

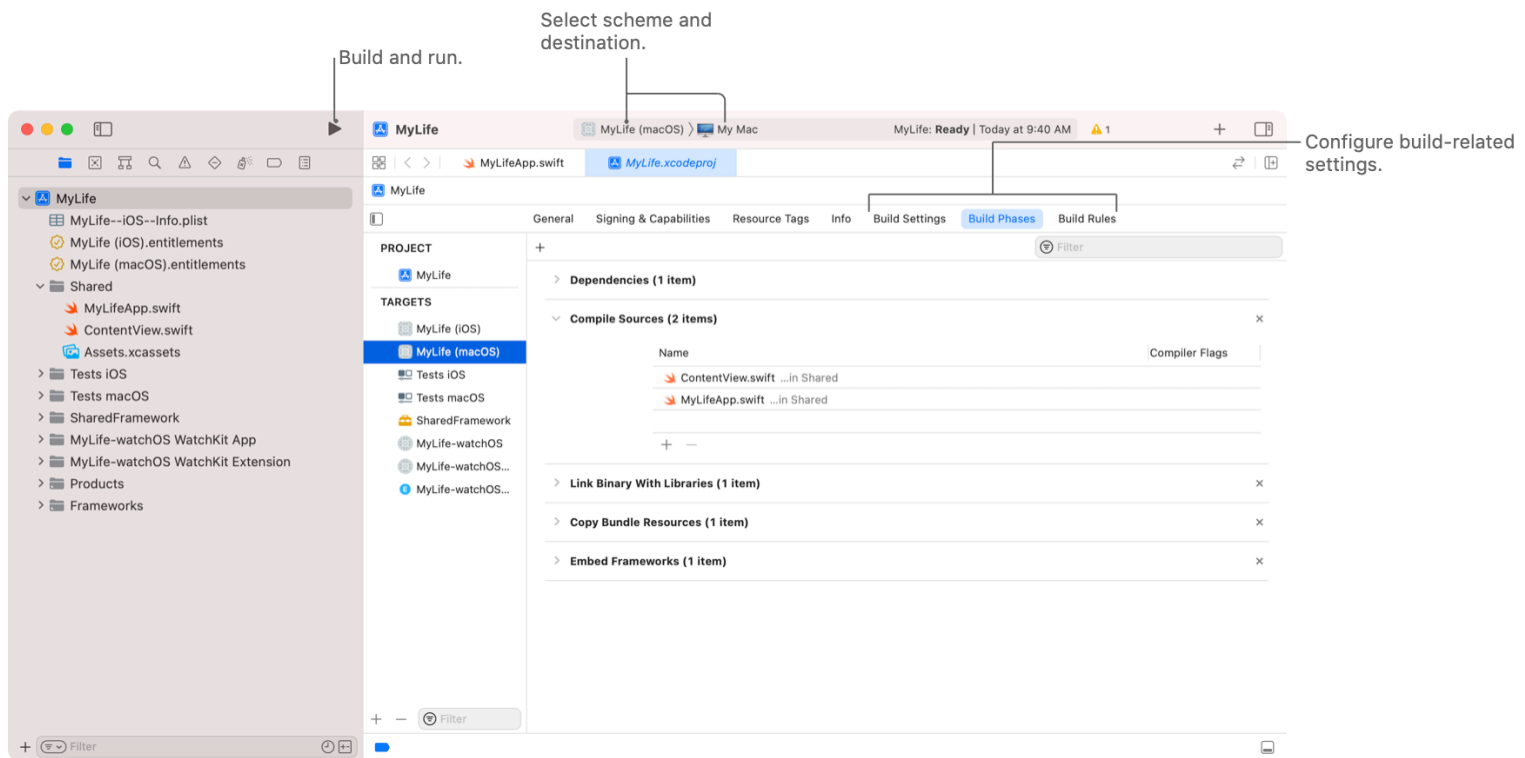
[Xcode](#) / Build system

Build system

Compile your code into a binary format, and customize your project settings to build your code.

Overview

The Xcode build system manages the tools that transform your code and resource files into a finished app. When you tell Xcode to build your project, the build system analyzes your files and uses your project settings to assemble the set of tasks to perform. Use your project settings to modify the build process and add tasks that you need to complete your builds.



Topics

Essentials

Configuring a new target in your project

Configure your project to build a new product, and add the code and resources the product requires.

Configuring a multiplatform app

Share project settings and code across platforms in a single app target.

Build settings

Configuring the build settings of a target

Specify the options you use to compile, link, and produce a product from a target, and identify settings inherited from your project or the system.

Adding a build configuration file to your project

Specify your project's build settings in plain-text files, and supply different settings for debug and release builds.

Build settings reference

A detailed list of individual Xcode build settings that control or change the way a target is built.

Identifying and addressing framework module issues

Detect and fix common problems found in framework modules with the module verifier.

Understanding build product layout changes in Xcode

Build customization

Customizing the build schemes for a project

Specify which targets to build, and customize the settings Xcode uses to build, run, test, and profile those targets.

Customizing the build phases of a target

Specify the tasks to perform during a build, including the source files to compile, the scripts to run, and the resources to include in the final product.

Creating build rules for custom file types

Tell Xcode how to build your project's custom file types, and provide dependency information to optimize the build process for each file.

Running custom scripts during a build

Execute custom shell scripts during the build process, and run tools or other commands that your project requires.

Running code on a specific platform or OS version

Add conditional compilation markers around code that requires a particular family of devices or minimum operating system version to run.

Performance

Configuring your project to use mergeable libraries

Use mergeable dynamic libraries to get app launch times similar to static linking in release builds, without losing dynamically linked build times in debug builds.

Improving the speed of incremental builds

Tell the Xcode build system about your project's target-related dependencies, and reduce the compiler workload during each build cycle.

Improving build efficiency with good coding practices

Shorten compile times by reducing the number of symbols your code exports and by giving the compiler the explicit information it needs.

Building your project with explicit module dependencies

Reduce compile times by eliminating unnecessary module variants using the Xcode build system.

Security and privacy

Verifying the origin of your XCFrameworks

Discover who signed a framework, and take action when that changes.

Enabling enhanced security for your app

Detect out-of-bounds memory access, use of freed memory, and other potential vulnerabilities.

Creating enhanced security helper extensions

Reduce opportunities for an attacker to target your app through its extensions.



Adopting type-aware memory allocation

Reduce the opportunities to treat pointers as data in your code.



Conforming to Mach IPC security restrictions

Avoid crashes and potentially insecure situations associated with Mach messages.

See Also

Xcode IDE



Projects and workspaces

Manage the code and resources you use to build apps, libraries, and other software for Apple platforms.



Source control management

Back up your files, collaborate with others, and tag your releases with Git source control support in Xcode.



Capabilities

Enable services that Apple provides, such as In-App Purchase, Push Notifications, Apple Pay, iCloud, and many others.



Command-line tools

Develop and customize your projects in Terminal.