

MOMAP

Installation Guide

Version 2020A

MOMAP

Molecular Material Property Prediction Package

Version 2020A

March, 2020

MOMAP Installation Guide

Version 2020A edited by:

Dr. Qikai Li

Released by Hongzhiwei Technology (Shanghai) Co., Ltd
and Z.G. Shuai Group

The information in this document applies to MOMAP version 2020A

CONTENTS

Chapter 1. Installation.....	1
Introduction	1
Installing MOMAP – Basic Procedure	1
Installing MOMAP – In Details	1
Install MOMAP	1
Chapter 2. Licensing.....	5
Generate MOMAP license data	5
Testing the installation	7
Troubleshooting installation problems.....	7

Chapter 1. Installation

Introduction

Here we will describe how to install MOMAP 2020A Package.

We assume that you have the basic knowledge on how to install programs and manage your operating system. If this is not the case, please consult your local system administrator or consult HZW support.

Installing MOMAP – Basic Procedure

Installing MOMAP and configuring the license can be done in a few steps, that is,

1. Install MOMAP package
2. Obtain and install MOMAP license

Installing MOMAP – In Details

MOMAP is mainly targeted to be running in a UNIX-like environment, however, it can also be used on Windows or MacOS environment.

Install MOMAP

First download the MOMAP package from website:

<http://www.momap.net.cn/index.php/download>, by simply clicking the related link or DOWNLOAD button to download the file.

a) Install MOMAP for Linux

The MOMAP package for **Linux** is downloaded as a single zipped installable run file, e.g.,

```
momap-2020A-linux-mpich2.run.gz
```

Unzip the file and add executable attribute to the file, and install MOMAP by running:

```
$ gunzip momap-2020A-linux-mpich2.run.gz
$ chmod a+x momap-2020A-linux-mpich2.run
$ ./momap-2020A-linux-mpich2.run
```

Or more generally, install MOMAP by running:

```
$ sh momap-2020A-linux-mpich2.run
```

Before using the MOMAP, users should first have a read of the files `README` and `INSTALL`, and take `env.sh` as a template to setup one's own running environment if the Bash shell is used.

Modify `env.sh` according to your specific situation, and add:

```
. $MOMAP_ROOT/env.sh
```

to one's `~/.bashrc`.

If MOMAP is to be run under the **Ubuntu** Linux system, before we start to install MOMAP, we need to promote the user rights and make the user to be an administrator. Here, we will show you how to make a user an administrator in Ubuntu through the command line:

```
$ sudo usermod -aG sudo sampleuser
```

You can verify that the user is now in the "sudo" group by checking the groups a user belongs to, through the following command:

```
$ groups sampleuser
```

Log out and log back in for it to take effect, then we can proceed to have a normal MOMAP installation.

In addition, more environment variables can be added to one's `~/.bashrc` as described in the User Guide.

In a typical situation, for example, when using Bash as the shell, at least three lines like the following in your configuration file (`~/.bashrc`, or `~/.bash_profile`, or `/etc/bashrc`) must exist in order to run the MOMAP software:

```
export MOMAP_ROOT=$HOME/MOMAP-2020A
export PATH=$MOMAP_ROOT/bin:$PATH
export MOMAP_LICENSE=$MOMAP_ROOT/license.lic
```

b) Install MOMAP for macOS

For the **macOS** case, the installation process is similar to that in the Linux case, however, we need to install the necessary libraries used in the MOMAP package, and here the **Homebrew** package manager for macOS is used. The installation procedure is as follows:

a) Install **Homebrew** if not installed already

```
$ /usr/bin/ruby -e "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/master
/install)"
```

b) Install GCC 9, MPICH 3.3, FFTW3 and Lapack 3.8

```
$ brew install gcc@9
$ brew install mpich
$ brew install fftw
$ brew install lapack
```

Other settings are similar to those in the Linux situation. Once the above installations are finished, we can use the MOMAP for macOS as in the Linux's case.

c) Install MOMAP for Windows

To install MOMAP for Windows, we need first to install minGW64, Git Bash, Python, and MSMPI as detailed below.

Download **minGW64** at the link: <https://sourceforge.net/projects/mingw-w64/>, and install minGW64 to location C:\mingw-w64.

Note: Remember to change the Architecture to x86_64.

Download **Git for Windows** at the link: <https://gitforwindows.org/>, and install to C:\mingw-w64\Git.

Download **Python for Windows** at the link: <https://www.python.org/downloads/>, and install to C:\mingw-w64\Python37 (version may not be the same).

Download **MSMPI** at the link: <https://docs.microsoft.com/en-us/message-passing-interface/microsoft-mpi>, and install to C:\mingw-w64\msmpi.

Next, we need to modify the environment variable **PATH** in order to let the MOMAP to run properly.

In both Windows 7 and Windows 10, open the **Explorer**, Right click on **This PC (My Computer)**, then Left Click **Properties**, and further Left Click **Advanced system settings**. In the pop-up window, Click the **Advanced** tab, and Left Click **Environment Variables**, then you can add or edit the various environment variables. In our case, we add the following lines to the PATH variable:

```
C:\mingw-w64\mingw64\bin
C:\mingw-w64\msmpi\bin
C:\mingw-w64\Python37
```

The current Git Bash has problem with python, before running python, we need first to add an alias for python:

Open **Git Bash**, edit `~/ .bashrc` and add the following line:

```
alias python='winpty python.exe'
```

Also make a symbolic link to `<installed-dir>/python.exe`:

```
$ ln -s <installed-dir>/python.exe /usr/bin/python
```

Exit and reopen Git Bash, issue command `python` should be okay now.

Once the above MOMAP running environment settings are done, we can further install the **MOMAP for Windows**. For example, put the downloaded file to `D:\`, extract the archived files to the directory `D:\MOMAP-2020A`. Check the file `env-mingw.sh` to see if the environment variable `MOMAP_ROOT` points to the installed directory.

Again open Git Bash, edit `~/ .bashrc` and add the following line:

```
source /d/MOMAP-2020A/env-mingw.sh
cd $MOMAP_ROOT
```

Exit and reopen Git Bash, then we can use MOMAP now.

For example, we would like to do some tests, then go to the directory `D:/MOMAP-2020A/tests` by issuing:

```
$ cd tests
$ ls
$ cd azulene
```

Then we can do our work just as we do in the Linux situation.

Chapter 2. Licensing

Generate MOMAP license data

Before we can use MOMAP package, we need a license. There exist three license options, that is, `Network`, `Single`, and `MacFree`.

- **Network**: Choose this option if you will be using a floating license server.
- **Single**: Choose this option if you will be using node-locked license.
- **MacFree**: Choose this option if you will be using a trial license.

The program to collect license data is `get_license_data.exe`, users can use `--help` option to run the program to get the help message like the following:

```
$ get_license_data.exe --help
MOMAP license data collection utility, version 2020A (2.2.0)

Usage: get_license_data.exe [opts]
  -outfile license.dat           : set output filename, default to 'license.dat'
  -type [Network | Single | MacFree] : set license type, default to 'Network'
  -cat [Purchased | Free]       : set license category, default to 'Purchased'
  -hosts machinefile            : set machine list file, default to rhosts
  -year_inc yy                  : set increment for purchased license, default to 3 years
  -month_inc mm                 : set increment for free license, default to 1 month
  -version x.y.z                : set version number, default to 2.2.0

e.g.:
  get_license_data.exe
  get_license_data.exe -outfile license.dat
  get_license_data.exe -version 2.2.0
  get_license_data.exe -type MacFree -version 2.2.0
  get_license_data.exe -type Network -version 2.2.0
  get_license_data.exe -type Single -version 2.2.0 -hosts hosts
```

If the program is run without any option, it will generate the `license.dat` file for the current node (assumed to be master node) as `Network` type licensing, the MAC information will be automatically filled in as follows:

```

$ cat license.dat
sales_name = 'HONGZHIWEI TECHNOLOGY(SHANGHAI) CO,.LTD';
sales_webSite = 'www.hzwtech.com';
sales_email = 'sales@hzwtech.com';

licenseInput.user_name = 'Change here';
licenseInput.user_researchGroup = 'Change here';
licenseInput.user_institution = 'Change here';
licenseInput.user_email = 'Change here';
licenseInput.user_address = 'Change here';

licenseInput.user_type = 'Group';

licenseInput.product_name = 'MOMAP';
licenseInput.product_moduleName = 'dushin, transport';
licenseInput.product_version = '2.2.0';

licenseInput.license_type = 'Network';
licenseInput.license_releaseDate = '20200331';
licenseInput.license_maintenanceDate = '20230331';
licenseInput.license_expireDate = '20230331';
licenseInput.license_category = 'Purchased';

licenseInput.computer_MAC = {
'94-57-A5-57-D4-00'
'94-57-A5-57-D4-01'
'94-57-A5-57-D4-02'
'94-57-A5-57-D4-03'
};

```

Users need to fill the user information marked with “Change here” with your actual name, group, institution *etc.* as required, and send this `license.dat` file to the email address as shown in `license.dat`.

Later on, a MOMAP license file, for example, `license.lic`, will be sent to you by a sales representative, you simply copy the license file to `$HOME/MOMAP/`, modify `~/.bashrc` and let the environment variable `MOMAP_LICENSE` point to this file if you have not done so:

```

$ export MOMAP_LICENSE=$MOMAP_ROOT/license.lic

```

Testing the installation

To verify that the MOMAP package has been properly installed, and the license is correctly configured and installed, users can run a short test to verify the installation.

Troubleshooting installation problems

If you encounter any problems during the installation, please contact the HZW support.