# VM2
## Version 2.8.2

# Quick Start: Installation

**VeraChem LLC**

VeraChem has been issued a patent **(USPTO Patent No. 8,140,268)** for the VM2 method.

Contact:

For information regarding VM2 software package licensing contact VeraChem LLC at sales@verachem.com

For technical support contact VeraChem LLC at support@verachem.com

For general enquiries contact VeraChem LLC at info@verachem.com

# Quick Start: Installation

## VM2 Package installation

# VM2 Package Installation

## 1. Obtaining the VM2 package, and package choices

### 1.1. Commercial licensing

To obtain the VM2 package for commercial use contact sales@verachem.com.

Commercial licensing available includes one and two year licenses as well as a perpetual license. Multi-site licenses are available.

### 1.2 Trial license

To obtain a trial license for the VM2 package contact sales@verachem.com

Free three-month licenses are available for users to trial the fully functional parallel processor enabled VM2 package.

### 1.3 Academic licensing

To obtain the VM2 package for academic use contact info@verachem.com

Provide your name, position, and institution, and outline in general terms your intended use of the software.

### 1.4 Package choices

A number of choices are available, which range in capability from ligand only calculations in serial processor mode to protein-ligand binding affinity calculations run in parallel processor modes. The following table shows the various packages available and their capabilities:

| VM2 Package | Parallelization | Maximum atom count | |
|---|---|---|---|
| | | Real atoms | Live atoms |
| Ligand only | Serial, MPI | - | 200 |
| Host+ligand | Serial, MPI | - | 600 |
| Protein+ligand | Serial, MPI, MPI+OpenMP, MPI+CUDA, MPI+OpenMP+CUDA | 10000 | 3000 |
| Full suite | Serial, MPI, MPI+OpenMP, MPI+CUDA, MPI+OpenMP+CUDA | 10000 | 3000 |
| Full suite - large | Serial, MPI, MPI+OpenMP, MPI+CUDA, MPI+OpenMP+CUDA | 10000 | 5000 |

## 2. Operating systems and hardware

The VM2 package currently runs on Linux desktops, workstations, and clusters. It can also take advantage of GPU acceleration.

### 2.1. Linux workstations

The serial, MPI, and MPI-OpenMP VM2 packages can be installed any workstation with Intel CPU(s) and two gigabytes of RAM per compute core or more available, which is running Linux kernel 2.6.32 or later e.g. CentOS 6.9+, Ubuntu 14.04+, etc. It is recommended that a minimum of 8 CPU cores is available for computation.

### 2.2. Linux desktops

These VM2 packages can also run on commodity desktop Intel PCs running Linux kernel 2.6.32 or later that have adequate memory, though recommended use would be for smaller calculations (ligand, hosts, host-ligand complexes), with dedicated workstations more suitable for the more computationally demanding protein and protein-ligand complex calculations.

### 2.3. Linux clusters

The MPI and MPI-OpenMP VM2 packages can run across clusters of workstations (or clusters of commodity machines in the case of Beowulf clusters). Given that the MPI parallelization schemes are not communication bound slower Ethernet interconnects are adequate, though parallel MPI also works with the faster InfiniBand interconnects if present.

### 2.4. Linux workstations and clusters with NVIDIA GPU acceleration

The MPI-CUDA and MPI-OpenMP-CUDA VM2 packages can take advantage of NVIDIA GPUs (Fermi and Kepler architectures) for acceleration of parts of its algorithm. This includes use of multi-GPU workstations and clusters of workstations each with multiple GPUs.

### 2.5. OSX

VM2 is not currently available for OSX.

### 2.6. MS Windows

VM2 is not currently available for MS Windows.

## 3. Installation procedure

### 3.1. Download the VM2 package

After downloading the VM2 package and the example set

vcCompChem_<version>.tar.bz2

vcCompChem_<version>_examples.tar.bz2

where version is the major, minor, and sub-minor version numbers. (e.g. 2_7_050), uncompress and untar them in location of your choice, e.g.

tar xvf vcCompChem_<version>.tar.bz2

tar xvf vcCompChem_<version>_examples.tar.bz2

will create the directories

vcCompChem_<version>/

vcCompChem_<version>_examples/

in the directory you are currently in.

### 3.2. License file

Copy your license file, named vm2_license.LIC, into the vcCompChem_<version>/exe directory.

### 3.3. Environment variables for installation

These installation instructions assume the bash shell is being used. Place the following shell commands and environment variable settings in your **.bashrc** file, which should then be sourced prior to running the installation script. You may use another default shell as you wish, as long as the equivalent command/same environment variables are set.

Modify the variable **VCHOME** to reflect the location of the directory resulting from the tar file extraction above.

```
ulimit -s unlimited
export VCHOME=/home/<my_user_name>/vcCompChem_<version>
export VM2HOME=$VCHOME
export VCPYTHON=$VCHOME/exe/vc_python
export VM2PYTHON=$VCPYTHON
```

### 3.4. Requirements for installation

It may be necessary, depending on the Linux flavor being used, to install packages such as tcsh and g77.

zlib-devel.x86_64 might be required to compile python and gcc-c++.x86_64 for the extensions. In most cases these packages will already be installed on the system.

To check for already installed libraries:

CentOS, RHEL:
  yum list zlib-devel
  yum list gcc
  yum list g++

Debian, Ubuntu:
  dpkg -l zlib-devel
  dpkg -l gcc

### 3.5. Installation script

The following sequence of commands should complete the installation.

```
cd vcCompChem_<version>
cd build
./install_vcCompChem.sh
```

The installation will take several minutes. At the conclusion of the installation steps an automated test set will run, which may also take several minutes to complete.
If any of the automated tests fail, relevant information will be found in the log files they generate in the vcCompChem_<version>/tests directory. One common issue is that the VCHOME and/or VCPYTHON environment variable(s) are not set or set incorrectly. Check this by typing:

```
echo $VCHOME

echo $VCPYTHON
```

Please contact VeraChem for support at support@verachem.com if you have difficulties with installation.

## 4. Installed VM2 package structure

The installed VM2 package directories of interest are:

$VCHOME/documentation
$VCHOME/exe
$VCHOME/lib
$VCHOME/tests

The documentation directory contains a PDF of the package manual and a text file containing the installation directions. The exe directory contains helper software tools and the VM2 executables themselves.

### 4.1. Helper tools

A set of helper command line software tools are present in the $VCHOME/exe directory. Currently, the most useful of these are:

#### 4.1.1. VCharge : assignment of partial atomic charges

VCharge provides fast, easy access to accurate partial charges for virtually any drug-like compound. As input it requires an sdf/mol file. In addition to the Linux command line version supplied with this package, a GUI version is available.

#### 4.1.2. VConf : 2D to 3D and small molecule conformational search

VConf is a standalone conformational search application, which processes an SD file of drug-like compounds containing an initial 2D or 3D conformation of each molecule. In addition to the Linux command line version supplied with this package, a GUI version is available.

#### 4.1.3. prm2top : AMBER formatted input data files to VM2 input data files

This tool given AMBER format .prmtop and .inpcrd files, outputs VM2 input data files – see main user's manual Section V.5.1.

#### 4.1.4. psf2top : CHARMM formatted input data files to VM2 input data files

This tool given a CHARMM format .psf file and .sdf/.mol file, outputs VM2 input data files – see main user's manual Section V. 5.2.

#### 4.1.5. mmo2top : Schrodinger mmo file to VM2 input data files

This tool given a Schrodinger .mmo file, output VM2 data files – see main user's manual Section V. 5.3.

### 4.2. VM2 executables

The VM2 executables present in the $VCHOME/exe directory depends on the licensing level – see Section 1.4 above.

| | |
|---|---|
| Ligand only: | VC_CompChemPackage_serial.exe<br>VC_CompChemPackage_mpi.exe |
| Host+ligand: | VC_CompChemPackage_serial.exe<br>VC_CompChemPackage_mpi.exe |
| Protein+ligand: | VC_CompChemPackage_serial.exe<br>VC_CompChemPackage_mpi.exe<br>VC_CompChemPackage_mpi_openmp.exe |

| | |
|---|---|
| | VC_CompChemPackage_mpi_openmp_cuda.exe |
| Full suite: | VC_CompChemPackage_serial.exe<br>VC_CompChemPackage_mpi.exe<br>VC_CompChemPackage_mpi_openmp.exe<br>VC_CompChemPackage_mpi_openmp_cuda.exe |
| Full suite - large: | VC_CompChemPackage_serial.exe<br>VC_CompChemPackage_mpi.exe<br>VC_CompChemPackage_mpi_openmp.exe<br>VC_CompChemPackage_mpi_openmp_cuda.exe |

### 4.3. Supplied libraries

The following run time libraries are supplied in $VCHOME/lib :

| | |
|---|---|
| /intel | Required Intel math, linear algebra, and parallel processing libraries (MPI, OpenMP.) |
| /cuda | Required Nvidia CUDA libraries for running on GPUs. |
| /magma | Required linear algebra libraries for running on GPUs. |

## 5. Environment variables for running validation calculations

The following environment variables must be set before running a calculation.
They can either be set in the user's **.bashrc** or, preferably, within a script used to launch the calculation. Note that the actual values of OMP_NUM_THREADS and MKL_NUM_THREADS will depend on the type of parallel run being requested. See **Sections VI 4.** and **VI 5.** below for examples of different runs and alternatives to **bash** shell scripts e.g C-shell, PBS, SLURM.

```
---------------------------------------------------------------------------------------------------------
ulimit -s unlimited

INTEL_LIBS=$VCHOME/lib/intel
INTEL_MKL_LIBS=$INTEL_LIBS/mkl
INTEL_MPI_LIBS=$INTEL_LIBS/mpi

CUDA_LIBS=$VCHOME/lib/cuda:$VCHOME/lib/magma

LD_LIBRARY_PATH=$INTEL_LIBS:$INTEL_MKL_LIBS:$INTEL_MPI_LIBS:$CUDA_LIBS
export LD_LIBRARY_PATH

PATH=$INTEL_MPI_LIBS:$PATH
export PATH

export OMP_NUM_THREADS=1
export MKL_NUM_THREADS=1
export I_MPI_PIN_DOMAIN=omp
export KMP_STACKSIZE=16m
---------------------------------------------------------------------------------------------------------
```

Since other software besides VM2 may depend on existing MPI and CUDA configurations, care should be taken when setting the variables to ensure that they only affect the environment in which VM2 software is being run.

## 6. Validation tests

A set of validation tests is available in the vcCompChem_<version>/tests directory. A subset of these tests is run automatically after installation, as alluded to above. These tests are a basic confirmation of installation. It is recommended that the user run all the tests appropriate to their intended use of the package (set up pathway type and hardware configurations) to confirm correct installation.

A python script that automates the full set of tests is provided:

run_all_tests.py

This script will run the entire verachem test suite and validate the results.
The command line argument -py will only run python helper tools tests, -vm2 will only run vm2 tests , -c will add cluster tests, and -g will add gpu tests. The default with no arguments will run the python helper tools tests, then vm2 tests, but no cluster or gpu tests.

To run the subset of tests run automatically after installation use:

run_install_tests.py

### 6.1. Helper tools validation tests

The supplied helper tool validation tests check that the file format conversions for the Maestro/Macromodel, AmberTools, and Biovia Discovery Studio set up pathways (see Section V.5.) are functioning correctly.

#### 6.1.1. Maestro/Macromodel pathway

This test is run automatically after installation. To run manually, carry out the following commands, monitor for error messages, and examine log.out for differences with reference values:

cd vcCompChem_<version>/tests/mmo2top/ligand_08

./run.sh

./verify.sh

#### 6.1.2. AmberTools pathway

This test is run automatically after installation. To run manually, carry out the following commands, monitor for error messages, and examine log.out for differences with reference values:

cd vcCompChem_<version>/tests/prm2top/1ke5/ligand

./run.sh

./verify.sh

cd vcCompChem_<version>/tests/prm2top/1ke5/protein

./run.sh

./verify.sh

### 6.1.3. Biovia Discovery Studio Visualizer (DSV) pathway

To run this test the variable VCDSPATH must be set to the location of the CHARMm forcefield files from your installation of Discovery Studio Visualizer. (For the 2016 version on the PC this is DiscoveryStudio_2016/share/forcefield/CHARMm.)

Carry out the following commands, monitor for error messages, and examine log.out for differences with reference values:

cd vcCompChem_<version>/tests/psf2top/1ke5/ligand

./run.sh

./verify.sh

cd vcCompChem_<version>/tests/prm2top/ psf2top/1ke5/protein

./run.sh

./verify.sh

### 6.1.4. VCharge validation test

This test is run automatically after installation. To run manually, carry out the following commands, monitor for error messages, and examine log.out for differences with reference values:

cd vcCompChem_<version>/tests/vcharge

./run.sh

./verify.sh

### 6.1.5. VConf validation test

This test is run automatically after installation. To run manually, carry out the following commands, monitor for error messages, and examine log.out for differences with reference values:

cd vcCompChem_<version>/tests/vconf

./run.sh

./verify.sh

## 6.2. VM2 validation tests

The tests for vm2 are located in vcCompChem_<version>/tests/vm2 . The test mpi_4 is run automatically after installation.

mpi_16/
mpi_4/
mpi_8/
mpi_cuda/
mpi_openmp_8_2/
mpi_openmp_8_4/
mpi_openmp_cuda/

Each test is named for a different configuration of mpi, openmp, and cuda. Most tests include scripts for use with PBS / Torque and when running interactively. The PBS scripts will need to be modified to match your computing environment, queue names, run time limits, etc.

Example output is provided in the reference subdirectory of each test. If you open either **.out** file, the time required for the test on our hardware will be found at the bottom of the file.