

Installing PLATON on a Mac and getting Olex² to use it

Anthony Linden, Department of Chemistry, University of Zurich, Switzerland, 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 assume you are using OS 10.11 (El Capitan). The procedure seems to change a little for each new Mac OS version, mainly because X11 libraries always seem to 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 works from then on 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 at least once a month, because it is constantly being improved.

If you are familiar with Unix/Linux, the following is probably clear to you. If not, please simply try to follow the instructions below exactly. Please read and type carefully, as there are important spaces and dashes within some commands. All commands typed in the XQuartz window 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!

The following is based on Ton Spek's readme at <http://www.platonsoft.nl/xraysoft/Mac-OSX/platon>, but I have modified it a little to update the steps and remove the unnecessary installation of the full Xcode developer kit. Everything below is case sensitive. Any other instructions or even a compiled version you find on the PLATON web site might not be up to date.

1 - Download XQuartz 2.7.9 from <http://xquartz.macosforge.org>

2 - Install it and start it (located in Applications/Utilities). When the window opens, you are in the top level directory (folder) for your Mac user account. To return here 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`

4 - Download gcc and gfortran from hpc.sourceforge.net

The binaries are near the top of the page: `gcc-6.1-bin.tar.gz`, `gfortran-6.1-bin.tar.gz`. It is recommended you use Safari for the download to avoid the tar files being completely unpacked, after which the instructions below will not work (Firefox unfortunately unpacks the downloads).

5 - Move these two files to your Desktop (more convenient that way). Probably your Mac automatically uncompresses them, so the `.gz` is removed, but you still must have the single files with the `.tar` ending for the next step.

6 - In the XQuartz window, type (add `.gz` to the file names if they are still present with that extension):
`cd`

`cd Desktop`

`sudo tar -xvzf gcc-6.1-bin.tar -C /`

`sudo tar -xvzf gfortran-6.1-bin.tar -C /`

sudo asks for your login password the first time, assuming you have administration rights. Otherwise the password for the admin account on your Mac.

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

8 - Download the files: platon.f.gz, xdrv.c.gz, check.def, sucrose.spf from:
'http://www.platonsoft.nl/xraysoft/unix/platon' and move them into that folder.

9 - In XQuartz, go to that folder. If already at Desktop from the above, then just type 'cd platon'. If lost, type:

```
cd  
cd Desktop/platon
```

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

```
gunzip platon.f.gz  
gunzip xdrv.c.gz
```

The file check.def might have acquired the .txt ending during the download, if so, type:

```
mv check.def.txt check.def
```

11 - Compile PLATON. This is the most critical and often troublesome step. Type carefully and exactly:

```
sudo gfortran -o platon platon.f xdrv.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.

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!

If this step fails with error messages after the long wait, either you did not type the above correctly, or the X11 libraries are not where they are expected to be, which might be the case if you are not using El Capitan.

If you have an earlier version of the Mac OS, e.g. Yosemite or Mavericks, one of the following might work instead (these will certainly not work if you use El Capitan):

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

If this also fails, try to find the path to the `/X11/lib` folders and use that with the `-L` switch.

12 - Copy the newly created file 'platon' (the executable) and check.def 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 `/Applications` folder. With El Capitan, if you do not already have the folder `/usr/local/bin`, you might not be allowed to create it. Simply 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/.  
cp check.def /usr/local/bin/.  
chmod -R 755 /usr/local/bin/platon  
chmod -R 755 /usr/local/bin/check.def
```

If these commands do not work, try prefixing them with 'sudo '.

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

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

13 - Set up an environment variable for the location of the check.def file to be used as part of CheckCIF validation. If this step is skipped, it is not important because PLATON will create a new check.def in your current working folder each time if it cannot find check.def otherwise.

If you wish to define the environment variable, it has to be done each time you start XQuartz, so for convenience it needs to be in the `.bash` file if you use the default bash shell in XQuartz, or in `.cshrc` if you change to `csh` or `tcsh`.

For bash: `export CHECKDEF=<path>/check.def`

For tcsh: `setenv CHECKDEF <path>/check.def`

Where '`<path>/check.def`' might be `'/usr/local/bin/check.def'` if you put `check.def` where I suggest above.

14 - 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, such as `'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 with your current structure files in them.

To update PLATON at any time, you need to repeat steps 7-12. The availability of a new version is indicated on the PLATON opening window (red text starting with WEB:). The compilation and installation steps can be simplified by creating a makefile with the relevant instructions. See hints at <http://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. Simply save this as a file called `'makefile'` and run it in XQuartz with the command `'make'` initially to compile and then, if no errors, `'make install'` to move things to the right places.

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

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

How to get Olex² to use PLATON

On a Mac, to get Olex² 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. This only works from version 1.2.8 of Olex² released in August 2016. Earlier versions had bugs which made getting PLATON to work more complicated, although it can be done. Therefore, please ensure you have the latest version of Olex² installed.

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.

2 - Start Olex². In the main GUI instruction panel, select the HOME panel and then click on the SETTINGS tab. The second last item in the list that appears is "PATH:". Enter the full path to your PLATON executable here and then click out of the path definition box. Quit Olex² 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 - You are done! Platon should work now, but you have to have a refined structure with a CIF or `.res` file to test this, e.g. try the sucrose sample structure and refine it with SHELXL with the option ACTA activated.

It is possible that PLATON still does not know where `check.def` is, but that is no longer a problem, because if PLATON cannot find `check.def`, it creates one each time in the current working directory and uses that.

Note that if the executables for all external programs like SHELXL, SHELXT, etc. are in the same location as PLATON, Olex² will have no trouble finding them and they do not then need to be placed in the Olex² application folder itself.