BCS 371 Mobile Application Development I

Arthur Hoskey, Ph.D.
Farmingdale State College
Computer Systems Department

Today's Lecture

- Externalizing Resources

- Content From Textbook:

Professional Android by Reto Meier and Ian Lake

Published by John Wiley and Sons Inc., 2018

Externalizing Resources

- It is good practice to keep non-code resources external to your code.
- Examples of non-code resources:
  - Images
  - String constants
- Easier to maintain, update, and manage.
- Makes it easy to define alternative resources.

- **Dynamic Selection.** Android dynamically selects resources from resource trees that contain different values for alternative hardware configurations, languages, and locations.
- When an application starts Android automatically selects the correct resources without you having to write a line of code.
- You just have to define the resources for each situation.

Externalizing Resources

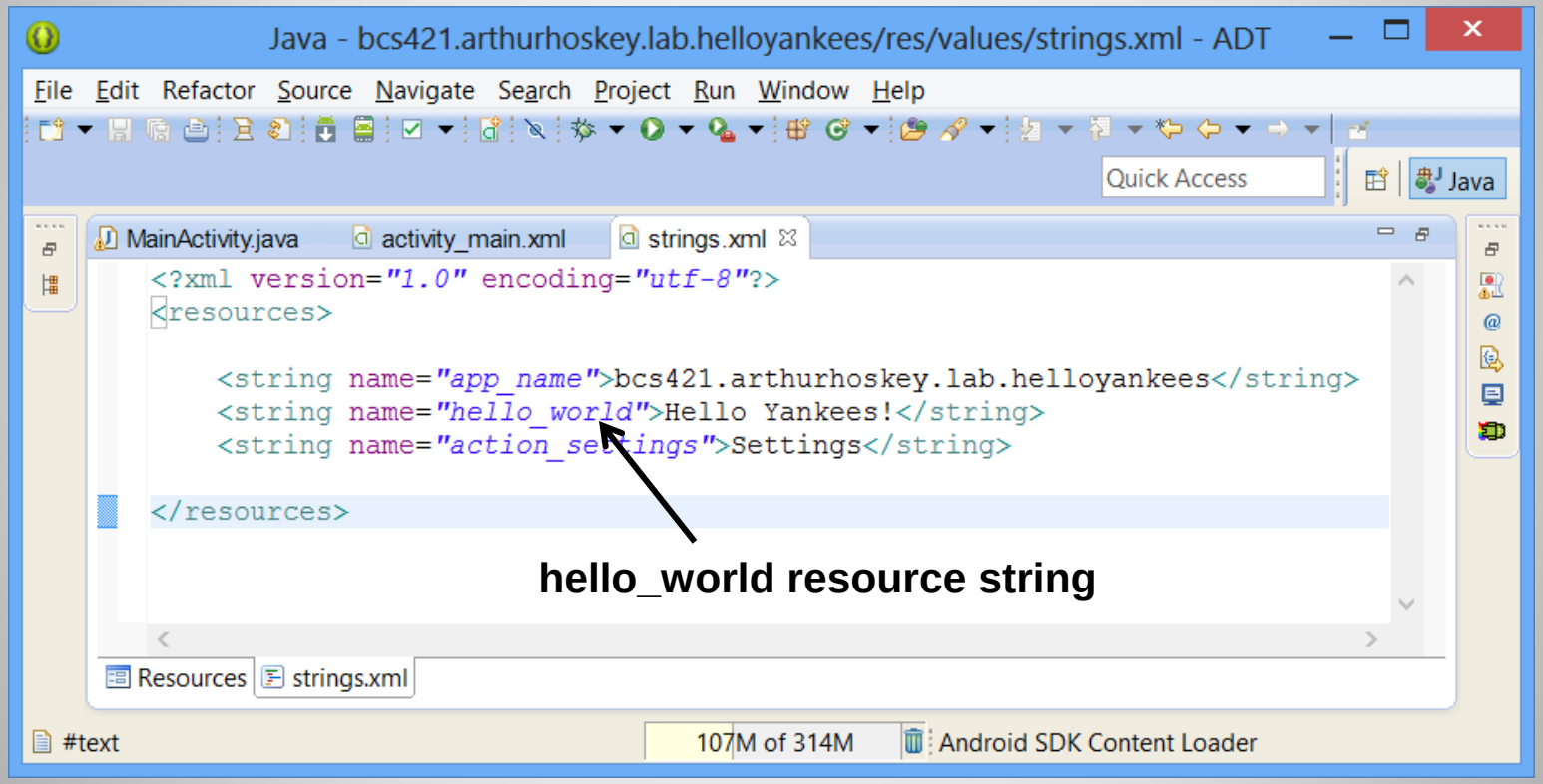
- Now on to string resources…

String Resources

String Resources

- All strings should be defined in the string resource file(s).
- **Always use string resources.** When referencing a string resource in your application you should use the string resource id (do not hard code strings in your app).
- It is better to use string resources as opposed to hardcoding strings into your app.
- For example…

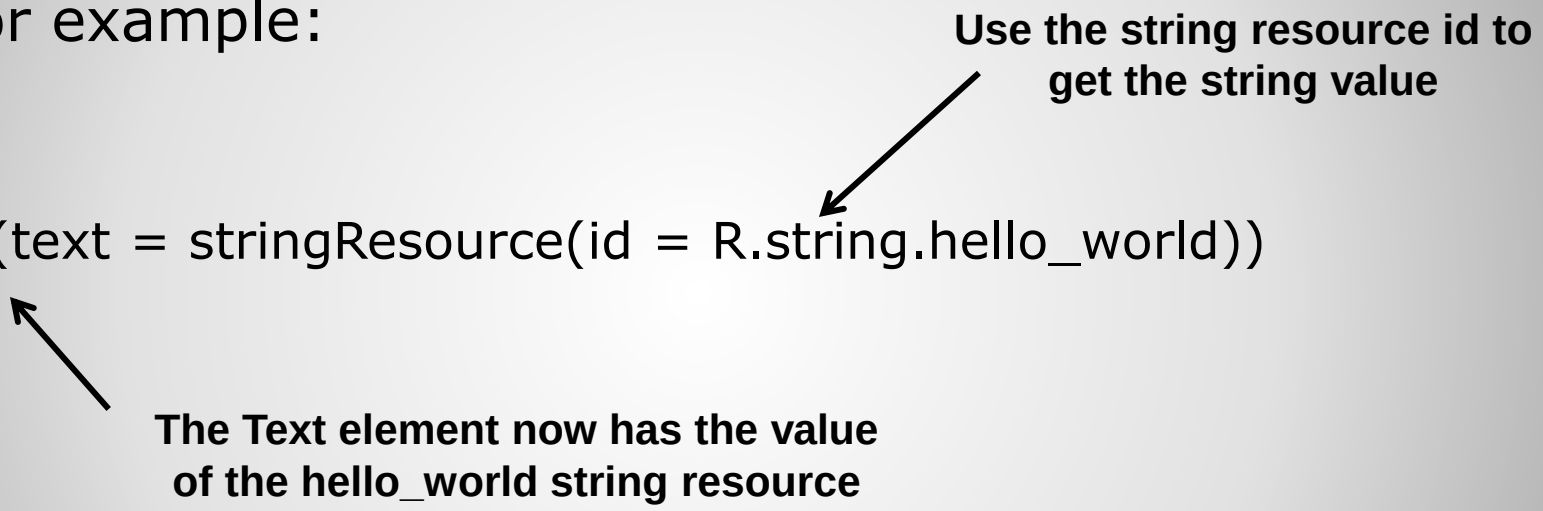
- Here is a string resource file that defines a resource string named hello_world.
- The value of hello_world is "Hello Yankees!".



Define String Resources

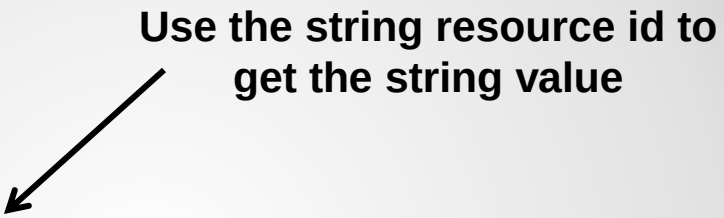
Using String Resource in Composable

- **Use string resource in composable code.**
- Use the **stringResource** function to access string resources in composable code.
- For example:

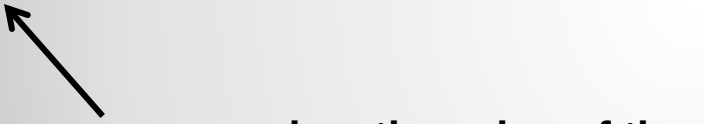


Using String Resources in Code

- **Use string resource in normal code.**
- Use the **getString** function to access strings in normal code.
- For example:




```
var s = getString(R.string.hello_world)
```



Note: getString is a method on the Context class.

- Resource id creation…

Resource Id Creation

R.txt

- **Generated automatically** based on external resources.
- Created when your project is compiled.
- Contains the definition of each resource.

R.txt

- The following XML code is from a strings.xml file.
- It defines two string resources:
  - app_name
  - hello_world
- When this project is compiled the R.txt file will contain variables for each of the string resources.

```
<?xml version="1.0" encoding="utf-8"?>
<resources>

    <string name="app_name">bcs421.arthurhoskey.lab.helloyankees blah blah</string>
    <string name="hello_world">Hello Yankees!</string>

</resources>
```

strings.xml

- **This resource ids are defined in R.text. R.txt is located in the subdirectory: \app\build\intermediates\runtime_symbol_list\debug**
- **DO NOT EDIT THIS FILE**

- Here is some of the contents of an R.txt file:

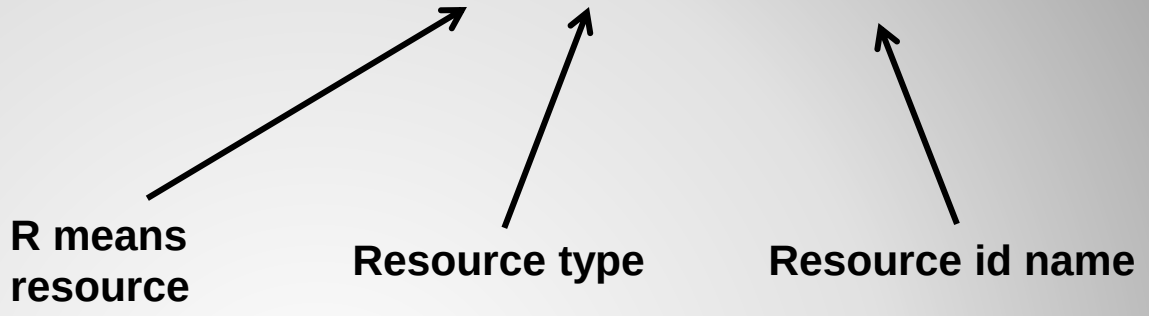
int anim abc_fade_in 0x7f010000
int anim abc_fade_out 0x7f010001
int anim abc_grow_fade_in_from_bottom 0x7f010002
…

int string app_name 0x7f0f001c
…
int string hello_world 0x7f0f007c
…

resource name (hello_world)
hello_world unique resource id number
Resource type (string)

R.txt

```
Text(text = stringResource(id = R.string.hello_world))
```



R.Txt File

```
int string hello_world 0x7f0f007c
```

Using Resource ID

- Now on to using different languages and hardware…

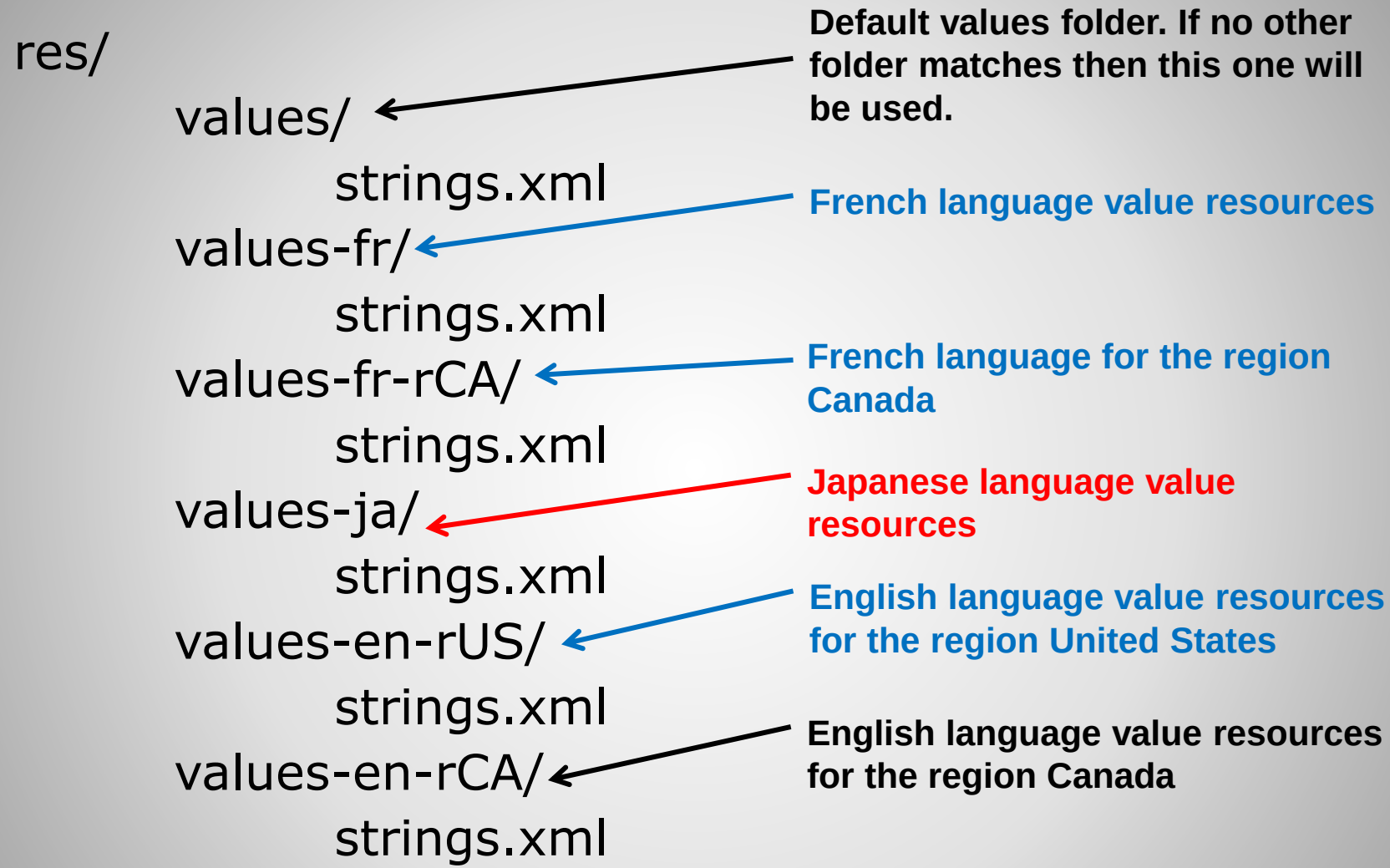
Different Languages and Hardware

Different Languages and Hardware

- You can also create different resources for the following:
  - Specific languages
  - Locations
  - Hardware configurations
- Use a parallel directory structure within the res folder.
- Each directory gets its own copy of the resource file.
- Good Localization Link:

http://developer.android.com/guide/topics/resources/localization.html

- For example…



Different Languages and Hardware

Runtime Configuration Changes

- Android handles runtime changes to the following:
  - Language
  - Location
  - Hardware
- Android terminates and restarts the active Activity to do this.
- You can bypass this behavior
- You can detect and react to these changes yourself.
- Textbook gives details of this process.

Other Resource Types

- You can also create external resources for the following:
  - Colors
  - Dimensions
  - Styles and Themes
  - Drawables
  - Animations
  - Property Animations
  - View Animations
  - Frame-by-Frame Animations
- Usage and setup for these are similar to what we have seen.

Review

- What is the benefit of externalizing resources?
- What folder are all resources stored in?
- What folder are the string resources stored in?
- How would you go about using different languages in an app?

- End of Slides

End of Slides