Don’t Fear the Project

Jared Sorge
https://jsorge.net
jared@jsorge.net

My Work

Let’s Survey the Project Landscape

Tim Ekl
@timothyekl

I'd love to see a #nextcoders talk (lightning or otherwise) that just has a Venn diagram of Xcode project, scheme, target, workspace, etc.

11:25 AM · Mar 5, 2018 · Tweetbot for Mac

4 Likes

Target

Target

- Inputs
  - Source files
  - Build settings
  - Build phases
  - Capabilities
- Outputs
  - Application, dynamic framework, static library, extension…

Scheme

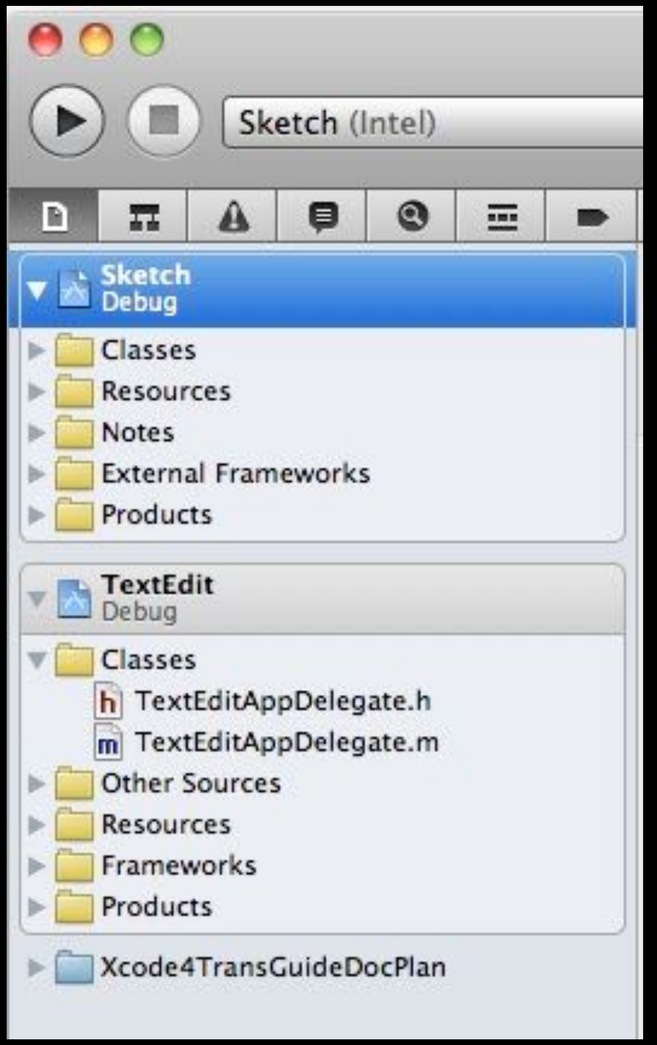
- Associated with a target
- Responsible for doing the building, running, testing of its target
- Applies configurations
  - debug & release are standards
- Determines which tests run
- Applies diagnostics such as sanitizers

Project

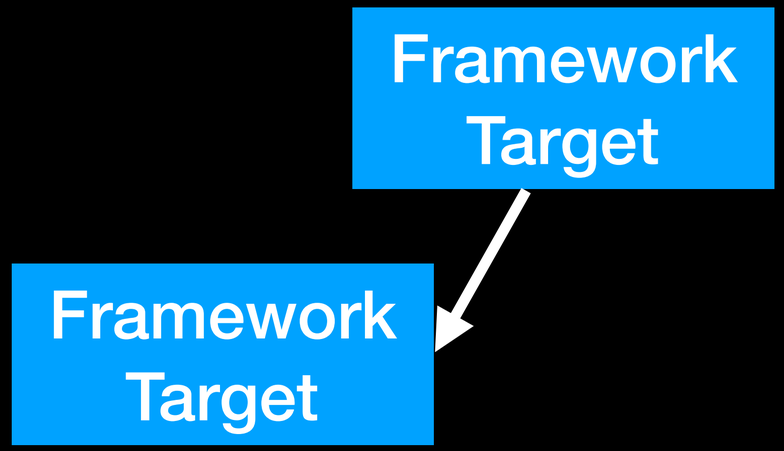
- Target management
  - Source files, build settings, extra resources
- Scheme management
- Can invoke the build process
- Run tests
- Syntax highlighting
- Indexing
- Much more

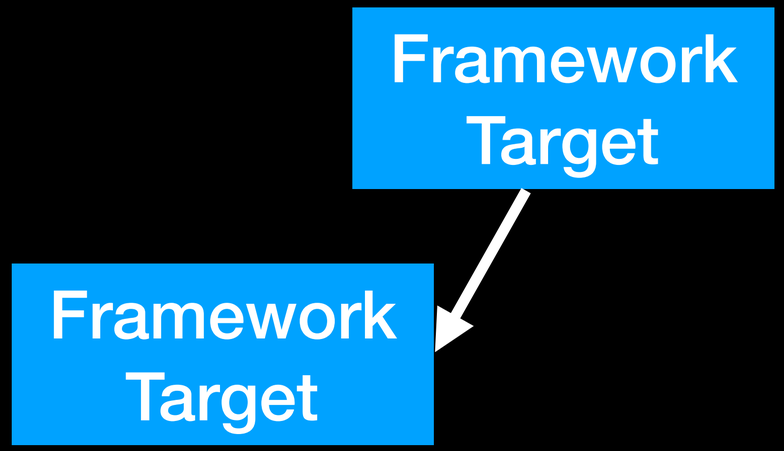
Workspace

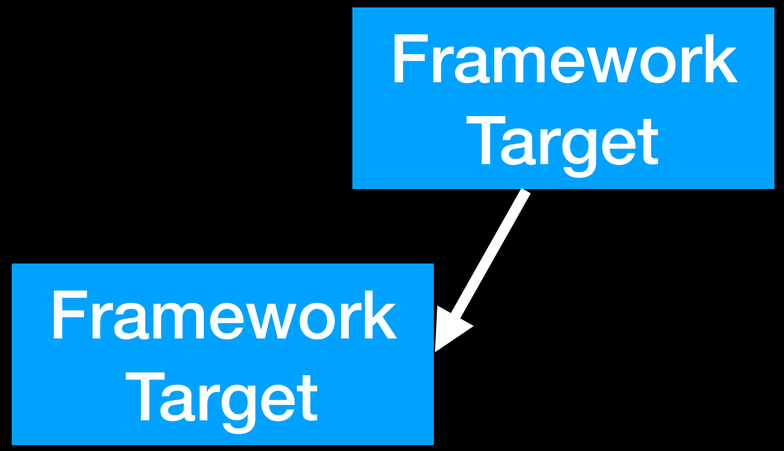
- Work with multiple projects at a time
- Most common in things like CocoaPods

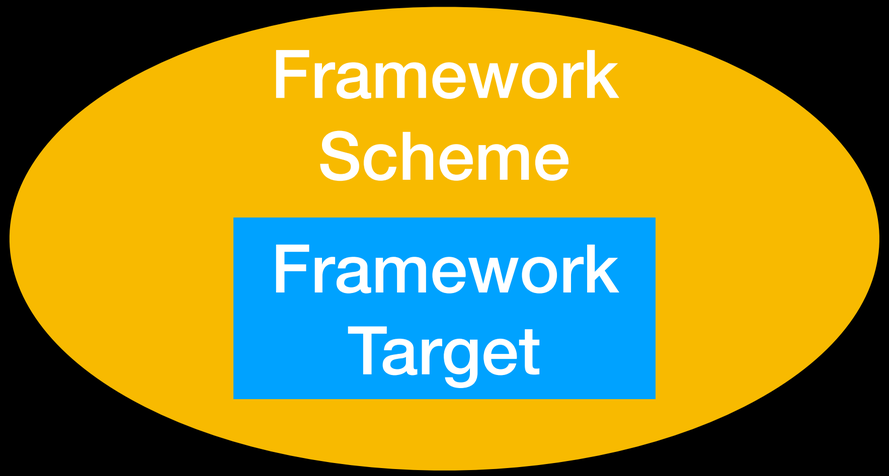


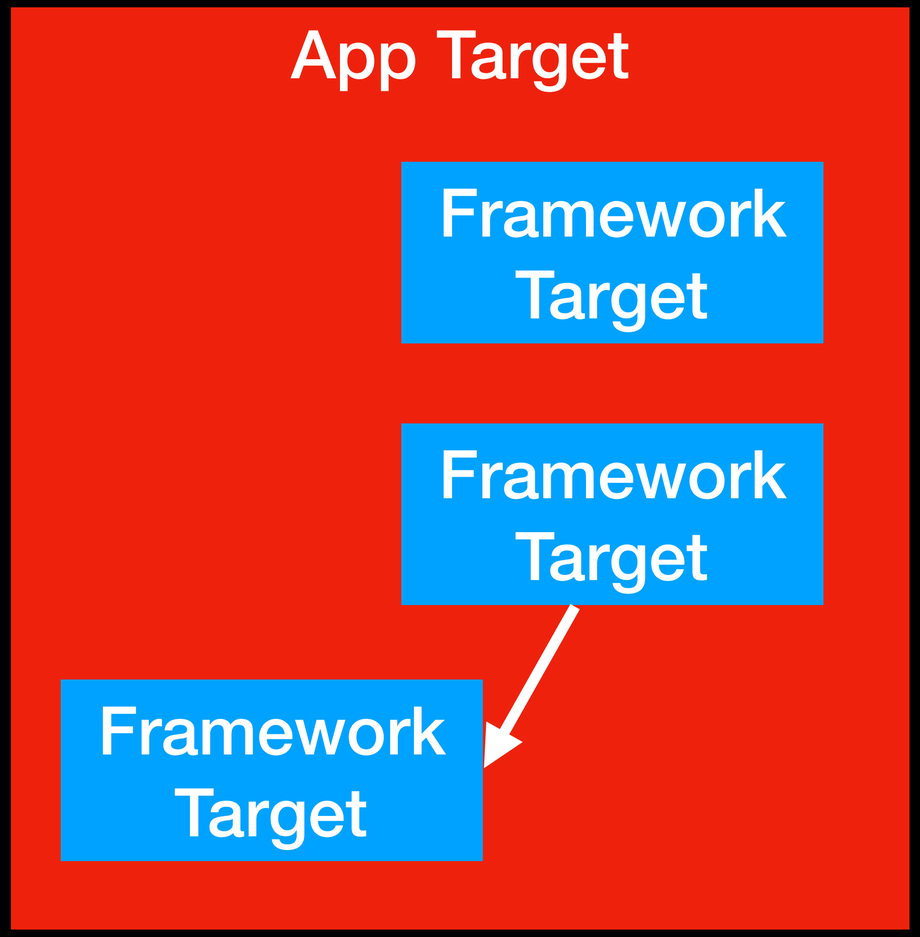
Framework
Target


Framework
Target
Framework
Target

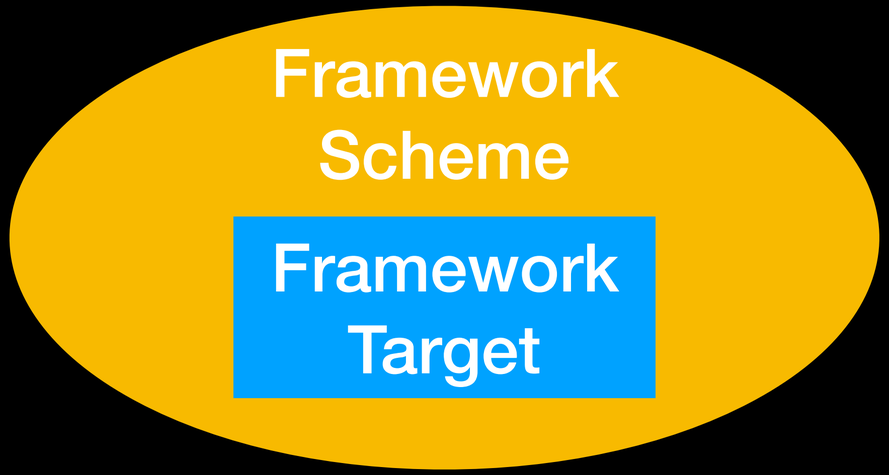

Framework
Target
Framework
Target

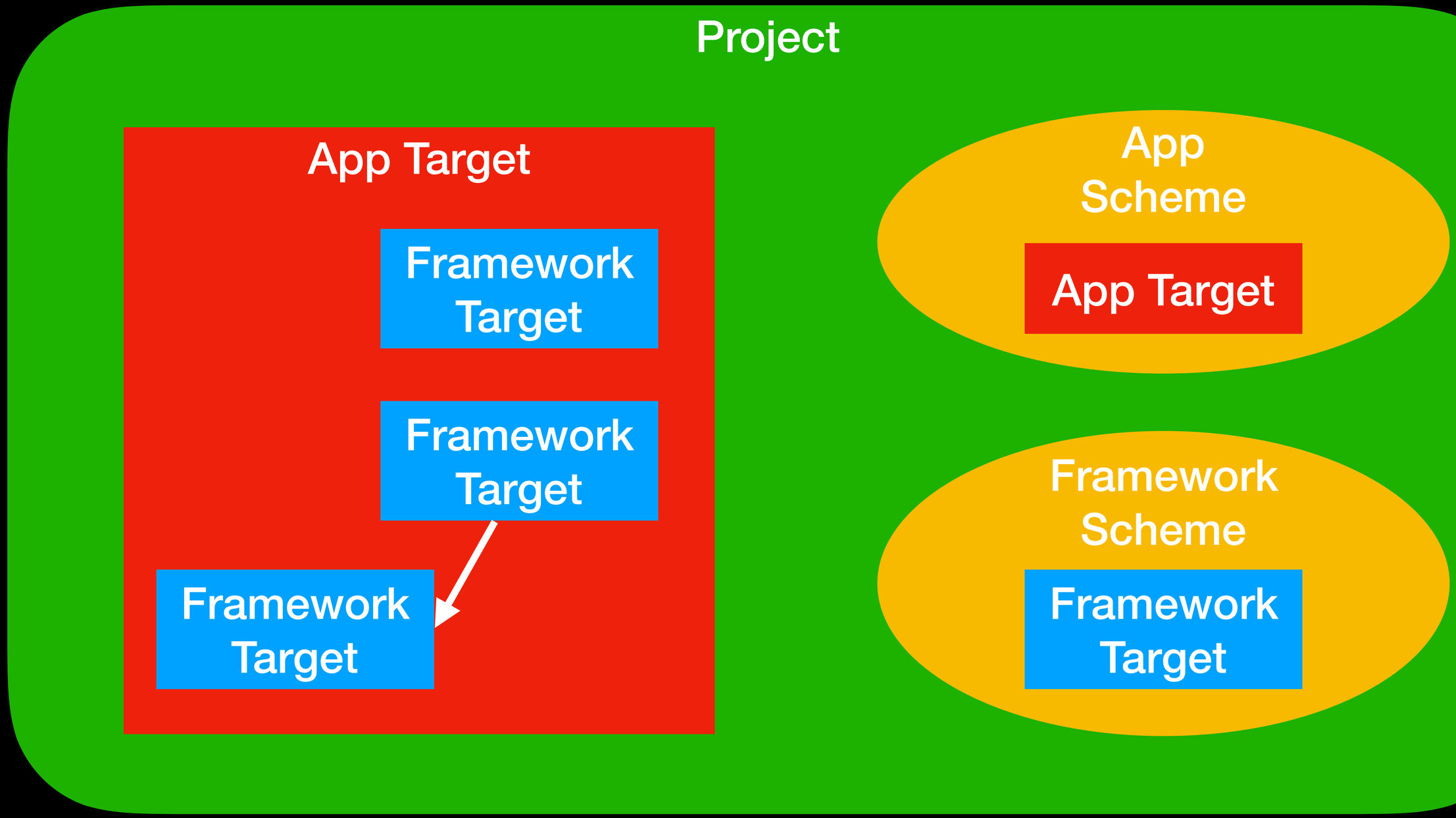

Framework
Target
Framework
Target

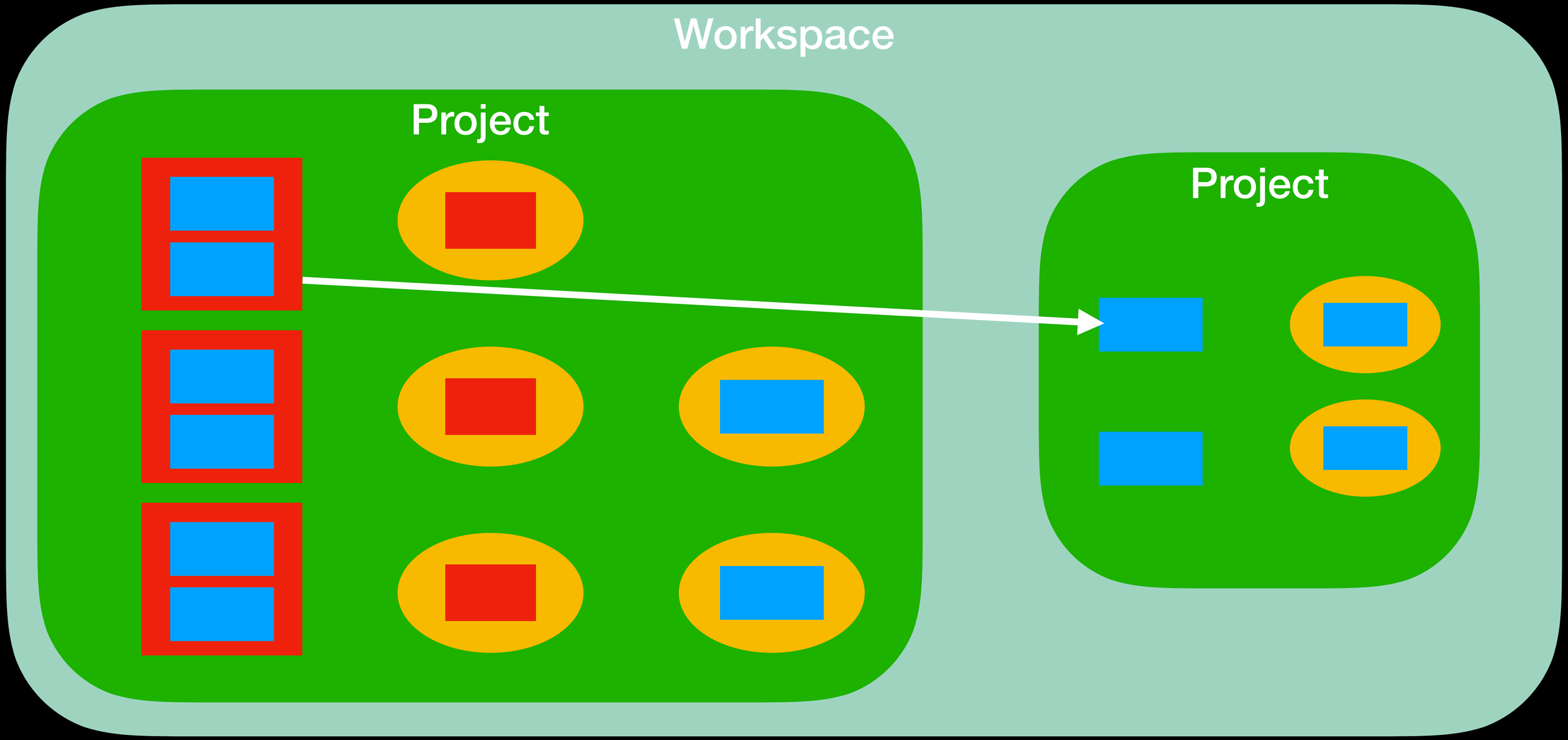

Framework
Scheme
Framework
Target


App Target
Framework Target
Framework Target
Framework Target

App Scheme
App Target


Framework Scheme
Framework Target


Project
App Target
Framework Target
Framework Target
Framework Target
App Scheme
App Target
Framework Scheme
Framework Target


Workspace
Project
Project

Resources

- WWDC 2018: Behind the Scenes of the Xcode Build Process
  - https://developer.apple.com/videos/play/wwdc2018/415/

December, 2017

“We don’t check in Xcode projects”

–iOS co-worker at Lyft

Generating Xcode Projects

- XcodeGen
  - https://github.com/yonaskolb/XcodeGen
  - Define your project in yml or json files
- Swift Package Manager
  - `swift package generate-xcodeproj`
  - Define your project in the Package.swift manifest

Why do this?

- Groups and files in Xcode are always in sync with the filesystem
  - Great Developer Habits, WWDC 2019
    - https://developer.apple.com/videos/play/wwdc2019/239/
- Human readable project configurations stored in source control
- One less file to worry about in source control and code reviews
- No more merge conflicts in project files

“Only 123 lines of conflict in my project file.
Rather, blocks of conflicts.
Probably a couple thousand lines.
On the other hand, I know what I’m doing today.”

–Beleaguered Developer

```
// !$*UTF8*$!
{
  archiveVersion = 1;
  classes = {
  };
  objectVersion = 50;
  objects = {

/* Begin PBXBuildFile section */
    935B0BCC22F27614007FC7C1 /* AppDelegate.swift in Sources
*/ = {isa = PBXBuildFile; fileRef = 935B0BCB22F27614007FC7C1
/* AppDelegate.swift */; };
    935B0BCE22F27614007FC7C1 /* ViewController.swift in
Sources */ = {isa = PBXBuildFile; fileRef =
935B0BCD22F27614007FC7C1 /* ViewController.swift */; };
    935B0BD122F27614007FC7C1 /* Main.storyboard in Resources
*/ = {isa = PBXBuildFile; fileRef = 935B0BCF22F27614007FC7C1
/* Main.storyboard */; };
```

```
Embed Frameworks */,
        );
        name = "Embed Frameworks";
        runOnlyForDeploymentPostprocessing = 0;
    };
/* End PBXCopyFilesBuildPhase section */

/* Begin PBXFileReference section */
    935B0BC822F27614007FC7C1 /* MyContactApp.app */ = {isa =
PBXFileReference; explicitFileType = wrapper.application;
includeInIndex = 0; path = MyContactApp.app; sourceTree =
BUILT_PRODUCTS_DIR; };
    935B0BCB22F27614007FC7C1 /* AppDelegate.swift */ = {isa
= PBXFileReference; lastKnownFileType = sourcecode.swift;
path = AppDelegate.swift; sourceTree = "<group>"; };
    935B0BCD22F27614007FC7C1 /* ViewController.swift */ =
{isa = PBXFileReference; lastKnownFileType =
sourcecode.swift; path = ViewController.swift; sourceTree =
"<group>"; };
```

```
935B0BE922F27824007FC7C1 /* DataModel.framework in
Frameworks */,
      );
      runOnlyForDeploymentPostprocessing = 0;
    };
    935B0BDF22F27624007FC7C1 /* Frameworks */ = {
      isa = PBXFrameworksBuildPhase;
      buildActionMask = 2147483647;
      files = (
      );
      runOnlyForDeploymentPostprocessing = 0;
    };
/* End PBXFrameworksBuildPhase section */

/* Begin PBXGroup section */
    935B0BBF22F27614007FC7C1 = {
      isa = PBXGroup;
      children = (
        935B0BCA22F27614007FC7C1 /* MyContactApp */,
```

```
        );
        path = DataModel;
        sourceTree = "<group>";
    };
/* End PBXGroup section */

/* Begin PBXHeadersBuildPhase section */
    935B0BDD22F27624007FC7C1 /* Headers */ = {
        isa = PBXHeadersBuildPhase;
        buildActionMask = 2147483647;
        files = (
            935B0BE622F27624007FC7C1 /* DataModel.h in Headers
*/,
        );
        runOnlyForDeploymentPostprocessing = 0;
    };
/* End PBXHeadersBuildPhase section */

/* Begin PBXNativeTarget section */
```

```
            CreatedOnToolsVersion = 10.2.1;
          };
          935B0BE122F27624007FC7C1 = {
            CreatedOnToolsVersion = 10.2.1;
          };
        };
      };
      buildConfigurationList = 935B0BC322F27614007FC7C1 /*
Build configuration list for PBXProject "MyContactApp" */;
      compatibilityVersion = "Xcode 9.3";
      developmentRegion = en;
      hasScannedForEncodings = 0;
      knownRegions = (
        en,
        Base,
      );
      mainGroup = 935B0BBF22F27614007FC7C1;
      productRefGroup = 935B0BC922F27614007FC7C1 /*
Products */;
```

A brand new shell app + framework project file contains 513 lines

Scott Berrevoets
@ScottBerrevoets

When the entire tree for both apps is generated into 1 Xcode project, the .pbxproj file is 750k lines or 41MB

1:36 PM · Jul 16, 2019 · Tweetbot for Mac

5 Retweets **41** Likes

XcodeGen Primer

The Project Spec

```
Repo Root
 |-- project.yml
 |-- Modules
  |-- App
   |-- Sources
    |-- Main.swift
    |-- // other sources
   |-- xcconfigs
  |-- DataModel
   |-- Sources
    |-- Contact.swift
   |-- xcconfigs
```

The Project Spec

```
name: MyContactApp
options:
  bundleIdPrefix: com.myapp
targets:
  MyContactApp:
    type: application
    platform: iOS
    deploymentTarget: "10.0"
    sources: [Modules/App/Sources]
    dependencies:
      - target: DataModel
      - sdk: Contacts.framework
     configFiles:
       Debug: xcconfigs/App-Debug.xcconfig
       Release: xcconfigs/App-Release.xcconfig
DataModel:
    type: framework
    platform: iOS
      sources: [Modules/DataModel/Sources]
    configFiles:
      Debug: xcconfigs/DataModel-Debug.xcconfig
      Release: xcconfigs/DataModel-Release.xcconfig
```

```
Repo Root
 |-- project.yml
 |-- Modules
  |-- App
   |-- Sources
    |-- Main.swift
    |-- // other sources
   |-- xcconfigs
  |-- DataModel
   |-- Sources
    |-- Contact.swift
   |-- xcconfigs
```

Breaking Up

```
name: MyContactApp
include:
  - Modules/App/app.yml
  - Modules/DataModel/DataModel.yml
```

```
// Modules/DataModel.yml
targets:
  DataModel:
    type: framework
    platform: iOS
    sources:
      - path: Sources
        name: DataModel
    configFiles:
      Debug: xcconfigs/DataModel-Debug.xcconfig
      Release: xcconfigs/DataModel-Release.xcconfig
```

Breaking Up

```
name: MyContactApp
include:
  - Modules/App/app.yml
  - Modules/DataModel/DataModel.yml
  - Modules/Networking/Networking.yml

// Modules/DataModel.yml
targets:
  DataModel:
    type: framework
    platform: iOS
    sources:
      - path: Sources
        name: DataModel
    configFiles:
      Debug: xcconfigs/DataModel-Debug.xcconfig
      Release: xcconfigs/DataModel-Release.xcconfig

// Modules/Networking.yml
targets:
  Networking:
    type: framework
    platform: iOS
    sources:
      - path: Sources
        name: Networking
    configFiles:
      Debug: xcconfigs/Networking-Debug.xcconfig
      Release: xcconfigs/Networking-Release.xcconfig
```

Target Templates

```
targetTemplates:
  Framework:
    type: framework
    platform: iOS
    configFiles:
      Debug: Modules/${target_name}/xcconfigs/${target_name}-Debug.xcconfig
      Release: Modules/${target_name}/xcconfigs/${target_name}-Release.xcconfig
    sources:
      - path: Modules/${target_name}/Sources
        name: ${sourceName}

// Updated Modules/DataModel/DataModel.yml
targets:
  DataModel:
    templates:
      - Framework
    templateAttributes:
      sourceName: AwesomeFramework
```

Schemes

- Auto-generated for each target
- Target scheme
  - Add test targets
  - Supply your own config variants (other than debug/release)
  - Add pre/post actions
  - Cannot rename the scheme from the default
- Project scheme
  - Allows for additional control than a target scheme
  - Can fully configure the scheme like you can in Xcode’s scheme editor

Workflow Integration

- Installation
  - Mint (package manager for Swift CLI tools)
  - Homebrew
  - Download and run `make`
  - Specify a version and let a script handle it (👍)
- Triggering
  - `xcodegen generate`
  - Use as part of an automated process

iOS Project Template

https://github.com/jsorge/ios-project-template

Executable File | 36 lines (27 sloc) | 808 Bytes

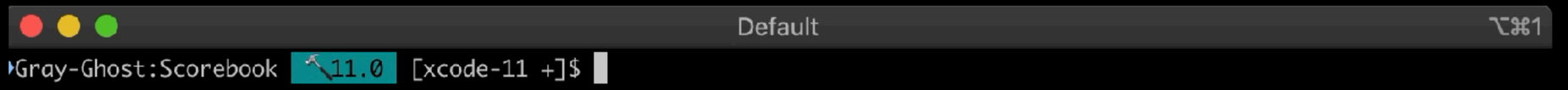
Raw | Blame | History

```
#!/usr/bin/env bash

set -e
set -o pipefail
set -u

required_version="$(cat .xcodegen-version)"
install_location=./vendor

install() {
  if [ ! -d $install_location ]; then
    mkdir $install_location;
  fi;

  rm -f ./tmp/XcodeGen ./tmp/xcodegen.tar.gz

  curl --location --fail --retry 5 \
    https://github.com/yonaskolb/XcodeGen/releases/download/"$required_version"/xcodegen.zip \
    --output $install_location/xcodegen.zip

  (
    cd $install_location
    unzip -o xcodegen.zip -d download > /dev/null
    mv download/xcodegen/bin/xcodegen XcodeGen
    rm -rf xcodegen.zip download
  )

  echo "Installed XcodeGen locally"
}

if [ ! -x $install_location/XcodeGen ]; then
  install
elif ! diff <(echo "Version: $required_version") <($install_location/XcodeGen version) > /dev/null; then
  install
fi
```

Using Make

In your Makefile:

```
.PHONY: project
project:
	@./tools/ensure-xcodegen.sh
	./vendor/XcodeGen generate
```


Default
⌥⌘1
Gray-Ghost:Scorebook
11.0
[xcode-11 +]$

Pain Points

- When you do a pull, merge, or otherwise get changes from upstream you'll have to re-make your project
- Xcode sometimes doesn't like the project file changing from underneath it
  - Script closing project, re-making, and re-opening the project
- CI setup may be more complicated if your CI provider assumes a project is checked in for setting their service up
  - In a post-checkout CI step, run your project generation command

Next Steps

- Move your build settings to xcconfig files, don’t put any in your project
  - Olof’s Xcoders xcconfig talk
    - https://vimeo.com/274817680
  - James Dempsey’s Build Settings Extractor Mac App
    - https://github.com/dempseyatgithub/BuildSettingExtractor
- Delete the project file from your repo, add `*.xcodeproj` to your git ignore file

Daniel Duan
@daniel_duan

1. put all Xcode build settings into xcconfigs (github.com/dempseyatgithu...)
2. set up XcodeGen (github.com/yonaskolb/Xcod...)
3. delete and git ignore .xcodeproj
4. enlightenment

yonaskolb/XcodeGen
A Swift command line tool for generating your Xcode project - yonaskolb/XcodeGen
github.com

6:57 PM · Jul 28, 2019 · Twitter Web App

3 Retweets **37** Likes

Don’t Fear the Project

Jared Sorge
https://jsorge.net
jared@jsorge.net