MOLMOL

Manual

Reto Koradi
Institut für Molekularbiologie und Biophysik
ETH-Hönggerberg
CH-8093 Zürich
E-mail: molmol@mol.biol.ethz.ch

Spectrospin AG
Industriestr. 26
CH-8117 Fällanden
Fax: (+41) 1 825 96 96
E-mail: molmol@spectrospin.ch

1. Introduction

MOLMOL is a program for displaying, analysing, and manipulating molecules. It was written from scratch, replacing the old CONFOR program originally written by Martin Billeter and later modified by Tai-he Xia, both in the group of Prof. K. Wüthrich at the ETH Zürich. The last version was running on a VAX driving an Evans & Sutherland graphics system. Since this machine is completely outdated by now, it was decided to write a new program running on modern graphics workstations.

MOLMOL was first developed under the name COSMOS, but it had to be renamed due to a name collision with a different program.

It was tried to keep the program as general as possible. However, there are some functions that make it especially useful for studying structures of macromolecules obtained by NMR.

The program is freely available by anonymous ftp from one of the following servers, the corresponding directories are given in parentheses:

- `ftp.mol.biol.ethz.ch (/pub/software/MOLMOL)`
- `ftp.spectrospin.ch (/pub/sag/MOLMOL)`
- `ftp.bruker.de (/pub/bruker/MOLMOL)`
- `ftp.bruker.com (/pub/nmr/mirror.bruker.de/MOLMOL)`

All users are required to carefully read and follow the conditions in the file COPYING supplied with the program.

The latest information about the program can always be found on the following web page:

```
http://www.mol.biol.ethz.ch/wuthrich/software/molmol/
```

Please direct comments, suggestions and bug reports to the address on the title page of this manual, preferably by E-mail.

2. About This Manual

This manual does not explain all the details of the commands. The user is encouraged to use the online help system of the program to get this information. A hardcopy of the online manual pages is appended at the end of this manual.

This document is also stored as DVI (`manual.dvi`) and PostScript (`manual.ps`) file in the `man` subdirectory of the installation directory. The reference manual (online manual pages) is stored as `ref.ps` in the same directory, as well as a tutorial in HTML format `tutorial.html`.

It is assumed that the user has a basic familiarity with UNIX for installing the program. For using the program with this manual, the user should be familiar with the basic concepts of using applications with graphical user interfaces.

3. Hardware

MOLMOL is currently running on Silicon Graphics (SGI), Sun, IBM and DEC workstations, as well as on PCs running Linux. On SGI and Sun machines, it makes use of the full graphics capabilities of the machine, but the program can also display on any machine running the X Window System (X11), even on inexpensive black and white X terminals.

4. Installation

See the file `README` for installation instructions.

5. Features

- supported hardware and operating systems
  - Silicon Graphics (IRIX 4.X, 5.X)
  - Sun SPARC (SunOS 4.X, Solaris 2.X)
  - IBM RS6000 (AIX)
  - Digital ALPHA (OSF/1)
  - Linux
  - PCs with Intel Pentium (Windows NT/95)
- display systems
  - GL (Silicon Graphics), including stereo display
  - OpenGL, including stereo display
  - XGL (Sun)
  - X11
  - TTY (no graphics)
- user interface
  - command line with automatic command completion and history
  - pulldown menus (created from ASCII file)
  - popup menu (created from ASCII file)
  - buttons (created from ASCII file)
  - dialog boxes
  - slider box

  - online help (optionally in web broser)
  - log window
  - macro recorder
  - user-defined commands
  - undo (multiple level)
- interactive manipulations (selected subset of molecules)
  - rotations (virtual trackball)
  - translations
  - zoom
- data input/output
  - read residue library
  - write entries for residue library
  - read/write PDB file
  - read list of PDB files
  - read/write DIANA angle file
  - read list of DIANA angle files
  - read/write DG file
  - read list of DG files
  - read SYBYL Mol2 files
  - read/write sequence
  - read/write binary file (with full graphics information)
  - replace coordinates with new ones from PDB file
  - replace coordinates with new ones from list of PDB files
  - replace coordinates with new ones from DG file
  - replace coordinates with new ones from list of DG files
  - read/write file with distance constraints
  - read file with angle constraints
  - read file with chemical shifts
  - save/load orientation
- create entries in residue library (from DG and PDP format)
- structure manipulation
  - create new molecule

  - change molecule name
  - add residue at start or end
  - change residue (mutation)
  - remove residue
  - add/remove atoms
  - add pseudo atom
  - flip ring/methyl atoms for better superposition
  - add bond
  - generate bonds between atoms with close distance
  - remove bonds
  - add/remove angles
  - add/remove angle constraints
  - calculate mean structure
- definition of distances (constraints, H-bonds)
- setting of dihedral angles (construction of helices,...)
- interactive rotation about single bonds
- calculation of superpositions and principal axes
- selection
  - with general expressions
  - interactively
- display possibilities
  - select light position
  - intensity of ambient light
  - select projection (orthogonal or perspective)
  - anti aliasing
  - depth cueing (set fog density)
  - normal bonds or bonds to pseudo atoms
  - bonds: invisible, lines, half lines, cylinders, half cylinders, cones, half cones
  - atoms: invisible, spheres or tetrahedrons
  - distances: invisible, line, cylinder, cone (also dashed)
  - atom labels
  - circles for B-factors

  - constraints
  - violations
  - plates for rings (e. g. for DNA)
  - dottet or shaded, colored molecule surfaces (VdW, solvent accessible, contact), optionally trimmed
  - isosurfaces of electrostatic potential
  - MD trajectories (with lines or cylinders)
  - MD runs (animation)
  - automatic rotation
  - stereo (left, right, side-by-side, cross-eye)
- labels and annotation
  - placement in 2D and 3D (stereo)
  - super-/subscript
  - symbols (greek letters)
  - edit
  - interactive move/resize
  - draw lines, rectangles and circles
- display attributes
  - color
  - line width
  - line style
  - reflection coefficients (ambient, diffuse, specular, shininess)
  - reflection, texture/image, bumpiness (for ray tracing)
  - shading (none, flat, Gouraud, Phong, dots, lines)
  - radius (cylinders, spheres and splines)
  - list current attributes, create macro
- schematic display
  - read secondary structure information from PDB file
  - calculate secondary structure (Kabsch/Sander algorithm)
  - ribbons (for helices, sheets), splines (turns, coil)
  - split/merge elements
  - change start and end position of elements
  - various profiles (elliptical, rectangular, ...)

  - end styles: sharp, soft, arrow
  - splines with variable radius (e. g. for mobility)
  - change width, thickness, size of arrow, ...
  - one, two or variable colors
  - cylinders (can be moved and resized), optionally with arrow head
  - dipolar moments as arrows
  - surfaces for beta-sheets
  - solids (sphere, ellipsoid, cylinder, cone, box)
- plotting
  - PostScript (portrait/landscape, also with stereo, shading, depth cueing and hidden surface elimination)
  - FrameMaker (MIF), for FrameMaker 3 (only 8 colors) and FrameMaker 4 (many colors)
  - Enhanced MetaFile (Windows)
  - image formats: TIFF, JPEG, PNG, BMP
  - POV-Ray (ray tracing)
  - VRML
- calculations
  - global RMSDs with average, standard deviation, minimum and maximum
  - global displacements
  - local RMSDs
  - local displacements
  - RMSD calculations for groups of structures
  - calculate best matching structure parts
  - reduce number of structures (calculate clusters of similar structures)
  - solvent accessible surface of residues
  - electrostatic potential
  - ring current and bond polarization shifts
  - missing atom coordinates (protons, pseudo atoms)
  - angular order parameters
  - angles between helix axes
  - relative lengths of principal axes
  - angles of bonds relative to principal axes
  - find/generate bonds between close atoms

  - find/generate H-bonds
  - find Van der Waals violations
  - find short distances between atoms, generate peak list
  - check distance constraints
  - check angle constraints
  - all functions listed above also for groups (complexes)
  - interactive measurement of bond lengths, bond angles and dihedrals
- figures
  - Ramachandran plots (in color, with allowed regions and angle constraints)
  - angle distribution plots with dots inside bars (in color)
  - angle distribution plots in circles (for MD)
  - distance vs. structure number (for MD)
  - contact maps
  - maps of distance constraints
- history
- command macros

6. Starting

The program is supplied with a shell script that processes options and arguments and sets the necessary environment variables. The following options are recognized by this shell script:

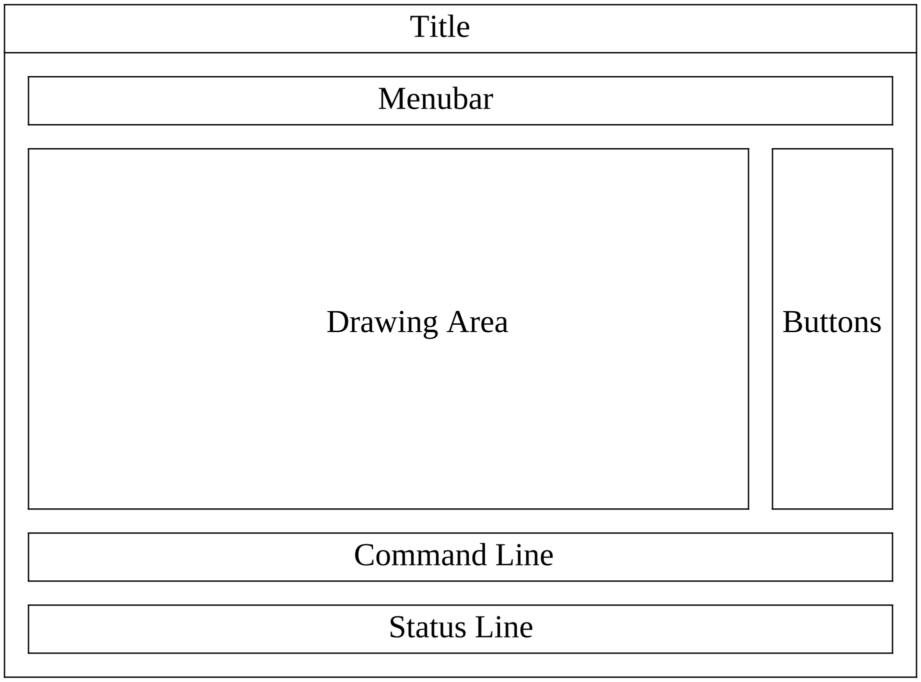
argument	description
`-h`	give list of options
`-t`	TTY interface (no graphics)
`-o name`	alternate input/output device
`-2`	stereo mode (OpenGL only)
`-a`	display all atoms instead of backbone
`-s`	superimpose complete backbone
`-r range`	superimpose backbone of given range
`-f file`	execute macro file (`-` for standard input)

argument	description
files	DG, PDB, Angle or Dump files

If the program is started without names of structure files, it will recover the saved state of the last invocation and neglect all options but `-f macro`.

Besides the listed options, the program will also recognize standard Xt options, like `-bg color` for setting the background color.

7. Main Window Layout



You can switch most of these elements with the `UserInterface` command to adapt the interface to your personal preferences.

8. Executing Commands

There are different ways to enter commands:

- All commands can be executed by selecting them in a pulldown menu. Commands that need arguments will ask for them using a dialog box.

- Users that prefer keyboard input can enter commands on the command line. Parts of commands that are unique are completed automatically. It is possible to also enter the command arguments on the command line. If no or only part of the arguments are given, a dialog box will appear.
- Some frequently used commands can also be found in the popup menu associated with the right mouse button.
- Some frequently used commands and command combinations can be executed by pushing one of the buttons on the right side of the drawing area.
- Some commands have keyboard accelerators. They can be seen in the pulldown menu. These commands can be executed by using the corresponding key combination anywhere in the main window.
- If you have the log window open (use the `UserInterface` command to open it if it is closed), you can repeat previous commands by clicking on them in there.
- The program executes commands that it receives on standard input. This can be used to couple MOLMOL with other programs, e. g. by writing a program (shell script) that generates commands and then piping the output of this program into MOLMOL.

Instead of giving just a constant number, you can always use expressions for numerical command arguments. Commands that act on multiple items (like atoms or bonds) will evaluate this expression for each item seperately. See chapter “Expressions” on page 13 for a description of the expression syntax.

9. Online Help

There is a help text for each command. There are two possibilities to get help:

- Commands that have a dialog for their arguments have a Help button in this dialog. A mouse click on this button will show the help text.
- Executing the `HelpCmd` command puts the program into help mode. The next command that is activated, either from the command line or in a menu, will not be executed, but its help text will be displayed instead.

If you are looking for a command and do not know the name, you can use the `HelpApropos` command to locate it.

10. Interactive Manipulations

10. 1. Rotation

Molecules can be rotated by pressing the left mouse button in the drawing area and then moving the mouse. The virtual trackball model is used.

10. 2. Moving

Molecules can be moved by pressing the middle mouse button in the drawing area and then moving the mouse.

10. 3. Moving/Resizing Text

Text annotations can be modified by pressing the middle mouse button inside the box that is displayed when the text is selected, and then moving it. The box is subdivided into different region with dashed lines. The central region is used for moving the text, the other regions are used for resizing the text in the corresponding direction.

10. 4. Zooming

Zooming is done by pressing the left and middle mouse buttons at the same time. Moving the mouse to the right and/or the top will zoom in, moving it to the left and/or bottom will zoom out.

11. Selection

MOLMOL uses the selection concept wherever possible. This means that the user first selects a set of items, and then executes the command that performs the desired action on the selected items.

There are two ways to make selections:

- Interactive selection. Items (like atoms or bonds) can be selected by clicking on them with the left or middle mouse button. Doing that will normally deselect all other items of the same class. To prevent that, either the Shift or the Ctrl key must be pressed on the keyboard while making the selection. For text annotations there is a difference between the function of the left and middle mouse buttons: the left mouse button only selects texts if their bottom-left corner is the nearest item, while the middle mouse button gives priority to texts and selects them whenever the position is within the text.
- Use commands. Items can be selected by using various commands. The most convenient way to use these commands is the selection dialog box, which can be swiched on with the `DialSelect` command. All these commands take expressions that specify whether an item is selected or not. See chapter “Expressions” on page 13 for a description of the expression syntax.

Unfortunatley there is no way to see what items are currently selected. But the status line will give feedback after a selection (interactively or by a command) is made.

12. Properties

Every item (e. g. atom) can have a (virtually) random amount of properties. A property is a boolean value for each item, which can be either true (set) or false (not set). Each property has a unique name.

Some properties have a special meaning to the program. The user should not try to use them for anything else but for the intended purpose:

all Always set for all items. The user should never try to modify this property, otherwise there may be unpredictable effects.

visible True for all atoms, bonds, distances and primitives that have their display attributes set in a way to make them visible. It is maitained by the program, and should not be modified by the user. It is convenient for selecting only visible items.

selected Set for all selected items. This property is modified by the methods explained in chapter “Selection” on page 12 , most commands act on items that have this property set.

displayed Only items that have this property set will be visible. Setting it to false can be used to make items invisible.

movable Only molecules that have this property set will be modified by interactive manipulations. Setting this property to false for part of the molecules can be used for moving molecules relative to each other.

Some properties are predefined and are calculated whenever a molecule is loaded. They are stored in the file `setup/PropDef` of the MOLMOL installation directory.

The user can define additional properties by using the `DefProp*` (e. g. `DefPropAtom`) commands. This is useful for simplifying the expressions used for selections. Also some commands take property names as arguments, the user can either use one of the predefined properties there, or one defined with `DefProp*`. Consult the online manual page of the `DefProp*` commands for examples of defining and using properties.

13. Expressions

Expressions are valid for all numerical command arguments. They are also used for making selections. The online manual contains a formal description of expressions, while this chapter gives a more informal introduction. The online manual also contains a lot of examples on how to use expressions for selection.

13. 1. Constants

13. 1. 1. Integer Constants

Integer constants are written in the ususal decimal notation.

Examples

- `7`
- `395`

13. 1. 2. Floating Point Constants

Floating point constants are written in the ususal notation, with a decimal point. The scientific notation with an exponent (indicated with the letter `E` or `e` is also supported. The decimal point is optional for numbers with exponent. Since integer values are automatically converted to floating point values wherever a floating point value is expected, the user can also use integer constants.

Examples

- `5.3`
- `23.478e17`
- `3.59e-5`
- `1E3`

13. 1. 3. String Constants

String constants are enclosed in double quotes. If double quotes are used inside a string, they have to be escaped by a `\` (backslash). If a string is used inside a command argument, single quotes also have to be escaped by using `\'` because they are used for enclosing each argument.

Examples

- `"hello"`
- `"Hello World!"`

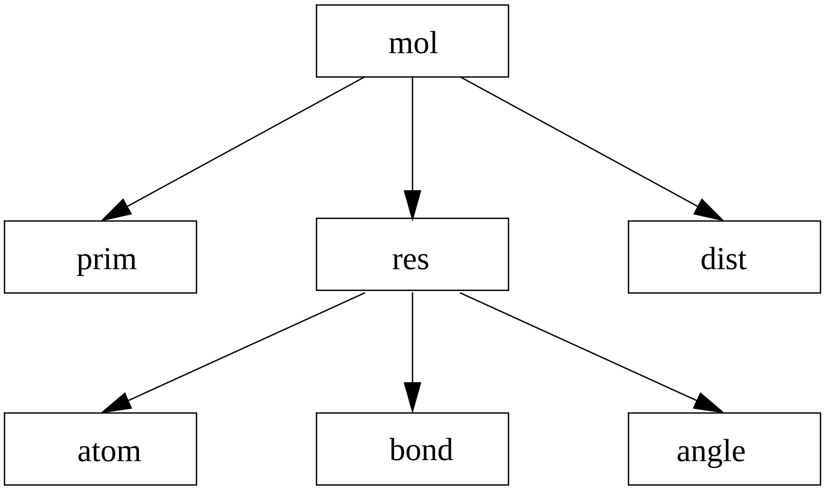
13. 2. Values

Depending on the context, an expression can use values of a data item. If an expression is evaluated for a data item, it can use the values of this item. E. g. an expression that is an argument of a command that operates on atoms can use the values of this atom, like its name. The following is a table of the values that are valid for each data item:

item	value	type	explanation
`mol`	`num`	integer	molecule number
`mol`	`number`	integer	molecule number
`res`	`num`	integer	residue number
`res`	`number`	integer	residue number
`prim`	`num`	integer	primary number
`prim`	`number`	integer	primary number

item	value	type	explanation
`mol`	`name`	string	molecule name
`res`	`name`	string	residue name
`atom`	`name`	string	atom name
`angle`	`name`	string	angle name
`atom`	`shift`	float	chemical shift of atom
`atom`	`bfactor`	float	B factor of atom
`atom`	`vdw`	float	van der Waals radius of atom
`atom`	`charge`	float	partial charge of atom
`atom`	`heavycharge`	float	charges on heavy atoms
`atom`	`avgcharge`	float	averaged charges on heavy atoms
`atom`	`simplecharge`	float	simple charges from setup file
`atom`	`d`	float	distance from reference atom(s)
`angle`	`val`	float	angle
`dist`	`val`	float	distance
`dist`	`limit`	float	limit of constraint
`dist`	`viol`	float	violation of constraint
`dist`	`upl`	bool	true if distance is upper limit
`dist`	`lol`	bool	true if distance is lower limit
`dist`	`hbond`	bool	true if distance is H-bond
`atom`	`attr`	integer	graphics attribute index
`bond`	`attr`	integer	graphics attribute index
`dist`	`attr`	integer	graphics attribute index
`prim`	`attr`	integer	graphics attribute index

Values of an item can be referenced by just using their name. It is also possible to access values of items that are higher in the data hierarchy. This is done in the form item.value, e. g. `res.name` in an expression that is evaluated for an atom. The following figure shows the data hierarchy:



Not all primitives belong to a molecule. For the ones that do not (titles), accessing molecule values will result in null values (`0` for the number, empty string for the name).

For bonds, distances and angles, it is also possible to access the values of the atoms involved. For bonds this is done by using the items `atom1` and `atom2`, for dihedral angles the 4 atoms defining it can be accessed over `atom1`, `atom2`, `atom3`, and `atom4`.

For bonds and distances, it is also possible to access the values of both residues involved. This is done by using the items `res1` and `res2`.

For distances, it is also possible to access the values of both molecules involved. This is done by using the items `mol1` and `mol2`.

Examples

- `name`
- `res.num`
- `atom1.name`

13. 3. Properties

The definition of properties is described in chapter "Properties" on page 12 . They can be used in expressions like the values in section "Values" on page 14 , i. e. by just writing the name or in the form item.property for data items that are higher in the hierarchy. The type of the value is always boolean.

Examples

- `heavy`
- `mol.movable`
- `atom2.bb`

13. 4. Unary Operators

The exclamation mark (!) or not is used for logical negation. It is only valid for boolean values.

The minus sign (-) is used for numerical negation. It is valid for integer and floating point values.

sqrt is used for the calculation of square roots. It is valid for floating point values. If an integers is used, it is converted to floating point.

log is used for the calculation of natural logarithms. It is valid for floating point values. If an integers is used, it is converted to floating point.

Examples

- `- 5.7`
- `- res.num`
- `! bb`

13. 5. Numerical Operators

The usual numerical operators + (addition), - (subtraction), * (multiplication), and / (division) are defined for integer and floating point values. The type of the result is the same as the type of the operators. If integer and floating point values are mixed, the integer value is converted to floating point and the calculation is done in floating point.

For integers, the operator % gives the remainder of the division.

Examples

- `res.num + 2`
- `3 * bfactor`

13. 6. Comparison Operators

The operators = (equal), != (not equal), < (less), <= (less or equal), > (greater), and >= (greater or equal) can be used to compare numerical values. The result is of type boolean.

= and != can also be used to compare strings. It is possible to use the wildcards ? (matches one character) and * (matches any number of characters) for string comparison.

Examples

- `res.num > 30`
- `name != "Q*"`

13. 7. Logical Operators

The operators & (or and) and | (or or) can be used to combine boolean values, with a boolean value as result.

Examples

- `res.num > 30 & bfactor > 3.2`
- `name = "N" or name = "CA" or name = "C"`
- `name = "CB" & visible`

13. 8. Operator Precedence

Operators can be combined at will, as long as the values have the correct type. If operators are mixed in an expression, they are evaluated in the following order:

1. `-` (unary), `!` (`not`)
2. `*`, `/`, `%`
3. `+`, `-`
4. `=`, `!=`, `<`, `<=`, `>`, `>=`
5. `&` (`and`)
6. `|` (`or`)

Operators that have the same precedence are evaluated from left to right.

The evaluation order can be changed by using parantheses. Subexpressions enclosed in parantheses are evaluated first.

Examples

- `2.3 * (bfactor + 1.8)`
- `(res.num < 10 | res.name = "ARG") & name = "C"`

13. 9. Short Notation

For simple selections, there is a shorter notation. It consists of 3 parts, which are all optional:

1. a `#` followed by a list of molecule names and numbers
2. a `:` followed by a list of residue names and numbers
3. a `@` followed by a list of atom names

Spaces are not allowed within such an expression, strings are not quoted. It is possible to give a list of such expressions, separated by spaces. This will select all parts, which corresponds to the logical "or" in the full notation.

Expressions in the short notation can be used within expressions in the full notation, they have a boolean value.

Examples

- `@CA`
- `:10-20`
- `#1-3,5:10-20,25,LYS@N,CA,C`
- `:10@HN :17@HA`
- `heavysc & :13-27`

14. Command Overview

14. 1. Input/Output

`ReadAng`	read DIANA dihedral angle file
`ReadListAng`	read list of DIANA dihedral angle files
`ReadDg`	read DG file
`ReadListDg`	read list of DG files
`ReadPdb`	read PDB file
`ReadListPdb`	read list of PDB files
`ReadSybyl`	read SYBYL Mol2 file
`ReadXyz`	read XYZ file
`ReadSeq`	read sequence
`ReadDump`	read dump file
`ReadLol`	read file with lower limits
`ReadUpl`	read file with upper limits
`ReadAco`	read file with angle constraints
`ReadShift`	read file with chemical shifts
`ReadPot`	read potentials for surface coloring
`ReplaceDg`	replace with DG file
`ReplaceListDg`	replace with list of DG files
`ReplacePdb`	replace with PDB file
`ReplaceListPdb`	replace with list of PDB files
`ReadLib`	read and error check residue library
`WriteAng`	write DIANA dihedral angle file
`WriteDg`	write DG file
`WritePdb`	write PDB file
`WriteSeq`	write sequence
`WriteDump`	write dump file
`WriteLol`	write file with lower limits
`WriteUpl`	write file with upper limits
`WriteLib`	write entries for residue library
`WriteRotation`	write current rotation
`WriteSecondary`	write macro with secondary structure

14. 2. Modify Molecules

`NewMol`	create new molecule
`MeanMol`	create mean molecule
`NameMol`	change molecule name
`FirstMol`	move molecules to front of molecule list
`RemoveMol`	remove molecules
`AddRes`	add residue to molecule
`ChangeRes`	change residue type
`RemoveRes`	remove residues
`AddAtom`	add new atom
`AddPseudo`	add new pseudo atom
`FlipAtom`	flip atoms for better superposition
`RemoveAtom`	remove atoms
`AddBond`	add bond between two atoms
`RemoveBond`	remove bonds
`AddDist`	add distance between two atoms
`AddHbond`	add H-bond between two atoms
`AddLol`	add lower limit between two atoms
`AddUpr`	add upper limit between two atoms
`RemoveDist`	remove distances
`AddAngle`	add dihedral angle
`RotateAngle`	rotate dihedral angles
`SetAngle`	rotate dihedral angles
`RemoveAngle`	remove angles
`AddAco`	add angle constraints
`RemoveAco`	remove angle constraints

14. 3. Primitives

`AddTitle`	add 2D text annotation
`AddText`	add 3D text annotation
`EditText`	edit text annotation
`SizeText`	resize text
`AddDrawobj`	add line, rectangle or circle
`MoveDrawobj`	move draw object

StyleDrawobj	set display style of draw objects
SetNeigh	set sheet neighbours of residues
AddRibbon	add ribbons (schematic drawing)
SplitRibbon	split ribbons
TypeRibbon	set type of ribbons
StyleRibbon	set display style of ribbons
SizeRibbon	set sizes of ribbons
PaintRibbon	set coloring style of ribbons
AddPlates	add plates for rings
SizePlate	set sizes of plates
AddSurface	add surface
TrimSurface	cut away part of a surface
PaintSurface	set coloring style of surfaces
AddIsosurface	add isosurface of potential
AddTrajec	add trajectories
StyleTrajec	set display style of trajectories
AddCircles	add circles around atoms
AddCylinder	add cylinder for schematic drawing
AddDipole	add arrow showing dipolar moment
StyleCylinder	set display style of cylinders
AddSheet	add sheet for schematic drawing
AddSolid	add solid for schematic drawing
ColorPrim	change color of primitives
TintPrim	change secondary color of primitives
LinePrim	set style and width of primitive lines
ShadePrim	set shading of primitives
RadiusPrim	set radius of primitives
NearPrim	set near clipping plane of primitives
MovePrim	move primitive
LengthPrim	set length of primitives
RemovePrim	remove primitive

14. 4. Display

`ZoomAbs`	zoom absolute
`ZoomRel`	zoom relative
`Fullscreen`	switch fullscreen display on and off
`DrawSize`	set size of drawing area
`Stereo`	set stereo mode
`StartRock`	start rock
`StopRock`	stop rock
`StartAnim`	start animation
`StopAnim`	stop animation
`BondMode`	choose bond display mode
`BackColor`	change background color
`Projection`	choose projection
`ClipPos`	set position of clipping planes
`ClipSize`	set distance of clipping planes
`ViewAngle`	set view angle
`Light`	setup of light source
`Fog`	setup of fog (depth cueing)
`Rendering`	set rendering options
`DrawDelay`	set drawing delay
`DrawPrec`	set drawing precision
`PathNames`	set path names

14. 5. Movement

`Center`	change rotation center
`Fit`	calculate structure superpositions
`MoveX`	move molecule in X direction
`MoveY`	move molecule in Y direction
`MoveZ`	move molecule in Z direction
`RotateInit`	undo all rotations
`RotateX`	rotate molecule around X axis
`RotateY`	rotate molecule around Y axis
`RotateZ`	rotate molecule around Z axis

14. 6. User Interface

`UserInterface`	switch elements of user interface
`DialSelect`	switch selection dialog on and off
`DialMol`	switch molecule dialog on and off
`DialColor`	switch color dialog on and off
`DialStyle`	switch style dialog on and off
`DialMeasure`	switch measurement dialog on and off

14. 7. Selection And Properties

`SelectMol`	select molecules
`SelectRes`	select residues
`SelectAtom`	select atoms
`SelectBond`	select bonds
`SelectAngle`	select angles
`SelectDist`	select distances
`SelectPrim`	select primitives
`DefPropMol`	define molecule property
`DefPropRes`	define residue property
`DefPropAtom`	define atom property
`DefPropBond`	define bond property
`DefPropAngle`	define angle property
`DefPropDist`	define distance property
`DefPropPrim`	define primitive property
`ListPropMol`	list all molecule properties
`ListPropRes`	list all residue properties
`ListPropAtom`	list all atom properties
`ListPropBond`	list all bond properties
`ListPropAngle`	list all angle properties
`ListPropDist`	list all distance properties
`ListPropPrim`	list all primitive properties
`SetRef`	set reference atoms
`UndefProp`	undefine property

14. 8. Attributes

`ColorAtom`	set color of atoms
`ColorBond`	set color of bonds
`ColorDist`	set color of distances
`TintDist`	set secondary color of distances
`LabelAtom`	set format of atom label
`LabelDist`	choose label of distances
`LineAtom`	set style and width of atom lines
`LineBond`	set style and width of bond lines
`LineDist`	set style and width of distance lines
`RadiusAtom`	set radius of atoms
`RadiusBond`	set radius of bonds
`RadiusDist`	set radius of distances
`MaterialAtom`	set material properties of atoms
`MaterialBond`	set material properties of bonds
`MaterialDist`	set material properties of distances
`MaterialPrim`	set material properties of primitives
`TextureAtom`	set texture of atoms
`TextureBond`	set texture of bonds
`TextureDist`	set texture of distances
`TexturePrim`	set texture of primitives
`ShadeAtom`	set shading of atoms
`ShadeBond`	set shading of bonds
`ShadeDist`	set shading of distances
`StyleAtom`	set display style of atoms
`StyleBond`	set display style of bonds
`StyleDist`	set display style of distances
`AttrAtom`	list attributes of atoms
`AttrBond`	list attributes of bonds
`AttrDist`	list attributes of distances
`AttrPrim`	list attributes of primitives

14. 9. Plotting

`PlotUnit`	set unit of plot parameters
`PlotPar`	set plot parameters
`PlotPs`	make a PostScript plot
`PlotFm3`	make a FrameMaker 3 plot
`PlotFm4`	make a FrameMaker 4 plot
`PlotMeta`	make a Windows enhanced metafile plot
`PlotTiff`	make a TIFF plot
`PlotJpeg`	make a JPEG plot
`PlotPng`	make a PNG plot
`PlotBmp`	make a BMP (Windows bitmap) plot
`PlotPov`	make a POV-Ray plot
`PlotVrml`	make a VRML plot

14. 10. Calculations

`CalcSecondary`	calculate secondary structure
`CalcRmsd`	calculate RMSDs and displacements
`CalcMatch`	calculate best matching structure parts
`CalcCluster`	reduce number of structures
`CalcSurface`	calculate solvent accessible surface
`CalcPot`	calculate electrostatic potential
`CalcShift`	calculate ring current shifts
`CalcAtom`	calculate missing atom coordinates
`CalcBond`	calculate bonds between close atoms
`CalcHbond`	find H-bonds
`CalcVdw`	find Van der Waals violations
`CalcDist`	find short distances between atoms
`CalcAngle`	calculate angular order parameters
`CalcHelix`	calculate angles between helix axes
`CalcAxis`	calculate lengths of principal axes
`CheckBond`	check number of bonds
`CheckUpl`	check upper limits
`CheckLol`	check lower limits
`CheckAco`	check angle constraints

14. 11. Figures

`FigRamach`	Ramachandran plot
`FigAngles`	show angle distribution bars
`FigCircles`	show angle distribution circles
`FigDist`	show distance vs. structure number
`FigContact`	show contact map
`FigLimit`	show map of distance constraints
`FigOff`	switch from figure to normal display

14. 12. Miscellaneous

`Group`	define molecule groups
`LengthBond`	set length of bonds for display
`HelpViewer`	set viewer for help files
`HelpApropos`	locate commands by keyword
`HelpCmd`	get help on command
`HelpMouse`	show help text about using the mouse
`HelpButton`	show definition of buttons
`HelpProp`	show definition of standard properties
`HelpAbout`	show program version and copyright
`SetUndo`	set the number of possible undos
`Undo`	undo effect of last command
`XMacStand`	execute standard macro
`XMacUser`	execute user macro
`RecordMac`	open macro recorder
`InitAll`	delete and initialize everything
`System`	execute operating system command
`Register`	register as MOLMOL user
`Quit`	quit the program

15. Standard Macros

There are some standard macros in the subdirectory `macros` of the MOLMOL installation directory. They are partly used by the buttons, but can also be executed with the `XMacStand` command, or be used as instructive examples.

color_atom.mac	Set color of selected atoms to "standard" colors (red for O, yellow for S, etc.), e. g. for CPK model.
cpk.mac	Display currently selected parts as CPK model.
ball_stick.mac	Display currently selected structure parts as ball and stick model.
ribbon.mac	Make ribbon drawing of currently selected residues.
sausage.mac	Draw spline with variable radius for bundle of structures.
schem_dna.mac	Schematic display of selected DNA residues.
temper.mac	Set color of selected atoms and bonds depending on B-factor.
black_white.mac	Set material properties for good looking black and white plot.

16. Known Problems

- Minor gaps in soft ends for ribbon style lense.

The latest information about problems with the crurrent release can be reached from the following web page:

```
http://www.mol.biol.ethz.ch/wuthrich/software/molmol/
```

17. Trouble Shooting

This chapters lists a few problems that users might have, and suggests possible solutions.

P: Program crashes on startup.

S: You might have a corrupted `dump` file. Remove the file `molmol/dump` in your home directory and try again.

P: Program starts up, but no menu bar is visible.

S: `MOLMOLHOME` is not properly set. Check the startup script.

18. Environment

The program reads two environment variables. They are set in the startup script, so the user normally does not have to care about them. In case of problems, the system administrator should look at the startup script and make the necessary modifications.

`MOLMOLHOME` must be set to the name of the directory where the program was installed.

`MOLMOLDEV` defines the output device (graphics system). This should be set to `Motif/GL` when running the program locally on an older SGI machine (e. g. Personal Iris), to `Motif/OpenGL` when running on newer SGI machines, to `Motif/XGL` when running the program on a Sun, and to `Motif/X11` for other configurations. When running the program on an SGI and displaying remotely on another SGI, you can also use `Motif/GL` or `Motif/OpenGL`, but in this case you might get better performance with using `Motif/GLD` or `Motif/OpenGLD`. Note that it is much better to run the program locally in most cases. For running the program without graphics, the variable must be set to `TTY/NO`, which is most conveniently done by giving the `-t` switch on the command line.

If you want to use a value different from the default for `MOLMOLDEV`, you can give an alternate value with the `-o` switch on the command line, or you can set the environment variable before calling the startup script.

19. User Files

The program creates a subdirectory `molmol` in the home directory of the user, and stores some files there:

history The command history of the last session. This can be executed as macro (command `XMacUser`).

dump A binary file that describes the complete program state. This file is written whenever the user quits the program, and it is automatically read on startup. It is recommended to remove `dump` (or move it to a different name) before starting work with different molecules.

par Contains values of some program parameters. These values are also stored in the `dump` file, so it is redundant if a `dump` exists. It is used to maintain preferred parameter values of the user when the `dump` is removed.

20. Configuration

All configuration files are stored in the subdirectory `setup` of the MOLMOL installation directory. The system admistrator can make site specific changes in these files.

With the `PathNames` command, each user can select files different from the default ones. For doing this, it is recommended to copy the default file to a place in the home directory and make the desired changes.

The following is a list of configuration files:

name	description
`MenuBar`	names and mnemonics of all pulldown menus

name	description
`*.menu`	entries of all pulldown menus
`Popup`	entries in the popup menu
`Buttons`	definition of buttons
`Valuator`	entries in the valuator box
`amber94.lib`	residue library
`PdbAtoms`	translation table for PDB atom names
`AtomRadius`	data for the commands `MakeBonds` and `CheckBonds`
AtomCharge	charges referenced in expressions by `simplecharge`
`FlipAtoms`	data for the command `FlipAtom`
`RingShift`	data for the command `CalcShift`
`PropDef`	predefined properties
`ColorList`	list of colors for color dialog

If there is a macro with the name `startup.mac` in the subdirectory `molmol` of the home directory of the user, it is automatically executed when the program is started up.