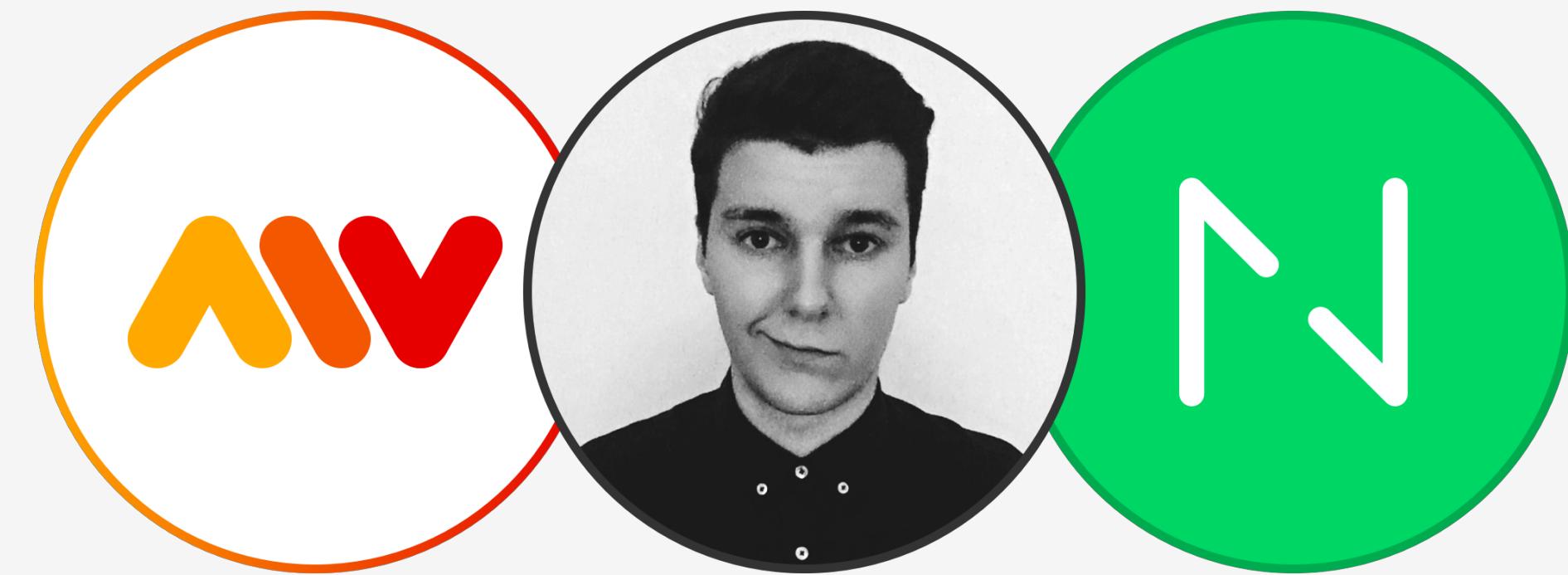


# Setting up projects with xcconfigs

Adrian Kashivskyy, Mobile Warsaw, February 2019



# Agenda

## 1. Build settings

What they are and what's the problem.

## 2. Xccconfigs

Syntax, usage and examples.

## 3. Shameless plug

Showcase of netguru/xccconfigs.

# Build settings

*"A build setting is a variable that determines how build tasks are performed. Xcode uses build settings to specify aspects of the build process followed to generate a product."*

— *Xcode Build System Guide*

## ▼ Build Options

Setting	 PJATK
Debug Information Format	DWARF with dSYM File ◊
Precompiled Header Uses Files From Build Directory	Yes ◊

## ▼ Deployment

Setting	 PJATK
Deployment Postprocessing	No ◊
Strip Style	All Symbols ◊
Targeted Device Family	◊

## ▼ Linking

Setting	 PJATK
Mach-O Type	◊

## ▼ Packaging

Setting	 PJATK
Create Info.plist Section in Binary	No ◊
Expand Build Settings in Info.plist File	Yes ◊
Force Package Info Generation	No ◊
Info.plist File	/Users/Adrian/Projects/pjatk/pjatk-app-ios/PJATK/Resources/-Info.plist
Info.plist Other Preprocessor Flags	
Info.plist Output Encoding	same-as-input ◊
Info.plist Preprocessor Definitions	
Info.plist Preprocessor Prefix File	
Preprocess Info.plist File	No ◊
Preserve HFS Data	No ◊
Product Bundle Identifier	

```
) ;  
    name = MainMenu.xib;  
    sourceTree = "<group>";  
};  
/* End PBXVariantGroup section */  
  
/* Begin XCBuildConfiguration section */  
3A9E302F2077F2DB001D645B /* Debug */ = {  
    isa = XCBuildConfiguration;  
    buildSettings = {  
        ALWAYS_SEARCH_USER_PATHS = NO;  
        CLANG_ANALYZER_NONNULL = YES;  
        CLANG_ANALYZER_NUMBER_OBJECT_CONVERSION = YES_AGGRESSIVE;  
        CLANG_CXX_LANGUAGE_STANDARD = "gnu++14";  
        CLANG_CXX_LIBRARY = "libc++";  
        CLANG_ENABLE_MODULES = YES;  
        CLANG_ENABLE_OBJC_ARC = YES;  
        CLANG_ENABLE_OBJC_WEAK = YES;  
        CLANG_WARN_BLOCK_CAPTURE_AUTORELEASING = YES;
```

# Why bother?

- **Prevent accidental modification**  
It might set unwanted or wrongly inferred build settings as a result of some action.
- **It's hard to review pbxproj files**  
How many of you actually look at diffs in this file?
- **More powerful features**  
Imports, inheritance, composition, conditions.



# xcconfigs

*"Xcode lets you define build setting in external, plaintext files, called xcconfigs"*

— Me

# Syntax

- **Include**

```
#include "path/to/other.xcconfig"
```

- **Assignment**

```
GREETING = hello
```

- **Interpolation**

```
PHRASE = $(GREETING), mobile warsaw
```

# Syntax

- **Inheritance** – *more on that later on*

```
PATHS = /usr/include $(inherited)
```

- **Conditional assignment**

```
XYZZY[os=iphone*] = sucks
```

```
XYZZY[os=macosx*] = rules
```

```
TRUTH = AppKit $(XYZZY)
```

# Example

```
#include "General.xcconfig"

PRODUCT_NAME = MobileWarsawTests
INFOPLIST_PATH = $(PROJECT_DIR)/Resources/$(PRODUCT_NAME)-Info.plist

DEVELOPMENT_TEAM = 1A2B3C4D5
CODESIGN_IDENTITY = iPhone Developer

SWIFT_TREAT_WARNINGS_AS_ERRORS = YES
FRAMEWORK_SEARCH_PATHS = $(inherited) $(PROJECT_DIR)/Carthage/Build/iOS
```

# Advantages

- **Easier code review**  
No clutter and boilerplate, all edits are intentional.
- **Better structure and composition**  
You can organize them as you want, not as Xcode wants.
- **Reusability**  
Decompose common settings into separate universal files.

# Advantages

- **Documentation**  
You can document reasons for setting certain build settings.
- **Ease of use**  
While most features are available in the visual editor, it's much simpler to use them in xcconfigs.

# Disadvantages

- **Multiple sources of truth**

There are multiple places build settings live in.

- **Manual maintenance**

You need to keep track of new build settings and recommended values.

# Solution?

# Libraries!

# Libraries!

[netguru/xccconfigs](https://github.com/netguru/xccconfigs)

## XcconfigsDemo



Info

Build Settings

## PROJECT

Basic Customized All Combined Levels +



## XcconfigsDemo

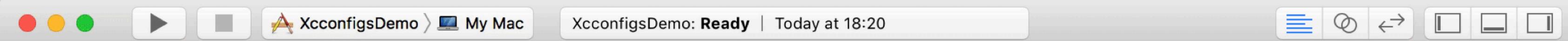
## TARGETS

## XcconfigsDemo

▼ Architectures		Resolved	XcconfigsDemo	iOS Default
Setting				
Additional SDKs				
Architectures		Standard architectures (armv7, ▾)		Standard architectures (armv7, ▾)
Base SDK		Latest iOS (iOS 10.3) ▾	Latest iOS (iOS 10.3) ▾	Latest iOS (iOS 10.3) ▾
▼ Build Active Architecture Only		<Multiple values> ▾	<Multiple values> ▾	No ▾
Debug		Yes ▾	Yes ▾	No ▾
Release		No ▾		No ▾
Supported Platforms		iOS ▾		iOS ▾
Valid Architectures		arm64 armv7 armv7s		arm64 armv7 armv7s

▼ Assets		Resolved	XcconfigsDemo	iOS Default
Setting				
Asset Pack Manifest URL Prefix				
Embed Asset Packs In Product Bundle		No ▾		No ▾
Enable On Demand Resources		No ▾		No ▾
On Demand Resources Initial Install Tags				
On Demand Resources Prefetch Order				

▼ Build Locations		Resolved	XcconfigsDemo	iOS Default
Setting				
Build Products Path		build		build
Intermediate Build Files Path		build		build
▼ Per-configuration Build Products Path		<Multiple values>		build
Debug		build/Debug-iphoneos		build



XcconfigsDemo

Info Build Settings

PROJECT

XcconfigsDemo

TARGETS

XcconfigsDemo

XcconfigsTest

Deployment Target

macOS Deployment Target Default

iOS Deployment Target Default

Configurations

Name	Based on Configuration File
------	-----------------------------

Debug

↳ XcconfigsDemo	None ▾
↳ XcconfigsDemo	None ▾
↳ XcconfigsTest	None ▾

Release

↳ XcconfigsDemo	None ▾
↳ XcconfigsDemo	None ▾
↳ XcconfigsTest	None ▾

+ -

Use  for command-line builds

Localizations

Language	Resources
----------	-----------

English — Development Language	2 Files Localized
--------------------------------	-------------------

+ - Filter



XcconfigsDemo

Info

Build Settings

PROJECT

XcconfigsDemo

TARGETS

XcconfigsDemo

XcconfigsTest

Deployment Target

iOS Deployment Target

Default

Configurations

Name

Based on Configuration File

Debug

3 Configurations Set

- ↳ XcconfigsDemo
- ↳ XcconfigsDemo
- ↳ XcconfigsTest

Debug ▾

Application ▾

Tests ▾

Release

3 Configurations Set

- ↳ XcconfigsDemo
- ↳ XcconfigsDemo
- ↳ XcconfigsTest

Release ▾

Application ▾

Tests ▾

+

Use

Release



for command-line builds

Localizations

Language

Resources

English — Development Language

2 Files Localized

+

-

Use Base Internationalization



# Structure

- Per configuration  
Debug and release.
- Per platform  
macOS, iOS, tvOS and watchOS.
- Per target type  
Application, extension, framework and test bundle.

# Showcase

# Further reading

- [The Unofficial Guide to xcconfig files](#)

Samantha Demi

- [Official documentation](#)

Apple

- [jspahrsummers/xcconfigs](#)

Justin Spahr-Summers

# Thank you!

GitHub: [akashivskyy/talks](https://github.com/akashivskyy/talks)

Twitter: [@akashivskyy](https://twitter.com/akashivskyy)