VMD Installation Guide

A. Dalke, J. Gullingsrud, W. Humphrey, S. Izrailev, J. Stone, J. Ulrich

Version 1.4

January 6, 2000

Theoretical Biophysics Group[1]
University of Illinois and Beckman Institute
405 N. Mathews
Urbana, IL 61801
VMD WWW home page: `http://www.ks.uiuc.edu/Research/vmd/`

Description

This document describes how to install one of the precompiled releases of VMD and contains instructions on how to compile VMD from the source code release.

[1] http://www.ks.uiuc.edu/

1 Credits and Program Reference

The authors request that any published work or images created using VMD include the following reference:

Humphrey, W., Dalke, A. and Schulten, K., "VMD - Visual Molecular Dynamics" *J. Molec. Graphics* **1996**, *14.1*, 33-38.

VMD has been developed by the Theoretical Biophysics group at the University of Illinois and the Beckman Institute. The main authors of VMD are A. Dalke, J. Gullingsrud, W. Humphrey, S. Izrailev, J. Stone, J. Ulrich. This work is supported by grants from the National Institutes of Health (grant number PHS 5 P41 RR05969-04), the National Science Foundation (grant number BIR-9423827 EQ), and the Roy J. Carver Charitable Trust.

2 Copyright and Disclaimer

3 Registering VMD

VMD is made available free of charge for non-commercial use to all interested users (but please see the Disclaimer below). We would like to request that you register with us that you are using VMD. This is so that we can maintain some idea of the number of users of the program and so that we know who to contact about program updates, bug fixes, etc. To register, please fill out the form at

`http://www.ks.uiuc.edu/Research/vmd/VMDregistration.html`.

We appreciate your help in this service.

4 Acknowledgments

The authors would particularly like to thank those individuals who have contributed improvements to VMD in the form of new features or entire replacement codes for old features. Special thanks go to Andrew Dalke, Paul Grayson, and Charles Schwieters for their VMD contributions. The entire VMD user community now benefits from your contributions.

The authors would like to thank the members of the Theoretical Biophysics group, past and present, who have helped tremendously in making suggestions, pushing for new features, and trying out quite often very bug-ridden code. Tom Bishop, the original (and for some time only) VMD user, has been a driving force in suggesting (well, demanding) new features and bug fixes. As well, thanks go to Alexander Balaeff, Ivo Hofacker, Xiche Hu, Barry Israeliwitz, Dorina Kosztin, Ilya Logunov, Jim Phillips, Ari Shinozaki, Svilen Tzonev, Willy Wriggers, Dong Xu, Feng Zhou, and our agent in the field, Daniel Barsky. Thanks also to all of you who have tried out the program.

Many external libraries and packages are used in VMD, and the program would not be possible without them. The authors wish to thank Jon Leech for the code to compute the uniform point distributions; T. C. Zhao and Mark Overmars, authors of the excellent XFORMS library; John Ousterhout and the other authors of the Tcl, Tk, TclX, TkX and Tcl-DP packages; the authors of the VRPN library from the University of North Carolina; Amitabh Varshney, author of SURF, also from UNC; Dmitrij Frishman at EMBL for developing STRIDE; Jack Lund for the url_get perl script; and Ethan Merrit from the University of Washington for developing the algorithm for drawing ribbons.

We also received invaluable assistance from people who got the source code and sent in patches and explicit bug reports. The VMD developers would like to thank Axel Berg, Andrew Dalke, Rick Kufrin, Joe Landman, Clare Macrae, Lukasz Salwinski, Stephen Searle, Charles Schwieters, Michael Tiemann, Raymond de Vries, and Simon Warfield for their bug fixes and correspondence.

5 Obtaining VMD Source and Binary Distributions

The VMD source code and binary distributions can be obtained after registering at the VMD web page. Download the appropriate distribution file with your web browser. For Windows binary distributions are self extracting, so once the distribution file is downloaded, proceed to the installation directions below.

For source distributions and Unix binary distributions, uncompress and untar the file. This will produce a subdirectory named `vmd-1.4`. Unless otherwise specified, all references to VMD code will be from this subdirectory, so `cd` there.

6 Installing a Pre-Compiled Version of VMD

To install the pre-compiled Windows version of VMD, simply run the self-extracting executable, and it will start the VMD Windows installer program, which includes built-in help.

To install the pre-compiled Unix version of VMD, then only three steps remain to be done after you uncompress and untar the distribution.

- Edit the `configure` script. If necessary, change the following values:

```
   $install_bin_dir

This is the location of the startup script 'vmd'.  It should
be located in the path of users interested in running VMD.

   $install_library_dir

This is the location of all other VMD files.  This includes
the binary and helper scripts.  It should not be in the path.
```

- Next generate the Makefile based on these configuration variables. This is done by running `./configure` .
- After configuration is complete, `cd` to the `src` directory and type `make install`. This will put the code in the two directories listed above. After this, you just type `vmd` to begin, provided that `vmd` is in your path.

7 What to Do If It Doesn't Work

If you are running a VMD binary which has been built with a native OpenGL implementation (i.e. not Mesa), you should make sure that you have the vendor provided OpenGL runtime libraries and the X server extensions correctly installed. SGI systems normally have the GL and OpenGL runtime support installed on them. Sun, HP, and IBM systems do not come with OpenGL support by default. If you don't have the OpenGL runtime libraries for these systems, they can be downloaded for free from the Sun, HP, and IBM web sites respectively. Each of the vendor's OpenGL implementations generally include "install check" programs which verify the correct installation and operation of the OpenGL libraries.

- Sun's OpenGL WWW site is at `http://www.sun.com/solaris/opengl/`
- HP's OpenGL WWW site is at `http://www.hp.com/unixwork/products/opengl.html`
- IBM's OpenGL WWW site is at `http://www.austin.ibm.com/software/OpenGL/`

We suggest that you check that you are doing everything correctly, and if it still doesn't work, report the problem by e-mail to `vmd@ks.uiuc.edu`. We'll try to help you.

8 Compiling Your Own Version of VMD

If for some reason you want to recompile VMD, then you will need to read the rest of this document. Most users will want to use the binary distributions we provide since they have been thorougly tested prior to release. It may be necessary for you to compile your own version of VMD in cases where we do not provide a binary for your platform, or when the provided binaries will not run correctly with a particular version of your operating system.

9 Working with the Configure Script

You must configure the compile time options you wish to use. These are set with the `./configure` script which produces Makefiles for the `src` and `$ARCH` directories. After doing the configuration you will do the make. The configuration can be set in two ways:

- On the command line, as in:

  ```
  ./configure IRIX5 GL TK XFORMS
  ```

  This will save the options to the file `./configure.options` so the next time you want to regenerate the Makefiles, you need but do

  ```
  ./configure
  ```

- Alternatively, you may edit the file `./configure.options` yourself. You may wish to keep your settings in several configure.options files if you intend to compile VMD on multiple platforms. This is a convenience which can save time when you want to recompile VMD more than once.

  For an SGI version one would do something like:

  ```
  cp ./configure.options.IRIX5.opengl ./configure.options
  ./configure
  ```

  Several configure.options files are included in the VMD distribution.

After you've set the options in the file, run `./configure` to propagate the new definitions to the Makefiles! The full list of compile-time options set by the configure script are:

- Which OS are you running? VMD can be compiled under several operating systems. The configure time options for these OSs are:

```
  Option:    Tested with:
      AIX4 -- IBM AIX 4.x
    HPUX10 -- HPUX 10.20
     IRIX5 -- SGI IRIX 5.3, SGI IRIX 6.x
     IRIX6 -- SGI IRIX 6.5.4
     LINUX -- RedHat Linux 5.2
  SOLARIS2 -- Solaris 2.6 Sparc
SOLARISX86 -- Solaris 7 x86
```

- Which type of display graphics will you use? There are four options which will be carefully explained below:

```
    Option:    Graphics Support:
     OPENGL -- OpenGL graphics library (overall best choice)
       MESA -- Free, Portable, Open Source, OpenGL workalike
         GL -- SGI's Iris GL graphics library (use for older systems)
 NOGRAPHICS -- Graphics support disabled
```

  OPENGL – VMD compiles with native OpenGL for DEC, HP, IBM, Sun, and SGI. OpenGL offers the best rendering performance on new systems, and will be the best supported graphics option for VMD as time goes on. On the newer SGIs, GL is implemented via OpenGL, so you should compile for OpenGL on these platforms. If you have a mix of GL and OpenGL-based machines in the same environment and want to get the best performance out of each, you'll have to do a bit of hackery to start the appropriate binary. Our suggestion is to compile and install one version as GL, and compile the OpenGL and rename the binary vmd_IRIX5_opengl. You can place that binary in the VMDDIR and modify the 'vmd' startup script to recognize which machine is running and start the appropriate application. Contact your vendor or support service to get OpenGL for your system.

  MESA – Mesa is a free OpenGL compatible library which can be used on systems which have no native OpenGL implementation. Mesa performs all of its rendering in software, and may run significantly slower than a native OpenGL implementation. The one exception to this is the 3dfx Voodoo boards running on Linux, for which Mesa does provide hardware accelerated rendering. With few exceptions, a native OpenGL implementation will provide the best rendering performance. We have compiled VMD with MESA 3.0 on all of the tested OSs. The Mesa library is available for free via ftp from `ftp.mesa3d.org` in the `mesa` directory. Mesa is also mirrored on sunsite in the directory `pub/packages/development/graphics/mesa`. See the web address
  `http://www.mesa3d.org/` for more information, and the `README` in `./lib/Mesa` for information about how to get Mesa working with VMD.

  GL – Iris GL is SGI's original hardware accelerated graphics API. Iris GL is no longer widely used now that OpenGL is available for all interesting platforms, but it is still supported in VMD. The Iris GL graphics support works with IRIX5. On very old SGI machines, OpenGL is emulated via GL, so you should probably compile for GL on these machines. To compile for GL you must have the appropriate header files and libraries installed on your system.

- VMD can be compiled to run in the CAVE. If you don't know what the CAVE is, see `http://evlweb.eecs.uic.edu/EVL/VR/` . To compile in the CAVE you will need to get the appropriate CAVE library. There are two, the GL (`libcave.a`) and OpenGL (`libcave_ogl.a`). VMD will compile in the correct one based on the graphics option you chose (GL or OPENGL).

  To get the CAVE library, please contact Pyramid Systems, Inc. The Pyramid WWW page is at `http://www.pyramidsystems.com/`. We cannot distribute the CAVE libraries, but the standard SGI binary distribution version is compiled with the CAVE library. To use that option, start VMD with the run-time option `-dispdev cave`.

  The configuration options are:

```
Option:    Graphics Support:
   CAVE -- include run-time support for CAVE display (GL and OpenGL).
```

- Which kind of GUI do you want to use? (The GUI is nearly independent from the graphics display method.) They are:

```
Option:    GUI Support:
 XFORMS -- the standard (soon to be "old") XForms GUI
   FLTK -- the in-development FLTK GUI
     TK -- (not yet supported) a rewrite of the GUI in Tk
```

  To date the GUI for VMD has been XForms or its GL-based predecessor, FORMS. In version 1.3 the GUI is still XForms based but we plan to convert to FLTK or Tk in future versions. A Tk based GUI could allow user-scripts to create or modify the display on the fly; like drawing an RMSD plot in a new 2D graphics window. If you want to try out Tk, go ahead, but there will be no GUI or mouse control.

  You should compile with XForms, in which case you need to get the pre-built binaries from `http://bragg.phys.uwm.edu/xforms/` or from the ftp site `ftp://einstein.phys.uwm.edu/` . There are instructions in `./lib/xforms/README` describing what you need to do.

  VMD was compiled against Tcl/Tk version 8.0.4 and TclX version 8.0.4 . You must have the 8.0 or better versions of both packages. The latest versions can be found at:

```
Tcl/Tk 8.0.4 is at http://www.scriptics.com/
TclX 8.0.4 is at ftp://www.neosoft.com/pub/tcl/TclX
```

  Because Tcl and TclX are very useful programs, we make the assumption that these will be installed on your system, so there is no `./lib/tcl` directory. To configure VMD to use the appropriate Tcl/TclX directory, you may have to edit `./configure` and change a few variables. The current settings look in `/usr/local/[include,lib]` unless the corresponding environment variable is set. The variables are:

  From the configure script:

```
################ Tcl / Tk
# location of TCL library and include file.
# If left blank, standard system  directories will be searched.
```

```
$stock_tcl_include_dir=$ENV{"TCL_INCLUDE_DIR"} || "/usr/local/include";
$stock_tcl_library_dir=$ENV{"TCL_LIBRARY_DIR"} || "/usr/local/lib";

# location of Tk (for TK option)
$stock_tk_include_dir=$ENV{"TK_INCLUDE_DIR"} || "/usr/local/include";
$stock_tk_library_dir=$ENV{"TK_LIBRARY_DIR"} || "/usr/local/lib";

# location of TclX
$stock_tclx_include_dir=$ENV{"TCLX_INCLUDE_DIR"} || "/usr/local/include";
$stock_tclx_library_dir=$ENV{"TCLX_LIBRARY_DIR"} || "/usr/local/lib";
```

Either change the default values or override them with the environment variables. As a note, if compiling TclX without Tk, you'll need use the option `./configure --with-tk=NO`.

- VMD uses Amitabh Varshney's SURF program to compute the molecular surface. This is actually an external program that VMD calls; it is not compiled into VMD. You will need to compile SURF yourself to use option. The configuration options are:

```
SURF -- add the external call to SURF (uses DrawMolItemSurf.\{Ch\}
        and adds the option to the graphics pop-up list)
```

  There is no reason not to compile SURF. It is actually distributed (with permission) as part of the VMD distribution. To compile surf, go to `./lib/surf` and follow the instructions in `README.VMD` (the `README` file if from surf itself). The end result will be a program named `surf_$ARCH` in that directory. Leave it there.

  For the main SURF distribution, see
  `ftp://ftp.cs.unc.edu/pub/projects/GRIP/SURF/surf.tar.Z`

- In addition to the CAVE library, VMD can use the VRPN library to get information about various 3DOF and 6DOF spatial trackers and other input devices. Our standard VMD builds now enable VRPN by default on most platforms. The configuration options for using VRPN input devices are:

```
VRPN -- (not fully supported yet) Use the VRPN library to support
        local and remote trackers and other input devices.
```

- VMD can connect to NAMD to perform interactive simulations; the protocols related to this connection are referred to as Interactive Molecular Dynamics (IMD). The configuration option for IMD is enabled by default:

```
IMD -- use IMD for doing remote simulations
```

- There are several compilation flags for VMD. They are:

```
DEBUG or NODEBUG
```

When turned on, these print all "msgdebug" messages. There are a lot of them and they really slow things down. (BTW, you can tell who wrote which sections of code by who uses msgDebug :) Also, set the -g flag when compiling. By default, VMD is compiled with NODEBUG. If you only want to add the -g flag, edit the configure script so the compiler is defined as "CC -g" (or "g++ -g").

```
SILENT or NOSILENT
```

The default is SILENT, in which case you don't see the details of the compilation as it suppresses the make command echo. All this does is add or not add the line ".SILENT" near the top of the produced Makefile.

10 Building Libraries for VMD

Now, a detour. There are few libraries you will have to build. These are part of the standard VMD distribution but not part of VMD proper. They are:

10.1 STRIDE

STRIDE is used by VMD to compute the secondary structure given the protein 3D coordinates. The appropriate STRIDE binary is included in the VMD binary distribution. To compile it yourself, see the web site at
`http://www.embl-heidelberg.de/stride/stride_info.html` for information on how to get the source and see `./lib/stride` for information on how to use it with VMD.

Change line 43 of `stride.h` from

```
#define MAX_AT_IN_RES             50
```

to

```
#define MAX_AT_IN_RES             75
```

because there are many structures with non-standard residues containing more than 50 atoms. Change line 96 of `stride.c` from

```
return(SUCCESS);
```

to

```
return(0);
```

since a program should return 0 if everything ended correctly.

10.2 Babel

Babel is used to convert different file formats automatically into PDB files(s) before reading them into VMD. Babel is a freely available general purpose molecular file conversion program. See the web site `http://www.eyesopen.com/babel.html` for more information about it. This program must be available on the path somewhere or you can set the BABEL_BIN environment to point directly to the binary.

10.3 Renderers

VMD supports many types of rendering formats. You may want to get the packages associated with them. They are (among others) Raster3D, Radiance, POV-Ray, and Tachyon. Information about these and other programs can be found at:
`http://www/Research/vmd/allversions/supported_output.html`.

10.4 xmgr

The Labels form allows you to call an external program to graph a selected geometry value over time. The default program, which we suggest using, is xmgr, available from
`ftp://plasma-gate.weizmann.ac.il/pub/xmgr4/`

11 Working with the VMD Source Code

Okay, so you've set the compile time options and run `./configure`, right? The next step would be to check out the latest versions of the files from RCS or CVS, but odds are you don't have your copy of VMD under revision control. In the future we intend to provide an anonymous CVS service to allow VMD users to track developments in the VMD source tree.

Note that the configure command also makes the subdirectory `$ARCH`.

Next you'll need to make the dependency list INSIDE THE SRC DIRECTORY. This is done with

```
  cd ./src
  make depend
  make
```

However, the current implementation needs `gcc` to get the list of include files needed by the different source files. The default Makedata.depend files for the different platforms is available in the appropriate binary distribution. After the dependencies are generated, run `make`, and the executable and object files will be deposited in the `$ARCH` directory

```
  cd ../$ARCH
(for example
  ../LINUX
  ../IRIX5
etc.)
```

This directory was made when you ran the `configure` script.