RasMol Version 2.6-beta-2 Reference Manual

(Manual de Referencia de RasMol)

http://www.ugr.es/~gebqmed/esrasmol.html

Molecular Visualisation Program

Roger Sayle
Glaxo Wellcome Research and Development
Stevenage, Hertfordshire, U.K.

Jump to Table of Contents
Date of latest revision

If you are not reading this reference manual from http://www.umass.edu/microbio/rasmol/distrib/rasman.htm, you may wish to check there in case revisions or additions have been made since the copy you are reading. Check the date of the latest revision (http://www.umass.edu/microbio/rasmol/distrib/rasman.htm#revhist). Other forms of this reference manual and other sources of information and help for RasMol are available at http://www.umass.edu/microbio/rasmol/getras.htm#rasmanual.

RasMol in versions for several operating systems, installation instructions, and extensive supplementary resources are available at the RasMol Home Page. Included are Frequently Asked Questions (FAQ), tutorials on how to use RasMol, several documents on how to create "movie" scripts, and Roger Sayle's personal account of how RasMol came to be.

Use the **Find** mechanism of your web browser to search for topics of interest. This is possible because this version of the Reference Manual is a **single html file** (rather than being broken into a separate file for each chapter, as are some versions of the Reference Manual you will find on the web). The disadvantage of this is that it may take some time to load the entire Reference Manual (>140 kilobytes) into your browser. If you click on a topic which **appears to be missing**, just wait until the entire document has loaded ("Document: Done" message at the bottom of Netscape).

If you will be referring to the Reference Manual frequently, it is recommended that you use your web browser's **File, Save** to make a copy of the Reference Manual on your local disk. You can then use your browser's local mode (**File, Open**) to view this file, which will load much more quickly. As long as you have installed your web browser on a computer, it does not need an Internet connection to view a local file.

Jump to Table of Contents

Roger Sayle, the creator of RasMol, maintained this reference manual through RasMol version 2.5. In July, 1996, this copy of the RasMol Reference Manual was partially updated from version 2.5 to version 2.6-beta-2 by Dr. Margaret Wong of the Chemistry Department, Swinburne University of Technology, Australia (marg@chem1.chem.swin.edu.au). Further updating, revisions, and maintenance have been by Eric Martz (emartz@microbio.umass.edu), who will be happy to receive suggestions for additional improvements. The complete history of revisions of this Reference Manual is detailed below. Bug fixes and some other details not included here will be found in the release documents 26beta1.txt and 26beta2.txt. A chronologic account of the development of RasMol and its debugging will be found in the original ChangeLog files which accompany the source code. RasMol's original distribution site is by ftp from the University of Edinburgh.

Sayle had a well designed system for converting a master copy of the reference manual into various formats including HTML versions, postscript, and the file which provides the built-in help in RasMol itself. Unfortunately, neither Margaret Wong nor I knew about this system until after we made changes directly to this HTML version. Anyone with serious plans to improve RasMol's documentation should consider obtaining the documentation management software from Roger (and please also inform me!).

Table of Contents

The information supplied in this document is believed to be true but no liability is assumed for its use or for the infringements of the rights of the others resulting from its use. Information in this document is subject to change without notice and does not represent a commitment on the part of the supplier. This product is not to be used in the planning, construction, maintenance, operation or use of any nuclear facility nor the flight, navigation or communication of aircraft or ground support equipment. The author shall not be liable, in whole or in part, for any claims or damages arising from such use, including death, bancruptcy or outbreak of war.

Introduction

RasMol is a molecular graphics program intended for the visualization of proteins, nucleic acids and small molecules. The program is aimed at display, teaching and generation of publication quality images. RasMol runs on wide range of architectures and operating systems including SGI, sun4, sun3, sun386i, DEC, HP and E&S workstations, DEC Alpha (OSF/1, Open VMS and Windows NT), IBM RS/6000, Cray, Sequent, VAX VMS (under DEC windows), IBM PC (under Microsoft Windows, Windows NT, OS/2, Linux, BSD386 and *BSD), Apple Macintosh and PowerMac. UNIX and VMS versions require an 8bit, 24bit or 32bit X Windows frame buffer (X11R4 or later). The X Windows version of RasMol provides optional support for a hardware dials box and accelerated shared memory communication (via the XInput and MIT- SHM extensions) if available on the current X Server.

The program reads in molecular co-ordinate files and interactively displays the molecule on the screen in a variety of representations and colour schemes. Supported input file formats include Brookhaven Protein Databank (PDB), Tripos Associates' Alchemy and Sybyl Mol2 formats, Molecular Design Limited's (MDL) Mol file format, Minnesota Supercomputer Centre's (MSC) XYZ (XMol) format and CHARMm format files. If connectivity information is not contained in the file this is calculated automatically. The loaded molecule can be shown as wireframe bonds, cylinder 'Dreiding' stick bonds, alpha-carbon trace, space-filling (CPK) spheres, macromolecular ribbons (either smooth shaded solid ribbons or parallel strands), hydrogen bonding and dot surface representations. Different parts of the molecule may be represented and coloured independently of the rest of the molecule or displayed in several representations simultaneously. The displayed molecule may be rotated, translated, zoomed and z-clipped (slabbed) interactively using either the mouse, the scroll bars, the command line or an attached dial box. RasMol can read a prepared list of commands from a 'script' file (or via inter-process communication) to allow a given image or viewpoint to be restored quickly. RasMol can also create a script file containing the commands required to regenerate the current image. Finally, the rendered image may be written out in a variety of formats including either raster or vector PostScript, GIF, PPM, BMP, PICT, Sun rasterfile or as a MolScript input script or a Kinemage.

RasMol has been developed at the University of Edinburgh's Biocomputing Research Unit and the BioMolecular Structure Department, Glaxo Research and Development, Greenford, U.K.

If you have any comments, questions, suggestions or complaints please do not hesitate to contact the author:

```
Roger Sayle,                Email:    ras32425@ggr.co.uk
Biomolecular Structure                rasmol@dcs.ed.ac.uk
Glaxo Research and Development,
Greenford Road, Greenford,    Tel:      (+44)(0)81 966 3567
Middlesex UB6 OHE.            Fax:      (+44)(0)81 966 4476
U.K.
```

General Operation

Running RasMol Under UNIX or VMS

To start RasMol from either the UNIX or VMS prompt, type the command 'rasmol'. This command can be followed by an optional filename. By default, immediately upon starting, the program displays the following message to identify the version number and display depth of the running program:

```
RasMol Molecular Renderer
Roger Sayle, October 1994
Version 2.5
[8bit version]
```

Immediately underneath this banner message appears the program's command line prompt 'RasMol>'. If the program is being executed under the X Window System, the program determines the type of the display being used. If the screen has either an 8 bit or 24 bit colour frame buffer, RasMol creates another window, which is used to display menu options and the rendered images. If a suitable screen is not available, RasMol may only be used from the command line. Commands may be typed to manipulate the model, and to output the generated image to a raster file.

If the program is run under the X Window System environment with a suitable colour screen, RasMol creates an additional window to display the rendered molecule interactively, as it is manipulated. If RasMol is not run under the X Window System, the program displays the message 'No suitable display detected!'. RasMol may instructed not to display a graphics window by using the command line option '-nodisplay'. This is particularly useful for using RasMol as a background or batch process.

It is possible to specify either a coordinate filename or a script filename or both on the UNIX/VMS command line. The format for specifying a script file to add the option '-script <filename>' to the command line. A molecule co-ordinate file may be specified by placing its name on the command line, optionally preceded by a file format option. If no format option is given, the specified co-ordinate file is assumed to be in PDB format.

Valid format options include '-pdb', '-mdl', '-mol2', '-xyz', '- alchemy' and '-charmm', which correspond to Brookhaven, MDL Mol file, Sybyl Mol2, MSC's xyz, Alchemy and CHARMm formats respectively. If both a co-ordinate file and a script file are specified on the command line, the molecule is loaded first, then the script commands are applied to it. If either file is not found, the program displays the error message 'Error: File not found!' and the user is presented the RasMol prompt.

In order to leave RasMol, the user can type the command quit at the RasMol prompt, and the program will return the user to the familiar unix prompt. Alternatively, if a prompt other than the main RasMol prompt is being displayed, the user may hit control-C (^C) to leave the program. The message '***Quit***' will be output to the terminal, before the usual unix prompt is redisplayed. The program may also be terminated by selecting the Quit menu option, on the bottom of the main menu.

Running RasMol Under Microsoft Windows

To start RasMol under Microsoft Windows, double click on the RasMol icon in the program manager. When RasMol first starts, the program displays a single main window (the display window) with a black background on the screen and provides the command line window minimized as a small icon at the bottom of the screen. The command line or terminal window may be opened by double clicking on this RasMol icon.

It is possible to specify either a coordinate filename or both on the windows command line. The format for specifying a script file to add the option '-script <filename>' to the command line. A molecule co-ordinate file may be specified by placing its name on the command line, optionally preceded by a file format option. If no format option is given, the specified co-ordinate file is assumed to be in PDB format. Valid format options include '- pdb', '-mdl', '-mol2', '-xyz', '-alchemy' and '-charmm', which correspond to Brookhaven, MDL Mol file, Sybyl Mol2, MSC's xyz, Alchemy and CHARMm formats respectively. If both a co-ordinate file and a script file are specified on the command line, the molecule is loaded first, then the script commands are applied to it. If either file is not found, the program displays the error message 'Error: File not found!' and the user is presented the RasMol prompt.

Running RasMol on the Apple Macintosh/PPC

To start RasMol on the Macintosh, double click on the RasMol icon using Finder. When RasMol first starts the program displays two windows, the top window (with the black background) is the graphics or canvas window and the window underneath it (with the white background) is the RasMol command line window.

RasMol on the Macintosh may also be started by double clicking on a file owned/created by the application with the signature 'RSML'. This will start up RasMol and pass the selected file to be loaded. There is no way of specifying the file format on the command line with a Macintosh so RasMol attempts to determine the file format by inspecting the file's type signature. Files with type signature 'RSML' are assumed to be RasMol scripts, files of type 'mMOL' are assumed to be MDL Mol files and all other types (principly 'TEXT') are assumed to be in PDB format. Unlike other versions of RasMol it is impossible to specify both a script and a co-ordinate file simultaneously.

Dragging and dropping multi-file 'movie' scripts onto aliases or copies of the RasMol application file may fail due to confusion about which is the correct script folder. Double-clicking on a script may lead to similar problems if copies are present. Further information will be found in the 'movie' script operation document , where you will find reference to an explanation of Macintosh signatures and how to change them.

Note that because on a Macintosh only one 'instance' of an application may be running at any one time, if you were double click on another file owned by 'RSML', the running copy of RasMol would 'zap' its molecule and load the newly specified file.

RasMol's Window

On all platforms RasMol displays two windows, the main **graphics** or **canvas** window with a black background and a **command line** or **terminal** window. At the top of the graphics window (or at the top of the screen for the Macintosh) is the RasMol menu bar. The contents of the menu bar change from platform to platform to support the local user interface guidelines, however all platforms support the 'File', 'Display', 'Colours', 'Export' and 'Options' pull-down menus. The Main graphics window also has two scroll bars, one on the right and one at the bottom, that may be used to rotate the molecule interactively.

The Macintosh implementation lacks the built-in hypertext help available from the 'Help' menu under MS Windows. The 'help' command is available in the command line window provided that the file 'rasmol.hlp' is in the same folder as the application itself, but this may be relatively cumbersome to use. Therefore it is recommended that the present hypertext reference manual be brought up in a web browser window when help is needed on the Macintosh. Since in RasMol 2.6 beta-2 the built-in hypertext reference manual is still at version 2.5, this is recommended for other operating systems as well. A local copy of this reference manual can be saved from the File menu in the web browser, and can be viewed in the browser's local mode even on computers lacking an Internet connection.

While the mouse pointer is located within the graphics area of the main display window, the mouse pointer is drawn as a cross- hair cursor, to enable the picking of objects being displayed; otherwise the mouse pointer is drawn as an arrowhead. Any characters that are typed at the keyboard while the display window is in 'focus' (meaning active or foreground) are redirected to the command line in the terminal window. Hence you do not need continually to switch focus between the command line and graphics windows.

The display window may be resized at any point during the session. This has the effect of simply rescaling the image displayed on the canvas. RasMol imposes limits on the size of the display window such that the window must be large enough to display the menu and scroll bars and yet small enough to fit on a single screen. Attempts to enlarge the screen may fail owing to insufficient memory on the host machine, in which case RasMol reports the error message 'Renderer Error: Unable to allocate frame buffer!' or some similar error.

On eight bit displays, when the number of colours required by the program exceeds the number of free colours on the screen, the program uses its own colourmap. This has the effect of temporarily displaying all windows other than the display window in false colours while the mouse pointer is within the display windows. If the mouse pointer is moved outside the display windows, the original colours of the other windows return, and the image on the canvas is shown in 'false colour'. Once the number of colours required by the program drops again, the presentation of colours returns to normal.

Mouse Controls

Here is a summary of RasMol's mouse click-and-drag controls. The set mouse command mode defaults to **set mouse rasmol**, which gives the controls summarized below. However, there are also **set mouse insight** and **set mouse quanta** modes (not shown below).

Action	Windows	Macintosh
Rotate X,Y	Left	Unmodified
Translate X,Y	Right	Command*
Rotate Z	Shift-Right	Shift-Command*
Zoom	Shift-Left	Shift
Slab Plane	Ctrl-Left	Ctrl

*On some Macs, the Option (Alt) key has the same effect on RasMol as the Command key.

Scroll Bars

The scroll bar across the bottom of the canvas area is used to rotate the molecule about the y-axis, i.e. to spin the nearest point on the molecule left or right; and the scroll bar to the right of the canvas rotates the molecule about the x-axis, i.e. the nearest point up or down. Each scroll bar has an 'indicator' to denote the relative orientation of the molecule, which is initially positioned in the centre of the scroll bar. These scroll bars may be operated in either of two ways. The first is by clicking any mouse button on the dotted scroll bar background to indicate a direct rotation relative to the current indicator position; the second is by clicking one of the arrows at either end of the scroll bar to rotate the molecule in fixed sized increments. Rotating the molecule by the second method may cause the indicators on the scroll bars to wrap around from one end of the bar to the other. A complete revolution is indicated by the indicator travelling the length of the scroll bar. The angle rotated by using the arrows depends upon the current size of the display window.

Picking

In order to identify a particular atom or bond being displayed, RasMol allows the users to 'pick' objects on the screen. The mouse is used to position the cross-hair cursor over the appropriate item, and then any of the mouse button is depressed. Provided that the pointer is located close enough to a visible object, the program determines the identity of the nearest atom to the point identified.

The program will display, in the terminal window, the atom's type, serial number, residue name and residue number. If the atom is a member of a named chain, the chain identifier is also displayed. Two examples of the output generated by selecting an atom are displayed below:

```
Atom:  CA 349      Group:  SER 70
Atom:  O  526      Hetero:  HOH 205    Chain:  P
```

The first line describes the alpha carbon of the serine-70 amino acid in a protein. The unique Brookhaven serial number for this atom is 349. The following line describes the oxygen atom in a water molecule attached to the P chain of the main molecule. The word 'Hetero' distinguishes heterogenous molecules (such as cofactors) from the residues in the main molecule, noted by 'Group'. [These two atoms are referred to by the two atom expressions 'SER70.CA' and 'HOH205:P.O', respectively, when using the RasMol commands select and restrict.]

Clicking the mouse on an atom can be used not only to **identify** it, but also to find **distances** between two atoms (or to display a **distance monitor**), **the bond angle** defined by three atoms, the **torsion angle** defined by four atoms, to toggle **labels** on or off, or to specify the **center of rotation** . See the set picking command for details.

Dials Box

If RasMol detects a 'dials box' attached to the user's workstation, it also allows the molecule to be manipulated interactively by the dials. Once RasMol starts up, it labels the LED displays above each dial, 'ROTATE X', 'ROTATE Y', 'ROTATE Z' and 'ZOOM' across the top row from left to right, and 'TRANS X', 'TRANS Y', 'TRANS Z' and 'SLAB' from left to right across the bottom row. Rotating any of the knobs will automatically transform and redisplay the molecule interactively. The dials only have effect while the mouse pointer is within the display window. If more than one application is using the dials box at a time, care must be taken to remember the dial labels assigned by each program, as each application may overwrite the dial-label LEDS.

The rotation about the X and Y axes automatically updates the indicators on the appropriate scroll bars. All the rotation dials rotate the molecule 180 degrees for a complete revolution of the dial. All the remaining dials clamp their values to permissible ranges; turning these dials past their limits has no effect. The centre of rotation of the molecule may be changed using the centre command on the command line, or the command set picking centre followed by a mouse click.

The 'ZOOM' dial allows the interactive zooming of the molecule between 10% and 200% of the original default magnification. Rotating the dial clockwise magnifies the molecule and anticlockwise shrinks it. A complete revolution of the dial corresponds to a 100% change in scale.

The 'SLAB' dial, which is only effective when slabbing is enabled, allows the user to move the front z-clipping plane from the nearest point on the molecule to the furthest. A complete rotation of the SLAB dial corresponds to moving the clipping plane half the distance between the front and back of the molecule. Turning the SLAB knob clockwise moves the clipping plane closer to the viewer (increasing the number of objects displayed), and turning it anticlockwise moves it further away (preventing more objects from being displayed).

Slabbing mode is enabled by typing the command 'slab on' on the command line or toggling the slab option on the options menu.

Translation along the X and Y axis allows the centre of the molecule to be moved within the canvas area of the screen. Rotation and zooming are still performed relative to the centre of rotation and the molecule respectively, which may often not be at the centre of the canvas. The TRANS Z dial currently has no effect.

Command Line Interface

RasMol maintains a history of recently used commands, so that the user never needs to type the same commands repeatedly. Typing ^P (Control-P) on the command line will display the previous command in the history and ^N will display the following command. These commands may be edited using the features described below. Moving forward or backward through the command history undoes the modifications made to the current line. The number of commands retained in the history depends upon their length. RasMol can retain more short command lines and fewer long ones.

Users with the Microsoft Windows version or the X windows version and with 'vt100' or compatible terminals (such as an 'xterm') can use the cursor control characters on the keyboard to abbreviate the control keys. The right and left cursor keys have the same affect as ^F and ^B, moving the cursor forward and back a single character respectively. Similarly, the up and down cursor keys have the same function as ^P and ^N, producing the previous and next entries in the command history respectively.

RasMol allows basic editing of the command line. Pressing either backspace, delete or ^H (Control-H) will delete the previous character, and the key ^D may be used to delete the character under the cursor. Several characters may be used to move the cursor along the command line. The characters ^B, ^F, ^A and ^E move the cursor back a single character, forward a single character, to the beginning of the line and to the end of the line, respectively. When the cursor is not at the end of the command line, typed characters are inserted into the line and do not overwrite existing characters. After a command line has been edited, a newline or carriage return will enter the entire line, regardless of where the cursor is positioned. Because RasMol is unable to move the cursor up to the previous line, care must be taken when editing commands that wrap over several lines. In the event that another process overwrites or corrupts the command line, the character ^L may be used to redisplay the line on the screen.

Dimensions within RasMol

All dimensions in RasMol, such as radii and distances, may be specified in either RasMol 'units' or Angstroms. The RasMol units were first introduced to allow reasonably sized values to be specified for most of the operations performed in RasMol. A single RasMol unit corresponds to 1/250th of an Angstrom, therefore the most frequently used values are in the hundreds. For this reason, if RasMol is given a distance parameter that doesn't contain a decimal point it is assumed to be in RasMol units. For example, the command 'spacefill 300' specifies a sphere radius of 300 RasMol units, or 1.2 Angstroms.

However, dimensions within RasMol can also be specified in Angstroms by placing a decimal point in the number. For example, 'spacefill 1.2' specifies a sphere radius in Angstroms. This is particularly useful for the cut-off distance parameter in 'within' expressions.

Start-up Initialisation Files

Each time RasMol is started, it searches for an initialisation file of commands to run before the command prompt is presented to the user. The file is called **.rasmolrc** on UNIX systems, and **RASMOL.INI** on VMS and Microsoft Windows Systems. the format and execution of this file is identical to that of the RasMol script command.

RasMol first looks for the initialisation file in the current directory, and if it is not found will look for it in the user's home directory. On all systems the environment variable **HOME** may be used to name the user's home appropriate directory. If no personal initialisation file is found the program looks for the file **rasmolrc** (or **RASMOLRC**) in the RasMol system directory pointed to by the environment variable **RASMOLPATH**. This directory should also contain the on-line help file **rasmol.hlp**. On UNIX systems RASMOLPATH is typically set to be '/usr/local/lib/rasmol'.

Unlike the command 'script ".rasmolrc"', the program will not generate an error message if the file is not found. The system rasmolrc file is commonly used by system managers to display information about the local installation and who to contact for help contain RasMol echo commands detailing a telephone number or e-mail address to be used for contacting.

Inter-process Communication

RasMol supports Inter Process Communication (IPC) in one form or another on all platforms. Under Microsoft Windows, IPC is implemented using Dynamic Data Exchange (DDE), on the MacIntosh IPC is implemented using Apple Events and on X Windows systems IPC is implemented using John Ousterhaut's Tcl/Tk communication protocol.

When RasMol starts up on an **X window** system it registers itself with the X window Server as a **Tcl interpreter**. From within a Tcl application such as 'wish', you can use the Tcl command 'winfo interps' to determine the currently register interpreters on that display. The first instance of RasMol registers itself as 'rasmol', the second as 'rasmol2', the third as 'rasmol #3' and so on. The Tcl interpreter can easily send a command to rasmol using the built-in 'send' command. RasMol interprets the string parameter to the send command not as a Tcl function to execute but as a RasMol command. Hence, typing 'send {rasmol} {background red}' into the wish interpreter will cause RasMol's display window to change colour. Using the same encoding as Microsoft's DDE Execute protocol, multiple commands may be sent in a single 'send' by placing the consecutive commands in square brackets. RasMol will execute all of the commands in a 'send' before refreshing the screen.

Under **Microsoft Windows**, RasMol supports a complete DDE protocol. The simplest layers of the protocol may be accessed by sending a DDE Execute command to application 'RasWin' and any topic. This will start a DDE conversation with the most recently launched instance of RasMol. Although any topic name can be used, the use of 'System' and/or 'RemoteControl' are recommended. Once again the contents of the execute package consists of a string for RasMol to execute. If the first non-whitespace character is an open bracket, the string is interpreted to be a sequence of consecutive commands enclosed in square brackets; otherwise the string consists of just a single command. Commands in square brackets may optionally be separated by whitespace and/or semi- colons. RasMol can also act as a 'data server' supporting hot, cold and warm links. Currently supported DDE items include 'Name', 'Image', 'Pick', 'Count' which denotes the Molecule name, the currently displayed image (in Microsoft DIB format), the atom expression of the last picked atom (or an empty string) and the number of selected atoms respectively. Using a hot or warm link on the 'Pick' item for example, allows an application such as Microsoft Word, Excel or Visual Basic to respond each time the user clicks on an atom in RasMol.

RasMol on the **Apple Macintosh** supports **AppleEvents**. Currently the only supported AppleEvents are the four 'core' events, Open Application, Open Document, Print Document and Quit. However, because OpenDocument determines its actions by the file's type signature this can be used to implement generic IPC. Because RasMol for the Macintosh treats all files of type 'RSML' as scripts, the sending applicatin need only place all the commands to be executed in a temporary file, set the type of the file to 'RSML' and then send RasMol an OpenDocument AppleEvent with the file as parameter.

Command Reference

RasMol allows the execution of interactive commands typed at the "**`RasMol>`**" prompt in the command-line window (a window separate from the graphics window). Each command must be given on a separate line. Keywords are case insensitive and may be entered in either upper or lower case letters. All whitespace characters are ignored except to separate keywords and their arguments.

The commands/keywords currently recognised by RasMol are given below.

Backbone
Background
Cartoon
Centre
Clipboard
Colour
Connect
CPK
Dots
Define
Echo
Exit
HBonds
Help
Label

Load
Monitor
Pause
Print
Quit
Renumber
Refresh
Reset
Restrict
Ribbons
Rotate
Save
Script
Select
Set*

Show
Slab
Source
Spacefill
SSBonds
Stereo
Strands
Structure
Trace
Translate
Wireframe
Write
Zap
Zoom

*Note that the Set command includes a large number of important options, including, for example, set picking which allows the measurement of **distances, angles and torsion angles** among other things.

Backbone

```
Syntax:  backbone {<boolean>}
         backbone <value>
         backbone dashes
```

The RasMol **`backbone`** command permits the representation of a polypeptide backbone as a series of bonds connecting the adjacent alpha carbons of each amino acid in a chain. The display of these backbone `bonds' is turned on and off by the command paramater the same as the **`wireframe`** command. The command **`backbone off`** turns off the selected `bonds', and **`backbone on`** or with a number turns them on. The number can be used to specify the cylinder radius of the representation in either angstrom or rasmol units. A parameter value of 500 (2.0 angstroms) or above results in a "Parameter value too large" error. Backbone objects may be coloured using the RasMol **`colour backbone`** command.

The reserved work **`backbone`** is also 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. Wireframe, backbone and strands representations may be displayed with dashed (dotted) lines. This is enabled by allowing the **`dash`** or **`dashes`** parameters to the wireframe, 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** 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** 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** 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** 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** 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 will give a list of all the predefined colour names recognised by RasMol.

Allowed objects are **atoms**, **bonds**, **backbone**, **ribbons**, **labels**, **dots**, **hbonds**, and **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**, **amino**, **chain**, **group**, **shapely**, **structure**, **temperature**, **charge**, and **user**. Hydrogen bonds can also be coloured by **type** and dot surfaces can also be coloured by **electrostatic potential**.

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** 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** command.

Echo

```
Syntax:  echo {<string>}
```

The RasMol **echo** command is used to display a message in the RasMol command/terminal window. The string parameter may optionally be delimited in double quote characters. If no parameter is specified, the **echo** command displays a blank line. This command is particularly useful for displaying text from within a RasMol **script** file.

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** command, in contrast, terminates the execution of RasMol itself.

HBonds

```
Syntax:  hbonds {<boolean>}
         hbonds <value>
```

The RasMol **hbond** 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 **hbond** 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** 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** command the alpha carbon positions of the appropriate residues may be used instead. This is especially useful when examining proteins in backbone representation.

Help

```
Syntax:  help {<topic> {<subtopic>}}
         ? {<topic> {<subtopic>}}
```

The RasMol **help** command provides on-line help on the given topic.

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** 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** 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** 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** 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.

Monitor

```
Syntax:  monitor <number>  <number>
         monitor {<boolean>}
```

The RasMol **monitor** command allows the display of distance monitors. A distance monitor is a dashed (dotted) line between an arbitrary pair of atoms, optionally labelled by the distance between them.
monitor <number> <number> adds such a distance monitor between the two atoms specified by the atom serial numbers given as parameters.

Distance monitors are turned off with the command **monitors off**. By default, monitors display the distance between its two end points as a label at the centre of the monitor. These distance labels may be turned off with the command set monitors off, and re-enabled with the command set monitors on. Like most other representations the colour of a monitor is taken from the colour of its end points unless specified by the colour monitors command.

Distance monitors may also be added to a molecule interactively with the mouse, using the set picking monitor command. Clicking on an atom results in it being identified on the rasmol command line. In addition every atom picked increments a modulo counter such that in monitor mode, every second atom displays the distance between this atom and the previous one. The shift key may be used to form distance monitors between a fixed atom and several consecutive positions. A distance monitor may also be removed (toggled) by selecting the appropriate pair of atom end points a second time.

Pause

```
Syntax:  pause
         wait
```

The RasMol **pause** command is used in script files to stop the script file for local manipulation by a mouse, until any key is ushed to restart the script file. **Wait** is synonymous with **pause**.

This command may be executed in RasMol script files to suspend the sequential execution of commands and allow the user to examine the current image. When RasMol executes a "pause" command in a script file, it suspends execution of the rest of the file, refreshes the image on the screen and allows the manipulation of the image using the mouse and scroll bars, or resizing of the graphics window. Once a key is pressed, control returns to the script file at the line following the "pause" command. While a script is suspended the molecule may be rotated, translated, scaled, slabbed and picked as normal, but all menu commands are disabled. The "pause" can probably be used most effectively with "echo" commands in education pre-scripted demonstrations, where a description of the current image is presented to the user/student. Typically the command before a "pause" should be "echo Press any key to continue".

Execution of a script can be cancelled by pressing Control-D or Control-Z (on VAX/VMS, Control-C) while standing at a pause. The command set picking none disables picking, which avoids the display of spurious messages whilst a script is suspended at a pause.

Print

```
Syntax:  print
```

The RasMol **print** command sends the currently displayed image to the local default printer using the operating system's native printer driver. Note: this command is not yet supported under UNIX or VMS. It is intended to take advantage of Microsoft Windows and Apple Macintosh printer drivers. For example, allowing images to be printed directly on a dot matrix printer.

When using RasMol on a UNIX or VMS system this functionality may be achieved by either generating a PostScript file using the RasMol **write ps** or **write vectps** commands and printing that or generating a raster image file and using a utility to dump that to the local printer.

Techniques for **high-resolution** printing are discussed in the FAQ.

Quit

```
Syntax:  quit
```

Exit from the RasMol program. The RasMol command **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,`** the centre of rotation is set to the geometric centre of the currently loaded molecule, **`centre all,`** 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`** 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`** command, except restrict disables the **`wireframe,`** **`spacefill`** and **`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`** 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`** command. This command is similar to the RasMol command **`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` command has been dropped. The only difference is that without a format specifier the **`save`** command generates a **`PDB`** file and the `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` command.

A RasMol script file can be generated with the `write script` or `write rasmol` 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 and refresh commands, can generate "movies". Detailed guides to script creation 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` and `hydrogen` parameters.

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

There is a separate introduction to Select commands in Chime and RasMol.

Set

```
Syntax:  set <parameter> {<option>}
```

The RasMol **set** command allows the user to alter various internal program parameters such as those controlling rendering options. Each parameter has its own set or permissible parameter options. Typically, omiting the parameter option resets that parameter to its default value. A list of valid parameter names is given below.

Ambient
Axes
Background
BackFade
BondMode
Bonds
BoundBox
Cartoon
Display
FontSize
HBonds
Hetero
HourGlass
Hydrogen
Kinemage
Menus
Monitor
Mouse
Picking
Radius
Shadow
SlabMode
Solvent
Specular
SpecPower
SSBonds
StereoViewing
Strands
TransparentGif
UnitCell
VectPS
Write

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.

Rotation rotates the molecule through a fixed slab plane. Therefore, **stereo** does not produce a valid stereo image when slab is on.

See also **set slabmode**.

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.

SSBonds

```
Syntax:  ssbonds {<boolean>}
         ssbonds <value>
```

The RasMol **ssbonds** command is used to represent the disulphide bridges of the protein molecule as either dotted lines or cylinders between the connected cysteines. The first time that the **ssbonds** command is used, the program searches the structure of the protein to find half-cysteine pairs (cysteines whose sulphurs are within 3 angstroms of each other) and reports the number of bridges to the user. The command **ssbonds on** displays the selected `bonds' as dotted lines, and the command **ssbonds off** disables the display of ssbonds in the currently selected area. Selection of disulphide bridges is identical to normal bonds, and may be adjusted using the RasMol **set bondmode** command. The colour of disulphide bonds may be changed using the **colour ssbonds** command. By default, each disulphide bond has the colours of its connected atoms.

By default disulphide bonds are drawn between the sulphur atoms within the cysteine groups. By using the **set ssbonds** command the position of the cysteine's alpha carbons may be used instead.

Stereo

```
Syntax:  stereo on
         stereo [-] <number>
         stereo off
```

The RasMol **stereo** command provides side-by-side stereo display of images. Stereo viewing of a molecule may be turned on (and off) either by selecting **Stereo** from the **Options menu**, or by typing the commands **stereo on** or **stereo off**. The separation angle between the two views may be adjusted with the set stereo [-] <number> command, where positive values result in crossed eye viewing and negative values in relaxed (wall-eyed) viewing. The inclusion of **[-] <number>** in the **stereo** command, as for example in **stereo 3** or **stereo -5**, also controls angle and direction.

As of RasMol version 2.6-beta-2, the stereo command is only partially implemented. When stereo is turned on, the image is not properly recentered. (This can be done with a **translate x -<number>** command.) It is not supported in vector PostScript output files, is not saved by the **write script** command, and in general is not yet properly interfaced with several other features of the program. When **slab** is on together with stereo, the image which results is not a stereoscopic view.

Strands

```
Syntax:  strands {<boolean>}
         strands <value>
         strands dashes
```

The RasMol **strands** command displays the currently loaded protein or nucleic acid as a smooth "ribbon" of depth-cued curves passing along the backbone of the protein. The ribbon is composed of a number of strands that run parallel to one another along the peptide plane of each residue. 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** 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 colour of the central and outermost strands may be coloured independently using the **colour ribbon1** and **colour ribbon2** commands respectively. The number of strands in the ribbon may be altered using the RasMol **set strands** command.

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 for nucleic acids (which produces a ribbon 2.88 Angstroms wide). 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** command. This command is similar to the RasMol command **ribbons** which renders the biomolecular ribbon as a smooth shaded surface.

Wireframe, backbone and strands representations may be displayed with dashed (dotted) lines. This is enabled by allowing the **dash** or **dashes** parameters to the wireframe, backbone and strands commands.

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**, **strands**, and **cartoons**. Note that each residue may be displayed as either a ribbon, strands, cartoon or trace, and enabling one of these representations 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** command.

Wireframe, backbone and strands representations may be displayed with dashed (dotted) lines. This is enabled by allowing the **dash** or **dashes** parameters to the wireframe, backbone and 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** 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 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.

The distinction between this command and the RasMol **save** command has been dropped. The only difference is that without a format specifier the **save** command generates a **PDB** file and the **write** command generates a **GIF** image.

The **set write** 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.

Internal Parameters

RasMol has a number of internal parameters that may be modified using the **set** command. These parameters control a number of program options such as rendering options and mouse button mappings.

A complete list of internal parameter names is given under the **set** command.

Set Ambient

```
Syntax:  set ambient {<value>}
```

The RasMol **ambient** parameter is used to control the amount of ambient (or surrounding) light in the scene. The **ambient** value must be between 0 and 100 that controls the percentage intensity of the darkest shade of an object. For a solid object, this is the intensity of surfaces facing away from the light source or in shadow. For depth-cued objects this is the intensity of objects furthest from the viewer.

This parameter is commonly used to correct for monitors with different "gamma values" (brightness), to change how light or dark a hardcopy image appears when printed or to alter the feeling of depth for wireframe or ribbon representations.

Set Axes

```
Syntax:  set axes <boolean>
         set axes on
         set axes off
```

The RasMol **axes** parameter controls the display of orthogonal co-ordinate axes on the current display. The co-ordinate axes are those used in the molecule data file, and the origin is the centre of the molecule's bounding box. The **set axes** command is similar the the commands **set boundbox** and **set unitcell** that display the bounding box and the crystallographic unit cell respectively.

With **set axes on** the characters X, Y and Z are displayed on the postive direction of the cartesian axes.

Set Backfade

```
Syntax:  set backfade on
         set backfade off
```

Shading to an arbitrary background colour, rather than just black, can be performed. This is controlled by the commands "set backfade on" and "set backfade off". For example, this may be used to generate depth-cued wireframe images that fade to white, rather than black.

Set Background

```
Syntax:  set background {<colour>}
```

The RasMol **background** parameter 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, Blue (RGB) components enclosed in square brackets. Typing the command **help colours** 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 command **set background** is synonymous with the RasMol command **background.**

The RasMol **set bondmode** command controls the mechanism used to select individual bonds. When using the

Set BondMode

```
Syntax:  set bondmode and
         set bondmode or
```

The RasMol **set bondmode** command controls the mechanism used to select individual bonds. When using the **select** and **restrict** commands, a given bond will be selected if i) the bondmode is **or** and either of the connected atoms is selected, or ii) the bondmode is **and** and both atoms connected by the bond are selected. Hence an individual bond may be uniquely identified by using the command "**set bondmode and**" and then uniquely selecting the atoms at both ends.

Set Bonds

```
Syntax:  set bonds <boolean>
```

The RasMol **set bonds** command controls display of double and triple bonds as multiple lines or cylinders. Currently bond orders are only read from MDL Mol files, Sybyl Mol2 format files, Tripos Alchemy format files and suitable Brookhaven PDB files. Double (and triple) bonds are specified in PDB files by specifying a given bond twice (and three times) in CONECT records. The command "set bonds on" enables the display of bond order, and the commands "set bonds off" and "set bonds off" disable them. See the FAQ for detailed examples.

Set BoundBox

```
Syntax:  set boundbox <boolean>
```

The RasMol **boundbox** parameter controls the display of the current molecules bounding box on the display. The bounding box is orthogonal to the data file's original co-ordinate axes. The **set boundbox** command is similar the the commands **set axes** and **set unitcell** that display orthogonal co-ordinate axes and the bounding box respectively.

Set Cartoon

```
Syntax:  set cartoon <boolean>
         set cartoon <number>
```

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.

Set Display

```
Syntax:  set display selected
         set display normal
```

This command controls the display mode within RasMol. By default, **set display normal,** RasMol displays the molecule in the representation specified by the user. The command **set display selected** changes the display mode such that the molecule is temporarily drawn so as to indicate currently selected portion of the molecule. The user specified colour scheme and representation remains unchanged. In this representation all selected atoms are shown in yellow and all non selected atoms are shown in blue. The colour of the background is also changed to a dark grey to indicate the change of display mode. This command is typically only used by external Graphical User Interfaces (GUIs).

Set HBonds

```
Syntax:  set hbonds backbone
         set hbonds sidechain
```

The RasMol **hbonds** parameter determines whether hydrogen bonds are drawn between the donor and acceptor atoms of the hydrogen bond, **set hbonds sidechain** or between the alpha carbon atoms of the protein backbone and between the phosphorous atoms of the nucleic acid backbone, **set hbonds backbone.** The actual display of hydrogen bonds is controlled by the **hbonds** command. Drawing hydrogen bonds between protein alpha carbons or nucleic acid phosphorous atoms is useful when the rest of the molecule is shown in only a schematic representation such as **backbone,** **ribbons** or **strands.** his parameter is similar to the RasMol **ssbonds** parameter.

Set FontSize

```
Syntax:  set fontsize {<value>}
```

The RasMol **set fontsize** command is used to control the size of the characters that form atom labels. This value corresponds to the height of the displayed character in pixels. The maximum value of **fontsize** is 32 pixels, and the default value is 8 pixels high. To display atom labels on the screen use the RasMol **label** command and to change the colour of displayed labels, use the **colour labels** command.

Set Hetero

```
Syntax:  set hetero <boolean>
```

The RasMol **hetero** parameter is used to modify the `default' behaviour of the RasMol **select** command, i.e. the behaviour of **select** without any parameters. When this value is **false,** the default **select** region does not include any heterogenous atoms (refer to the predefined set **hetero**). When this value is **true,** the default **select** region may contain hetero atoms. This parameter is similar to the RasMol **hydrogen** parameter which determines whether hydrogen atoms should be included in the default set. If both **hetero** and **hydrogen** are **true,** **select** without any parameters is equivalent to **select all.** See also the group colour scheme, which is affected by the value of the **hetero** parameter. The *Hetero Atoms* item on RasMol's *Options* menu can also be used to toggle the value of the **hetero** parameter.

Set HourGlass

```
Syntax:  set hourglass <boolean>
```

The RasMol **hourglass** parameter allows the user to enable and disable the use of the `hour glass' cursor used by RasMol to indicate that the program is currently busy drawing the next frame. The command **set hourglass on** enable the indicator, whilst **set hourglass off** prevents RasMol from changing the cursor. This is useful when spinning the molecule, running a sequence of commands from a script file or using interprocess communication to execute complex sequences of commands. In these cases a `flashing' cursor may be distracting.

Set Hydrogen

```
Syntax:  set hydrogen <boolean>
```

The RasMol **hydrogen** parameter is used to modify the `default' behaviour of the RasMol **select** command, i.e. the behaviour of **select** without any parameters. When this value is **false,** the default **select** region does not include any hydrogen or deuterium atoms (refer to the predefined set **hydrogen**). When this value is **true,** the default **select** region may contain hydrogen atoms. This parameter is similar to the RasMol **hetero** parameter which determines whether heterogenous atoms should be included in the default set. If both **hydrogen** and **hetero** are **true,** **select** without any parameters is equivalent to **select all.**

Set Kinemage

```
Syntax:  set kinemage <boolean>
```

The RasMol **set kinemage** command controls the amount of detail stored in a Kinemage output file generated by the RasMol **write kinemage** command. The output kinemage files are intended to be displayed by David Richardson's Mage program. **set kinemage false,** the default, only stores the currently displayed representation in the generated output file. The command **set kinemage true,** generates a more complex Kinemage that contains both the wireframe and backbone representations as well as the co-ordinate axes, bounding box and crystal unit cell.

Set Menus

```
Syntax:  set menus <boolean>
```

The RasMol **set menus** command enables the canvas window's menu buttons or menu bar. This command is typically only used by graphical user interfaces or to create as large as image as possible when using Microsoft Windows.

Set Monitor

```
Syntax:  set monitor on
         set monitor off
```

The **distance monitor labels** may be turned off with the command **set monitors off**, and re-enabled with the command **set monitors on**.

Set Mouse

```
Syntax:  set mouse rasmol
         set mouse insight
         set mouse quanta
```

The RasMol **set mouse** command sets the rotation, translation, scaling and zooming mouse bindings. The default value is **rasmol** which is suitable for two button mice (for three button mice the second and third buttons are synonymous); X-Y rotation is controlled by the first button, and X-Y translation by the second. Additional functions are controlled by holding a modifier key on the keyboard. [Shift] and the first button performs scaling, [shift] and the second button performs Z-rotation, and [control] and the first mouse button controls the clipping plane. The **insight** and **quanta** provide the same mouse bindings as other packages for experienced users.

Set Picking

```
Syntax:  set picking on
         set picking off
         set picking none
         set picking ident
         set picking distance
         set picking monitor
         set picking angle
         set picking torsion
         set picking label
         set picking centre
```

The **set picking** series of commands affects how a user may interact with a molecule displayed on the screen in Rasmol.

Enabling/Disabling Atom Picking.
Clicking on an atom with the mouse results in identification and the display of it residue name, residue number, atom name, atom serial number and chain in the command window. This behaviour may be disabled with the command **set picking none** and restored with the command **set picking ident**.

Disabling picking, by using **set picking off** is useful when executing the **pause** command in RasMol scripts as it prevents the display of spurious message on the command line whist the script is suspended.

Measuring Distances, Angles and Torsions.
Interactive measurement of distances, angles and torsions is achieved using the commands
set picking distance
set picking monitor
set picking angle
and
set picking torsion
respectively. In these modes, clicking on an atom results in it being identified on the rasmol command line. In addition every atom picked increments a modulo counter such that in distance mode, every second atom displays the distance (or distance monitor) between this atom and the previous one. In angle mode, every third atom displays the angle between the previous three atoms and in torsion mode every fourth atom displays the torsion between the last four atoms. By holding down the shift key while picking an atom, this modulo counter isn't incremented and allows, for example, the distances of consecutive atoms from a fixed atom to be displayed. See the monitor command for how to control the display of distance monitor lines and labels.

Labelling Atoms with the Mouse.
The mouse may also be used to toggle the display of an atom label on a given atom. The RasMol command **set picking label** removes a label from a picked atom if it already has one, or displays a concise label at that atom position otherwise.

Centering Rotation with the Mouse.
A molecule may be centered on a specified atom position using the RasMol command
set picking centre or **set picking center**. In this mode, picking an atom causes all futher rotations to be about that point.

Set Radius

```
Syntax:  set radius {<value>}
```

The RasMol **`set radius`** command is used to alter the behaviour of the RasMol **`dots`** command depending upon the value of the **`solvent`** parameter. When **`solvent`** is **`true,`** the **`radius`** parameter controls whether a true Van der Waal's surface is generated by the **`dots`** command. If the value of **`radius`** is anything other than zero, that value is used as the radius of each atom instead of it true VdW value. When the value of **`solvent`** is **`true,`** this parameter determines the `probe sphere' (solvent) radius. The parameter may be given as an integer in rasmol units or containing a decimal point in Angstroms. The default value of this parameter is determined by the value of **`solvent`** and changing **`solvent`** resets **`radius`** to its new default value.

Set Shadow

```
Syntax:  set shadow <boolean>
```

The RasMol **set shadow** command enables and disables raytracing of the currently rendered image. Currently only the spacefilling representation is shadowed or can cast shadows. Enabling shadowing will automatically disable the Z-clipping (slabbing) plane using the command slab off. Raytracing typically takes about 10s for a moderately sized protein. It is recommended that shadowing is normally disabled whilst the molecule is being transformed or manipulated, and only enabled once an appropiate viewpoint is selected, to provide a greater impression of depth.

Set SlabMode

```
Syntax:  set slabmode <slabmode>
```

The RasMol **slabmode** parameter controls the rendering method of *spacefilled* atoms cut by the slabbing (z-clipping) plane. The rendering of *bonds, cartoons, ribbons, and strands* is a reject mode (see below), and is not affected by the slab mode. Valid slabmode parameters are **reject** (the **default** for bonds and cartoons), **half**, **hollow**, **solid** (the **default** for spacefilled atoms), and **section**.

There is no slab mode in RasMol which displays a true slab, namely, a section of constant nonzero thickness. Such a slab is defined by two cutting planes, one behind the other. Everything behind the rear plane, or in front of the front plane, is hidden. (In contrast to RasMol, the slab mode in MAGE is a true slab; slab thickness is controlled by the "zslab" slider, and the center of the slab can be moved to the front or rear by the "ztran" slider.)

- **reject**: All atoms and bonds are rendered whole. Atoms or bonds cut by the z-clipping plane are entirely hidden. Cartoons, ribbons, and strands are cut at or near the z-clipping plane. This is the **unconditional rendering for everything except spacefilled atoms**.
- **half**: Only the front halves of spacefilled atoms which are cut by the z-clipping plane are shown, and they are rendered as a hollow spherical surface with the portion in front of the z-clipping plane hidden. Therefore you can see through the holes in the fronts of these "ring" atoms to atoms behind.
- **hollow**: Spacefilled atoms which are cut by the z-clipping plane are rendered as a hollow spherical surface with the portion in front of the z-clipping plane hidden. Therefore you can see inside these atoms to either the rear inside surface of the atom or to portions of other intersecting (bonded) atoms.
- **solid**: Spacefilled atoms cut by the z-clipping plane are rendered as a solid sphere with a flat cut surface (as though made of clay and cut with a knife). This is the **default for spacefilled atoms**.
- **section**: Only the cut surfaces of (solid) atoms cut by the z-clipping plane are rendered. All atoms behind or in front of the z-clipping plane are hidden. The image is a zero-thickness plane parallel to the viewing screen.

Set Solvent

```
Syntax:  set solvent <boolean>
```

The RasMol **set solvent** command is used to control the behaviour of the RasMol dots command. Depending upon the value of the **solvent** parameter, the dots command either generates a Van der Waal's or a solvent acessible surface around the currently selected set of atoms. Changing this parameter automatically resets the value of the RasMol radius parameter. The command **set solvent false,** the default value, indicates that a Van der Waal's surface should be generated and resets the value of radius to zero. The command **set solvent true** indicates that a `Connolly' or `Richards' solvent accessible surface should be drawn and sets the radius parameter, the solvent radius, to 1.2 Angstroms (or 300 RasMol units).

Set Specular

```
Syntax:  set specular <boolean>
```

The RasMol **set specular** command enables and disables the display of specular highlights on solid objects drawn by RasMol. Specular highlights appear as white reflections of the light source on the surface of the object. The current RasMol implementation uses an approximation function to generate this highlight.

The specular highlights on the surfaces of solid objects may be altered by using the specular reflection coefficient, which is altered using the RasMol **set specpower** command.

Set SpecPower

```
Syntax:  set specpower {<value>}
```

The **specpower** parameter determines the shininess of solid objects rendered by RasMol. This value between 0 and 100 adjusts the reflection coeffient used in specular highlight calculations. The specular highlights are enabled and disabled by the RasMol **set specular** command. Values around 20 or 30 produce plastic looking surfaces. High values represent more shiny surfaces such as metals, while lower values produce more diffuse/dull surfaces.

Set SSBonds

```
Syntax:  set ssbonds backbone
         set ssbonds sidechain
```

The RasMol **ssbonds** parameter determines whether disulphide bridges are drawn between the sulphur atoms in the sidechain (the default) or between the alpha carbon atoms in the backbone of the cysteines residues. The actual display of disulphide bridges is controlled by the **ssbonds** command. Drawing disulphide bridges between alpha carbons is useful when the rest of the protein is shown in only a schematic representation such as **backbone,** **ribbons** or **strands.** his parameter is similar to the RasMol **hbonds** parameter.

Set Stereo

command.

```
Syntax:  set stereo on
         set stereo off
         set stereo <boolean>
```

The RasMol **set stereo** parameter controls the separation between the left and right images. Turning stereo on and off doesn't reposition the center of the molecule.

The command **stereo on** . The separation angle between the two views may be adjusted with the **set stereo [-] <number>** command, where positive values result in relaxed viewing and negative values in crossed viewing. Currently, stereo viewing is not supported in vector PostScript output files.

Set Strands

```
Syntax:  set strands {<value>}
```

The RasMol **strands** parameter controls the number of parallel strands that are displayed in the ribbon representations of proteins. The permissible values for this parameter are 1, 2, 3, 4, 5 and 9. The default value is

5. The number of strands is constant for all ribbons being displayed. However, the ribbon width (the separation between strands) may be controlled on a residue by residue basis using the RasMol `ribbons` command.

Set Transparent

```
Syntax:  set transparent on
         set transparent off
```

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

Set UnitCell

```
Syntax:  set unitcell <boolean>
```

The RasMol **`unitcell`** parameter controls the display of the crystallographic unit cell on the current display. The crystal cell is only enabled if the appropriate crystal symmetry information is contained in the PDB data file. The RasMol command `show symmetry` display details of the crystal's space group and unit cell axes. The **`set unitcell`** command is similar the the commands `set axes` and `set boundbox` that display orthogonal co-ordinate axes and the bounding box respectively.

Set VectPS

```
Syntax:  set vectps <boolean>
```

The RasMol **`vectps`** parameter is use to control the way in which the RasMol `write` command generates vector PostScript output files. The command **`set vectps on`** enables to use of black outlines around spheres and cylinder bonds producing `cartoon-like' high resolution output. However, the current implementation of RasMol incorrectly cartoons spheres that are intersected by more than one other sphere. Hence `ball and stick' models are rendered correctly by not large spacefilling spheres models. Cartoon outlines can be disabled, the default, by the command **`set vectps off`**

Set Write

```
Syntax:  set write <boolean>
```

The RasMol **`set write`** command enables (and disables) the use of **`save`** and **`write`** in scripts, but may only be executed from the command line. By default, this value is **`false`**, prohibiting the generation of files in any scripts executed at start-up (such as those launched from a WWW browser such as Mosaic or NetScape). However, animators may start up RasMol interactively, type **`set write on`** and then execute a script to generate each frame using the source command.

Atom Expressions

RasMol atom expressions uniquely identify an arbitrary group of atoms within a molecule. Atom expressions are composed of either primitive expressions, predefined sets, comparison operators, **`within`** expressions, or logical (boolean) combinations of the above expression types.

The logical operators allow complex queries to be constructed out of simpler ones using the standard boolean connectives **and**, **or** and **not**. These may be abbreviated by the symbols "**&**", "**|**", and "**!**", respectively. Parentheses (brackets) may be used to alter the precedence of the operators. For convenience, a comma may also be used for boolean disjunction.

The atom expression is evaluated for each atom, hence **protein and backbone** selects protein bacbone atoms, not the protein and [nucleic] acid backbone atoms!

```
Examples:    backbone and not helix
             within( 8.0, ser70 )
             not (hydrogen or hetero)
             not *.FE and hetero
             8, 12, 16, 20-28
             arg, his, lys
```

Example Expressions

The following table gives some useful examples of RasMol atom expressions.

Expression	**Interpretation**
*	All atoms
cys	Atoms in cysteines
hoh	Atoms in heterogenous water molecules
as?	Atoms in either asparagine or aspartic acid
*120	Atoms at residue 120 of all chains
*p	Atoms in chain P
*.n?	Nitrogen atoms
cys.sg	Sulphur atoms in cysteine residues
ser70.c?	Carbon atoms in serine-70
hem*p.fe	Iron atoms in the Heme groups of chain P

Primitive Expressions

RasMol primitive expressions are the fundamental building blocks of atom expressions. There are two types of primitive expression. The first type is used to identify a given residue number or range of residue numbers. A single residue is identified by its number (position in the sequence), and a range is specified by lower and upper bounds separated by a hyphen character. For example **select 5,6,7,8** is also **select 5-8.** Note that this selects the given residue numbers in all macromolecule chains.

The second type of primitive expression specifies a sequence of fields that must match for a given atom. The first part specifies a residue (or group of residues) and an optional second part specifies the atoms within those residues. The first part consists of a residue name, optionally followed by a residue number and/or chain identifier.

A residue name typically consists of up to three alphabetic characters, which are case insensitive. Hence the primitive expressions **SER** and **ser** are equivalent, identifying all serine residues. Residue names that contain non-alphabetic characters, such as sulphate groups, may be delimited using square brackets, i.e. **[SO4]**

The residue number is the residue's position in the macromolecule sequence. Negative sequence numbers are permited. For example, **SER70** Care must be taken when specifying both residue name and number, it the group at the specified position isn't the specified residue no atoms are selected.

The chain identifier is typically a single case-insensitive alphabetic or numeric character. Numeric chain identifiers must be distinguished or separated from residue numbers by a colon character. For example, **SER70A** or **SER70:1**

The second part consists of a period character followed by an atom name. An atom name may be up to four alphabetic or numeric characters.

An asterisk may be used as a wild card for a whole field and a question mark as a single character wildcard.

Comparison Operators

Parts of a molecule may also be distinguished using equality, inequality and ordering operators on their properties. The format of such comparison expression is a property name, followed by a comparison operator and then an integer value.

The atom properties that may be used in RasMol are **atomno** for the atom serial number, **elemno** for the atom's atomic number (element), **resno** for the residue number, **radius** for the spacefill radius in RasMol units (or zero if not represented as a sphere) and **temperature** for the PDB anisotropic temperature value.

The equality operator is denoted either **=** or **==**. The inequality operator as either **<>**, **!=**, or **/=**. The ordering operators are **<** for less than, **<=** for less than or equal to, **>** for greater than, and **>=** for greater than or equal to.

```
Examples:    resno < 23
             temperature >= 900
             atomno == 487
```

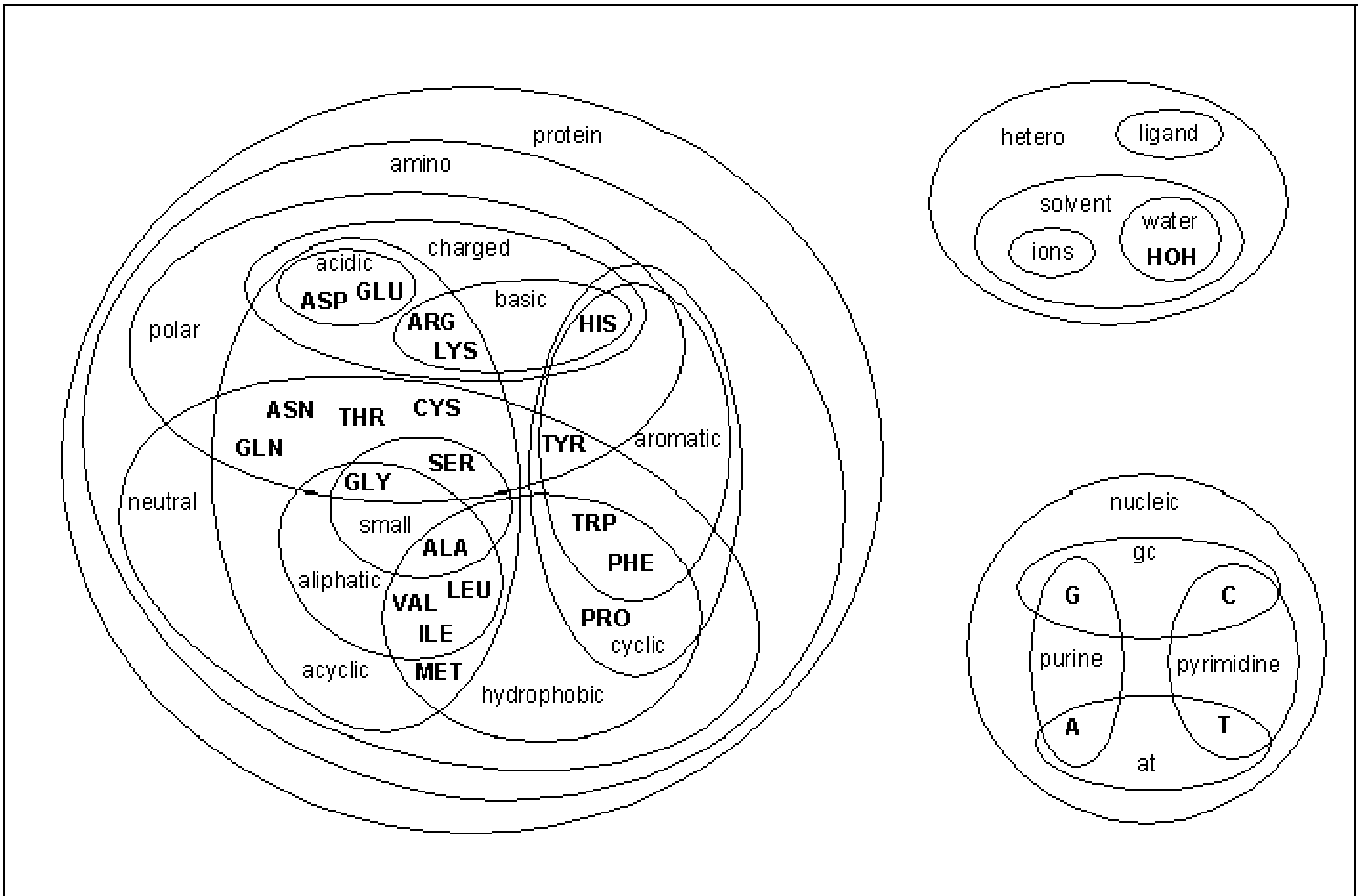
Within Expressions

A RasMol **within** expression allows atoms to be selected on their proximity to another set of atoms. A **within** expression takes two parameters separated by a comma and surrounded by parenthesis. The first argument is an integer value called the "cut-off" distance of the within expression and the second argument is any valid atom expression. The cut-off distance is expressed in either integer RasMol units or Angstroms containing a decimal point. An atom is selected if it is within the cut-off distance of any of the atoms defined by the second argument. This allows complex expressions to be constructed containing nested **within** expressions.

For example, the command **select within(3.2,backbone)** selects any atom within a 3.2 Angstrom radius of any atom in a protein or nucleic acid backbone. **Within** expressions are particularly useful for selecting the atoms around an active site.

Predefined Sets

RasMol atom expressions may contain predefined sets. These sets are single keywords that represent portions of a molecule of interest. Predefined sets are often abbreviations primitive atom expressions, and in some cases of selecting areas of a molecule that could not otherwise be distinguished. The Venn diagram below, kindly contributed by Kurt Giles, summarizes most of the predefined terms. A hyperlinked list of the currently predefined sets is tabulated below the diagram. In addition to the sets listed here, RasMol also treats element names (and their plurals) as predefined sets containing all atoms of that element type, i.e. the command `select oxygen` is equivalent to the command `select elemno=8.`



RasMol/Chime Venn diagram v1.2
Drawn by Kurt Giles (kurt@inn-prot.weizmann.ac.il) for the Israeli National Node of EMBnet

AT
Acidic
Acyclic
Aliphatic
Alpha
Amino
Aromatic
Backbone
Basic
Bonded
Buried
CG
Charged

Cyclic
Cystine
Helix
Hetero
Hydrogen
Hydrophobic
Ions
Large
Ligand
Medium
Neutral
Nucleic

Polar
Protein
Purine
Pyrimidine
Selected
Sheet
Sidechain
Small
Solvent
Surface
Turn
Water

AT Set

This set contains the atoms in the complementary nucleotides adenosine and thymidine (A and T respectively). All nucleotides are classified as either the set `at` or the set `cg` This set is equivalent to the RasMol atom expressions "`a,t`" and "`nucleic and not cg`"

Acidic Set

The set of acidic amino acids. These are the residue types Asp and Glu. All amino acids are classified as either **`acidic`**, [**`basic`**]() **`or`** [**`neutral.`**]() This set is equivalent to the RasMol atom expressions "**`asp, glu`**" and "**`amino and not (basic or neutral)`**"

Acyclic Set

The set of atoms in amino acids not containing a cycle or ring. All amino acids are classified as either [**`cyclic`**]() or **`acyclic.`** This set is equivalent to the RasMol atom expression "**`amino and not cyclic`**"

Aliphatic Set

This set contains the aliphatic amino acids. These are the amino acids Ala, Gly, Ile, Leu and Val. This set is equiavlent to the RasMol atom expression "**`ala, gly, ile, leu, val`**"

Alpha Set

The set of alpha carbons in the protein molecule. This set is approximately equivalent to the RasMol atom expression "**`*.CA`**" This command should not be confused with the predefined set [**`helix`**]() which contains the atoms in the amino acids of the protein's alpha helices.

Amino Set

This set contains all the atoms contained in amino acid residues. This is useful for distinguishing the protein from the nucleic acid and heterogenous atoms in the current molecule database.

Aromatic Set

The set of atoms in amino acids containing aromatic rings. These are the amino acids His, Phe, Trp and Tyr. Because they contain aromatic rings all members of this set are member of the predefined set [**`cyclic.`**]() This set is equivalent to the RasMol atom expressions "**`his, phe, trp, tyr`**" and "**`cyclic and not pro`**"

Backbone Set

This set contains the four atoms of each amino acid that form the polypeptide N-C-C-O backbone of proteins, and the atoms the sugar phosphate backbone of nucleic acids. Use the RasMol predefined sets **`protein`** and **`nucleic`** to distinguish between the two forms of backbone. Atoms in nucleic acids and proteins are either **`backbone`** or [**`sidechain.`**]() This set is equivalent to the RasMol expression "**`(protein or nucleic) and not sidechain`**"

The predefined set **`mainchain`** is synonymous with the set **`backbone.`**

Basic Set

The set of basic amino acids. These are the residue types Arg, His and Lys. All amino acids are classified as either `acidic,` `basic` or `neutral.` This set is equivalent to the RasMol atom expressions "`arg, his, lys`" and "`amino and not (acidic or neutral)`"

Bonded Set

This set contain all the atoms in the current molecule database that are bonded to atleast one other atom.

Buried Set

This set contains the atoms in those amino acids that tend (prefer) to buried inside protein, away from contact with solvent molecules. This set refers to the amino acids preference and not the actual solvent acessibility for the current protein. All amino acids are classified as either `surface` or `buried.` This set is equivalent to the RasMol atom expression "`amino and not surface`", or "`ala, leu, val, ile, phe, cys, met, trp`", or "`(hydrophobic and not pro) or cys`".

CG Set

This set contains the atoms in the complementary nucleotides cytidine and guanoine (C and G respectively). All nucleotides are classified as either the set `at` or the set `cg` This set is equivalent to the RasMol atom expressions "`c,g`" and "`nucleic and not at`"

Charged Set

This set contains the charged amino acids. These are the amino acids that are either `acidic` or `basic.` Amino acids are classified as being either `charged` or `neutral.` This set is equivalent to the RasMol atom expressions "`acidic or basic`" and "`amino and not neutral`"

Cyclic Set

The set of atoms in amino acids containing a cycle or rings. All amino acids are classified as either `cyclic` or `acyclic.` This set consists of the amino acids His, Phe, Pro, Trp and Tyr. The members of the predefined set `aromatic` are members of this set. The only cyclic but non-aromatic amino acid is proline. This set is equivalent to the RasMol atom expressions "`his, phe, pro, trp, tyr`" and "`aromatic or pro`" and "`amino and not acyclic`"

Cystine Set

This set contains the atoms of cysteine residues that form part of a disulphide bridge, i.e. half cystines. RasMol automatically determines disulphide bridges, if neither the predefined set `cystine` nor the RasMol `ssbonds` command have been used since the molecule was loaded. The set of free cysteines may be determined using the RasMol atom expression "`cys and not cystine`"

Helix Set

This set contains all atoms that form part of a protein alpha helix as determined by either the PDB file author or Kabsch and Sander's DSSP algorithm. By default, RasMol uses the secondary structure determination given in the PDB file if it exists. Otherwise, it uses the DSSP algorithm as used by the RasMol `structure` command.

This predefined set should not be confused with the predefined set `alpha` which contains the alpha carbon atoms of a protein.

Hetero Set

This set contains all the heterogenous atoms in the molecule. These are the atoms described by HETATM entries in the PDB file. These typically contain water, cofactors and other solvents and ligands. All `hetero` atoms are classified as either `ligand` or `solvent` atoms. These heterogenous `solvent` atoms are further classified as either `water` or `ions.`

Hydrogen Set

This predefined set contains all the hydrogen and deuterium atoms of the current molecule. This predefined set is equivalent to the RasMol atom expression "`elemno=1`"

Hydrophobic Set

This set contains all the hydrophobic amino acids. These are the amino acids Ala, Leu, Val, Ile, Pro, Phe, Met and Trp. All amino acids are classified as either `hydrophobic` or `polar.` This set is equivalent to the RasMol atom expressions "`ala, leu, val, ile, pro, phe, met, trp`" and "`amino and not polar`"

Ions Set

This set contains all the heterogenous phosphate and sulphate ions in the current molecule data file. A large number of these ions are sometimes associated with protein and nucleic acid structures determined by X-ray crystallography. These atoms tend to clutter an image. All `hetero` atoms are classified as either `ligand` or `solvent` atoms. All `solvent` atoms are classified as either `water` or `ions.`

Large Set

All amino acids are classified as either `small,` `medium` or `large.` This set is equivalent to the RasMol atom expression "`amino and not (small or medium)`"

Ligand Set

This set contains all the heterogenous cofactor and ligand moieties that are contained in the current molecule data file. At this set is defined to be all `hetero` atoms that are not `solvent` atoms. Hence this set is equivalent to the RasMol atom expression "`hetero and not solvent`"

Medium Set

All amino acids are classified as either small, **medium** or large. This set is equivalent to the RasMol atom expression "**amino and not (large or small)**", or "**val, thr, asp, asn, pro, cys**".

Neutral Set

The set of neutral amino acids. All amino acids are classified as either acidic, **basic** or neutral. This set is equivalent to the RasMol atom expression "**amino and not (acidic or basic)**"

Nucleic Set

The set of all atoms in nucleic acids, which consists of the four nucleotide bases adenosine, cytidine, guanosine and thymidine (A, C, G and T respectively). All neucleotides are classified as either purine or pyrimidine. This set is equivalent to the RasMol atom expressions "**a,c,g,t**" and "**purine or pyrimidine**"

Polar Set

This set contains the polar amino acids. All amino acids are classified as either hydrophobic or **polar.** This set is equivalent to the RasMol atom expression "**amino and not hydrophobic**"

Protein Set

The set of all atoms in proteins. This consists of the RasMol predefined set amino and common post-translation modifications.

Purine Set

The set of purine nucleotides. These are the bases adenosine and guanosine (A and G respectively). All nucleotides are either **purines** or pyrimidines. This set is equivalent to the RasMol atom expressions "**a,g**" and "**nucleic and not purine**"

Pyrimidine Set

The set of pyrimidine nucleotides. These are the bases cytidine and thymidine (C and T respectively). All nucleotides are either purines or **pyrimidines.** This set is equivalent to the RasMol atom expressions "**c,t**" and "**nucleic and not pyrimidine**"

Selected Set

This set contains the set of atoms in the currently selected region. The currently selected region is defined by the preceding select or restrict command and not the atom expression containing the **selected** keyword.

Sheet Set

This set contains all atoms that form part of a protein beta sheet as determined by either the PDB file author or Kabsch and Sander's DSSP algorithm. By default, RasMol uses the secondary structure determination given in the PDB file if it exists. Otherwise, it uses the DSSP algorithm as used by the RasMol `structure` command.

Sidechain Set

This set contains the functional sidechains of any amino acids and the base of each nucleotide. These are the atoms not part of the polypeptide N-C-C-O backbone of proteins or the sugar phosphate backbone of nucleic acids. Use the RasMol predefined sets `protein` and `nucleic` to distinguish between the two forms of sidechain. Atoms in nucleic acids and proteins are either `backbone` or `sidechain.` This set is equivalent to the RasMol expression "`(protein or nucleic) and not backbone`"

Small Set

All amino acids are classified as either `small,` `medium` or `large.` This set is equivalent to the RasMol atom expression "`amino and not (medium or large)`", or "`ala, gly, ser`".

Solvent Set

This set contains the solvent atoms in the molecule co-ordinate file. These are the heterogenous water molecules, phosphate and sulphate ions. All `hetero` atoms are classified as either `ligand` or `solvent` atoms. All `solvent` atoms are classified as either `water` or `ions.` This set is equivalent to the RasMol atom expressions "`hetero and not ligand`" and "`water or ions`"

Surface Set

This set contains the atoms in those amino acids that tend (prefer) to be on the surface of proteins, in contact with solvent molecules. This set refers to the amino acids preference and not the actual solvent accessibility for the current protein. All amino acids are classified as either `surface` or `buried.` This set is equivalent to the RasMol atom expression "`amino and not buried`" or "`gly, ser, thr, lys, asp, asn, glu, pro, arg, gln, tyr, his`" or "`(polar and not cys) or pro`".

Turn Set

This set contains all atoms that form part of a protein turns as determined by either the PDB file author or Kabsch and Sander's DSSP algorithm. By default, RasMol uses the secondary structure determination given in the PDB file if it exists. Otherwise, it uses the DSSP algorithm as used by the RasMol `structure` command.

Water Set

This set contains all the heterogenous water molecules in the current database. A large number of water molecules are sometimes associated with protein and nucleic acid structures determined by X-ray crystallography. These atoms tend to clutter an image. All `hetero` atoms are classified as either `ligand` or `solvent` atoms. The `solvent` atoms are further classified as either `water` or `ions.`

Colours and Colour Schemes

The RasMol colour command allows different objects (such as atoms, bonds and ribbon segments) to be given a specified colour. Typically this colour is either a RasMol predefined colour name or an RGB triple. Additionally RasMol also supports:

- cpk
- amino
- chain
- group
- shapely
- structure
- temperature
- charge and

- user colour schemes for atoms,
- hbond type for hydrogen bonds, and the
- electrostatic potential colour scheme for dot surfaces.

The currently predefined colour names are listed below with their corresponding RGB triplet.

NAME	RASMOL	HTML *
black	[0,0,0]	00 00 00
white	[255,255,255]	FF FF FF
blue	[0,0,255]	00 00 FF
green	[0,255,0]	00 FF 00
red	[255,0,0]	FF 00 00
yellow	[255,255,0]	FF FF 00
magenta	[255,0,255]	FF 00 FF
cyan	[0,255,255]	00 FF FF
orange	[255,165,0]	FF A5 00
redorange	[255,69,0]	FF 45 00
violet	[238,130,238]	EE 82 EE
purple	[160,32,240]	A0 20 F0
greenblue	[46,139,87]	2E 8B 57

*** RGB = Red, Green, Blue. Decimal values are given for assigning colors in RasMol with its 'color' command. Hexadecimal is given to match text colors in HTML, e.g. <font color="#8F8FFF">Nitrogen</font>, which looks like this: Nitrogen.**

Here is a CPK color key as an HTML table which could be used, for example, next to a Chime display:

C H O N S

Click here for the CPK color key alone. You can **view source**, then cut and paste the HTML code into your document.

If you frequently wish to use a colour not predefined, you can write a one-line script. For example, if you make the file **grey.spt** containing the line

```
colour [180,180,180] #grey
```

then the command **script grey.spt** colours the currently selected atom set grey. (Or, you can color it gray. :-)

Amino Colours

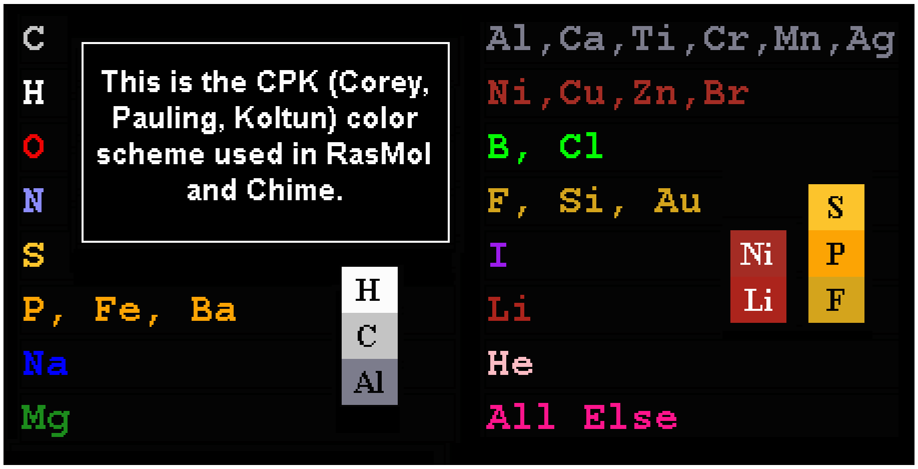
The RasMol **`amino`** colour scheme colours amino acids according to traditional amino acid properties. The purpose of colouring is to identify amino acids in an unusual or surprising environment. The outer parts of a protein that are polar are visible (bright) colours and non-polar residues darker. Most colours are hallowed by tradition. This colour scheme is similar to the **`shapely`** scheme.

```
 [verified against 2.6 source code 99/03/13]
   ASP,GLU    bright red [230,10,10]     E6 0A 0A
   LYS,ARG    blue       [20,90,255]     14 5A FF
   PHE,TYR    mid blue   [50,50,170]     32 32 AA
   GLY        light grey [235,235,235]   EB EB EB
   ALA        dark grey  [200,200,200]   C8 C8 C8
   HIS        pale blue  [130,130,210]   82 82 D2
   CYS,MET        yellow [230,230,0]     E6 E6 00
   SER,THR        orange [250,150,0]     FA 96 00
   ASN,GLN        cyan   [0,220,220]     00 DC DC
   LEU,VAL,ILE    green  [15,130,15]     0F 82 0F
   TRP            pink   [180,90,180]    B4 5A B4
   PRO, PCA, HYP flesh  [220,150,130]   DC 96 82
```

Chain Colours

The RasMol **`chain`** colour scheme assigns each macromolecular chain a unique colour. This colour scheme is particularly useful for distinguishing the parts of multimeric structure or the individual `strands' of a DNA chain.

CPK Colours



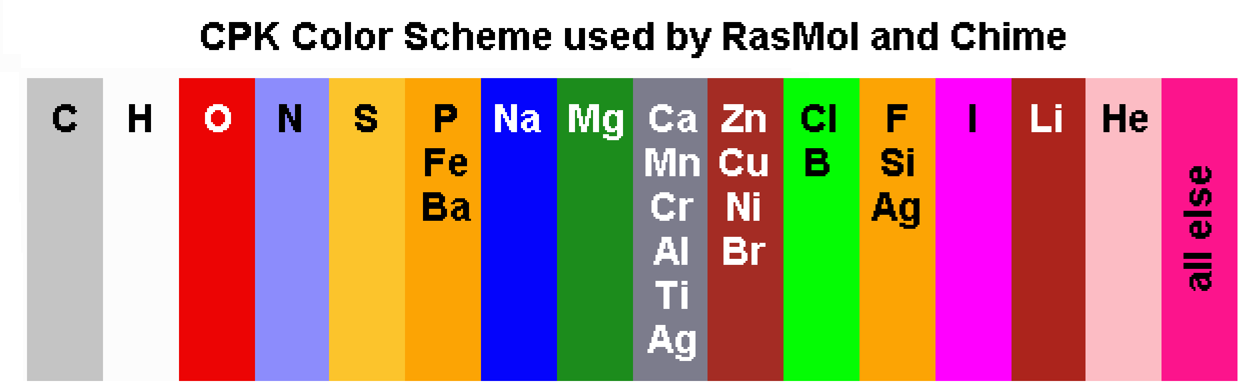
The RasMol **`cpk`** colour scheme is based upon the colours of the popular plastic spacefilling models which were developed by Corey, Pauling and later improved by Koltun. This colour scheme colours `atom' objects by the atom (element) type. This is the scheme conventionally used by chemists. The assignment of element type to colours is given below.

(Information below is based on RasMol's source code and as of February 19, 1998, is believed to be complete and correct here for the first time.)

```
    ELEMENT                    COLOR NAME       RGB DECIMAL    RGB HEXADECIMAL
    Carbon                     light grey       [200,200,200]  C8 C8 C8
    Oxygen                     red              [240,0,0]      F0 00 00
    Hydrogen                   white            [255,255,255]  FF FF FF

    Nitrogen                   light blue       [143,143,255]  8F 8F FF
    Sulphur                    sulphur yellow   [255,200,50]   FF C8 32
    Chlorine, Boron            green            [0,255,0]      00 FF 00
```

Phosphorus, Iron, Barium	orange	[255,165,0]	FF A5 00
Sodium	blue	[0,0,255]	00 00 FF
Magnesium	forest green	[34,139,34]	22 8B 22
Zn, Cu, Ni, Br	brown	[165,42,42]	A5 2A 2A
Ca, Mn, Al, Ti, Cr, Ag	dark grey	[128,128,144]	80 80 90
F, Si, Au	goldenrod	[218,165,32]	DA A5 20
Iodine	purple	[160,32.240]	A0 20 F0
Lithium	firebrick	[178,34,34]	B2 22 22
Helium	pink	[255,192,203]	FF C0 CB
Unknown	deep pink	[255,20,147]	FF 14 93


CPK Color Scheme used by RasMol and Chime

Group Colours

The RasMol `group` colour scheme colour codes residues by their position in a macromolecular chain. Each chain is drawn as a smooth spectrum from blue through green, yellow and orange to red. The N termini of proteins are coloured blue (similar to the CPK colour for nitrogen) and the C termini, red (similar to the CPK colour for oxygen). The 5' termini of nucleic acids are coloured blue, and the 3' termini, red.

If a chain has a large number of heterogenous molecules associated with it, the macromolecule may not be drawn in the full range of the spectrum. When RasMol performs group colouring it decides the range of colours it uses from the residue numbering given in the PDB file. Hence the lowest residue number is displayed in blue and the highest residue number is displayed as red. Unfortunately, if a PDB file contains a large number of hetatoms, such as water molecules, that occupy the high residue numbers, the protein is displayed in the blue-green end of the spectrum and the waters in the yellow-red end of the spectrum. This is aggrevated by there typically being many more water molecules than amino acid residues. The solution to this problem is to use the command `set hetero off` before applying the group colour scheme. This can also be achieved by toggling *Hetero atoms* on the *Options* menu before selecting *Group* on the *Colour* menu. This command instructs RasMol to only use non-hetero residues in the group colour scaling.

Shapely Colours

The RasMol `shapely` colour scheme colour codes residues by amino acid property. This scheme is based upon Bob Fletterick's "Shapely Models". Each amino acid and nucleic acid residue is given a unique colour. The `shapely` colour scheme is used by David Bacon's Raster3D program. This colour scheme is similar to the `amino` colour scheme.

Structure Colours

[Corrected against 2.6 source code 99/03/13 -- the helix and sheet rgb values were incorrect here prior to this date.]

The RasMol **structure** colour scheme colours the molecule by protein secondary structure. Alpha helices are coloured magenta, [255,0,128], beta sheets are coloured yellow, [255,200,0], turns are coloured pale blue, [96,128,255] and all other residues are coloured white. The secondary structure is either read from the PDB file (HELIX and SHEET records), if available, or determined using Kabsch and Sander's DSSP algorithm. The RasMol **structure** command may be used to force DSSP's structure assignment to be used.

Temperature Colours

The RasMol **temperature** colour scheme colour codes each atom according to the anisotropic temperature (beta) value stored in the PDB file. Typically this gives a measure of the mobility/uncertainty of a given atom's position. High values are coloured in warmer (red) colours and lower values in colder (blue) colours. This feature is often used to associate a "scale" value [such as amino acid variability in viral mutants] with each atom in a PDB file, and colour the molecule appropriately.

The difference between the **temperature** and **charge** colour schemes is that increasing temperature values proceed from blue to red, whereas increasing charge valuse go from red to blue.

Charge Colours

The RasMol **charge** colour scheme colour codes each atom according to the charge value stored in the input file (or beta factor field of PDB files). High values are coloured in blue (positive) and lower values coloured in red (negative). Rather than use a fixed scale this scheme determines the maximum and minimum values of the charge/temperature field and interpolates from red to blue appropriately. Hence, green cannot be assumed to be `no net charge' charge.

The difference between the **charge** and **temperature** colour schemes is that increasing temperature values proceed from blue to red, whereas increasing charge valuse go from red to blue.

If the charge/temperature field stores reasonable values it is possible to use the RasMol **colour dots potential** command to colour code a dot surface (generated by the **dots** command) by electrostatic potential.

User Colours

The RasMol **user** colour scheme allows RasMol to use the colour scheme stored in the PDB file. The colours for each atom are stored in COLO records placed in the PDB data file. This convention was introduced by David Bacon's Raster3D program.

If you wish to use a custom colouring scheme, it is typically much simpler to construct a short script to accomplish this than to insert values into the PDB file. The script also has the advantage that it can be invoked for any currently displayed molecule. An example of such a script is available.

HBond Type Colours

The RasMol **type** colour scheme applies only to hydrogen bonds, hence is used in the command "**colour hbonds type**" This scheme colour codes each hydrogen bond according to the distance along a protein chain between hydrogen bond donor and acceptor. This schematic representation was introduced by Belhadj-Mostefa and

Milner-White. This representation gives a good insight into protein secondary structure (hbonds forming alpha helices appear red, those forming sheets appear yellow and those forming turns appear magenta).

Offset	Colour	Triple
+2	white	[255,255,255]
+3	magenta	[255,0,255]
+4	red	[255,0,0]
+5	orange	[255,165,0]
-3	cyan	[0,255,255]
-4	green	[0,255,0]
default	yellow	[255,255,0]

Potential Colours

The RasMol **`potential`** colour scheme applies only to dot surfaces, hence is used in the command "**`colour dots potential`**" This scheme colours each currently displayed dot by the electrostatic potential at that point in space. This potential is calculated using Coulomb's law taking the temperature/charge field of the input file to be the charge assocated with that atom. This is the same interpretation used by the **`colour charge`** command. Like the **`charge`** colour scheme low values are blue/white and high values are red. The table below shows the static assignment of colours using a dielectric constant value of 10.

25 < V	red	[255,0,0]
10 < V < 25	orange	[255,165,0]
3 < V < 10	yellow	[255,255,0]
0 < V < 3	green	[0,255,0]
-3 < V < 0	cyan	[0,255,255]
-10 < V < 3	blue	[0,0,255]
-25 < V < -10	purple	[160,32,240]
V < -25	white	[255,255,255]

Revision History

- 99/03/13: Corrected spelling of Koltun (the K in CPK). Corrected structure color rgb values by comparison with the source code. Checked Amino colors against the source code but no corrections were needed.
- 98/08/10: Added Venn diagram of predefined terms contributed by Kurt Giles.
- 98/04/26: Added definitions of Medium, Small (whence Large), Buried, and Surface predefined terms obtained by examination of the source code. Prepared a separate introduction to select expressions, linked under the Select command.
- 98/03/23: Added paragraph at the very beginning with link to the original source of this reference manual, and reorganized the section before the table of contents.
- 98/02/28: Provided more detailed descriptions of the slab and set slabmode commands.
- 98/02/19: Added the **vectps** option to the write command. Provided link to discussion of high-resolution printing in the FAQ under the write and print commands. Completed the table of CPK color assignments used by RasMol, and added a colored GIF image illustrating them.
- 97/07/25: Added some missing RGB values in the CPK color scheme, and also a sample color key table in HTML.
- 97/07/04: Added hexadecimal RGB values for amino and predefined colors.
- 97/06/29: Added RGB values for the CPK Color Scheme in both decimal and hexidecimal.
- 97/03/16: Expanded the section on the group colour scheme, explaining the use of **`set hetero off`** to control the spectrum of colors. Also corrected the colours given for the ends of proteins, which were inadvertantly reversed in the reference manual.
- 96/09/02: Added new section on the **trace** command, and revised **backbone** accordingly. Added **dashes** option to backbone, strands, and wireframe commands. Added section for **set bonds**. Added more detail

about use of **pause** including synonymy with **wait**. Added section on **set bonds**. Indexed the section on **set write** under **set** (it had inadvertantly been omitted). Fixed incorrectly formatted "disappearing" instances of "<filename>". Added new section on **exit** which is now distinct from **quit**.

- 96/08/14: Filename changed from rasman26.htm to **rasman.htm** so links will remain version-independent. Minor typos and formatting changes.
- 96/08/13: Updated Introduction and added General Operations sections, which were taken from the non-html reference manual bundled with version 2.5. These were revised to bring them into accord with version 2.6 beta-2 (see comments in html source).

→