

Installation of PhreePlot on Linux and Mac systems

1. Requirements

PhreePlot will run natively under a variety of operating systems but this may require the installation of some external libraries, or an update to existing ones. These supporting libraries include those used by **PhreePlot** itself along with those used by the embedded **PHREEQC** (**phreeqcRM**) library. The actual libraries required depend on your operating system and the compilers that have been used to compile **PhreePlot** and **phreeqcRM**. You may need administrator rights (root permission) to install or update these components. No compilation of **PhreePlot** is required. **Ghostscript** (gs) is also required for converting the native **Postscript** output to alternative file formats.

For the gfortran version of **PhreePlot**, the following libraries are required:

- gfortran (part of the gcc distribution)
- libgfortran5 (runtime library)

Other libraries used are: libstdc++, libc, libm, libgomp, linux-vdso, libgcc, libpthread, libquadmath and libdl.

If you have an older version of Linux installed, you may have to update your libraries, especially gfortran (if it is 9.3.0 or above, this should be sufficient). Instructions for how to update older systems can be found at https://fortran-lang.org/learn/os_setup/install_gfortran and some instructions are also included below.

The various releases of **PhreePlot** are named according to their build characteristics:

pp version: 1

OS: Linux | OSX

Version: xx.x for OSX (i.e. 11.5)

compiler: gfortran

chip: intel

PHREEQC: build number

giving the compressed archive the name pp_1-Linux-gfortran_intel-15968.tar.bz2.

All of the shell scripts included in the distribution are for the bash shell. In other shell environments, these should either be modified accordingly or be run explicitly with the bash shell, e.g.:

```
/bin/bash phreeplot.sh
```

2. Getting ready for installation

2.1 LINUX

The source of the libraries required varies with the flavour of Linux being used.

2.1.1 UBUNTU

You need `gcc` and **Ghostscript** (`gs`) installed to run to full capability.

Check in your terminal (or console) if these are installed with:

```
which gcc
```

and

```
which gs
```

If a full path is not returned, then they are not installed. To install, enter the following as needed:

```
sudo apt install gcc
```

```
sudo apt-get install gfortran
```

```
sudo apt-get install ghostscript
```

Once installed, check the version

```
gfortran -v
```

If it is 9.3.0 or above, then this should be sufficient. If not, you may need to update your system:

```
sudo add-apt-repository ppa:ubuntu-toolchain-r/test
```

```
sudo apt update
```

```
sudo apt dist-upgrade
```

```
sudo apt-get install gfortran-10
```

```
sudo update-alternatives --install /usr/bin/gfortran gfortran  
/usr/bin/gfortran-10 100
```

These instructions also apply to WSL (Windows Sub Linux) and other Debian-based flavours.

2.1.2 CENTOS

```
sudo dnf groupinstall "Development Tools"
```

or

```
sudo yum install gcc-gfortran
```

2.2 MAC OS X

The current release for Mac OS is compiled for x86_64 architecture and it is not yet ready for the native M1 chip. However, **PhreePlot** can be run on an M1 machine using the x86_64 emulation mode outlined below.

The actual binary (pp_1-OSX-11.5-gfortran_intel_15968.tar.bz2) release of **PhreePlot** is based on the latest version of OS Big Sur (11.0 to 11.6).

After **PhreePlot** has been installed, untar by double-clicking or by using the command:

```
tar -xvf pp_1-OSX-11.5-gfortran_intel_15968.tar.bz2
```

If Xcode is not already installed, you need to install it with root permission using the sudo command.

Xcode can be downloaded using 'Command Line Tools from Terminal' (in Utility folder):

```
sudo xcode-select --install
```

This can be installed directly from the Apple Developer website (Mac Ports) or from the Mac App Store. In the latter case, the complete Xcode will be installed. This is very large and for non-developers is unnecessary and a waste of hard disk space. Therefore, the Mac Ports approach should be sufficient.

macports can be downloaded from <https://www.macports.org/install.php>, and is an open source package to compile, manage and upgrade packages for Mac OS X, and will allow the packages and libraries included with **PhreePlot** to run and to produce images:

```
sudo port install gcc11 +gfortran
```

NOTE: +gfortran adds the Fortran compiler to gcc since it is not included by default.

Ghostscript can also be installed if necessary:

```
sudo port install ghostscript
```

If no development tools are already installed, then all the necessary code dependencies will be automatically installed. This installation can take several minutes.

As an alternative to macports (for intel version but mandatory for Apple Silicon), the homebrew installer can be used. This requires a slightly different sequence of commands:

```
brew install gcc
```

```
brew install ghostscript
```

```
brew install gfortran
```

Once the packages are installed it may be necessary to declare a global variable to make them available, e.g. in a terminal:

```
export DYLD_LIBRARY_PATH="/usr/local/lib/gcc/11/"
```

A slightly different procedure is needed on an M1 machine depending on the shell used:

```
env /usr/bin/arch -x86_64 /bin/zsh
```

or

```
env /usr/bin/arch -x86_64 /bin/bash
```

If a Bourne shell is preferred, then

```
export DYLD_LIBRARY_PATH="/usr/local/lib/gcc/11/"
```

should enable **PhreePlot** to run.

The **PhreePlot** distribution contains a precompiled version of the `libPhreeqcRM`. a library, but this can be compiled from source after downloading from the USGS **Phreeqc3** website <https://www.usgs.gov/software/phreeqc-version-3> and following the **PhreeqcRM** readme notes, e.g. after downloading

```
cd /directory_where_phreeqcrm_is/phreeqcrm-3.7.3-15968
```

declaring the `gfortran` version installed on your mac (i.e. `gfortran-mp-11`) and OS version (i.e. 11.5).

TARGET is detectable with the command:

```
sw_vers
```

then

```
export TARGET=11.5
```

```
export VERS=mp-11
```

```
export MACOSX_DEPLOYMENT_TARGET=${TARGET}
```

```
rm -fr build
```

```
mkdir build
```

```
cmake -S . -B build -D CMAKE_C_COMPILER=`which gcc-${VERS}` -D  
CMAKE_CXX_COMPILER=`which g++-${VERS}` -D CMAKE_FC_COMPILER=`which  
gfortran-${VERS}` -D CMAKE_SYSTEM_NAME=Darwin -D CMAKE_SYSTEM_PROCES-  
SOR="x86_64" -D CMAKE_OSX_ARCHITECTURES="x86_64" -D CMAKE_OSX_DEPLOY-  
MENT_TARGET=${TARGET}
```

```
cd build
```

```
make
```

3. Installation and Setup

Copy the appropriate `pp*.bz2` archive to your preferred directory.

Unzip using, e.g. for Ubuntu

```
tar -xvf pp_1-Linux-gfortran_10_intel_15968.tar.bz2
```

When unzipped, it will create a directory 'phreeplot' in your current directory containing the shell script `phreeplot.sh` and below this, the following sub-directories:

<code>/bin</code>	contains the <code>pp</code> binary
<code>/demo</code>	contains a number of demo examples
<code>/doc</code>	contains the documentation
<code>/lib</code>	contains <code>libPhreeqcRM.a</code>
<code>/system</code>	contains various files used by PhreePlot

PhreePlot makes use of two environment variables, `PHREEPLOT` and `PHREEPLOT_PATH` to specify the location of files. If the `export` command is available, the can be set with:

```
export PHREEPLOT=/location of the installation, specifically of the system directory/.
```

```
export PHREEPLOT_PATH=/location of the pp binary file/.
```

`PHREEPLOT_PATH` can be added to your `PATH`, as in the script below, making the **PhreePlot** executable, `pp`, accessible from any location:

```
export=$PHREEPLOT_PATH:$PATH
```

`phreeplot.sh` declares these variables and should be executed as

```
source phreeplot.sh
```

or alternatively, a single line could be added to your `.bashrc` file such as:

```
source /directory_where_phreeplot_is/phreeplot/phreeplot.sh
```

so that the file will be executed each time a terminal is opened.

Note that `phreeplot.sh` prepends the executable path to the beginning of the existing `PATH` without deleting any other possible occurrences of the path.

The libraries, and their versions, required to run a given distribution can be found using

```
ldd pp
```

or its equivalent.

If the `export` command is not available (as with non-bash shells) , use:

LD_LIBRARY_PATH "/directory/you/choose"

Finally, an internet search with any error messages can often be useful.

4. Testing and running

If you have sourced the phreeplot.sh script, enter

```
pp -v
```

If the installation was successful, and if it was installed in say the /home/ directory, you should see something like (versions and dates may differ):

```
Usage:  pp input_file_name [other command line parameters...]
```

```
PhreePlot version:                1 (Linux-x64) 09:05:14 14 Dec 2021
```

```
PHREEPLOT environment variable:    /home/phreeplot
```

```
PHREEPLOT_PATH environment variable: /home/phreeplot/bin
```

```
Phreeqc version:                  3.7.3-15968
```

```
Ghostscript version and source:    GPL Ghostscript 9.50 (2019-10-15)(in-  
stalled)
```

```
System directory:                  /home/phreeplot/system/
```

PhreePlot is a console application (not a GUI) and expects an input file to be specified on the command line, i.e.

```
pp file.ppi [other command line parameters]
```

where file.ppi is the name of the input file. Other command line parameters are for testing and for possible future use. Scripts can be made to run one or more instances of **PhreePlot** in the normal way.

A test script, test1.sh, can be found in the /demo directory. This runs the demo/test/test.ppi file which produces a predominance diagram for Fe and, if **Ghostscript** is installed, graphical output files in other formats.

```
source ./demo/test.sh
```

A much larger range of examples can be found in the demo directory.

```
cd demo
```

```
source ./demo.sh
```

5. Contacts

PhreePlot general enquiries: David Kinniburgh, david@phreeplot.org

Linux/Mac installation enquiries: Marino Vetuschi Zuccolini, <mailto:marino.zuccolini@unige.it>