ReflectCode

Automated source code transformation

Re-cycle / Re-use / Re-purpose

Technical Guide for Developers

Contents

1. Purpose

Android and iOS are huge and complex systems with various techniques, standards and libraries. Both platforms are widely different and uses different technologies for developing the apps. ReflectCode's aim is to bridge the difference and provide the translated Xcode project which is developer friendly. While doing this we also provide our attention to make sure that Xcode project is generated as per standard tools and techniques used by iPhone developers.

Both the platforms have some good and some not so good ways of doing things e.g. iOS platform do not provide mechanism to refer to assets in type safe manner, we need to provide exact string of resource. On the other hand Android provides the type safe way using R class (R.id., R.string. etc.) During generating the Xcode project we make sure that the good part of the Android project is transferred to Xcode. In this case we generate 'R.swift' class which records all the resources. For easy access we also include the 'RCGetResources.swift' in the project.

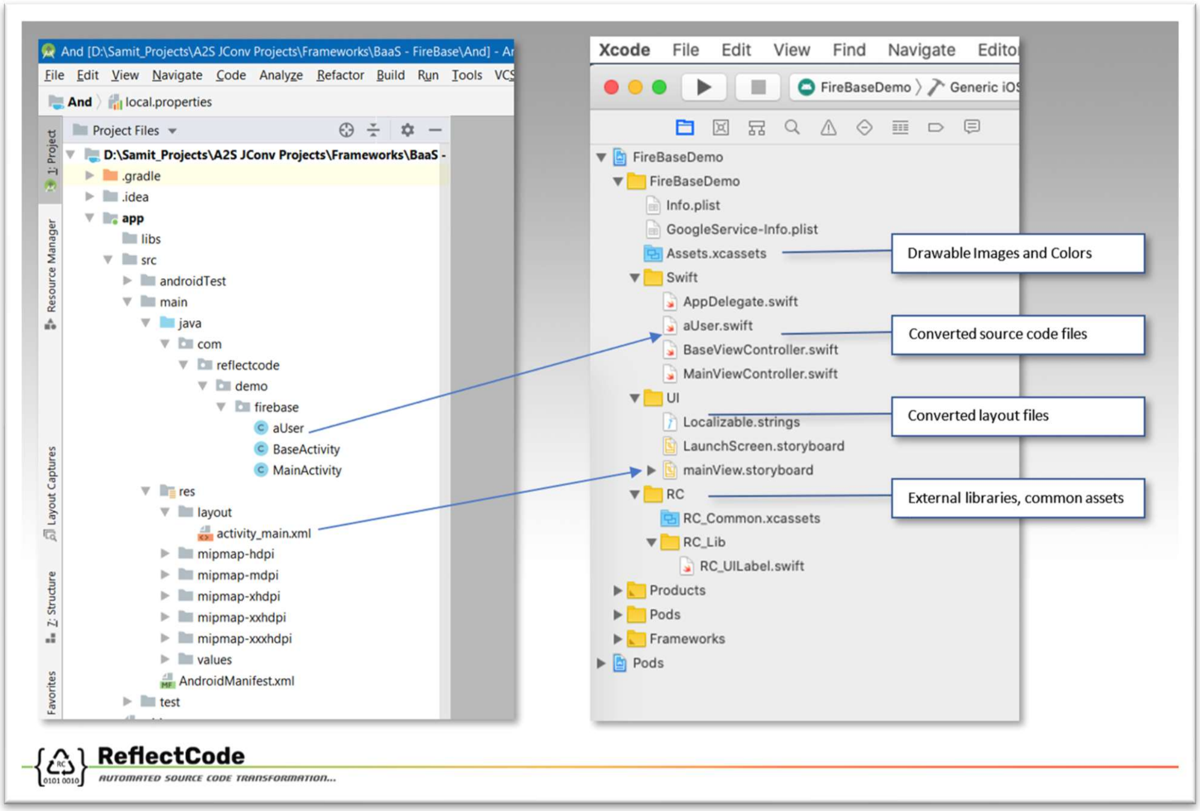
During translation our first preference is to use the APIs provided by platform. However in few complex areas, we make use of additional wrapper classes to bridge the platform differences. For this we have developed various custom controls and classes and include those in the project as per need basis. These are open source and provided with MIT license.

Aim of this 'DevGuide' is to make developer aware of the project structure and techniques used in the generated Xcode project. We hope that this will help developers to quickly get started with the new Xcode project for further fine tuning and enhancements.

2. Project structure

The new Xcode project structure is almost mirror image of Android project. All the android project folder structure will be preserved and migrated to new Xcode project. Also names of all files, classes and its members are preserved. To make sure that Xcode project follows the iOS standards minor changes are done as mentioned below.

Folder structure comparison :



Drawable Images and Colors

Converted source code files

Converted layout files

External libraries, common assets

File name changes :

As per iOS code standards, names of classes are updated as below :

File type	Android file name	Xcode file name
Source code	LoginActivity.java	LoginViewController.swift
Layout	login_screen_activity.xml	login_screen_view.xml

3. Additional notes in code

For better understanding, we insert the informative notes in the comments for developers. These notes starts with “// RC Note :” Few examples are -

- *// RC Note : source android method is onCreate()*
- *// RC Note : Code moved to >>> @objc func onClick(_ view: UIView?){*
  Android code makes heavy use of Lambada expressions (Closures in iOS) such as OnClickListener. On iOS platform most classes do not use closure, instead they use target action methods. In this situation the code inside android lambada expression is converted to a method and appropriate #selector is assigned to the control. When code is changed into method, its location in the file changes and this comment will help to locate the original code.

- *// RC Note (Java code not required)*
  Few Java APIs are not required in the iOS code because it is implemented in some other way.

- *// RC Note : Java code not required >>> resultDialog.setCancelable(false);*
- *// RC Note : Android specific path is used, this needs refactoring for iOS platform*
- *// RC Note : Inline anonymous classes not supported in Swift, it is converted to Nester class*

4. TODOs added in code

In few cases additional steps are required for code to compile and behave as per expectation on iPhone. For such instances, we add TODO comments and assign unique number to it. This should help in tracking those items. E.g. –

- *//TODO: RC-J039-012 : Android Build Version check is not applicable on iOS platform, The code below needs refactoring*

  Android code may contain version specific code. During after code is translated, we need to review these comment and delete the code which is not required

Few more examples :

- *//TODO: RC-J002-003 : Class 'Response' in below statement is not supported*
- *//TODO: RC-J036-003 : UITapGestureRecognizer' used, change method signature to - @objc*
- *//TODO: RC-J036-001 : Custom font "Roboto-Regular" used. This is a file name, replace it with actual font name*

We follow the naming convention -

RC : is identify that comment is added by our system from original project comments
J002 : Unique id assigned to each Java file. For Kotlin files, id starts with K
003 : at the end is the unique serial number within the file

Note : The list of ToDo items are provided in the project report. After conversion, ReflectCode's post processing algorithm scans all the ToDo items in the list and tries to resolve few of them. These items will have (**Resolved by RC**) tag assigned for identification.

5. Code which is removed from the project

In some cases the iOS SDK provides a totally different approach. E.g. Button click event. In Android we use "`.setOnClickListener(this);`". In iOS, the best approach to handle events is @IBAction. ReflectCode analyse such calls to generate appropriate @IBAction and no code is generated for this Android API.

Exception : Android allows to assign the click listener to its ImageView control, iOS UIImageView do not allow the click event handling. In this case we assign the Gesture Recognizer

Platform	Code sample
Android	`btnLogin.setOnClickListener(this);`
Xcode	`@IBAction func onClick(_ v: UIButton?){`
Xcode	`imageView?.addGestureRecognizer( UITapGestureRecognizer(target: self, action: #selector( onClick(_:) )))`

Similarly to access UI controls on Android we use "`.findViewById(R.id.btnLogin);`" and this converted to @IBOutlet.

Platform	Code sample
Android	`btnLogin = (Button) findViewById(R.id.btnLogin);`
Xcode	`@IBOutlet weak var btnLoginOutlet: UIButton!`

6. New classes added to the project

During conversion we may introduce new classes as need basis. All the new classes will be added to separate folder "\RC\RC_Lib".

Few of these classes are developed in house by ReflectCode. The name prefixed "RC" e.g. :

- RCToast.swift (Android like toast message)
- RCGetResources.swift (to access the resources)
- RCAsyncTask.swift (to provide Asynchronous programming)

7. New variables inserted in code

In some cases we need to introduce the new temporary variables to match the platform difference. These are used to save intermediate objects. All such variable names start with ‘rc’ prefix. E.g. :

- var rcURLCollection: [URL]?

8. License and copy right

All the classes and libraries developed by ReflectCode are open source and made available as MIT License.

We may also use some 3rd party open source classes / libraries. We have ensured that all the external source code included in the project are open source and provided with open source license. Their original copy right information is maintained in the files.

The list of all the libraries with their license information and URL can be found in project report. Please refer to ‘External framework included’ table.

9. Sample Projects

ReflectCode’s open source contribution and sample projects are available on GitHub :

Calculator App	- https://github.com/ReflectCode/Calculator-App
Layout Demo	- https://github.com/ReflectCode/Layout_Demo
UI Controls	- https://github.com/ReflectCode/UI_Demo
Glide	- https://github.com/ReflectCode/Library-Glide
Firebase	- https://github.com/ReflectCode/Library-Firebase
Gallery App	- https://github.com/ReflectCode/Layout_RecyclerView
Drawer	- https://github.com/ReflectCode/Navigation-Drawer

10. About ReflectCode

You can find more information about RC on below links -

Website	- https://www.reflectcode.com/
LinkedIn	- https://www.linkedin.com/in/reflectcode
LinkedIn	- https://in.linkedin.com/in/samit27p
Twitter	- https://twitter.com/ReflectCode
Skype	- live:reflectcode
YouTube Channel	- https://www.youtube.com/channel/UCIWqyJ0Roy8T_Q4BFrVDriw
Angle.co	- https://angel.co/company/reflectcode