

Signing code for older versions of macOS on Apple Silicon

This thread has been locked. Questions are automatically locked after two months of inactivity, or sooner if deemed necessary by a moderator.



176

Note This content is a repost of [info from another thread](#) because that thread is not world readable (it’s tied to the DTK programme).

A number of folks have reported problems where:

- They have a product that supports older versions of macOS (anything prior to 10.11).
- If they build their product on Intel, everything works.
- If they build their product on Apple Silicon, it fails on those older versions of macOS.

A developer filed a bug about this (FB8830007) and, based on the diagnosis of that bug, I have some info to share as to what’s going wrong and how you can prevent it. Let’s start with some background.

macOS’s code signing architecture supports two different hash formats:

- `sha1`, the original hash format, which is now deprecated
- `sha256`, the new format, support for which was added in macOS 10.11

`codesign` should choose the signing format based on the deployment target:

- If your deployment target is 10.11 or later, you get `sha256`.
- If your deployment target is earlier, you get both `sha1` and `sha256`.

This problem crops up because, when building for both Intel and Apple Silicon, your deployment targets are different. You might set the deployment target to 10.9 but, on Apple Silicon, that’s raised to the minimum Apple Silicon system, 11.0. So, which deployment target does it choose?

Well, the full answer to that is complex but the executive summary is that it chooses the deployment target of the current architecture, that is, Intel if you’re building on Intel and Apple Silicon if you’re building on Apple Silicon. For example:

```
1 intel% codesign -d --arch x86_64 -vvv Test664892.app
2 ...
3 Hash  choices=sha1,sha256
4 ...
5 intel% codesign -d --arch arm64 -vvv Test664892.app
6 ...
7 Hash  choices=sha1,sha256
8 ...
9 ...
10 arm% codesign -d --arch x86_64 -vvv Test664892.app
11 ...
12 Hash  choices=sha256
13 ...
14 arm% codesign -d --arch arm64 -vvv Test664892.app
15 ...
16 Hash  choices=sha256
17 ...
```

The upshot is that you have problems if your deployment target is less than 10.11 and you sign on Apple Silicon. When you run on, say, macOS 10.10, the system looks for a `sha1` hash, doesn’t find it, and complains.

The workaround is to supply the `--digest-algorithm=sha1,sha256`, which overrides the hash choice logic in `codesign` and causes it to include both hashes:

```
1 arm% codesign -s - --digest-algorithm=sha1,sha256 Test664892.app
2 arm% codesign -d --arch x86_64 -vvv Test664892.app
3 ...
4 Hash  choices=sha1,sha256
5 ...
6 % codesign -d --arch arm64 -vvv Test664892.app
7 ...
8 Hash  choices=sha1,sha256
9 ...
```

Share and Enjoy

Quinn “The Eskimo!” @ Developer Technical Support @ Apple

`let myEmail = "eskimo" + "1" + "@apple.com"`

Code Signing

Asked 3 weeks ago by [eskimo](#)  

Reply to this question

This site contains user submitted content, comments and opinions and is for informational purposes only. Apple disclaims any and all liability for the acts, omissions and conduct of any third parties in connection with or related to your use of the site. All postings and use of the content on this site are subject to the [Apple Developer Forums Participation Agreement](#).

Apple Developer > Apple Developer Forums

Discover

macOS

iOS

watchOS

tvOS

Safari and Web

Games

Business

Education

WWDC

Design

Human Interface Guidelines

Resources

Videos

Apple Design Awards

Fonts

Accessibility

Internationalization

Accessories

Develop

Xcode

Swift

Swift Playgrounds

TestFlight

Documentation

Videos

Downloads

Distribute

Developer Program

App Store

App Review

Mac Software

Apps for Business

Safari Extensions

Marketing Resources

Trademark Licensing

Support

Articles

Developer Forums

Feedback & Bug Reporting

System Status

Contact Us

Account

Certificates, Identifiers & Profiles

App Store Connect

To view the latest developer news, visit [News and Updates](#).