRawResource()方法，可以获得原始对象的输入流，以方便后续使用。



目录

- 1 XML资源概述
- 2 使用res目录下资源
- 3 使用assets目录下资源**
- 4 本地化Android应用



assets/目录下资源简介

- assets/目录下也可以添加Android应用中所使用的资源，这些资源可以是任意格式的，且可以位于assets目录中的任意位置。
 - assets目录中可以建立子目录，建立更灵活的目录结构。
 - assets目录中的文件格式是任意的，不一定必须是XML文件。



assets/目录下资源简介

- assets/目录下的文件与res/raw/目录下文件区别：
 - assets/目录下文件不会被R类索引，即assets/目录下资源不能使用Resources对象获取。
 - assets/目录下资源不会被打包到APK中，即assets目录中适合存放过大的文件。



assets/目录下资源的使用

- 在Android中使用assets目录下资源：
 - assets目录下的资源**不会被R类处理**，因此不能使用res目录下资源的访问方式来访问。
 - 在Java代码中，一般是通过**getAssets()**方法获得**AssetManager**对象，然后再加载指定资源，处理该资源。

```
AssetManager assetManager = getAssets();
try {
    InputStream is = assetManager.open("test.txt");
} catch (IOException e) {
    e.printStackTrace();
}
```



assets/目录下资源的使用

- 在Android中使用assets目录下资源：
 - AssetManager对象中的常用方法：

方法名	描述
String[] list(String path)	返回path目录下所有文件组成的字符串数组形式（若path为空，则表示assets目录）
InputStream open(String filename)	打开指定fileName表示的文件流，返回该输入流
XmlResourceParser openXmlResourceParser(String filename)	从assets目录下的fileName文件中加载XML解析器
void close()	关闭当前AssetManager对象

- 详细信息，请参考：

<http://www.android-doc.com/reference/android/content/res/AssetManager.html>



目录

- 1 XML资源概述
- 2 使用res目录下资源
- 3 使用assets目录下资源
- 4 本地化Android应用**



国际化和本地化Android应用

- 我们编写的Android应用程序不仅在国内可以使用，也期望在国外得到推广；而不同国家使用的语言环境不一致，因此我们的Android应用应该适应**国际化和本地化**的能力。
 - 国际化：是指应用具有适用于任何地方的“潜力”。
 - 本地化：是指为了更适合于“特定”地方的使用，而另外增添的特色。



国际化和本地化Android应用

- 在Android中实现应用的国际化和本地化十分简单，只需要把应用的资源进行修改即可。
 - 这种操作的简洁性正是基于Android程序MVC分离的机制。
 - 实现Android国际化和本地化，需要做的就是，在res/目录下，创建适合特定国家的子目录即可（以values子目录为例，其它目录类似）。
- "values"目录，表示对于任何国家均适用，即若Android找不到某个国家的特别目录时，就采用当前默认目录中的资源。
- "values-语言代码-r国家代码"目录，表示某一个地区适用的资源目录（如values-zh-rCN表示中国大陆地区）。
- 常用的国家代码和语言代码，请参考：

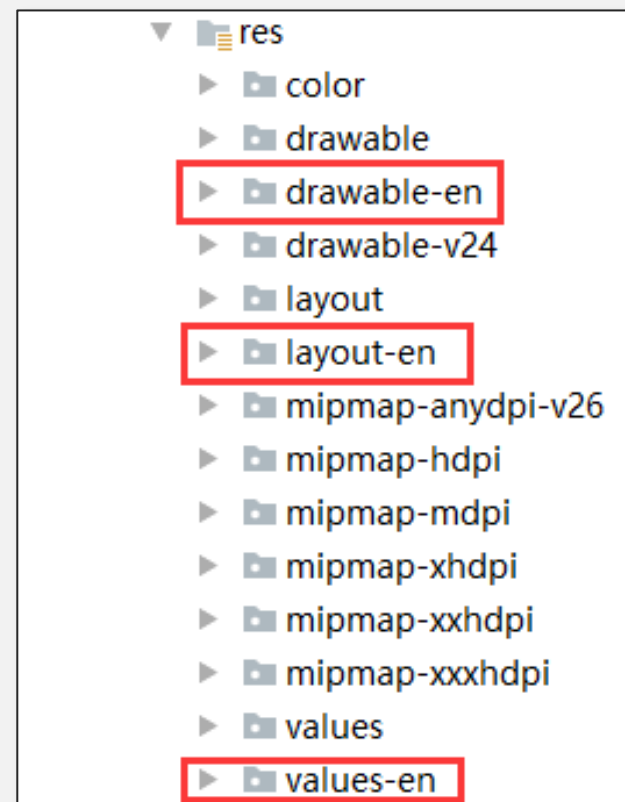
<http://www.cnblogs.com/Mien/archive/2008/08/22/1273950.html>



国际化和本地化Android应用



- 在各自的res/子目录中，定义当前地区所适用的资源文件即可完成本地化的实现。
- Android项目实现本地化后，可以**设置Android手机或模拟器的语言**，来测试在不同语言环境下，应用的本地化效果。

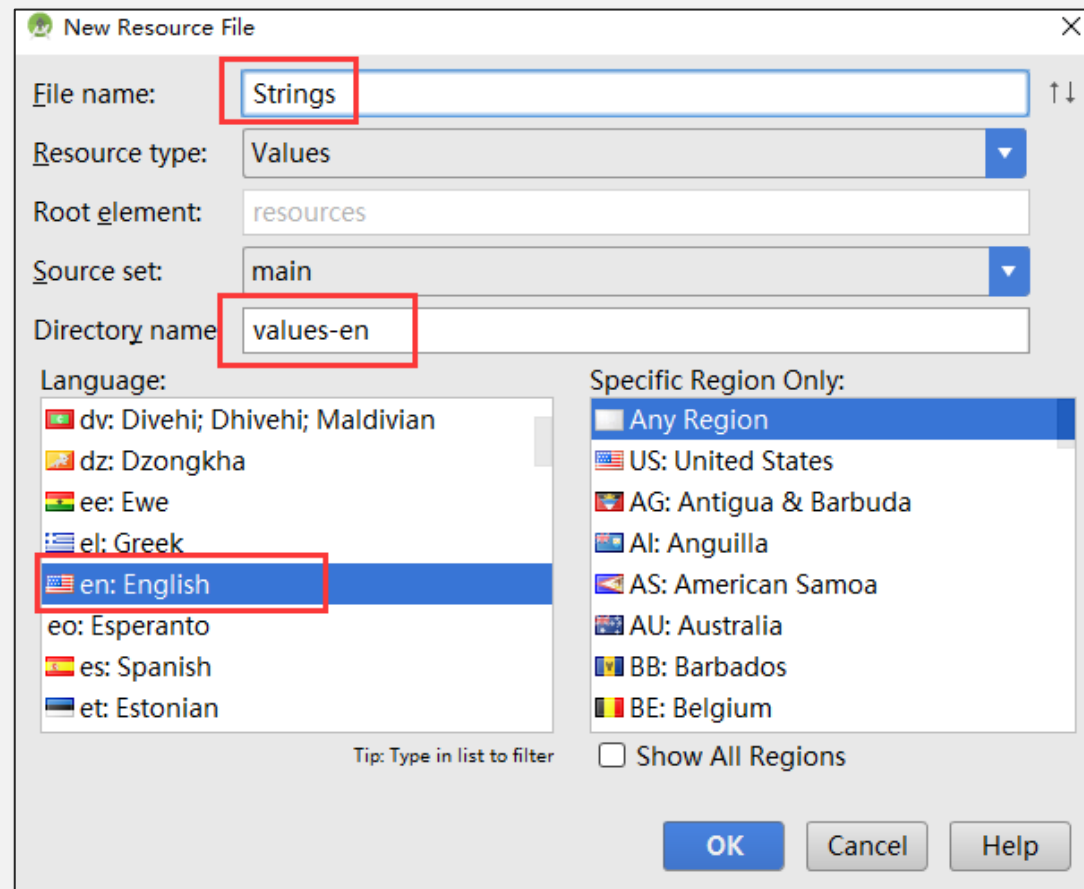
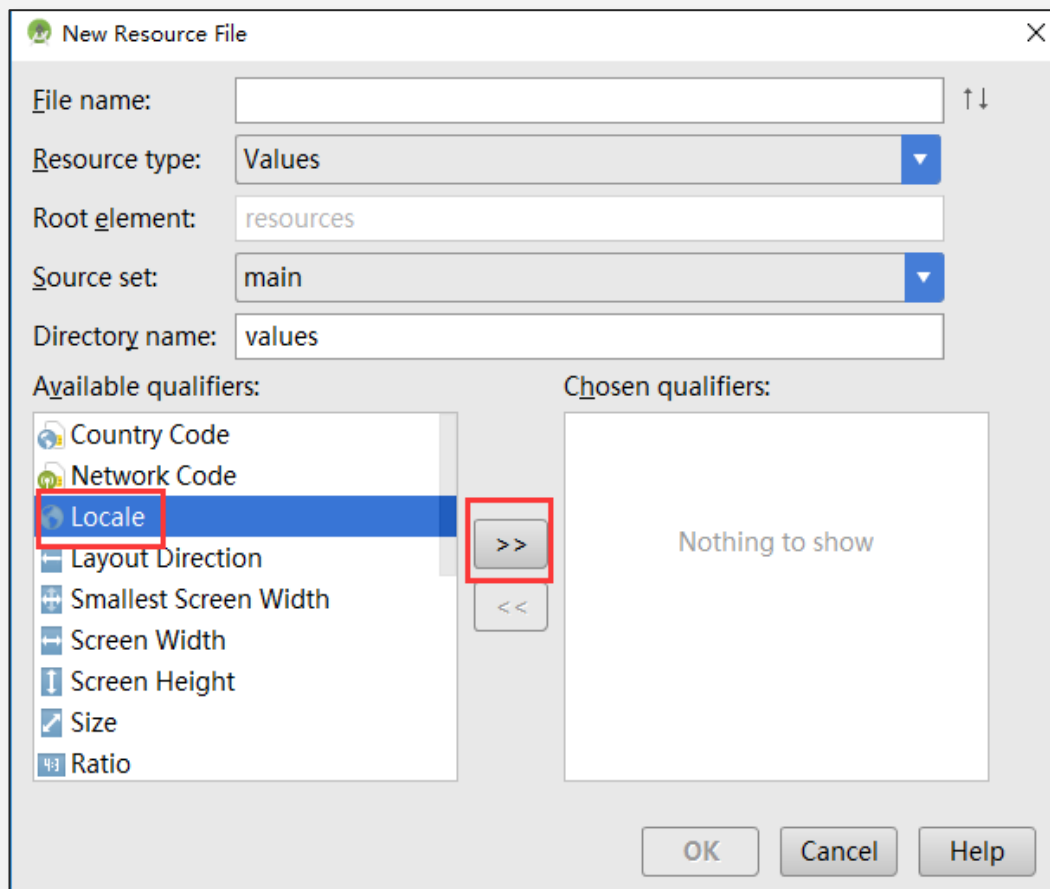




国际化和本地化Android应用



- 在res目录右击new->Android Resources File。





内容回顾

- 1 XML资源概述
- 2 使用res目录下资源
- 3 使用assets目录下资源
- 4 本地化Android应用



Thank you!

