#[1]$[2]+[3]

RasMol V2.3

Molecular Visualisation Program

Roger Sayle
Glaxo Research and Development
Greenford, Middlesex, U.K.

Table of Contents

1rasmol
2RasMol V2.3
3doc

+[5]#[6]$[7]Introduction

RasMol2 is an X11 windows system tool intended for the visualisation of proteins and nucleic acids. RasMol requires either an 8bit pseudo colour or a 24bit (32bit) true colour display. The program reads in a specified Brookhaven protein databank (PDB) file and determines the connectivity from the residue information provided. This may then rendered on the screen in a variety of formats and colour schemes. Currently available molecule representations include depth-cued wireframes, sticks, space filling `union of spheres', ball and stick models and protein ribbon diagrams.

+[8]#[9]$[10]**Command Reference**

RasMol allows the execution of interactive commands typed at the "**RasMol>**" prompt in the terminal 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	colour	exit	hbond
help	load	quit	reset
restrict	ribbon	rotate	save
script	select	set	show
slab	spacefill	structure	ssbond
translate	wireframe	write	zap

8doc
9chcomref
10Command Reference

+[11]#[12]$[13]K[14]**Backbone**

Syntax: **backbone {<boolean>}**
backbone <value>

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 determine the cylinder radius of the representation in 0.004 angstrom units. Backbone objects may be coloured using the RasMol colour backbone command. A parameter value of 500 (2 angstroms) or above results in an "Integer argument too large" error.

11doc
12backbone
13Backbone
14backbone

+[15]#[16]$[17]K[18]**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.

15doc
16background
17Background
18background

+[19]#[20]#[21]$[22]K[23]**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.

19doc
20center
21centre
22Centre
23centre

+[24]#[25]#[26]$[27]K[28]**Colour**

Syntax: **colour {<object>} <colour>**
color {<object>} <colour>

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. Typing the command help colours will give a list of all the predefined colour names recognised by RasMol.

Allowed objects are **atoms**, **bonds**, backbone, hbonds, ribbons 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 all objects accept atoms, 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). **Atom** objects can also be coloured by amino, cpk, chain, group, shapely, structure, temperature and user and hydrogen bond objects can also be coloured by type. For more information type help colour <colour>.

24doc
25color
26colour
27Colour
28colour

+[29]#[30]#[31]$[32]K[33]**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.

29doc
30hbond
31hbonds
32HBonds
33hbonds

+[34]#[35]$[36]K[37]**Help**

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

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

34doc
35help
36Help
37help

+[38]#[39]$[40]K[41]**Load**

```
Syntax:   load {pdb} <filename>
          load alchemy <filename>
```

Load either a Brookhaven Protein Databank (PDB) file or Alchemy(tm) format file into RasMol2. Only a single PDB file may be loaded at a time. This command selects all the atoms in the molecule, and sets the default representation to be a cpk coloured wireframe model.

38doc
39load
40Load
41load

+[42]#[43]#[44]$[45]K[46]**Quit**

Syntax: **quit**
exit

Exit from the RasMol program.

42doc
43exit
44quit
45Quit
46quit

+[47]#[48]#[49]$[50]K[51]**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.

47doc
48renum
49renumber
50Renumber
51renumber

+[52]#[53]$[54]K[55]**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.

52doc
53reset
54Reset
55reset

+[56]#[57]$[58]K[59]**Restrict**

Syntax: restrict {<expression>}

The RasMol **restrict** command both defines the currently active zone 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 zone. 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-active zone.

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

56doc
57restrict
58Restrict
59restrict

+[60]#[61]#[62]$[63]K[64]**Ribbons**

Syntax: ribbons {<boolean>}
ribbons <value>

The RasMol **ribbons** command displays the currently loaded protein 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 width of the ribbon at each position is determined by the optional parameter in the usual RasMol units. By default this value is 380, which produces a ribbon 1.52 Angstroms wide. The number of strands in the ribbon may be altered using the RasMol set strands command. The rendering of the ribbon may also be changed using the set ribbons command.

60doc
61ribbon
62ribbons
63Ribbons
64ribbons

+[65]#[66]$[67]K[68]**Rotate**

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

Rotate the molecule about the specified axis. Permited values for the axis parameter are "**x**", "**y**" and "**z**". 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.

65doc
66rotate
67Rotate
68rotate

+[69]#[70]$[71]K[72]**Save**

Syntax: **save {pdb} <filename>**
save alchemy <filename>

Save the currently selected set of atoms in either a Brookhaven Protein Database (PDB) or Alchemy(tm) format file. This command should not be confused with the RasMol write command which generates either image or script files.

69doc
70save
71Save
72save

+[73]#[74]$[75]K[76]**Script**

Syntax: script <filename>

The RasMol **script** command reads a set of commands sequentially from a text file and executes them. This allows sequences of commonly used commands to be stored and performed by a 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.

73doc
74script
75Script
76script

+[77]#[78]$[79]K[80]**Select**

Syntax: select {<expression>}

Define the currently active zone of the molecule. All subsequent RasMol commands that manipulate a molecule or modify its colour or representation, only effects the currently selected zone. The parameter of a **select** command is a RasMol expression that is evaluated for every atom of the current molecule. The currently selected (active) zone of the molecule are those atoms that cause the expression to evaluate true. To select the whole molecule use the RasMol command **select all.**

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

77doc
78select
79Select
80select

+[81]#[82]$[83]K[84]**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, ommiting the paramter option resets that parameter to its default value. A list of valid parameter names is given below. For more information on each internal parameter type help set <parameter>.

ambient	background	bondmode	hbonds
hetero	hourglass	hydrogen	mouse
shadow	slabmode	specular	specpower
ssbonds	strands		

81doc
82set
83Set
84set

+[85]#[86]$[87]K[88]**Show**

```
Syntax:    show information
           show sequence
```

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.

85doc
86show
87Show
88show

+[89]#[90]$[91]K[92]**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. Integer 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.

89doc
90slab
91Slab
92slab

+[93]#[94]$[95]K[96]**Spacefill**

```
Syntax:   spacefill {<boolean>}
          spacefill temperature
          spacefill user
          spacefill <value>
```

Represent the currently selected zone as a spacefilling union of spheres model. An integer parameter may be used to specify the radius of each atom given in 4nm units. If no parameter is given, each atom is drawn as a sphere of its Van der Waals radius.

The **`temperature`** option is used to set the radius of each selected sphere to the value in the temperature field of the molecule file. A zero or negative value causes no change in the selected atom. Temperature values greater than 2.00 are truncated to 2.00 Angstrom radius.

The **`user`** option allows the radius of the selected spheres to be determined by matching each atom against optional lines in the input data file. Details of the wildcard pattern matching used by Raster3D's COLOR records is given in the manual.

93doc
94spacefill
95Spacefill
96spacefill

+[97]#[98]#[99]$[100]K[101]**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.

97doc
98ssbond
99ssbonds
100SSBonds
101ssbonds

+[102]#[103]$[104]K[105]**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 and ladders found.

102doc
103structure
104Structure
105structure

+[106]#[107]$[108]K[109]**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.

106doc
107translate
108Translate
109translate

+[110]#[111]$[112]K[113]**Wireframe**

Syntax: **wireframe {<boolean>}**
wireframe <value>

Represent each bond within the selected zone of the molecule as either a cylinder or depth-cued vector. If no parameter is given, RasMol draws each bond as a hither-and-yon shaded narrow vector. An integer parameter specifies the radius of a cylinder, given in 4nm units, to be used as a stick bond.

110doc
111wireframe
112Wireframe
113wireframe

+[114]#[115]$[116]K[117]**Write**

Syntax: write {<format>} <filename>

Write the current image to a file in a standard raster format. Currently supported file formats include "**gif**" (Compuserve GIF), "**ppm**" (Portable Pixmap), "**ras**" (Sun rasterfile), "**ps**" and "**epsf**" (Encapsulated PostScript), "**monops**" (Monochrome Encapsulated PostScript) and "**bmp**" (Microsoft bitmap). This command should not be confused with the RasMol save command which save the currently selected portion of the molecule.

114doc
115write
116Write
117write

+[118]#[119]$[120]K[121]**Zap**

Syntax: zap

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

118doc
119zap
120Zap
121zap

+[122]#[123]$[124]K[125]**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 between 10 and 200 specifies the desired magnification as a percentage of the default scale.

122doc
123zoom
124Zoom
125zoom

+[126]#[127]$[128]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 below.

ambient	background	bondmode	hbonds
hetero	hourglass	hydrogen	mouse
shadow	slabmode	specular	specpower
ssbonds	strands		

126doc
127chsetopt
128Internal Parameters

+[129]#[130]$[131]K[132]**Set Ambient**

K[133]**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.

129doc
130setambient
131Set Ambient
132set ambient
133ambient

+[134]#[135]$[136]K[137]**Set Background**

K[138]**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.

134doc
135setbackground
136Set Background
137set background
138background

+[139]#[140]$[141]K[142]**Set BondMode**

K[143]**Syntax: set bondmode and**
set bondmode or

set bondmode

139doc
140setbondmode
141Set BondMode
142set bondmode
143bondmode

+[144]#[145]$[146]K[147]**Set Display**

K[148]**Syntax: set display selected**
set display normal

set display

144doc
145setdisplay
146Set Display
147set display
148display

+[149]#[150]$[151]K[152]**Set HBonds**

K[153]K[154]K[155]**Syntax: set hbonds backbone**
set hbonds sidechain

set hbonds

149doc
150sethbonds
151Set HBonds
152set hbonds
153hbonds
154sidechain
155backbone

+[156]#[157]$[158]K[159]**Set Hetero**

K[160]**Syntax: set hetero <boolean>**

set hetero

156doc
157sethetero
158Set Hetero
159set hetero
160hetero

+[161]#[162]$[163]K[164]**Set HourGlass**

K[165]**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.

161doc
162sethourglass
163Set HourGlass
164set hourglass
165hourglass

+[166]#[167]$[168]K[169]**Set Hydrogen**

K[170]**`Syntax:  set hydrogen <boolean>`**

set hydrogen

166doc
167sethydrogen
168Set Hydrogen
169set hydrogen
170hydrogen

+[171]#[172]$[173]K[174]**Set Mouse**

K[175]K[176]K[177]K[178]**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.

171doc
172setmouse
173Set Mouse
174set mouse
175mouse
176rasmol
177insight
178quanta

+[179]#[180]$[181]K[182]**Set Ribbons**

K[183]K[184]K[185]**Syntax: set ribbons strands**
set ribbons solid

The RasMol **set ribbons** command controls the way that macromolecular ribbons are displayed. The default value **strands** display macromolecular ribbons as parallel depth-cued strands that pass along the protein or nucleic acid backbone. The number of strands in the ribbon may be altered using the RasMol set strands command. The **set ribbons solid** command renders the macromolecular ribbon as a solid shaded ribbon.

179doc
180setribbons
181Set Ribbons
182set ribbons
183ribbons
184strands
185solid

+[186]#[187]$[188]K[189]**Set Shadow**

K[190]**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.

186doc
187setshadow
188Set Shadow
189set shadow
190shadow

+[191]#[192]$[193]K[194]**Set SlabMode**

K[195]K[196]K[197]K[198]K[199]K[200]**Syntax: set slabmode <slabmode>**

The RasMol **slabmode** parameter controls the rendering method of objects cut by the slabbing (z-clipping) plane. Valid slab modes are "**reject**", "**half**", "**hollow**", "**solid**" and "**section**".

191doc
192setslabmode
193Set SlabMode
194set slabmode
195slabmode
196reject
197half
198hollow
199solid
200section

+[201]#[202]$[203]K[204]**Set Specular**

K[205]**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.

201doc
202setspecular
203Set Specular
204set specular
205specular

+[206]#[207]$[208]K[209]**Set SpecPower**

K[210]**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.

206doc
207setspecpower
208Set SpecPower
209set specpower
210specpower

+[211]#[212]$[213]K[214]**Set SSBonds**

K[215]K[216]K[217]**Syntax: set ssbonds backbone**
set ssbonds sidechain

set ssbonds

211doc
212setssbonds
213Set SSBonds
214set ssbonds
215ssbonds
216backbone
217sidechain

+[218]#[219]$[220]K[221]**Set Strands**

K[222]**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.

218doc
219setstrands
220Set Strands
221set strands
222strands

+[223]#[224]$[225]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!

223doc
224chexprs
225Atom Expressions

+[226]#[227]$[228]K[229]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

226doc
227exampleexpressions
228Example Expressions
229example expressions

+[230]#[231]$[232]K[233]**Primitive Expressions**

RasMol primitive expressions are the fundamental building blocks of atom expressions. There a two basic 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 cysteine 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.

230doc
231primitiveexpressions
232Primitive Expressions
233primitive expressions

+[234]#[235]$[236]K[237]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, **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
```

234doc
235comparisonoperators
236Comparison Operators
237comparison operators

+[238]#[239]$[240]K[241]**Within Expressions**

K[242]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 RasMol 0.004 Angstrom units. 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(800,backbone)`** selects any atom within a 3.2 Angstrom radius of any atom in a protein or nucleic acid backbone. **`Within`** expressions are particularly usefull for selecting the atoms around an active site.

238doc
239withinexpressions
240Within Expressions
241within expressions
242within

+[243]#[244]$[245]K[246]**Predefined Sets**

K[247]RasMol atom expressions may contain predefined sets. Thsese 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. A list of the currently predfined sets is given below.

at	acidic	acyclic	aliphatic
alpha	amino	aromatic	backbone
basic	buried	cg	charged
cyclic	cystine	helix	hetero
hydrogen	hydrophobic	large	medium
neutral	nucleic	polar	protein
purine	pyrimidine	selected	sheet
sidechain	small	surface	turn
water			

243doc
244predefinedsets
245Predefined Sets
246predefined sets
247sets

+[248]#[249]$[250]K[251]**AT Set**

K[252]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`**"

248doc
249atset
250AT Set
251at set
252at

+[253]#[254]$[255]K[256]**Acidic Set**

K[257]The set of acidic amino acids. These are the residue types Asp, Glu and Tyr. All amino acids are classified as either **`acidic,`** basic **`or`** neutral. This set is equivalent to the RasMol atom expressions "**`asp, glu, tyr`**" and "**`amino and not (basic or neutral)`**"

253doc
254acidicset
255Acidic Set
256acidic set
257acidic

+[258]#[259]$[260]K[261]**Acyclic Set**

K[262]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**"

258doc
259acyclicset
260Acyclic Set
261acyclic set
262acyclic

+[263]#[264]$[265]K[266]**Aliphatic Set**

K[267]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`**"

263doc
264aliphaticset
265Aliphatic Set
266aliphatic set
267aliphatic

+[268]#[269]$[270]K[271]**Alpha Set**

K[272]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.

268doc
269alphaset
270Alpha Set
271alpha set
272alpha

+[273]#[274]$[275]K[276]**Amino Set**

K[277]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.

273doc
274aminoset
275Amino Set
276amino set
277amino

+[278]#[279]$[280]K[281]**Aromatic Set**

K[282]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`**"

278doc
279aromaticset
280Aromatic Set
281aromatic set
282aromatic

+[283]#[284]$[285]K[286]**Backbone Set**

K[287]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

283doc
284backboneset
285Backbone Set
286backbone set
287backbone

+[288]#[289]$[290]K[291]**Basic Set**

K[292]The set of basic amino acids. These are the residue types Asp, Glu and Tyr. All amino acids are classified as either acidic, **basic** or neutral. This set is equivalent to the RasMol atom expressions "**asp, glu, tyr**" and "**amino and not (acidic or neutral)**"

288doc
289basicset
290Basic Set
291basic set
292basic

+[293]#[294]$[295]K[296]**Buried Set**

K[297]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`**"

293doc
294buriedset
295Buried Set
296buried set
297buried

+[298]#[299]$[300]K[301]**CG Set**

K[302]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**"

298doc
299cgset
300CG Set
301cg set
302cg

+[303]#[304]$[305]K[306]**Charged Set**

K[307]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**"

303doc
304chargedset
305Charged Set
306charged set
307charged

+[308]#[309]$[310]K[311]**Cyclic Set**

K[312]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`**"

308doc
309cyclicset
310Cyclic Set
311cyclic set
312cyclic

+[313]#[314]$[315]K[316]**Cystine Set**

K[317]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`**"

313doc
314cystineset
315Cystine Set
316cystine set
317cystine

+[318]#[319]$[320]K[321]**Helix Set**

K[322]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 command should not be confused with the predefined set alpha which contains the alpha carbon atoms of a protein.

318doc
319helixset
320Helix Set
321helix set
322helix

+[323]#[324]$[325]K[326]**Hetero Set**

K[327]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. The RasMol predefined set water is often used to partition this set.

323doc
324heteroset
325Hetero Set
326hetero set
327hetero

+[328]#[329]$[330]K[331]**Hydrogen Set**

K[332]This predefined set contains all the hydrogen and deuterium atoms of the current molecule.

328doc
329hydrogenset
330Hydrogen Set
331hydrogen set
332hydrogen

+[333]#[334]$[335]K[336]**Hydrophobic Set**

K[337]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`**"

333doc
334hydrophobicset
335Hydrophobic Set
336hydrophobic set
337hydrophobic

+[338]#[339]$[340]K[341]**Large Set**

K[342]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)`**"

338doc
339largeset
340Large Set
341large set
342large

+[343]#[344]$[345]K[346]**Medium Set**

K[347]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)`**"

343doc
344mediumset
345Medium Set
346medium set
347medium

+[348]#[349]$[350]K[351]**Neutral Set**

K[352]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)**"

348doc
349neutralset
350Neutral Set
351neutral set
352neutral

+[353]#[354]$[355]K[356]**Nucleic Set**

K[357]The set of all atoms in nucleic acids.

353doc
354nucleicset
355Nucleic Set
356nucleic set
357nucleic

+[358]#[359]$[360]K[361]**Polar Set**

K[362]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`**"

358doc
359polarset
360Polar Set
361polar set
362polar

+[363]#[364]$[365]K[366]**Protein Set**

K[367]The set of all atoms in proteins. This consists of the RasMol predefined set amino and common post-translation modifications.

363doc
364proteinset
365Protein Set
366protein set
367protein

+[368]#[369]$[370]K[371]**Purine Set**

K[372]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**"

368doc
369purineset
370Purine Set
371purine set
372purine

+[373]#[374]$[375]K[376]**Pyrimidine Set**

K[377]The set of pyrimidine nucleotides. These are the bases cytidine and thymidine (C and T respectively). All nucleotides are either **`purineset purines`** or **`pyrimidines.`** This set is equivalent to the RasMol atom expressions "**`c,t`**" and "**`nucleic and not pyrimidine`**"

373doc
374pyrimidineset
375Pyrimidine Set
376pyrimidine set
377pyrimidine

+[378]#[379]$[380]K[381]**Selected Set**

K[382]This set contains the set of atoms in the currently active zone. The currently active zone is defined by the preceding select or restrict command and not the atom expression containing the **`selected`** keyword.

378doc
379selectedset
380Selected Set
381selected set
382selected

+[383]#[384]$[385]K[386]**Sheet Set**

K[387]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.

383doc
384sheetset
385Sheet Set
386sheet set
387sheet

+[388]#[389]$[390]K[391]**Sidechain Set**

K[392]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

388doc
389sidechainset
390Sidechain Set
391sidechain set
392sidechain

+[393]#[394]$[395]K[396]**Small Set**

K[397]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)`**"

393doc
394smallset
395Small Set
396small set
397small

+[398]#[399]$[400]K[401]**Surface Set**

K[402]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 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 buried`**"

398doc
399surfaceset
400Surface Set
401surface set
402surface

+[403]#[404]$[405]K[406]**Turn Set**

K[407]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.

403doc
404turnset
405Turn Set
406turn set
407turn

+[408]#[409]$[410]K[411]**Water Set**

K[412]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.

408doc
409waterset
410Water Set
411water set
412water

+[413]#[414]$[415]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 amino, chain, group, shapely, structure, temperature, user and hbond type colour schemes. The currently predefined colour names are

413doc
414chcolours
415Colour Schemes

+[416]#[417]$[418]K[419]**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 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.

416doc
417aminocolours
418Amino Colours
419amino colours

+[420]#[421]$[422]K[423]**Chain Colours**

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

420doc
421chaincolours
422Chain Colours
423chain colours

+[424]#[425]$[426]K[427]**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 Kultun. This colour scheme colour `atom' objects by the atom (element) type. This is the scheme conventionally used by chemists.

424doc
425cpkcolours
426CPK Colours
427cpk colours

+[428]#[429]$[430]K[431]**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. Hence the N terminus of proteins and 5' terminus of nucleic acids are coloured red and the C terminus of proteins and 3' terminus of nucleic acids are drawn in blue. 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.

428doc
429groupcolours
430Group Colours
431group colours

+[432]#[433]$[434]K[435]**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.

+[436]#[437]$[438]K[439]**Structure Colours**

The RasMol **`structure`** colour scheme colours the molecule by protein secondary structure. Alpha helices are coloured magenta, beta sheets are coloured yellow, 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.

436doc
437structurecolours
438Structure Colours
439structure colours

+[440]#[441]$[442]K[443]**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.

440doc
441temperaturecolours
442Temperature Colours
443temperature colours

+[444]#[445]$[446]K[447]**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 introducted by David Bacon's Raster3D program.

444doc
445usercolours
446User Colours
447user colours

+[448]#[449]$[450]K[451]**HBond Type Colours**

The RasMol **type** colour scheme applies only to hydrogen bonds, hence is used in the command "**colour hbonds type**" This colour 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).

448doc
449hbondtypecolours
450HBond Type Colours
451hbond type colours