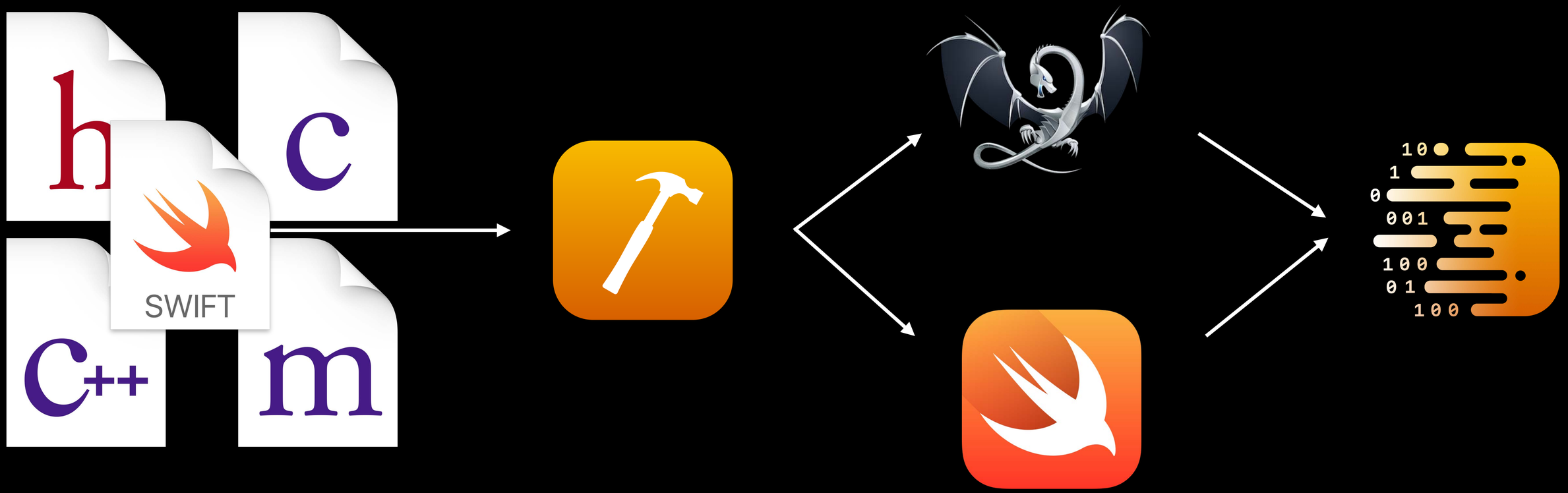
#WWDC18

Behind the Scenes of the Xcode Build Process

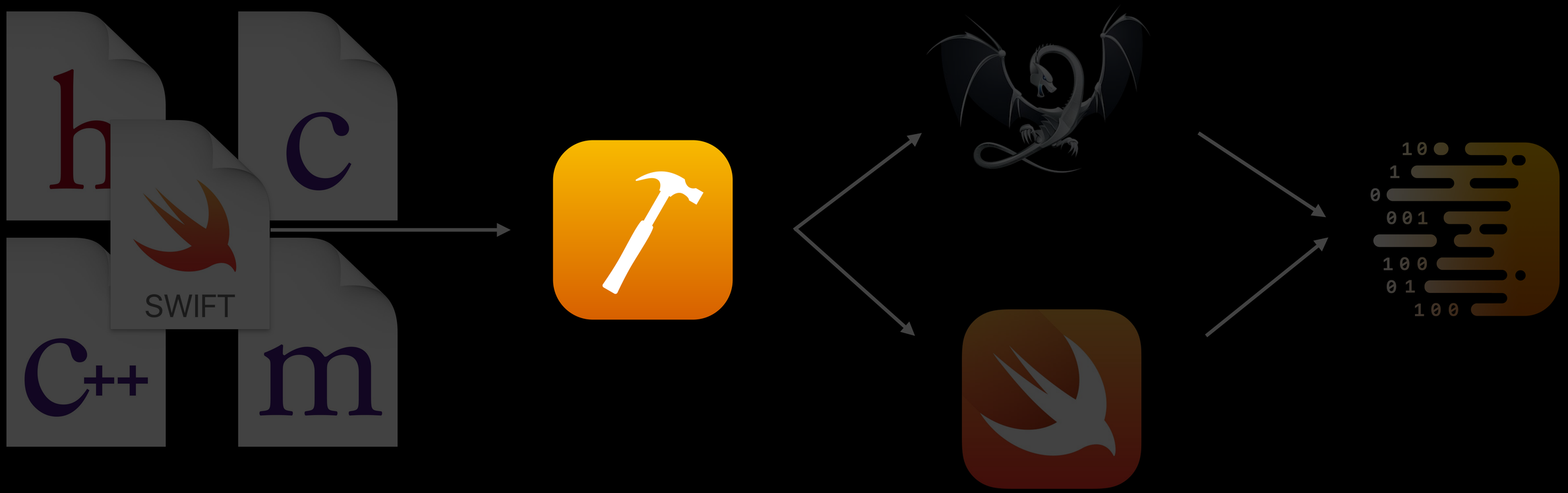
Session 415

Jake Petroules, Xcode Build System Team
Jürgen Ributzka, Clang Frontend Team
Devin Coughlin, Program Analysis Team
Louis Gerbarg, Linker Team

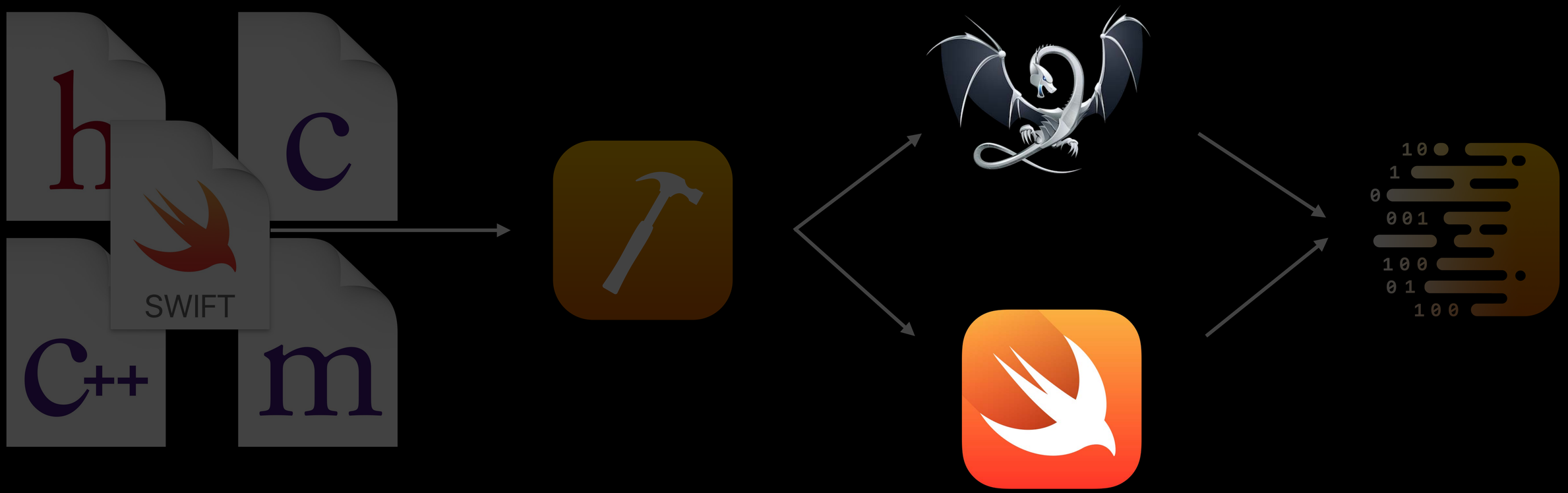
Exploring the Xcode Build Process



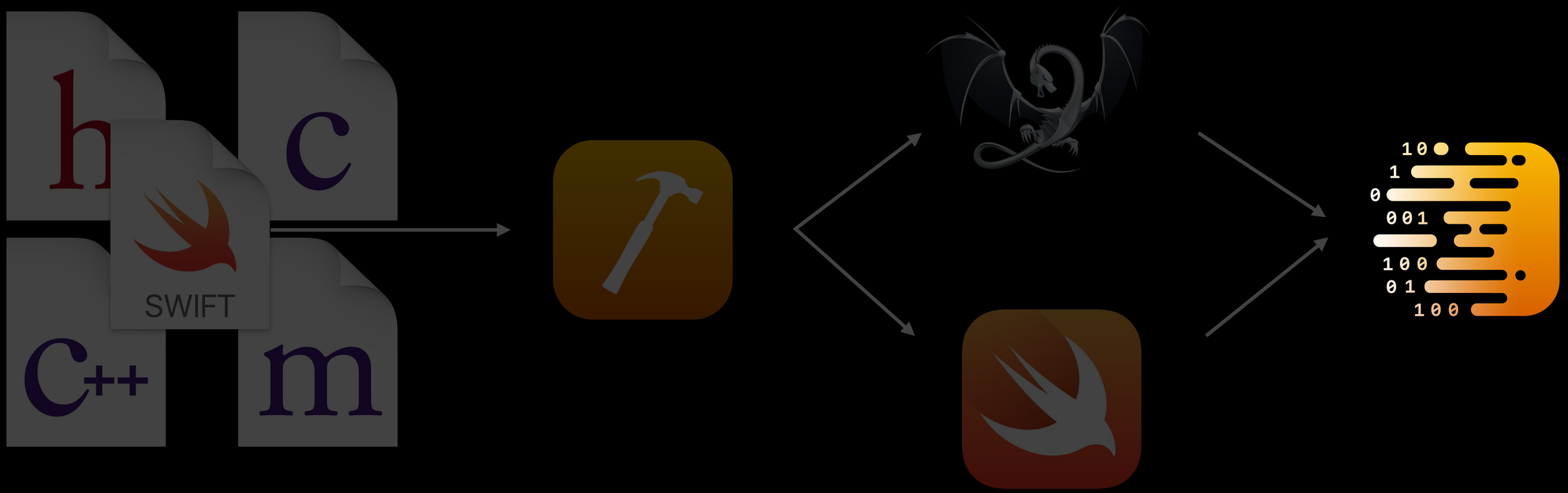
Exploring the Xcode Build Process: Build System



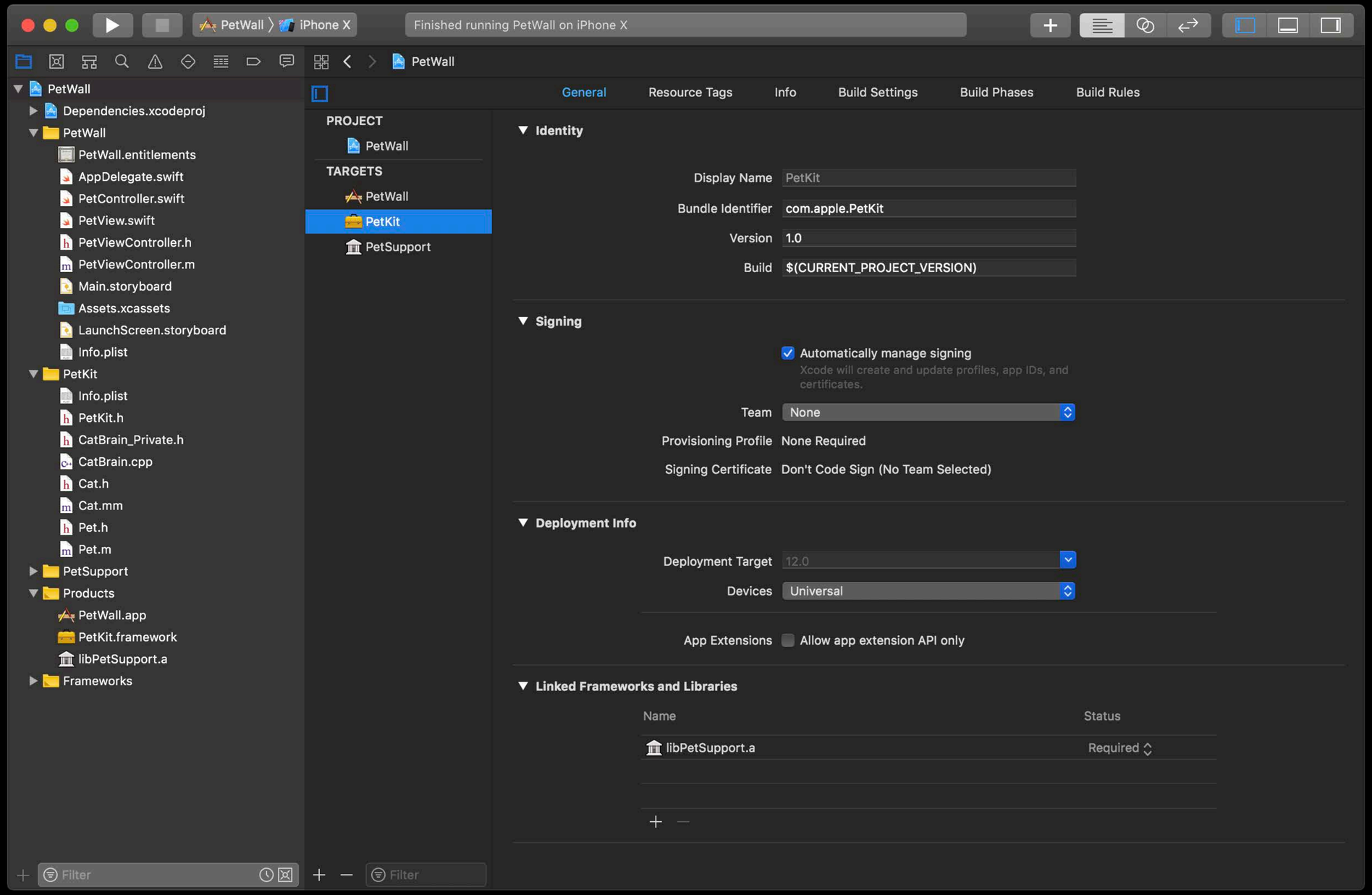
Exploring the Xcode Build Process: Clang and Swift



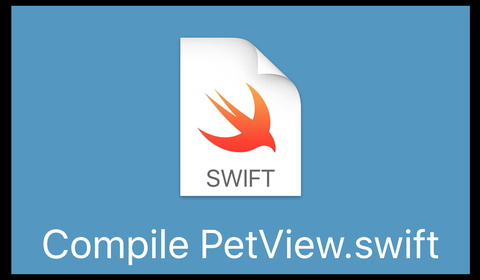
Exploring the Xcode Build Process: Linker

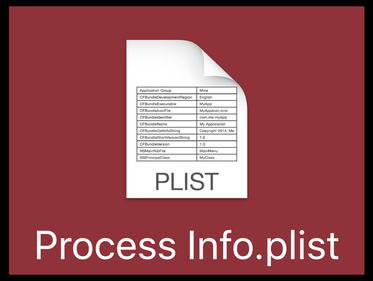



9:41
Feed Kitty
Kitty is hungry

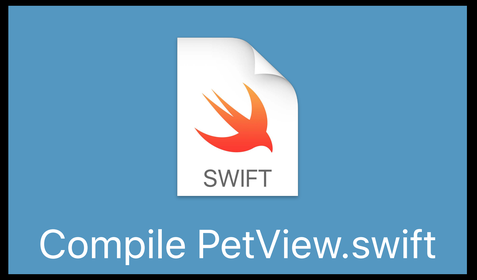

PetWall 〉 iPhone X
Finished running PetWall on iPhone X
PetWall
PetWall
Dependencies.xcodeproj
PetWall
PetWall.entitlements
AppDelegate.swift
PetController.swift
PetView.swift
PetViewController.h
PetViewController.m
Main.storyboard
Assets.xcassets
LaunchScreen.storyboard
Info.plist
PetKit
Info.plist
PetKit.h
CatBrain_Private.h
CatBrain.cpp
Cat.h
Cat.mm
Pet.h
Pet.m
PetSupport
Products
PetWall.app
PetKit.framework
libPetSupport.a
Frameworks
Filter
PetWall
General
Resource Tags
Info
Build Settings
Build Phases
Build Rules
PROJECT
PetWall
TARGETS
PetWall
PetKit
PetSupport
Filter
Identity
Display Name
PetKit
Bundle Identifier
com.apple.PetKit
Version
1.0
Build
$(CURRENT_PROJECT_VERSION)
Signing
Automatically manage signing
Xcode will create and update profiles, app IDs, and certificates.
Team
None
Provisioning Profile
None Required
Signing Certificate
Don't Code Sign (No Team Selected)
Deployment Info
Deployment Target
12.0
Devices
Universal
App Extensions
Allow app extension API only
Linked Frameworks and Libraries
Name
Status
libPetSupport.a
Required

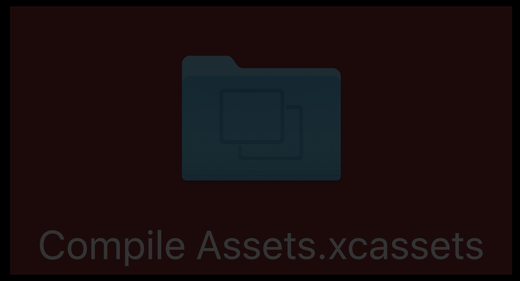
Build Process: Execution of a Collection of Tasks



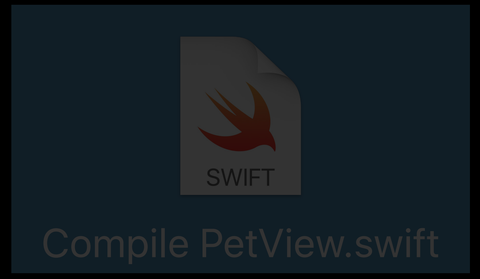


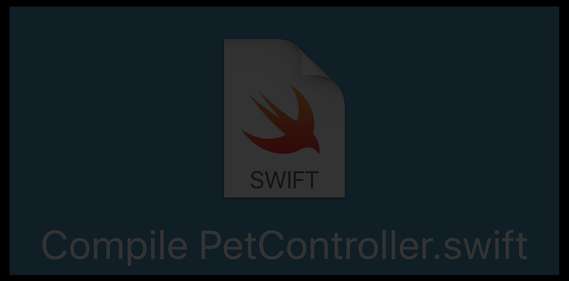
Build Process: Execution of a Collection of Tasks

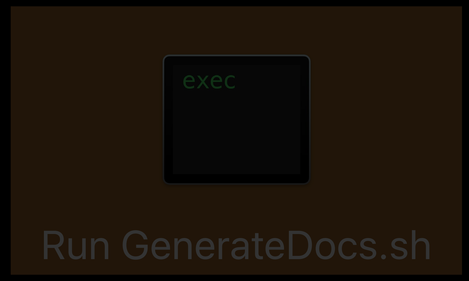


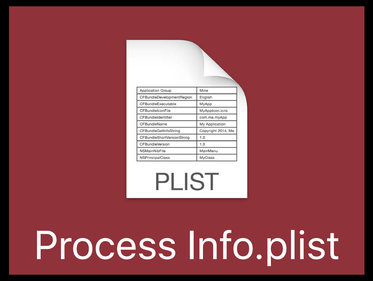


Build Process: Execution of a Collection of Tasks

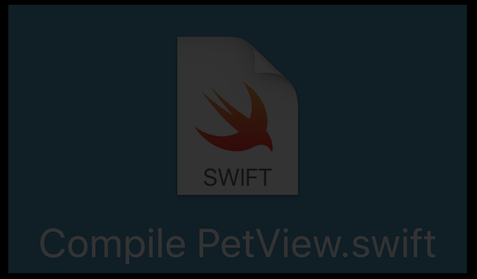


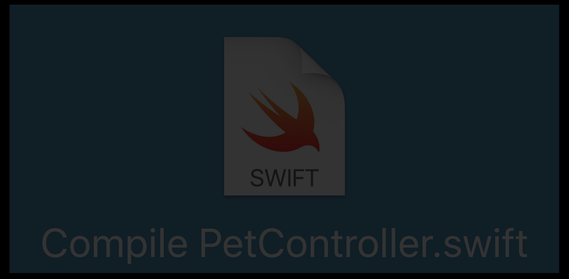


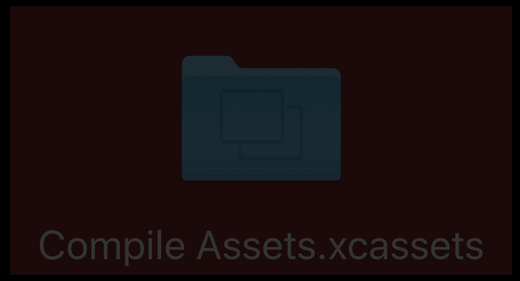


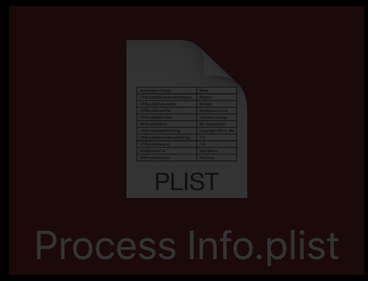


Build Process: Execution of a Collection of Tasks









Build Process: Execution of a Collection of Tasks

```
$ swiftc -module-name PetWall -target arm64-apple-ios12.0 -swift-version 4.2 ...
$ clang -x objective-c -arch arm64 ... PetViewController.m -o PetViewController.o
$ ld -o PetWall -framework PetKit PetViewController.o ...
$ actool --app-icon AppIcon ... Assets.xcassets
$ ...
$ [thousands more]
```

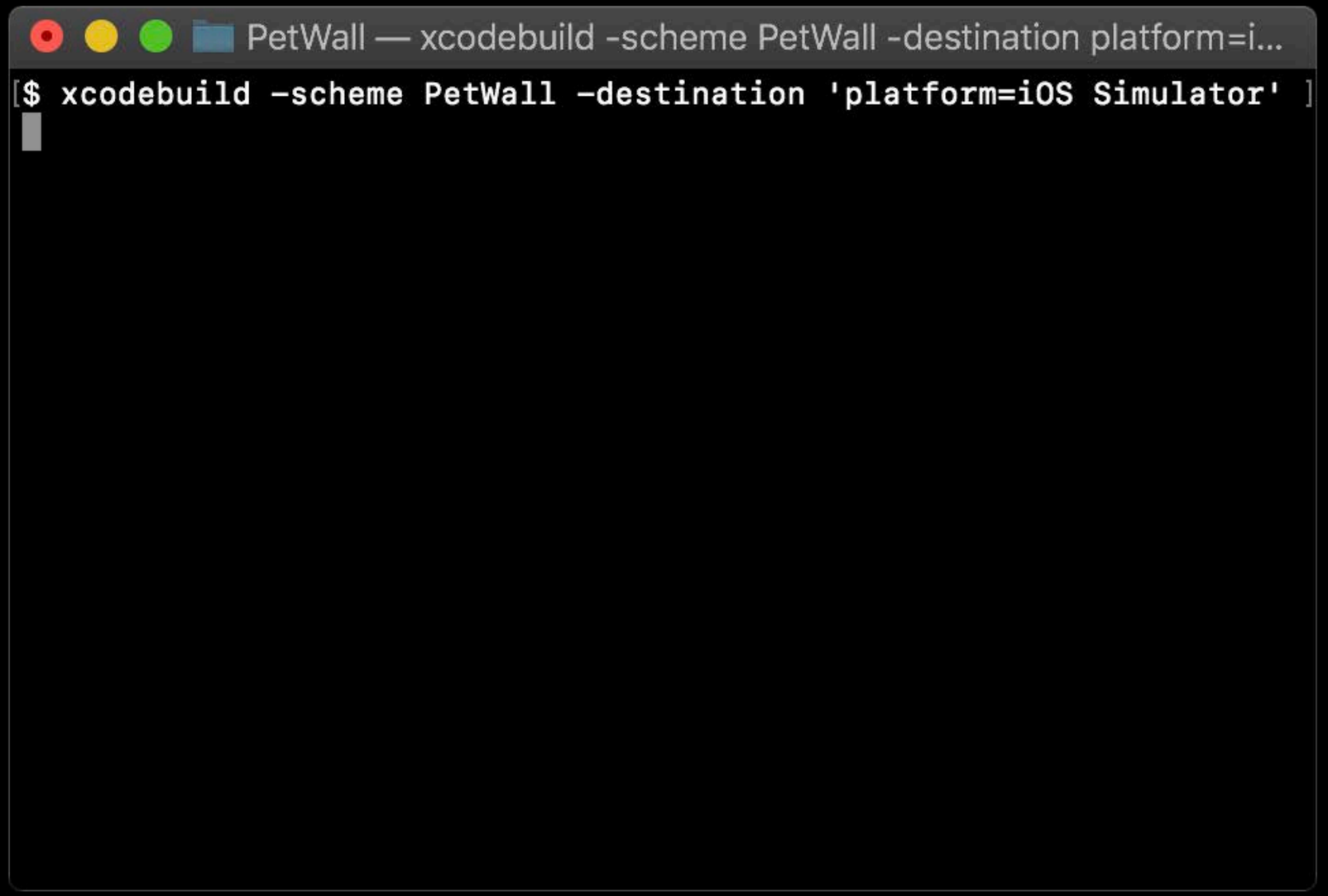
The build process
automates a series of tasks

PetKit — -bash — 67×21

```
[$ xcrun -sdk iphoneos clang -arch arm64 -miphoneos-version-min=12.0]
 -I.. -I../PetSupport -o Cat.o -c Cat.mm
[$ xcrun -sdk iphoneos clang -arch arm64 -miphoneos-version-min=12.0]
 -I.. -I../PetSupport -o Cat.o -c Cat.mm
[$ xcrun -sdk iphoneos clang -arch arm64 -miphoneos-version-min=12.0]
 -I.. -I../PetSupport -o Cat.o -c Cat.mm
[$ xcrun -sdk iphoneos clang -arch arm64 -miphoneos-version-min=12.0]
 -I.. -I../PetSupport -o Cat.o -c Cat.mm
[$ xcrun -sdk iphoneos clang -arch arm64 -miphoneos-version-min=12.0]
 -I.. -I../PetSupport -o Cat.o -c Cat.mm
[$ xcrun -sdk iphoneos clang -arch arm64 -miphoneos-version-min=12.0]
 -I.. -I../PetSupport -o Cat.o -c Cat.mm
[$ xcrun -sdk iphoneos clang -arch arm64 -miphoneos-version-min=12.0]
 -I.. -I../PetSupport -o Cat.o -c Cat.mm
[$ xcrun -sdk iphoneos clang -arch arm64 -miphoneos-version-min=12.0]
 -I.. -I../PetSupport -o Cat.o -c Cat.mm
[$ xcrun -sdk iphoneos clang -arch arm64 -miphoneos-version-min=12.0]
 -I.. -I../PetSupport -o Cat.o -c Cat.mm
[$ xcrun -sdk iphoneos clang -arch arm64 -miphoneos-version-min=12.0]
 -I.. -I../PetSupport -o Cat.o -c Cat.mm
$ 
```

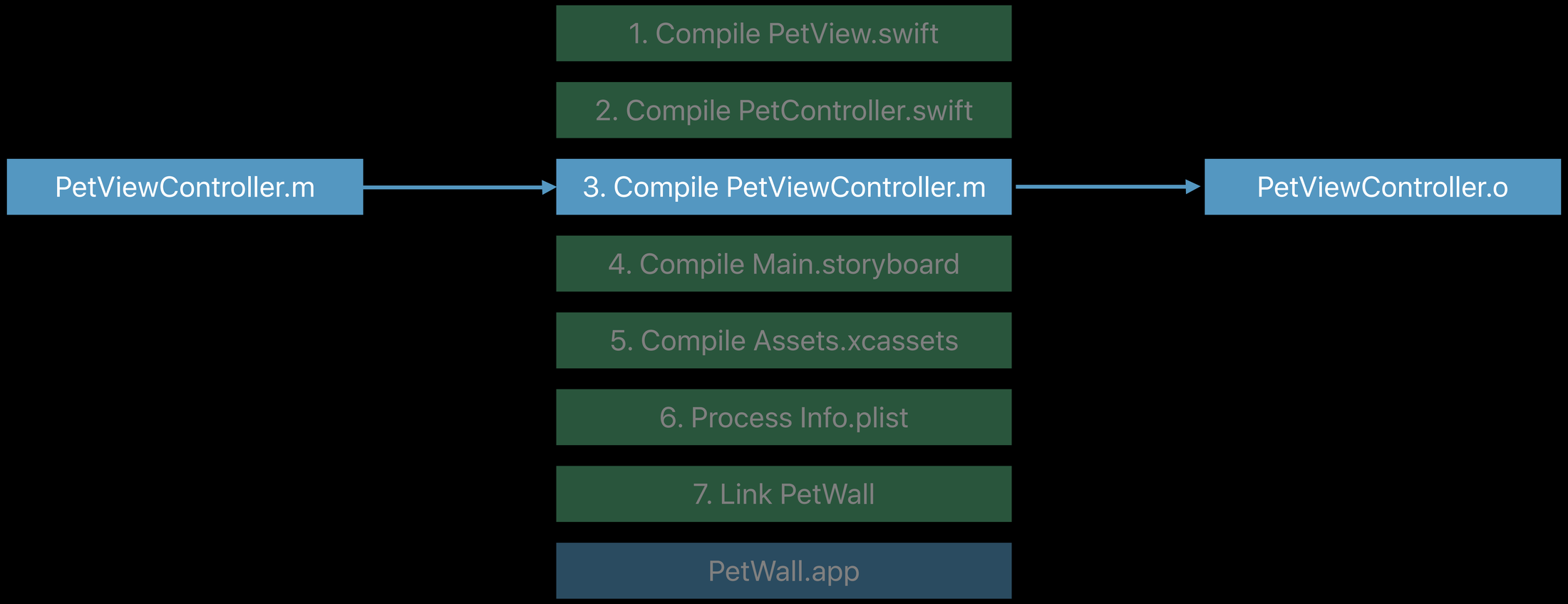
PetKit — -bash — 67×21

```
[$ xcrun -sdk iphoneos clang -arch arm64 -miphoneos-version-min=12.0]
 -I.. -I../PetSupport -o Cat.o -c Cat.mm
[$ xcrun -sdk iphoneos clang -arch arm64 -miphoneos-version-min=12.0]
 -I.. -I../PetSupport -o Cat.o -c Cat.mm
[$ xcrun -sdk iphoneos clang -arch arm64 -miphoneos-version-min=12.0]
 -I.. -I../PetSupport -o Cat.o -c Cat.mm
[$ xcrun -sdk iphoneos clang -arch arm64 -miphoneos-version-min=12.0]
 -I.. -I../PetSupport -o Cat.o -c Cat.mm
[$ xcrun -sdk iphoneos clang -arch arm64 -miphoneos-version-min=12.0]
 -I.. -I../PetSupport -o Cat.o -c Cat.mm
[$ xcrun -sdk iphoneos clang -arch arm64 -miphoneos-version-min=12.0]
 -I.. -I../PetSupport -o Cat.o -c Cat.mm
[$ xcrun -sdk iphoneos clang -arch arm64 -miphoneos-version-min=12.0]
 -I.. -I../PetSupport -o Cat.o -c Cat.mm
[$ xcrun -sdk iphoneos clang -arch arm64 -miphoneos-version-min=12.0]
 -I.. -I../PetSupport -o Cat.o -c Cat.mm
[$ xcrun -sdk iphoneos clang -arch arm64 -miphoneos-version-min=12.0]
 -I.. -I../PetSupport -o Cat.o -c Cat.mm
[$ xcrun -sdk iphoneos clang -arch arm64 -miphoneos-version-min=12.0]
 -I.. -I../PetSupport -o Cat.o -c Cat.mm
$
```

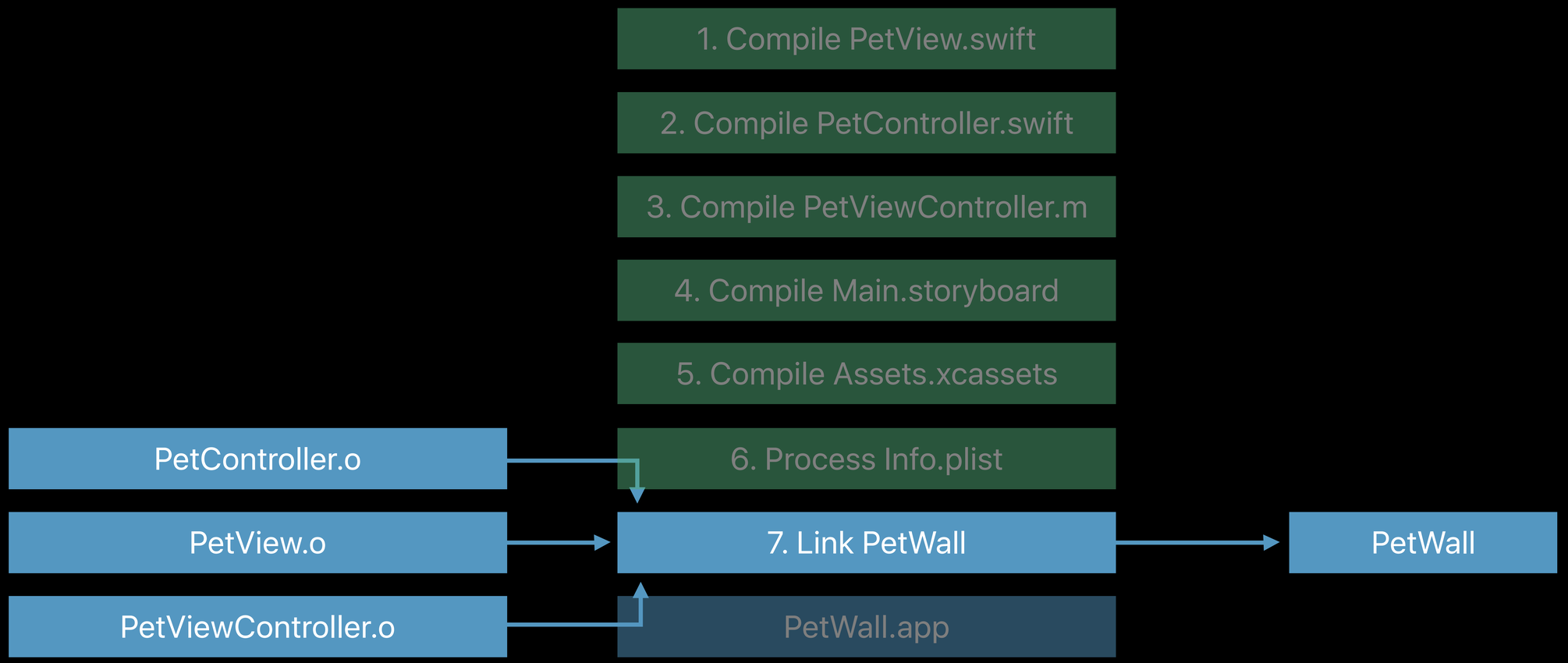

PetWall — xcodebuild -scheme PetWall -destination platform=i...
$ xcodebuild -scheme PetWall -destination 'platform=iOS Simulator'

Build Task Execution Order

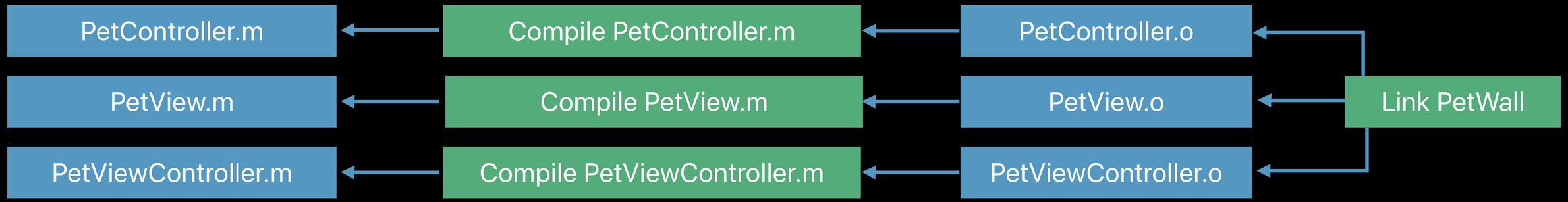
Build Task Execution Order



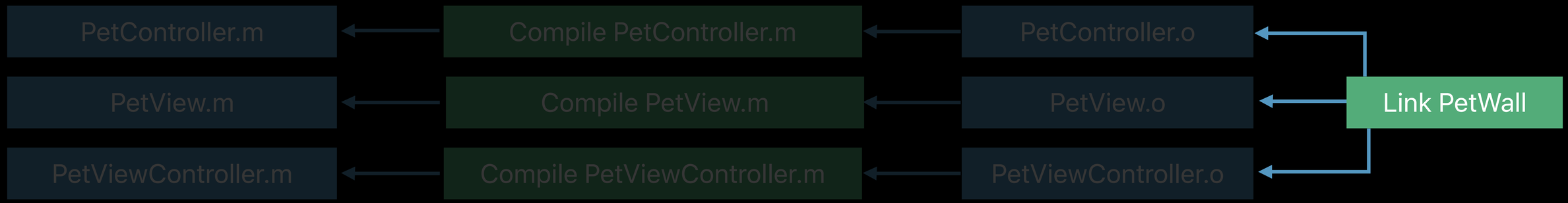
Build Task Execution Order



Build Task Dependency Order



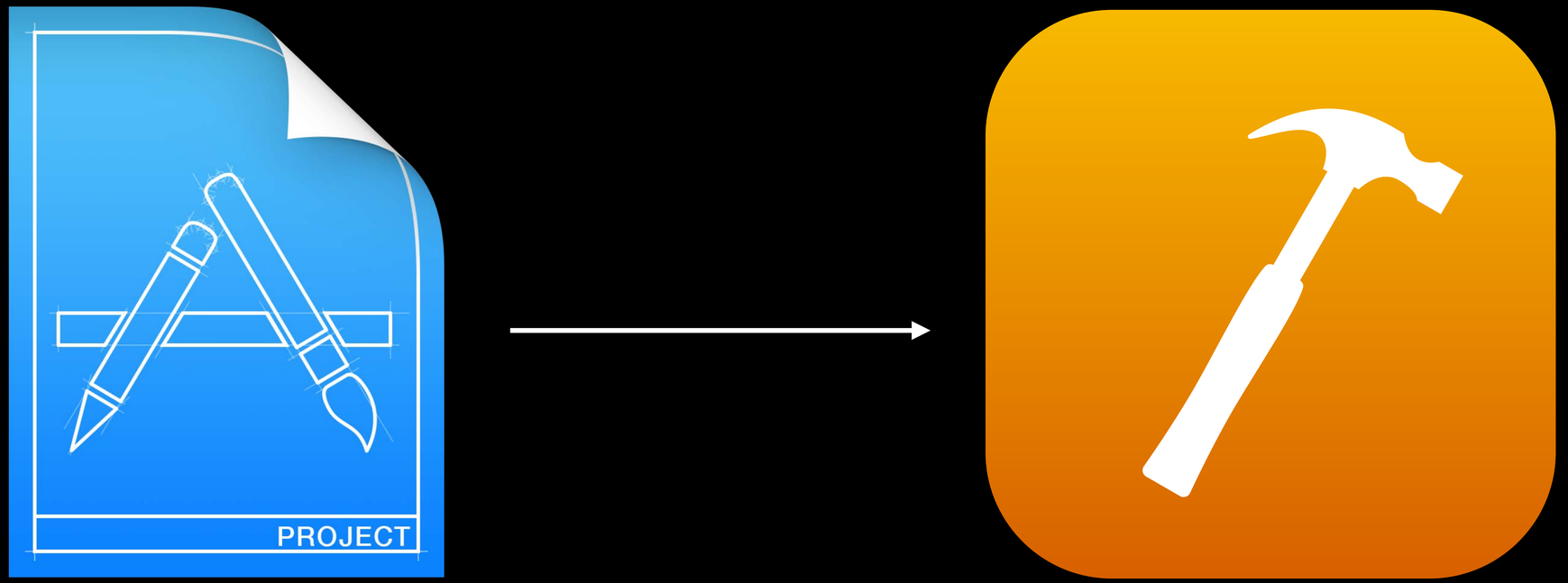
Build Task Dependency Order



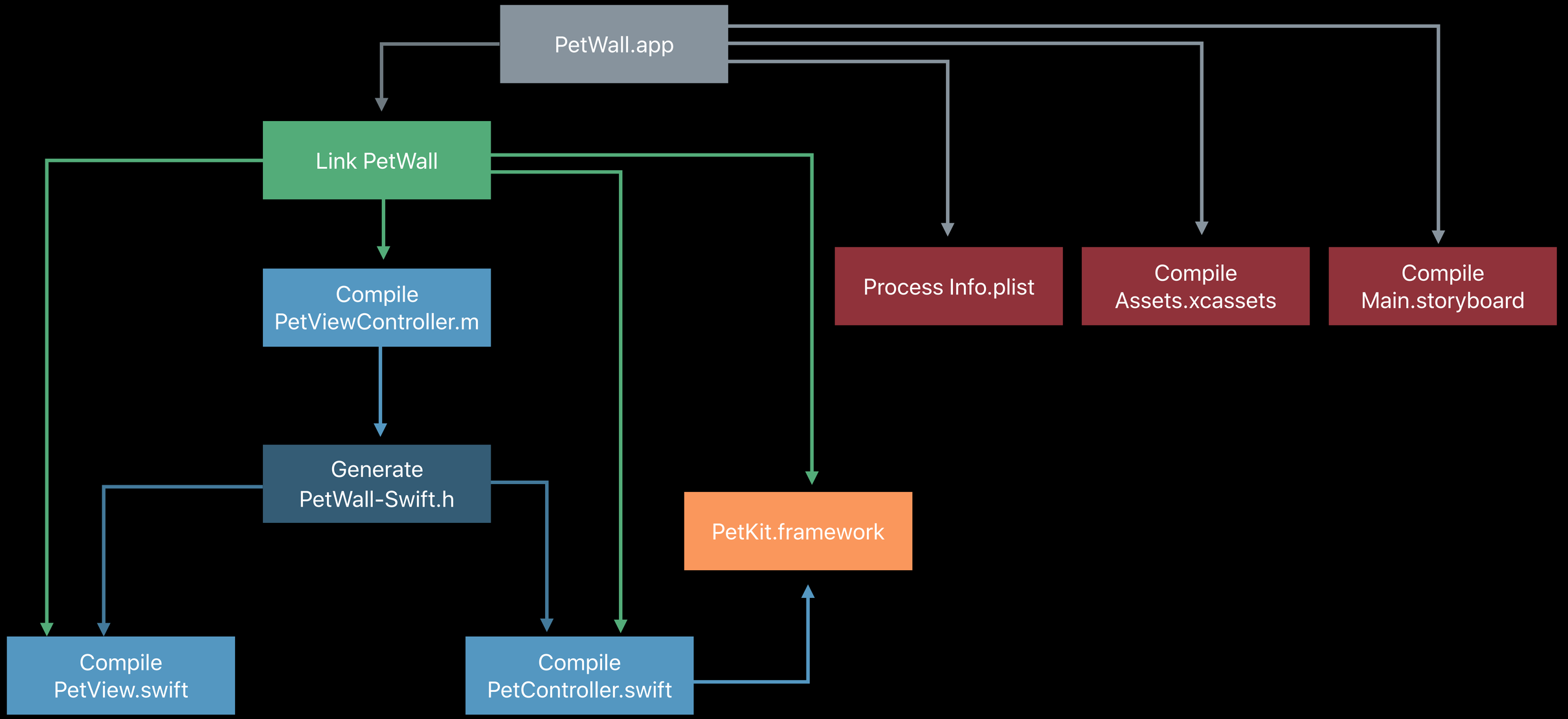
Build tasks are executed in
dependency order

What Happens When You Press Build?

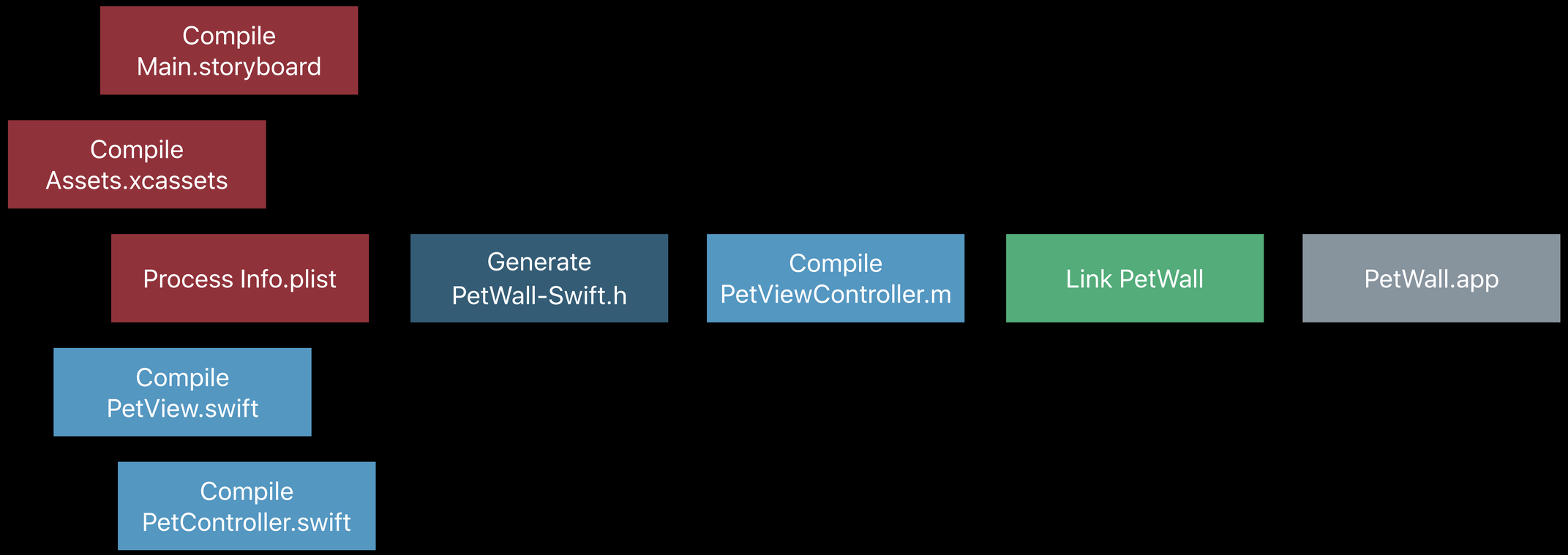
What Happens When You Press Build?



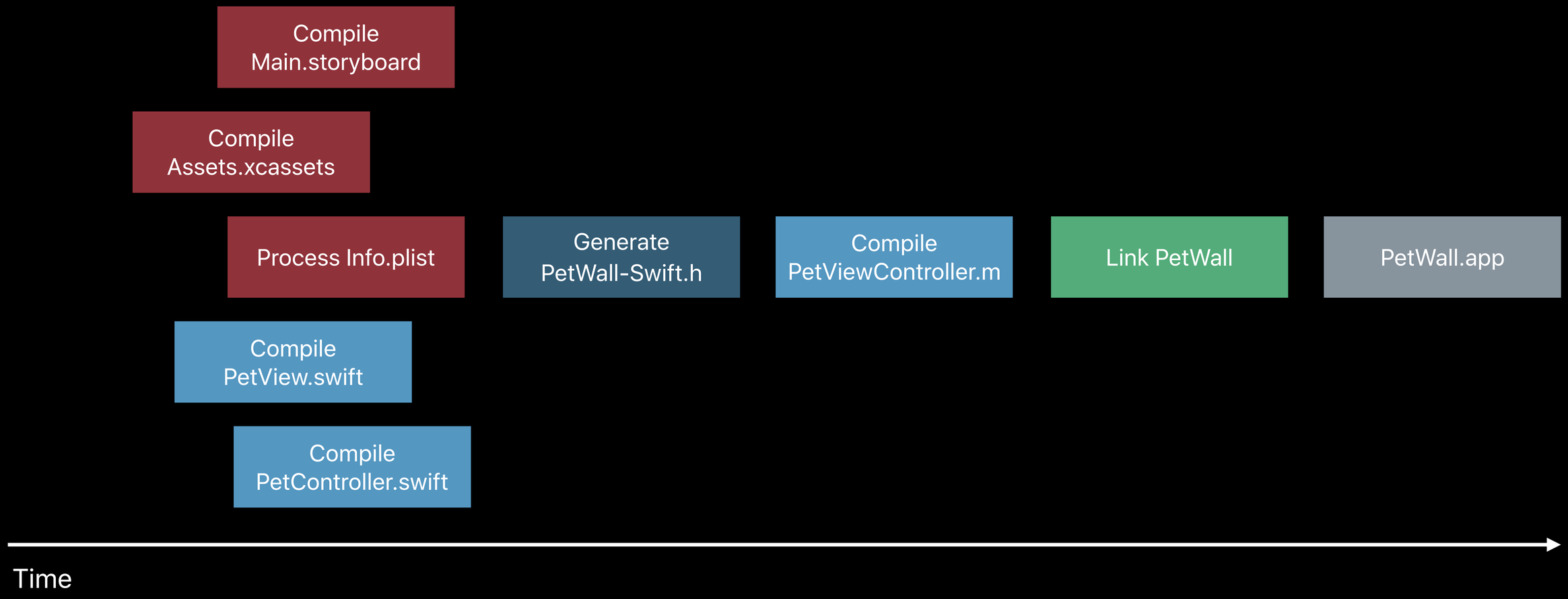
Build Process Represented as a Directed Graph



Execution of the Build Graph

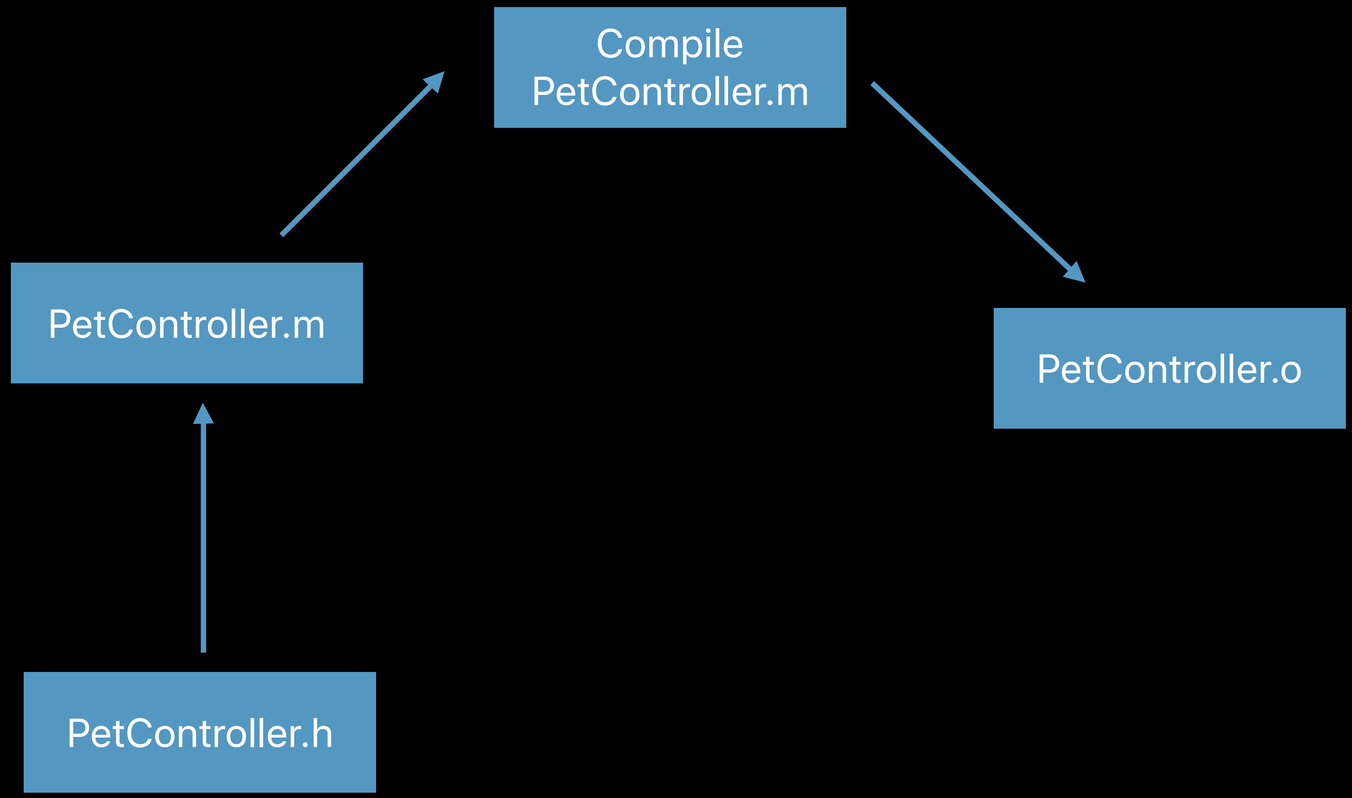


Execution of the Build Graph

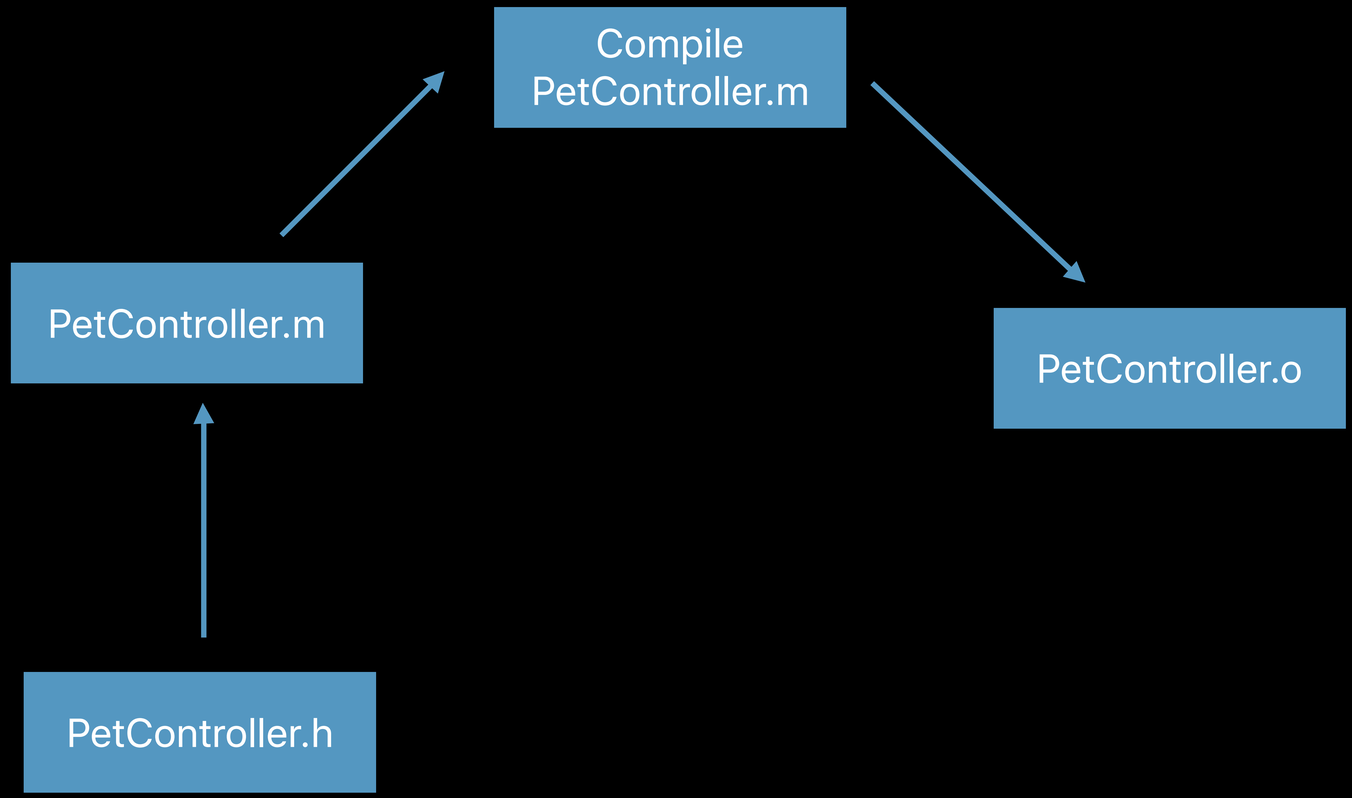


llbuild Execution Engine: Open Source

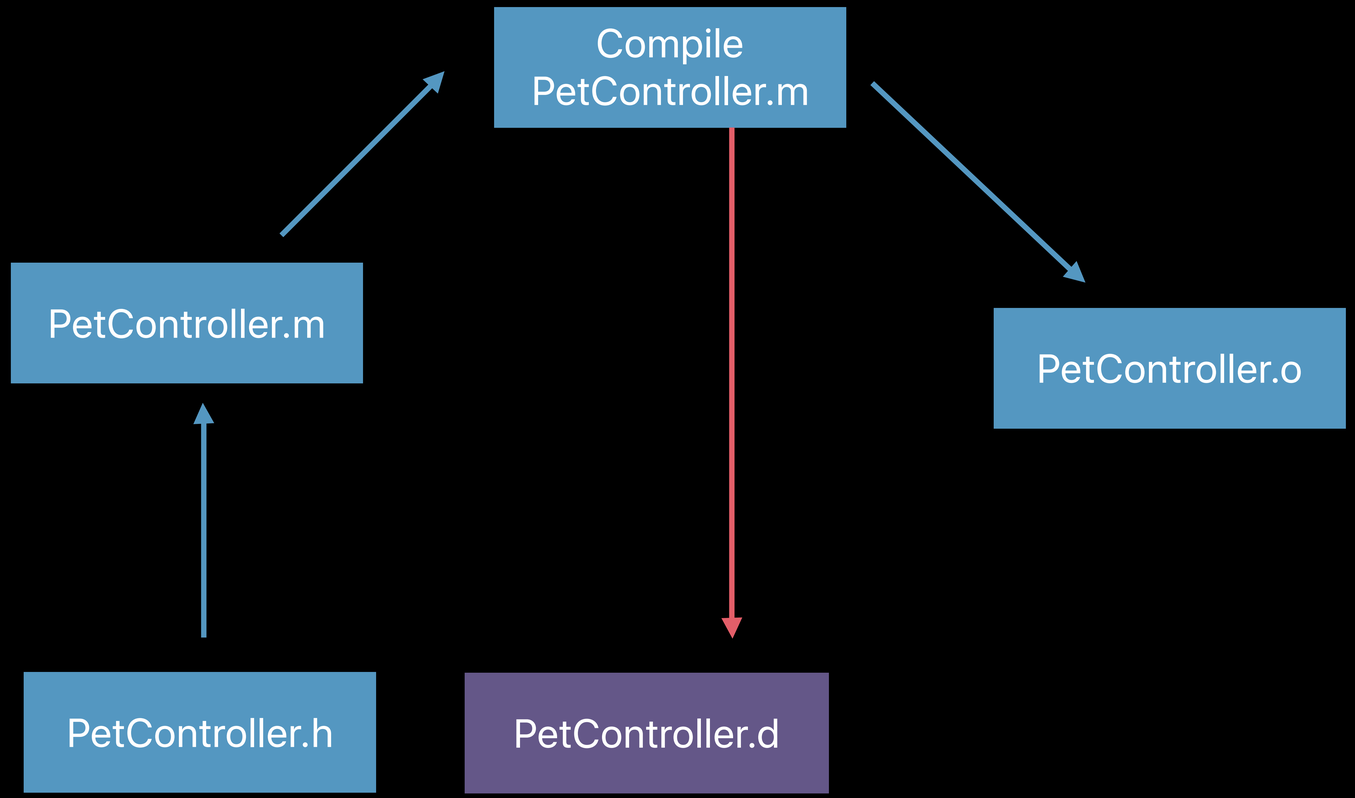
Discovered Dependencies



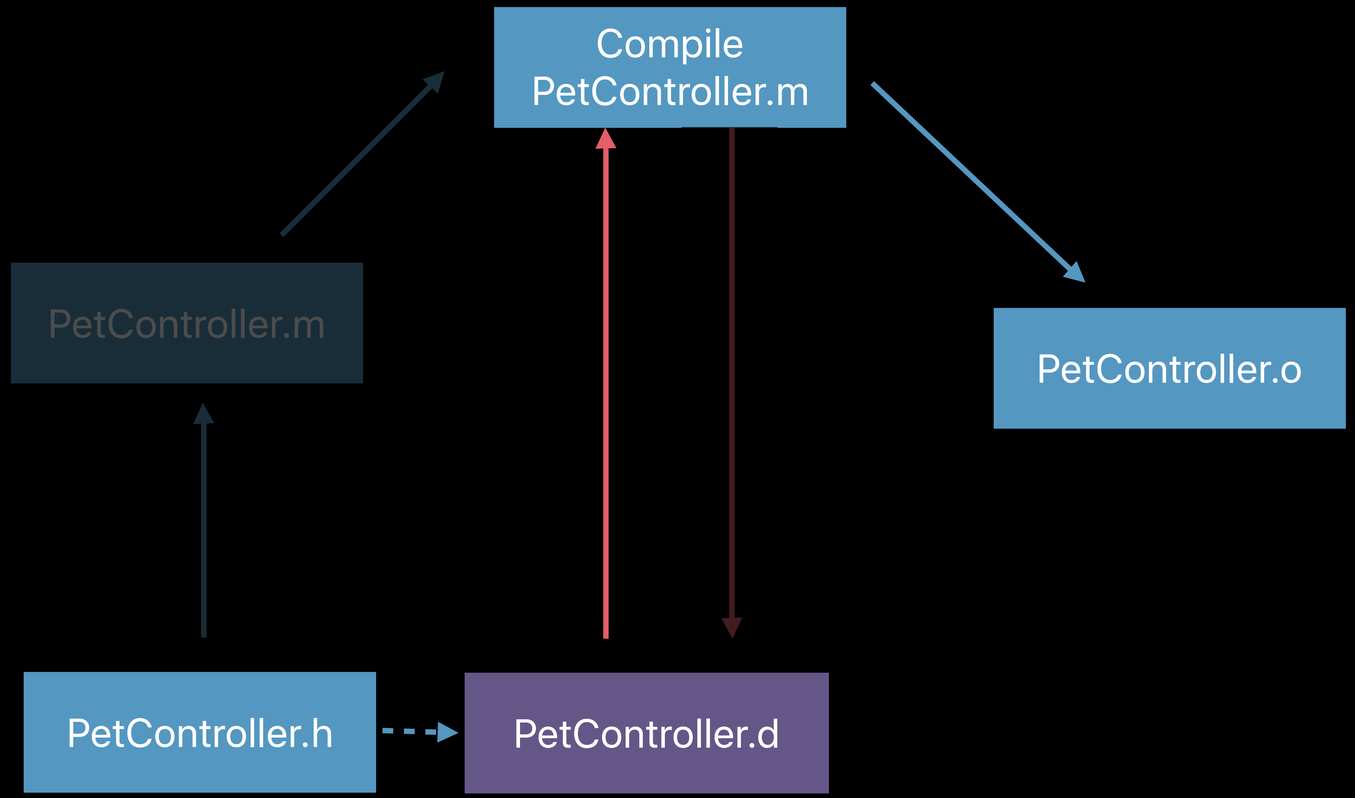
Discovered Dependencies



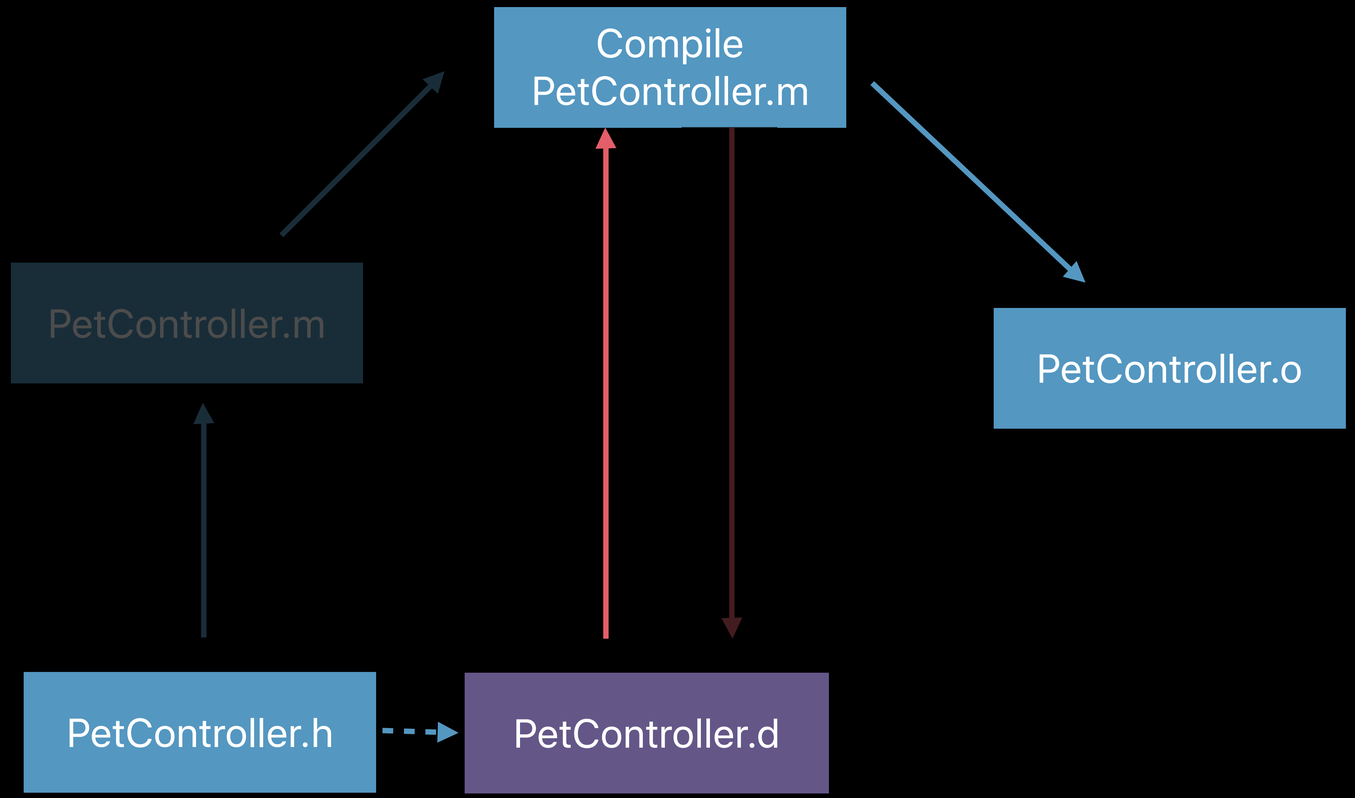
Discovered Dependencies



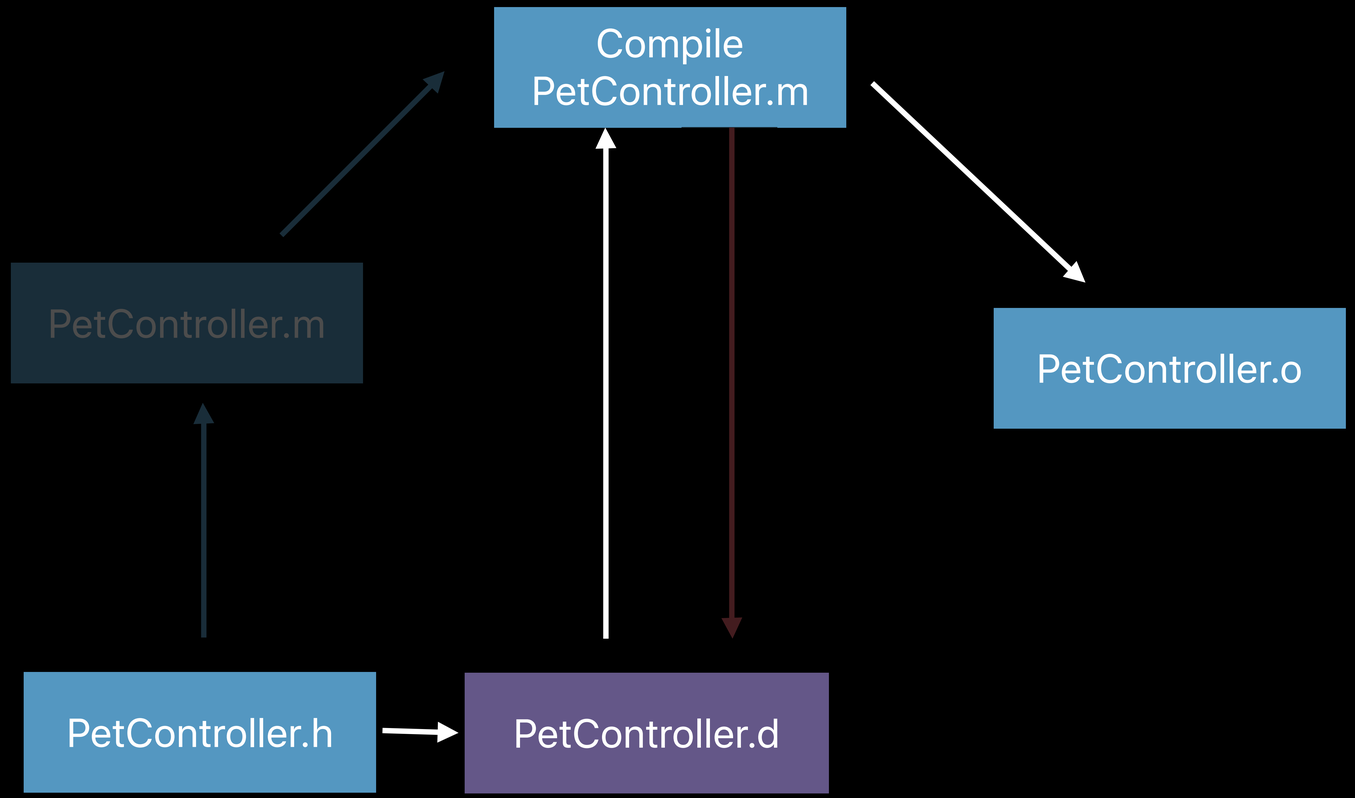
Discovered Dependencies



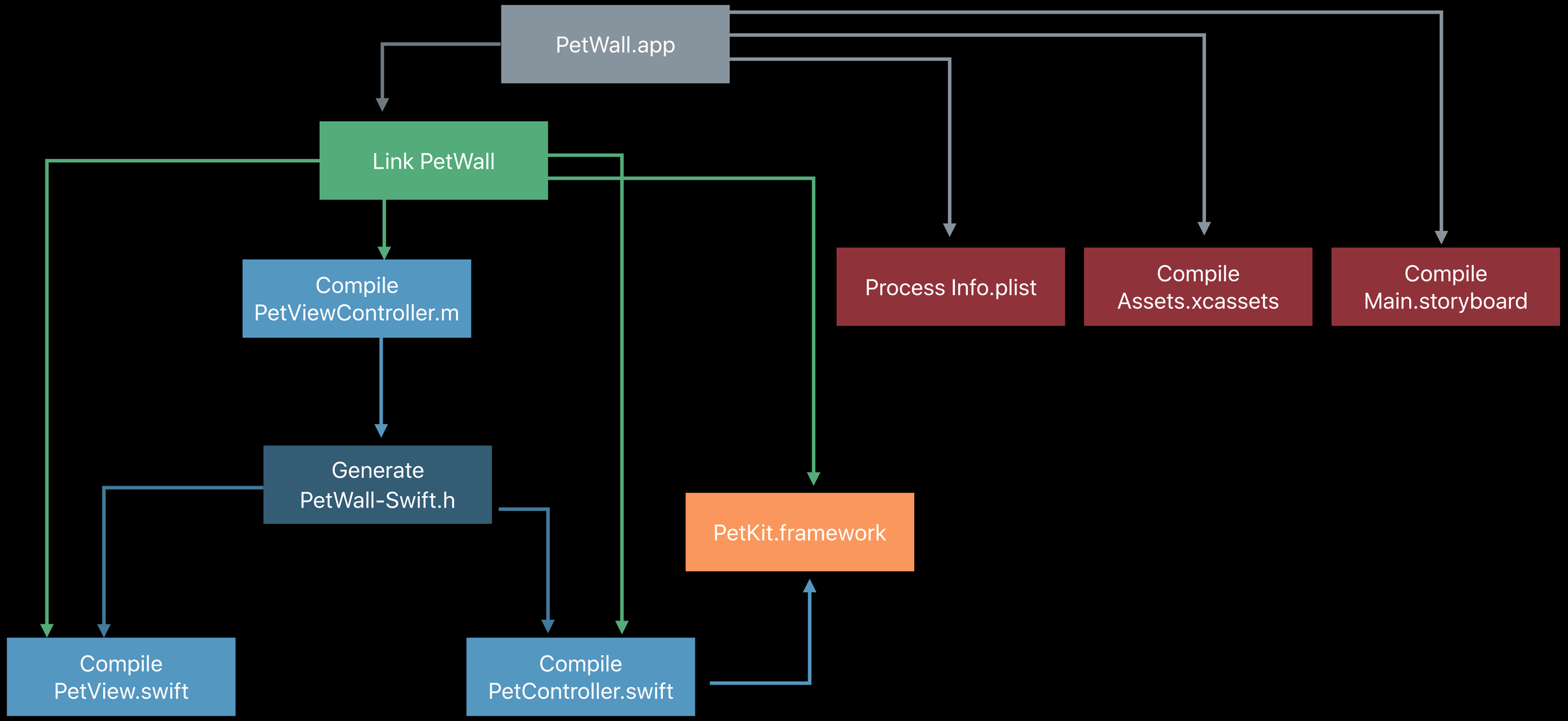
Discovered Dependencies



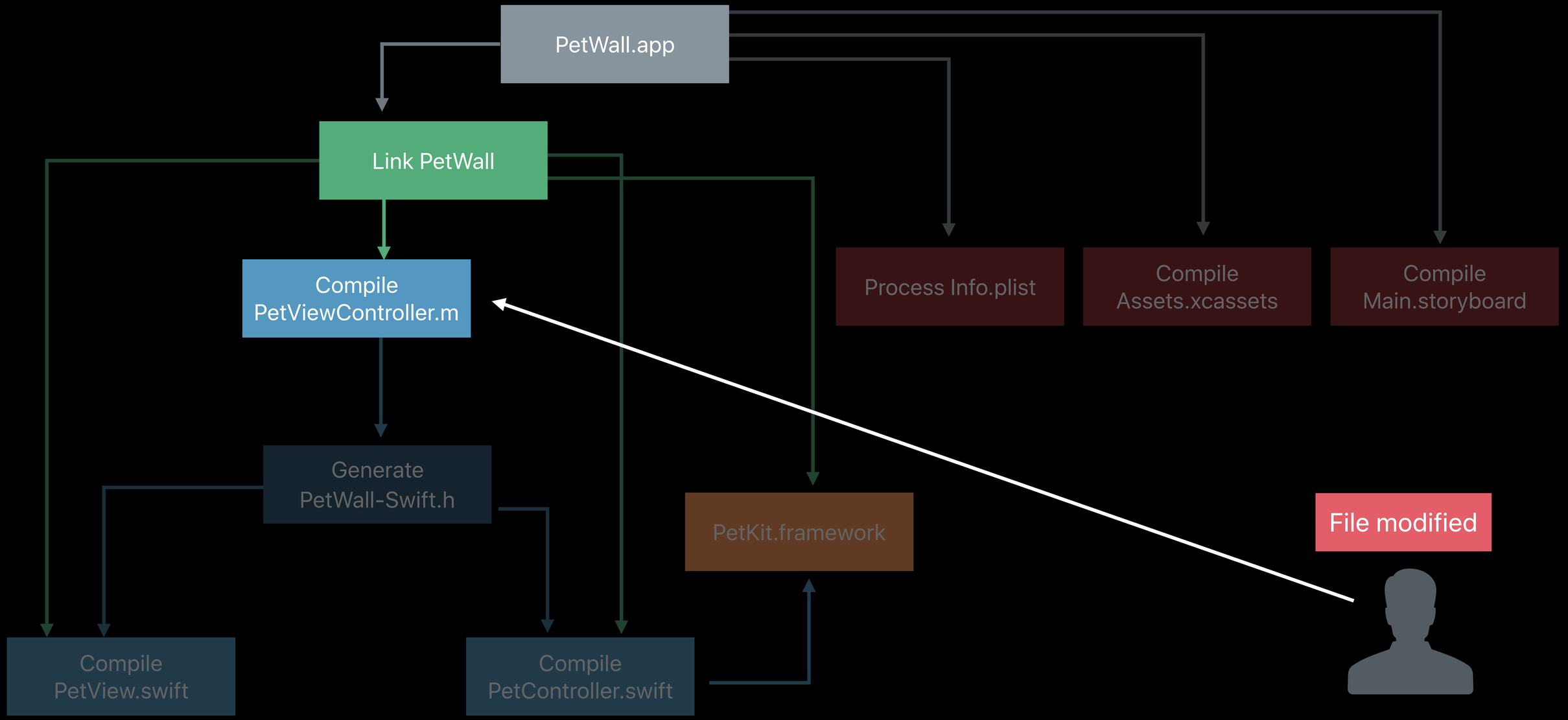
Discovered Dependencies



Incremental Builds



Incremental Builds



Change Detection and Task Signatures

Change Detection and Task Signatures

Each task in the build graph has a *signature*

Change Detection and Task Signatures

Each task in the build graph has a *signature*

Computed from stat info of inputs and other task metadata

Change Detection and Task Signatures

Each task in the build graph has a *signature*

Computed from stat info of inputs and other task metadata

Build system tracks task signatures of current and previous build

Change Detection and Task Signatures

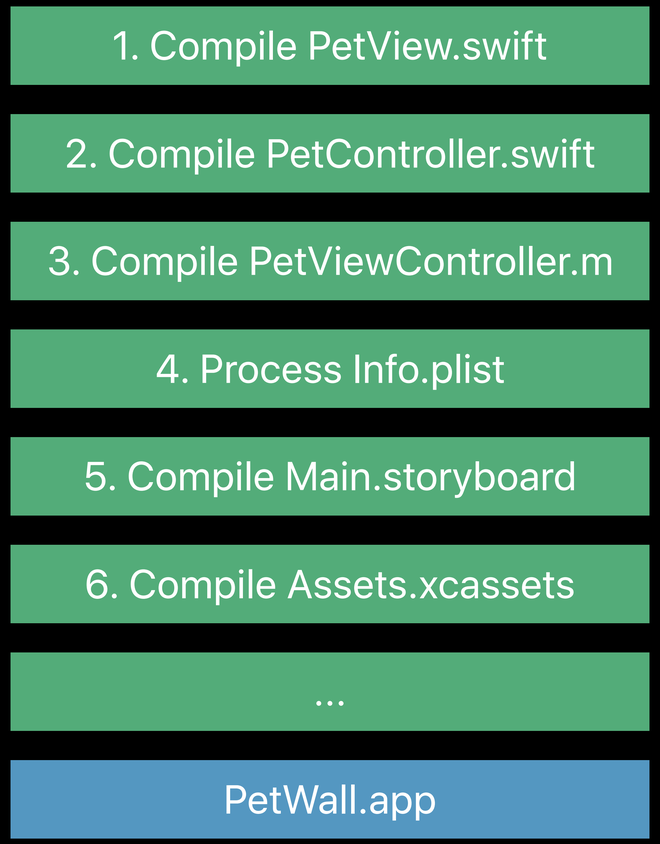
Each task in the build graph has a *signature*

Computed from stat info of inputs and other task metadata

Build system tracks task signatures of current and previous build

Compared to determine whether a task should be run

How can you help the build system?

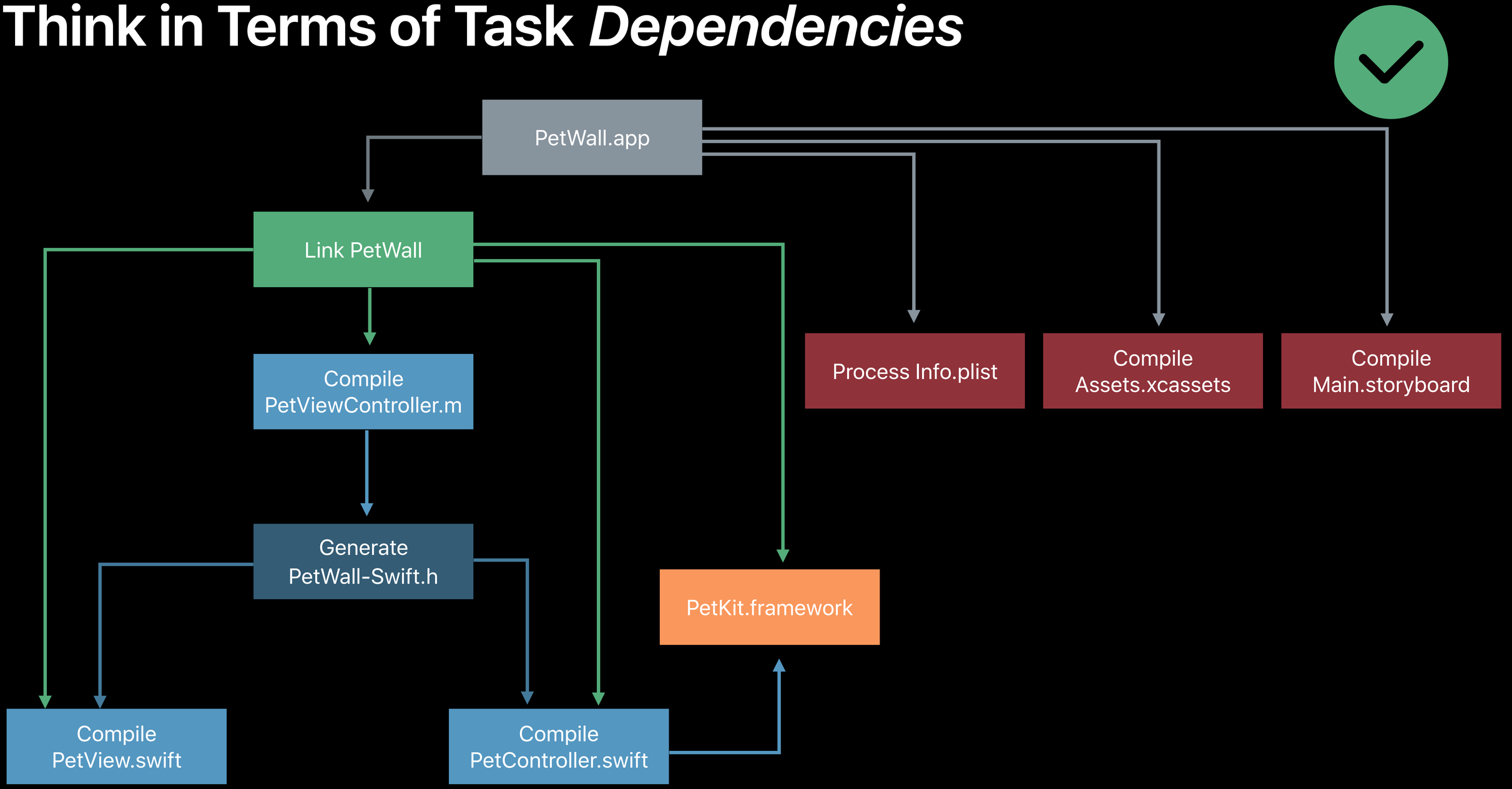

1. Compile PetView.swift
2. Compile PetController.swift
3. Compile PetViewController.m
4. Process Info.plist
5. Compile Main.storyboard
6. Compile Assets.xcassets
...
PetWall.app

Don't Think About the *Order* of Task Execution!

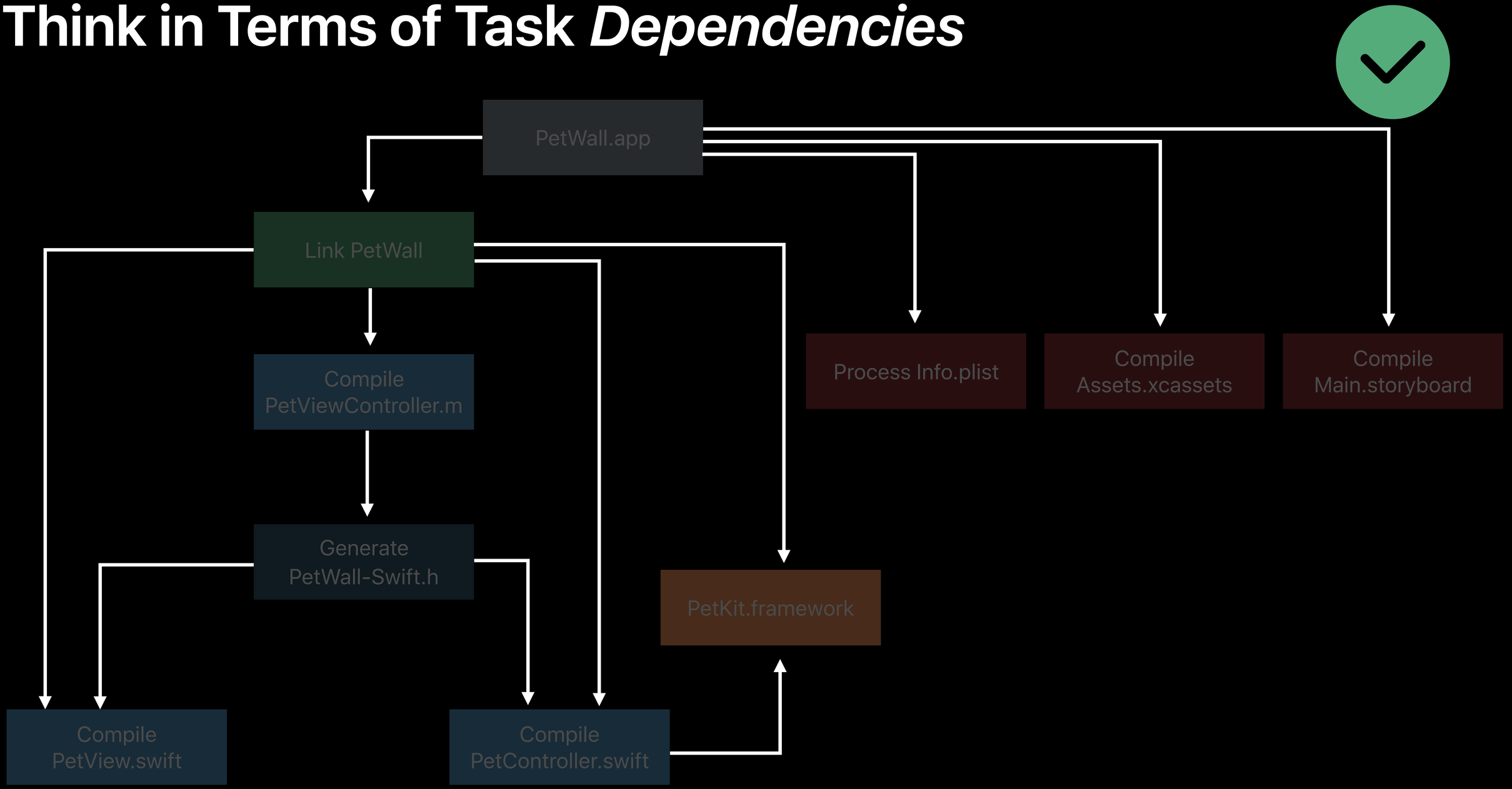
1. Compile PetView.swift
2. Compile PetController.swift
3. Compile PetViewController.m
4. Process Info.plist
5. Compile Main.storyboard
6. Compile Assets.xcassets
...

PetWall.app

Think in Terms of Task *Dependencies*



Think in Terms of Task *Dependencies*



Where Do Dependencies Come From?

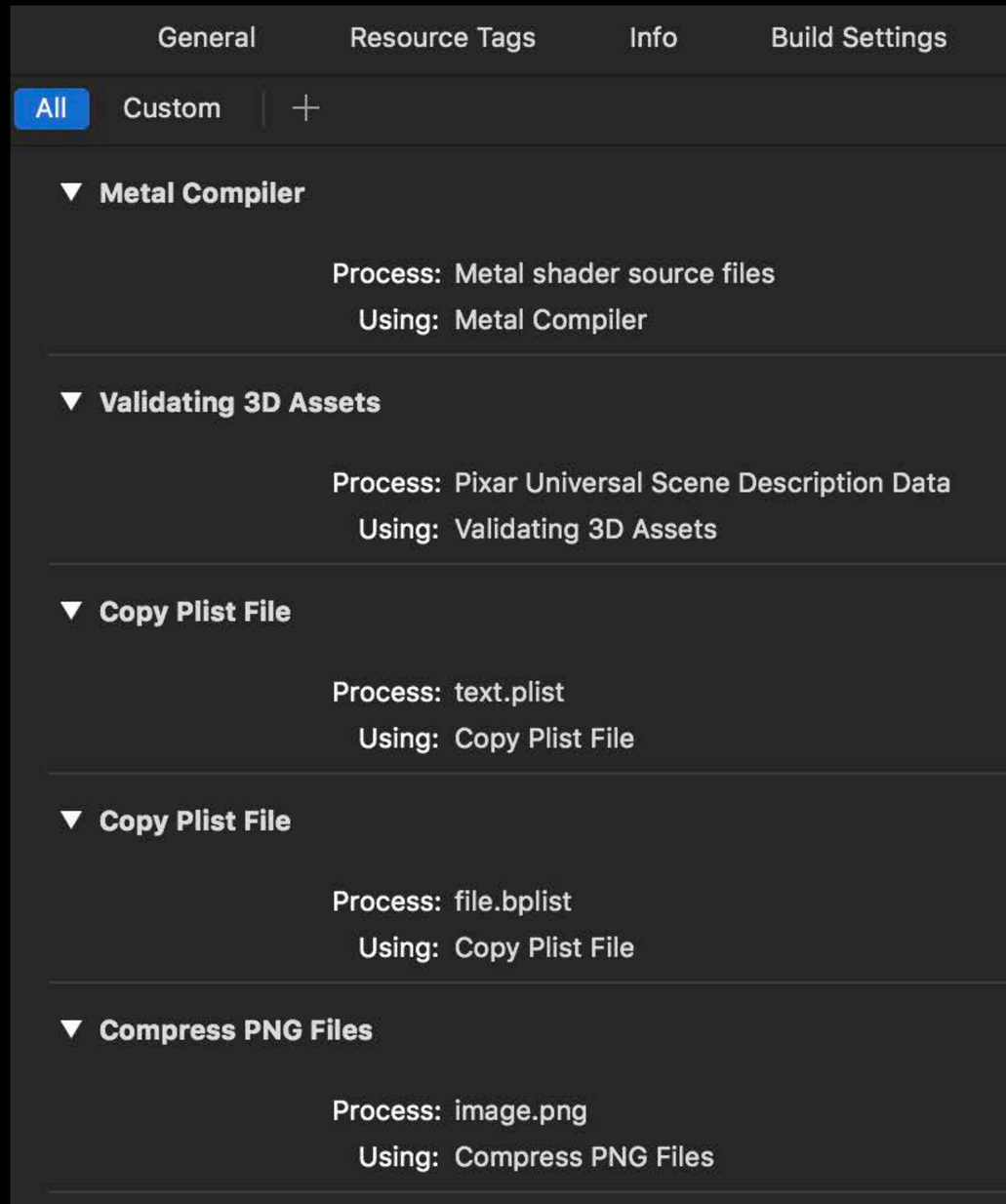
Built in

Target dependencies

Implicit dependencies

Build phase dependencies

Scheme order dependencies



Where Do Dependencies Come From?

Built in

Target dependencies

Implicit dependencies

Build phase dependencies

Scheme order dependencies

Where Do Dependencies Come From?

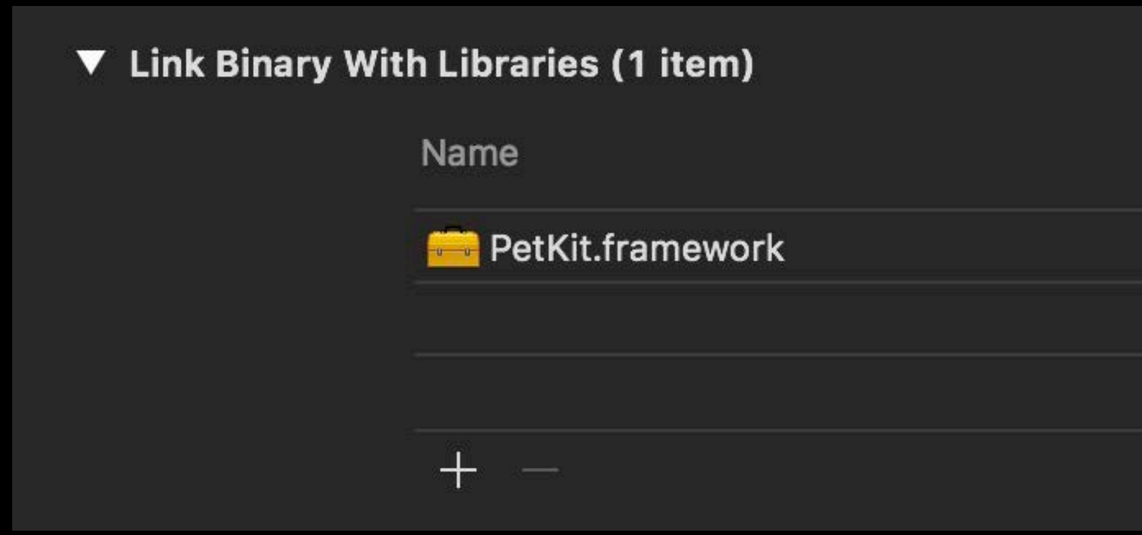
Built in

Target dependencies

Implicit dependencies

Build phase dependencies

Scheme order dependencies



Where Do Dependencies Come From?

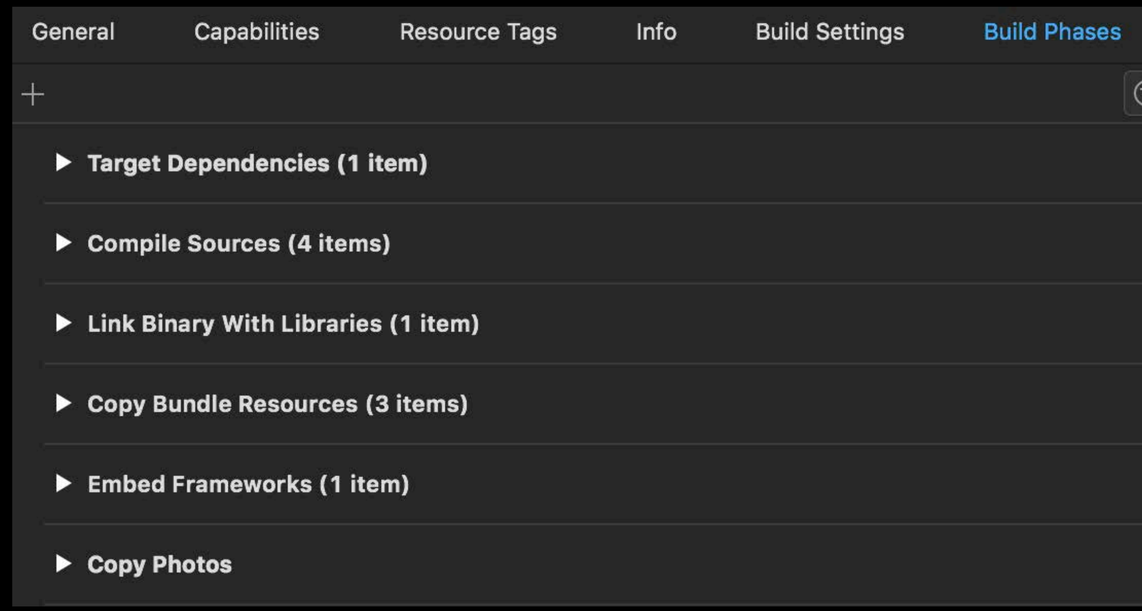
Built in

Target dependencies

Implicit dependencies

Build phase dependencies

Scheme order dependencies



Where Do Dependencies Come From?

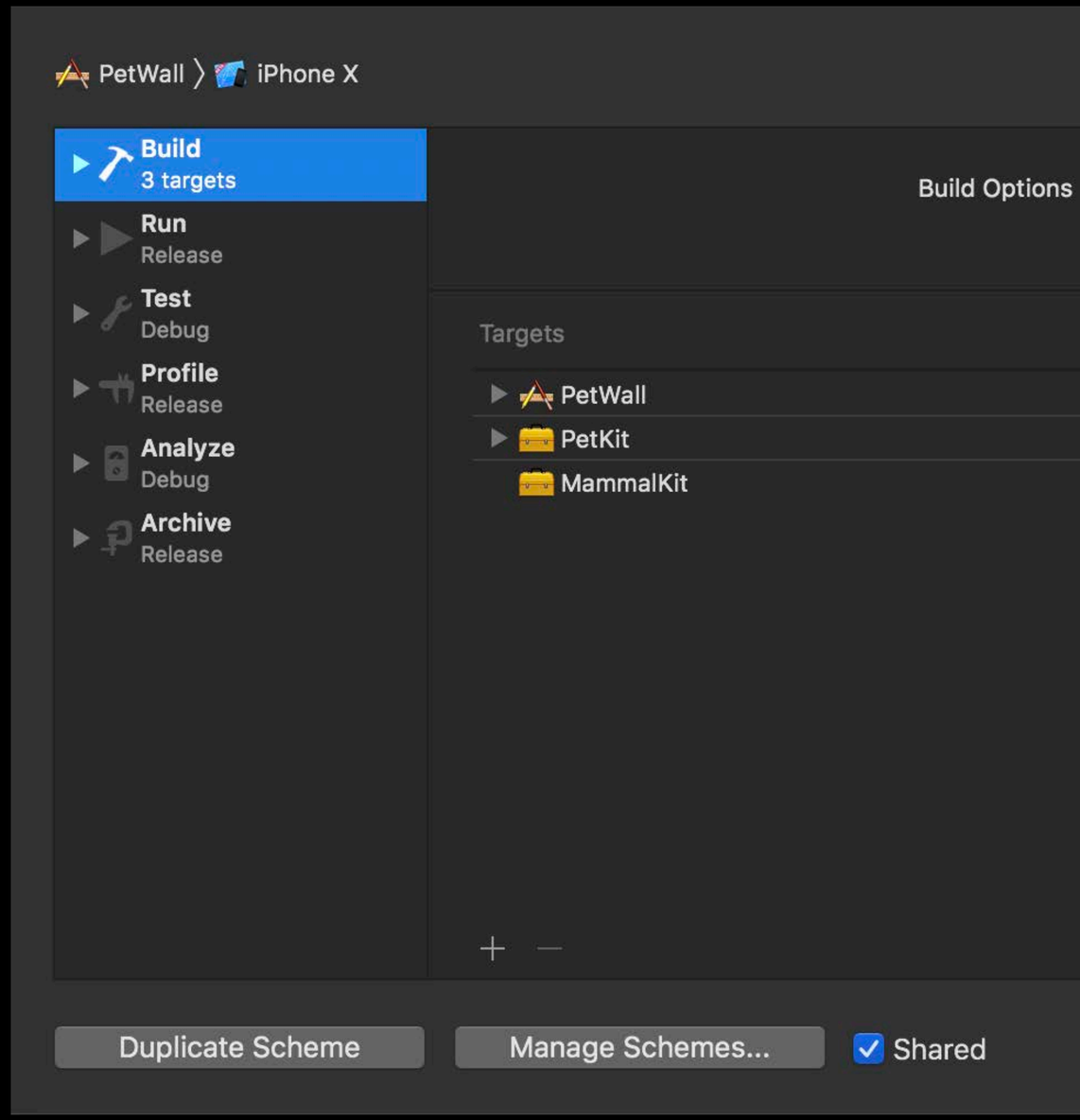
Built in

Target dependencies

Implicit dependencies

Build phase dependencies

Scheme order dependencies



Where Do Dependencies Come From?

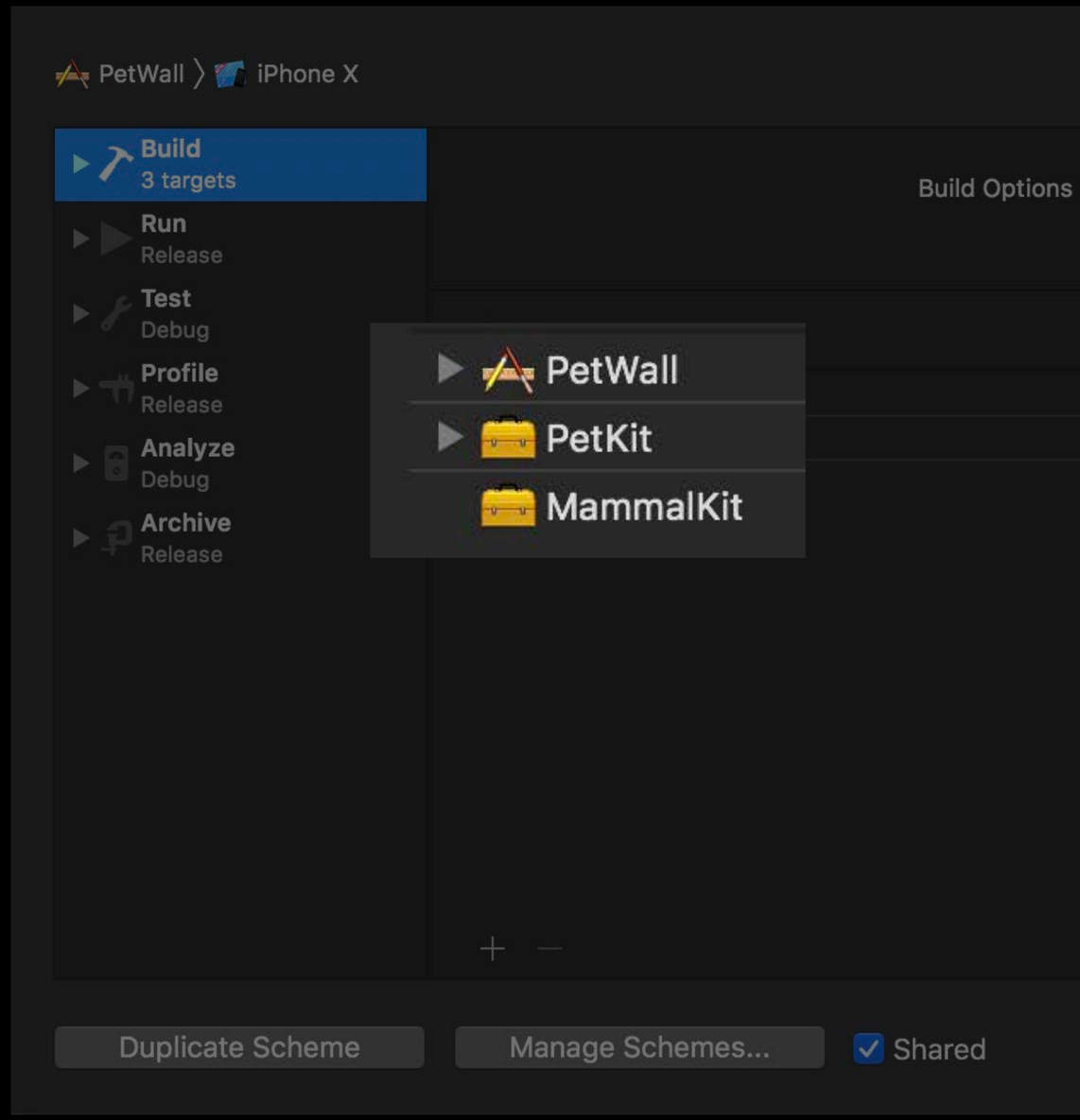
Built in

Target dependencies

Implicit dependencies

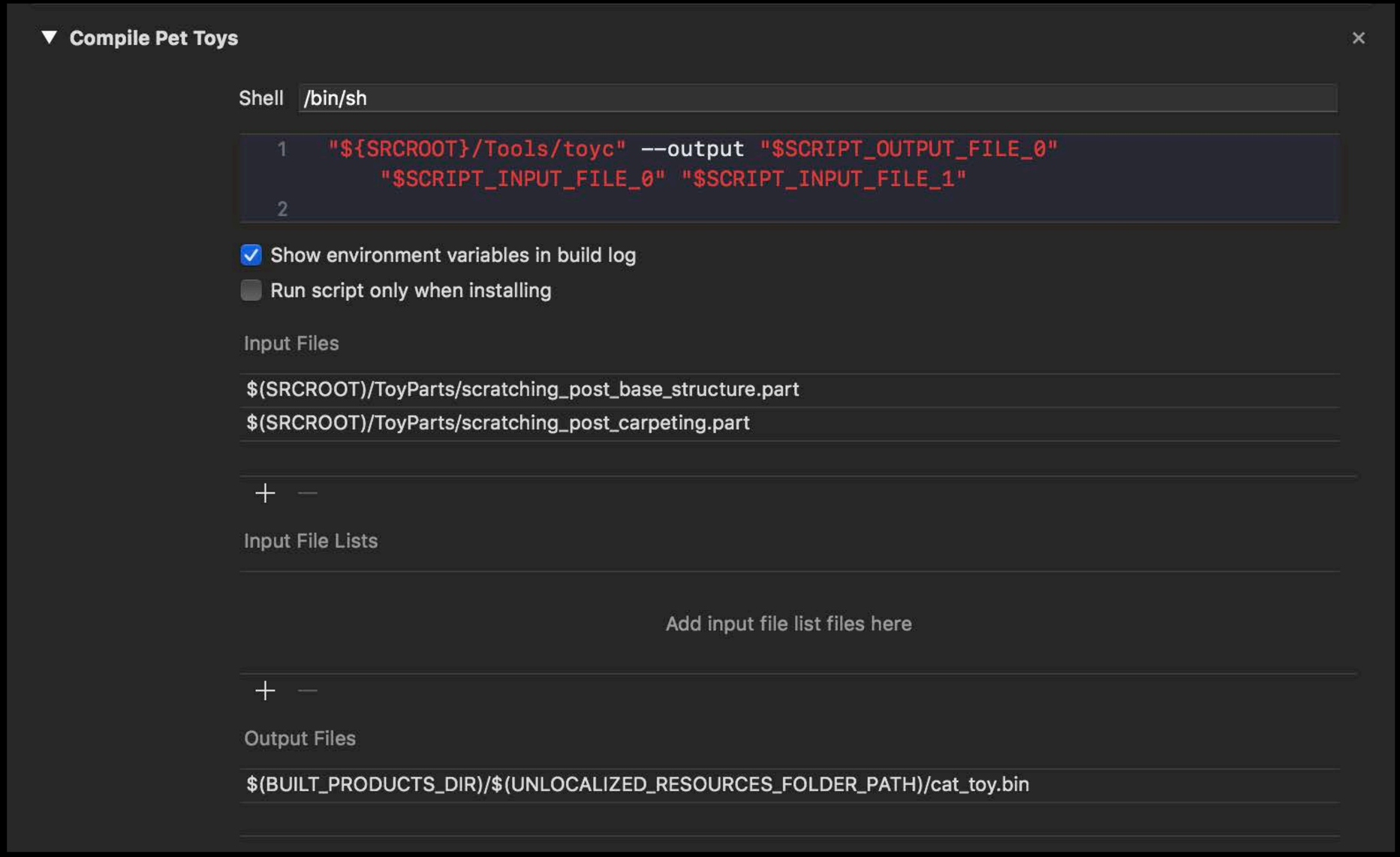
Build phase dependencies

Scheme order dependencies

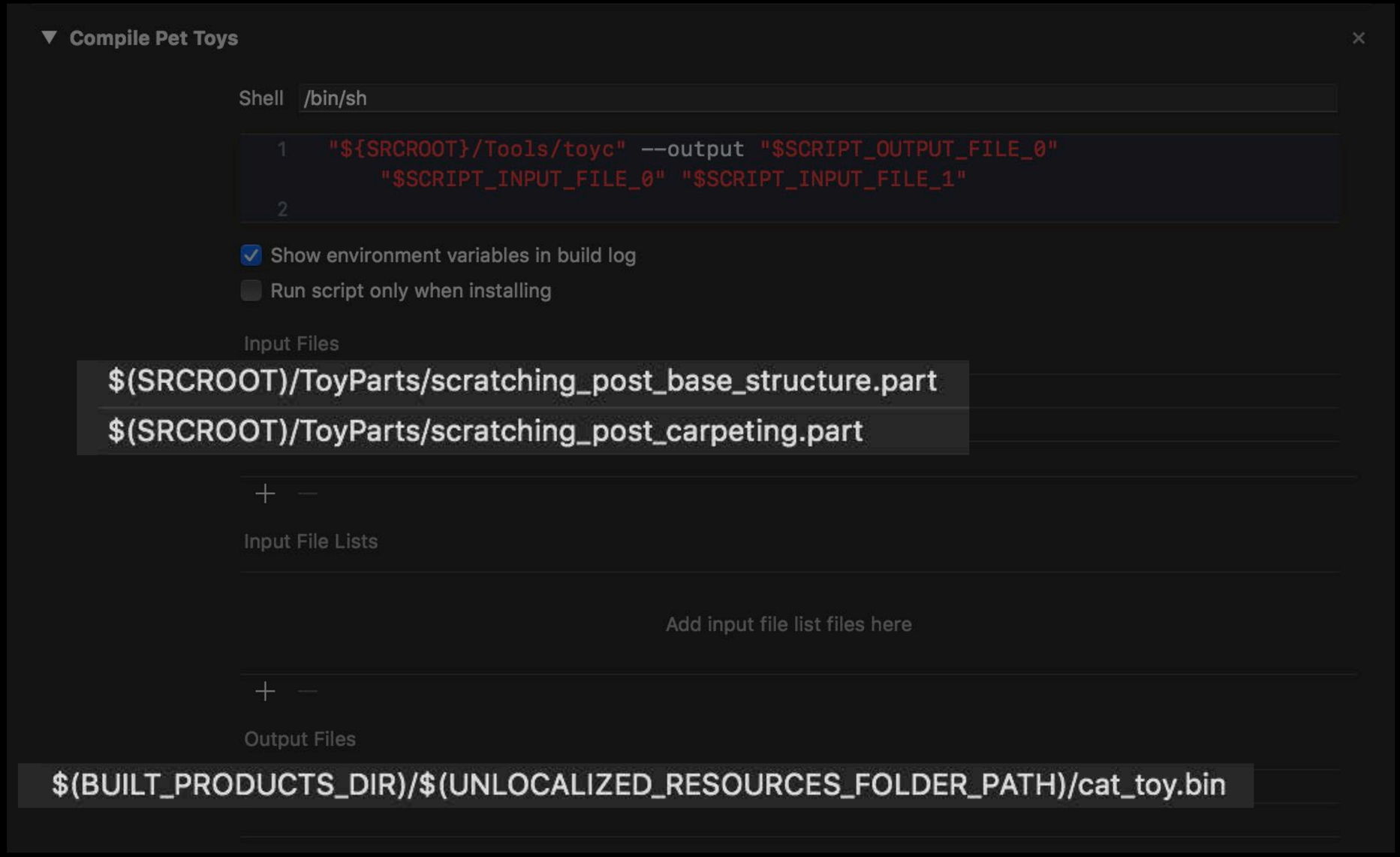


You!

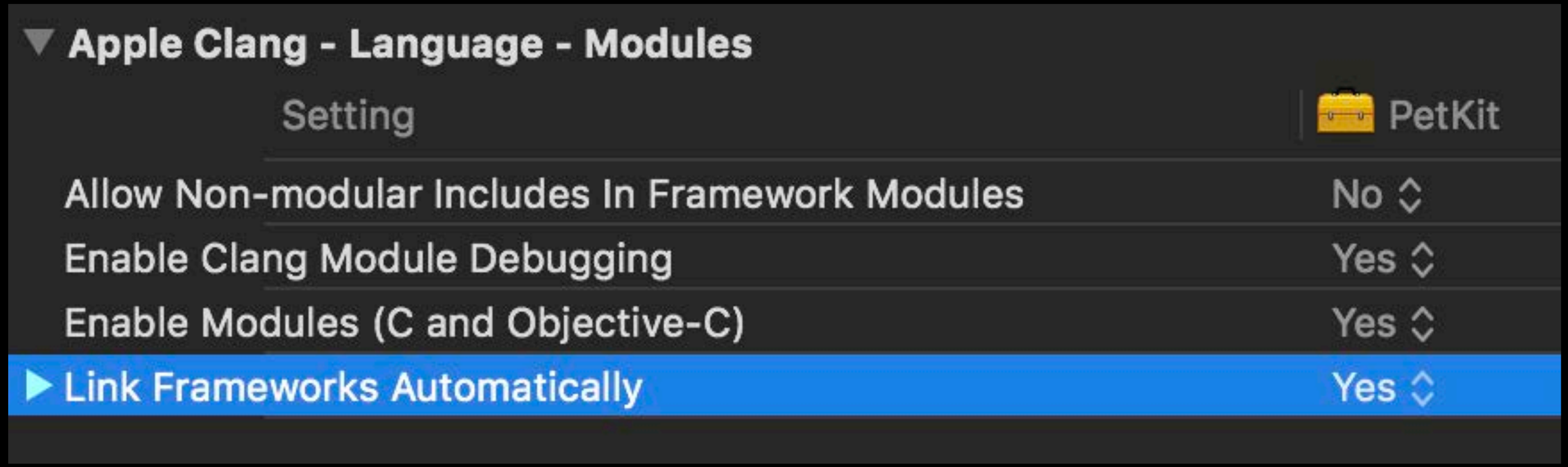
Declare Inputs and Outputs!



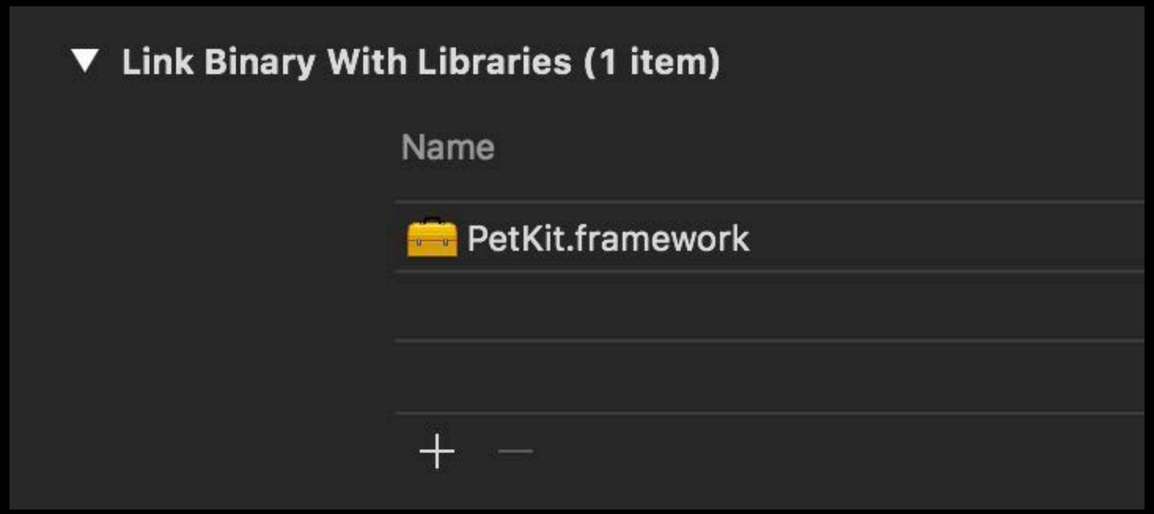
Declare Inputs and Outputs!



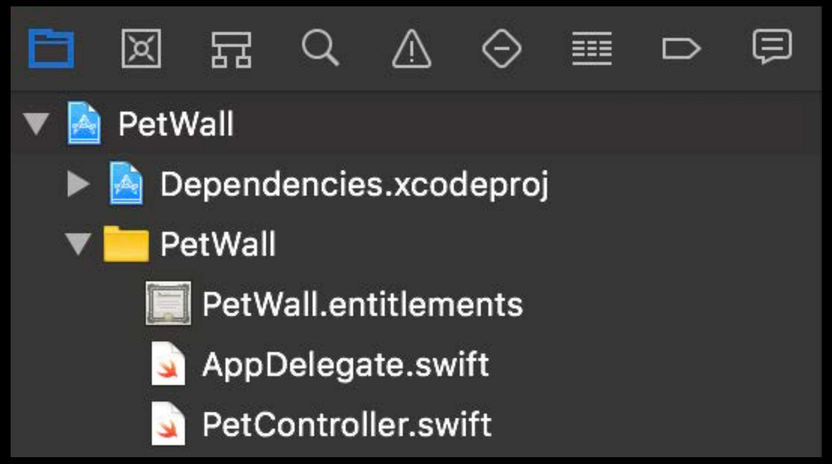
Avoid Auto-Link for Project Dependencies

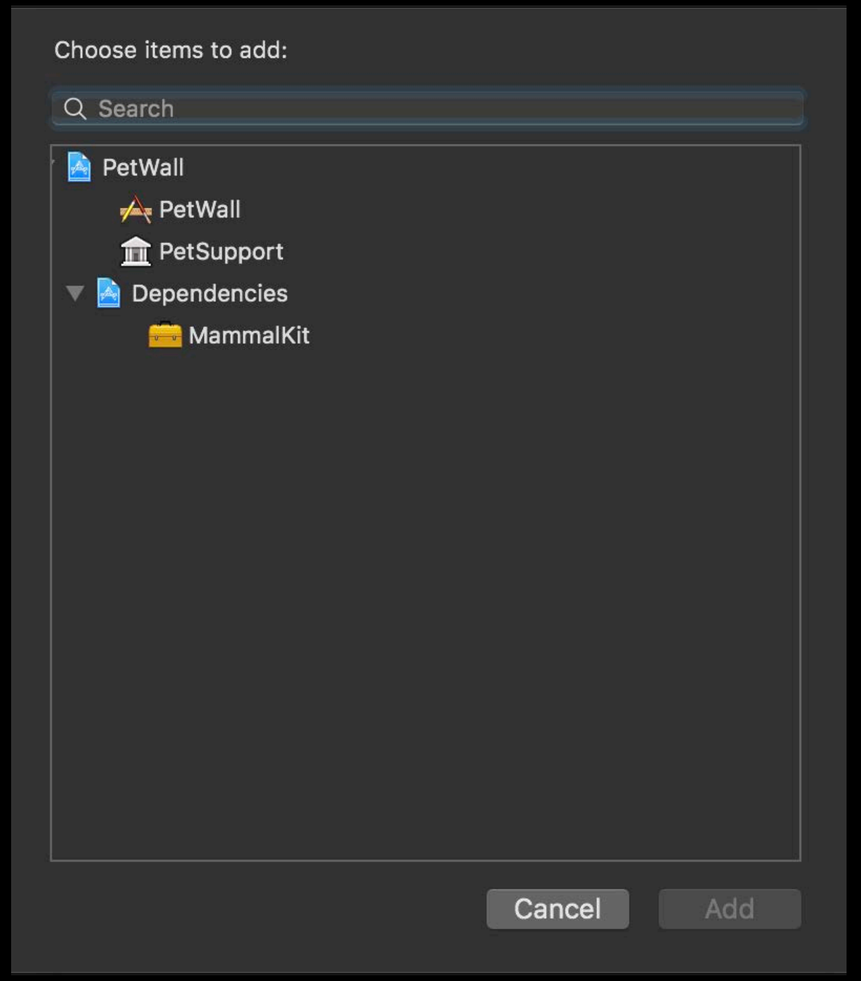


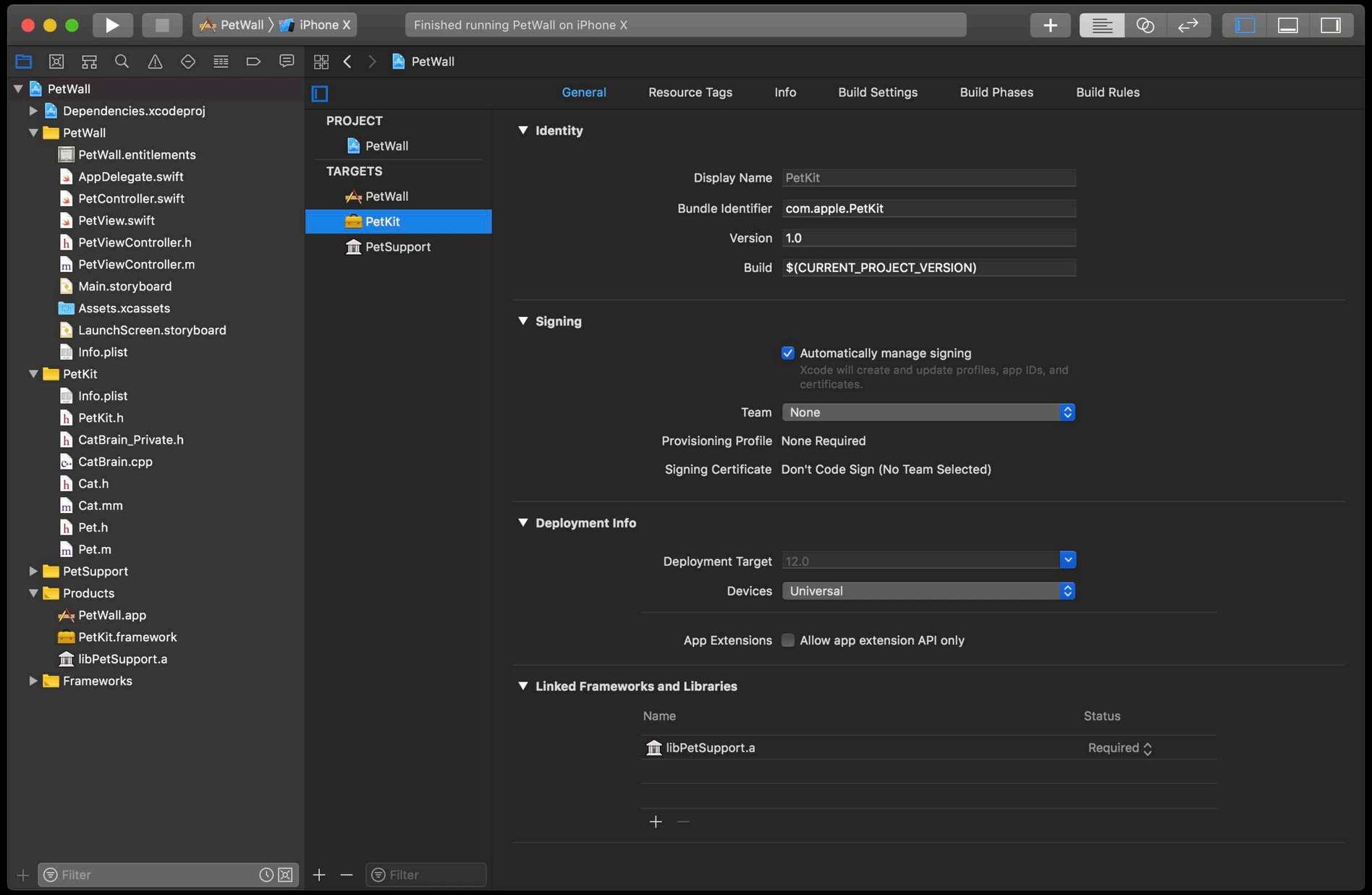
Add Explicit Dependencies

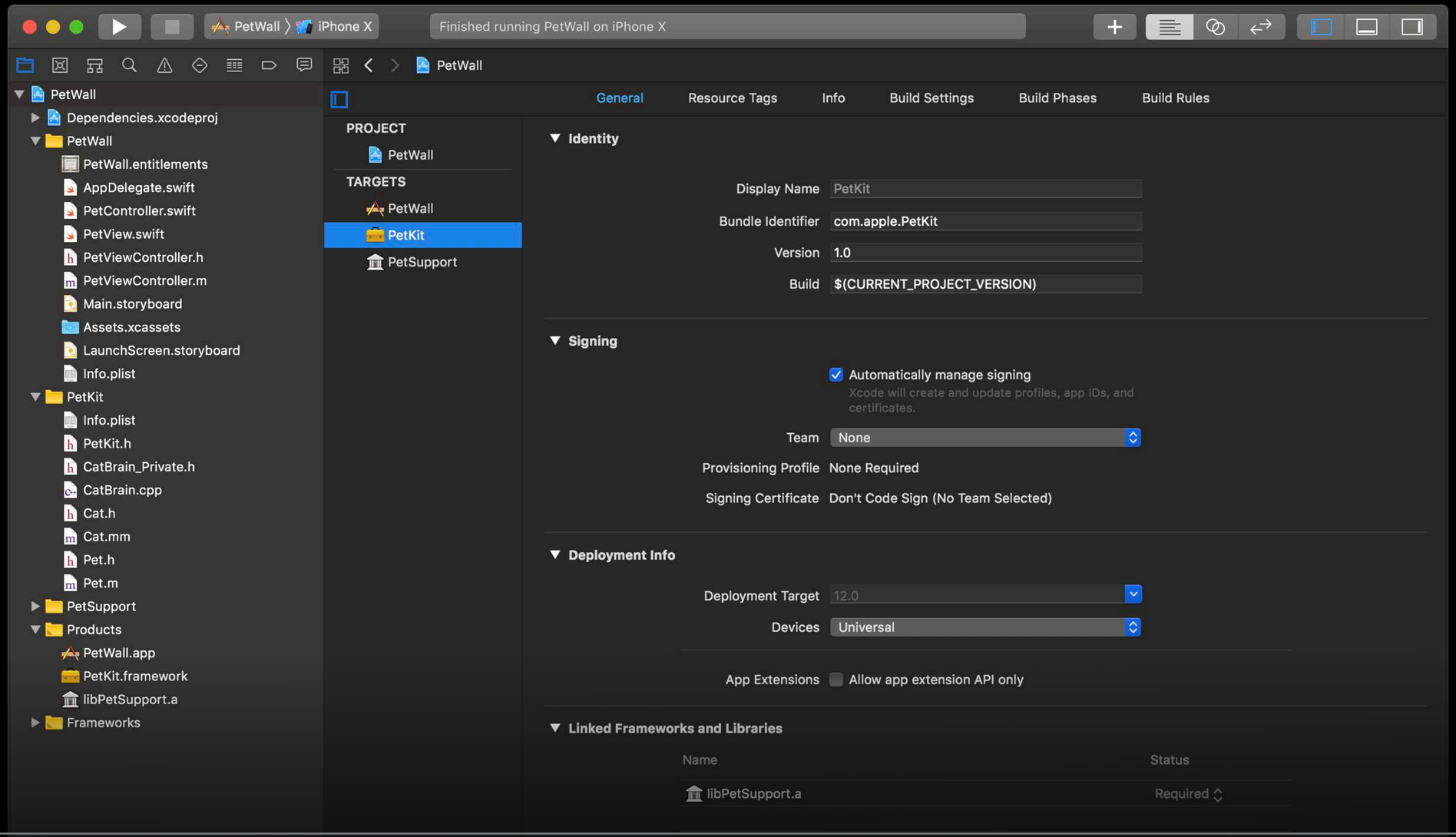


Create Workspaces and Project References






PetWall 〉 iPhone X
Finished running PetWall on iPhone X
PetWall
PetWall
Dependencies.xcodeproj
PetWall
PetWall.entitlements
AppDelegate.swift
PetController.swift
PetView.swift
PetViewController.h
PetViewController.m
Main.storyboard
Assets.xcassets
LaunchScreen.storyboard
Info.plist
PetKit
Info.plist
PetKit.h
CatBrain_Private.h
CatBrain.cpp
Cat.h
Cat.mm
Pet.h
Pet.m
PetSupport
Products
PetWall.app
PetKit.framework
libPetSupport.a
Frameworks
Filter
PetWall
General
Resource Tags
Info
Build Settings
Build Phases
Build Rules
PROJECT
PetWall
TARGETS
PetWall
PetKit
PetSupport
Filter
Identity
Display Name
PetKit
Bundle Identifier
com.apple.PetKit
Version
1.0
Build
$(CURRENT_PROJECT_VERSION)
Signing
Automatically manage signing
Xcode will create and update profiles, app IDs, and certificates.
Team
None
Provisioning Profile
None Required
Signing Certificate
Don't Code Sign (No Team Selected)
Deployment Info
Deployment Target
12.0
Devices
Universal
App Extensions
Allow app extension API only
Linked Frameworks and Libraries
Name
Status
libPetSupport.a
Required



Building Faster in Xcode

WWDC 2018

Behind the Scenes of the Xcode Build Process: Clang Builds

Jürgen Ributzka, Clang Frontend Team


10
1
0
001
100
01
100

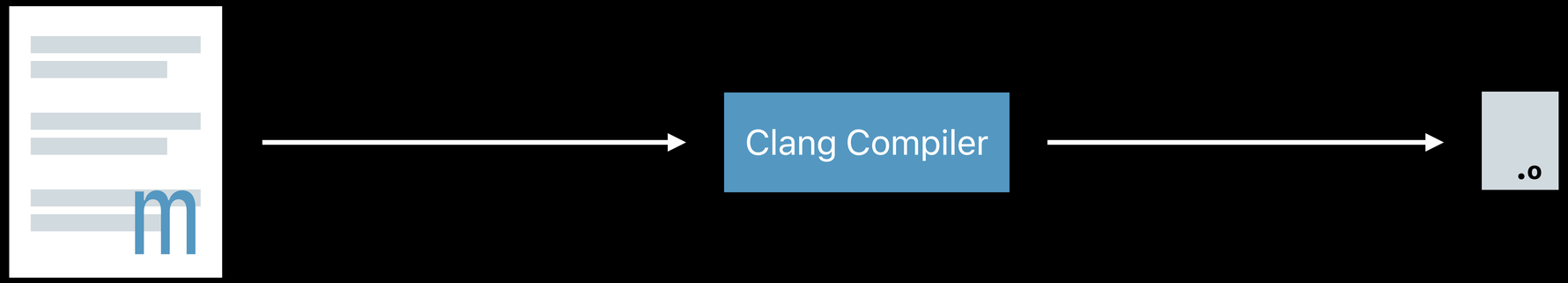
What Is Clang?

What Is Clang?

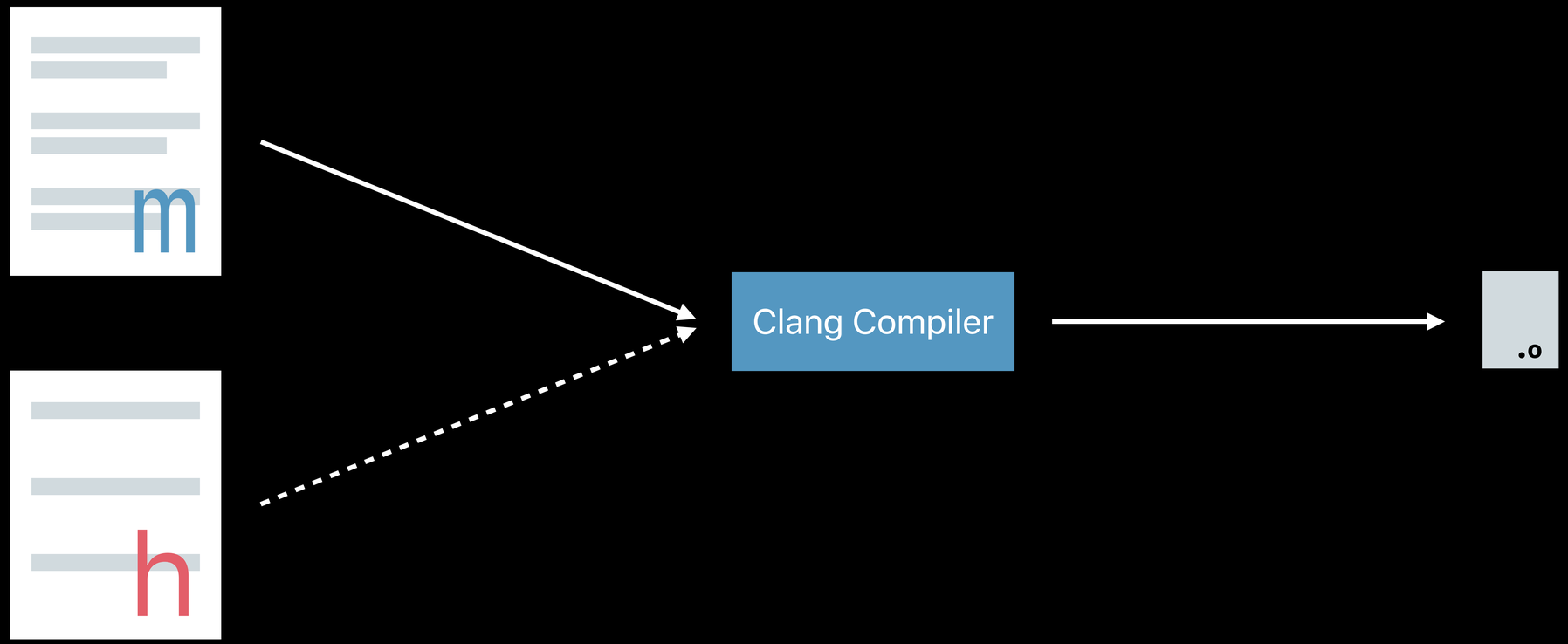
Apple’s official compiler for the C language family

- C
- C++
- Objective-C
- Objective-C++

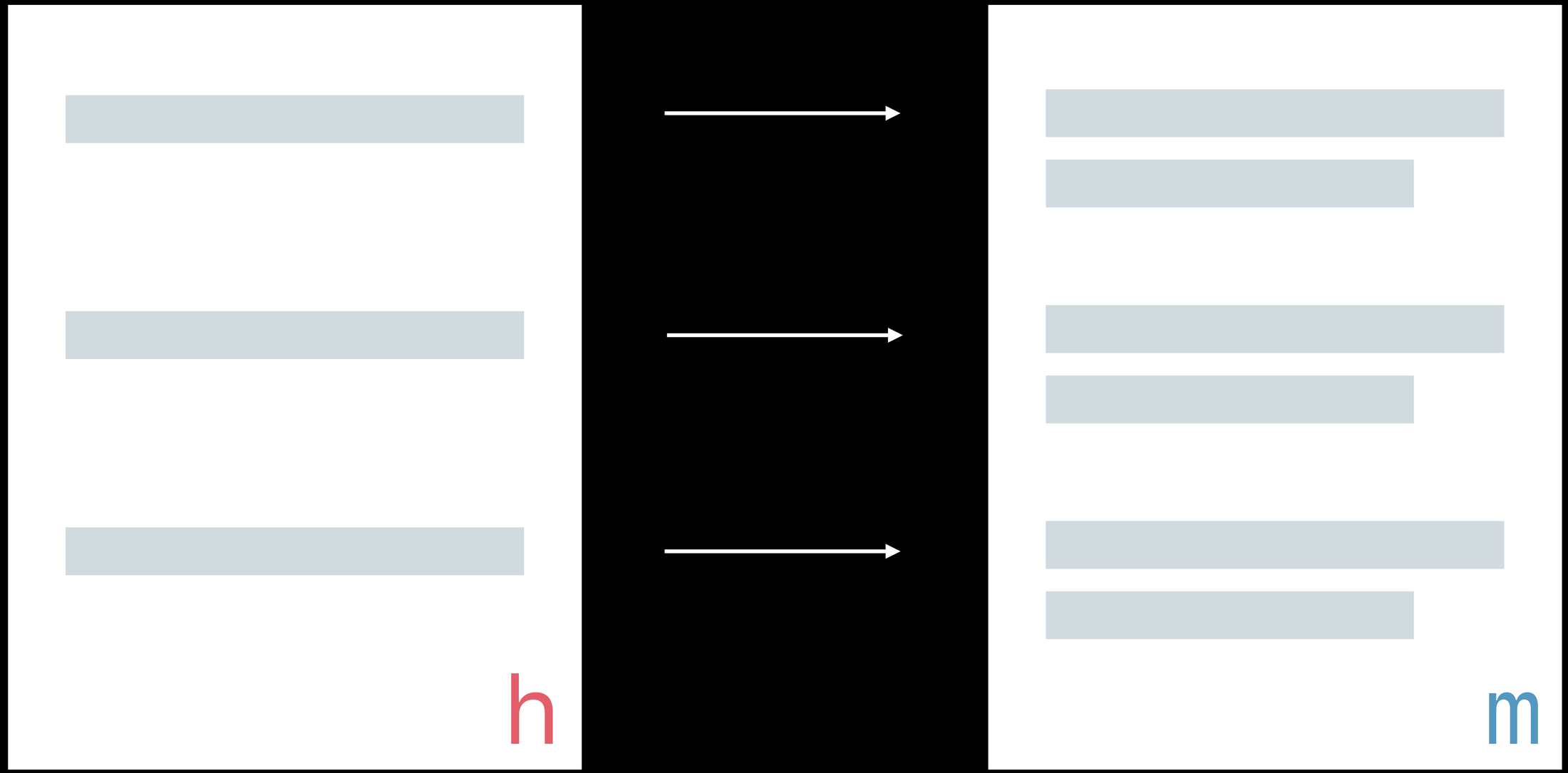
Traditional C Language Compilation



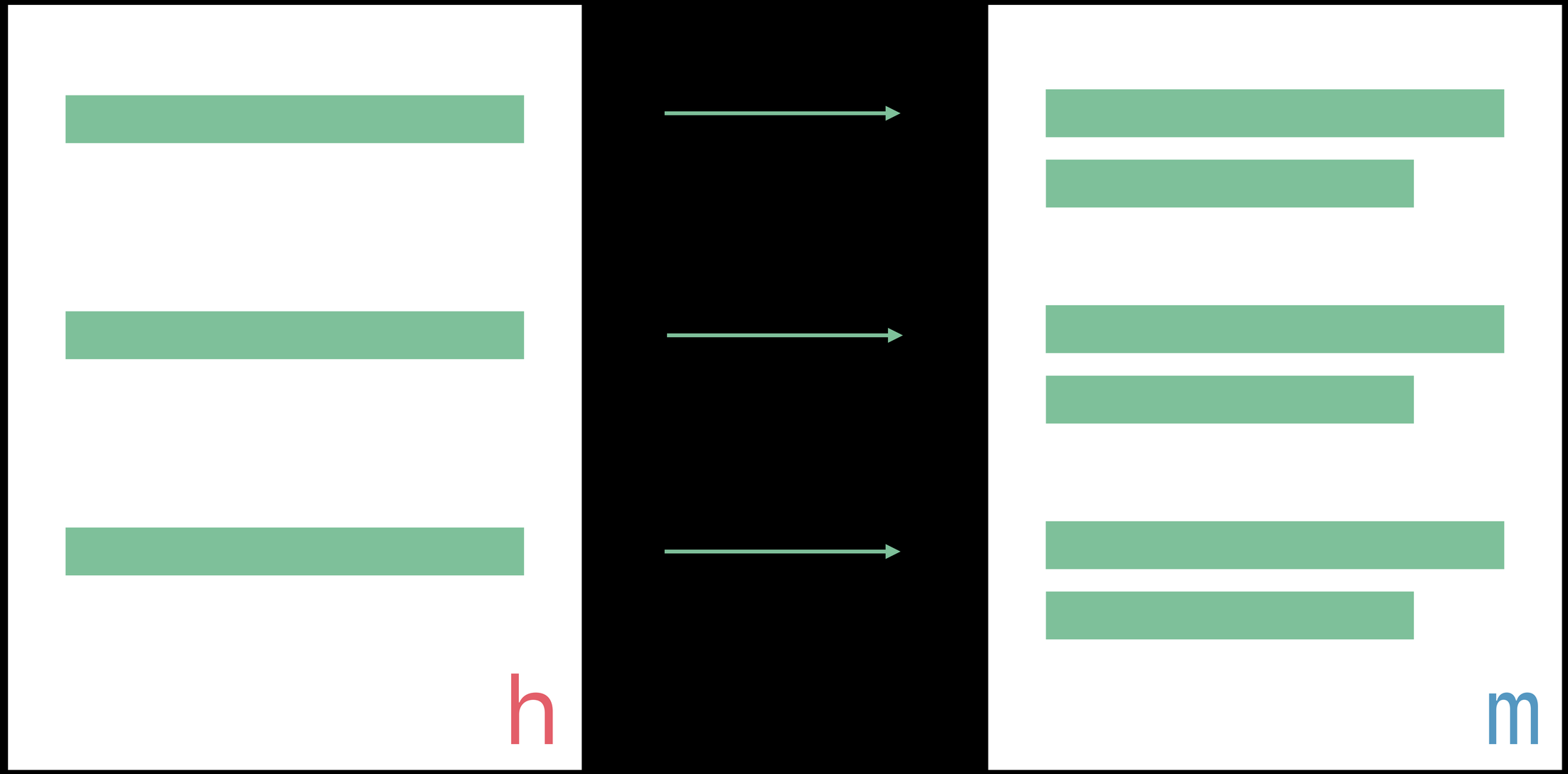
Traditional C Language Compilation



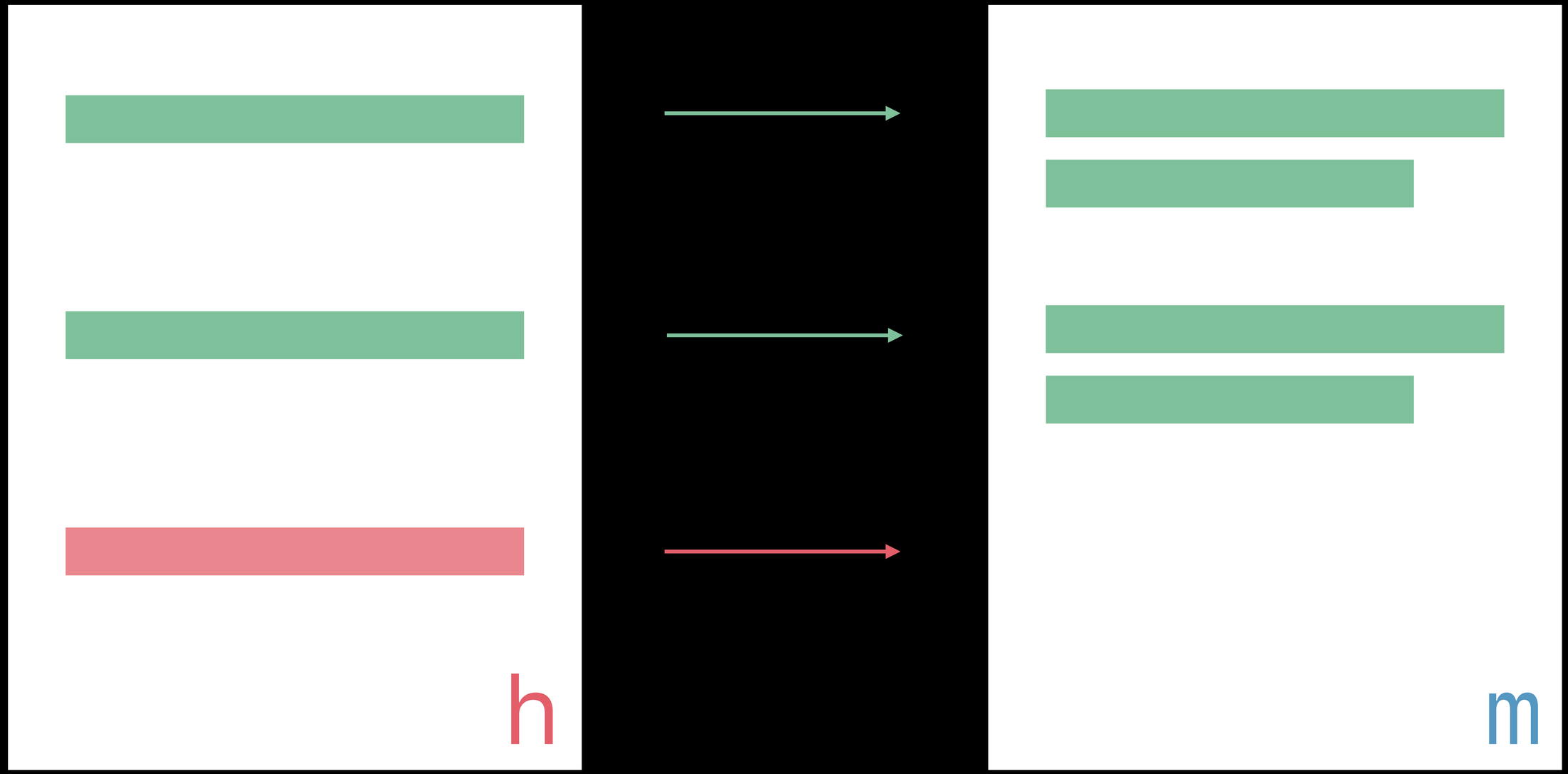
Traditional C Language Compilation



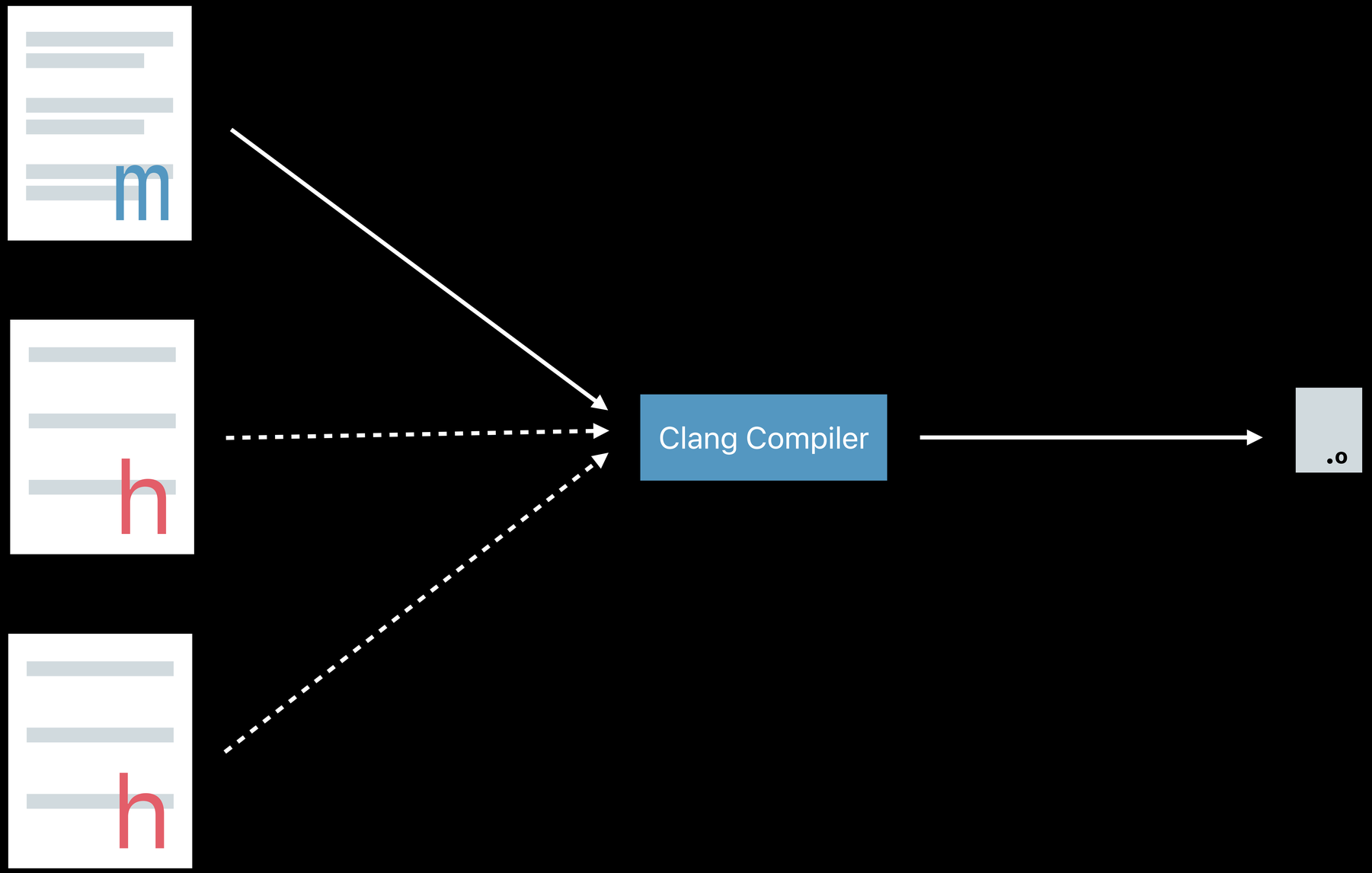
Traditional C Language Compilation



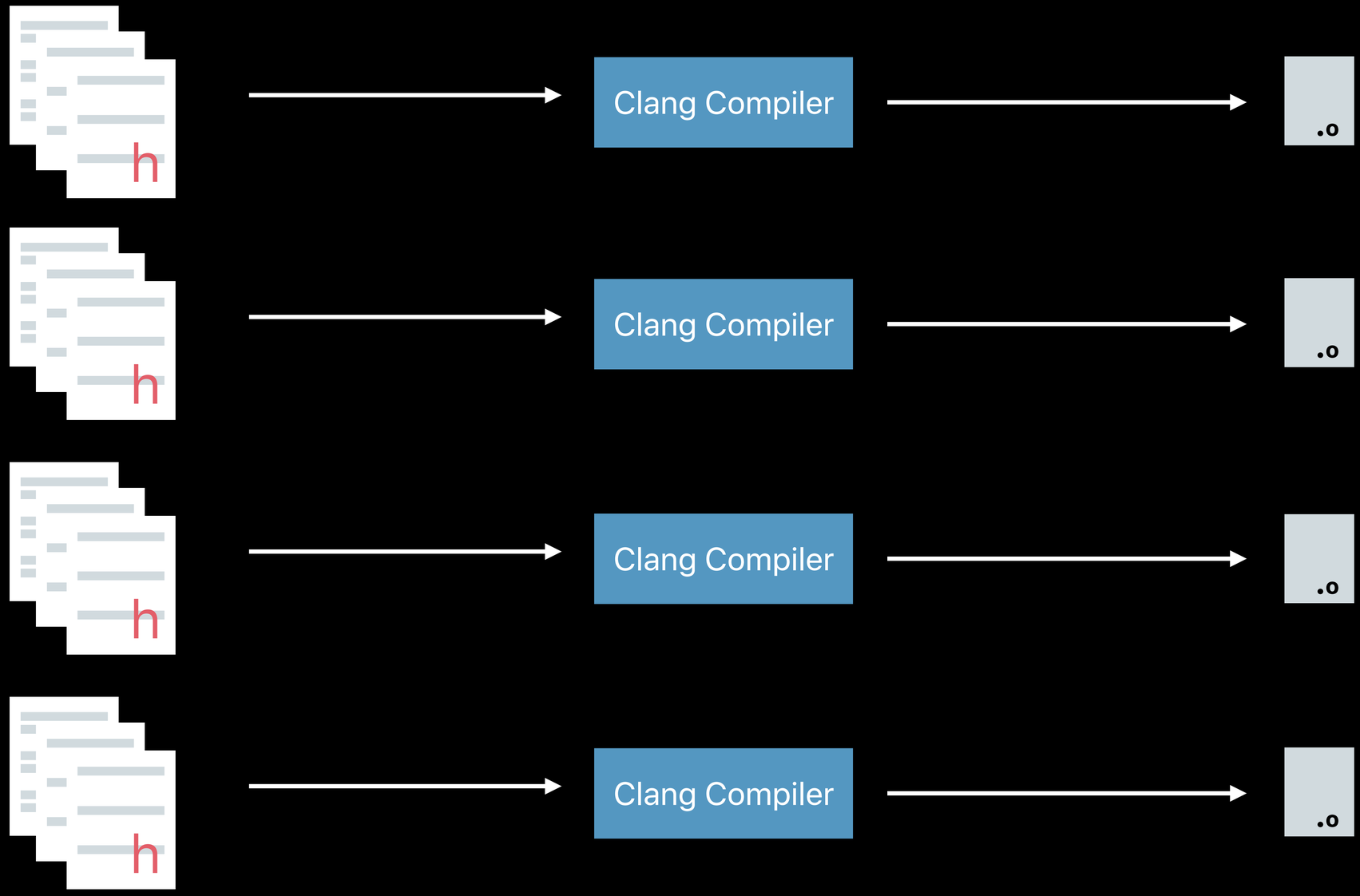
Traditional C Language Compilation

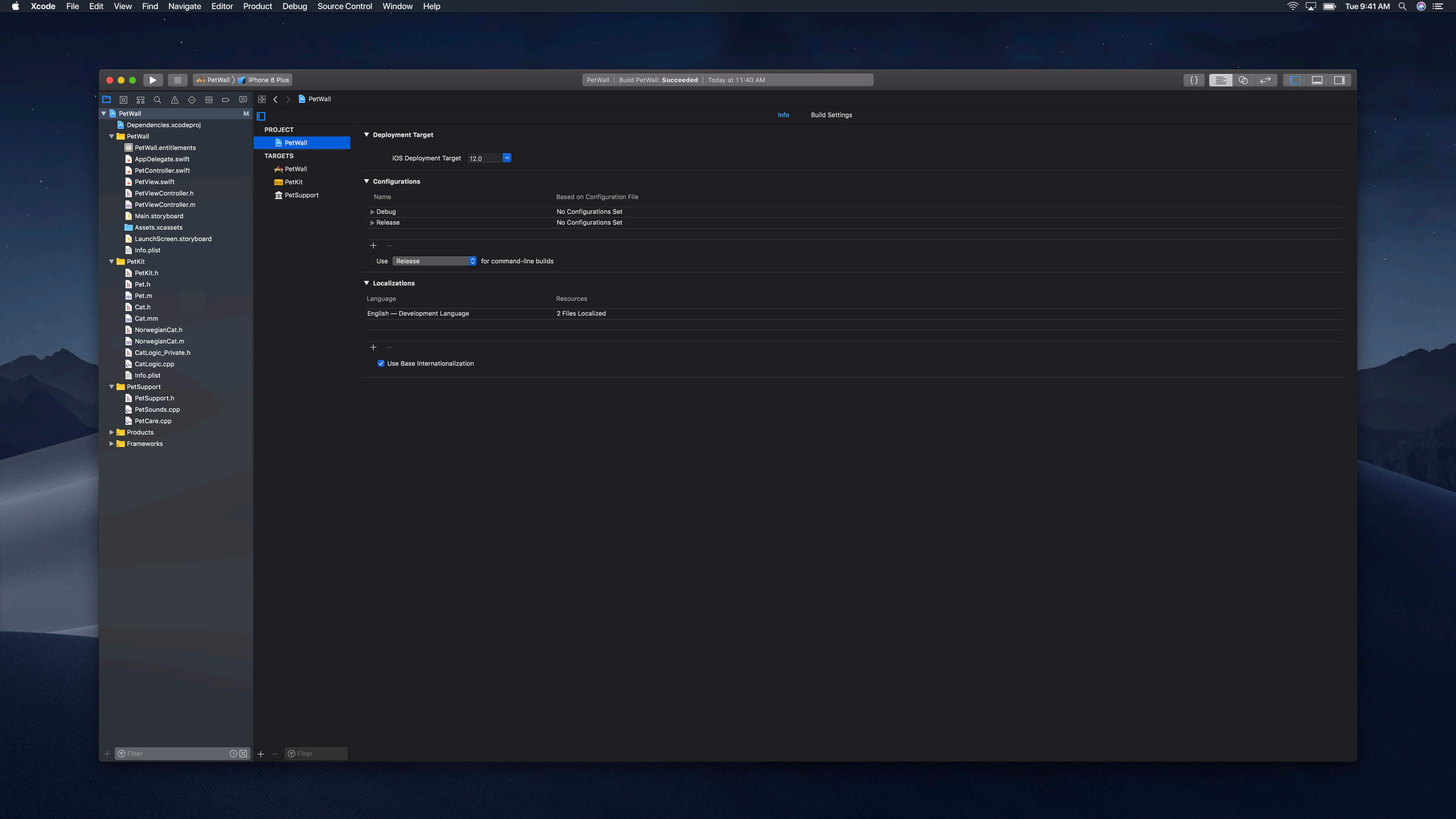


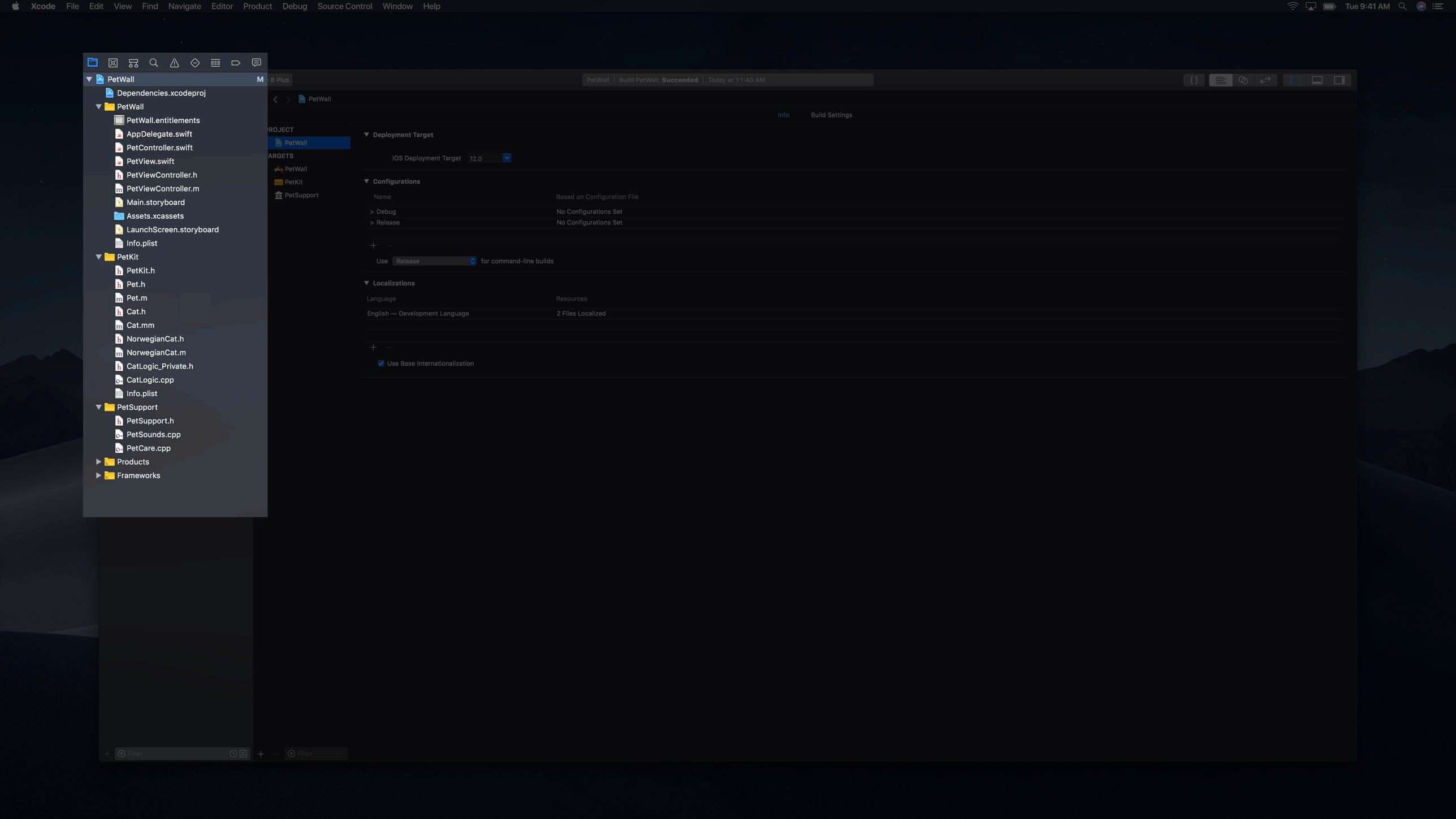
Traditional C Language Compilation



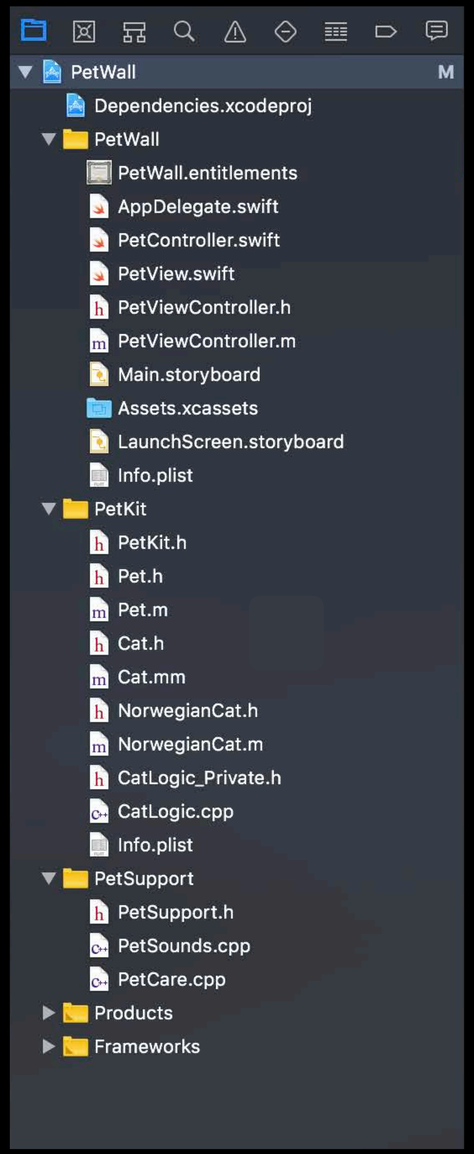
Traditional C Language Compilation



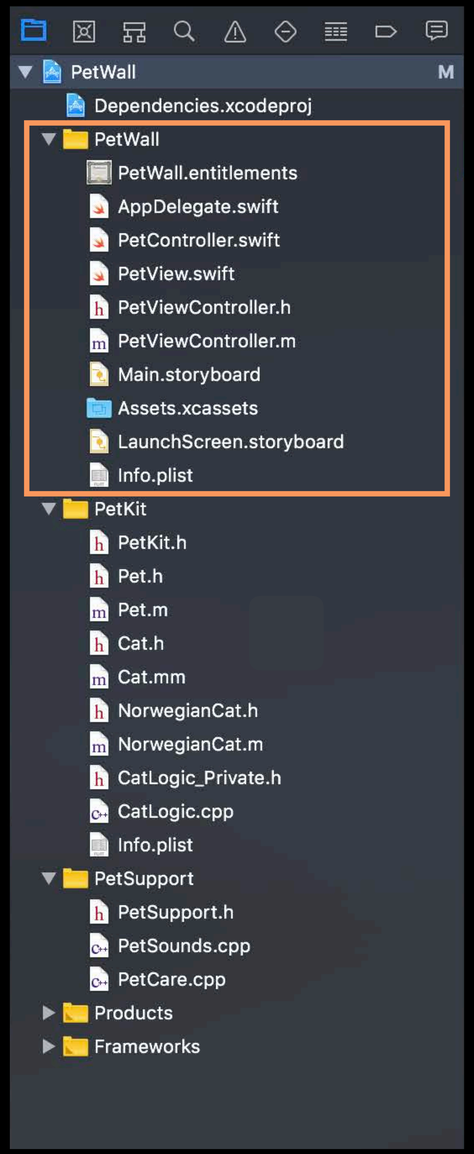

Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help
Tue 9:41 AM
PetWall 〉 iPhone 8 Plus
PetWall | Build PetWall: Succeeded | Today at 11:40 AM
PetWall
PetWall M
Dependencies.xcodeproj
PetWall
PetWall.entitlements
AppDelegate.swift
PetController.swift
PetView.swift
PetViewController.h
PetViewController.m
Main.storyboard
Assets.xcassets
LaunchScreen.storyboard
Info.plist
PetKit
PetKit.h
Pet.h
Pet.m
Cat.h
Cat.mm
NorwegianCat.h
NorwegianCat.m
CatLogic_Private.h
CatLogic.cpp
Info.plist
PetSupport
PetSupport.h
PetSounds.cpp
PetCare.cpp
Products
Frameworks
Filter
PROJECT
PetWall
TARGETS
PetWall
PetKit
PetSupport
Info
Build Settings
Deployment Target
iOS Deployment Target 12.0
Configurations
Name
Based on Configuration File
Debug
No Configurations Set
Release
No Configurations Set
Use Release for command-line builds
Localizations
Language
Resources
English — Development Language
2 Files Localized
Use Base Internationalization


Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help
Tue 9:41 AM
PetWall M
Dependencies.xcodeproj
PetWall
PetWall.entitlements
AppDelegate.swift
PetController.swift
PetView.swift
PetViewController.h
PetViewController.m
Main.storyboard
Assets.xcassets
LaunchScreen.storyboard
Info.plist
PetKit
PetKit.h
Pet.h
Pet.m
Cat.h
Cat.mm
NorwegianCat.h
NorwegianCat.m
CatLogic_Private.h
CatLogic.cpp
Info.plist
PetSupport
PetSupport.h
PetSounds.cpp
PetCare.cpp
Products
Frameworks
8 Plus
PetWall | Build PetWall: Succeeded | Today at 11:40 AM
PetWall
Info
Build Settings
PROJECT
PetWall
TARGETS
PetWall
PetKit
PetSupport
Deployment Target
iOS Deployment Target 12.0
Configurations
Name
Based on Configuration File
Debug
No Configurations Set
Release
No Configurations Set
Use Release for command-line builds
Localizations
Language
Resources
English — Development Language
2 Files Localized
Use Base Internationalization
Filter
Filter

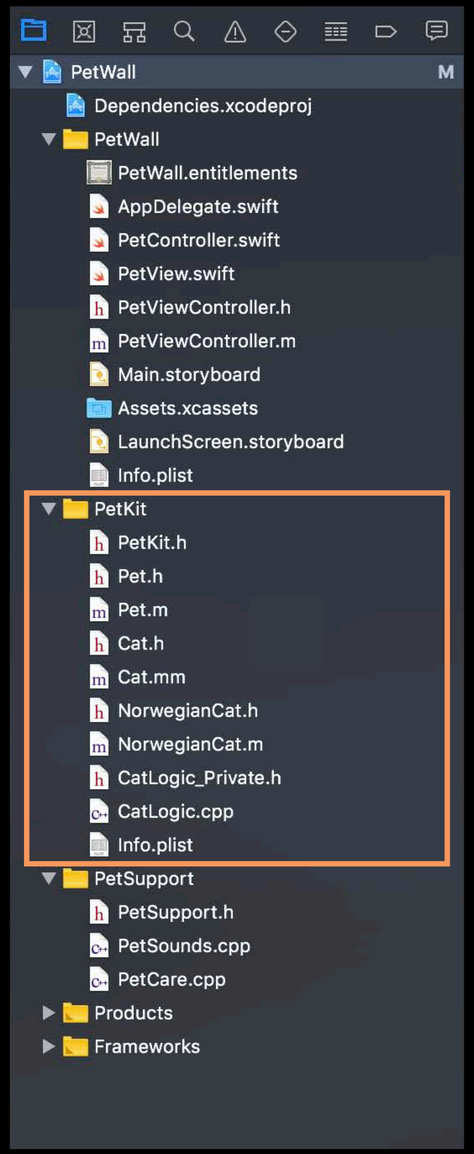
PetWall Example



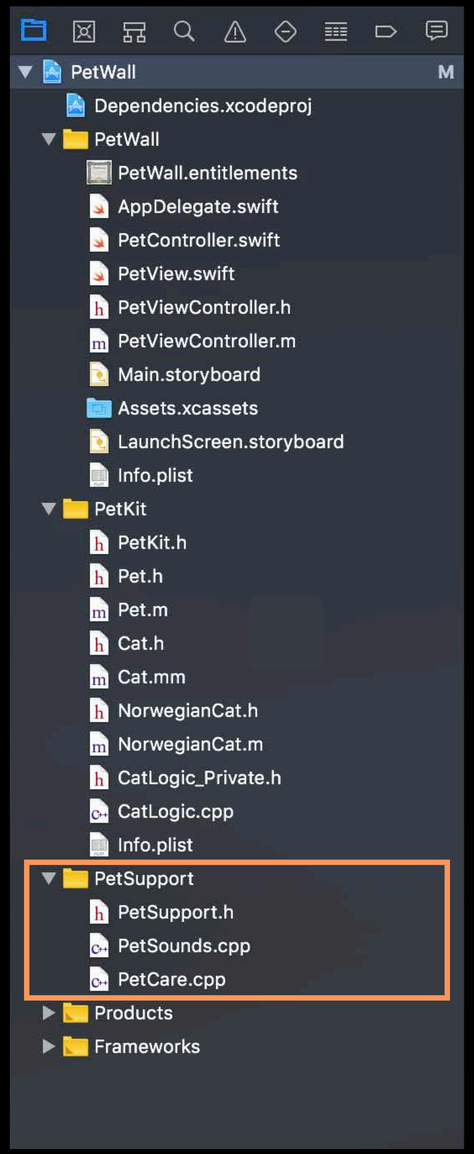
PetWall Example



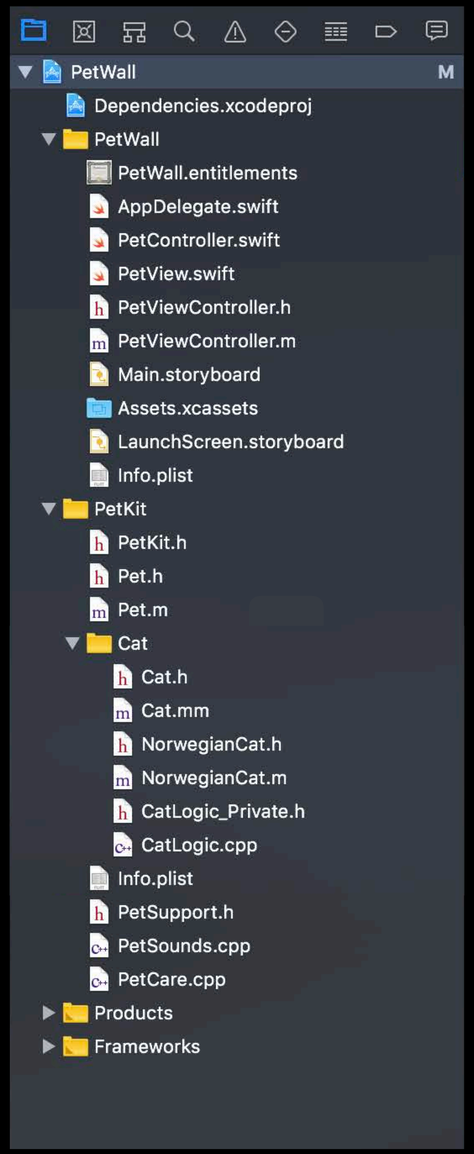
PetWall Example



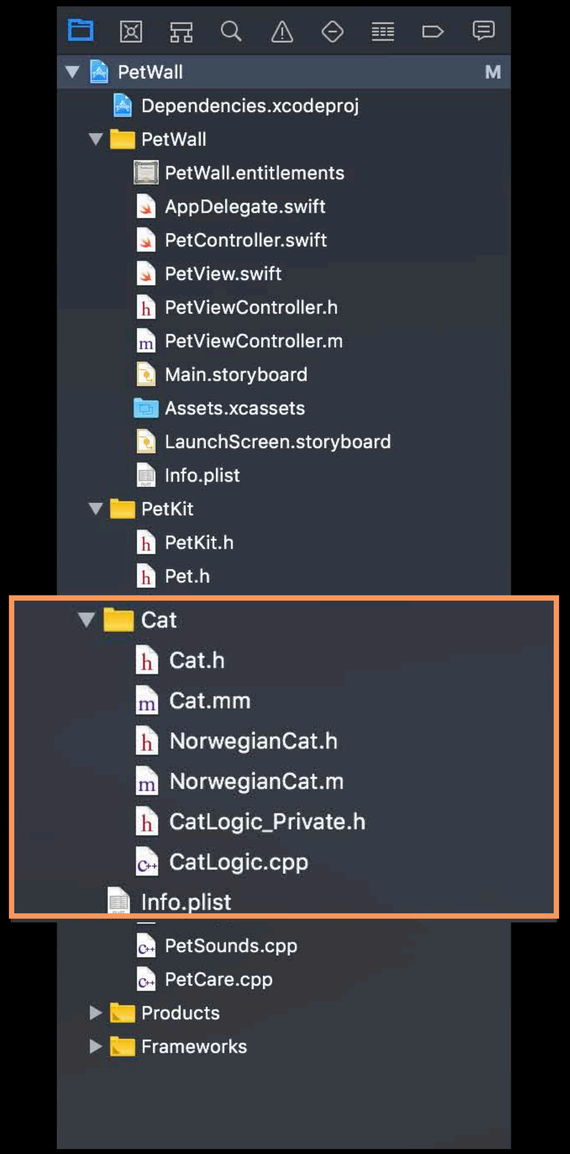
PetWall Example



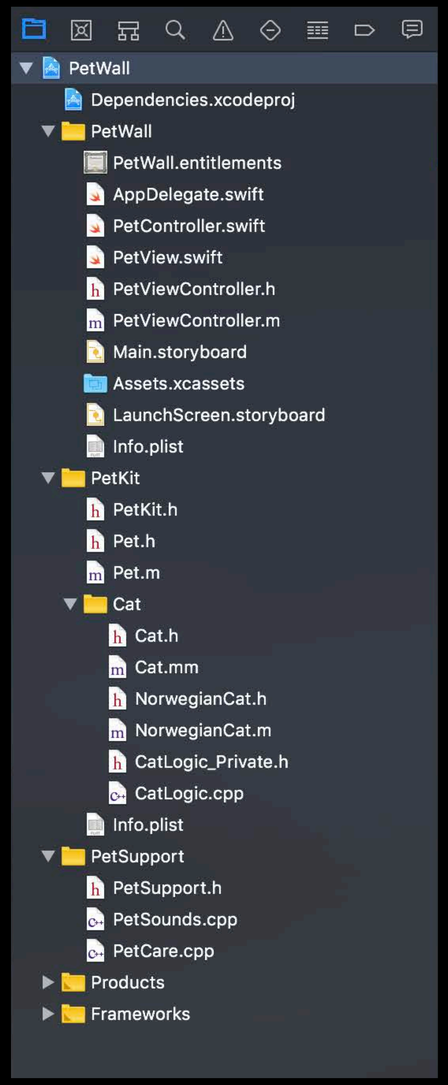
PetWall Example



PetWall Example



PetWall Example




```
//
//  Cat.mm
//  PetKit
//
#import <PetKit/Cat.h>
#import "PetSupport.h"
```

```
//
//  NorwegianCat.h
//  PetKit
//
#import <PetKit/Cat.h>
```

How do we find your headers?

How Do We Find Our Cat?

```
//
//  Cat.mm
//  PetKit
//
#import <PetKit/Cat.h>

@implementation Cat

- (NSInteger)_sizeOfCat {
    return 10;
}
- (void)meow {
    theCatsMeow((int)self._sizeOfCat);
```

How Do We Find Our Cat?

Write /x/src/PetWall/DerivedData/PetWall/Build/Intermediates.noindex/PetWall.build/Release-iphonesimulator/PetKit.build/PetKit-generated-files.hmap 0.1 seconds
Write /x/src/PetWall/DerivedData/PetWall/Build/Intermediates.noindex/PetWall.build/Release-iphonesimulator/PetKit.build/PetKit-all-non-framework-target-headers.hmap 0.1 seconds
Write /x/src/PetWall/DerivedData/PetWall/Build/Intermediates.noindex/PetWall.build/Release-iphonesimulator/PetKit.build/module.modulemap 0.1 seconds
Copy module.modulemap 0.1 seconds
Copy PetKit.h 0.1 seconds
Copy Pet.h 0.1 seconds
Copy NorwegianCat.h 0.1 seconds
Copy Cat.h 0.1 seconds
Copy CatLogic_Private.h 0.1 seconds
Write /x/src/PetWall/DerivedData/PetWall/Build/Intermediates.noindex/PetWall.build/Release-iphonesimulator/PetKit.build/DerivedSources/PetKit_vers.c 0.1 seconds
Process /x/src/PetWall/PetKit/Info.plist 0.1 seconds
Compile CatLogic.cpp 0.3 seconds
Write /x/src/PetWall/DerivedData/PetWall/Build/Intermediates.noindex/PetWall.build/Release-iphonesimulator/PetKit.build/Objects-normal/x86_64/PetKit.LinkFileList 0.1 seconds
Compile NorwegianCat.m 1.1 seconds
Compile Cat.mm 0.2 seconds

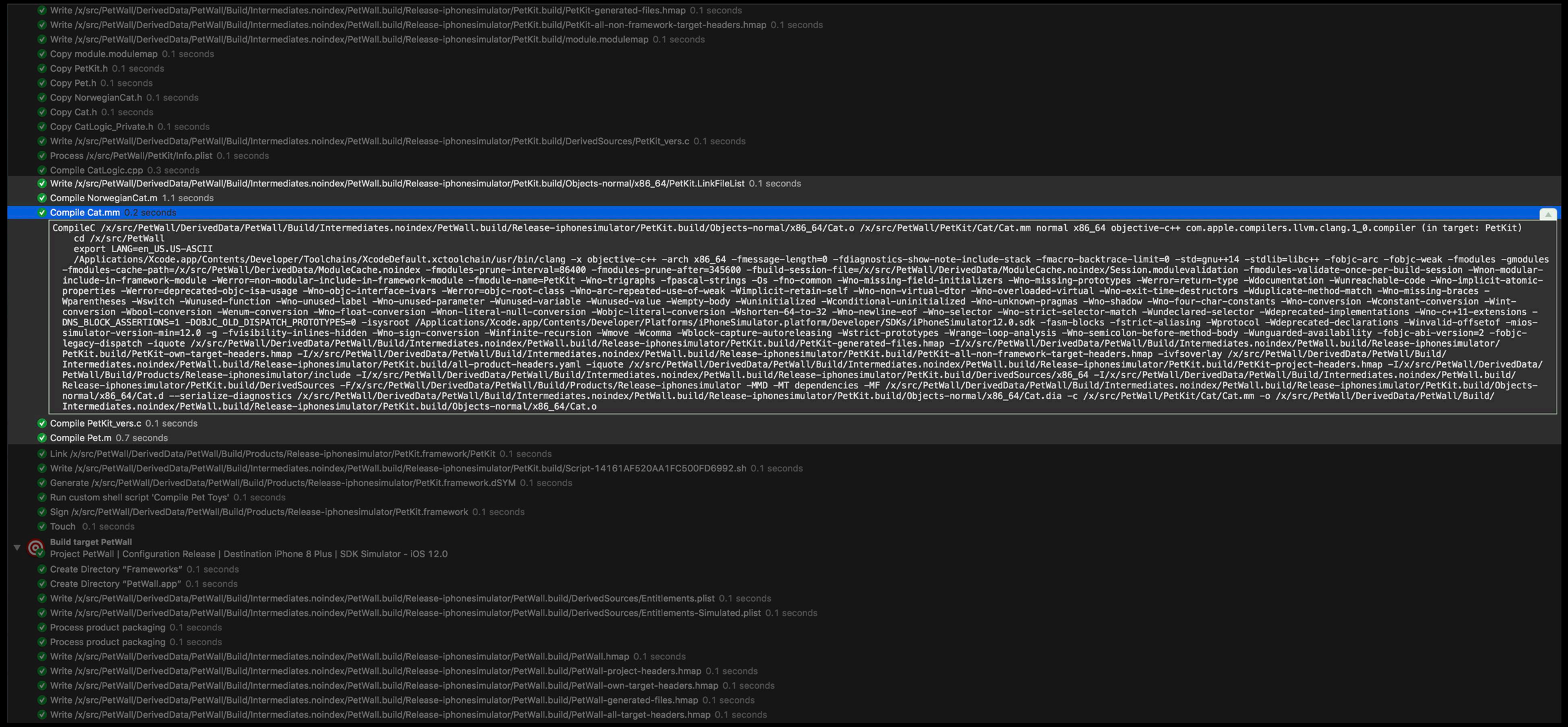
```
CompileC /x/src/PetWall/DerivedData/PetWall/Build/Intermediates.noindex/PetWall.build/Release-iphonesimulator/PetKit.build/Objects-normal/x86_64/Cat.o /x/src/PetWall/PetKit/Cat/Cat.mm normal x86_64 objective-c++ com.apple.compilers.llvm.clang.1_0.compiler (in target: PetKit)
    cd /x/src/PetWall
    export LANG=en_US.US-ASCII
    /Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/bin/clang -x objective-c++ -arch x86_64 -fmessage-length=0 -fdiagnostics-show-note-include-stack -fmacro-backtrace-limit=0 -std=gnu++14 -stdlib=libc++ -fobjc-arc -fobjc-weak -fmodules -gmodules
  -fmodules-cache-path=/x/src/PetWall/DerivedData/ModuleCache.noindex -fmodules-prune-interval=86400 -fmodules-prune-after=345600 -fbuild-session-file=/x/src/PetWall/DerivedData/ModuleCache.noindex/Session.modulevalidation -fmodules-validate-once-per-build-session -Wnon-modular-
  include-in-framework-module -Werror=non-modular-include-in-framework-module -fmodule-name=PetKit -Wno-trigraphs -fpascal-strings -Os -fno-common -Wno-missing-field-initializers -Wno-missing-prototypes -Werror=return-type -Wdocumentation -Wunreachable-code -Wno-implicit-atomic-
  properties -Werror=deprecated-objc-isa-usage -Wno-objc-interface-ivars -Werror=objc-root-class -Wno-arc-repeated-use-of-weak -Wimplicit-retain-self -Wno-non-virtual-dtor -Wno-overloaded-virtual -Wno-exit-time-destructors -Wduplicate-method-match -Wno-missing-braces -
  Wparentheses -Wswitch -Wunused-function -Wno-unused-label -Wno-unused-parameter -Wunused-variable -Wunused-value -Wempty-body -Wuninitialized -Wconditional-uninitialized -Wno-unknown-pragmas -Wno-shadow -Wno-four-char-constants -Wno-conversion -Wconstant-conversion -Wint-
  conversion -Wbool-conversion -Wenum-conversion -Wno-float-conversion -Wnon-literal-null-conversion -Wobjc-literal-conversion -Wshorten-64-to-32 -Wno-newline-eof -Wno-selector -Wno-strict-selector-match -Wundeclared-selector -Wdeprecated-implementations -Wno-c++11-extensions -
  DNS_BLOCK_ASSERTIONS=1 -DOBJC_OLD_DISPATCH_PROTOTYPES=0 -isysroot /Applications/Xcode.app/Contents/Developer/Platforms/iPhoneSimulator.platform/Developer/SDKs/iPhoneSimulator12.0.sdk -fasm-blocks -fstrict-aliasing -Wprotocol -Wdeprecated-declarations -Winvalid-offsetof -mios-
  simulator-version-min=12.0 -g -fvisibility-inlines-hidden -Wno-sign-conversion -Winfinite-recursion -Wmove -Wcomma -Wblock-capture-autoreleasing -Wstrict-prototypes -Wrange-loop-analysis -Wno-semicolon-before-method-body -Wunguarded-availability -fobjc-abi-version=2 -fobjc-
  legacy-dispatch -iquote /x/src/PetWall/DerivedData/PetWall/Build/Intermediates.noindex/PetWall.build/Release-iphonesimulator/PetKit.build/PetKit-generated-files.hmap -I/x/src/PetWall/DerivedData/PetWall/Build/Intermediates.noindex/PetWall.build/Release-iphonesimulator/
  PetKit.build/PetKit-own-target-headers.hmap -I/x/src/PetWall/DerivedData/PetWall/Build/Intermediates.noindex/PetWall.build/Release-iphonesimulator/PetKit.build/PetKit-all-non-framework-target-headers.hmap -ivfsoverlay /x/src/PetWall/DerivedData/PetWall/Build/
  Intermediates.noindex/PetWall.build/Release-iphonesimulator/PetKit.build/all-product-headers.yaml -iquote /x/src/PetWall/DerivedData/PetWall/Build/Intermediates.noindex/PetWall.build/Release-iphonesimulator/PetKit.build/PetKit-project-headers.hmap -I/x/src/PetWall/DerivedData/
  PetWall/Build/Products/Release-iphonesimulator/include -I/x/src/PetWall/DerivedData/PetWall/Build/Intermediates.noindex/PetWall.build/Release-iphonesimulator/PetKit.build/DerivedSources/x86_64 -I/x/src/PetWall/DerivedData/PetWall/Build/Intermediates.noindex/PetWall.build/
  Release-iphonesimulator/PetKit.build/DerivedSources -F/x/src/PetWall/DerivedData/PetWall/Build/Products/Release-iphonesimulator -MMD -MT dependencies -MF /x/src/PetWall/DerivedData/PetWall/Build/Intermediates.noindex/PetWall.build/Release-iphonesimulator/PetKit.build/Objects-
  normal/x86_64/Cat.d --serialize-diagnostics /x/src/PetWall/DerivedData/PetWall/Build/Intermediates.noindex/PetWall.build/Release-iphonesimulator/PetKit.build/Objects-normal/x86_64/Cat.dia -c /x/src/PetWall/PetKit/Cat/Cat.mm -o /x/src/PetWall/DerivedData/PetWall/Build/
  Intermediates.noindex/PetWall.build/Release-iphonesimulator/PetKit.build/Objects-normal/x86_64/Cat.o
```

Compile PetKit_vers.c 0.1 seconds
Compile Pet.m 0.7 seconds
Link /x/src/PetWall/DerivedData/PetWall/Build/Products/Release-iphonesimulator/PetKit.framework/PetKit 0.1 seconds
Write /x/src/PetWall/DerivedData/PetWall/Build/Intermediates.noindex/PetWall.build/Release-iphonesimulator/PetKit.build/Script-14161AF520AA1FC500FD6992.sh 0.1 seconds
Generate /x/src/PetWall/DerivedData/PetWall/Build/Products/Release-iphonesimulator/PetKit.framework.dSYM 0.1 seconds
Run custom shell script 'Compile Pet Toys' 0.1 seconds
Sign /x/src/PetWall/DerivedData/PetWall/Build/Products/Release-iphonesimulator/PetKit.framework 0.1 seconds
Touch 0.1 seconds

Build target PetWall
Project PetWall | Configuration Release | Destination iPhone 8 Plus | SDK Simulator - iOS 12.0

Create Directory “Frameworks” 0.1 seconds
Create Directory “PetWall.app” 0.1 seconds
Write /x/src/PetWall/DerivedData/PetWall/Build/Intermediates.noindex/PetWall.build/Release-iphonesimulator/PetWall.build/DerivedSources/Entitlements.plist 0.1 seconds
Write /x/src/PetWall/DerivedData/PetWall/Build/Intermediates.noindex/PetWall.build/Release-iphonesimulator/PetWall.build/DerivedSources/Entitlements-Simulated.plist 0.1 seconds
Process product packaging 0.1 seconds
Process product packaging 0.1 seconds
Write /x/src/PetWall/DerivedData/PetWall/Build/Intermediates.noindex/PetWall.build/Release-iphonesimulator/PetWall.build/PetWall.hmap 0.1 seconds
Write /x/src/PetWall/DerivedData/PetWall/Build/Intermediates.noindex/PetWall.build/Release-iphonesimulator/PetWall.build/PetWall-project-headers.hmap 0.1 seconds
Write /x/src/PetWall/DerivedData/PetWall/Build/Intermediates.noindex/PetWall.build/Release-iphonesimulator/PetWall.build/PetWall-own-target-headers.hmap 0.1 seconds
Write /x/src/PetWall/DerivedData/PetWall/Build/Intermediates.noindex/PetWall.build/Release-iphonesimulator/PetWall.build/PetWall-generated-files.hmap 0.1 seconds
Write /x/src/PetWall/DerivedData/PetWall/Build/Intermediates.noindex/PetWall.build/Release-iphonesimulator/PetWall.build/PetWall-all-target-headers.hmap 0.1 seconds

How Do We Find Our Cat?



On the Search for Our Cat

```
$ clang <list of arguments> -c Cat.mm -o Cat.o -v
```

On the Search for Our Cat

```
#include "..." search starts here:



#include <...> search starts here:
```

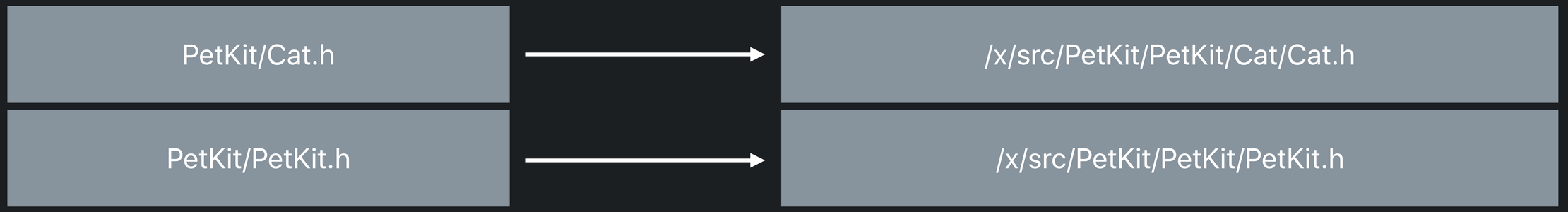
On the Search for Our Cat

```
#include "..." search starts here:
PetKit-generated-files.hmap (headermap)
PetKit-project-headers.hmap (headermap)

#include <...> search starts here:
PetKit-own-target-headers.hmap (headermap)
PetKit-all-target-headers.hmap (headermap)
DerivedSources
Build/Products/Debug (framework directory)
Header and Framework Locations in the SDK
```

What Are Header Maps?

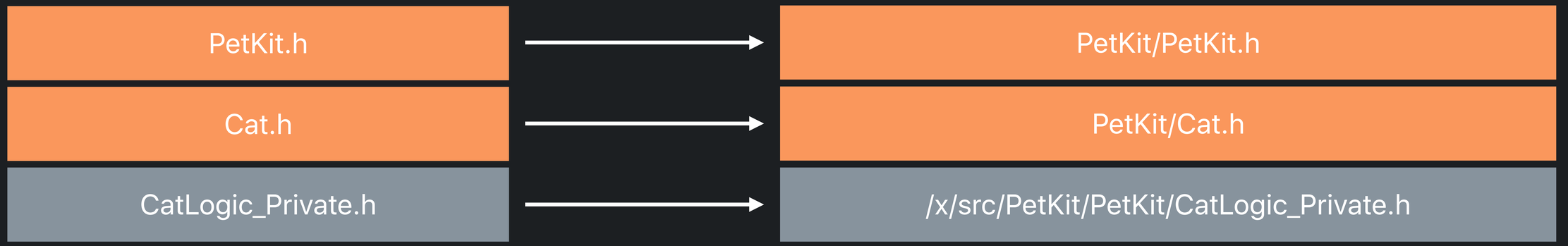
PetKit-project-headers.hmap:

PetKit-own-target-headers.hmap:

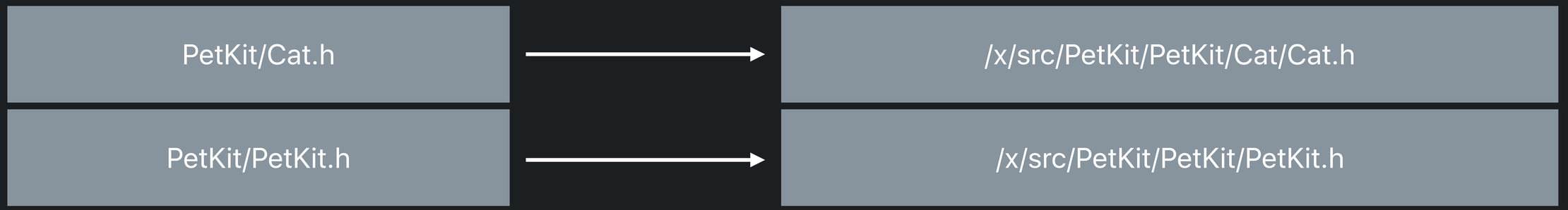


What Are Header Maps?

PetKit-project-headers.hmap:

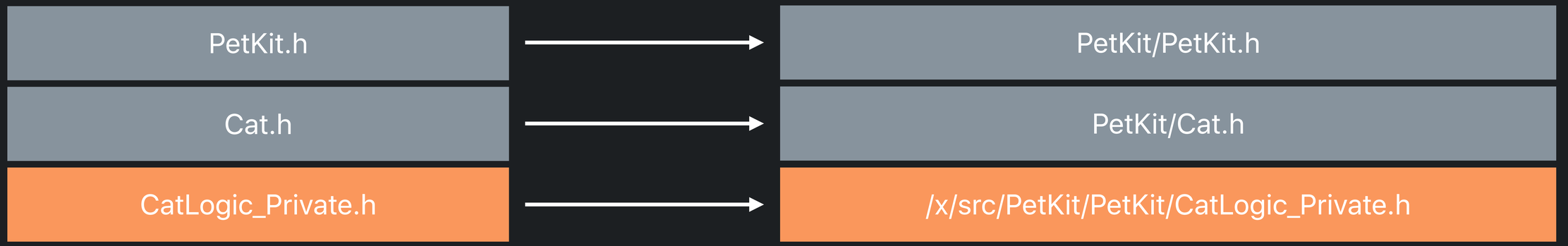


PetKit-own-target-headers.hmap:

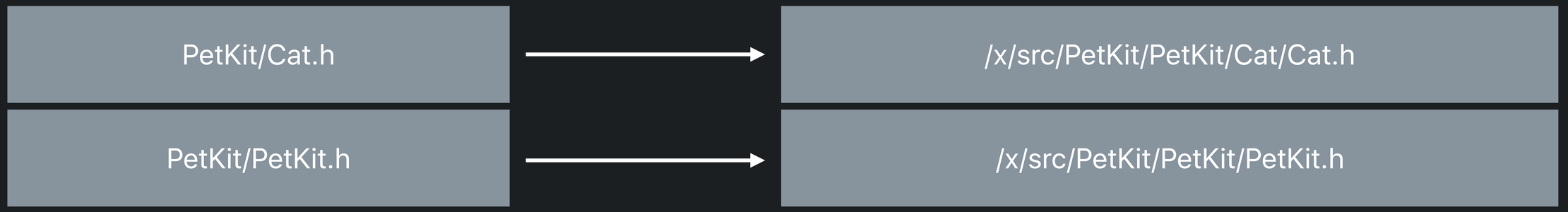


What Are Header Maps?

PetKit-project-headers.hmap:



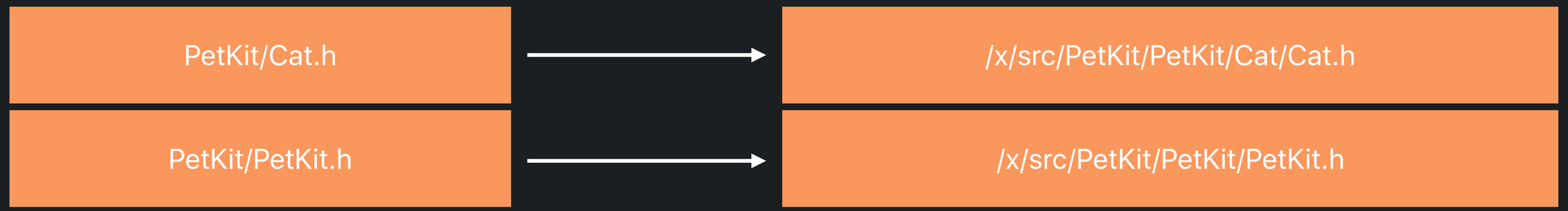
PetKit-own-target-headers.hmap:



What Are Header Maps?

PetKit-project-headers.hmap:

PetKit-own-target-headers.hmap:



Common Project Issues with Header Maps

Header is not part of the project

Headers with the same name shadow each other

How do we find system headers?

How Do We Find Foundation.h?

```
//
//  Pet.h
//  PetKit
//

#import <Foundation/Foundation.h>

NS_ASSUME_NONNULL_BEGIN
@interface Pet : NSObject
- (instancetype)initWithName:(NSString *)name;
@property (readonly) NSString *name;
@property (readonly) BOOL isHungry;
- (void)eat;
```

On the Search for Foundation.h

```
#include "..." search starts here:
PetKit-generated-files.hmap (headermap)
PetKit-project-headers.hmap (headermap)

#include <...> search starts here:
PetKit-own-target-headers.hmap (headermap)
PetKit-all-target-headers.hmap (headermap)
DerivedSources
Build/Products/Debug (framework directory)
$(SDKROOT)/usr/include
$(SDKROOT)/System/Library/Frameworks (framework directory)
```

On the Search for Foundation.h

```
#include <...> search starts here:
$(SDKROOT)/usr/include
$(SDKROOT)/System/Library/Frameworks (framework directory)
```

On the Search for Foundation.h

```
#include <...> search starts here:
$(SDKROOT)/usr/include
$(SDKROOT)/System/Library/Frameworks (framework directory)
```

On the Search for Foundation.h

```
#include <...> search starts here:
$(SDKROOT)/usr/include
$(SDKROOT)/System/Library/Frameworks (framework directory)
```

On the Search for Foundation.h

```
$(SDKROOT)/usr/include/Foundation/Foundation.h
```

On the Search for Foundation.h

```
$(SDKROOT)/usr/include/Foundation/Foundation.h
```

On the Search for Foundation.h

```
#include <...> search starts here:
$(SDKROOT)/usr/include
$(SDKROOT)/System/Library/Frameworks (framework directory)
```

On the Search for Foundation.h

```
$SDKROOT/System/Library/Frameworks/Foundation.framework
```

On the Search for Foundation.h

```
$SDKROOT/System/Library/Frameworks/Foundation.framework
```

On the Search for Foundation.h

```
$SDKROOT/System/Library/Frameworks/Foundation.framework
```

```
$SDKROOT/System/Library/Frameworks/Foundation.framework/Headers/Foundation.h
```

On the Search for Foundation.h

```
$SDKROOT/System/Library/Frameworks/Foundation.framework
```

```
$SDKROOT/System/Library/Frameworks/Foundation.framework/Headers/Foundation.h
```

On the Search for Bogus.h

```
$SDKROOT/System/Library/Frameworks/Foundation.framework
```

```
$SDKROOT/System/Library/Frameworks/Foundation.framework/Headers/Bogus.h
```

On the Search for Bogus.h

```
$SDKROOT/System/Library/Frameworks/Foundation.framework
```

```
$SDKROOT/System/Library/Frameworks/Foundation.framework/Headers/Bogus.h
```

On the Search for Bogus.h

```
$SDKROOT/System/Library/Frameworks/Foundation.framework
```

```
$SDKROOT/System/Library/Frameworks/Foundation.framework/Headers/Bogus.h
```

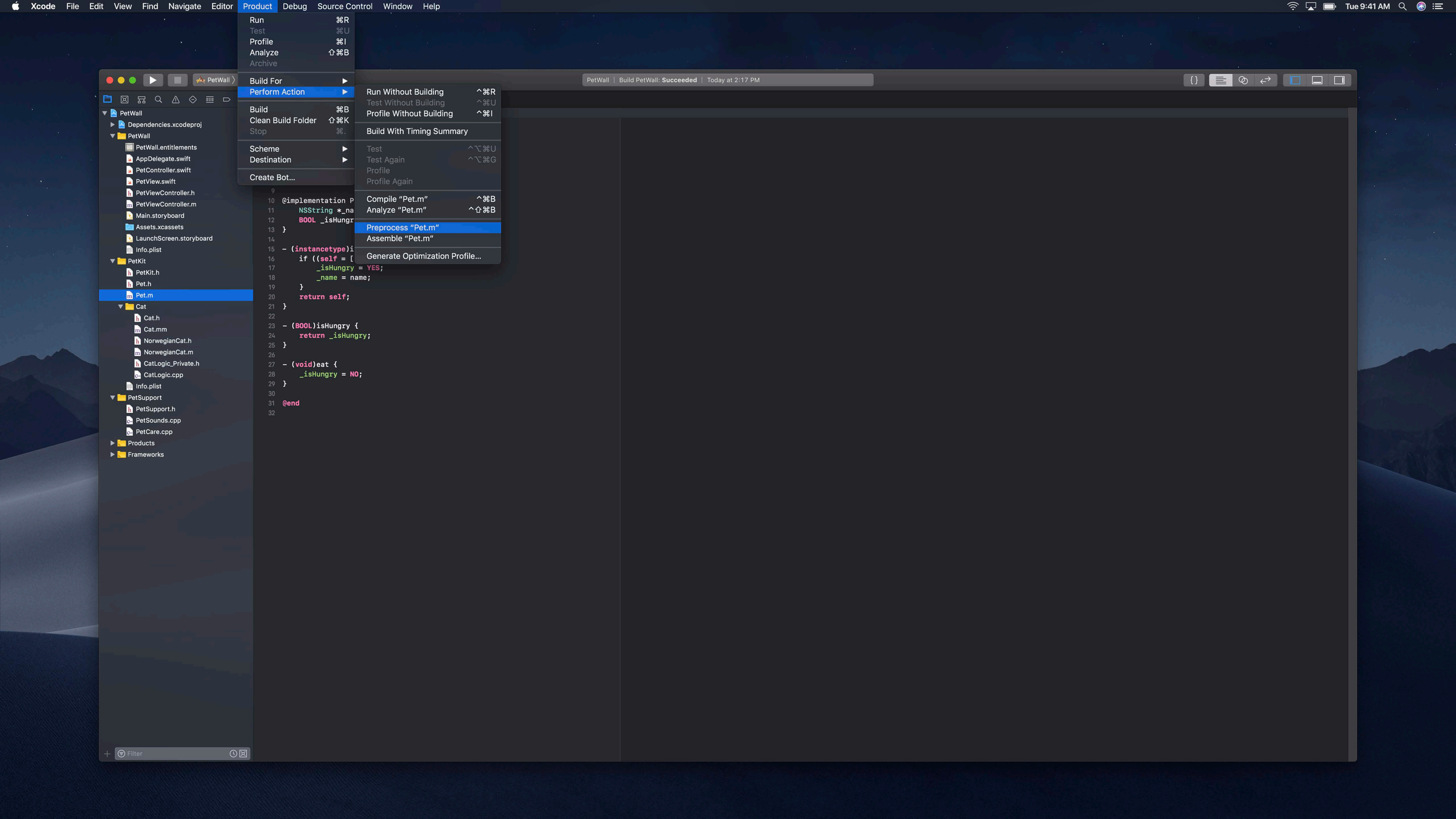
```
$SDKROOT/System/Library/Frameworks/Foundation.framework/PrivateHeaders/Bogus.h
```

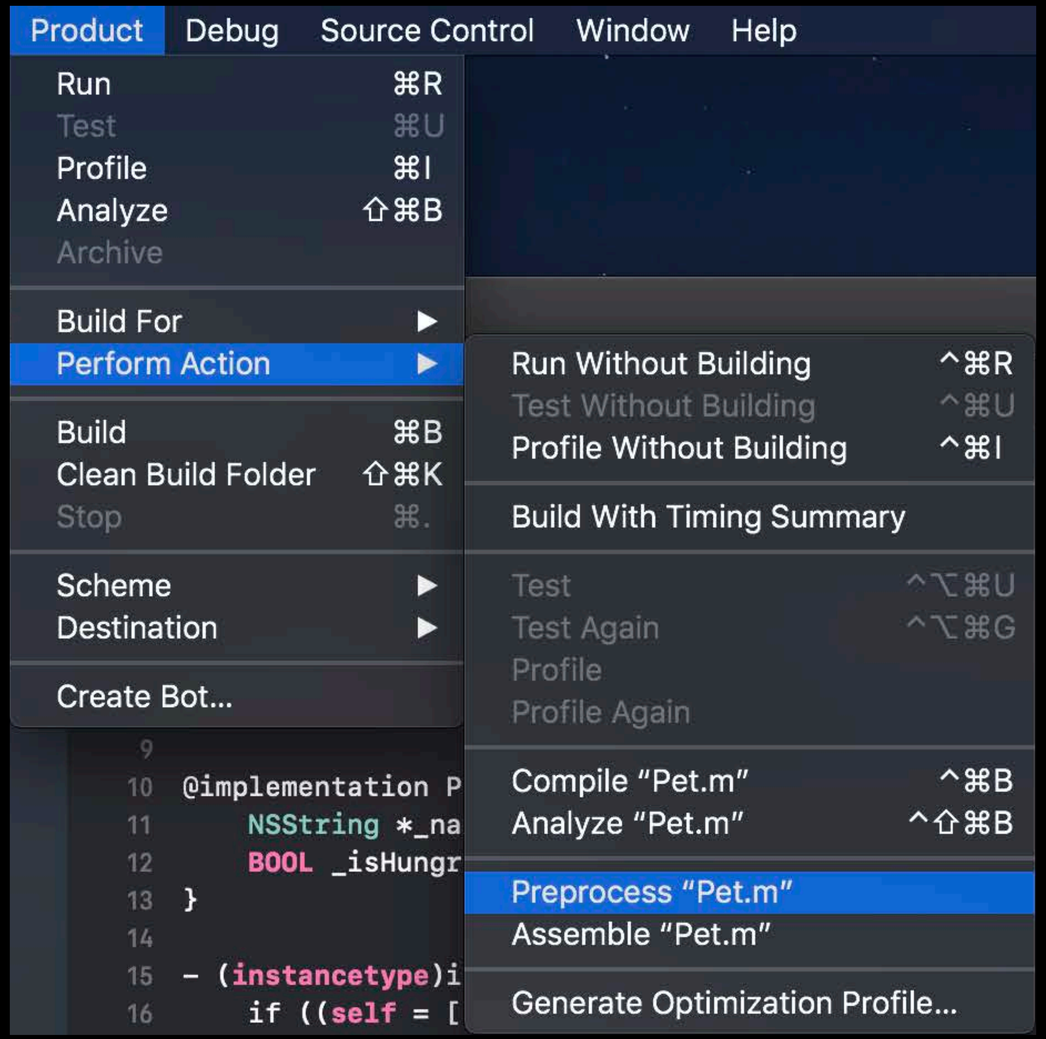
On the Search for Bogus.h

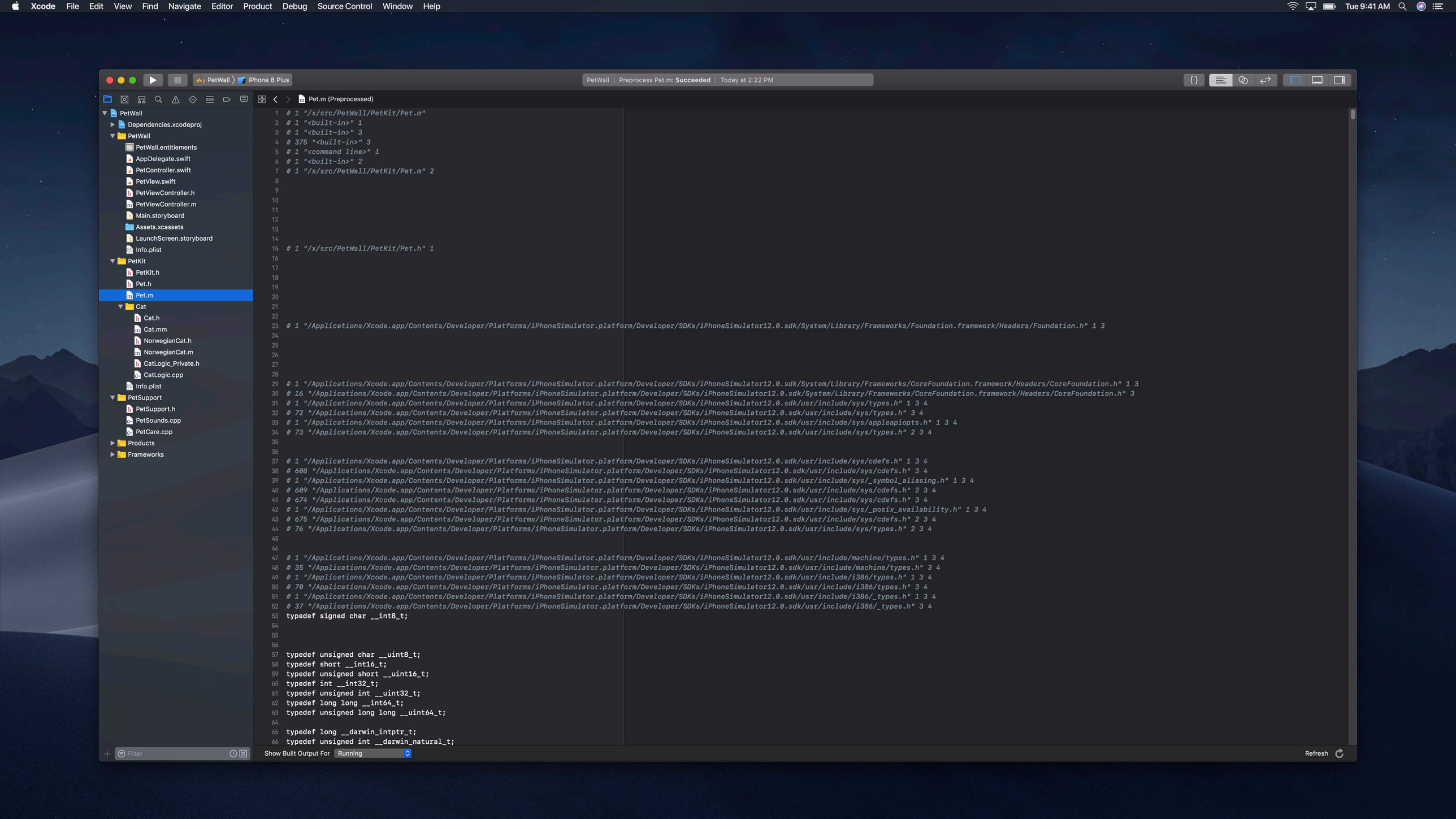
```
$SDKROOT/System/Library/Frameworks/Foundation.framework
```

```
$SDKROOT/System/Library/Frameworks/Foundation.framework/Headers/Bogus.h
```

```
$SDKROOT/System/Library/Frameworks/Foundation.framework/PrivateHeaders/Bogus.h
```


Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help
Tue 9:41 AM
Run ⌘R
Test ⌘U
Profile ⌘I
Analyze ⇧⌘B
Archive
Build For
Perform Action
Build ⌘B
Clean Build Folder ⇧⌘K
Stop ⌘.
Scheme
Destination
Create Bot...
Run Without Building ^⌘R
Test Without Building ^⌘U
Profile Without Building ^⌘I
Build With Timing Summary
Test ^⌥⌘U
Test Again ^⌥⌘G
Profile
Profile Again
Compile "Pet.m" ^⌘B
Analyze "Pet.m" ^⇧⌘B
Preprocess "Pet.m"
Assemble "Pet.m"
Generate Optimization Profile...
PetWall | Build PetWall: Succeeded | Today at 2:17 PM
PetWall
Dependencies.xcodeproj
PetWall
PetWall.entitlements
AppDelegate.swift
PetController.swift
PetView.swift
PetViewController.h
PetViewController.m
Main.storyboard
Assets.xcassets
LaunchScreen.storyboard
Info.plist
PetKit
PetKit.h
Pet.h
Pet.m
Cat
Cat.h
Cat.mm
NorwegianCat.h
NorwegianCat.m
CatLogic_Private.h
CatLogic.cpp
Info.plist
PetSupport
PetSupport.h
PetSounds.cpp
PetCare.cpp
Products
Frameworks
Filter
@implementation P
NSString *_na
BOOL _isHungr
}
- (instancetype)i
if ((self = [
_isHungry = YES;
_name = name;
}
return self;
}
- (BOOL)isHungry {
return _isHungry;
}
- (void)eat {
_isHungry = NO;
}
@end


Product
Debug
Source Control
Window
Help
Run ⌘R
Test ⌘U
Profile ⌘I
Analyze ⇧⌘B
Archive
Build For ▶
Perform Action ▶
Build ⌘B
Clean Build Folder ⇧⌘K
Stop ⌘.
Scheme ▶
Destination ▶
Create Bot...
Run Without Building ^⌘R
Test Without Building ^⌘U
Profile Without Building ^⌘I
Build With Timing Summary
Test ^⌥⌘U
Test Again ^⌥⌘G
Profile
Profile Again
Compile “Pet.m” ^⌘B
Analyze “Pet.m” ^⇧⌘B
Preprocess “Pet.m”
Assemble “Pet.m”
Generate Optimization Profile...
9
10 @implementation P
11 NSString *_na
12 BOOL _isHungr
13 }
14
15 - (instancetype)i
16 if ((self = [


Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help
Tue 9:41 AM
PetWall 〉 iPhone 8 Plus
PetWall | Preprocess Pet.m: Succeeded | Today at 2:22 PM
Pet.m (Preprocessed)
PetWall
Dependencies.xcodeproj
PetWall
PetWall.entitlements
AppDelegate.swift
PetController.swift
PetView.swift
PetViewController.h
PetViewController.m
Main.storyboard
Assets.xcassets
LaunchScreen.storyboard
Info.plist
PetKit
PetKit.h
Pet.h
Pet.m
Cat
Cat.h
Cat.mm
NorwegianCat.h
NorwegianCat.m
CatLogic_Private.h
CatLogic.cpp
Info.plist
PetSupport
PetSupport.h
PetSounds.cpp
PetCare.cpp
Products
Frameworks
Filter
1 "/x/src/PetWall/PetKit/Pet.m"
1 "<built-in>" 1
1 "<built-in>" 3
375 "<built-in>" 3
1 "<command line>" 1
1 "<built-in>" 2
1 "/x/src/PetWall/PetKit/Pet.m" 2
1 "/x/src/PetWall/PetKit/Pet.h" 1
1 "/Applications/Xcode.app/Contents/Developer/Platforms/iPhoneSimulator.platform/Developer/SDKs/iPhoneSimulator12.0.sdk/System/Library/Frameworks/Foundation.framework/Headers/Foundation.h" 1 3
1 "/Applications/Xcode.app/Contents/Developer/Platforms/iPhoneSimulator.platform/Developer/SDKs/iPhoneSimulator12.0.sdk/System/Library/Frameworks/CoreFoundation.framework/Headers/CoreFoundation.h" 1 3
16 "/Applications/Xcode.app/Contents/Developer/Platforms/iPhoneSimulator.platform/Developer/SDKs/iPhoneSimulator12.0.sdk/System/Library/Frameworks/CoreFoundation.framework/Headers/CoreFoundation.h" 3
1 "/Applications/Xcode.app/Contents/Developer/Platforms/iPhoneSimulator.platform/Developer/SDKs/iPhoneSimulator12.0.sdk/usr/include/sys/types.h" 1 3 4
72 "/Applications/Xcode.app/Contents/Developer/Platforms/iPhoneSimulator.platform/Developer/SDKs/iPhoneSimulator12.0.sdk/usr/include/sys/types.h" 3 4
1 "/Applications/Xcode.app/Contents/Developer/Platforms/iPhoneSimulator.platform/Developer/SDKs/iPhoneSimulator12.0.sdk/usr/include/sys/appleapiopts.h" 1 3 4
73 "/Applications/Xcode.app/Contents/Developer/Platforms/iPhoneSimulator.platform/Developer/SDKs/iPhoneSimulator12.0.sdk/usr/include/sys/types.h" 2 3 4
1 "/Applications/Xcode.app/Contents/Developer/Platforms/iPhoneSimulator.platform/Developer/SDKs/iPhoneSimulator12.0.sdk/usr/include/sys/cdefs.h" 1 3 4
608 "/Applications/Xcode.app/Contents/Developer/Platforms/iPhoneSimulator.platform/Developer/SDKs/iPhoneSimulator12.0.sdk/usr/include/sys/cdefs.h" 3 4
1 "/Applications/Xcode.app/Contents/Developer/Platforms/iPhoneSimulator.platform/Developer/SDKs/iPhoneSimulator12.0.sdk/usr/include/sys/_symbol_aliasing.h" 1 3 4
609 "/Applications/Xcode.app/Contents/Developer/Platforms/iPhoneSimulator.platform/Developer/SDKs/iPhoneSimulator12.0.sdk/usr/include/sys/cdefs.h" 2 3 4
674 "/Applications/Xcode.app/Contents/Developer/Platforms/iPhoneSimulator.platform/Developer/SDKs/iPhoneSimulator12.0.sdk/usr/include/sys/cdefs.h" 3 4
1 "/Applications/Xcode.app/Contents/Developer/Platforms/iPhoneSimulator.platform/Developer/SDKs/iPhoneSimulator12.0.sdk/usr/include/sys/_posix_availability.h" 1 3 4
675 "/Applications/Xcode.app/Contents/Developer/Platforms/iPhoneSimulator.platform/Developer/SDKs/iPhoneSimulator12.0.sdk/usr/include/sys/cdefs.h" 2 3 4
76 "/Applications/Xcode.app/Contents/Developer/Platforms/iPhoneSimulator.platform/Developer/SDKs/iPhoneSimulator12.0.sdk/usr/include/sys/types.h" 2 3 4
1 "/Applications/Xcode.app/Contents/Developer/Platforms/iPhoneSimulator.platform/Developer/SDKs/iPhoneSimulator12.0.sdk/usr/include/machine/types.h" 1 3 4
35 "/Applications/Xcode.app/Contents/Developer/Platforms/iPhoneSimulator.platform/Developer/SDKs/iPhoneSimulator12.0.sdk/usr/include/machine/types.h" 3 4
1 "/Applications/Xcode.app/Contents/Developer/Platforms/iPhoneSimulator.platform/Developer/SDKs/iPhoneSimulator12.0.sdk/usr/include/i386/types.h" 1 3 4
70 "/Applications/Xcode.app/Contents/Developer/Platforms/iPhoneSimulator.platform/Developer/SDKs/iPhoneSimulator12.0.sdk/usr/include/i386/types.h" 3 4
1 "/Applications/Xcode.app/Contents/Developer/Platforms/iPhoneSimulator.platform/Developer/SDKs/iPhoneSimulator12.0.sdk/usr/include/i386/_types.h" 1 3 4
37 "/Applications/Xcode.app/Contents/Developer/Platforms/iPhoneSimulator.platform/Developer/SDKs/iPhoneSimulator12.0.sdk/usr/include/i386/_types.h" 3 4
typedef signed char __int8_t;
typedef unsigned char __uint8_t;
typedef short __int16_t;
typedef unsigned short __uint16_t;
typedef int __int32_t;
typedef unsigned int __uint32_t;
typedef long long __int64_t;
typedef unsigned long long __uint64_t;
typedef long __darwin_intptr_t;
typedef unsigned int __darwin_natural_t;
Show Built Output For
Running
Refresh

```
#import <Foundation/Foundation.h>
```

>800

headers included

>9MB

of source code

How can we do better?

Clang Modules

On-disk cached header representation

Reusable

Faster build times

Module Properties

Module Properties

Context-free:

Module Properties

Context-free:

```
#define ENABLE_CHINCHILLA 1 // ignored
#import <PetKit/PetKit.h>
```

```
#define ENABLE_DOG 1 // ignored
#import <PetKit/PetKit.h>
```

Module Properties

Context-free:

```
#define ENABLE_CHINCHILLA 1 // ignored
#import <PetKit/PetKit.h>
```

```
#define ENABLE_DOG 1 // ignored
#import <PetKit/PetKit.h>
```

Self-contained:

Module Properties

Context-free:

```
#define ENABLE_CHINCHILLA 1 // ignored
#import <PetKit/PetKit.h>
```

```
#define ENABLE_DOG 1 // ignored
#import <PetKit/PetKit.h>
```

Self-contained:

```
#import <Foundation/Foundation.h> // import not required before PetKit module
#import <PetKit/Cat.h>
```

```
#import <Foundation/NSString.h>
```

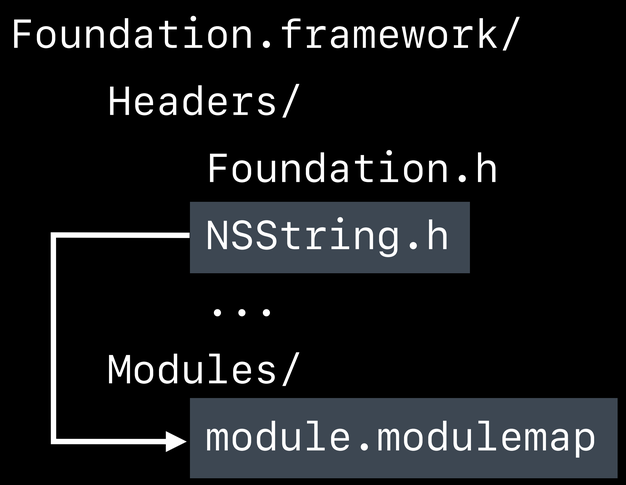
Foundation Framework

```
Foundation.framework/
    Headers/
        Foundation.h
        NSString.h
        ...
    Modules/
        module.modulemap
```

Foundation Framework

```
Foundation.framework/
    Headers/
        Foundation.h
        NSString.h
        ...
    Modules/
        module.modulemap
```

Foundation Framework



```
// Module Map - Foundation.framework/Modules/module.modulemap

framework module Foundation [extern_c] [system] {
    umbrella header "Foundation.h"

    export *
    module * {
        export *
    }

    explicit module NSDebug {
        header "NSDebug.h"
        export *
    }
}
```

```
// Module Map - Foundation.framework/Modules/module.modulemap

framework module Foundation [extern_c] [system] {
    umbrella header "Foundation.h"

    export *
    module * {
        export *
    }

    explicit module NSDebug {
        header "NSDebug.h"
        export *
    }
}
```

```
// Module Map - Foundation.framework/Modules/module.modulemap

framework module Foundation [extern_c] [system] {
    umbrella header "Foundation.h"

    export *
    module * {
        export *
    }

    explicit module NSDebug {
        header "NSDebug.h"
        export *
    }
}
```

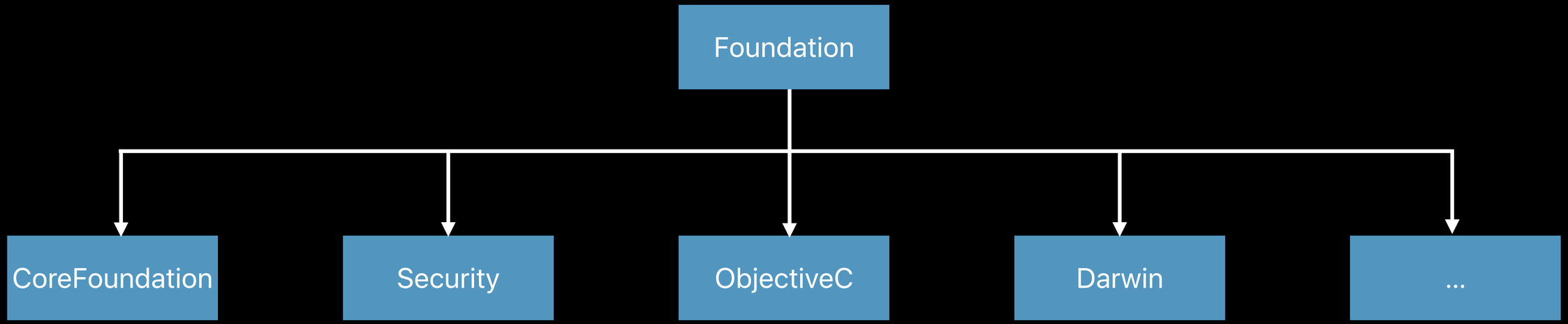
```
// Module Map - Foundation.framework/Modules/module.modulemap

framework module Foundation [extern_c] [system] {
    umbrella header "Foundation.h"

    export *
    module * {
        export *
    }

    explicit module NSDebug {
        header "NSDebug.h"
        export *
    }
}
```

```
// Foundation.h

…
#import <Foundation/NSScanner.h>
#import <Foundation/NSSet.h>
#import <Foundation/NSSortDescriptor.h>
#import <Foundation/NSStream.h>
#import <Foundation/NSString.h>
#import <Foundation/NSTextCheckingResult.h>
#import <Foundation/NSThread.h>
#import <Foundation/NSTimeZone.h>
…
```

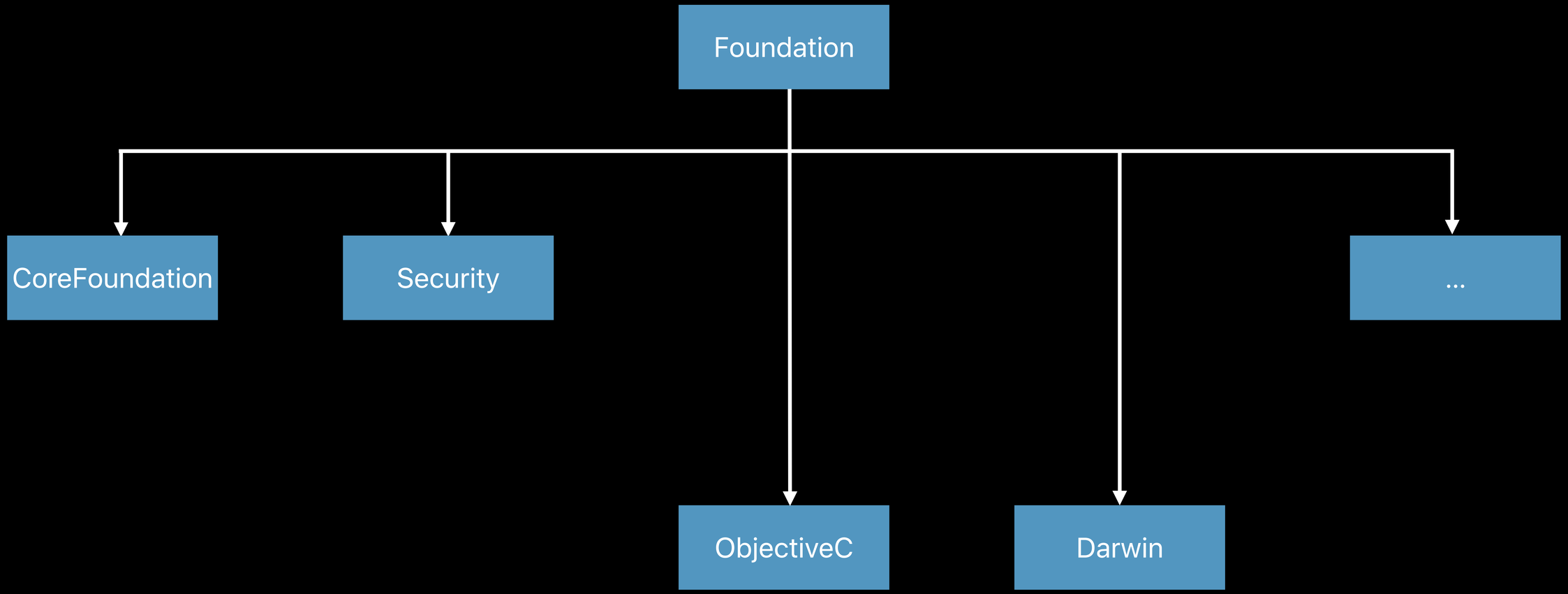
```
// Foundation.h

…
#import <Foundation/NSScanner.h>
#import <Foundation/NSSet.h>
#import <Foundation/NSSortDescriptor.h>
#import <Foundation/NSStream.h>
#import <Foundation/NSString.h>
#import <Foundation/NSTextCheckingResult.h>
#import <Foundation/NSThread.h>
#import <Foundation/NSTimeZone.h>
…
```

Building the Module

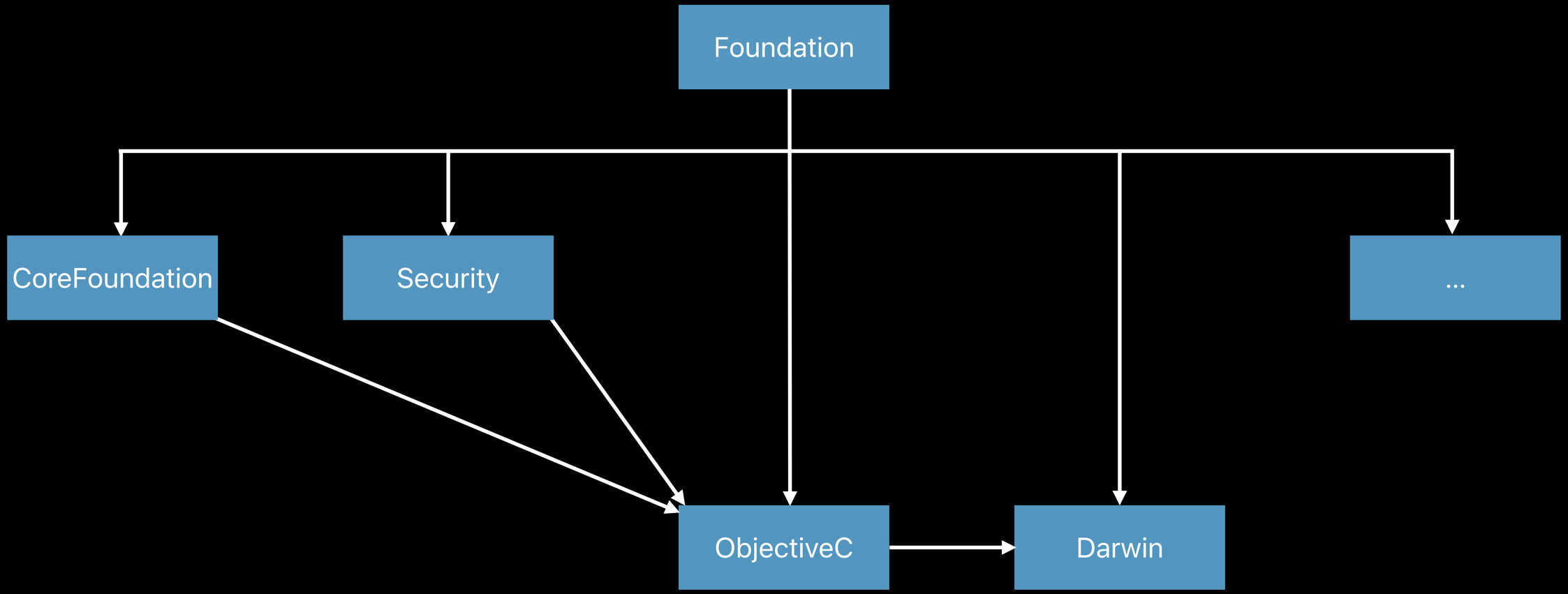
Building the Module



Building the Module



Building the Module



Module Cache

```
$ clang -fmodules -DENABLE_CHINCHILLA=1 …
```

Module Cache

```
$ clang -fmodules -DENABLE_CHINCHILLA=1 …
```

```
98XN8P5QH5OQ/
  CoreFoundation-2A5I5R2968COJ.pcm
  Security-1A229VWPAK67R.pcm
  Foundation-1RDF848B47PF4.pcm
```

Module Cache

```
$ clang -fmodules -DENABLE_CHINCHILLA=1 …
```

```
98XN8P5QH5OQ/
  CoreFoundation-2A5I5R2968COJ.pcm
  Security-1A229VWPAK67R.pcm
  Foundation-1RDF848B47PF4.pcm
```

```
$ clang -fmodules -DENABLE_CAT=1 …
```

Module Cache

```
$ clang -fmodules -DENABLE_CHINCHILLA=1 …
```

```
98XN8P5QH5OQ/
  CoreFoundation-2A5I5R2968COJ.pcm
  Security-1A229VWPAK67R.pcm
  Foundation-1RDF848B47PF4.pcm
```

```
$ clang -fmodules -DENABLE_CAT=1 …
```

```
1GYDULU5XJRF/
  CoreFoundation-2A5I5R2968COJ.pcm
  Security-1A229VWPAK67R.pcm
  Foundation-1RDF848B47PF4.pcm
```

How Do We Find Our Cat... Again?

This time with Clang Modules

```
//
//  NorwegianCat.h
//  PetKit
//

#import <PetKit/Cat.h>

NS_ASSUME_NONNULL_BEGIN
@interface NorwegianCat: Cat
...
@end
NS_ASSUME_NONNULL_END
```

PetWall Project Directory

```
PetWall/
    PetKit/
        Cat/
            Cat.h
            Cat.mm
            ...
    Info.plist
    Pet.h
    Pet.m
    PetKit.h
```

Clang’s Virtual File System (VFS)

```
PetKit.framework/
    Headers/
        Cat.h
        PetKit.h
        ...
    Modules/
        module.modulemap
```

Clang’s Virtual File System (VFS)

```
PetWall/
    PetKit/
        Cat/
            Cat.h
            Cat.mm
            ...
    Info.plist
    Pet.h
    Pet.m
    PetKit.h
```

```
PetKit.framework/
    Headers/
        Cat.h
        PetKit.h
        ...
    Modules/
        module.modulemap
```

Potential Modules Pitfalls

```
#import <PetKit/PetKit.h>
#import “Cat.h”
```

Potential Modules Pitfalls

```
#define ENABLE_CUTE_KITTENS 1

#import <PetKit/PetKit.h>
#import “Cat.h”
```

Potential Modules Pitfalls

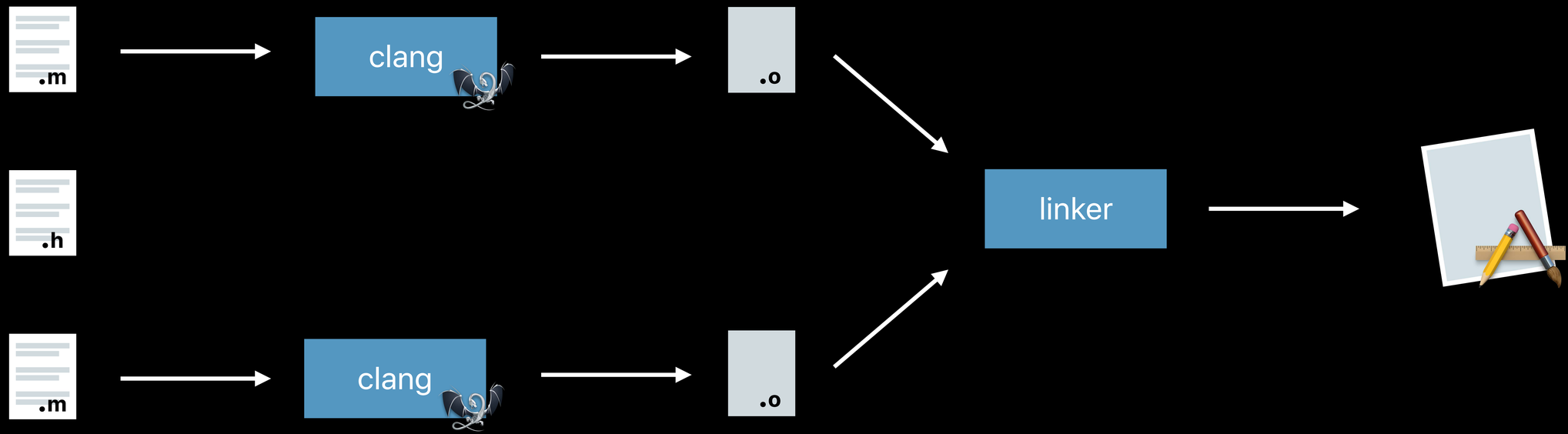
```
#define ENABLE_CUTE_KITTENS 1

#import <PetKit/PetKit.h>
#import <PetKit/Cat.h>
```

Behind the Scenes of the Xcode Build Process: Swift Builds

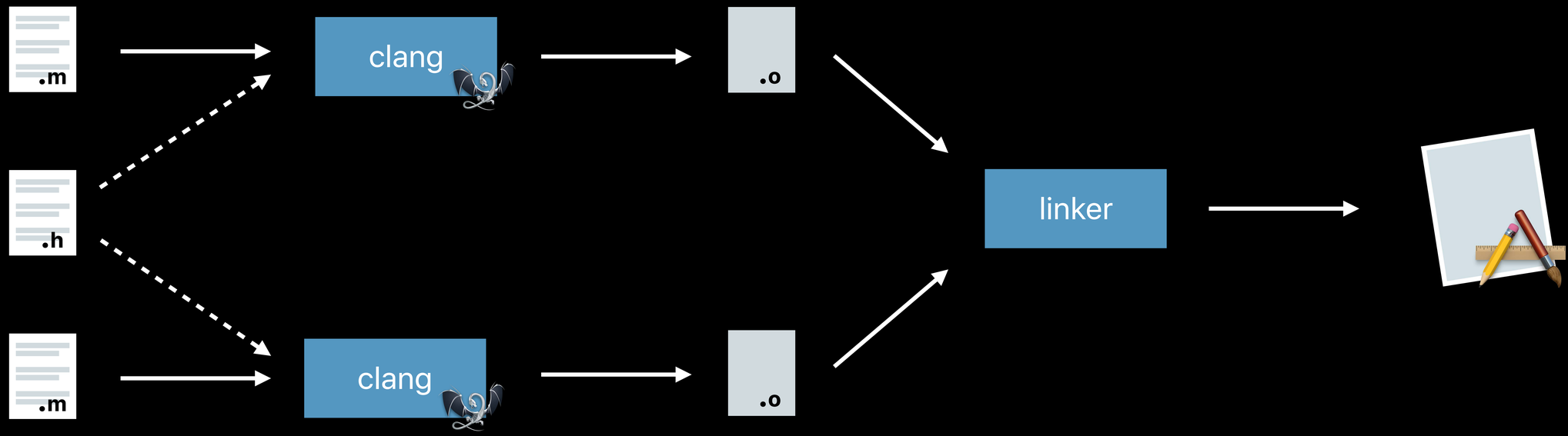
Devin Coughlin, Program Analysis Team


10
1
0
001
100
01
100

Recap: Clang Uses Headers to Find Declarations



Recap: Clang Uses Headers to Find Declarations



Swift Does Not Have Headers

Easier for beginners to get started

Avoids repeating declarations in separate file

Compiler does book-keeping to find declarations

```
import UIKit
class PetViewController: UIViewController {
  var view = PetView(name: "Fido", frame: frame)
  …
}
```

```swift
import UIKit
class PetViewController: UIViewController {
  var view = PetView(name: "Fido", frame: frame)
  …
}
```

```objc
#import "PetWall-Swift.h"
@implementation AppDelegate
…
@end
```

```swift
import UIKit
class PetViewController: UIViewController {
  var view = PetView(name: "Fido", frame: frame)
  …
}
```

```objc
#import "PetWall-Swift.h"
@implementation AppDelegate
…
@end
```

```swift
@testable import PetWall
class TestPetViewController: XCTestCase {
}
```

```swift
import UIKit
class PetViewController: UIViewController {
  var view = PetView(name: "Fido", frame: frame)
  …
}
```

```objc
#import "PetWall-Swift.h"
@implementation AppDelegate
…
@end
```

```swift
@testable import PetWall
class TestPetViewController: XCTestCase {
}
```

Finding declarations

```
import UIKit
class PetViewController: UIViewController {
  var view = PetView(name: "Fido", frame: frame)
  …
}
```

```
#import "PetWall-Swift.h"
@implementation AppDelegate
…
@end
```

```
@testable import PetWall
class TestPetViewController: XCTestCase {
}
```

Finding declarations

- Within a Swift target

```swift
import UIKit
class PetViewController: UIViewController {
  var view = PetView(name: "Fido", frame: frame)
  …
}
```

```objc
#import "PetWall-Swift.h"
@implementation AppDelegate
…
@end
```

```swift
@testable import PetWall
class TestPetViewController: XCTestCase {
}
```

Finding declarations

- Within a Swift target
- From Objective-C

```swift
import UIKit
class PetViewController: UIViewController {
  var view = PetView(name: "Fido", frame: frame)
  …
}
```

```objc
#import "PetWall-Swift.h"
@implementation AppDelegate
…
@end
```

```swift
@testable import PetWall
class TestPetViewController: XCTestCase {
}
```

Finding declarations

- Within a Swift target
- From Objective-C

Generating interfaces

```swift
import UIKit
class PetViewController: UIViewController {
  var view = PetView(name: "Fido", frame: frame)
  …
}
```

```objc
#import "PetWall-Swift.h"
@implementation AppDelegate
…
@end
```

```swift
@testable import PetWall
class TestPetViewController: XCTestCase {
}
```

Finding declarations

- Within a Swift target
- From Objective-C

Generating interfaces

- To use in Objective-C

```swift
import UIKit
class PetViewController: UIViewController {
  var view = PetView(name: "Fido", frame: frame)
  …
}
```

```swift
#import "PetWall-Swift.h"
@implementation AppDelegate
…
@end
```

```swift
@testable import PetWall
class TestPetViewController: XCTestCase {
}
```

Finding declarations

- Within a Swift target
- From Objective-C

Generating interfaces

- To use in Objective-C
- To use in other Swift targets

```swift
import UIKit
class PetViewController: UIViewController {
  var view = PetView(name: "Fido", frame: frame)
  …
}
```

```objc
#import "PetWall-Swift.h"
@implementation AppDelegate
…
@end
```

```swift
@testable import PetWall
class TestPetViewController: XCTestCase {
}
```

Finding declarations

- Within a Swift target
- From Objective-C

Generating interfaces

- To use in Objective-C
- To use in other Swift targets

Finding Declarations Within a Swift Target

```
import UIKit
class PetViewController: UIViewController {
  var view = PetView(name: "Fido", frame: frame)
  …
}
```

Finding Declarations Within a Swift Target

```
import UIKit
class PetViewController: UIViewController {
  var view = PetView(name: "Fido", frame: frame)
  …
}
```

Finding Declarations Within a Swift Target

PetViewController.swift

```
import UIKit
class PetViewController: UIViewController {
  var view = PetView(name: "Fido", frame: frame)
  …
}
```

Finding Declarations Within a Swift Target

PetViewController.swift

```
import UIKit
class PetViewController: UIViewController {
  var view = PetView(name: "Fido", frame: frame)
  …
}
```

Finding Declarations Within a Swift Target

PetViewController.swift

```
import UIKit
class PetViewController: UIViewController {
  var view = PetView(name: "Fido", frame: frame)
  …
}
```

PetView.swift

```
class PetView: UIView {
  init(name: String, frame: CGRect) {
    self.name = name
    super.init(frame: frame)
  }
}
```

Load and parse file

Finding Declarations Within a Swift Target

PetViewController.swift

```
import UIKit
class PetViewController: UIViewController {
  var view = PetView(name: "Fido", frame: frame)
  …
}
```

PetView.swift

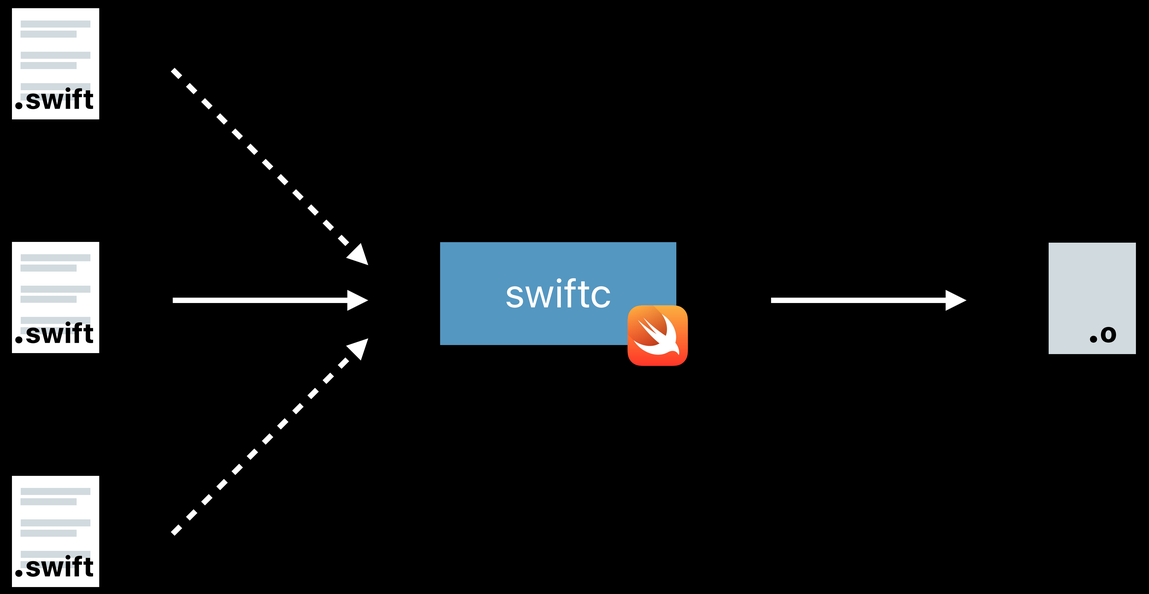
```
class PetView: UIView {
  init(name: String, frame: CGRect) {
    self.name = name
    super.init(frame: frame)
  }
}
```

Load and parse file

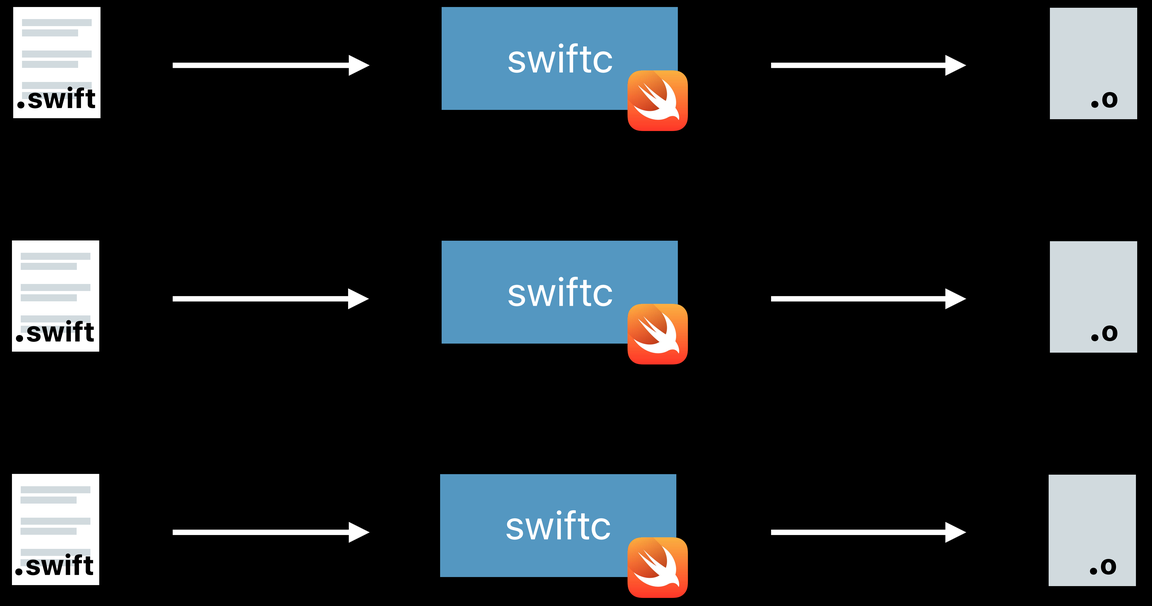
Does not check types in bodies

Compiler Looks at All Files Within Target

Compiler Looks at All Files Within Target



Xcode 9: Repeated Work in Debug Builds

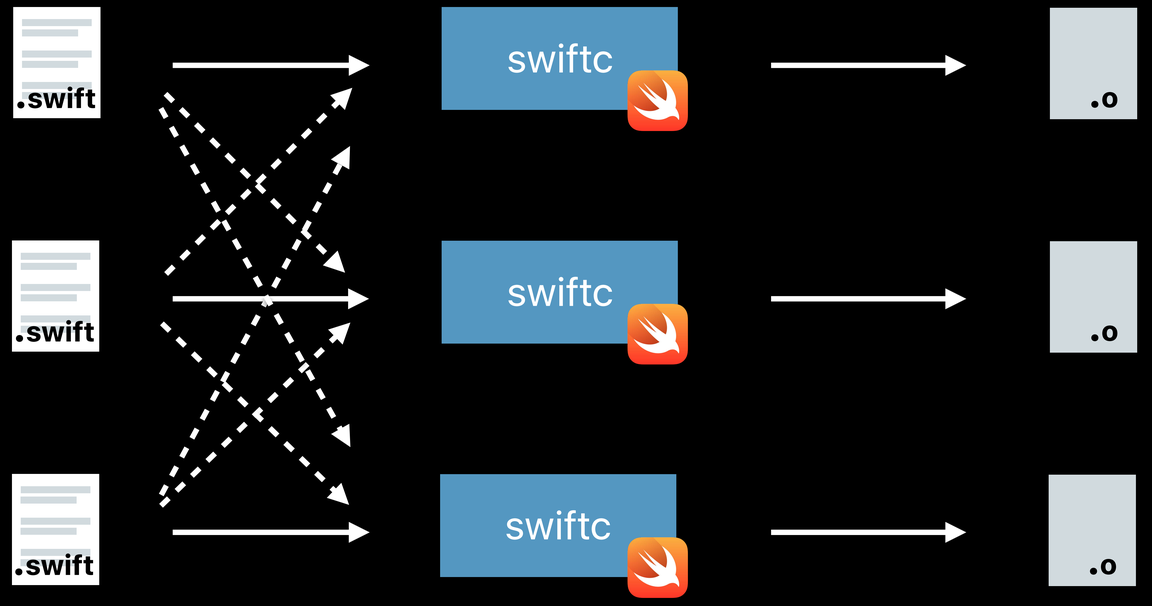


Compiled each file separately

Enabled parallel, incremental compilation

Compiler had to repeatedly parse all files to find declarations

Xcode 9: Repeated Work in Debug Builds



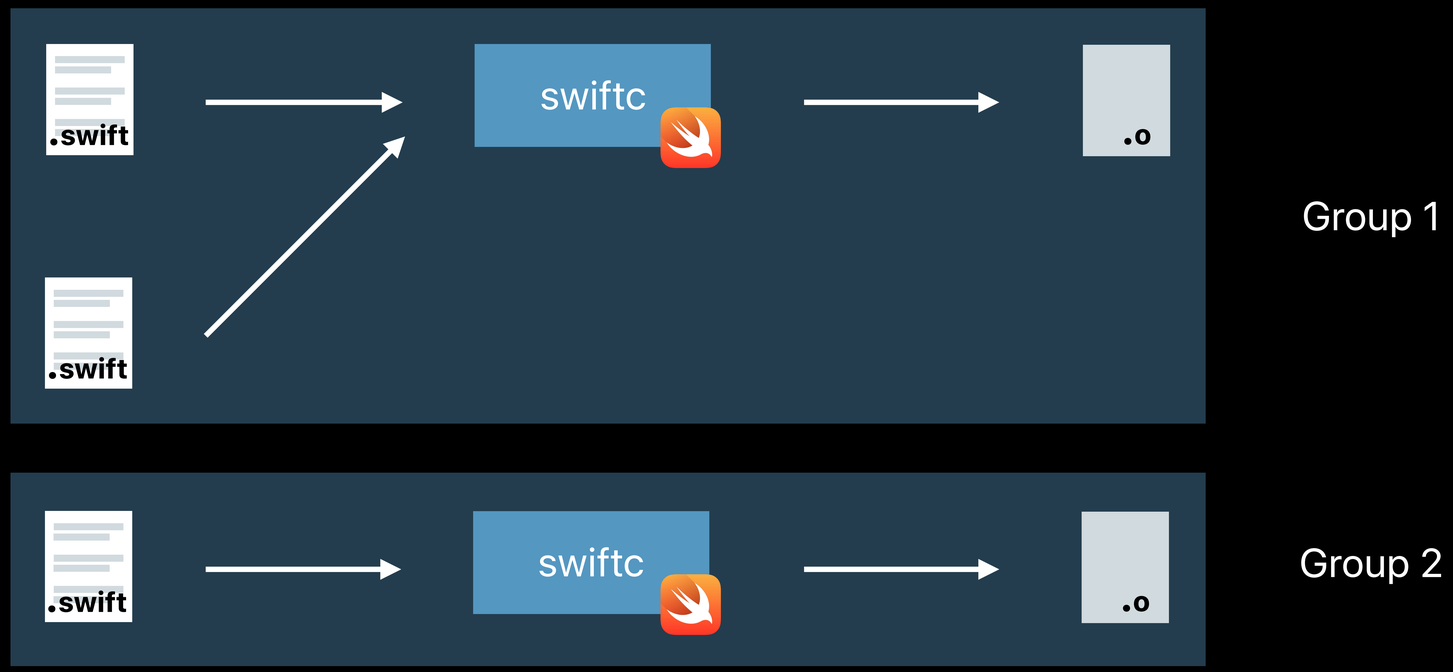
Compiled each file separately

Enabled parallel, incremental compilation

Compiler had to repeatedly parse all files to find declarations

Xcode 10: More Sharing in Debug Builds

NEW



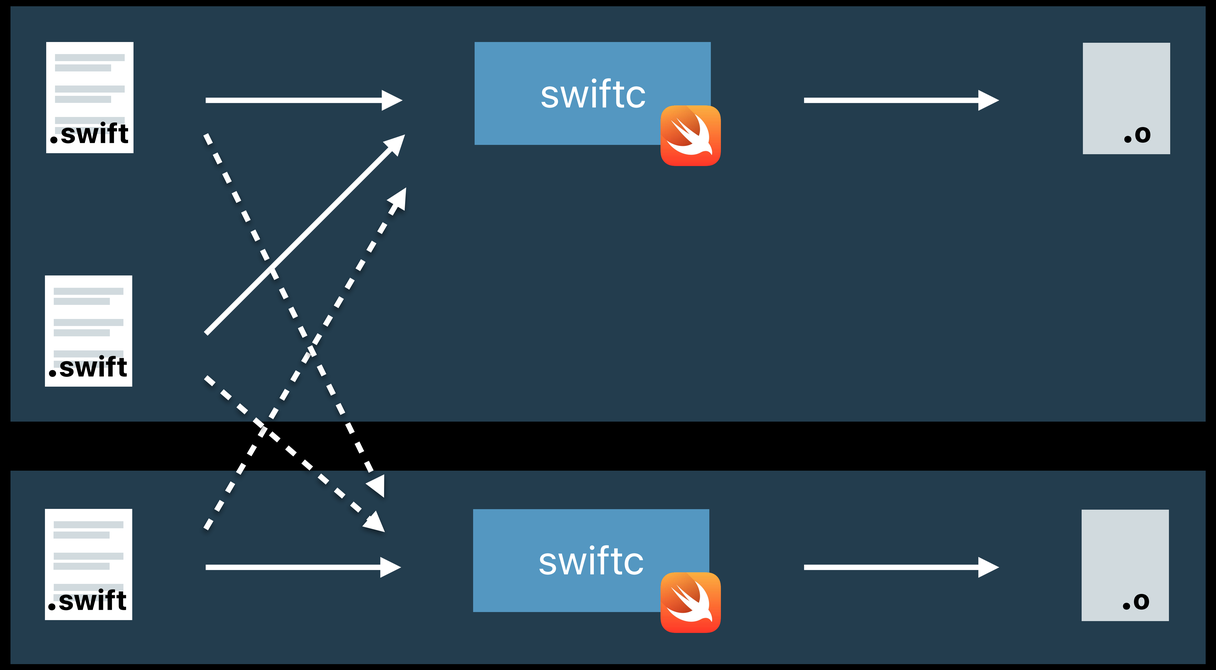
Group multiple files into common compilation process per core

Shares work of parsing within a process

Only repeat parsing across processes

Xcode 10: More Sharing in Debug Builds

NEW



Group multiple files into common compilation process per core

Shares work of parsing within a process

Only repeat parsing across processes

```swift
import UIKit
class PetViewController: UIViewController {
  var view = PetView(name: "Fido", frame: frame)
  …
}
```

```objc
#import "PetWall-Swift.h"
@implementation AppDelegate
…
@end
```

```swift
@testable import PetWall
class TestPetViewController: XCTestCase {
}
```

Finding declarations

- Within a Swift target
- From Objective-C

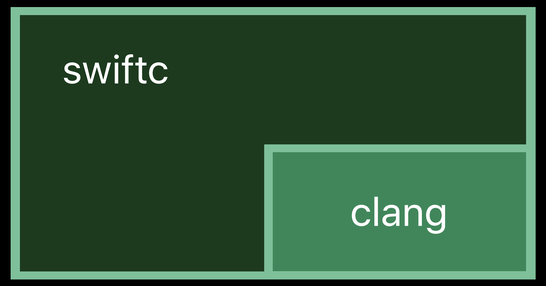
Generating interfaces

- To use in Objective-C
- To use in other Swift targets

Finding Declarations from Objective-C

```
import UIKit
class PetViewController: UIViewController {
  var view = PetView(name: "Fido", frame: frame)
  …
}
```

Compiler Uses Clang as a Library to Import Objective-C



Avoids messiness of a foreign-function interface

Can import clang frameworks directly

Objective-C Declarations Come from Headers

Objective-C Declarations Come from Headers

Imported Objective-C frameworks:

- From headers exposed in Clang module map

Objective-C Declarations Come from Headers

Imported Objective-C frameworks:

- From headers exposed in Clang module map

Within frameworks that mix Swift and Objective-C:

- From the umbrella header

Objective-C Declarations Come from Headers

Imported Objective-C frameworks:

- From headers exposed in Clang module map

Within frameworks that mix Swift and Objective-C:

- From the umbrella header

Within applications and unit test bundles:

- From target’s bridging header

Clang Importer Makes Methods More “Swifty”

Objective-C

```
- (BOOL)validateCuteness:(nonnull Pet *)pet
                   error:(NSError **)err;
```

Swift

```
do {
  try validateCuteness(fluffy)
} catch let e { … }
```

Clang Importer Makes Methods More “Swifty”

Objective-C

```
- (BOOL)validateCuteness:(nonnull Pet *)pet
                   error:(NSError **)err;
```

Swift

```
do {
  try validateCuteness(fluffy)
} catch let e { … }
```

Rename Methods Based on Part of Speech

Objective-C

```
- (void)drawPet:(nonnull Pet *)pet
        atPoint:(CGPoint)point;
```

Swift

```
draw(fluffy, at: origin)
```

Drops parameter type names following verbs and prepositions

Rename Methods Based on Part of Speech

Objective-C

```
- (void)drawPet:(nonnull Pet *)pet
        atPoint:(CGPoint)point;
```

Swift

```
draw(fluffy, at: origin)
```

Drops parameter type names following verbs and prepositions

Rename Methods Based on Part of Speech

Objective-C

```
- (void)drawPet:(nonnull Pet *)pet
        atPoint:(CGPoint)point;
```

Swift

```
draw(fluffy, at: origin)
```

Drops parameter type names following verbs and prepositions

Rename Methods Based on Part of Speech

Objective-C

```
- (void)drawPet:(nonnull Pet *)pet
        atPoint:(CGPoint)point;
```

Swift

```
draw(fluffy, at: origin)
```

Drops parameter type names following verbs and prepositions

```
// https://github.com/apple/swift/blob/master/lib/Basic/PartsOfSpeech.def
...
VERB(extend)
VERB(face)
VERB(fade)
VERB(fail)
VERB(fancy)
VERB(fasten)
VERB(fax)
VERB(fear)
VERB(feel)
VERB(fence)
VERB(fetch)
VERB(fight)
VERB(fill)
VERB(film)
...
```

… But It Is Just a List

Objective-C

```
- (void)feedPet:(nonnull Pet *)pet;
```

Swift

```
feedPet(fluffy)
```

Not renamed because ‘feed’ not in list

Use `NS_SWIFT_NAME` to change imported name explicitly

… But It Is Just a List

Objective-C

```
- (void)feedPet:(nonnull Pet *)pet
NS_SWIFT_NAME(feed(_:));
```

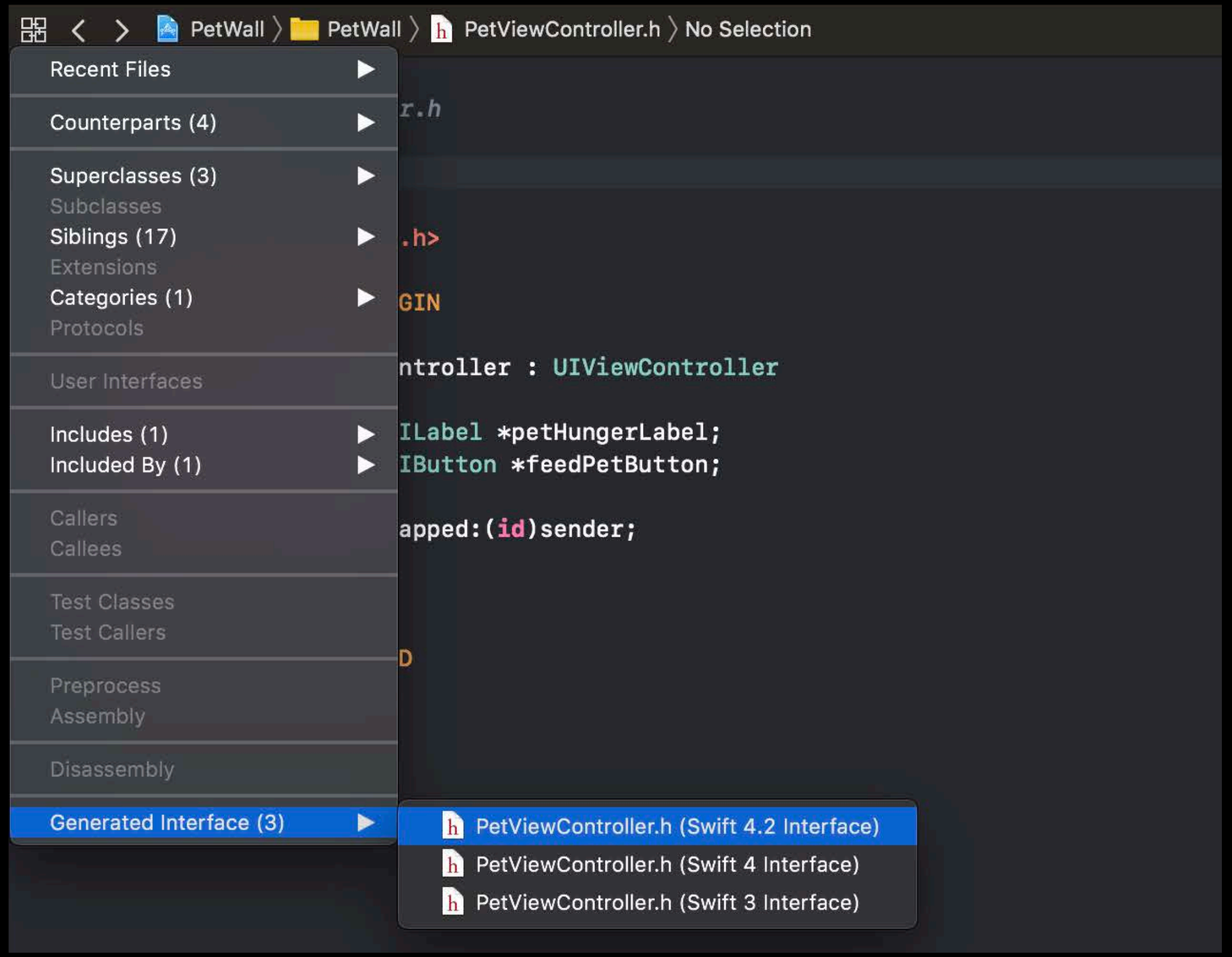
Swift

```
feed(fluffy)
```

Not renamed because ‘feed’ not in list

Use `NS_SWIFT_NAME` to change imported name explicitly

Xcode Shows Imported Swift Interface



```swift
import UIKit
class PetViewController: UIViewController {
  var view = PetView(name: "Fido", frame: frame)
  …
}
```

```objc
#import "PetWall-Swift.h"
@implementation AppDelegate
…
@end
```

```swift
@testable import PetWall
class TestPetViewController: XCTestCase {
}
```

Finding declarations

- Within a Swift target
- From Objective-C

Generating interfaces

- To use in Objective-C
- To use in other Swift targets

Generating Interfaces to Use in Objective-C

```
#import "PetWall-Swift.h"
```

Generating Interfaces to Use in Objective-C

```
#import "PetWall-Swift.h"
```

Generating Interfaces to Use in Objective-C

Write in Swift:

```
class PetCollar: NSObject {
  @objc
  var color: UIColor = .blue
  var name: String = "Fido"
}
```

```
#import "PetWall-Swift.h"
```

Generating Interfaces to Use in Objective-C

Write in Swift:

```
class PetCollar: NSObject {
  @objc
  var color: UIColor = .blue
  var name: String = "Fido"
}
```

Use from Objective-C:

```
#import "PetWall-Swift.h"

- (void)resetCollarColor {
   self.collar.color = [UIColor redColor];
}
```

Compiler Generates Objective-C Header

Swift

```
class PetCollar: NSObject {
  @objc
  var color: UIColor = .blue
  var name: String = "Fido"
}
```

Compiler Generates Objective-C Header

Swift

```
class PetCollar: NSObject {
  @objc
  var color: UIColor = .blue
  var name: String = "Fido"
}
```

Generated Objective-C Header

```
SWIFT_CLASS("_TtC7PetWall9PetCollar")
@interface PetCollar: NSObject
@property (nonatomic, strong) UIColor * _Nonnull color;
- (nonnull instancetype)init;
@end
```

For classes extending `NSObject` and methods/properties marked `@objc`

Compiler Generates Objective-C Header

Swift

```
class PetCollar: NSObject {
  @objc
  var color: UIColor = .blue
  var name: String = "Fido"
}
```

Generated Objective-C Header

```
SWIFT_CLASS("_TtC7PetWall9PetCollar")
@interface PetCollar: NSObject
@property (nonatomic, strong) UIColor * _Nonnull color;
- (nonnull instancetype)init;
@end
```

For classes extending `NSObject` and methods/properties marked `@objc`

Apps and unit tests: both public and internal declarations

Compiler Generates Objective-C Header

Swift

```
class PetCollar: NSObject {
  @objc
  var color: UIColor = .blue
  var name: String = "Fido"
}
```

Generated Objective-C Header

```
SWIFT_CLASS("_TtC7PetWall9PetCollar")
@interface PetCollar: NSObject
@property (nonatomic, strong) UIColor * _Nonnull color;
- (nonnull instancetype)init;
@end
```

For classes extending `NSObject` and methods/properties marked `@objc`

Apps and unit tests: both public and internal declarations

Frameworks: Only public declarations

Changing the Class Name in Objective-C

Swift

```
class PetCollar: NSObject {
  @objc
  var color: UIColor = .blue
  var name: String = "Fido"
}
```

Generated Objective-C Header

```
SWIFT_CLASS("_TtC7PetWall9PetCollar")
@interface PetCollar : NSObject
@property (nonatomic, strong) UIColor * _Nonnull color;
- (nonnull instancetype)init;
@end
```

Compiler ties Objective-C class to mangled Swift class name

Prevents conflict when two modules define class with same name

Changing the Class Name in Objective-C

Swift

```
class PetCollar: NSObject {
  @objc
  var color: UIColor = .blue
  var name: String = "Fido"
}
```

Generated Objective-C Header

```
SWIFT_CLASS("_TtC7PetWall9PetCollar")
@interface PetCollar : NSObject
@property (nonatomic, strong) UIColor * _Nonnull color;
- (nonnull instancetype)init;
@end
```

Compiler ties Objective-C class to mangled Swift class name

Prevents conflict when two modules define class with same name

Changing the Class Name in Objective-C

Swift

```
class PetCollar: NSObject {
  @objc
  var color: UIColor = .blue
  var name: String = "Fido"
}
```

Generated Objective-C Header

```
SWIFT_CLASS("_TtC7PetWall9PetCollar")
@interface PetCollar : NSObject
@property (nonatomic, strong) UIColor * _Nonnull color;
- (nonnull instancetype)init;
@end
```

Compiler ties Objective-C class to mangled Swift class name

Prevents conflict when two modules define class with same name

Use `@objc(Name)` to provide custom name — but must not conflict!

Changing the Class Name in Objective-C

Swift

```
@objc(PWLPetCollar)
class PetCollar: NSObject {
  @objc
  var color: UIColor = .blue
  var name: String = "Fido"
}
```

Generated Objective-C Header

```
SWIFT_CLASS_NAMED("PetCollar")
@interface PWLPetCollar : NSObject
@property (nonatomic, strong) UIColor * _Nonnull color;
- (nonnull instancetype)init;
@end
```

Compiler ties Objective-C class to mangled Swift class name

Prevents conflict when two modules define class with same name

Use `@objc(Name)` to provide custom name — but must not conflict!

```swift
import UIKit
class PetViewController: UIViewController {
  var view = PetView(name: "Fido", frame: frame)
  …
}
```

```objc
#import "PetWall-Swift.h"
@implementation AppDelegate
…
@end
```

```swift
@testable import PetWall
class TestPetViewController: XCTestCase {
}
```

Finding declarations

- Within a Swift target
- From Objective-C

Generating interfaces

- To use in Objective-C
- To use in other Swift targets

Module: A Distributable Unit of Declarations

```
@testable import PetWall
class TestPetViewController: XCTestCase {




}
```

Must first import other modules to see their declarations

Can import Clang modules

In Xcode each Swift target produces a separate module

Module: A Distributable Unit of Declarations

```
import XCTest
@testable import PetWall
class TestPetViewController: XCTestCase {
  func testInitialPet() {
    let controller = PetViewController()
    XCTAssertEqual(controller.view.name, "Fido")
  }
}
```

Must first import other modules to see their declarations

Can import Clang modules

In Xcode each Swift target produces a separate module

Module: A Distributable Unit of Declarations

```
import XCTest
@testable import PetWall
class TestPetViewController: XCTestCase {
  func testInitialPet() {
    let controller = PetViewController()
    XCTAssertEqual(controller.view.name, "Fido")
  }
}
```

Must first import other modules to see their declarations

Can import Clang modules

In Xcode each Swift target produces a separate module

Module: A Distributable Unit of Declarations

```
import XCTest
@testable import PetWall
class TestPetViewController: XCTestCase {
  func testInitialPet() {
    let controller = PetViewController()
    XCTAssertEqual(controller.view.name, "Fido")
  }
}
```

Must first import other modules to see their declarations

Can import Clang modules

In Xcode each Swift target produces a separate module

Importing Swift Declarations from Other Modules

```
import XCTest
@testable import PetWall
class TestPetViewController: XCTestCase {
  func testInitialPet() {
    let controller = PetViewController()
    XCTAssertEqual(controller.view.name, "Fido")
  }
}
```

Importing Swift Declarations from Other Modules

```
import XCTest
@testable import PetWall
class TestPetViewController: XCTestCase {
  func testInitialPet() {
    let controller = PetViewController()
    XCTAssertEqual(controller.view.name, "Fido")
  }
}
```

Deserialize to check types

PetWall.swiftmodule

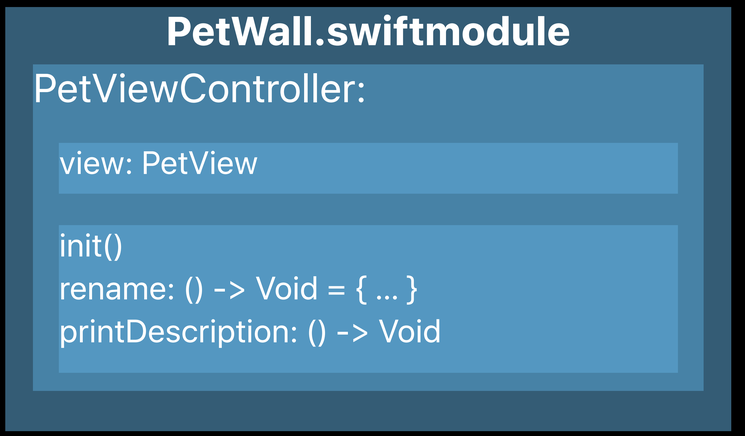
PetViewController:

view: PetView

init()
rename: () -> Void = { ... }
printDescription: () -> Void

Expose Declarations to Other Modules via .swiftmodule

```
public class PetViewController {
  public let view = PetView()

  @inlinable
  public func rename() {
    view.name = "Fido"
  }
  private func printDescription() { … }
}
```



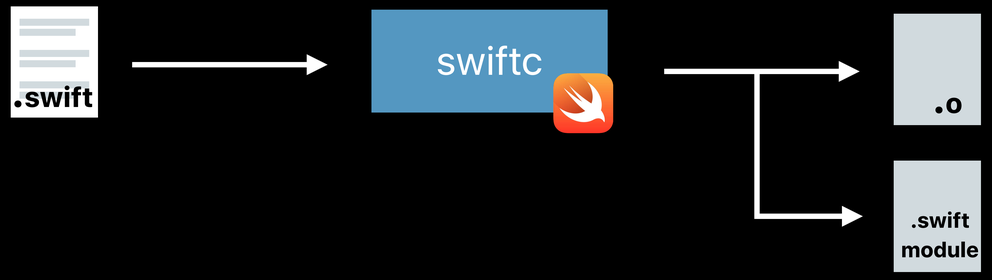
Serialized, binary representation of module's declarations

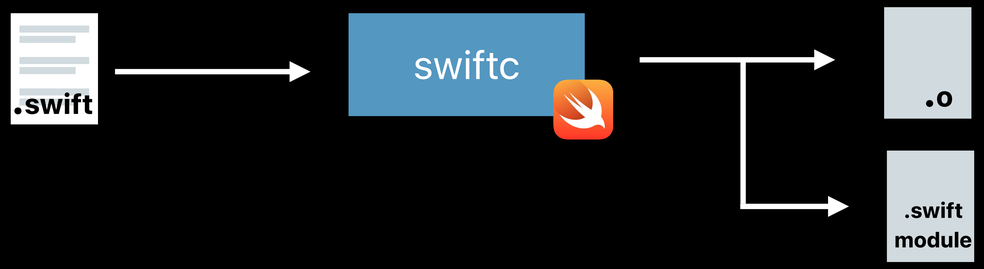
Includes bodies of inlinable functions

Includes private declarations for debugging

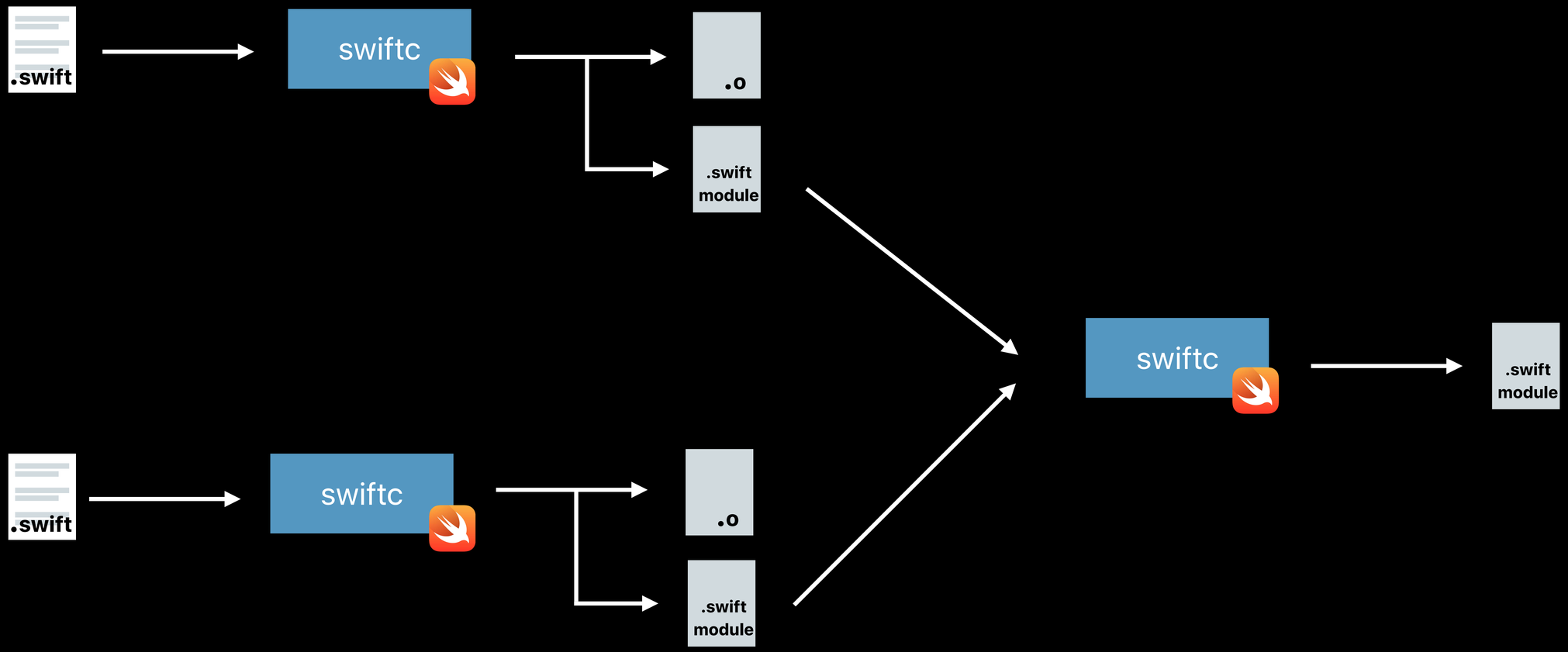
Produced During Compilation and Merged

Produced During Compilation and Merged

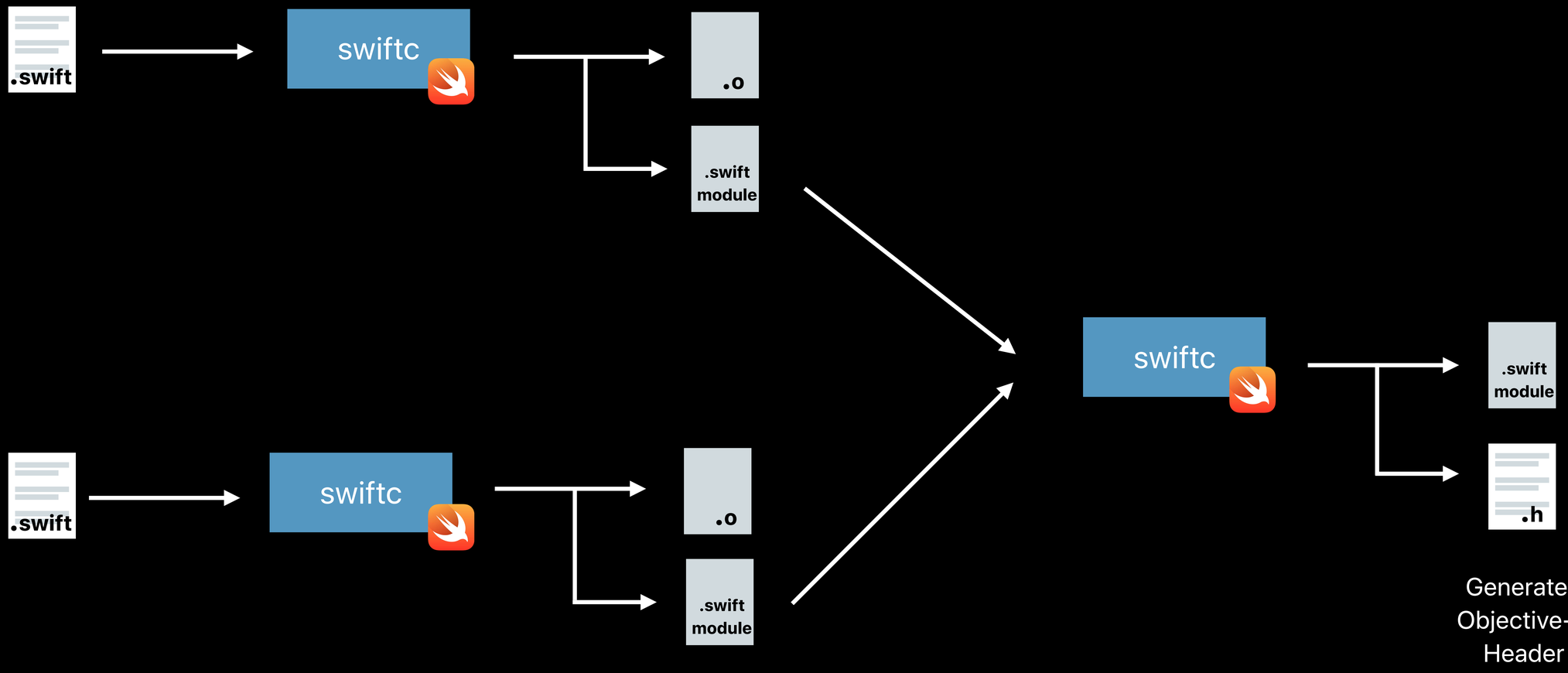




Produced During Compilation and Merged



Produced During Compilation and Merged



Behind the Scenes of the Xcode Build Process: Linking

Louis Gerbarg, Linker Team


10
1
0
001
100
01
100

What does the linker actually do?

What are symbols?

What are object files?

What are libraries?

The Linker

The Linker

Final task in building an executable Mach-O

The Linker

Final task in building an executable Mach-O

Combines the output of all compiler invocations into a single file

The Linker

Final task in building an executable Mach-O

Combines the output of all compiler invocations into a single file

- Moves and patches code generated by the compilers

The Linker

Final task in building an executable Mach-O

Combines the output of all compiler invocations into a single file

- Moves and patches code generated by the compilers

Takes two kinds of input files

The Linker

Final task in building an executable Mach-O

Combines the output of all compiler invocations into a single file

- Moves and patches code generated by the compilers

Takes two kinds of input files

- Object files (.o)

The Linker

Final task in building an executable Mach-O

Combines the output of all compiler invocations into a single file

- Moves and patches code generated by the compilers

Takes two kinds of input files

- Object files (.o)
- Libraries (.dylib, .tbd, .a)

Symbols

Symbols

A symbol is a name for a fragment of code or data

Symbols

A symbol is a name for a fragment of code or data

Fragments of code may reference other symbols

Symbols

A symbol is a name for a fragment of code or data

Fragments of code may reference other symbols

Symbols can have attributes on them that alter the linker’s behavior

Symbols

A symbol is a name for a fragment of code or data

Fragments of code may reference other symbols

Symbols can have attributes on them that alter the linker’s behavior

- Weak symbols

Symbols

A symbol is a name for a fragment of code or data

Fragments of code may reference other symbols

Symbols can have attributes on them that alter the linker’s behavior

- Weak symbols

Languages often encode data into a symbol “mangling” the symbol

Object Files

Object Files

Output of individual compiler actions

Object Files

Output of individual compiler actions

A non-executable Mach-O file containing code and data fragments

Object Files

Output of individual compiler actions

A non-executable Mach-O file containing code and data fragments

- Each fragment is represented by a symbol

Object Files

Output of individual compiler actions

A non-executable Mach-O file containing code and data fragments

- Each fragment is represented by a symbol
- Fragments may reference “undefined” symbols

Libraries

Libraries

Libraries define symbols that are not built as part of your target

Libraries

Libraries define symbols that are not built as part of your target

- Dylibs: Dynamic libraries

Libraries

Libraries define symbols that are not built as part of your target

- Dylibs: Dynamic libraries
  - Mach-O file that exposes code and data fragments executables can use

Libraries

Libraries define symbols that are not built as part of your target

- Dylibs: Dynamic libraries
  - Mach-O file that exposes code and data fragments executables can use
- TBDs: Text Based Dylib Stubs

Libraries

Libraries define symbols that are not built as part of your target

- Dylibs: Dynamic libraries
  - Mach-O file that exposes code and data fragments executables can use
- TBDs: Text Based Dylib Stubs
  - Only contains symbols

Libraries

Libraries define symbols that are not built as part of your target

- Dylibs: Dynamic libraries
  - Mach-O file that exposes code and data fragments executables can use
- TBDs: Text Based Dylib Stubs
  - Only contains symbols
- Static archives

Libraries

Libraries define symbols that are not built as part of your target

- Dylibs: Dynamic libraries
  - Mach-O file that exposes code and data fragments executables can use
- TBDs: Text Based Dylib Stubs
  - Only contains symbols
- Static archives
  - An archive of multiple .o files built with the “ar” tool

Libraries

Libraries define symbols that are not built as part of your target

- Dylibs: Dynamic libraries
  - Mach-O file that exposes code and data fragments executables can use
- TBDs: Text Based Dylib Stubs
  - Only contains symbols
- Static archives
  - An archive of multiple .o files built with the “ar” tool
  - Only the .o files with symbols you reference are included in your app

Example

Example

```
// Cat.mm

extern void playSound(const char * file);

static const char * purrFile = “purr.aac”;
@implementation Cat
- (void) purr {
    playSound(purrFile);
}
@end
```

Example

```
// Cat.mm

extern void playSound(const char * file);

static const char * purrFile = “purr.aac”;
@implementation Cat
- (void) purr {
    playSound(purrFile);
}
@end
```

Cat.o

```
l_.str:    .asciz "purr.aac"
"-[Cat purr]":
    adrp    x0, l_.str@PAGE
    add x0, x0, l_.str@PAGEOFF
    b   __Z9playSoundPKc
```

Example

```
// Cat.mm

extern void playSound(const char * file);

static const char * purrFile = “purr.aac”;
@implementation Cat
- (void) purr {
    playSound(purrFile);
}
@end
```

Cat.o

```
l_.str:    .asciz "purr.aac"
"-[Cat purr]":
    adrp   x0, l_.str@PAGE
    add x0, x0, l_.str@PAGEOFF
    b   __Z9playSoundPKc
```

Example

```
// Cat.mm

extern void playSound(const char * file);

static const char * purrFile = “purr.aac”;
@implementation Cat
- (void) purr {
    playSound(purrFile);
}
@end
```

Cat.o

```
l_.str:    .asciz "purr.aac"
"-[Cat purr]":
    adrp   x0, l_.str@PAGE
    add x0, x0, l_.str@PAGEOFF
    b   __Z9playSoundPKc
```

Example

```
// Cat.mm

extern void playSound(const char * file);

static const char * purrFile = “purr.aac”;
@implementation Cat
- (void) purr {
    playSound(purrFile);
}
@end
```

Cat.o

```
l_.str:    .asciz "purr.aac"
"-[Cat purr]":
    adrp    x0, l_.str@PAGE
    add x0, x0, l_.str@PAGEOFF
    b   __Z9playSoundPKc
```

Example

```
// Cat.mm

extern void playSound(const char * file);

static const char * purrFile = “purr.aac”;
@implementation Cat
- (void) purr {
    playSound(purrFile);
}
@end
```

Cat.o

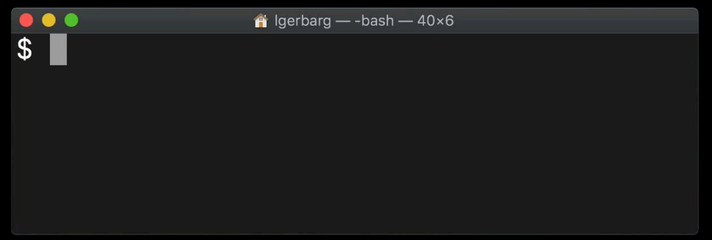
```
l_.str:    .asciz "purr.aac"
"-[Cat purr]":
    adrp    x0, l_.str@PAGE
    add x0, x0, l_.str@PAGEOFF
    b   __Z9playSoundPKc
```

Example

```
// Cat.mm

extern void playSound(const char * file);

static const char * purrFile = “purr.aac”;
@implementation Cat
- (void) purr {
    playSound(purrFile);
}
@end
```

Cat.o

```
l_.str:    .asciz "purr.aac"
"-[Cat purr]":
    adrp    x0, l_.str@PAGE
    add x0, x0, l_.str@PAGEOFF
    b   __Z9playSoundPKc
```



Example

Cat.o

```
l_.str:    .asciz "purr.aac"
-[Cat purr]:
  adrp x0, l_.str@PAGE
  add  x0, x0, l_.str@PAGEOFF
  b    __Z9playSoundPKc
```

Example

Cat.o

```
l_.str:    .asciz "purr.aac"
-[Cat purr]:
  adrp x0, l_.str@PAGE
  add  x0, x0, l_.str@PAGEOFF
  b    __Z9playSoundPKc
```

PetKit

Example

Cat.o

```
l_.str:    .asciz "purr.aac"
-[Cat purr]:
  adrp x0, l_.str@PAGE
  add  x0, x0, l_.str@PAGEOFF
  b    __Z9playSoundPKc
```

Example

Cat.o

```
l_.str:    .asciz "purr.aac"
-[Cat purr]:
  adrp x0, l_.str@PAGE
  add  x0, x0, l_.str@PAGEOFF
  b    __Z9playSoundPKc
```

PetKit

__TEXT

```
-[Cat purr]:
  adrp x0, l_.str@PAGE
  add  x0, x0, l_.str@PAGEOFF
  b    __Z9playSoundPKc
```

```
__cstring:
   .asciz    "purr.aac"
```

Example

Cat.o

```
l_.str:    .asciz "purr.aac"
-[Cat purr]:
  adrp x0, l_.str@PAGE
  add  x0, x0, l_.str@PAGEOFF
  b    __Z9playSoundPKc
```

PetKit

__TEXT

```
-[Cat purr]:
  adr  x0, #0x6f4
  nop
  b    __Z9playSoundPKc
```

```
__cstring:
   .asciz    "purr.aac"
```

Example

Cat.o

```
l_.str:    .asciz "purr.aac"
-[Cat purr]:
  adrp x0, l_.str@PAGE
  add  x0, x0, l_.str@PAGEOFF
  b    __Z9playSoundPKc
```

PetKit

__TEXT

```
-[Cat purr]:
  adr  x0, #0x6f4
  nop
  b    __Z9playSoundPKc
```

```
__cstring:
   .asciz    "purr.aac"
```

Example

Cat.o

```
l_.str:    .asciz "purr.aac"
-[Cat purr]:
  adrp x0, l_.str@PAGE
  add  x0, x0, l_.str@PAGEOFF
  b    __Z9playSoundPKc
```

PetKit

__TEXT

```
-[Cat purr]:
  adr  x0, #0x6f4
  nop
  b    __Z9playSoundPKc
```

```
__cstring:
   .asciz    "purr.aac"
```

Example

Cat.o

```
l_.str:    .asciz "purr.aac"
-[Cat purr]:
  adrp x0, l_.str@PAGE
  add  x0, x0, l_.str@PAGEOFF
  b    __Z9playSoundPKc
```

PetSupport.a

PetKit

__TEXT

```
-[Cat purr]:
  adr  x0, #0x6f4
  nop
  b    __Z9playSoundPKc
```

```
__cstring:
   .asciz    "purr.aac"
```

Example

Cat.o

```
l_.str:    .asciz "purr.aac"
-[Cat purr]:
  adrp x0, l_.str@PAGE
  add  x0, x0, l_.str@PAGEOFF
  b    __Z9playSoundPKc
```

PetSupport.a

PetSounds.o

```
__Z9playSoundPKc:
  …
   bl _open
```

PetCare.o

```
_callVet:
   …
```

PetKit

__TEXT

```
-[Cat purr]:
  adr  x0, #0x6f4
  nop
  b    __Z9playSoundPKc
```

```
__cstring:
   .asciz    "purr.aac"
```

Example

Cat.o

```
l_.str:    .asciz "purr.aac"
-[Cat purr]:
  adrp x0, l_.str@PAGE
  add  x0, x0, l_.str@PAGEOFF
  b    __Z9playSoundPKc
```

PetSupport.a

PetSounds.o

```
__Z9playSoundPKc:
  …
   bl _open
```

PetCare.o

```
_callVet:
   …
```

PetKit

__TEXT

```
-[Cat purr]:
  adr  x0, #0x6f4
  nop
  b    __Z9playSoundPKc
```

```
__cstring:
   .asciz    "purr.aac"
```

Example

Cat.o

```
l_.str:    .asciz "purr.aac"
-[Cat purr]:
  adrp x0, l_.str@PAGE
  add  x0, x0, l_.str@PAGEOFF
  b    __Z9playSoundPKc
```

PetSupport.a

PetSounds.o

```
__Z9playSoundPKc:
  …
   bl _open
```

PetCare.o

```
_callVet:
   …
```

PetKit

__TEXT

```
-[Cat purr]:
  adr  x0, #0x6f4
  nop
  b    __Z9playSoundPKc
```

```
__Z9playSoundPKc:
  …
   bl _open$stub
```

```
__cstring:
   .asciz    "purr.aac"
```

Example

Cat.o

```
l_.str:    .asciz "purr.aac"
-[Cat purr]:
  adrp x0, l_.str@PAGE
  add  x0, x0, l_.str@PAGEOFF
  b    __Z9playSoundPKc
```

PetSupport.a

PetSounds.o

```
__Z9playSoundPKc:
  …
   bl _open
```

PetCare.o

```
_callVet:
   …
```

PetKit

__TEXT

```
-[Cat purr]:
  adr  x0, #0x6f4
  nop
  b    __Z9playSoundPKc
```

```
__Z9playSoundPKc:
  …
   bl _open$stub
```

```
__cstring:
   .asciz    "purr.aac"
```

Example

Cat.o

```
l_.str:    .asciz "purr.aac"
-[Cat purr]:
  adrp x0, l_.str@PAGE
  add  x0, x0, l_.str@PAGEOFF
  b    __Z9playSoundPKc
```

PetSupport.a

PetSounds.o

```
__Z9playSoundPKc:
  …
   bl _open
```

PetCare.o

```
_callVet:
   …
```

PetKit

__TEXT

```
-[Cat purr]:
  adr  x0, #0x6f4
  nop
  b    __Z9playSoundPKc
```

```
__Z9playSoundPKc:
  …
   bl _open$stub
```

```
__cstring:
   .asciz    "purr.aac"
```

Example

Cat.o

```
l_.str:    .asciz "purr.aac"
-[Cat purr]:
  adrp x0, l_.str@PAGE
  add  x0, x0, l_.str@PAGEOFF
  b    __Z9playSoundPKc
```

PetSupport.a

PetSounds.o

```
__Z9playSoundPKc:
  …
   bl _open
```

PetCare.o

```
_callVet:
   …
```

PetKit

__TEXT

```
-[Cat purr]:
  adr  x0, #0x6f4
  nop
  b    __Z9playSoundPKc
```

```
__Z9playSoundPKc:
  …
   bl _open$stub
```

```
__cstring:
   .asciz    "purr.aac"
```

Example

Cat.o

```
l_.str:    .asciz "purr.aac"
-[Cat purr]:
  adrp x0, l_.str@PAGE
  add  x0, x0, l_.str@PAGEOFF
  b    __Z9playSoundPKc
```

PetSupport.a

PetSounds.o

```
__Z9playSoundPKc:
  …
   bl _open
```

PetCare.o

```
_callVet:
   …
```

libSystem.tbd

```
---
archs:           [ arm64 ]
platform:        ios
install-name:    libSystem.B.dylib
exports:
  - archs:       [ arm64 ]
  - symbols:     [ ___open ]
...
```

PetKit

__TEXT

```
-[Cat purr]:
  adr  x0, #0x6f4
  nop
  b    __Z9playSoundPKc
```

```
__Z9playSoundPKc:
  …
   bl _open$stub
```

```
__cstring:
   .asciz    "purr.aac"
```

Example

Cat.o

```
l_.str:    .asciz "purr.aac"
-[Cat purr]:
  adrp x0, l_.str@PAGE
  add  x0, x0, l_.str@PAGEOFF
  b    __Z9playSoundPKc
```

PetSupport.a

PetSounds.o

```
__Z9playSoundPKc:
  …
   bl _open
```

PetCare.o

```
_callVet:
   …
```

libSystem.tbd

```
---
archs:           [ arm64 ]
platform:        ios
install-name:    libSystem.B.dylib
exports:
  - archs:       [ arm64 ]
  - symbols:     [ ___open ]
...
```

PetKit

__TEXT

```
-[Cat purr]:
  adr   x0, #0x6f4
  nop
  b     __Z9playSoundPKc
```

```
__Z9playSoundPKc:
  …
   bl _open$stub
```

```
_open$stub:
  ldr  x16, _open$ptr
  br x16
```

```
__cstring:
   .asciz    "purr.aac"
```

Example

Cat.o

```
l_.str:    .asciz "purr.aac"
-[Cat purr]:
  adrp x0, l_.str@PAGE
  add  x0, x0, l_.str@PAGEOFF
  b    __Z9playSoundPKc
```

PetSupport.a

PetSounds.o

```
__Z9playSoundPKc:
  …
   bl _open
```

PetCare.o

```
_callVet:
   …
```

libSystem.tbd

```
---
archs:           [ arm64 ]
platform:        ios
install-name:    libSystem.B.dylib
exports:
  - archs:       [ arm64 ]
  - symbols:     [ ___open ]
...
```

PetKit

__TEXT

```
-[Cat purr]:
  adr   x0, #0x6f4
  nop
  b     __Z9playSoundPKc
```

```
__Z9playSoundPKc:
  …
   bl _open$stub
```

```
_open$stub:
  ldr  x16, _open$ptr
  br x16
```

```
__cstring:
   .asciz     "purr.aac"
```

Example

Cat.o

```
l_.str:    .asciz "purr.aac"
-[Cat purr]:
  adrp x0, l_.str@PAGE
  add  x0, x0, l_.str@PAGEOFF
  b    __Z9playSoundPKc
```

PetSupport.a

PetSounds.o

```
__Z9playSoundPKc:
  …
   bl _open
```

PetCare.o

```
_callVet:
   …
```

libSystem.tbd

```
---
archs:           [ arm64 ]
platform:        ios
install-name: libSystem.B.dylib
exports:
  - archs:     [ arm64 ]
  - symbols:   [ ___open ]
...
```

PetKit

__TEXT

```
-[Cat purr]:
  adr   x0, #0x6f4
  nop
  b     __Z9playSoundPKc
```

```
__Z9playSoundPKc:
  …
   bl _open$stub
```

```
_open$stub:
  ldr  x16, _open$ptr
  br x16
```

```
__cstring:
   .asciz    "purr.aac"
```

Example

Cat.o

```
l_.str:    .asciz "purr.aac"
-[Cat purr]:
  adrp x0, l_.str@PAGE
  add  x0, x0, l_.str@PAGEOFF
  b    __Z9playSoundPKc
```

PetSupport.a

PetSounds.o

```
__Z9playSoundPKc:
  …
   bl _open
```

PetCare.o

```
_callVet:
   …
```

libSystem.tbd

```
---
archs:           [ arm64 ]
platform:        ios
install-name:    libSystem.B.dylib
exports:
  - archs:       [ arm64 ]
  - symbols:     [ ___open ]
...
```

PetKit

__TEXT

```
-[Cat purr]:
  adr   x0, #0x6f4
  nop
  b     __Z9playSoundPKc
```

```
__Z9playSoundPKc:
  …
   bl _open$stub
```

```
_open$stub:
  ldr  x16, _open$ptr
  br x16
```

```
__cstring:
   .asciz    "purr.aac"
```

__DATA

Pointers

```
_open$ptr: .long 0x0
```

Example

Cat.o

```
l_.str:    .asciz "purr.aac"
-[Cat purr]:
  adrp x0, l_.str@PAGE
  add  x0, x0, l_.str@PAGEOFF
  b    __Z9playSoundPKc
```

PetSupport.a

PetSounds.o

```
__Z9playSoundPKc:
  …
   bl _open
```

PetCare.o

```
_callVet:
   …
```

libSystem.tbd

```
---
archs:           [ arm64 ]
platform:        ios
install-name:    libSystem.B.dylib
exports:
  - archs:       [ arm64 ]
  - symbols:     [ ___open ]
...
```

PetKit

__TEXT

```
-[Cat purr]:
  adr   x0, #0x6f4
  nop
  b     __Z9playSoundPKc
```

```
__Z9playSoundPKc:
  …
   bl _open$stub
```

```
_open$stub:
 ldr  x16, _open$ptr
 br x16
```

```
__cstring:
   .asciz    "purr.aac"
```

__DATA

Pointers

```
_open$ptr: .long 0x0
```

Example

Cat.o

```
l_.str:    .asciz "purr.aac"
-[Cat purr]:
  adrp x0, l_.str@PAGE
  add  x0, x0, l_.str@PAGEOFF
  b    __Z9playSoundPKc
```

PetSupport.a

PetSounds.o

```
__Z9playSoundPKc:
  …
   bl _open
```

PetCare.o

```
_callVet:
   …
```

libSystem.tbd

```
---
archs:           [ arm64 ]
platform:        ios
install-name: libSystem.B.dylib
exports:
  - archs:     [ arm64 ]
  - symbols:   [ ___open ]
...
```

PetKit

__TEXT

```
-[Cat purr]:
  adr   x0, #0x6f4
  nop
  b     __Z9playSoundPKc
```

```
__Z9playSoundPKc:
  …
   bl _open$stub
```

```
_open$stub:
 ldr  x16, _open$ptr
 br x16
```

```
__cstring:
   .asciz    "purr.aac"
```

__DATA

Pointers

```
_open$ptr: .long 0x0
```

__LINKEDIT

```
Bind _open$ptr ->
   libSystem`_open
```

Example

Cat.o

```
l_.str:    .asciz "purr.aac"
-[Cat purr]:
  adrp x0, l_.str@PAGE
  add  x0, x0, l_.str@PAGEOFF
  b    __Z9playSoundPKc
```

PetSupport.a

PetSounds.o

```
__Z9playSoundPKc:
  …
   bl _open
```

PetCare.o

```
_callVet:
   …
```

libSystem.tbd

```
---
archs:           [ arm64 ]
platform:        ios
install-name:    libSystem.B.dylib
exports:
  - archs:       [ arm64 ]
  - symbols:     [ ___open ]
...
```

PetKit

__TEXT

```
-[Cat purr]:
  adr  x0, #0x6f4
  nop
  b    __Z9playSoundPKc
```

```
__Z9playSoundPKc:
  …
   bl _open$stub
```

```
_open$stub:
 ldr  x16, _open$ptr
 br x16
```

```
__cstring:
   .asciz    "purr.aac"
```

__DATA

Pointers

```
_open$ptr: .long 0x0
```

__LINKEDIT

```
Bind _open$ptr ->
   libSystem`_open
```

Example

Cat.o

```
l_.str:    .asciz "purr.aac"
-[Cat purr]:
  adrp x0, l_.str@PAGE
  add  x0, x0, l_.str@PAGEOFF
  b    __Z9playSoundPKc
```

PetSupport.a

PetSounds.o

```
__Z9playSoundPKc:
  …
   bl _open
```

PetCare.o

```
_callVet:
   …
```

libSystem.tbd

```
---
archs:          [ arm64 ]
platform:       ios
install-name:   libSystem.B.dylib
exports:
  - archs:      [ arm64 ]
  - symbols:    [ ___open ]
...
```

PetKit

__TEXT

```
-[Cat purr]:
  adr  x0, #0x6f4
  nop
  b    __Z9playSoundPKc

__Z9playSoundPKc:
  …
   bl _open$stub

_open$stub:
 ldr  x16, _open$ptr
 br x16

__cstring:
   .asciz    "purr.aac"
```

__DATA

Pointers

```
_open$ptr: .long 0x0
```

__LINKEDIT

```
Bind _open$ptr ->
```


10
1
0
001
100
01
100


10
1
0
001
100
01
100


10
1
0
001
100
01
100


10
1
0
001
100
01
100


10
1
0
001
100
01
100

Open Source

Swift

- https://swift.org/

Clang

- https://www.llvm.org/

llbuild

- https://github.com/apple/swift-llbuild/

More Information

https://developer.apple.com/wwdc18/415


WWDC18