# HARLEM

# (HAmiltonians for Response properties of LargE Molecules)

HARLEM is a multipurpose interactive Molecular Modeling package. It is designed to combine modern electronic structure, statistical mechanics and Machine Learning techniques controlled by a graphical interface to provide an effective theoretical tool to study complex properties and function of macromolecular systems such as biocatalysis, long-rang electron transfer, ion membrane permeation, ligand binding and protein-protein interactions.

HARLEM provides an interface to popular molecular modeling packages such as GAUSSIAN and AMBER, GROMACS, HARLEM is an open developing platform. HARLEM is currently being developed and maintained by Igor Kurnikov (http://harlemprog.org) and members of Maria Kurnikova’s lab at Carnegie Mellon University (http://crete.chem.cmu.edu).

## Why HARLEM?

Yes. This is definitely a pertinent question. Even though software exists that can do a number of sophisticated operations, why is HARLEM suggested? Simply because HARLEM is capable of doing all those operations and more and further improves upon the simplicity of the task. Here is why the principal developer likes to use HARLEM:

 *To overcome shortcomings of the architecture of the existing quantum chemical software*.

 *To achieve a high degree of the program usability*.

 *The program should have a modular structure which permit easy modifications and extensibility of the program functionality*.

 *Interact with other molecular modeling programs smoothly to allow exchange of data*.

## Installation

Although HARLEM is designed to work with both Linux and Microsoft Windows platform, it is easier to install and work with the Microsoft Windows version. In order to install HARLEM follow these instructions:

* Download harlem.zip file from HARLEM download page (http://harlemprog.org).
* Unpack content of harlem.zip file into a directory, for example, c: directory, this should create HARLEM executable harlem.exe and several dll files in the directory C:\HARLEM
  + Subdirectory *C:\HARLEM\residues\_db* with residues and force-field databases (currently files amber\_94\_ff.dat, aminoacids.hlm, cofactors.hlm, nucleotides.hlm, water.hlm)
  + Subdirectory *C:\HARLEM\scripts* with Python scripts
  + Subdirectory *C:\HARLEM\basis* with basis sets in DALTON format
  + Subdirectory *C:\HARLEM\harlem\_manual* containing HARLEM documentation, useful for developers (in case you are interested in developing HARLEM).
  + Subdirectory *C:\HARLEM\examples* with a few input files and scripts to run HARLEM jobs.

The advantage of using the Microsoft Windows is that there is no dependency on the version of various libraries that HARLEM requires, unlike the Linux version, where several libraries are required to be installed. The rationale behind this deliberate design was that HARLEM on Microsoft Windows would effectively act as clients where the user could prepare the various inputs and formatted files required to perform several calculations and perform simple visualization of results on the molecules. The actual calculations are performed actually a Linux cluster. However, the Linux version is developed for more advanced user, who prefers to format and work on Linux clusters. It is recommended that the Microsoft Windows version be used for beginners.

## Front End

The front-end of HARLEM consists of two windows: (1) the HARLEM console and (2) the HARLEM control interface. The HARLEM console acts as an interactive tool that passes on information from the program to the user. It is essentially a detailed status indicator that allows the user to decide the course of action whenever something goes wrong. It also gives essential details on the organization of the current molecule loaded into the view on HARLEM control interface.

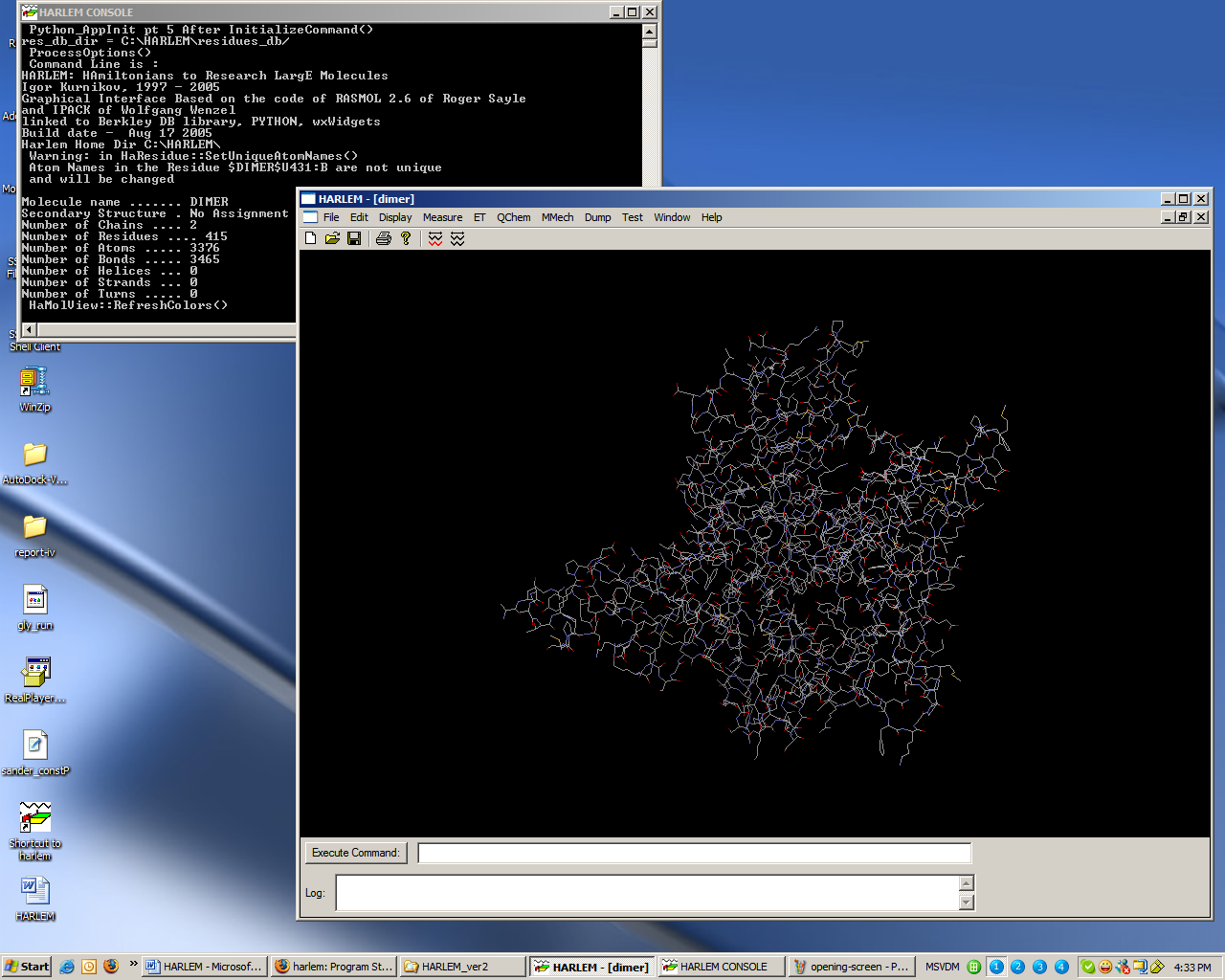
The HARLEM control interface is where all the action occurs. It provides the interactive interface to model all the molecules that the user is interested in. It is composed of two parts, namely, the visualization window (where one can visualize molecules, observe changes made to the molecule and so on) and the command line interface that allows the user to type in commands, one at a time, so that the user can model the system according to his/ her wish.

The rest of the user’s manual for HARLEM is organized as follows. The following section gives an introduction for the command line interface and its use in HARLEM. Through out the manual, several examples are provided to illustrate the power of the command line interface. The next section illustrates several types of calculations that can be performed using HARLEM. Also, the various functions available on the main visualization interface are also explained.

## Command Line Interface

The command line interface extends upon the powerful query mechanism built in with RASMOL molecular viewer package. Apart from that, the mouse controls and the scroll buttons also resemble RASMOL capabilities. The mouse control buttons and their respective functions are shown below in table 1. Most of the mouse control options are of the “click-and-drag” type operations. This helps in selection/ viewing and instant feedback to the user about how the controls are being used.

|  |  |  |
| --- | --- | --- |
| **Number** | **Mouse Control** | **Action** |
| 1 | Left button held down | Rotate X, Y axes |
| 2 | Right button held down | Translate X, Y axes |
| 3 | Shift + Left button held down | Zoom |
| 4 | Shift + Right button held down | Rotate Z axis |
| 5 | Control + Left button held down | Slab pane |



Synchronizing the mouse movements along with the command line interface as well as the main visualization interface will provide the user with a good mechanism to manipulate molecules. The general syntax for a command is as follows:

(<command\_name>)\* <selection>

The commands can be anything legal listed in table 2, and the selection specifies a set of atoms a user is interested in. Usually it is an accepted practice to allow multiple commands to be used followed by the selection. This allows for some nifty operations to be done, which otherwise would have been difficult. The commands are usually interpreted by the command line interpreter inbuilt with the system.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Backbone | Background | Bond | Cartoon | Centre | Clipboard | Colour | Connect |
| CPK | CPKnew | Define | Depth | Dots | Echo | English | Exit |
| HBonds | Help | Label | Load | Molecule | Monitor | Pause | Print |
| Quit | Refresh | Renumber | Reset | Restrict | Ribbons | Rotate | Save |
| Script | Select | Set | Show | Slab | Source | Spacefill | SSBonds |
| Star | Stereo | Strands | Structure | Surface | Trace | Translate | UnBond |
| Write | Zap | Zoom | Overlapmol | Within | Group |  |  |

Most of the commands in the above table are also explained in RASMOL user manual [1] because HARLEM inherits RASMOL interface (Another popular molecular viewer that inherited RASMOL looks and commands is Jmol or JSmol available at the PDB databank).

HARLEM interface provides extensions to the command language for manipulating the molecules too. These commands are mostly attached with the “select” command or “define” command. These two commands essentially allow an intuitive way to think of molecules and manipulate them for calculations.

The actual introduction to most commands can be found in the second part of the manual, which deals exclusively with the use of RASMOL commands. One can also find it online at <http://www.umass.edu/microbio/rasmol/coulson.txt>.

While working with the large molecules, a convenient way to break them up is by using a selection/ query language that suits the user. Hence it is important to recognize the various naming conventions used to uniquely identify an atom within the molecule. This hierarchy can be drawn as follows (as illustrated in figure 2):

Molecule

Chain

Residue

Atom

Atom Name

Residue Name & Residue Number

Chain Name

Molecule Name

Identified by

Identified by

Identified by

Identified by

In HARLEM the molecular structures are also identified by the same convention, and hence an atom like CA, located in residue 1141 chain A and belonging to MET residue and 3BTA molecule will be shown/ identified as in figure 3:

$3BTA

$MET

1143: A . SD

Molecule Name

Residue Name

Residue Number

Chain Name

Atom Name

The RASMOL commands are explained in detail in the next few pages. These commands are illustrated in a tutorial like fashion and many useful resources are available for users who need more help [2 – 5]. A brief summary of all commands and their usage is given below.

### Backbone

The reserved work **backbone** is used as a predefined set and as a parameter to the **set hbond** and **set ssbond** commands. The RasMol command **trace** renders a smoothed backbone, in contrast to **backbone** which connects alpha carbons with straight lines. Wire frame, backbone and strands representations may be displayed with dashed (dotted) lines. This is enabled by allowing the **dash** or **dashes** parameters to the wire frame, backbone and strands commands.

### Background

Syntax: background <colour>

The RasMol **background** command is used to set the colour of the "canvas" background. The colour may be given as either a colour name or a comma separated triple of Red, Green and Blue (RGB) components enclosed in square brackets. Typing the command [**help colours**](http://www.umass.edu/microbio/rasmol/distrib/rasman.htm#help#help) will give a list of the predefined colour names recognised by RasMol. When running under X Windows, RasMol also recognises colours in the X server's colour name database.

The **background** command is synonymous with the RasMol [**set background**](http://www.umass.edu/microbio/rasmol/distrib/rasman.htm#setbackground#setbackground) command.

### Cartoon

Syntax: cartoon <number>

The RasMol **cartoon** The ribbons representation in RasMol has been extended to allow the display of Richardson (MolScript) style Normal protein cartoons. They are currently implemented as thick (deep) ribbons. The easiest way to obtain a cartoon representation of a protein is to use the new cartoon option on the display menu. The cartoon or cartoons command on the RasMol command line represents the currently selected residues as a deep ribbon with width specified by the commands argument. Using the command cartoons without a parameter the ribbons width is taken from the proteins secondary structure, as described in the ribbons command. By default, the C-terminus of beta-sheets are displayed as arrow heads.

This may be enabled and disabled using the set cartoons command. The depth of the cartoon may be adjusted using the set cartoons command. The set cartoons command without any parameters returns these two options to their default values.

### Centre

Syntax: center {<expression>}

centre {<expression>}

The RasMol **centre** command defines the point about which the [**rotate**](http://www.umass.edu/microbio/rasmol/distrib/rasman.htm#rotate#rotate) command and the scroll bars rotate the current molecule. Without a parameter the centre command resets the centre of rotation to be the centre of gravity of the molecule. If an atom expression is specified, RasMol rotates the molecule about the centre of gravity of the set of atoms specified by the expression. Hence, if a single atom is specified by the expression, that atom will remain `stationary' during rotations.

Type [**help expression**](http://www.umass.edu/microbio/rasmol/distrib/rasman.htm#help#help) for more information on RasMol atom expressions.

### Clipboard

Syntax: clipboard

The RasMol **clipboard** command places a copy of the currently displayed image on the local graphics `clipboard'. Note: this command is not yet supported on UNIX or VMS machines. It is intended to make transfering images between applications easier under Microsoft Windows or on an Apple Macintosh.

When using RasMol on a UNIX or VMS system this functionality may be achieved by generating a raster image in a format that can be read by the receiving program using the RasMol [**write**](http://www.umass.edu/microbio/rasmol/distrib/rasman.htm#write#write) command.

### Colour

Syntax: colour {<object>} <colour>

color {<object>} <colour>

color {<object>} <[RGB triplet]>

Colour the atoms (or other objects) of the selected zone. The colour may be given as either a colour name or a comma separated triple of Red, Green and Blue (RGB) components enclosed in square brackets. A typical RGB triplet is [255,255,255] which is the colour white. Typing the command [help colours](http://www.umass.edu/microbio/rasmol/distrib/rasman.htm#chcolours#chcolours) will give a list of all the predefined colour names recognised by RasMol.

Allowed objects are **atoms**, **bonds**, [**backbone**](http://www.umass.edu/microbio/rasmol/distrib/rasman.htm#backbone#backbone), [**ribbons**](http://www.umass.edu/microbio/rasmol/distrib/rasman.htm#ribbons#ribbons), [**labels**](http://www.umass.edu/microbio/rasmol/distrib/rasman.htm#labels#labels), [**dots**](http://www.umass.edu/microbio/rasmol/distrib/rasman.htm#dots#dots), [**hbonds**](http://www.umass.edu/microbio/rasmol/distrib/rasman.htm#hbonds#hbonds), and [**ssbonds**](http://www.umass.edu/microbio/rasmol/distrib/rasman.htm#ssbonds#ssbonds). If no object is specified, the default keyword **atom** is assumed. Some colour schemes are defined for certain object types.

The colour scheme **none** can be applied to all **objects** except atoms and dots, stating that the selected objects have no colour of their own, but use the colour of their associated atoms (i.e. the atoms they connect). This command is especially useful in script files.

**Atom** objects can also be coloured by [**cpk**](http://www.umass.edu/microbio/rasmol/distrib/rasman.htm#cpkcolours#cpkcolours), [**amino**](http://www.umass.edu/microbio/rasmol/distrib/rasman.htm#aminocolours#aminocolours), [**chain**](http://www.umass.edu/microbio/rasmol/distrib/rasman.htm#chaincolours#chaincolours), [**group**](http://www.umass.edu/microbio/rasmol/distrib/rasman.htm#groupcolours#groupcolours), [**shapely**](http://www.umass.edu/microbio/rasmol/distrib/rasman.htm#shapelycolours#shapelycolours), [**structure**](http://www.umass.edu/microbio/rasmol/distrib/rasman.htm#structurecolours#structurecolours), [**temperature**](http://www.umass.edu/microbio/rasmol/distrib/rasman.htm#temperaturecolours#temperaturecolours), [**charge**](http://www.umass.edu/microbio/rasmol/distrib/rasman.htm#chargecolours#chargecolours), and [**user**](http://www.umass.edu/microbio/rasmol/distrib/rasman.htm#usercolours#usercolours). Hydrogen bonds can also be coloured by [**type**](http://www.umass.edu/microbio/rasmol/distrib/rasman.htm#hbondtypecolours#hbondtypecolours) and dot surfaces can also be coloured by [**electrostatic potential**](http://www.umass.edu/microbio/rasmol/distrib/rasman.htm#potentialcolours#potentialcolours).

For more information see colours.

### Connect

Syntax: connect {<boolean>}

The RasMol **connect** command is used to force RasMol to (re)calculate the connectivity of the current molecule. If the original input file contained connectivity information, this is discarded. The command **connect false** uses an extremely fast heuristic algorithmm that is suitable for determing bonding in large bio-molecules such as proteins and nucleic acids. The command **connect true** uses a slower more accurate algorithm based upon covalent radii that is more suitable for small molecules containing inorganic elements or strained rings. If no parameters are given, RasMol determines which algorithm to use based on the number of atoms in the file. Greater than 255 atoms causes RasMol to use the faster implementation. This is the method used to determine bonding, if necessary, when a molecule is first read in using the [**load**](http://www.umass.edu/microbio/rasmol/distrib/rasman.htm#load#load) command.

### Define

Syntax: define <identifier> <expression>

The RasMol **define** command allows the user to associate an arbitrary set of atoms with a unique identifier. This allows the definition of user-defined sets. These sets are declared statically, i.e. once defined the contents of the set do not change, even if the expression defining them depends on the current transformation and representation of the molecule.

### Dots

Syntax: dots {<boolean>}

dots <value>

The RasMol **dots** command is used to generate a Van der Waal's dot surface around the currently selected atoms. Dot surfaces display regularly spaced points on a sphere of Van der Waals' radius about each selected atom. Dots that would are `buried' within the Van der Waal's radius of any other atom (selected or not) are not displayed. The command **dots on** deletes any existing dot surface and generates a dots surface around the currently selected atom set with a default dot density of 100. The command **dots off** deletes any existing dot surface. The dot density may be specified by providing a numeric parameter between 1 and 1000. This value approximately corresponds to the number of dots on the surface of a medium sized atom.

By default, the colour of each point on a dot surface is the colour of it's closest atom at the time the surface is generated. The colour of the whole dot surface may be changed using the [**colour dots**](http://www.umass.edu/microbio/rasmol/distrib/rasman.htm#colour#colour) command.

### Exit

Syntax: exit

The RasMol **exit** command is used to terminate execution of a script (returning to the command line, or the calling script), or of inter-process communication, closing the link between programs.

The [**quit**](http://www.umass.edu/microbio/rasmol/distrib/rasman.htm#quit#quit) command, in contrast, terminates the execution of RasMol itself.

### HBonds

Syntax: hbonds {<boolean>}

hbonds <value>

The RasMol **hbonds** command is used to represent the hydrogen bonding of the protein molecule's backbone. This information is useful in assessing the protein's secondary structure. Hydrogen bonds are represented as either dotted lines or cylinders between the donor and acceptor residues. The first time the **hbonds** command is used, the program searches the structure of the molecule to find hydrogen bonded residues and reports the number of bonds to the user. The command **hbonds on** displays the selected `bonds' as dotted lines, and the **hbonds off** turns off their display. The colour of hbond objects may be changed by the [**colour hbond**](http://www.umass.edu/microbio/rasmol/distrib/rasman.htm#colour#colour) command. Initially, each hydrogen bond has the colours of its connected atoms.

By default the dotted lines are drawn between the accepting oxygen and the donating nitrogen. By using the [**set hbonds**](http://www.umass.edu/microbio/rasmol/distrib/rasman.htm#sethbonds#sethbonds) command the alpha carbon positions of the appropriate residues may be used instead. This is especially useful when examining proteins in backbone representation.

### Label

Syntax: label {<string>}

label <boolean>

The RasMol **label** command allows an arbitrary formatted text string to be associated with each currently selected atom. This string may contain embedded `expansion specifiers' which display properties of the atom being labelled. An expansion specifier consists of a `%' character followed by a single alphabetic character specifying the property to be displayed (similar to C's printf syntax). An actual '%' character may be displayed by using the expansion specifier `%%'.

Atom labelling for the currently selected atoms may be turned off with the command **label off.** By default, if no string is given as a parameter RasMol uses labels appropriate for the current molecule. RasMol uses the label "%n%r:%c.%a" if the molecule contains more than one chain, "%e%i" if the molecule has only a single residue (a small molecule) and "%n%r.%a" otherwise.

The colour of each label may be changed using the [**colour label**](http://www.umass.edu/microbio/rasmol/distrib/rasman.htm#colour#colour) command. By default, each label is drawn in the same colour as the atom to which it is attached. The size of the displayed text may be changed using the [**set fontsize**](http://www.umass.edu/microbio/rasmol/distrib/rasman.htm#setfontsize#setfontsize) command.

The following table lists the current expansion specifiers:

%a Atom Name

%b %t B-factor/Temperature

%c %s Chain Identifier

%e Element Atomic Symbol

%i Atom Serial Number

%m single letter amino acid code

%n Residue Name; three letter code

%r Residue Number

The syntax of RasMol atom expressions allows the selection of individual molecule conformations if present in an **NMR file**. The simplest form of the atom expression is the syntax

::25 to select model 25 from the molecule.

This is equivalent to the atom expression  
model = 25  
as the keyword **model** may now be used in comparison expressions. The most general form of atom expression is now CYS32:A:25.SG which denotes the gamma sulphur of residue cysteine-32 in chain A of model 25.

Individual chains may be specified by the syntax ":A" for chain A, or ":1" for chain 1 (i.e. the wildcard may be dropped from the expression "\*:A"). This may also be extended to NMR models; ":A:4" denotes chain A of model 4, and even more terse means all atoms in all chains of NMR model 4.

### Load

Syntax: load {<format>} <filename>

load {<format>} inline

Load a molecule co-ordinate file into RasMol2. Valid molecule file formats are **pdb** (Brookhaven Protein Databank), **mdl** (Molecular Design Limited's MOL file format), **alchemy** (Tripos' Alchemy file format), **mol2** (Tripos' Sybyl Mol2 file format), **mopac** (mopac file format; either cartesian or z-matrix format), **nmrpdb** (nmr multi-pdb file format), **charmm** (CHARMm file format) or **xyz** (MSC's XMol XYZ file format). If no file format is specified, **pdb** is assumed by default. Only a single molecule may be loaded at a time. To delete a molecule prior to loading another use the RasMol [**zap**](http://www.umass.edu/microbio/rasmol/distrib/rasman.htm#zap#zap) command.

The **load** command selects all the atoms in the molecule, centres it on the screen and renders it as a CPK coloured wireframe model. If the molecule contains no bonds (i.e. contains only alpha carbons), it is drawn as an alpha carbon backbone. If the file specifies less bonds than atoms, RasMol determines connectivity using the [**connect**](http://www.umass.edu/microbio/rasmol/distrib/rasman.htm#connect#connect) command.

The **load inline** command also allows **Storing Atom Co-ordinates in Scripts** to allow better integration with WWW browsers. A load command executed inside a script file may now specify the keyword **inline** instead of a conventional filename. This option specifies that the co-ordinates of the molecule to load are stored in the same file as the currently executing commands. Typically this is used in the command **load pdb inline**, which is followed by a number of RasMol commands terminated by the command **exit**. The **exit** command terminates execution of the current script and returns control to the command line (or the calling script). This means any lines following **exit** are never interpreted by RasMol. These may be used to store atomic co-ordinates in PDB file format. Because in Brookhaven PDB file format, any line not recognised by the parser should be ignored, only lines beginning ATOM, HETATM, TER, etc. are examined. Hence a file may be both a RasMol script and a PDB file simultaneously. This allows both co-ordinate and representation data to be transmitted as a single file. One possible use is a standard RasMol script prefix that may be concatenated with an appropriate PDB file on-the-fly.

### Quit

Syntax: quit

Exit from the RasMol program. The RasMol command [**exit**](http://www.umass.edu/microbio/rasmol/distrib/rasman.htm#exit#exit) has a different function.

### Refresh

Syntax: refresh

The RasMol **refresh** command is used in script files to redraw the local image.

### Renumber

Syntax: renumber {{-} <value>}

The RasMol **renumber** command sequentially numbers the residues in a macromolecular chain. The optional parameter specifies the value of the first residue in the sequence. By default, this value is one. For proteins, each amino acid is numbered consecutively from the N terminus to the C terminus. For nucleic acids, each base is numbered from the 5' terminus to 3' terminus. All chains in the current database are renumbered and gaps in the original sequence are ignored. The starting value for numbering may be negative.

### Reset

Syntax: reset

The RasMol **reset** command restores the original viewing transformation and centre of rotation. The scale is set to it default value, [**zoom 100,**](http://www.umass.edu/microbio/rasmol/distrib/rasman.htm#zoom#zoom) the centre of rotation is set to the geometric centre of the currently loaded molecule, [**centre all,**](http://www.umass.edu/microbio/rasmol/distrib/rasman.htm#centre#centre) this centre is translated to the middle of the screen and the viewpoint set to the default orientation.

This command should not be mistaken for the RasMol [**zap**](http://www.umass.edu/microbio/rasmol/distrib/rasman.htm#zap#zap) command which deletes the currently stored molecule, returning the program to its initial state.

### Restrict

Syntax: restrict {<expression>}

The RasMol **restrict** command both defines the currently selected region of the molecule and disables the representation of (most of) those parts of the molecule no longer selected. All subsequent RasMol commands that modify a molecule's colour or representation effect only the currently selected region. The parameter of a **restrict** command is a RasMol atom expression that is evaluated for every atom of the current molecule. This command is very similar to the RasMol [**select**](http://www.umass.edu/microbio/rasmol/distrib/rasman.htm#select#select) command, except restrict disables the [**wireframe,**](http://www.umass.edu/microbio/rasmol/distrib/rasman.htm#wireframe#wireframe) [**spacefill**](http://www.umass.edu/microbio/rasmol/distrib/rasman.htm#spacefill#spacefill) and [**backbone**](http://www.umass.edu/microbio/rasmol/distrib/rasman.htm#backbone#backbone) representations in the non-selected region.

The **restrict** command now turns off the display of ribbons, strands, cartoons and backbones outside of the given atom expression.

Type "help expression" for more information on RasMol atom expressions.

### Ribbons

Syntax: ribbons {<boolean>}

ribbons <value>

The RasMol **ribbons** command displays the currently loaded protein or nucleic acid as a smooth solid "ribbon" surface passing along the backbone of the protein. The ribbon is drawn between each amino acid whose alpha carbon is currently selected. The colour of the ribbon is changed by the RasMol [**colour ribbon**](http://www.umass.edu/microbio/rasmol/distrib/rasman.htm#colour#colour) command. If the current ribbon colour is **none** (the default), the colour is taken from the alpha carbon at each position along its length.

The width of the ribbon at each position is determined by the optional parameter in the usual RasMol units. By default the width of the ribbon is taken from the secondary structure of the protein or a constant value of 720 (2.88 Angstroms) for nucleic acids. The default width of protein alpha helices and beta sheets is 380 (1.52 Angstroms) and 100 (0.4 Angstroms) for turns and random coil. The secondary structure assignment is either from the PDB file or calculated using the DSSP algorithm as used by the [**structure**](http://www.umass.edu/microbio/rasmol/distrib/rasman.htm#structure#structure) command. This command is similar to the RasMol command [**strands**](http://www.umass.edu/microbio/rasmol/distrib/rasman.htm#strands#strands) which renders the biomolecular ribbon as parallel depth-cued curves.

### Rotate

Syntax: rotate <axis> {-} <value>

Rotate the molecule about the specified axis. Permited values for the axis parameter are "<tt><b>x</b></tt>", "<tt><b>y</b></tt>" and "<tt><b>z</b></tt>". The integer parameter states the angle in degrees for the structure to be rotated. For the X and Y axes, positive values move the closest point up and right, and negative values move it down and left respectively. For the Z axis, a positive rotation acts clockwise and a negative angle anti-clockwise.

### Save

Syntax: save {pdb} <filename>

save alchemy <filename>

save mdl <filename>

Save the currently selected set of atoms in either a Brookhaven Protein Database (PDB) or Alchemy(tm) format file. The distinction between this command and the RasMol [**write**](http://www.umass.edu/microbio/rasmol/distrib/rasman.htm#write#write) command has been dropped. The only difference is that without a format specifier the **save** command generates a **PDB** file and the [**write**](http://www.umass.edu/microbio/rasmol/distrib/rasman.htm#write#write) command generates a **GIF** image.

### Script

Syntax: script <filename>

The RasMol **script** command reads a set of RasMol commands sequentially from a text file and executes them. This allows sequences of commonly used commands to be stored and performed by single command. A RasMol script file may contain a further script command up to a maximum "depth" of 10, allowing compilicated sequences of actions to be executed. RasMol ignores all characters after the first '#' character on each line allowing the scripts to be annotated. Script files are often also annotated using the RasMol [**echo**](http://www.umass.edu/microbio/rasmol/distrib/rasman.htm#echo#echo) command.

A RasMol script file can be generated with the [**write script**](http://www.umass.edu/microbio/rasmol/distrib/rasman.htm#write#write) or [**write rasmol**](http://www.umass.edu/microbio/rasmol/distrib/rasman.htm#write#write) commands to output the sequence of commands that are needed to regenerate the current view, representation and colouring of the currently displayed molecule. Such automatically-generated scripts generate only a single image.

RasMol script files can also be created manually with a text editor. Such scripts, through use of the [pause](http://www.umass.edu/microbio/rasmol/distrib/rasman.htm#pause#pause) and [refresh](http://www.umass.edu/microbio/rasmol/distrib/rasman.htm#refresh#refresh) commands, can generate "movies". Detailed [guides to script creation](http://www.umass.edu/microbio/rasmol/scripts.htm) are available.

The RasMol command **source** is synonymous with the **script** command.

### Select

Syntax: select {<expression>}

Define the currently selected region of the molecule. All subsequent RasMol commands that manipulate a molecule or modify its colour or representation, only effects the currently selected region. The parameter of a **select** command is a RasMol expression that is evaluated for every atom of the current molecule. The currently selected (active) region of the molecule are those atoms that cause the expression to evaluate true. To select the whole molecule use the RasMol command **select all.** The behaviour of the **select** command without any parameters is determined by the RasMol [**hetero**](http://www.umass.edu/microbio/rasmol/distrib/rasman.htm#sethetero#sethetero) and [**hydrogen**](http://www.umass.edu/microbio/rasmol/distrib/rasman.htm#set#set) parameters.

Type "help expression" for more information on RasMol atom expressions.

### Show

Syntax: show information

show sequence

show symmetry

The RasMol **show** command display details of the status of the currently loaded molecule. The command **show information** lists the molecule's name, classification, PDB code and the number of atoms, chains, groups it contains. If hydrogen bonding, disulphide bridges or secondary structure have been determined, the number of hbonds, ssbonds, helices, ladders and turns are also displayed respectively. The command **show sequence** lists the residues that compose each chain of the molecule.

### Slab

Syntax: slab {<boolean>}

slab <value>

The RasMol **slab** command enables, disables or positions the z-clipping plane of the molecule. The program only draws those portions of the molecule that are further from the viewer than the slabbing plane. Values range from zero at the very back of the molecule to 100 which is completely in front of the molecule. Intermediate values determine the percentage of the molecule to be drawn.

### Spacefill

Syntax: spacefill {<boolean>}

spacefill temperature

spacefill user

spacefill <value>

The RasMol **spacefill** command is used to represent all of the currently selected atoms as solid spheres. This command is used to produce both union-of-spheres and ball-and-stick models of a molecule. The command, **spacefilll true,** the default, represents each atom as a sphere of Van der Waals radius. The command **spacefill off** turns off the representation of the selected atom as spheres. A sphere radius may be specified as an integer in RasMol units (1/250th Angstrom) or a value containing a decimal point. A value of 500 (2.0 Angstroms) or greater results in a "Parameter value too large" error.

The **temperature** option sets the radius of each sphere to the value stored in its temperature field. Zero or negative values causes have no effect and values greater than 2.0 are truncated to 2. The **user** option allows the radius of each spheres to be specified by additional lines in the molecule's PDB file using Raster 3D's COLOR record extension.

The RasMol command **cpk** is synonymous with the **spacefill** command.

### Structure

Syntax: structure

The RasMol **structure** command calculates secondary structure assignments for the currently loaded protein. If the original PDB file contained structural assignment records (HELIX and SHEET) these are discarded. Initially, the hydrogen bonds of the current molecule are found, if this hasn't been done already. The secondary structure is the determined using Kabsch and Sander's DSSP algorithm. Once finished the program reports the number of helices, strands and turns found.

### Trace

Syntax: trace {<boolean>}

trace <value>

trace temperature

The RasMol **trace** command displays a smooth spline between consecutive alpha carbon positions. This spline does not pass exactly through the alpha carbon position of each residue, but follows the same path as [**ribbons**](http://www.umass.edu/microbio/rasmol/distrib/rasman.htm#ribbons#ribbons), [**strands**](http://www.umass.edu/microbio/rasmol/distrib/rasman.htm#strands#strands), and [**cartoons**](http://www.umass.edu/microbio/rasmol/distrib/rasman.htm#cartoons#cartoons). Note that each residue may be displayed as either a ribbon, strands, cartoon or trace, and enabling one of these representation disables the others. However, a residue may be displayed simultaneously as backbone and one of the above representations [though this may change in future versions of RasMol]. [Prior to version 2.6, **trace** was synonymous with **backbone**.]

**Trace temperature** displays the backbone as a wider cylinder at high temperature factors and thinner at lower. This representation is useful to x-ray crystallographers and NMR spectroscopists.

### Translate

Syntax: translate <axis> {-} <value>

The RasMol **translate** command moves the position of the centre of the molecule on the screen. The axis parameter specifies along which axis the molecule is to be moved and the integer parameter specifies the absolute position of the molecule centre from the middle of the screen. Permited values for the axis parameter are **x**, **y**, and **z**. Displacement values must be between -100 and 100 which correspond to moving the current molecule just off the screen. A positive **x** displacement moves the molecule to the right, and a positive **y** displacement moves the molecule down the screen. The pair of commands **translate x 0** and **translate y 0** centres the molecule on the screen.

### Wireframe

Syntax: wireframe {<boolean>}

wireframe <value>

wireframe dashes

The RasMol **wireframe** command represents each bond within the selected region of the molecule as either a cylinder, a line or depth-cued vector. The display of bonds as depth-cued vectors (drawn darker the further away from the viewer) is turned on by the command **wireframe** or **wireframe on.** The selected bonds are displayed as cylinders by specifying a radius either as an integer in RasMol units or containing a decimal point as a value in Angstroms. A parameter value of 500 (2.0 angstroms) or above results in an "Parameter value too large" error. Bonds may be coloured using the [**colour bonds**](http://www.umass.edu/microbio/rasmol/distrib/rasman.htm#colour#colour) command.

[Wireframe](http://www.umass.edu/microbio/rasmol/distrib/rasman.htm#wireframe#wireframe), [backbone](http://www.umass.edu/microbio/rasmol/distrib/rasman.htm#backbone#backbone) and [strands](http://www.umass.edu/microbio/rasmol/distrib/rasman.htm#strands#strands) representations may be displayed with dashed (dotted) lines. This is enabled by allowing the **dash** or **dashes** parameters to the [wireframe](http://www.umass.edu/microbio/rasmol/distrib/rasman.htm#wireframe#wireframe), [backbone](http://www.umass.edu/microbio/rasmol/distrib/rasman.htm#backbone#backbone) and [strands](http://www.umass.edu/microbio/rasmol/distrib/rasman.htm#strands#strands) commands.

### Write

Syntax: write {<format>} <filename>

Write the current image to a file in a standard raster format. Currently supported image file formats include "**gif**" (Compuserve GIF), "**iris**" (IRIS RBG format), "**ppm**" (Portable Pixmap), "**ras**" (Sun rasterfile), "**ps**" and "**epsf**" (Encapsulated PostScript), "**monops**" (Monochrome Encapsulated PostScript), "**vectps**" (Vector PostScript, *see below*), "**bmp**" (Microsoft bitmap) and "**pict**" (Apple PICT). The **write** command may also be used to generate command scripts for other graphics programs. The format **script** writes out a file containing the RasMol [**script**](http://www.umass.edu/microbio/rasmol/distrib/rasman.htm#script#script) commands to reproduce the current image. The format **molscript** writes out the commands required to render the current view of the molecule as ribbons in Per Kraulis' Molscript program and the format **kinemage** the commands for David Richardson's program Mage.

The RasMol command **write vectps <filename>** creates a postscript file at printer resolution, which can then be sent to your printer. (This command is not on RasMol's Export menu nor is it documented in the on-line help. The command write ps filename writes raster postscript at screen resolution.) The disadvantage of vector postscript is that at present it **does not support** ribbons, cartoons, strands, or traces. Note that the [set vectps on](http://www.umass.edu/microbio/rasmol/distrib/rasman.htm#setvectps#setvectps) command adds outlines to cylinder bonds or spheres. However, it presently does not work for spheres intersecting more than one other sphere. Thus, it works well for stick or ball-and-stick images but not for most spacefilling images.

Techniques for **high-resolution** printing are discussed in the [FAQ](http://www.umass.edu/microbio/rasmol/faq.htm).

The distinction between this command and the RasMol [**save**](http://www.umass.edu/microbio/rasmol/distrib/rasman.htm#save#save) command has been dropped. The only difference is that without a format specifier the [**save**](http://www.umass.edu/microbio/rasmol/distrib/rasman.htm#save#save) command generates a **PDB** file and the **write** command generates a **GIF** image.

The [**set write**](http://www.umass.edu/microbio/rasmol/distrib/rasman.htm#setwrite#setwrite) command enables and disables the use of "save" and "write" in scripts.

The "write gif <filename>" command allows generation of transparent GIFs. This may be controlled by the "set transparent on" and "set transparent off" commands.

### Zap

Syntax: zap

Deletes the contents of the current database and resets parameter variables to their initial default state.

### Zoom

Syntax: zoom {<boolean>}

zoom <value>

Change the magnification of the currently displayed image. Boolean parameters either magnify or reset the scale of current molecule. An integer parameter specifies the desired magnification as a percentage of the default scale. The minimum parameter value is 10, the maximum parameter value is dependent upon the size of the molecule being displayed. For medium sized proteins this is about 500.

### Overlap

Syntax: overlapmol <atom\_selection1> <atom\_selection2>

The extensions to RASMOL commands are explained here in detail. The first set of commands is the usage of overlapmol. This command allows a user to easily superpose two molecules and visualize them on screen. This command can be used in the following way:

Ex: overlapmol ($1TLP\*) ($3BTA\*)

This would then direct HARLEM to overlap the two molecules with names 1TLP and 3 BTA. Note that however, you must use the names of the molecules assigned in HARLEM. Otherwise HARLEM will not know what selections to overlap. Also note that even thought the HARLEM name for the molecule does not begin with a $, it is expected that the user provide a $ prefix to allow HARLEM recognize that they are separate molecules. The two molecules are 1TLP and 3BTA (corresponding to their PDB codes – but they can be something that can be named using the “File” option too). The \* operator indicates to the program that all the atoms in the molecule must be overlapped, and this will allow a full overlap of molecules. Suppose, it is of interest only to overlap a particular region or selection of atoms in the two proteins: such a command would look something like this:

overlapmol ($3BTA$\*.CA and (1-63, 72-132, 142-200)) ($1TLP$\*.CA)

This command essentially overlaps the entire 1TLP CA atoms (c-alpha atoms) with the selected residues (highlighted) CA atoms of 3BTA. Thus, the command line options give high flexibility to select and manipulate bio-molecules on the screen.

### Select (Part 2)

The next command that is presented here is the select command. Selection of atoms is one of the most important aspects while working with proteins and bio-molecules in general. The powerful query mechanism using select has already been shown in the RASMOL tutorial outlined [1]. However there are certain extensions that may be more interesting and those commands are illustrated here. The general format for the select command is:

Syntax: select (within) <atom\_selection>

Ex: select $3BTA$ZN1311.A:ZN

This simple command selects the Zinc atom located on residue named Zn at location 1311. In order to ensure accuracy in all selections, currently it is important to keep track of all PDB related naming conventions (especially residue numbers, since residue numbers and residues are preserved in HARLEM as per the original PDB file; it is also assumed that the user knows enough about the protein molecule with which he/she is working). More complex selections are also possible. Suppose the user wants to select all atoms within a radius of 5.0 A from the active site of 3BTA (the zinc forms one of the active site components). This command can be written as follows:

select within (5, 0, $3BTA$ZN1311.A:ZN)

### Define (Part 2)

All these commands also naturally lead to the definition of custom groups of atoms that the user may want to define. Hence a group can be defined using the define keyword followed by the selection of atoms that form the group. This is illustrated below:

Syntax: define <name\_of\_group> <atom\_selection>

Ex: define myset within (5, 0, $3BTA$ZN1311.A:ZN)

Naturally, a combination of these commands leads to a powerful manipulation language for bio-molecules. Especially, the creation of groups and such abstractions can be good for defining custom calculations on the molecules as a whole. An example of such a command set is given in the next few sections, which will be more general for users to use as a standard template.

Before those programs can be completely shown, however, it is important to understand the HARLEM file format, where all information regarding the molecules is stored. This is the focus of the next section in this manual.

## HARLEM File Format

HARLEM is capable of reading a number of file formats including its own format called the HARLEM file format. This format acts as a broker between various applications, and represents the core data structure that represents the molecule in memory. It is designed with a view towards integrating several applications, and hence the format is more or less a free formatted text file, with every section following a unique header format. A simple HARLEM file may look as shown in figure 2.

As illustrated, every HARLEM file has a record indicator followed by a set of records associated with it. The beginning of every file consists of #BEGIN MOL\_SET record, which for all practical purposes signals the beginning of a new molecule set. In HARLEM every file is considered as a set of molecules, and hence the top level represents this using a MOL\_SET record. Immediately following the MOL\_SET record is the name of the molecule set, and followed by a set of MOLECULE records. Every molecule is formed by a set of ATOMS-, BONDS- and GROUPS- records. This allows a hierarchical definition of molecules within the HARLEM program, and also an easy mechanism to manipulate these molecules.

The ATOMS records consist of the atom number (like a serial number to keep count), the atomic number, the name of the atom for easy identification, the X-Y-Z coordinates, the residue name, atomic charge, whether it is hybridized or not and the atomic mass.

The BONDS records are similarly described using the “from” atom number (indexed from the ATOMS records), the “to” atom number (indexed again from the ATOMS records), and the type of bond that is present.

#BEGIN MOL\_SET

MOL\_SET\_NAME=benzene

#MOLECULE

MOLNAME=BENZENE

#ATOMS

1 6 "C01" 0.000000000 0.000000000 0.000000000 " " "UNK" 1 0.00000 "NO\_HYBRID" "" "" 12.01100

2 6 "C02" -0.702000000 1.216000000 0.000000000 " " "UNK" 1 0.00000 "NO\_HYBRID" "" "" 12.01100

3 6 "C03" -2.107000000 1.216000000 0.000000000 " " "UNK" 1 0.00000 "NO\_HYBRID" "" "" 12.01100

4 6 "C04" -2.809000000 0.000000000 0.000000000 " " "UNK" 1 0.00000 "NO\_HYBRID" "" "" 12.01100

5 6 "C05" -2.107000000 -1.216000000 0.000000000 " " "UNK" 1 0.00000 "NO\_HYBRID" "" "" 12.01100

6 6 "C06" -0.702000000 -1.216000000 0.000000000 " " "UNK" 1 0.00000 "NO\_HYBRID" "" "" 12.01100

7 1 "H07" 1.103000000 0.000000000 0.000000000 " " "UNK" 1 0.00000 "NO\_HYBRID" "" "" 1.00790

8 1 "H08" -0.150000000 2.172000000 0.000000000 " " "UNK" 1 0.00000 "NO\_HYBRID" "" "" 1.00790

9 1 "H09" -2.658000000 2.172000000 0.000000000 " " "UNK" 1 0.00000 "NO\_HYBRID" "" "" 1.00790

10 1 "H10" -3.912000000 0.000000000 0.000000000 " " "UNK" 1 0.00000 "NO\_HYBRID" "" "" 1.00790

11 1 "H11" -2.658000000 -2.172000000 0.000000000 " " "UNK" 1 0.00000 "NO\_HYBRID" "" "" 1.00790

12 1 "H12" -0.150000000 -2.172000000 0.000000000 " " "UNK" 1 0.00000 "NO\_HYBRID" "" "" 1.00790

#END ATOMS

#BONDS

1 2 SINGLE

1 6 SINGLE

1 7 SINGLE

2 3 SINGLE

2 8 SINGLE

3 4 SINGLE

3 9 SINGLE

4 5 SINGLE

4 10 SINGLE

5 6 SINGLE

5 11 SINGLE

6 12 SINGLE

#END BONDS

#END MOLECULE

#GROUPS

#END GROUPS

#ATOM LISTS

#END ATOM LISTS

#END MOL\_SET