Using Drawable and Values Resources

Jim Wilson

MOBILE SOLUTIONS DEVELOPER & ARCHITECT

@hedgehogjim   blog.jwhh.com

What to Expect from This Module

Resource Basics

Drawable Resources

Values Resources

Accessing Resources from Java Code

Resources

Externalize content from source code

- Maintained separately from code
- Simplify adaptability
  - More on this later in the series

Located under res project folder

Variety of resource types are supported

- Layout
- Menu
- Many others

Drawable Resource

Something that can be drawn to the screen

- Often used with ImageView, ImageButton
- Icons for menus, navigation drawer

Project location

- res/drawable

A variety of drawable types supported

Drawable Resource

Raster

- Non-scalable graphics file
- *.png preferred, can also use *.jpg, *.gif

Vector

- Scalable vector graphics file
- Work well for simple graphics
- Import with Vector Asset Studio

Values Resource

Allow storing values as resources

- Strings
- Colors
- Dimensions
- Integers
- Many more

Project location

- res/values

Values Resource Organization

Organized into XML files

- File name doesn’t matter
- A file can have a mix of specific types
- Can have multiple files

Declaring values resources

- Under a root element of resources
- Declared as child element
  - Element indicates resource type
  - Named by name attribute

Declaring Values Resources

```
<resources>
    <integer name="my_value">1000</integer>
    <!-- declare other values resources -->
</resources>
```

Accessing Resources From Java Code

Accessed through a Resources reference

- Available through any Context reference
  - Generally we use the current Activity
- Use getResources method

Accessing Resources From Java Code

Retrieving a specific resource

- Use Resources.getXXX methods
- Pass the resource’s name

Resource names in code

- Part of the generated R class
- Qualified by resource type
  - R.*type*.*name*
  - Example: R.integer.my_value
- Built-in resources in android.R class
  - android.R.*type*.*name*

Accessing Resources From Java Code

Accessing string resources

- Value accessible with Resources.getString
- Don't need Resources reference
- Can call getString directly from Context

Using String resources effectively

- Often not necessary to retrieve value
- Methods often have resource-friendly overload available

Summary

Resources

- Externalize content from source code

Summary

Drawable resources

- Something that can be drawn to screen
- Raster graphics
  - Non-scalable files such as *.png, etc.
- Vector graphics
  - Scalable vector graphics files
  - Import with Vector Asset Studio

Summary

Values resources

- Store values such as strings, colors, etc.
- Organized in XML files
- Element indicates resource type
- Resource named as part of declaration

Summary

Accessing resources from Java code

- Use Resources reference
  - Available from Context.getResources
- Access values with Resources.getXXX

String resources

- Accessible directly from Context
- Many methods accept string resource

Resource name

- Part of generated R class
- Qualified by resource type