Build ChordPro from source on a Mac

While it is not too easy to build **ChordPro** from source, it is doable. It will take some time and a lot of space; mainly because of the size of Xcode when you want to build the SwiftUI GUI.

Note: *These instructions are tested on macOS Sonoma, Ventura and Monterey.*

Build ChordPro CLI

This is needed to build any of the three options:

- Command Line version
- SwiftUI GUI version for macOS Monterey or higher
- Classic GUI version

Homebrew

Install homebrew and follow its instructions carefully.

Perl

Once homebrew is installed, install the following formulas:

```
brew install perl
brew install cpanminus
```

Again, follow the instructions. It is important to add stuff to your `~/.zprofile`. In the end, the content should look like this:

```
eval "$(/opt/homebrew/bin/brew shellenv)"
eval "$(perl -I$HOME/perl5/lib/perl5 -Mlocal::lib=$HOME/perl5)"
```

Note: *The pre-installed version of Perl cannot be used to build ChordPro. It contains a ‘universal’ dynamic library, both for Intel and ARM and ChordPro needs it for a specific architecture.*

Then, instal the following Perl package:

```
cpanm PAR::Packer
```

***Note**: This package comes pre-installed on the Mac but it is insisting on using the 'system-perl'. So we have to add a local version and that is why it is so important to have a correct `~/.zprofile`.*

Build ChordPro CLI

- Download or clone the dev branch of *ChordPro*.
- Open the downloaded folder in the Terminal (right-click folder in the Finder and choose 'New Terminal at Folder')

In the Terminal:

```
cpanm --installdeps .
```

***Note**: Don't forget the '.' at the end!*

This will install all the needed dependencies to build *ChordPro*.

***Note**: Sometimes, ChordPro will add new dependencies. If compiling does not work anymore after a checkout, run above comment again.*

Now you can build the CLI version of *ChordPro*:

```
cd pp/macos
make ppl TARGET=chordpro
```

You will get some warnings but the building should complete and there is a *ChordPro* binary in the `pp/macos/build` directory.

***Note**: If you build on an Apple Silicon Mac, this binary will -not- run because it is unsigned. No worries, we deal with that later when building a GUI.*

Build ChordPro SwiftUI GUI

If you are able to build the CLI version, you can build the SwiftUI GUI. It will be build with Xcode, again from the command-line.

Install Xcode

Install Xcode; the `command line tools` are unfortunately not enough. Best is to download it directly from the Apple developer website or use Xcodes. Downloading Xcode from the Mac App Store often gives problems and is not recommended.

Build the GUI

```
cd ../macosswift
make
```

This will build a DMG with an ad-hoc signed application for the architecture of the Mac you are using now.

Note*: You cannot build an Intel version on an Apple Silicon Mac or an Apple Silicon version on an Intel. Also, ChordPro cannot be build as a ‘universal application’.*

You should now have a DMG in the `pp/macosswift` directory that is ready to use.

ChordPro Classic GUI

While the SwiftUI wrapper is fresh and new; the Classic GUI can also still be build on a Mac. However, this is *absolutely* not easy.

Homebrew

Install wxWidgets with Homebrew; the cross-platform GUI toolkit used for Classic.

```
brew install wxwidgets
```

Instal an additional formula:

```
brew install zlib
```

Perl

Extra dependencies you have to install:

```
cpanm Alien::wxWidgets
cpanm ExtUtils::XSpp
```

Now comes the biggest challenge; install xwPerl from source. Unfortunately, wxPerl is currently not well maintained and Johan Vromans, the maintainer of *ChordPro*, created an independent fork. Download the latest release from his repo.

Open the `Wx-x.xxx` folder in the terminal and do the following:

```
perl ./Makefile.PL
make
make install
```

Build the GUI

Go to the `pp/macos` directory again and build ChordPro:

```
make
```

This will build a DMG for the architecture of the Mac you are using now.

***Note**: An Apple Silicon version will be ad-hoc signed or else it will simply not run.*

You should now have a DMG in the `pp/macos` directory that is ready to use.