

What is CocoaPods?

- manages library dependencies for XCode projects
- dependencies? modules to expand and improve your projects
- dependencies are specified in a single text file called a Podfile
- CocoaPods fetch source code then link it together in your XCode workspace to your project
- CONCLUSION:** improve discoverability of, and engagement in third part open-source libraries

Ruby Programming

- built with Ruby, installable with the default by Ruby available on MacOS
- What is Ruby? A programming language used in dependencies like Cocoapods.
- it is popular because of its simplicity of language
 - > Why Cocoapods is written in Ruby programming because:
 - *easy to share with Bundler/Rubygems
 - *a good choice to let users monkey-patch whatever they needed for their projects (monkey-patch: extend or modify code without altering the original source code)
 - *excellent language for CLI tools that do a lot of string manipulation (string manipulation: a class of problems where a user is asked to Process a given string and use/change its data)
(CLI tools: Command-Line Interface / terminal in MacOS)

CocoaPods Installation

- *sudo gem install cocoapods
(sudo: allows you to run programs with the security privileges of another user, administrator configures, prompts you for your password)
- *to update Cocoapods, simply install the gem again (above commands) or
For a pre-release version: sudo gem install cocoapods -pre (two dashes)
- *What is a Podfile? Is a specification that describes the dependencies of the XCode projects. Text file simply named PodFile.

Pod install or pod update

- *pod install - install new pods in your project, retrieve pods for the project. Whenever user edits Podfile to **add, update or remove a pod** (pod is a dependency/ies that you add to your project)
- *pod outdated - **~>x.y** means starting from this version but feel free to update higher (version restrictions)
- *pod update - CocoaPods will try to find an updated version of the pod without taking into account the version listed in Podfile.lock, will update all pods in the Podfile.lock
- *pod update *PODNAME* - will only update the specific podname inside the Podfile.lock

Files in CocoaPods

- *Podfile.lock - this file tracks the installed versions of each pod and locks those versions.

It also means the pod version is *locked* to a specific version unless told otherwise.

*Podfile - this the file made during the pod init or you can make one yourself (see below). This file contains the platform you are using and the library you want installed

Using Cocoapods

Making a Podfile (in your project directory)

*pod init - slightly filled out for user OR

*start with an empty file - touch Podfile (touch - make a new file, Podfile - name of the file), you can find this file in the project folder

< Creating a new Xcode project with CocoaPods

To create a new project with CocoaPods, follow these simple steps:

- Create a new project in Xcode as you would normally.
- Open a terminal window, and `$ cd` into your project directory.
- Create a Podfile. This can be done by running `$ pod init`.
- Open your Podfile. The first line should specify the platform and version supported.

```
platform :ios, '9.0'
```

- In order to use CocoaPods you need to define the Xcode target to link them to. So for example if you are writing an iOS app, it would be the name of your app. Create a target section by writing `target '$TARGET_NAME' do` and an `end` a few lines after.
- Add a CocoaPod by specifying `pod '$PODNAME'` on a single line inside your target block.

```
target 'MyApp' do
  pod 'ObjectiveSugar'
end
```

During trial of Cocoapods

Got this error:

```
[NOTE]
You may have encountered a bug in the Ruby interpreter or extension libraries.
Bug reports are welcome.
For details: https://www.ruby-lang.org/bugreport.html

[IMPORTANT]
Don't forget to include the Crash Report log file under
DiagnosticReports directory in bug reports.

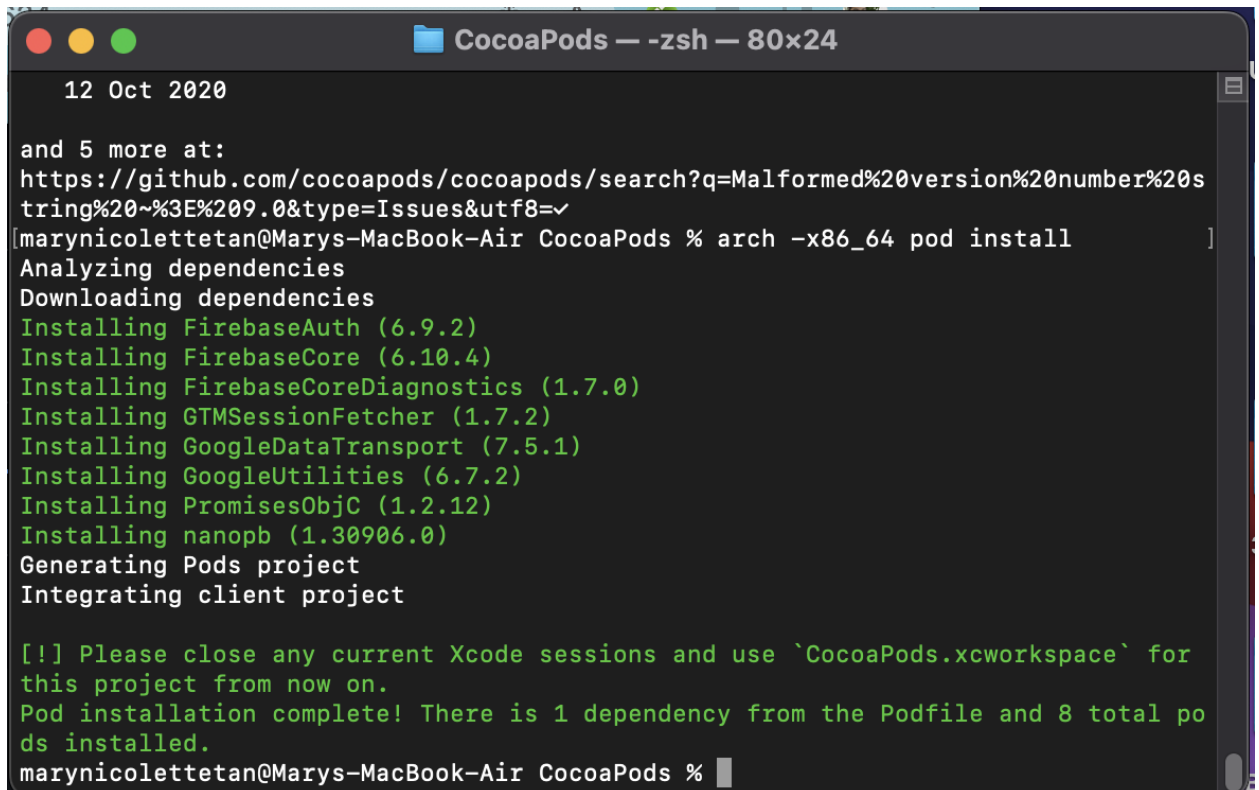
zsh: abort      pod install
marynicolettetan@Marys-MacBook-Air CocoaPods %
```

To resolve this error:

From the project folder

```
*sudo arch -x86_64 gem install ffi
```

```
*arch -x86_64 pod install
```

A terminal window titled "CocoaPods — -zsh — 80x24" showing the output of a pod install command. The terminal displays the date "12 Oct 2020" and a link to a GitHub issue. The command executed is "arch -x86_64 pod install". The output shows the installation of various dependencies: FirebaseAuth (6.9.2), FirebaseCore (6.10.4), FirebaseCoreDiagnostics (1.7.0), GTMSessionFetcher (1.7.2), GoogleDataTransport (7.5.1), GoogleUtilities (6.7.2), PromisesObjC (1.2.12), and nanopb (1.30906.0). The terminal also shows "Generating Pods project" and "Integrating client project". A green message indicates that the installation is complete and provides instructions to close Xcode sessions and use the generated workspace. The prompt "marynicolettetan@Marys-MacBook-Air CocoaPods %" is visible at the bottom.

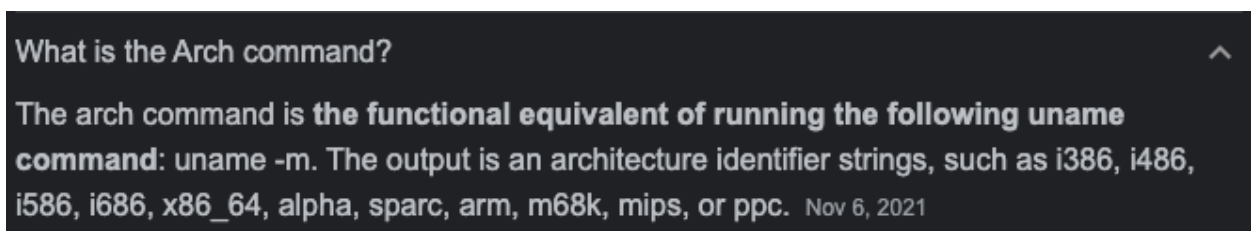
```
12 Oct 2020

and 5 more at:
https://github.com/cocoapods/cocoapods/search?q=Malformed%20version%20number%20s
tring%20~%3E%209.0&type=Issues&utf8=✓
[marynicolettetan@Marys-MacBook-Air CocoaPods % arch -x86_64 pod install
Analyzing dependencies
Downloading dependencies
Installing FirebaseAuth (6.9.2)
Installing FirebaseCore (6.10.4)
Installing FirebaseCoreDiagnostics (1.7.0)
Installing GTMSessionFetcher (1.7.2)
Installing GoogleDataTransport (7.5.1)
Installing GoogleUtilities (6.7.2)
Installing PromisesObjC (1.2.12)
Installing nanopb (1.30906.0)
Generating Pods project
Integrating client project

[!] Please close any current Xcode sessions and use `CocoaPods.xcworkspace` for
this project from now on.
Pod installation complete! There is 1 dependency from the Podfile and 8 total po
ds installed.
marynicolettetan@Marys-MacBook-Air CocoaPods %
```

A dig deeper into the error, during our research it mentioned that it had something to do with the M1 processor and the error also said “*You may have encountered a bug in the Ruby interpreter or extension libraries.*”

arch command

A screenshot of a search result for "What is the Arch command?". The result explains that the arch command is the functional equivalent of running the 'uname' command with the '-m' flag. It lists various architecture identifiers such as i386, i486, i586, i686, x86_64, alpha, sparc, arm, m68k, mips, and ppc. The date "Nov 6, 2021" is also visible.

```
What is the Arch command? ^

The arch command is the functional equivalent of running the following uname
command: uname -m. The output is an architecture identifier strings, such as i386, i486,
i586, i686, x86_64, alpha, sparc, arm, m68k, mips, or ppc. Nov 6, 2021
```


However, when you run the .xcworkspace all is well and the libraries are identified.

References:

Ruby used in CocoaPods

<https://medium.com/xcblog/a-history-of-ruby-inside-ios-development-427b5a09f91e>

CocoaPods Getting Started Guides

<https://guides.cocoapods.org/using/getting-started.html>

Unix commands - sudo

<https://kb.iu.edu/d/amyi#:~:text=The%20sudo%20command%20allows%20you.which%20the%20system%20administrator%20configures.>

What is a PodFile?

<https://guides.cocoapods.org/using/the-podfile.html>

Pod install error

<https://github.com/CocoaPods/CocoaPods/issues/10518#issuecomment-798912624>

Arch command

<https://www.computerhope.com/unix/uarch.htm>