iOS Development with `BUCK`

Issarapong

Growth Session #11 - January 18-19 2018

Intro

- Generally, native iOS project will be created with Xcode

  ⇒ Use CocoaPods or Carthage as Dependency management.

- Project & Dependencies are in `project_name.workspace`

  ⇒ `project_name.xcodeproject` & `Pods.xcodeproject`

- `project.pbxproj` is properties file of `.xcodeproject` which declared project settings, compiler settings, resources, targets, provisioning profile, certs, etc …

Xcode Project Structure

```
21F360DD1FE908F7001B9B2E /* PricePreviewViewController.swift in Sources */ = {isa = PBXBuildFile; fileRef = 21F360DC1FE908F7001B9B2E /* PricePreviewViewController.swift */; };
21F360DE1FE908F7001B9B2E /* PricePreviewViewController.swift in Sources */ = {isa = PBXBuildFile; fileRef = 21F360DC1FE908F7001B9B2E /* PricePreviewViewController.swift */; };
21F360DF1FE908F7001B9B2E /* PricePreviewViewController.swift in Sources */ = {isa = PBXBuildFile; fileRef = 21F360DC1FE908F7001B9B2E /* PricePreviewViewController.swift */; };
```

`PricePreviewViewController.swift` in 3 Targets

- AppStore
- Internal Staging
- Internal Production

Xcode Project Structure

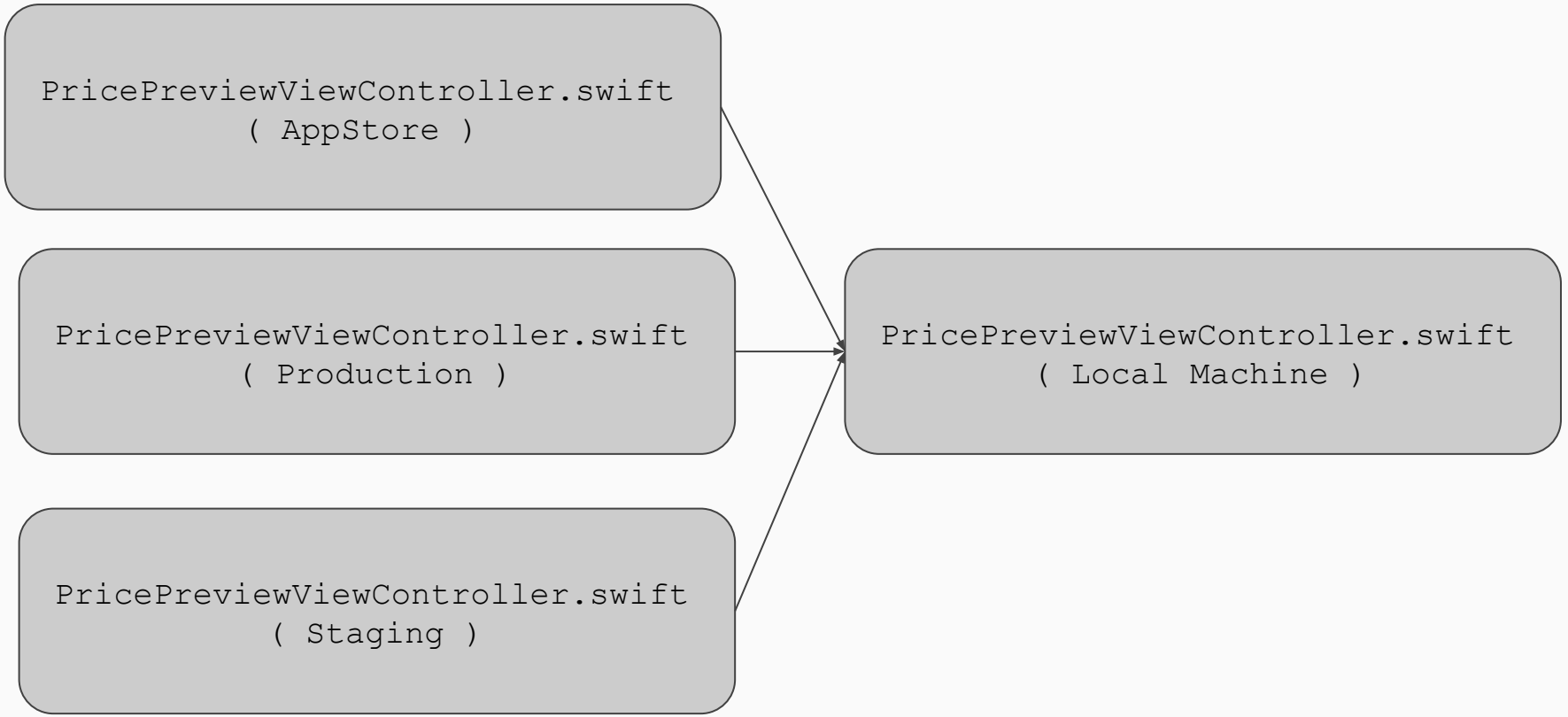
```
21F360DD1FE908F7001B9B2E /* PricePreviewViewController.swift in Sources */ = {isa = PBXBuildFile; fileRef = 21F360DC1FE908F7001B9B2E /* PricePreviewVie
21F360DE1FE908F7001B9B2E /* PricePreviewViewController.swift in Sources */ = {isa = PBXBuildFile; fileRef = 21F360DC1FE908F7001B9B2E /* PricePreviewViewController.swift */; };
21F360DF1FE908F7001B9B2E /* PricePreviewViewController.swift in Sources */ = {isa = PBXBuildFile; fileRef = 21F360DC1FE908F7001B9B2E /* PricePreviewViewController.swift */; };
```

The hex fileRef right is the reference of the file in the local machine.

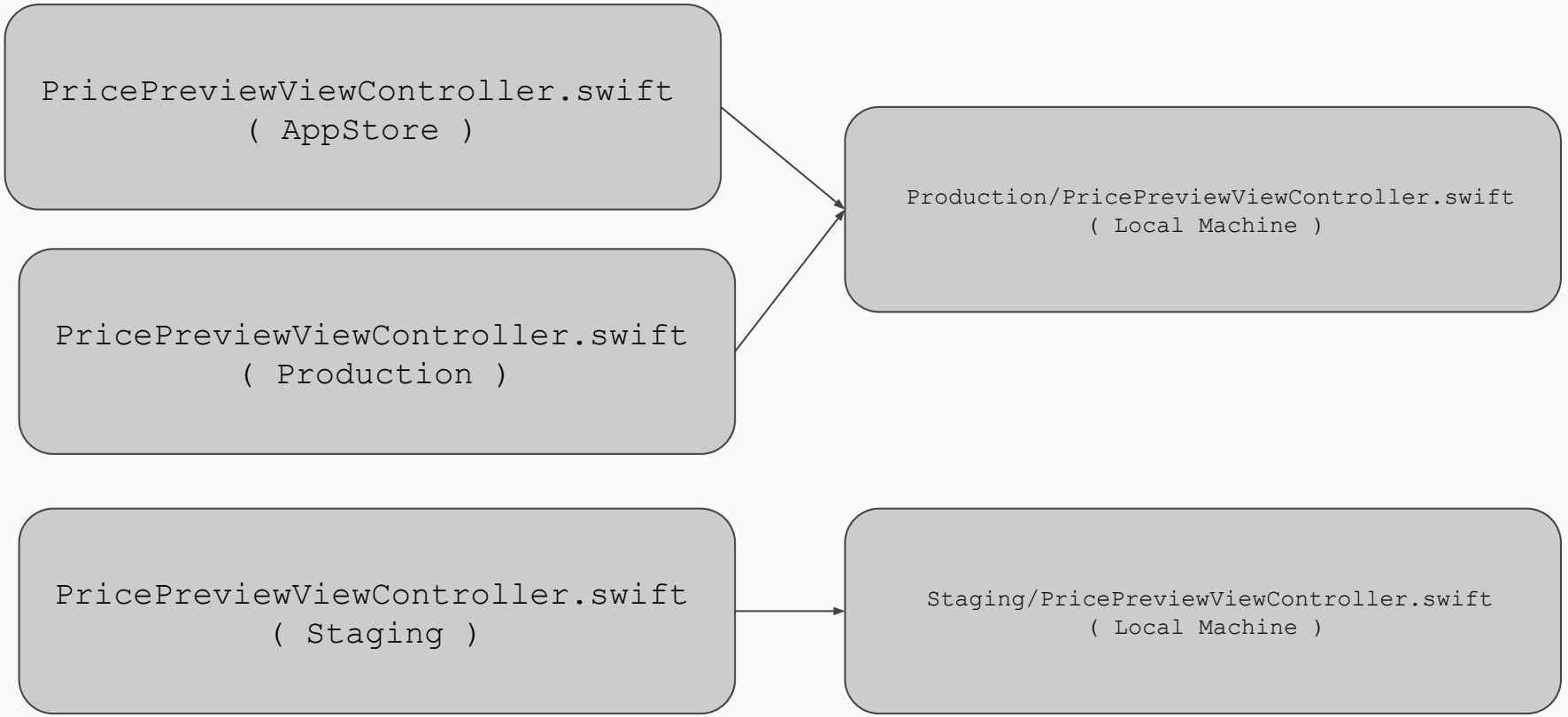
```
21F360DD1FE908F7001B9B2E /* PricePreviewViewController.swift in Sources */ = {isa = PBXBuildFile; fileRef = 21F360DC1FE908F7001B9B2E /* PricePreviewVie
21F360DE1FE908F7001B9B2E /* PricePreviewViewController.swift in Sources */ = {isa = PBXBuildFile; fileRef = 21F360DC1FE908F7001B9B2E /* PricePreviewViewController.swift */; };
21F360DF1FE908F7001B9B2E /* PricePreviewViewController.swift in Sources */ = {isa = PBXBuildFile; fileRef = 21F360DC1FE908F7001B9B2E /* PricePreviewViewController.swift */; };
```

The left fileRef is the actual reference of the file in the project (unique)

Xcode Project Structure

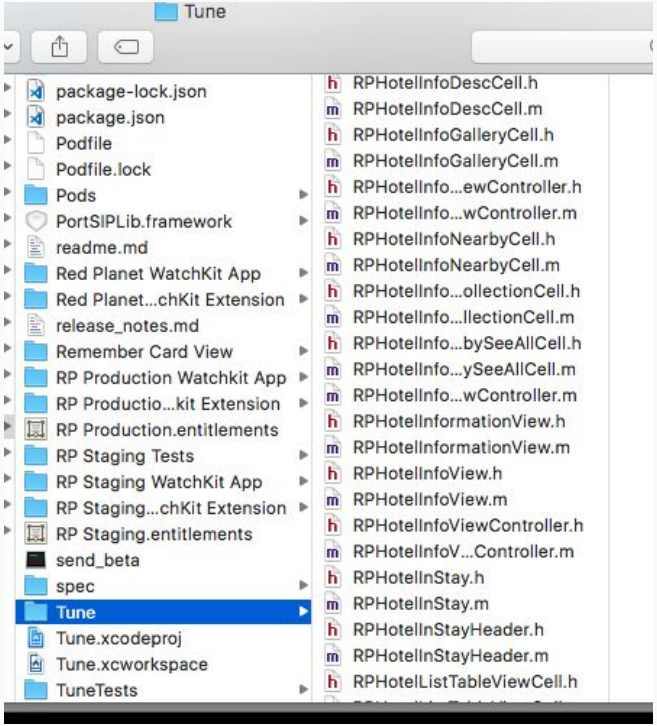


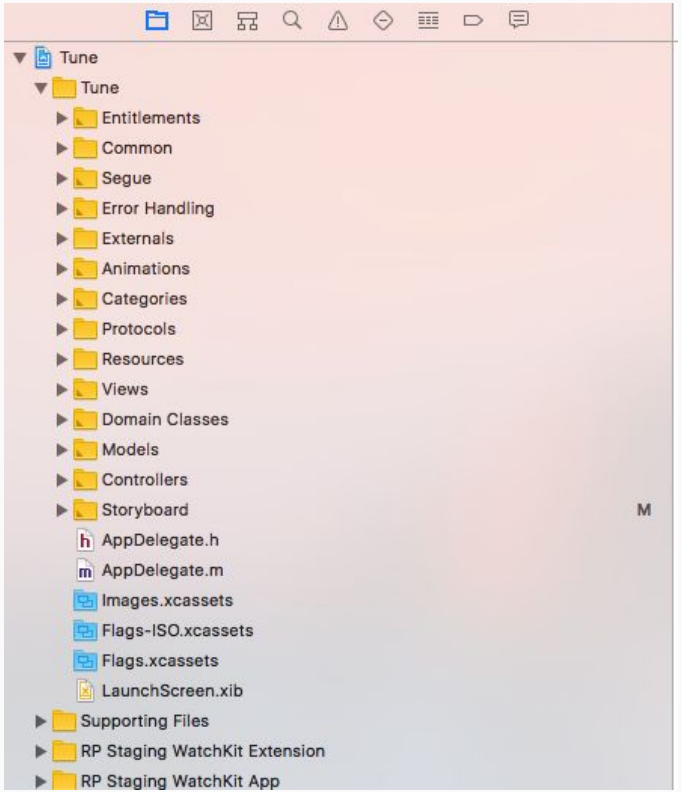
Xcode Project Structure



Xcode Project

XcodeProject use file references to create its own sources directory





What’s the problem ?

Xcode Project

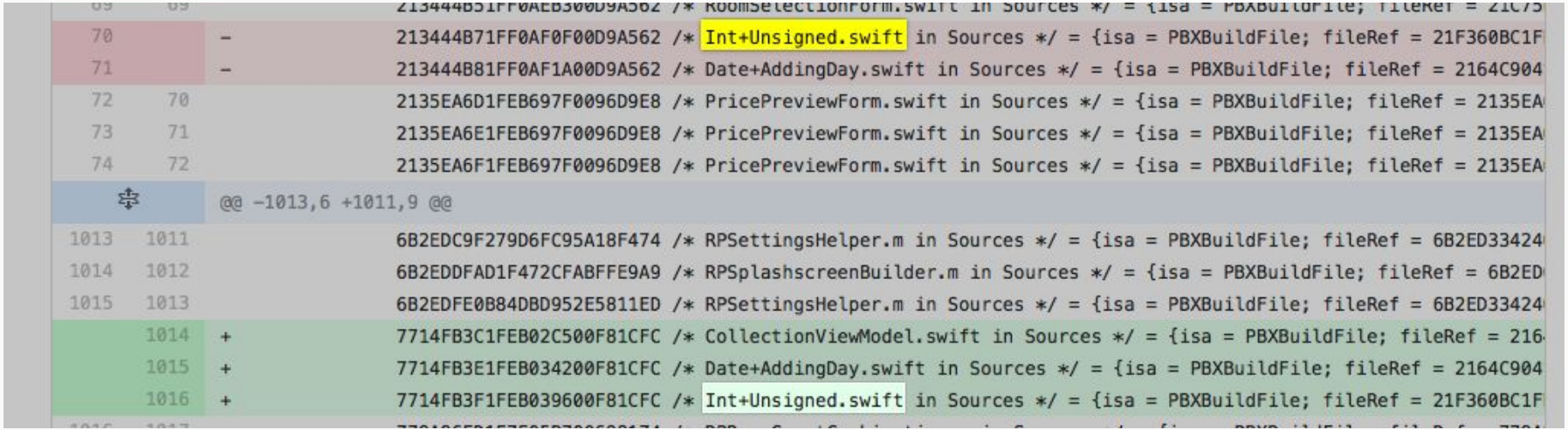
Someone switches branches then move files within Xcode?

⇒ Project files will have conflicts 😱 😱 😱

Xcode Project

Adding a file with the same name & directory in 2 different branches ?

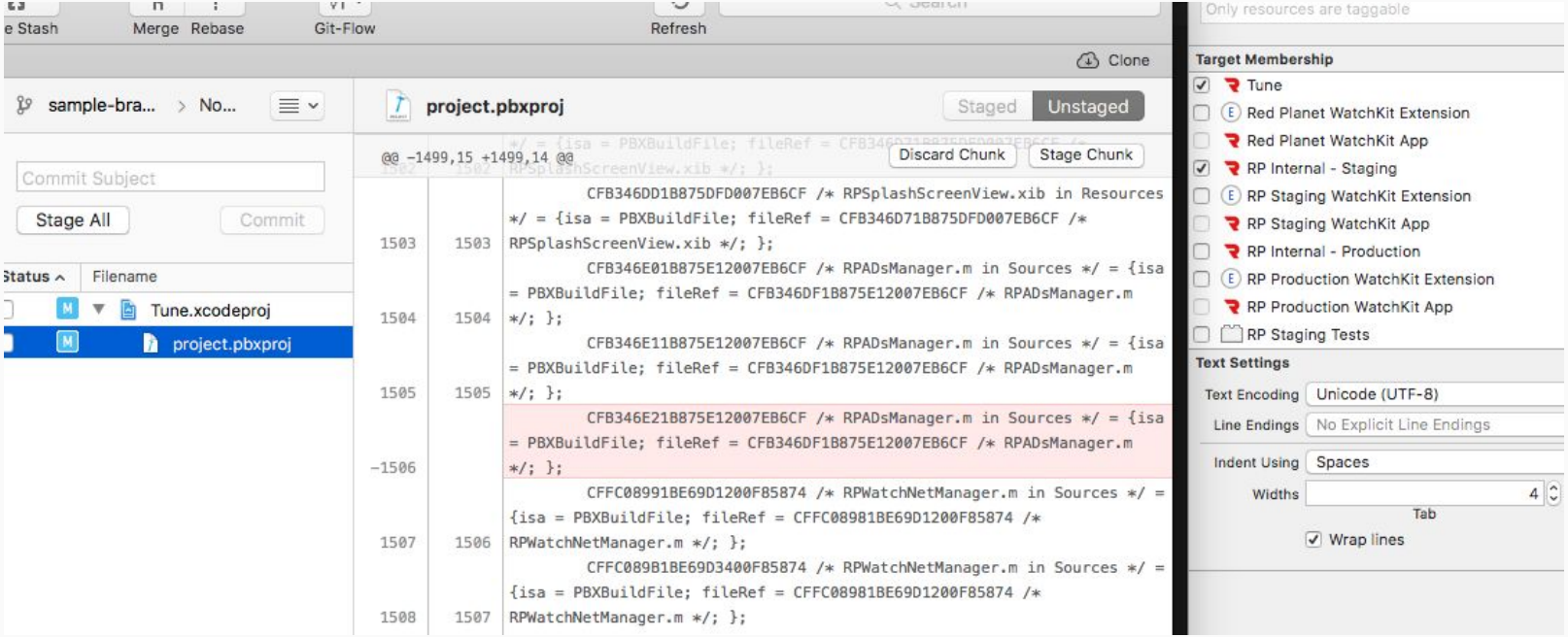
⇒ Project files will have conflicts 🤯🤯🤯



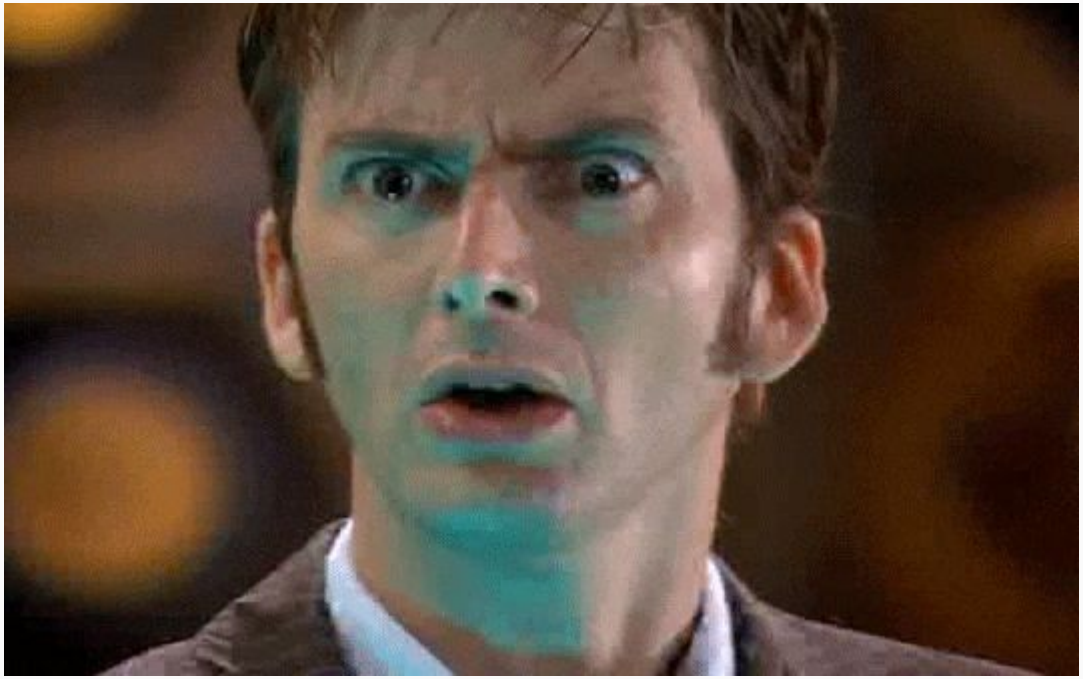
Xcode Project

Someone in the team re-add files in the target ?

⇒ Project files will have conflicts 🤯🤯🤯



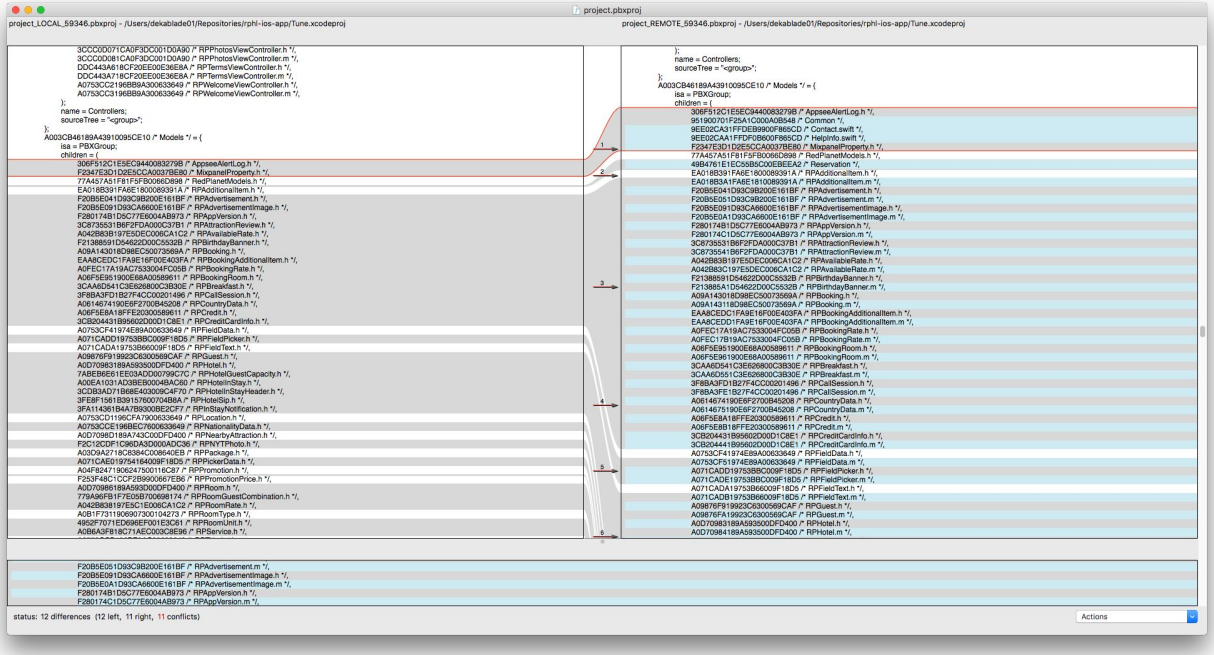
Xcode Project



Xcode Project

Someone in the team re-add files in the target ?

⇒ Project files will have conflicts 🦃 🦃 🦃



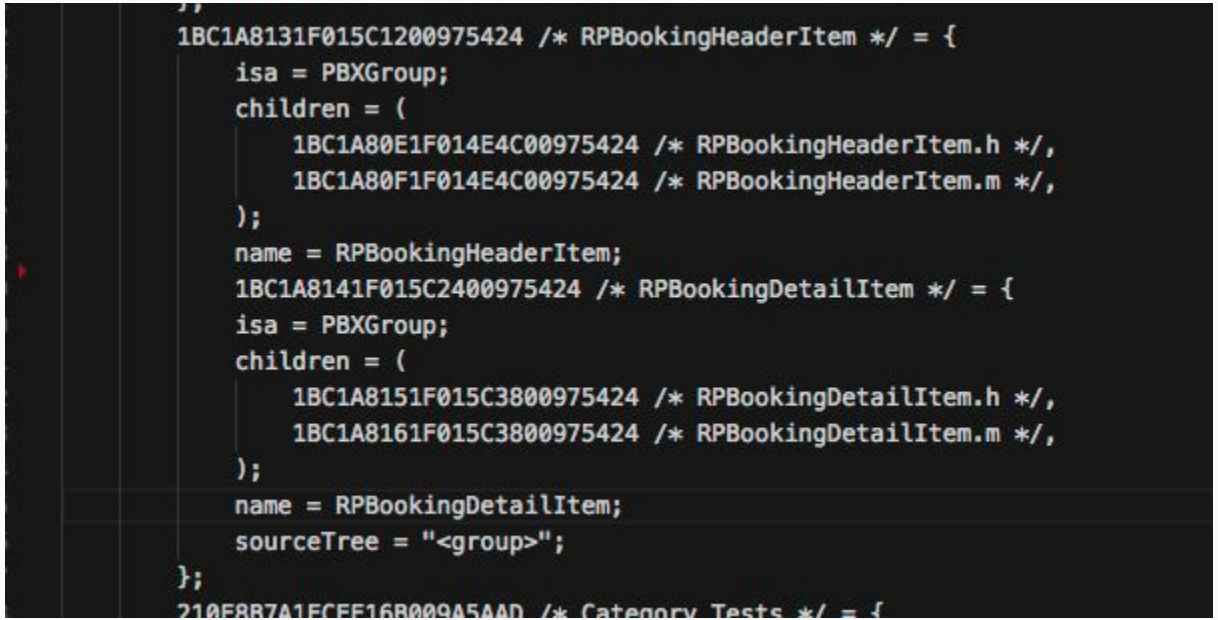
Xcode Project

Solving this kind of conflicts requires to dive into the project file and find the valid file reference

Xcode Project

Git is not that cleaner

⇒ Sometime git breaks the braces, then the project can’t be open 😱 😱 😱



```
1BC1A8131F015C1200975424 /* RPBookingHeaderItem */ = {
    isa = PBXGroup;
    children = (
        1BC1A80E1F014E4C00975424 /* RPBookingHeaderItem.h */,
        1BC1A80F1F014E4C00975424 /* RPBookingHeaderItem.m */,
    );
    name = RPBookingHeaderItem;
    1BC1A8141F015C2400975424 /* RPBookingDetailItem */ = {
    isa = PBXGroup;
    children = (
        1BC1A8151F015C3800975424 /* RPBookingDetailItem.h */,
        1BC1A8161F015C3800975424 /* RPBookingDetailItem.m */,
    );
    name = RPBookingDetailItem;
    sourceTree = "<group>";
};
210F8B7A1FCFE16B009A5AAD /* Category Tests */ = {
```

Introducing ‘BUCK’

BUCK

- Buck is a build system developed and used by Facebook.
- Speed up building
    - Parallel build with multiple core of machine.
    - Tracking files, build only file with change(s).
    - Buck Query know what to build for the target.
- Integratable with IDE (Xcode, IntelliJ)

BUCK

Build With BUCK

- Buck use its own rules in `./BUCK` to for the project structure and `./buckconfig` to configure project like cert, compiler versions, etc, …
- Faster than `project.pbxproj`
- BUCK build with rules, not file references.
- BUCK build with `` `$ buck build target_name` ``
- `.pbxproj` `.workspace` is not required for build.

BUCK

./BUCK

- Contain build rules, targets
- Rules (iOS)
  - apple_asset_catalog()
  - apple_binary()
  - apple_bundle()
  - apple_library()
  - apple_package()
  - apple_resource()
  - apple_test()
  - core_data_model()
  - prebuild_apple_framework()

BUCK

./BUCK

- apple_resource()
  - `name` // rule_name
  - `files` // files with name & extentions
  - `dirs` // application bundle directories

```
apple_resource(
  name = 'BxTraderResources',
  files = glob(['**/**/*.png', '**/**/*.storyboard', '**/**/*.xib']),
  dirs = [],
)
```

BUCK

./BUCK

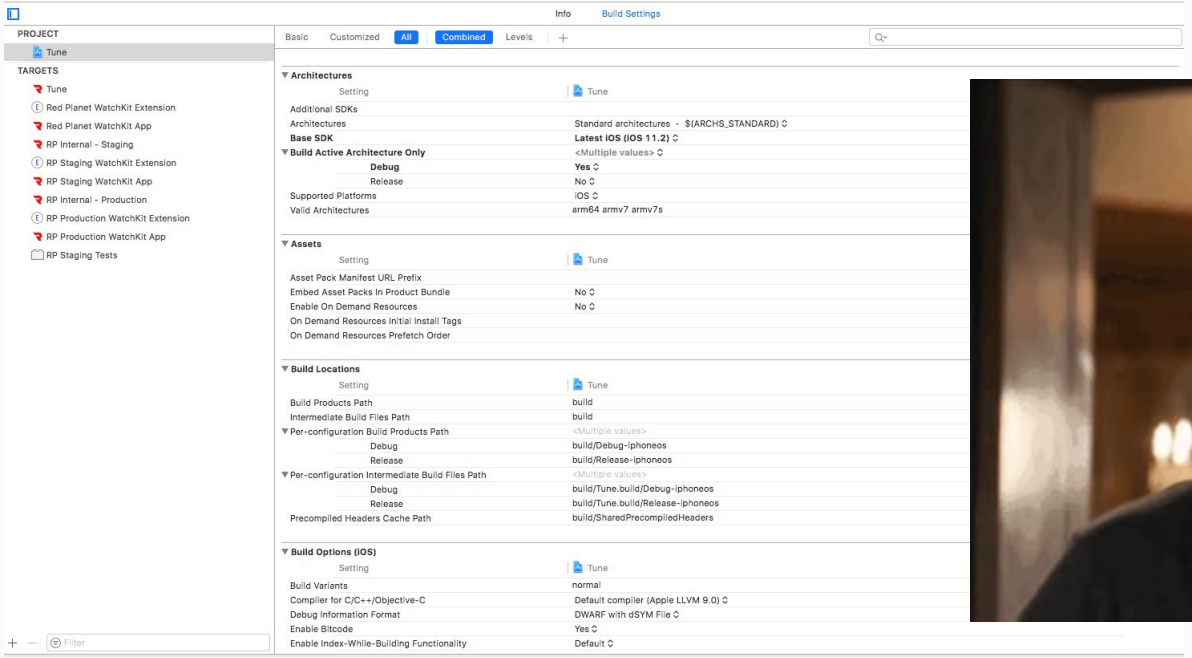
- apple_binary()
  - `name` // rule name
  - `srcs` // file sources
  - `frameworks` // frameworks

```
apple_binary(
  name = 'BxTraderAppBinary',
  srcs = glob([
    '*.m', '*.swift', '**/*.swift', '**/**/*.swift',
  ]),
  frameworks = [
    '$SDKROOT/System/Library/Frameworks/UIKit.framework',
    '$SDKROOT/System/Library/Frameworks/Foundation.framework',
    '$SDKROOT/System/Library/Frameworks/SystemConfiguration.framework',
  ],
)
```

BUCK > ./bugconfig

Project configurations

⇒ e.g.: Swift versions, pod lib swift versions, base sdk, bridge header, etc



BUCK > ./bugconfig

target_sdk_version

For each platform, you can specify the target SDK version to use. The format is `{platform}_target_sdk_version`.

```
[apple]
  iphonesimulator_target_sdk_version = 7.0
  iphoneos_target_sdk_version = 7.0
  macosx_target_sdk_version = 10.9
```

codesign

To override a default path to `codesign`, set this setting to either a file path or buck target.

```
[apple]
  codesign = //path/to/target/that/creates:codesign
```

BUCK > ./bugconfig

xctool_default_destination_specifier

This setting is passed directly to `xctool`, and then to `xcodebuild` as the `-destination` argument. See the man page for the proper syntax.

```
[apple]
  xctool_default_destination_specifier = platform=iOS Simulator
```

BUCK > ./bugconfig

.buckconfig

```
[cache]
  mode = dir

[cxx]
  default_platform = iphonesimulator-x86_64
  cflags = -g -std=c11
  cxxflags = -g -std=c++14
  # Many ARC APIs don't compile with separate preprocess and compile.
  combined_preprocess_and_compile = true


[alias]
  bx_trader_app = //:BxTraderApp

[apple]
  xctool_default_destination_specifier = platform=iOS Simulator


[project]
    ignore = .buckd, \
             .hg, \
             .git, \
             .idea, \
             buck-cache, \
             buck-out, \

[apple]
  xctool_zip_target = //third-party/ios/xctool:xctool-minimal-zip

[swift]
  version = 4.0
  compiler_flags = -DBUCK -whole-module-optimization -enable-testing -suppress-warnings

[project]
ide_force_kill = never
```

Pros & Cons

Pros

- Build faster up to 50% on CI [AirBNB]
- Smaller app size 30% [AirBNB]
- Buck still can generate `.workspace`, `.pbxproj` if we want to open project to build and run with xcode ⇒ `$ buck project target_name`
- `.pbxproj` , `.workspace` can be added to `.gitignore` to avoid conflicts

Cons

- Very complex to configure
- Requires extreme knowledge in iOS Project Structure. Because you need to Initialize a project with ./BUCK & ./buckconfig (so NOT with Xcode > File > New Project)
- Able to use CocoaPods but need to add every Pods headers manually in ./BUCK

DEMO

Thanks!

Contact Nimbl3

hello@nimbl3.com

399 Sukhumvit Road, Interchange 21
Klongtoey nua, Wattana
Bangkok 10110

20th Floor, Central Tower
28 Queen's Road
Central, Hong Kong

nimbl3.com