

## Installing PLATON on a Mac and getting Olex<sup>2</sup> to use it

Anthony Linden, Department of Chemistry, University of Zurich, Switzerland, [anthony.linden@chem.uzh.ch](mailto:anthony.linden@chem.uzh.ch)

Here is how to get PLATON running on a Mac. PLATON uses X11 for its GUI, so you need the X11 libraries and tools and you then need a Fortran compiler. Ton Spek does not create compiled versions of PLATON with static libraries, hence you must compile it yourself and the need for all this.

The instructions below should work with Mac OS 12 (Monterey), 13 (Ventura), 14 (Sonoma), and 15 (Sequoia) on both Intel and M1/M2/M3/M4 chipset Macs. The procedure sometimes changes a little for each new Mac OS version, mainly because X11 libraries sometimes change location and a PLATON executable compiled under one OS does not always work under another. Once you get this set up for one OS, it usually works from then on at least until the next OS is installed, so you usually only need to do the actual compile step when updating PLATON. It is advisable to update and recompile PLATON regularly, because it is constantly being improved.

If you are familiar with Unix/Linux, the following is probably clear to you. If not, please try to follow the instructions exactly. **Please read and type carefully, as there are important spaces, dots and dashes within some commands.** All line commands typed are case sensitive. Unix commands usually do not give any feedback when they work correctly. Messages often mean something might not have worked.

This looks long, but is quite quick if all goes well. Don't be discouraged! Please reach out if confused.

See also Ton Spek's readme at <https://www.platonsoft.nl/xraysoft/Mac-OSX/platon>. Note that it is unnecessary to install the full Xcode developer kit, just Command Line Tools (see below).

**Note:** In early 2024, some people experienced difficulties with compilation because of errors during the final link (ld) stage concerning missing library definitions, or similar. Apparently a previously installed version of Command Line Tools and/or gfortran had become incompatible with the MacOS after an upgrade or update. If this happens to you when you have previously been compiling PLATON successfully, try steps 3a and 4.

If needed, there is a stripped Fortran only version of PLATON (without graphics) available (special) that usually compiles without problems. It allows for easy validation of jobs such as 'platon -U name.cif'.

1 – Download and install the latest XQuartz (2.8.5 or later) from <https://xquartz.macosforge.org>, unless you have installed it already. XQuartz is needed and will start automatically when running PLATON later on, because it acts as an X11 server for the PLATON graphics.

2 - Start XQuartz or open Terminal (either can be used here; both are located in Applications/Utilities). When the window opens, you are in the top level directory (folder) for your Mac user account. To return to this folder at any time, type 'cd'. To find out which folder you are in at any time, type 'pwd'.

3 - Install Apple Command Line Tools; type in the XQuartz window: `xcode-select --install`

3a - If you want to update an existing version of Command Line Tools, type in the XQuartz window:  
`cd /Library/Developer/`  
`sudo rm -rf CommandLineTools`  
 (enter your password, then give it a little time)  
`xcode-select --install`

4 - Download and install gfortran:

Go to <https://github.com/fxcoudert/gfortran-for-macOS/releases>

Download the gfortran version appropriate for your Mac OS version and chipset.

Double-click the downloaded .dmg file and run the gfortran installer.

Alternatively, you can install Homebrew from <https://brew.sh>, then enter in XQuartz or Terminal the command: `brew install gcc`.

5 - Create a temporary folder named 'platon' at a convenient location. I use Desktop for now. If you use uppercase letters for any part of the folder name, you must type it exactly the same way from now on.

6 - Download the files: `platon.f.gz`, `xdrvr.c.gz`, `sucrose.spf` from: <https://www.platonsoft.nl/xraysoft/unix/platon> and move them into the temporary folder just created. If access to the site is blocked by a security window, you can force it to open by clicking "show more info" and then clicking "visit website" at the bottom of the pop-up window.

7 - In XQuartz, go to that folder; type:

```
cd
cd Desktop/platon
```

8 - gunzip the files `platon.f.gz` and `xdrvr.c.gz` (the Mac probably has done that automatically). If not:

```
gunzip platon.f.gz
gunzip xdrvr.c.gz
```

9 - Compile PLATON. This is the most critical and often troublesome step. Type carefully and exactly:  
`sudo gfortran -o platon platon.f xdrvr.c -I/opt/X11/include -L/opt/X11/lib -lX11`

Note that in the above, the character after the dash in `-I/opt` is an uppercase i and the character after the dash in `-lX11` is a lowercase L. X11 is "Xeleven".

If nothing appears to be happening, just wait. The compiler takes a minute or two. If the command prompt reappears after a while without any messages, you have succeeded; go to Step 10.

If this step fails with error messages after the long wait, either you did not type the compile instruction exactly as above (a common mistake, so try again carefully), or Command Line Tools or gfortran are not the latest versions; see the Note before Step 1 on the previous page.

If you have an earlier version of the Mac OS, e.g. Yosemite or Mavericks, the X11 libraries might not be where they are expected to be and one of the following might work instead (these will certainly not work if you use El Capitan or later):

```
gfortran -o platon platon.f xdrvr.c -L/opt/X11/lib -lX11
gfortran -o platon platon.f xdrvr.c -L/usr/X11/lib -lX11
```

10 - Copy the newly created file 'platon' (the executable) to `/usr/local/bin` or any other location you like your executables, such as `shelx`, etc., to be. I recommend not putting these in the system Applications folder, nor should you leave them in the temporary platon folder used for the compilation. If you do not have access to the folder `/usr/local/bin`, prefix the lines below with 'sudo'. Alternatively, use any convenient folder within your user account and use the path to it instead of `/usr/local/bin` in the following.

In the following, the dot at the end is important:

```
cp platon /usr/local/bin/.
chmod -R 755 /usr/local/bin/platon
```

Quit XQuartz or Terminal and start it again so the system knows about the new program (or type 'hash -r' if in the default bash shell, 'rehash' if in tcsh or csh). Only needed the first time you install PLATON.

11 - Test: type 'platon sucrose.spf' in the XQuartz window (you have to be in the platon folder again if you moved away), click on ORTEP in the main PLATON menu for an ORTEP plot.

If all is well, you have a working version of PLATON that you can now use quite independently of any other program simply by using the command line in XQuartz or a Terminal window with any CIF, for example by typing 'platon x.cif'. Similarly, if the SHELX programs are on a known path, you can run them manually from there too. All you need to do is 'cd' to the folder containing your current structure files.

To update PLATON at any time, repeat steps 6-10. The availability of a new version is indicated on the PLATON opening window (red text starting with WEB:) and you can download the new version by clicking on that information. The compilation and installation steps can be simplified by creating a makefile with the relevant instructions. See hints at <https://www.cs.colby.edu/maxwell/courses/tutorials/maketutor/>. A makefile for compiling PLATON is below. Remove the # from the first two lines if your Mac does not automatically unzip the downloads. <tab> means insert the tab character on those lines. Save this as a file called 'makefile' in the same place as the `platon.f` and `xdrvr.c` files and run it in XQuartz or Terminal with the command 'make' initially to compile and then, if no errors, 'make install' to move things to the right place.

```
platon:
#<tab>   gunzip -f platon.f.gz
#<tab>   gunzip -f xdrvr.c.gz
<tab>   gfortran -o platon platon.f xdrvr.c -I/usr/X11/include -L/opt/X11/lib -lX11

install:
<tab>   mv platon /usr/local/bin/.
<tab>   chmod 755 /usr/local/bin/platon
```

## How to get Olex<sup>2</sup> to use PLATON on a Mac

To get Olex<sup>2</sup> to know that PLATON exists, and thus show the 'PTON' button at the top of the main GUI instruction panel, and to start PLATON, you need to do the steps below.

1 - Install PLATON as described above. Make a note of the full path to where you put the PLATON executable. For example, /usr/local/bin if you put it there. It is convenient if you also put the executables for shelxt, shelxl, shredcif, etc. in the same location.

2 - Start Olex<sup>2</sup>. In the main GUI instruction panel, select the HOME panel and then click on the SETTINGS tab. The second last item in the list is "PATH:". Enter the full path to your PLATON executable here and then click out of the path definition box. If you wish to specify multiple paths here, the delimiter between them is a colon (:), not a semi-colon as used on Windows computers. Quit Olex<sup>2</sup> and restart it. Come back to the path definition box and ensure the path you typed is shown. If the box is empty try again.

3 - To test everything, run Olex<sup>2</sup>. Select the sucrose demo structure on the Home panel. Go to the Work tab, then click on the down arrow beside the solve tab. If the SHELX programs have been installed and are on the defined path, the dropdown at the program line should show ShelXT (and ShelXS if installed). Click on the down arrow beside the refine tab and again you should see ShelXL in the program dropdown. Olex<sup>2</sup>'s own routines are displayed by default, but some people, including this author, prefer to use the SHELX programs. At the very top right of the Olex<sup>2</sup> screen, there should be a PTON button if Olex<sup>2</sup> knows that PLATON exists. If all this is OK, then you are ready to go!

PLATON might not know where check.def is, but that is no longer a problem, because if PLATON cannot find check.def, it creates the file each time in the current working directory and uses that.