

In Xcode10, the i386 architecture is deprecated for macOS, so building the Flutter engine for armv7/armeabi-v7a fails. Specifically, libraries like CoreFoundation contain only code for the x86_64 architecture.

Figure 1: iOS ARMv7

To check if the logic is fine, run command below:

When `find_sdk.py` return 10.13, the ninja build will succeed for `gen_snapshot (i386)`, `Flutter.framework (ARMv7)` and `libflutter.so (armeabi-v7a)`.

To build the Flutter engine for iOS simulator on a modern Mac(x86_64), the `gn` command will generate a `target_cpu` value with `x64`. Henceforth, the `Flutter.framework` and `gen_snapshot` will be `x86_64`. However, sometimes you may want to develop Flutter on a 32bit simulator(like iPhone5), you will need both `Flutter.framework` and `gen_snapshot` to be `i386`.

[illegible]

Figure 2: Android armeabi-v7a

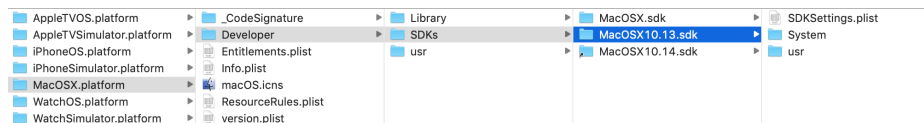


Figure 3: Uncompressed SDK in Xcode10

Follow instruction below to change the default behavior in gn command: 1. Edit your-flutter-engine-path/engine/src/flutter/tools/gn

Staged files		tools/gn
Filename	Path	
Unstaged files		
gn	tools	***

```

Hunk 1: Lines 119-143
119 119     elif runtime_mode == 'dynamic_release':
120 120         gn_args['dart_runtime_mode'] = 'release'
121 121     else:
122 122         gn_args['dart_runtime_mode'] = runtime_mode
123 123
124 124     if args.dart_debug:
125 125         gn_args['dart_debug'] = True
126 126
127 127     if args.target_os == 'android':
128 128         gn_args['target_cpu'] = args.android_cpu
129 129     elif args.target_os == 'ios':
130 130         if args.simulator:
131 131             - gn_args['target_cpu'] = 'x64'
132 132             + gn_args['target_cpu'] = 'x86'
133 133         else:
134 134             gn_args['target_cpu'] = args.ios_cpu
135 135     elif args.target_os == 'linux':
136 136         gn_args['target_cpu'] = args.linux_cpu
137 137     else:
138 138         # Building host artifacts
139 139         gn_args['target_cpu'] = 'x64'
140 140
141 141     # On iOS Devices, use the Dart bytecode interpreter so we don't incur
142 142     # snapshotting and linking costs of the precompiler during development.
143 143     # We can still use the JIT on the simulator though.
144 144     can_use_dbc = runtime_mode in ['debug', 'dynamic_profile', 'dynamic_release']

```