

Instructions for Installing KStars, Dependencies, and related software on OS X with Emerge instead (Not tested on 10.12)

(At least until we work out an automated and/or simpler method of doing it)

1. Make sure you have home-brew installed. brew.sh
2. Make sure that you have git, make, and cmake installed.
3. Install some dependencies using homebrew

```
brew install cmake wget coreutils p7zip gettext ninja python3  
ninja bison boost shared-mime-info
```
4. And some more:

```
brew tap homebrew/science  
brew install pkgconfig  
brew install cfitsio  
brew install cmake  
brew install eigen  
brew install astrometry-net
```
5. here If you do not already have qt 5 installed, then you will need it. By the way, this can take a very long time. You can also download QT5 from their website to your home directory. I did this with QT Creator and just specified the path to it. But it should also work with the homebrew version.

```
brew install qt5 --with-dbus
```

4. I followed these instructions to build INDI from source and get it installed on the Mac in the proper location: indilib.org/forum/general/210-howto-buil...st-libindi-ekos.html

Do this at the command line:

```
mkdir ~/Projects
cd ~/Projects
git clone github.com/indilib/indi.git
```

Before you can build, follow the instructions in the Mac OS X file in the download from the git. This includes downloading 0.15.0, libusb-1.0.19, cfit3380, and gsl-2.1, then placing the folders at the same level as your git download, and then configure and make from the command line in the gsl folder. screenshot to the right.

Also before you build, double check that CMakeLists.txt in the folder is correct. Open the file and make sure it says this, if change it.

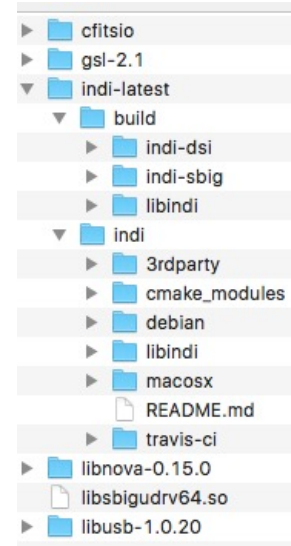
```
set (indiclient_SRCS
${CMAKE_CURRENT_SOURCE_DIR}/libs/indibase/basedevice.cpp
${CMAKE_CURRENT_SOURCE_DIR}/libs/indibase/baseclient.cpp
${CMAKE_CURRENT_SOURCE_DIR}/libs/indibase/indiproperty.cpp
${CMAKE_CURRENT_SOURCE_DIR}/base64.c
)
```

```
set (indiclientqt_SRCS
${CMAKE_CURRENT_SOURCE_DIR}/libs/indibase/basedevice.cpp
${CMAKE_CURRENT_SOURCE_DIR}/libs/indibase/baseclientqt.cpp
${CMAKE_CURRENT_SOURCE_DIR}/libs/indibase/indiproperty.cpp
${CMAKE_CURRENT_SOURCE_DIR}/base64.c
)
```

Now go back to the Projects directory and follow these instructions.

```
export PATH=$PATH:$(brew --prefix gettext)/bin
export Qt5_DIR=$(brew --prefix qt5)
export Qt5DBus_DIR=$Qt5_DIR
export Qt5Test_DIR=$Qt5_DIR
export Qt5Network_DIR=$Qt5_DIR
export ECM_DIR=$(brew --prefix kf5-extra-cmake-modules)/share/ECM
```

```
mkdir -p build/libindi
cd build/libindi
cmake -DCMAKE_INSTALL_PREFIX=/usr/local -DCMAKE_BUILD_TYPE=Debug
~/Projects/indi/libindi
sudo make install
```



readme
libnova-
unzipped
running
See the

libindi
not,

If you want the 3rd Party drivers like SBIG or DSI, you can try to build and install them or just copy them from somewhere else if you can find them. An alternative would be to install the free Cloudmakers Indiserver program:

<http://www.cloudmakers.eu/indiserver/>

And just run that to serve your 3rd Party devices

Check that the binary files for the drivers are all in /usr/local/bin and that the driver xml files are in /usr/local/share/indi

6. There is a problem with the Emerge routines where they access the wrong server. It is explained on this website: <http://unix.stackexchange.com/questions/304001/problem-building-kde-software-on-osx>

To solve it, create a file called ~/.gitconf and in this file you want to put these contents:

```
[url "git://anongit.kde.org/"]
  insteadOf = kde:
[url "ssh://git@git.kde.org/"]
  pushInsteadOf = kde:
```

7. Manually create the extra cmake modules because there seems to be an issue with using the version in homebrew.

```
git clone https://github.com/KDE/extra-cmake-modules.git
cd extra-cmake-modules
cmake .
make
sudo make install
```

8. Following the instructions from this page, we can set up emerge.

https://community.kde.org/Guidelines_and_HOWTOs/Build_from_source/Mac

```
mkdir ~/kf
cd ~/kf
git clone --branch unix3 git://anongit.kde.org/emerge.git
mkdir etc
cp emerge/kdesettings.mac etc/kdesettings.ini
export CMAKE_LIBRARY_PATH=/usr/local/Cellar/gettext/0.19.8.1/lib
export
CMAKE_INCLUDE_PATH=/usr/local/Cellar/gettext/0.19.8.1/include
```

```
export PATH=/usr/local/Cellar/bison/3.0.4/bin:$PATH
```

- **Setup environment (path must contain your Qt install, here 5.7 default location and brew's gettext)**

```
export
PATH=/Users/cullmann/Qt5.7.0/5.7/clang_64/bin:/usr/local/Cellar/g
```

```
ettxt/0.19.7/bin:$PATH
. emerge/kdeenv.sh
```

9. Now you are ready to get kstars. Type the following and hit enter:

```
emerge kstars
```

10. You will need to get the data folder to the correct location. It is located in:

```
~/kf/share/kstars
```

You will need to either copy and paste it to

```
~/Library/Application Support/kstars
```

or link it to its current location like this:

```
ln -s ~/kf/share/kstars/* ~/Library/Application\ Support/kstars/
```

11. If you would like gsc so that the simulators can have nice stars to work with, then install gsc following these instructions:

<http://www.indilib.org/support/tutorials/139-indi-library-on-raspberry-pi.html>

a. Find a folder on your hard drive in a terminal window and use these commands:

```
mkdir ~/gsc
cd ~/gsc
wget -O bincats_GSC_1.2.tar.gz http://cdsarc.u-
strasbg.fr/viz-bin/nph-Cat/tar.gz?bincats/GSC_1.2
tar -xvzf bincats_GSC_1.2.tar.gz
cd src
make
mv gsc.exe gsc
sudo cp gsc /usr/local/bin/
cd ..
cp ~/gsc/bin/regions.* /gsc
```

I had a couple of problems. First, it needed both of the regions files in a subfolder called bin inside the final gsc folder. Second, I had to copy all the folders that began with N and S to the final gsc folder (but NOT in the bin subfolder). Third, I had trouble getting the environment variable permanent, so you will note I made the final line install to /gsc not ~/gsc

12. Astrometry.net and xplanet are fairly easy to install if you follow the directions on their websites. They work extremely well. I hope to update this with more instructions about it.