Mergeable Libraries in Xcode 15

Sanju Naik, devX Gojek

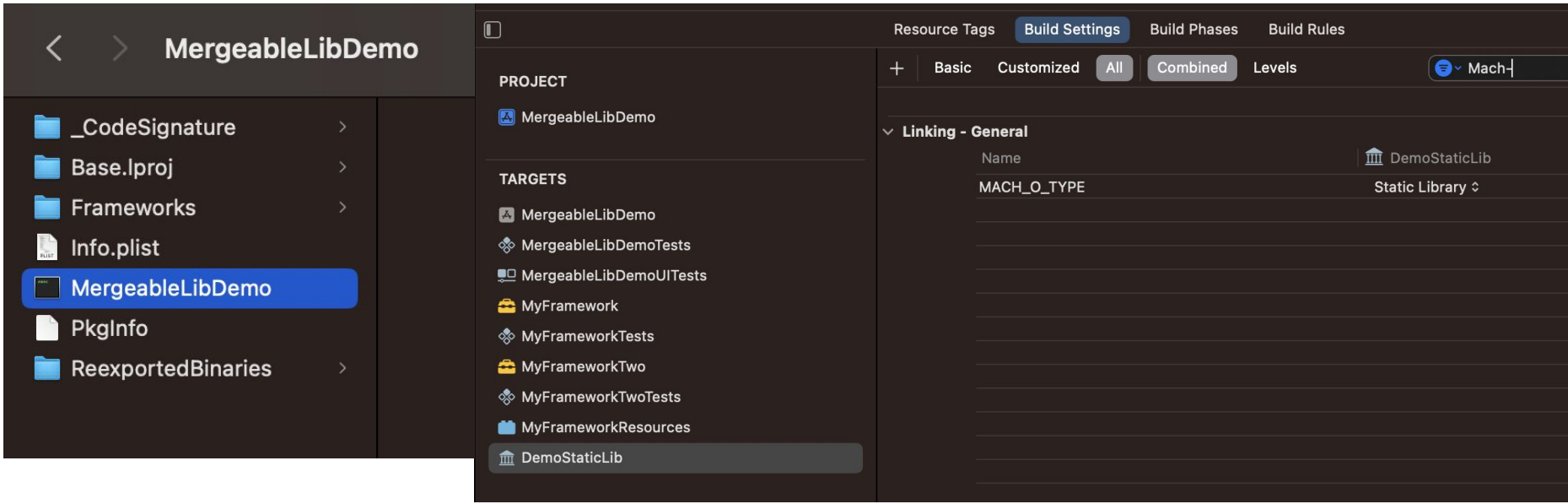
Agenda

- Static Libraries vs Dynamic Frameworks
- Pros and Cons of Static Libs vs Dynamic Frameworks
- Mergeable libraries - Xcode 15
  - Automatic merging
  - Manual merging
- Load resources from mergeable libs
- Use otool to inspect app binary
- How can you leverage mergeable libraries
- Q & A

Static Libraries vs Dynamic Frameworks

Static Library :

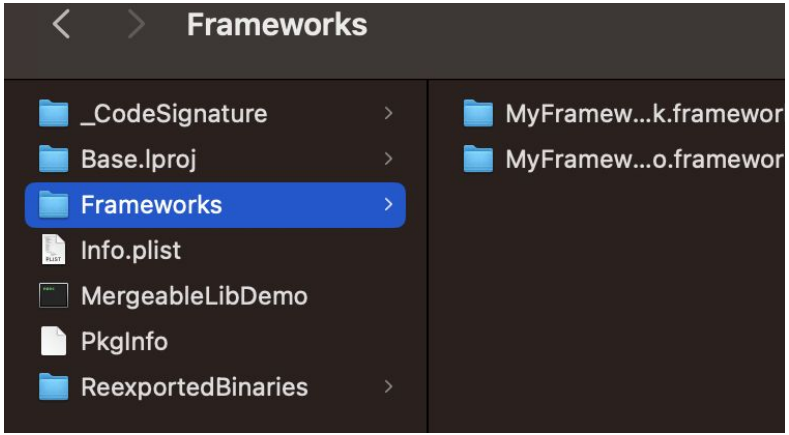
- Built and statically linked into main App Binary

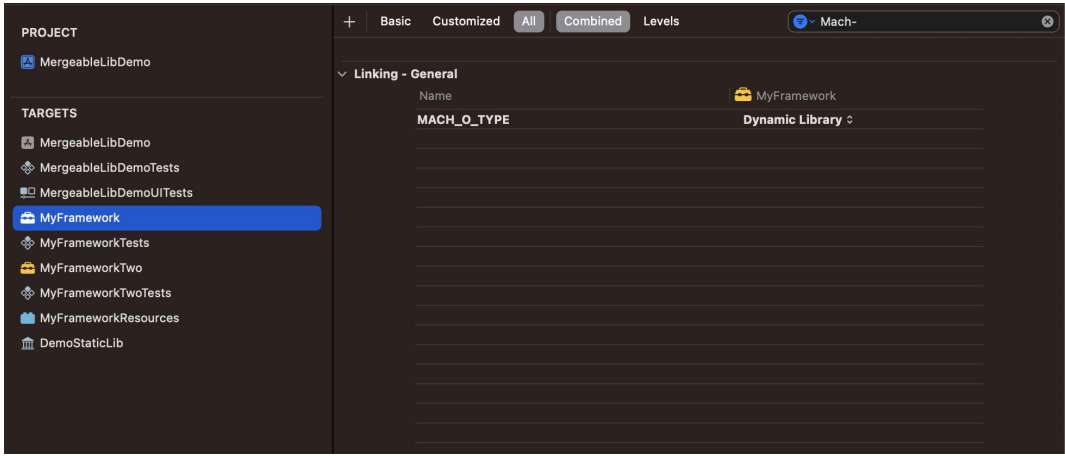


Static Libraries vs Dynamic Frameworks

Dynamic Framework:

- Built as a separate binary and placed in /Frameworks folder
- Loaded during app launch using dyld.





Static Libraries vs Dynamic Frameworks

Static Library :

Pros:

- Doesn’t have overhead on app launch time.
- Preferred by most developers when you a lot of dependencies.

Cons:

- Increase build time (link time).
- Link time is a constant addition in all of your build iterations

Static Libraries vs Dynamic Frameworks

Dynamic Frameworks :

Pros:

- Loaded at runtime so don’t increase build time
- Helps for quick iteration during development

Cons:

- Increases App launch time.

Problem

Since dynamic frameworks have overhead on App launch time, hence static libraries was the preferred way, which shifts cost to developers

Static linking time is a constant addition in all of the builds even including incremental builds.

We always had to make this choice and choose the trade off.

New

Mergeable Libraries in Xcode 15 💥

- Best of both worlds (static & dynamic) no more trade off

- Dependencies work as dynamic frameworks during Development, quick iterations

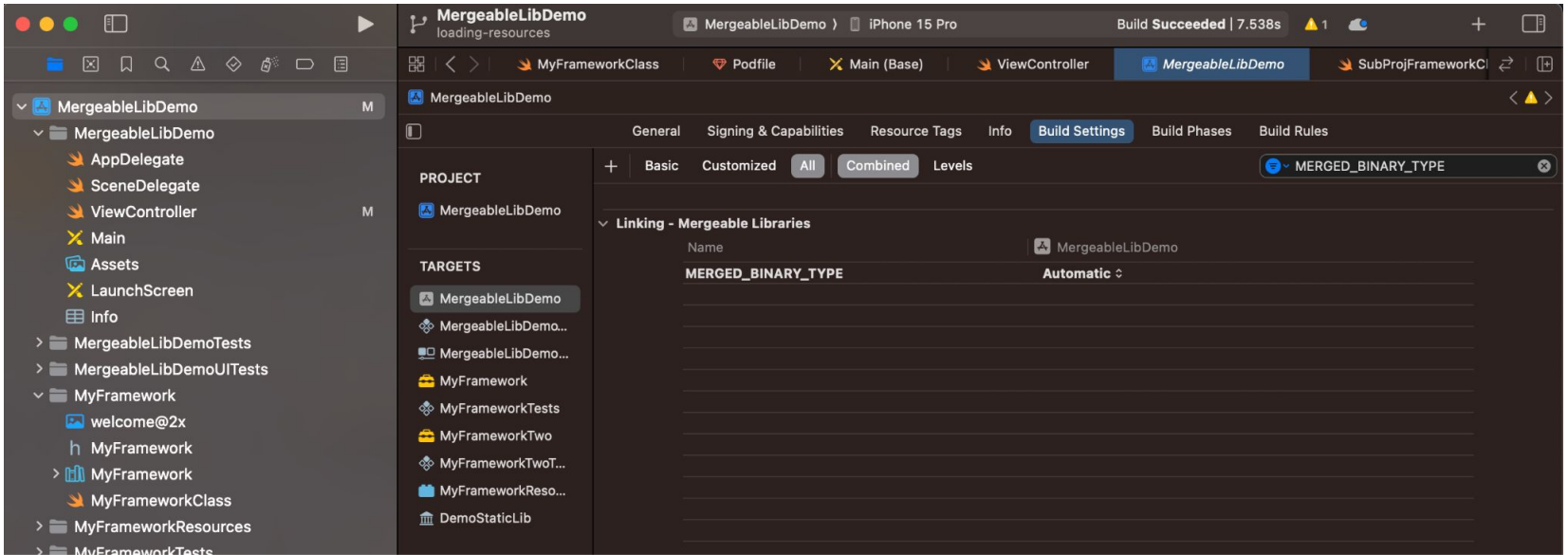
- Dependencies work like static libraries during release, faster App launch time.

Mergeable Libraries

- Automatic
  - Works for direct dependencies
  - Targets in the same Xcode project, that are mentioned in Link binary with libraries
- Manual
  - This is required when your dependencies are not direct dependencies
    - Indirect dependencies
    - Targets from other Xcode project

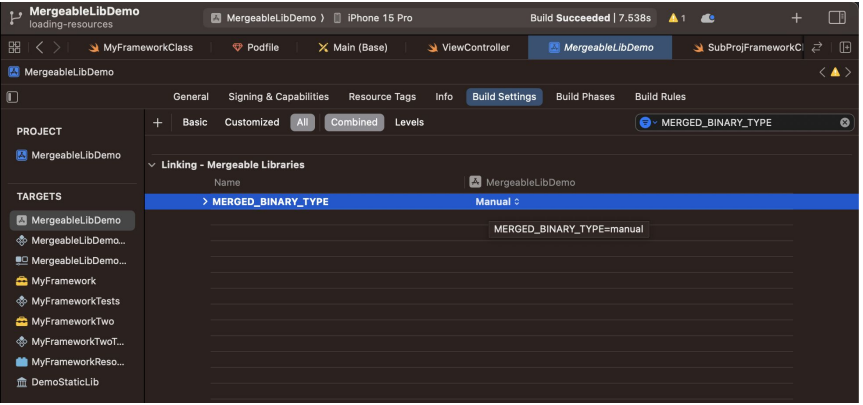
Demo

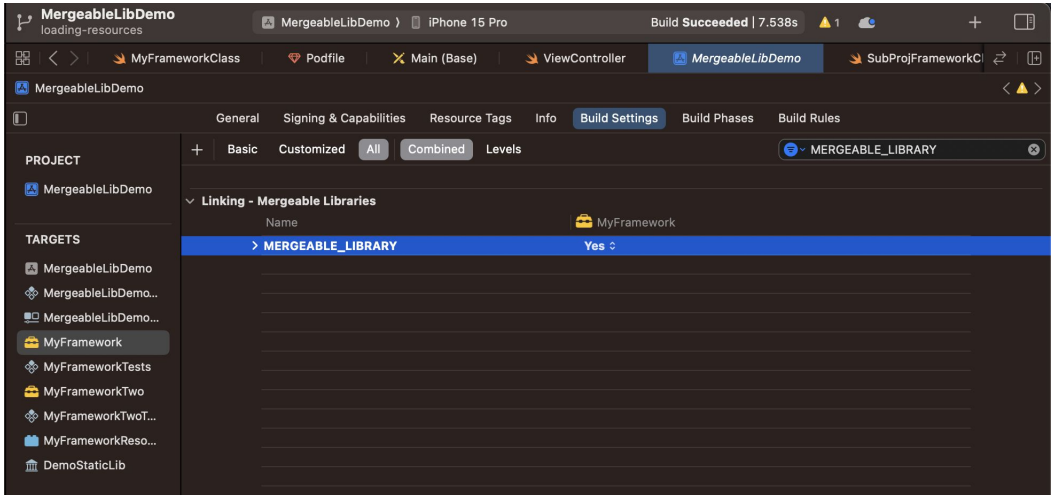
- Automatic Merging
  - MERGED_BINARY_TYPE = Automatic
- You are all set!



Demo

- Manual Merging
  - MERGED_BINARY_TYPE = Manual
  - On the targets you want to merge MERGEABLE_LIBRARY = Yes.
- You are all set!





Load resources from mergeable Libraries

- In Frameworks we put resources in framework bundle
- For static libs we create a resource bundle
- Mergeable libraries behaves as dynamic framework during development and static lib during release

Where to put resources?

Load resources from mergeable Libraries

Good News

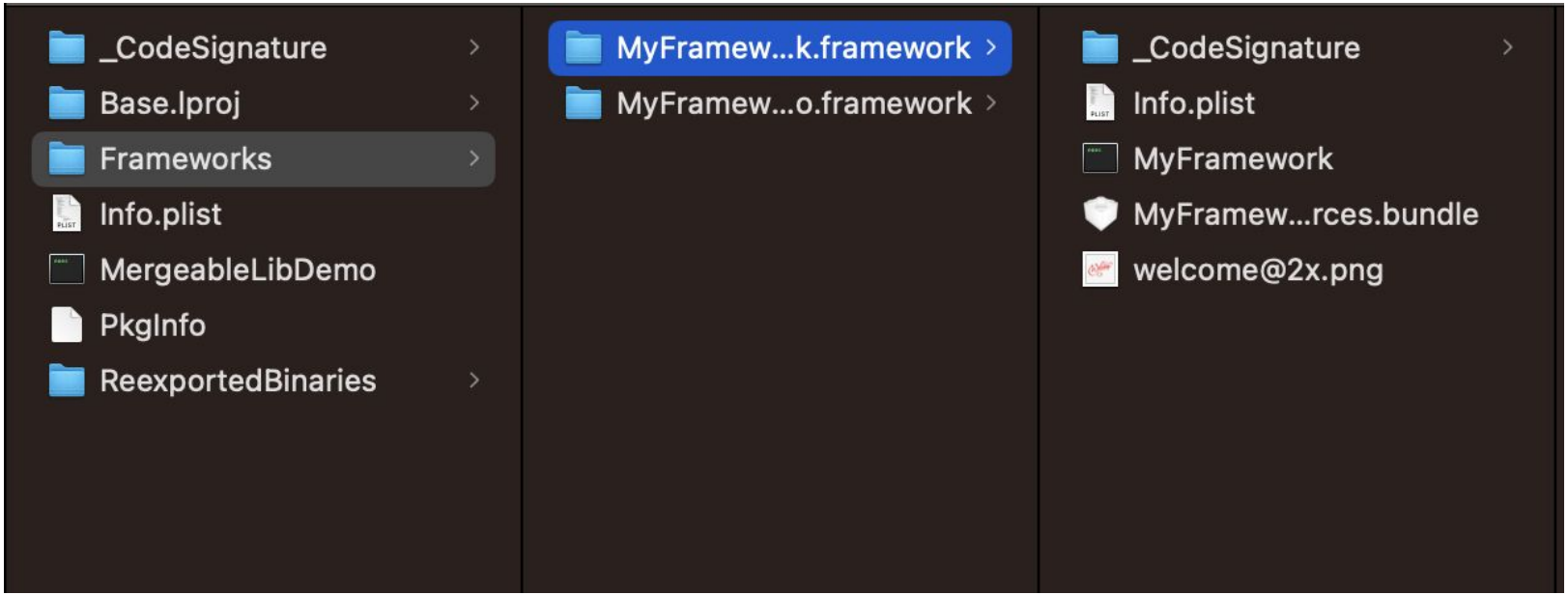
- We can always put them in Framework bundle and it works
- Apple added this method **objc_setHook_getImageName** in iOS 12 to consistently return framework bundle

Load resources from mergeable Libraries

- When you access bundle using **let** bundle = Bundle(for: MyFrameworkClass.**self**)
- It always returns framework bundle in both debug and release mode
- You can use both resource bundle and Asset Catalogue or assets individually
- You still need to embed Frameworks as resources are still kept under /Frameworks

App Bundle structure

- Binary which is part of /Frameworks is mostly empty doesn’t contain symbols
- Inspect using nm -j <path-to-binary>, prints no symbols



Inspect Mergeable Libraries

Otool -l <path-to-app-binary> , shows whether app is using Re exported binaries or not

```
➜  MergeableLibDemo git:(loading-resources) ✗ otool -l /Users/sanju/Library/Developer/Xcode/DerivedData/MergeableLibDemo-hlbygdmzxuubclbjkpryjdtggw
ld/Products/Debug-iphonesimulator/MergeableLibDemo.app/MergeableLibDemo | grep REEXPORT -A5
          cmd LC_REEXPORT_DYLIB
      cmdsize 72
         name @rpath/MyFrameworkTwo.framework/MyFrameworkTwo (offset 24)
   time stamp 2 Thu Jan  1 05:30:02 1970
      current version 1.0.0
compatibility version 1.0.0
--
          cmd LC_REEXPORT_DYLIB
      cmdsize 72
         name @rpath/MyFramework.framework/MyFramework (offset 24)
   time stamp 2 Thu Jan  1 05:30:02 1970
      current version 1.0.0
compatibility version 1.0.0
➜  MergeableLibDemo git:(loading-resources) ✗
```

Who can take advantage of Mergeable Libraries

- Ideal App Launch time is ~400 ms.

- You can measure App launch time using instruments - https://developer.apple.com/videos/play/wwdc2019/423/

- Dynamic frameworks are loaded using dyld which is part of the pre-main

References

- Medium article : https://medium.com/@SanjuNaik14/meet-mergeable-libraries-790a40aa89b8
- WWDC Video https://developer.apple.com/videos/play/wwdc2023/10268/
- Apple documentation - https://developer.apple.com/documentation/xcode/configuring-your-project-to-use-mergeable-libraries
- Demo project - https://github.com/sanju-naik/MergeableLib-Demo

Thank you

Q & A