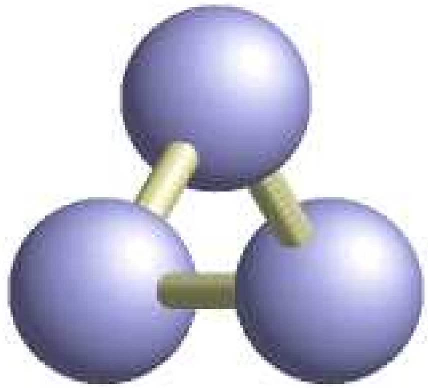
Manual

RasMol 2.7.3



Molecular Graphics Visualisation Tool
6 February 2005

Based on RasMol 2.6 by Roger Sayle
Biomolecular Structures Group
Glaxo Wellcome Research & Development
Stevenage, Hertfordshire, UK
Version 2.6, August 1995, Version 2.6.4, December 1998

and Based on Mods by

Author	Version, Date	Copyright
Arne Mueller	RasMol 2.6x1 May 1998	© Arne Mueller 1998
Gary Grossman and Marco Molinaro	RasMol 2.5-ucb November 1995 RasMol 2.6-ucb November 1996	© UC Regents/ModularCHEM Consortium 1995, 1996
Philippe Valadon	RasTop 1.3 August 2000	© Philippe Valadon 2000
Herbert J. Bernstein	RasMol 2.7.0 March 1999 RasMol 2.7.1 June 1999 RasMol 2.7.1.1 January 2001 RasMol 2.7.2 August 2000 RasMol 2.7.2.1 April 2001 RasMol 2.7.2.1.1 January 2004 RasMol 2.7.3 February 2005	© Herbert J. Bernstein 1998-2005

with RasMol 2.7.3 incorporating changes by Clarice Chigbo, Ricky Chachra, and Mamoru Yamanishi. Work on RasMol 2.7.3 supported in part bygrants DBI-0203064, DBI-0315281 and EF-0312612 from the U.S. National Science Foundation and grant DE-FG02-03ER63601 from the U.S. Department of Energy.

and Incorporating Translations by

Author	Item	Language
Isabel Serván Martínez, José Miguel Fernández Fernández	2.6 Manual	Spanish
José Miguel Fernández Fernández	2.7.1 Manual	Spanish

Fernando Gabriel Ranea	2.7.1 menus and messages	Spanish
Jean-Pierre Demailly	2.7.1 menus and messages	French
Giuseppe Martini, Giovanni Paolella, A. Davassi, M. Masullo, C. Liotto	2.7.1 menus and messages 2.7.1 help file	Italian

This Release by
Herbert J. Bernstein, Bernstein + Sons, P.O. Box 177, Bellport, NY, USA
yaya@bernstein-plus-sons.com

The original RasMol manual was created by Roger Sayle. In July 1996, Dr. Margaret Wong of the Chemistry Department, Swinburne University of Technology, Australia, made extensive revisions to the RasMol 2.5 manual to accurately reflect the operation of RasMol 2.6. Eric Martz of the University of Massachusetts made further revisions. In May 1997, William McClure of Carnegie Mellon University reorganized the HTML version of the manual into multiple sections which could be downloaded quickly and added use of frames. Portions of the 2.7.1 version of the RasMol manual were derived with permission from William McClure's version using Roger Sayle's rasmol.doc for version 2.6.4 as the primary source. Changes have been made in August 2000 for RasMol version 2.7.2, January 2001 for RasMol version 2.7.1.1, April 2001 for RasMol version 2.7.2.1 and February 2005 for RasMol version 2.7.3.

Documentation Last Updated 19 April 2005
Edited by Herbert J. Bernstein and Frances C. Bernstein

Translations

Thanks to the efforts of José Miguel Fernández Fernández (Departamento de Bioquímica y Biología Molecular. Universidad de Granada. España (jmfernan@ugr.es)) a translation of the Manual for Rasmol version 2.7.1 into Spanish is now available. La traducción española del manual de la versión de la Dra. Wong revisada por Eric Martz fue realizada por Isabel Serván Martínez y José Miguel Fernández Fernández. La actual traducción del Manual de RasMol 2.7.1 ha sido realizada usando como base la anterior de RasMol 2.6 por José Miguel Fernández Fernández.

Thanks to translations by Fernando Gabriel Ranea in late 2000 and early 2001, RasMol is now capable of rendering most menu items and messages in Spanish. Jean-Pierre Demailly provided French translations of menus and messages in January 2001. Giuseppe Martini and Giovanni Paolella with contributions by A. Davassi, M. Masullo and C. Liotto provided Italian translations of menus and messages in March 2001.

THIS IS A PRELIMINARY RELEASE INVOLVING EXTENSIVE MODIFICATIONS
***** USE WITH CAUTION ******

IMPORTANT

This version is based in directly on RasMol version 2.7.2.1.1, on RasMol version 2.7.2, on RasMol version 2.7.1, on RasMol version 2.6_CIF.2, on RasMol version 2.6x1, on RasMol version 2.6.4, and RasMol 2.5-ucb and 2.6-ucb.

Please read the file NOTICE for important notices which apply to this package and for license terms (GPL or RASLIC).

Table of Contents

rasmol@rasmol.org

Notices

This software has been created from several sources. Much of the code is from RasMol 2.6, as created by Roger Sayle.

See: http://www.dcs.ed.ac.uk/home/rasmol

The torsion angle code, new POVRAY3 code and other features are derived from the RasMol2.6x1 revisions by Arne Mueller.

See: ftp://nexus.roko.goe.net/pub/rasmol

The Ramachandran printer plot code was derived from fisipl created by Frances C. Bernstein. See the Protein Data Bank program tape.

The code to display multiple molecules and to allow bond rotation is derived in large part from the UCB mods by Gary Grossman and Marco Molinaro, included with permission of Eileen Lewis of the ModularCHEM Consortium.

See: http://mc2.CCHem.Berkeley.EDU/RasMol

The CIF modifications make use of a library based in part on CBFlib by Paul J. Ellis and Herbert J. Bernstein.

See: http://www.bernstein-plus-sons.com/software/CBF

Parts of CBFlib is loosely based on the CIFPARSE software package from the NDB at Rutgers university.

See http://www.iucr.org/iucr-top/cif/mmcif/ndb/software/CIFPARSE/

Please read the following notices below:

Copying RasMol
GPL
RASLIC
General Notice
RasMol 2.6 Notice
IUCr Policy
CBFlib V0.1 Notice
CIFPARSE notice

Copying RasMol

This version is based in large part on RasMol version 2.7.2.1.1, Rasmol version 2.7.2, RasMol version 2.7.1.1 and RasTop version 1.3 and indirectly on the RasMol 2.5-ucb and 2.6-ucb versions and version 2.6_CIF.2, RasMol 2.6x1 and RasMol_2.6.4.

RasMol 2.7.3 may be distributed under the terms of the GNU General Public License (the GPL), see

http://www.gnu.org/licenses/gpl.txt

or the file GPL or type the command '**help GPL**'

or RasMol 2.7.3 may be distributed under the RASMOL license. See the file NOTICE or type the command '**help RASLIC**'

GPL

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by

modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
    Gnomovision version 69, Copyright (C) year name of author
    Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
    This is free software, and you are welcome to redistribute it
    under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
  Yoyodyne, Inc., hereby disclaims all copyright interest in the program
  `Gnomovision' (which makes passes at compilers) written by James Hacker.

  <signature of Ty Coon>, 1 April 1989
  Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

RASLIC

If you do not use the GPL, the following license terms apply:

RasMol License

Even though the authors of the various documents and software found here have made a good faith effort to ensure that the documents are correct and that the software performs according to its documentation, and we would greatly appreciate hearing of any problems you may encounter, the programs and documents any files created by the programs are provided **AS IS** without any warranty as to correctness, merchantability or fitness for any particular or general use.

THE RESPONSIBILITY FOR ANY ADVERSE CONSEQUENCES FROM THE USE OF PROGRAMS OR DOCUMENTS OR ANY FILE OR FILES CREATED BY USE OF THE PROGRAMS OR DOCUMENTS LIES SOLELY WITH THE USERS OF THE PROGRAMS OR DOCUMENTS OR FILE OR FILES AND NOT WITH AUTHORS OF THE PROGRAMS OR DOCUMENTS.

Subject to your acceptance of the conditions stated above, and your respect for the terms and conditions stated in the notices below, if you are not going to make any modifications or create derived works, you are given permission to freely copy and distribute this package, provided you do the following:

1. Either include the complete documentation, especially the file NOTICE, with what you distribute or provide a clear indication where people can get a copy of the documentation; and

2. Please give credit where credit is due citing the version and original authors properly; and

3. Please do not give anyone the impression that the original authors are providing a warranty of any kind.

If you would like to use major pieces of RasMol in some other program, make modifications to RasMol, or in some other way make what a lawyer would call a "derived work", you are not only permitted to do so, you are encouraged to do so. In addition to the things we discussed above, please do the following:

4. Please explain in your documentation how what you did differs from this version of RasMol; and

5. Please make your modified source code available.

This version of RasMol is _not_ in the public domain, but it is given freely to the community in the hopes of advancing science. If you make changes, please make them in a responsible manner, and please offer us the opportunity to include those changes in future versions of RasMol.

General Notice

RasMol V2.6 Notice

The following notice applies to RasMol V 2.6 and older RasMol versions.

Information in this document is subject to change without notice and does not represent a commitment on the part of the supplier. This package is sold/distributed subject to the condition that it shall not, by way of trade or otherwise, be lent, re-sold, hired out or otherwise circulated without the supplier's prior consent, in any form of packaging or cover other than that in which it was produced. No part of this manual or accompanying software may be reproduced, stored in a retrieval system on optical or magnetic disk, tape or any other medium, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise for any purpose other than the purchaser's personal use.

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, bankruptcy or outbreak of war.

IUCR Policy

The IUCr Policy for the Protection and the Promotion of the STAR File and CIF Standards for Exchanging and Archiving Electronic Data.

Overview

The Crystallographic Information File (CIF)[1] is a standard for information interchange promulgated by the International Union of Crystallography (IUCr). CIF (Hall, Allen & Brown, 1991) is the recommended method for submitting publications to Acta Crystallographica Section C and reports of crystal structure determinations to other sections of Acta Crystallographica and many other journals. The syntax of a CIF is a subset of the more general STAR File[2] format. The CIF and STAR File approaches are used increasingly in the structural sciences for data exchange and archiving, and are having a significant influence on these activities in other fields.

Statement of intent

The IUCr's interest in the STAR File is as a general data interchange standard for science, and its interest in the CIF, a conformant derivative of the STAR File, is as a concise data exchange and archival standard for crystallography and structural science.

Protection of the standards

To protect the STAR File and the CIF as standards for interchanging and archiving electronic data, the IUCr, on behalf of the scientific community,

* holds the copyrights on the standards themselves,

* owns the associated trademarks and service marks, and

* holds a patent on the STAR File.

These intellectual property rights relate solely to the interchange formats, not to the data contained therein, nor to the software used in the generation, access or manipulation of the data.

Promotion of the standards

The sole requirement that the IUCr, in its protective role, imposes on software purporting to process STAR File or CIF data is that the following conditions be met prior to sale or distribution.

* Software claiming to read files written to either the STAR File or the CIF standard must be able to extract the pertinent data from a file conformant to the STAR File syntax, or the CIF syntax, respectively.

* Software claiming to write files in either the STAR File, or the CIF, standard must produce files that are conformant to the STAR File syntax, or the CIF syntax, respectively.

* Software claiming to read definitions from a specific data dictionary approved by the IUCr must be able to extract any pertinent definition which is conformant to the dictionary definition language (DDL)[3] associated with that dictionary.

The IUCr, through its Committee on CIF Standards, will assist any developer to verify that software meets these conformance conditions.

Glossary of terms

[1] CIF:

is a data file conformant to the file syntax defined at http://www.iucr.org/iucr-top/cif/spec/index.html

[2] STAR File:

is a data file conformant to the file syntax defined at http://www.iucr.org/iucr-top/cif/spec/star/index.html

[3] DDL:

is a language used in a data dictionary to define data items in terms of "attributes". Dictionaries currently approved by the IUCr, and the DDL versions used to construct these dictionaries, are listed at http://www.iucr.org/iucr-top/cif/spec/ddl/index.html

Last modified: 30 September 2000

CBFLIB

The following Disclaimer Notice applies to CBFlib V0.1, from which this code in part is derived.

CIFPARSE

Portions of this software are loosely based on the CIFPARSE software package from the NDB at Rutgers University. See

http://ndbserver.rutgers.edu/NDB/mmcif/software

CIFPARSE is part of the NDBQUERY application, a program component of the Nucleic Acid Database Project [H. M. Berman, W. K. Olson, D. L. Beveridge, J. K. Westbrook, A. Gelbin, T. Demeny, S. H. Shieh, A. R. Srinivasan, and B. Schneider. (1992). The Nucleic Acid Database: A Comprehensive Relational Database of Three-Dimensional Structures of Nucleic Acids. Biophys J., 63, 751-759.], whose cooperation is gratefully acknowledged, especially in the form of design concepts created by J. Westbrook.

Please be aware of the following notice in the CIFPARSE API:

This software is provided WITHOUT WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE OR ANY OTHER WARRANTY, EXPRESS OR IMPLIED. RUTGERS MAKE NO REPRESENTATION OR WARRANTY THAT THE SOFTWARE WILL NOT INFRINGE ANY PATENT, COPYRIGHT OR OTHER PROPRIETARY RIGHT.

Contents

Introduction

RasMol is a molecular graphics program intended for the visualisation 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 Microsoft Windows, Apple Macintosh, UNIX and VMS systems. UNIX and VMS versions require an 8, 24 or 32 bit colour X Windows display (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 a molecule coordinate file and interactively displays the molecule on the screen in a variety of colour schemes and molecule representations. Currently available representations include depth-cued wireframes, 'Dreiding' sticks, spacefilling (CPK) spheres, ball and stick, solid and strand biomolecular ribbons, atom labels and dot surfaces.

Up to 5 molecules may be loaded and displayed at once. Any one or all of the molecules may be rotated and translated.

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 coordinate files and interactively displays the molecule on the screen in a variety of representations and colour schemes. Supported input file formats include Protein Data Bank (PDB), Tripos Associates' Alchemy and Sybyl Mol2 formats, Molecular Design Limited's (MDL) Mol file format, Minnesota Supercomputer Center's (MSC) XYZ (XMol) format, CHARMm format, CIF format and mmCIF 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. Atoms may also be labelled with arbitrary text strings. Alternate conformers and multiple NMR models may be specially coloured and identified in atom labels. 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 Kinemage.

The RasMol help facility can be accessed by typing "help <topic>" or "help <topic> <subtopic>" from the command line. A complete list of RasMol commands may be displayed by typing "help commands". A single question mark may also be used to abbreviate the keyword "help". Please type "help notices" for important notices.

General Operation

- [RasMol's Window]
- [Mouse Controls]
- [Scroll Bars]
- [Picking]
- [Dials Box]
- [Command Line Interface]
- [Dimensions within RasMol]
- [Start-up Initialisation Files]
- [Inter-Process Communication (IPC)]

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. There will be some variation in this message depending on your choice of platform:

```
RasMol Molecular Renderer
Roger Sayle, August 1995
Copyright (C) Roger Sayle 1992-1999
Version 2.7.3 February 2005
Copyright (C) Herbert J. Bernstein 1998-2005
*** See "help notice" for further notices ***
[32-bit 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 be instructed not to display a graphics window by using the command line option '-nodisplay'. This is particularly useful for running 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. A script file may be specified by adding the option '-script <filename>' to the command line. A molecule coordinate 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 coordinate file is assumed to be in PDB, CIF or mmCIF format. Valid format options include '-pdb', '-mdl', '-mol2', '-xyz', '- alchemy', '-charmm', '-mopac' and '-cif' which correspond to Protein Data Bank format, Molecular Design Limited's Mol file format, Tripos's Sybyl Mol2 file format, MSC's XMOL XYZ file format, Tripos's Alchemy file format, CHARMm file format, J. P. Stewart's MOPAC file format and IUCr CIF or mmCIF file format, respectively. If both a coordinate 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 or exit 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 a script filename or both on the windows command line. A script file may be specified by adding the option '-script <filename>' to the command line. A molecule coordinate 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 coordinate file is assumed to be in PDB, CIF or mmCIF format. Valid format options include '-pdb', '-mdl', '-mol2', '-xyz', '- alchemy', '-charmm', '-mopac' and '-cif' which correspond to Protein Data Bank format, Molecular Design Limited's Mol file format, Tripos's Sybyl Mol2 file format, MSC's XMOL XYZ file format, Tripos's Alchemy file format, CHARMm file format, J. P. Stewart's MOPAC file format and IUCr CIF or mmCIF file format, respectively. If both a coordinate 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 (principally 'TEXT') are assumed to be in PDB format. Unlike other versions of RasMol it is impossible to specify both a script and a coordinate 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.

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', 'Options' and 'Settings' 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.

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.

The normal behavior of the scroll bars can be changed by the '`rotate bond`' and '`rotate all`' commands and restored to normal operation by the '`rotate molecule`' command. Alternatively the equivalent items in the "Settings" menu may be used. When '`rotate bond`' is selected, the bottom scroll bar controls. rotation around a bond selected by the '**bond <src> <dst> pick**' command (or by use of the "Pick Bond" item in the "Settings" menu). When '**rotate all**' is selected, the scroll bars control rotation of all the loaded molecules instead of just rotating the currently selected molecule.

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 buttons 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 Protein Data Bank 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 heterogeneous 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 the **coordinates**, the **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, to specify the **centre of rotation**, or to specify a **bond as the axis 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 allows the execution of interactive commands typed at the RasMol prompt in the terminal window. Characters typed into either the terminal or the display window are processed on the command line. Each command must be given on a separate line terminated by a newline or carriage return character. Keywords are case insensitive and may be entered in both lower and upper case letters. All whitespace (space, tab and formfeed) characters are ignored, except to separate the keyword and the arguments of a command. Blank lines (those containing only whitespace) are ignored. There is an internal restriction that command lines are limited to a maximum of 256 characters. Strings may be delimited by matching single or double quotation marks. Placing a hash '#' character anywhere outside quotes terminates the line. RasMol will ignore the rest of the line, which may be used to comment on the command.

If a syntax error is detected on entering an interactive command, RasMol indicates the location of the error on the command line by placing the '^' character under the offending word or character, and writing an error message on the following line. If a command is not recognised by RasMol, the program will generate an 'Unrecognised command!' error and redisplay the main prompt. If surplus information is given at the end of a command line, RasMol will execute the recognised command, but issue the warning message 'Warning: Ignoring rest of command!'. Some commands may prompt the user for more information. These commands display a different prompt and are discussed in the command reference.

Whenever RasMol outputs diagnostic or error messages to the screen owing to selecting options from the menu or picking objects on the screen, the current command line is cleared. The prompt is redisplayed after any text has been displayed.

Command Line Editing

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.

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.

Users with the Macintosh version can use the four 'arrow keys' to move up and down through previous command line entries; and back and forth within a single command line statement. Hitting 'return' or 'enter' at any time will result in the execution of the current, *e.g.* selected or edited, command line contents.

Dimensions within RasMol

All dimensions in RasMol, such as radii and distances, may be specified in either 'RasMol units' or Ångstroms (Å). 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 Ångstrom; therefore the most frequently used values are in the hundreds. For this reason, if RasMol is given a distance parameter that does not 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 Å.

However, dimensions within RasMol can also be specified in Ångstroms by placing a decimal point in the number. For example, 'spacefill 1.2' specifies a sphere radius of 1.2 Å. 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 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. Such system rasmolrc files will contain RasMol '**`echo`**' commands detailing a telephone number or e-mail address to be used for contacting somebody for local assistance.

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 'rasmol #2', 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 application 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 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	Background	Bond	Cartoon	Centre	Clipboard	Colour	Connect
CPK	CPKnew	Define	Depth	Dots	Echo	English	Exit
French	HBonds	Help	Italian	Label	Load	Molecule	Monitor
Pause	Print	Quit	Refresh	Renumber	Reset	Restrict	Ribbons
Rotate	Save	Script	Select	Set	Show	Slab	Source
Spacefill	Spanish	SSBonds	Star	Stereo	Strands	Structure	Surface
Trace	Translate	UnBond	Wireframe	Write	Zap	Zoom	

Backbone

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

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 parameter in the same way as with 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 Ångstrom or RasMol units. A parameter value of 500 (2.0 Ångstroms) or above results in a "Parameter value too large" error. Backbone objects may be coloured using the RasMol '**colour backbone**' command.

The reserved word backbone is also used as a predefined set ("help sets") 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.

The backbone may be displayed with dashed lines by use of the '**backbone dash**' command.

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.

Bond

```
Syntax:  bond <number>  <number> +
         bond <number>  <number> pick
         bond rotate {<boolean>}
```

The RasMol command '**bond <number> <number> +**' adds the designated bond to the drawing, increasing the bond order if the bond already exists. The command '**bond <number> <number> pick**' selects the two atoms specified by the atom serial numbers as the two ends of a bond around which the '**rotate bond <angle>**' command will be applied. If no bond exists, it is created.

Rotation around a previously picked bond may be specified by the '**rotate bond <angle>**' command, or may also be controlled with the mouse, using the '**bond rotate on/off**' or the equivalent '**rotate bond on/off**' commands.

Cartoon

```
Syntax:  cartoon {<number>}
```

The RasMol '**cartoon**' command does a display of a molecule '**ribbons**' as Richardson (MolScript) style protein '**cartoons**', implemented as thick (deep) ribbons. The easiest way to obtain a cartoon representation of a protein is to use the '**Cartoons**' option on the '**Display**' menu. The '**cartoon**' command represents the currently selected residues as a deep ribbon with width specified by the command's argument. Using the command without a parameter results in the ribbon's width being taken from the protein's secondary structure, as described in the '**ribbons**' command. By default, the C-termini 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 <number>**' command. The '**set cartoons**' command without any parameters returns these two options to their default values.

Centre

```
Syntax:  centre {<expression>} {translate|center}
         center {<expression>} {translate|center}
```

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.

Alternatively the centring may be given as a comma separated triple of [CenX, CenY, CenZ] offsets in RasMol units (1/250 of an Ångstrom) from the centre of gravity. The triple must be enclosed in square brackets.

The optional forms '**centre ... translate**' and '**centre ... center**' may be used to specify use of a translated centre of rotation (not necessarily in the centre of the canvas) or a centre of rotation which is placed at the centre of the canvas. Starting with RasMol 2.7.2, the default is to center the new axis on the canvas.

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>
```

Colour the atoms (or other objects) of the selected region. 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**', '**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). '**Atom**' objects can also be coloured by '**alt**', '**amino**', '**chain**', '**charge**', '**cpk**', '**group**', '**model**', '**shapely**', '**structure**', '**temperature**' or '**user**'. Hydrogen bonds can also be coloured by '**type**' and dot surfaces can also be coloured by '**electrostatic potential**'. For more information type '**help colour <colour>**'.

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 a fast heuristic algorithm that is suitable for determining 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 to 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 input 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.

Depth

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

The RasMol '**depth**' command enables, disables or positions the back-clipping plane of the molecule. The program only draws those portions of the molecule that are closer to the viewer than the clipping 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.

This command interacts with the '**slab <value>**' command, which clips to the front of a given z-clipping plane.

Dots

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

The RasMol '**dots**' command is used to generate a van der Waals' 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 Waals' 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 its 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.

English

```
Syntax:  English
```

The RasMol '**English**' command sets the menus and messages to the English versions. The commands '**French**', '**Italian**' and '**Spanish**' may be used to select French, Italian and Spanish menus and messages.

French

```
Syntax:  French
```

The RasMol '**French**' command sets the menus and messages to the French versions. The commands '**English**', '**Italian**' and '**Spanish**' may be used to select English, Italian and Spanish menus and messages.

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.

Italian

```
Syntax:  Italian
```

The RasMol '**Italian**' command sets the menus and messages to the Italian versions. The commands '**English**', '**French**' and '**Spanish**' may be used to select English, French and Spanish menus and messages.

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 and spacing of the displayed text may be changed using the '**set fontsize**' command. The width of the strokes in the displayed text may be changed using the '**set fontstroke**' 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
%n      Residue Name
%r      Residue Number
%M      NMR Model Number (with leading "/")
%A      Alternate Conformation Identifier (with leading ";")
```

Load

```
Syntax:  load {<format>} <filename>
```

Load a molecule coordinate file into RasMol. Valid molecule file formats are '**`pdb`**' (Protein Data Bank format), '**`mdl`**' (Molecular Design Limited's MOL file format), '**`alchemy`**' (Tripos' Alchemy file format), '**`mol2`**' (Tripos' Sybyl Mol2 file format), '**`charmm`**' (CHARMm file format), '**`xyz`**' (MSC's XMol XYZ file format), '**`mopac`**' (J. P. Stewart's MOPAC file format) or '**`cif`**' (IUCr CIF or mmCIF file format). If no file format is specified, '**`PDB`**', '**`CIF`**', or '**`mmCIF`**' is assumed by default. Up to 5 molecules may be loaded at a time. To delete a molecule prior to loading another use the RasMol '**`zap`**' command. To select a molecule for manipulation use the RasMol '**`molecule <n>`**' 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 fewer bonds than atoms, RasMol determines connectivity using the '**`connect`**' command.

The '**`load inline`**' command also allows the storing of atom coordinates in scripts to allow better integration with WWW browsers. A load command executed inside a script file may specify the keyword '**`inline`**' instead of a conventional filename. This option specifies that the coordinates 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 coordinates in PDB, CIF or mmCIF file format. One possible use is a standard RasMol script prefix that may be concatenated with an appropriate PDB file on-the-fly.

Molecule

```
Syntax:  molecule <number>
```

The RasMol '**`molecule`**' command selects one of up to 5 previously loaded molecules for active manipulation. While all the molcules are displayed and may be rotated collectively (see the '**`rotate all`**' command), only one molecule at a time time is active for manipulation by the commands which control the details of rendering.

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. The RasMol command '**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 its 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 pushed 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 usual, 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, this allows 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.

Quit

```
Syntax:  quit
         exit
```

Exit from the RasMol program. The RasMol commands '**exit**' and '**quit**' are synonymous, except within nested scripts. In that case, '**exit**' terminates only the current level, while '**quit**' terminates all nested levels of scripts.

Refresh

```
Syntax:  refresh
```

The RasMol '**refresh**' command redraws the current image. This is useful in scripts to ensure application of a complex list of parameter changes.

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 the 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 its 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 affect 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.

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 Ångstroms) for nucleic acids. The default width of protein alpha helices and beta sheets is 380 (1.52 Ångstroms) and 100 (0.4 Ångstroms) 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 bond {<boolean>}
         rotate molecule {<boolean>}
         rotate all {<boolean>}
```

Rotate the molecule about the specified axis. Permitted 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.

Alternatively, this command may be used to specify which rotations the mouse or dials will control. If '**rotate bond true**' is selected, the horizontal scroll bar will control rotation around the axis selected by the '**bond src dst pick**' command. If '**rotate all true**' is selected, and multiple molecules have been loaded, then all molecules will rotate together. In all other cases, the mouseand dials control the the rotation of the molecule selected by the '**molecule n**' command.

Save

```
Syntax:  save {pdb} <filename>
         save mdl <filename>
         save alchemy <filename>
         save xyz <filename>
```

Save the currently selected set of atoms in a Protein Data Bank (PDB), MDL, Alchemy(tm) or XYZ 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 complicated 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.

The most common way to generate a RasMol script file is to use 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.

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

Scripts may also be created with a text editor.

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 affect 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.

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, omitting the paramter 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
CisAngle	Display	FontSize	FontStroke	HBonds	Hetero	HourGlass	Hydrogen
Kinemage	Menus	Monitor	Mouse	Picking	Radius	ShadePower	Shadow
SlabMode	Solvent	Specular	SpecPower	Stereo	SSBonds	Strands	Transparent
UnitCell	VectPS	Write					

Show

```
Syntax:  show information
         show centre
         show phipsi
         show RamPrint
         show rotation
         show selected { group | chain | atom }
         show sequence
         show symmetry
         show translation
         show zoom
```

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 centre**' shows any non-zero centering values selected by the '**centre [CenX, CenY, CenZ]**' command. The command '**show phipsi**' shows the phi and psi angles of the currently selected residues and the omega angles of cis peptide bonds. The command '**show RamPrint**' (or 'show RPP' or 'show RamachandranPrinterPlot') shows a simple Ramachandran printer plot in the style of Frances Bernstein's fisipl program. The command '**show rotation**' (or 'show rot' or 'show 'rotate') shows the currently selected values of z, y, x and bond rotations, if any. The command '**show selected**' (or 'show selected group' or 'show selected chain' or 'show selected atom') shows the groups (default), chains or atoms of the current selection. The command '**show sequence**' lists the residues that comprise each chain of the molecule. The command '**show symmetry**' shows the space group and unit cell of the molecule. The command '**show translation**' shows any non-zero translation values selected by the '**translate <axis> <value>**' command. The command '**show zoom**' shows any non-zero zoom value selected by the '**zoom <value>**' command.

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.

This command interacts with the '**depth <value>**' command, which clips to the rear of a given z-clipping plane.

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, '**spacefill 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 Ångstrom) or a value containing a decimal point. A value of 500 (2.0 Ångstroms) 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 have no effect and values greater than 2.0 are truncated to 2.0. The '**user**' option allows the radius of each sphere to be specified by additional lines in the molecule's PDB file using Raster 3D's COLOUR record extension.

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

The RasMol command '**cpknew**' is synonymous with the '**spacefill**' command, except that a slightly different set of colours is used.

Spanish

```
Syntax:  Spanish
```

The RasMol '**Spanish**' command sets the menus and messages to the Spanish versions. The commands '**English**', '**French**' and '**Italian**' may be used to select English, French and Italian menus and messages.

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 Ångstroms 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.

Star

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

The RasMol '**star**' command is used to represent all of the currently selected atoms as stars (six strokes, one each in the x, -x, y, -y, z and -z directions). The commands '**select not bonded**' followed by '**star 75**' are useful to mark unbonded atoms in a '**wireframe**' display with less overhead than provided by '**spacefill 75**'. This can be done automatically for all subsequent wireframe displays with the command '**set bondmode not bonded**'.

The command '**star true**', the default, represents each atom as a star with strokes length equal to van der Waals radius. The command '**star off**' turns off the representation of the selected atom as stars. A star stroke length may be specified as an integer in RasMol units (1/250th Ångstrom) or a value containing a decimal point. A value of 500 (2.0 Ångstroms) or greater results in a "Parameter value too large" error.

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

The RasMol '**spacefill**' command can be used for more artistic rendering of atoms as spheres.

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**'.

Starting with RasMol version 2.7.2.1, the '**Stereo**' menu selection and the command '**stereo**' without arguments cycle from the initial state of '**stereo off**' to '**stereo on**' in cross-eyed mode to '**stereo on**' in wall-eyed mode and then back to '**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.

The stereo command is only partially implemented. When stereo is turned on, the image is not properly recentred. (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.

Strands

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

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 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 Ångstroms wide). The default width of protein alpha helices and beta sheets is 380 (1.52 Ångstroms) and 100 (0.4 Ångstroms) 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.

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, SHEET and TURN) these are discarded. Initially, the hydrogen bonds of the current molecule are found, if this hasn't been done already. The secondary structure is then determined using Kabsch and Sander's DSSP algorithm. Once finished the program reports the number of helices, strands and turns found.

Surface

```
Syntax:  surface molecule <value>
         surface solvent  <value>
```

The RasMol '**surface**' command renders a Lee-Richards molecular surface resulting from rolling a probe atom on the selected atoms. The value given specifies the radius of the probe. If given in the first form, the evolute of the surface of the probe is shown (the solvent excluded surface). If given in the second form, the envelope of the positions of the center of the probe is shown (the solvent accessible surface).

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 residues may be displayed as '**ribbons**', '**strands**', '**cartoons**' or as a '**trace**'. Enabling one of these representations disables the others. However, a residue may be displayed simultaneously as backbone and as one of the above representations. 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. Permitted 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.

UnBond

```
Syntax:  unbond <number>  <number>
         unbond
```

The RasMol command '**unbond <number> <number>**' removes the designated bond from the drawing.

The command '**unbond**' without arguments removes a bond previously picked by the '**bond <number> <number> pick**' command.

Wireframe

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

The RasMol '**wireframe**' command represents each bond within the selected region of the molecule as a cylinder, a line or a 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 Ångstroms. A parameter value of 500 (2.0 Ångstroms) or above results in an "Parameter value too large" error. Bonds may be coloured using the '**colour bonds**' command. If the selected bonds involved atoms of alternate conformers then the bonds are narrowed in the middle to a radius of .8 of the specified radius (or to the radius specifed as the optional second parameter).

Non-bonded atoms, which could become invisible in an ordinary '**wireframe**' display can be marked by a preceding '**set bondmode not bonded**' command. If nearly co-linear bonds to atoms cause them to be difficult to see in a wireframe display, the '**set bondmode all**' command will add markers for '**all**' atoms in subsequent '**wireframe**' command executions.

Write

```
Syntax:  write {<format>} <filename>
```

Write the current image to a file in a standard format. Currently supported image file formats include '**bmp**' (Microsoft bitmap) and '**gif**' (Compuserve GIF), '**iris**' (IRIS RGB), '**ppm**' (Portable Pixmap), '**ras**' (Sun rasterfile), '**ps**' and '**epsf**' (Encapsulated PostScript), '**monops**' (Monochrome Encapsulated PostScript), '**pict**' (Apple PICT), '**vectps**' (Vector Postscript). 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 following formats are useful for further processing: '**povray**' (POVRay 2), '**povray3**' (POVRay 3 -- under development), '**vrml**' (VRML file). Finally, several formats are provided to provide phi-psi data for listing or for '**phipsi**' (phi-psi data as an annotated list with cis omegas), '**ramachan**' and '**RDF**' and '**RamachandranDataFile**' (phi-psi data as columns of numbers for gnuplot), '**RPP**' and '**RamachandranPrinterPlot**' (phi-psi data as a printer plot).

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.

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 below.

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

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. It 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>
```

The RasMol '**axes**' parameter controls the display of orthogonal coordinate axes on the current display. The coordinate 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 to the commands '**set boundbox**' and '**set unitcell**' that display the bounding box and the crystallographic unit cell, respectively.

Set Backfade

```
Syntax:  set backfade <boolean>
```

The RasMol '**backfade**' parameter is used to control backfade to the specified background colour, rather than black. This is controlled by the commands '**set backfade on**' and '**set backfade off**'. For example, this may be used to generate depth-cued 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**'.

Set BondMode

```
Syntax:  set bondmode and
         set bondmode or
         set bondmode all
         set bondmode none
         set bondmode not bonded
```

The RasMol '**set bondmode**' command controls the mechanism used to select individual bonds and modifies the display of bonded and non-bonded atoms by subsequent '**wireframe**' commands.

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.

The '**bondmode [all | none | not bonded]**' commands add '**star 75**' or '**spacefill 75**' markers for the designated atoms to '**wireframe**' displays. Stars are used when the specified wireframe radius is zero.

Set Bonds

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

The RasMol '**bonds**' parameter is used to control 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, CIF and mmCIF, and suitable PDB files. Double (and triple) bonds are specified in some PDB files by specifying a given bond twice (and three times) in CONECT records. The command '**set bonds on**' enables the display of bond orders, and the command '**set bonds off**' disables them.

Set BoundBox

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

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

Set Cartoon

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

The RasMol '**cartoon**' parameter is used to control display of the cartoon version of the '**ribbons**' display. By default, the C-termini of beta-sheets are displayed as arrow heads. This may be enabled and disabled using the '**set cartoons <boolean>**' command. The depth of the cartoon may be adjusted using the '**cartoons <number>**' command. The '**set cartoons**' command without any parameters returns these two options to their default values.

Set CisAngle

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

The RasMol '**cisangle**' parameter controls the cutoff angle for identifying cis peptide bonds. If no value is given, the cutoff is set to 90 degrees.

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 FontSize

```
Syntax:   set fontsize {<value>} { FS | PS }
```

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 48 pixels, and the default value is 8 pixels high. Fixed or proportional spacing may be selected by appending the "FS" or "PS" modifiers, respectively. The default is "FS". 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 FontStroke

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

The RasMol '**set fontstroke**' command is used to control the size of the stroke width of the characters that form atom labels. This value is the radius in pixels of cylinders used to form the strokes. The special value of "0" is the default used for the normal single pixel stroke width, which allows for rapid drawing and rotation of the image. Non-zero values are provided to allow for more artistic atom labels for publication at the expense of extra time in rendering the image.

When wider strokes are used, a larger font size is recommend, e.g. by using the RasMol '**set fontsize 24 PS**' command, followed by '**set fontstroke 2**'

The character sets used by RasMol rendered with fixed spacing with single-pixel-width strokes and with proportional spacing with 2-pixel-radius cylinder strokes are shown in the following sample:

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 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`**'. This parameter is similar to the RasMol '**`ssbonds`**' parameter.

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 heterogeneous 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`**'.

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`**' enables 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, deuterium or tritium 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 heterogeneous 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 &lt;boolean&gt;
```

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 coordinate 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 an image as possible when using Microsoft Windows.

Set Monitor

```
Syntax:  set monitor <boolean>
```

The RasMol '**set monitor**' command enables '**monitors**'. The distance monitor labels may be turned off with the command '**set monitor off**', and re-enabled with the command '**set monitor 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**' options provide the same mouse bindings as other packages for experienced users.

Set Picking

```
Syntax:  set picking <boolean>
         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
set picking center
set picking coord
set picking bond
set picking atom
set picking group
set picking chain
```

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

Enabling/Disabling Atom Identification Picking: Clicking on an atom with the mouse results in identification and the display of its residue name, residue number, atom name, atom serial number and chain in the command window. This behavior may be disabled with the command '**set picking none**' and restored with the command '**set picking ident**'. The command '**set picking coord**' adds the atomic coordinates of the atom to the display.

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 while 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 is not 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.

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

Picking a Bond as a Rotation Axis: Any bond may be picked as an axis of rotation for the portion of the molecule beyond the second atom selected. This feature should be used with caution, since, naturally, it changes the conformation of the molecule. After executing '**set picking bond**' or using the equivalent "Pick Bond" in the "Settings" menu, a bond to be rotated is picked with the same sort of mouse clicks as are used for picking atoms for a distance measurement. Normally this should be done where a bond exists, but if no bond exists, it will be added. The bond cannot be used for rotation if it is part of a ring of any size. All bonds selected for rotation are remembered so that they can be properly reported when writing a script, but only the most recently selected bond may be actively rotated.

`Enabling Atom/Group/Chain Selection Picking`: Atoms, groups and chains may be selected (as if with the '**`select`**' command), with the '**`set picking atom`**', '**`set picking group`**', '**`set picking chain`**' commands. For each of these commands, the shift key may be used to have a new selection added to the old, and the control key may be used to have a new selection deleted from the old. When the '**`set picking atom`**' command is given, the mouse can be used to pick or to drag a box around the atoms for which selection is desired. When the '**`set picking group`**' command is given, picking any an atom will cause selection of all atoms which agree in residue number with the picked atom, even if in different chains. When the '**`set picking chain`**' command is given, picking any atom will cause selection of all atoms which agree in chain identifier with the picked atom.

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 Waals' 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 its 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 Ångstroms. The default value of this parameter is determined by the value of '**`solvent`**' and changing '**`solvent`**' resets '**`radius`**' to its new default value.

Set ShadePower

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

The '**`shadepower`**' parameter (adopted from RasTop) determines the shade repartition (the contrast) used in rendering solid objects. This value between 0 and 100 adjusts shading on an object surface oriented along the direction to the light source. Changing the shadepower parameter does not change the maximum or the minimum values of this shading, as does changing the '**`ambient`**' parameter. A value of 100 concentrates the light on the top of spheres, giving a highly specular, glassy rendering (see the '**`specpower`**' parameter). A value of 0 distributes the light on the entire object.

This implementation of shadepower differs from the one in RasTop only in the choice of range (0 to 100 versus -20 to 20 in RasTop).

Set Shadow

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

The RasMol '**`set shadow`**' command enables and disables ray-tracing 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`**'. Ray-tracing typically takes about several seconds for a moderately sized protein. It is recommended that shadowing be 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 objects cut by the slabbing (z-clipping) plane. Valid slabmode parameters are "**`reject`**", "**`half`**", "**`hollow`**", "**`solid`**" and "**`section`**".

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 Waals' or a solvent accessible 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 Waals' 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 Ångstroms (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 coefficient 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**'. This parameter is similar to the RasMol '**hbonds**' parameter.

Set Stereo

```
Syntax:   set stereo <boolean>
          set stereo [-] <number>
```

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

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. 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 <boolean>
```

The RasMol '**transparent**' parameter controls the writing of transparent GIFs by the '**write gif <filename>**' command. 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, CIF or mmCIF 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 to the commands '**set axes**' and '**set boundbox**' that display orthogonal coordinate 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 the 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 but 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 '**write**' parameter controls the use of the '**save**' and '**write**' commands within scripts, but it 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). 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 backbone 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
```

- Primitive Expressions
- Predefined Sets
- Comparison Operators
- Within Expressions
- Examples

Example Expressions

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

Expression	**Interpretation**
*	All atoms
cys	Atoms in cysteines
hoh	Atoms in heterogeneous 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
.;A	All atoms in alternate conformation A
*/4	All atoms in model 4

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 intended to be the residue's position in the macromolecule sequence, but negative sequence numbers, gaps in numbering, or even reverse numbering are permitted in the PDB format. Care must be taken when specifying both residue name and number. If the group at the specified position isn't the specified residue then 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`**" for the alphabetic chain identifier, "A", or "**`SER70:1`**" for the numeric chain identifier, "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 optional semicolon followed by an alternate conformation identifier may be appended. An optional slash followed by a model number may also be appended.

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 isotropic 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 parentheses. 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 Ångstroms 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 Ångstrom 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 of primitive atom expressions. In some cases the use of predefined sets allows selection of areas of a molecule that could not otherwise be distinguished. A list of the currently predefined sets is given below. 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**'.

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 equivalent 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 heterogeneous 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 of 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 at least one other atom.

Buried Set

This set contains the atoms in those amino acids that tend (prefer) to be buried inside protein, away from 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 surface`**".

CG Set

This set contains the atoms in the complementary nucleotides cytidine and guanosine (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 heterogeneous 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 heterogeneous '**solvent**' atoms are further classified as either '**water**' or '**ions**'.

Hydrogen Set

This predefined set contains all the hydrogen, deuterium and tritium 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 heterogeneous 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 heterogeneous cofactor and ligand moieties that are contained in the current molecule data file. 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)`**".

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**". The symbols for RNA nucleotides (U, +U, I, 1MA, 5MC, OMC, 1MG, 2MG, M2G, 7MG, OMG, YG, H2U, 5MU, and PSU) are also recognized as members of this set.

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 pyrimidine**".

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 purine**".

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)`**".

Solvent Set

This set contains the solvent atoms in the molecule coordinate file. These are the heterogeneous 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`**".

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 heterogeneous 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`**'.

Set Summary

The table below summarises RasMol's classification of the common amino acids.

Residues:	**ala**	**arg**	**asn**	**asp**	**cys**	**glu**	**gln**	**gly**	**his**	**ile**	**leu**	**lys**	**met**	**phe**	**pro**	**ser**	**thr**	**trp**	**tyr**	**val**
	A	**R**	**N**	**D**	**C**	**E**	**Q**	**G**	**H**	**I**	**L**	**K**	**M**	**F**	**P**	**S**	**T**	**W**	**Y**	**V**
Predefined Set																				
	A	R	N	D	C	E	Q	G	H	I	L	K	M	F	P	S	T	W	Y	V
acidic				*		*														
acyclic	*	*	*	*	*	*	*	*		*	*	*	*			*	*			*
aliphatic	*							*		*	*									*
aromatic									*					*				*	*	
basic		*							*			*								
buried	*				*					*	*		*	*				*		*
charged		*		*		*			*			*								
cyclic									*					*	*			*	*	
hydrophobic	*							*		*	*		*	*	*			*	*	*
large		*				*	*		*	*	*	*	*	*				*	*	
medium			*	*	*										*		*			*
negative				*		*														
neutral	*		*		*		*	*	*	*	*		*	*	*	*	*	*	*	*
polar		*	*	*	*	*	*		*			*				*	*			
positive		*							*			*								
small	*							*								*				
surface		*	*	*		*	*	*	*			*			*	*	*		*	

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 '**alt**', '**amino**', '**chain**', '**charge**', '**cpk**', '**group**', '**model**', '**shapely**', '**structure**', '**temperature**' or '**user**' colour schemes for atoms, and '**hbond type**' colour scheme for hydrogen bonds and '**electrostatic potential**' colour scheme for dot surfaces. The 24 currently predefined colour names are listed below with their corresponding RGB triplet and hexadecimal value.

Predefined colour	Sample	RGB Values	Hexadecimal
Black		[0, 0, 0]	000000
Blue		[0, 0,255]	0000FF
BlueTint		[175,214,255]	AFD7FF
Brown		[175,117,89]	AF7559
Cyan		[0,255,255]	00FFFF
Gold		[255,156, 0]	FC9C00
Grey		[125,125,125]	7D7D7D
Green		[0,255, 0]	00FF00
GreenBlue		[46,139,87]	2E8B57
GreenTint		[152,255,179]	98FFB3
HotPink		[255, 0,101]	FF0065
Magenta		[0,255,0]	FF00FF
Orange		[255,165, 0]	FFA500
Pink		[255,101,117]	FF6575
PinkTint		[255,171,187]	FFABBB
Purple		[160, 32,240]	A020F0
Red		[255, 0, 0]	FF0000
RedOrange		[255, 69, 0]	FF4500

SeaGreen		[0,250,109]	00FA6D
SkyBlue		[58,144,255]	3A90FF
Violet		[238,130,238]	EE82EE
White		[255,255,255]	FFFFFF
Yellow		[255,255, 0]	FFFF00
YellowTint		[246,246,117]	F6F675

Note that the rendering of the hexadecimal-equivalent colours shown here will depend on many factors. Thus, they only approximate how RasMol will render the RGB colours on your computer.

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.col**' containing the line, '**colour [180,180,180] #grey**', then the command '**script grey.col**' colours the currently selected atom set grey.

Alt Colours

The RasMol '**alt**' (Alternate Conformer) colour scheme codes the base structure with one colour and applies a limited number of colours to each alternate conformer. In a RasMol built for 8-bit colour systems, 4 colours are allowed for alternate conformers. Otherwise, 8 colours are available.

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.

Amino Acids	colour Name	Sample	RGB Values	Hexadecimal
ASP, GLU	**Bright Red**		[230,230, 10]	E60A0A
CYS, MET	**Yellow**		[230,230, 0]	E6E600
LYS, ARG	**Blue**		[20, 90,255]	145AFF
SER, THR	**Orange**		[250,150, 0]	FA9600
PHE, TYR	**Mid Blue**		[50, 50,170]	3232AA

ASN, GLN	**Cyan**		[0,220,220]	00DCDC
GLY	**Light Grey**		[235,235,235]	EBEBEB
LEU, VAL, ILE	**Green**		[15,130, 15]	0F820F
ALA	**Dark Grey**		[200,200,200]	C8C8C8
TRP	**Purple**		[180, 90,180]	B45AB4
HIS	**Pale Blue**		[130,130,210]	8282D2
PRO	**Flesh**		[220,150,130]	DC9682
Others	**Tan**		[190,160,110]	BEA06E

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. '**Chain**' can be selected from the RasMol '**Colours**' menu.

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 values 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.

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 colours 'atom' objects by the atom (element) type. This is the scheme conventionally used by chemists. The assignment of the most commonly used element types to colours is given below.

Element	Colour Name	Sample	RGB Values	Hexadecimal
Carbon	light grey		[200,200,200]	C8C8C8
Oxygen	red		[240,0,0]	F00000
Hydrogen	`white`		[255,255,255]	FFFFFF
Nitrogen	sky blue		[143,143,255]	8F8FFF
Sulfur	yellow		[255,200,50]	FFC832
Phosphorus	`orange`		[255,165,0]	FFA500
Chlorine	`green`		[0,255,0]	00FF00
Bromine, Zinc	brown		[165,42,42]	A52A2A
Sodium	`blue`		[0,0,255]	0000FF
Iron	`orange`		[255,165,0]	FFA500
Magnesium	forest green		[34,139,34]	228B22
Calcium	dark grey		[128,128,144]	808090
Unknown	deep pink		[255,20,147]	FF1493

Note that except for **`green`**, **`white`**, **`blue`**, and **`orange`**, these colour names are not the ones specified as "**Predefined colours**" in RasMol; thus, they can only be specified on the command line as RGB triplets.

In the CPK colouring scheme, RasMol will attempt to assign a colour to each element from the periodic table from a list of 16 colours (the colour codes listed are to help in understanding the mapping and are not used by RasMol):

Code	colour Name	Sample	RGB Values	Hexadecimal
LG	**`Light Grey`**		[200,200,200]	C8C8C8
SB	**`Sky Blue`**		[143,143,255]	8F8FFF
R	**`Red`**		[240, 0, 0]	F00000
Y	**`Yellow`**		[255,200, 50]	FFC832
W	**`White`**		[255,255,255]	FFFFFF

Pk	**Pink**		[255,192,203]	FFC0CB
Go	**Golden Rod**		[218,165, 32]	DAA520
Bl	**Blue**		[0, 0,255]	0000FF
Or	**Orange**		[255,165, 0]	FFA500
DG	**Dark Grey**		[128,128,144]	808090
Br	**Brown**		[165, 42, 42]	A52A2A
P	**Purple**		[160, 32,240]	A020F0
DP	**Deep Pink**		[255, 20,147]	FF1493
G	**Green**		[0,255, 0]	00FF00
FB	**Fire Brick**		[178, 34, 34]	B22222
FG	**Forest Green**		[34,139, 34]	228B22

In the CPKnew colouring scheme, RasMol uses brighter colours:

Code	colour Name	Sample	RGB Values	Hexadecimal
LG	**Light Grey**		[211,211,211]	D3D3D3
SB	**Sky Blue**		[135,206,235]	87CEE6
R	**Red**		[255, 0, 0]	FF0000
Y	**Yellow**		[255,255, 0]	FFFF00
W	**White**		[255,255,255]	FFFFFF
Pk	**Pink**		[255,192,203]	FFC0CB
Go	**Golden Rod**		[218,165, 32]	DAA520
Bl	**Blue**		[0, 0,255]	0000FF
Or	**Orange**		[255,170, 0]	FFAA00
DG	**Dark Grey**		[105,105,105]	696969

Br	**`Brown`**		`[128, 40, 40]`	`802828`
P	**`Purple`**		`[160, 32,240]`	`A020F0`
DP	**`Deep Pink`**		`[250, 22,145]`	`FA1691`
G	**`Green`**		`[ 0,255, 0]`	`00FF00`
FB	**`Fire Brick`**		`[178, 33, 33]`	`B22121`
FG	**`Forest Green`**		`[ 34,139, 34]`	`228B22`

1a	**2a**	**3b**	**4b**	**5b**	**6b**	**7b**	**8**			**1b**	**2b**	**3a**	**4a**	**5a**	**6a**	**7a**	**0**
H 1 W																	He 2 Pk
Li 3 FB	Be 4 DP											B 5 G	C 6 LG	N 7 SB	O 8 R	F 9 Go	Ne 10 DP
Na 11 Bl	Mg 12 FG											Al 13 DG	Si 14 Go	P 15 Or	S 16 Y	Cl 17 G	Ar 18 DP
K 19 DP	Ca 20 DG	Sc 21 DP	Ti 22 DG	V 23 DP	Cr 24 DG	Mn 25 DG	Fe 26 Or	Co 27 DP	Ni 28 Br	Cu 29 Br	Zn 30 Br	Ga 31 DP	Ge 32 DP	As 33 DP	Se 34 DP	Br 35 Br	Kr 36 DP
Rb 37 DP	Sr 38 DP	Y 39 DP	Zr 40 DP	Nb 41 DP	Mo 42 DP	Tc 43 DP	Ru 44 DP	Rh 45 DP	Pd 46 DP	Ag 47 DG	Cd 48 DP	In 49 DP	Sn 50 DP	Sb 51 DP	Te 52 DP	I 53 P	Xe 54 DP
Cs 55 DP	Ba 56 Or	La 57 DP	Hf 72 DP	Ta 73 DP	W 74 DP	Re 75 DP	Os 76 DP	Ir 77 DP	Pt 78 DP	Au 79 Go	Hg 80 DP	Tl 81 DP	Pb 82 DP	Bi 83 DP	Po 84 DP	At 85 DP	Rn 86 DP
Fr 87 DP	Ra 88 DP	Ac 89 DP															
Lanthinide Series		Ce 58 DP	Pr 59 DP	Nd 60 DP	Pm 61 DP	Sm 62 DP	Eu 63 DP	Gd 64 DP	Tb 65 DP	Dy 66 DP	Ho 67 DP	Er 68 DP	Tm 69 DP	Yb 70 DP	Lu 71 DP		
Actinide Series		Th 90 DP	Pa 91 DP	U 92 DP	Np 93 DP	Pu 94 DP	Am 95 DP	Cm 96 DP	Bk 97 DP	Cf 98 DP	Es 99 DP	Fm 100 DP	Md 101 DP	No 102 DP	Lr 103 DP		

For X-ray crystallographic models of proteins and nucleic acids (*i.e*. without hydrogens) the display can be 'brightened' by converting the O, C, and N atoms from the RasMol default **`cpk`** colors to "true red, white and blue" using RasMol's predefined colour scheme. Use the following sequence of commands to try it:

```
select all
select oxygen
colour red
select carbon
colour white
select nitrogen
colour blue
select all
```

Extension of this idea to other atoms and colour schemes is straightforward.

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 heterogeneous molecules associated with it, the macromolecule may not be drawn in the full 'range' of the spectrum. '**Group**' can be selected from the RasMol '**Colours**' menu.

If a chain has a large number of heterogeneous 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 heteroatoms, 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 aggravated 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.

NMR Model Colours

The RasMol '**model**' colour scheme codes each NMR model with a distinct colour. The NMR model number is taken as a numeric value. High values are coloured in blue and lower values coloured in red. Rather than use a fixed scale this scheme determines the maximum value of the NMR model number and interpolates from red to blue appropriately.

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.

Residues	Colour Name	Sample	RGB Values	Hexadecimal
ALA	**Medium Green**		[140,255,140]	8CFF8C
GLY	**White**		[255,255,255]	FFFFFF
LEU	**Olive Green**		[69, 94, 69]	455E45
SER	**Medium Orange**		[255,112, 66]	FF7042
VAL	**Light Purple**		[255,140,255]	FF8CFF
THR	**Dark Orange**		[184, 76, 0]	B84C00
LYS	**Royal Blue**		[71, 71,184]	4747B8
ASP	**Dark Rose**		[160,0,66]	A00042
ILE	**Dark Green**		[0, 76, 0]	004C00
ASN	**Light Salmon**		[255,124,112]	FF7C70
GLU	**Dark Brown**		[102, 0, 0]	660000
PRO	**Dark Grey**		[82, 82, 82]	525252
ARG	**Dark Blue**		[0, 0,124]	00007C
PHE	**Olive Grey**		[83, 76, 66]	534C42
GLN	**Dark Salmon**		[255, 76, 76]	FF4C4C
TYR	**Medium Brown**		[140,112,76]	8C704C
HIS	**Medium Blue**		[112,112,255]	7070FF
CYS	**Medium Yellow**		[255,255,112]	FFFF70
MET	**Light Brown**		[184,160, 66]	B8A042
TRP	**Olive Brown**		[79, 70, 0]	4F4600
ASX,GLX,PCA,HYP	**Medium Purple**		[255, 0,255]	FF00FF
A	**Light Blue**		[160,160,255]	A0A0FF

C	**Light Orange**		[255,140,75]	FF8C4B
G	**Medium Salmon**		[255,112,112]	FF7070
T	**Light Green**		[160,255,160]	A0FFA0
Backbone	**Light Grey**		[184,184,184]	B8B8B8
Special	**Dark Purple**		[94, 0, 94]	5E005E
Default	**Medium Purple**		[255, 0,255]	FF00FF

Structure Colours

The RasMol '**structure**' colour scheme colours the molecule by protein secondary structure. Alpha helices are coloured magenta, [240,0,128], beta sheets are coloured yellow, [255,255,0], 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, SHEET and TURN 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 values go from red to blue.

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.

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 associated 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]

Amino Acid Codes

The following table lists the names, single letter and three letter codes of each of the amino acids.

Alanine	A	ALA	Arginine	R	ARG
Asparagine	N	ASN	Aspartic acid	D	ASP
Cysteine	C	CYS	Glutamic acid	E	GLU
Glutamine	Q	GLN	Glycine	G	GLY
Histidine	H	HIS	Isoleucine	I	ILE

Leucine	L	LEU	Lysine	K	LYS
Methionine	M	MET	Phenylalanine	F	PHE
Proline	P	PRO	Serine	S	SER
Threonine	T	THR	Tryptophan	W	TRP
Tyrosine	Y	TYR	Valine	V	VAL

Booleans

A boolean parameter is a truth value. Valid boolean values are 'true' and 'false', and their synonyms 'on' and 'off'. Boolean parameters are commonly used by RasMol to either enable or disable a representation or option.

File Formats

Protein Data Bank Files

If you do not have the PDB documentation, you may find the following summary of the PDB file format useful. The Protein Data Bank is a computer-based archival database for macromolecular structures. The database was established in 1971 by Brookhaven National Laboratory, Upton, New York, as a public domain repository for resolved crystallographic structures. The Bank uses a uniform format to store atomic coordinates and partial bond connectivities as derived from crystallographic studies. In 1999 the Protein Data Bank moved to the Research Collaboratory for Structural Biology.

PDB file entries consist of records of 80 characters each. Using the punched card analogy, columns 1 to 6 contain a record-type identifier, the columns 7 to 70 contain data. In older entries, columns 71 to 80 are normally blank, but may contain sequence information added by library management programs. In new entries conforming to the 1996 PDB format, there is other information in those columns. The first four characters of the record identifier are sufficient to identify the type of record uniquely, and the syntax of each record is independent of the order of records within any entry for a particular macromolecule.

The only record types that are of major interest to the RasMol program are the ATOM and HETATM records which describe the position of each atom. ATOM/HETATM records contain standard atom names and residue abbreviations, along with sequence identifiers, coordinates in Ångstrom units, occupancies and thermal motion factors. The exact details are given below as a FORTRAN format statement. The "fmt" column indicates use of the field in all PDB formats, in the 1992 and earlier formats or in the 1996 and later formats.

```
FORMAT(6A1,I5,1X,A4,A1,A3,1X,A1,I4,A1,3X,3F8.3,2F6.2,1X,I3,2X,A4,2A2)
```

Column	Content	fmt
1-6	'ATOM' or 'HETATM'	all
7-11	Atom serial number (may have gaps)	all
13-16	Atom name, in IUPAC standard format	all
17	Alternate location indicator indicated by A, B or C	all
18-20	Residue name, in IUPAC standard format	all
23-26	Residue sequence number	all
27	Code for insertions of residues (i.e. 66A & 66B)	all
31-38	X coordinate	all
39-46	Y coordinate	all
47-54	Z coordinate	all
55-60	Occupancy	all
61-66	Temperature factor	all
68-70	Footnote number	92
73-76	Segment Identifier (left-justified)	96

77-78	Element Symbol (right-justified)	96
79-80	Charge on the Atom	96

Residues occur in order starting from the N-terminal residue for proteins and 5'-terminus for nucleic acids. If the residue sequence is known, certain atom serial numbers may be omitted to allow for future insertion of any missing atoms. Within each residue, atoms are ordered in a standard manner, starting with the backbone (N-C-C-O for proteins) and proceeding in increasing remoteness from the alpha carbon, along the side chain.

HETATM records are used to define post-translational modifications and cofactors associated with the main molecule. TER records are interpreted as breaks in the main molecule's backbone.

If present, RasMol also inspects HEADER, COMPND, HELIX, SHEET, TURN, CONECT, CRYST1, SCALE, MODEL, ENDMDL, EXPDTA and END records. Information such as the name, database code, revision date and classification of the molecule are extracted from HEADER and COMPND records, initial secondary structure assignments are taken from HELIX, SHEET and TURN records, and the end of the file may be indicated by an END record.

RasMol Interpretation of PDB fields

Atoms located at 9999.000, 9999.000, 9999.000 are assumed to be Insight pseudo atoms and are ignored by RasMol. Atom names beginning ' Q' are also assumed to be pseudo atoms or position markers.

When a data file contains an NMR structure, multiple conformations may be placed in a single PDB file delimited by pairs of MODEL and ENDMDL records. RasMol displays all the NMR models contained in the file.

Residue names "CSH", "CYH" and "CSM" are considered pseudonyms for cysteine "CYS". Residue names "WAT", "H20", "SOL" and "TIP" are considered pseudonyms for water "HOH". The residue name "D20" is consider heavy water "DOD". The residue name "SUL" is considered a sulphate ion "SO4". The residue name "CPR" is considered to be cis-proline and is translated as "PRO". The residue name "TRY" is considered a pseudonym for tryptophan "TRP".

RasMol uses the HETATM fields to define the sets hetero, water, solvent and ligand. Any group with the name "HOH", "DOD", "SO4" or "PO4" (or aliased to one of these names by the preceding rules) is considered a solvent and is considered to be defined by a HETATM field.

RasMol only respects CONECT connectivity records in PDB files containing fewer than 256 atoms. This is explained in more detail in the section on determining molecule connectivity. CONECT records that define a bond more than once are interpreted as specifying the bond order of that bond, *i.e*. a bond specified twice is a double bond and a bond specified three (or more) times is a triple bond. This is not a standard PDB feature.

PDB Colour Scheme Specification

RasMol also accepts the supplementary COLO record type in the PDB files. This record format was introduced by David Bacon's Raster3D program for specifying the colour scheme to be used when rendering the molecule. This extension is not currently supported by the PDB. The COLO record has the same basic record type as the ATOM and HETATM records described above.

Colours are assigned to atoms using a matching process. The Mask field is used in the matching process as follows. First RasMol reads in and remembers all the ATOM, HETATM and COLO records in input order. When the user-defined ('User') colour scheme is selected, RasMol goes through each remembered ATOM/HETATM record in turn, and searches for a COLO record that matches in all of columns 7 through 30. The first such COLO record to be found determines the colour and radius of the atom.

Column	**Content**
1-6	'COLOR' or 'COLOUR'
7-30	Mask (described below)
31-38	Red component
39-46	Green component
47-54	Blue component
55-60	Sphere radius in Ångstroms
61-70	Comments

Note that the Red, Green and Blue components are in the same positions as the X, Y, and Z components of an ATOM or HETA record, and the van der Waals radius goes in the place of the Occupancy. The Red, Green and Blue components must all be in the range 0 to 1.

In order that one COLO record can provide colour and radius specifications for more than one atom (e.g. based on residue, atom type, or any other criterion for which labels can be given somewhere in columns 7 through 30), a 'don't-care' character, the hash mark "#" (number or sharp sign) is used. This character, when found in a COLO record, matches any character in the corresponding column in a ATOM/HETATM record. All other characters must match identically to count as a match. As an extension to the specification, any atom that fails to match a COLO record is displayed in white.

Multiple NMR Models

RasMol loads all of the NMR models from a PDB file no matter which command is used: '**`load pdb <filename>`**' or '**`load nmrpdb <filename>`**'

Once multiple NMR conformations have been loaded they may be manipulated with the atom expression extensions described in '**`Primitive Expressions`**'. In particular, the command '**`restrict */1`**' will restrict the display to the first model only.

CIF and mmCIF Format Files

CIF is the IUCr standard for presentation of small molecules and mmCIF is intended as the replacement for the fixed-field PDB format for presentation of macromolecular structures. RasMol can accept data sets in either format.

There are many useful sites on the World Wide Web where information tools and software related to CIF, mmCIF and the PDB can be found. The following are good starting points for exploration:

The International Union of Crystallography (IUCr) provides access to software, dictionaries, policy statements and documentation relating to CIF and mmCIF at: IUCr, Chester, England (www.iucr.org/iucr-top/cif/) with many mirror sites.

The Nucleic Acid Database Project provides access to its entries, software and documentation, with an mmCIF page giving access to the dictionary and mmCIF software tools at Rutgers University, New Jersey, USA (http://ndbserver.rutgers.edu/NDB/mmcif) with many mirror sites.

This version of RasMol restricts CIF or mmCIF tag values to essentially the same conventions as are used for the fixed-field PDB format. Thus chain identifiers and alternate conformation identifiers are limited to a single character, atom names are limited to 4 characters, etc. RasMol interprets the following CIF and mmCIF tags:

mmCIF tag	**CIF tag**	**Used for**
_struct_biol.details		Info.classification
_database_2.database_code		Info.identcode
_entry.id		
_struct_biol.id		
_struct.title	Info.moleculename	
_chemical_name_common		
_chemical_name_systematic		
_chemical_name_mineral		
_symmetry.space_group_name_H-M	_symmetry_space_group_name_H-M	Info.spacegroup
_cell.length_a	_cell_length_a	Info.cell
_cell.length_b	_cell_length_b	
_cell.length_c	_cell_length_c	
_cell.angle_alpha	_cell_angle_alpha	
_cell.angle_alpha	_cell_angle_alpha	
_cell.angle_beta	_cell_angle_beta	
_cell.angle_gamma	_cell_angle_gamma	
_atom_sites.fract_transf_matrix[1][1]	_atom_sites_fract_tran_matrix_11	Used to compute orthogonal coords
...	...	
_atom_sites.fract_transf_vector[1]	_atom_sites_fract_tran_vector_1	

...	...	
_atom_sites.cartn_transf_matrix[1][1]	_atom_sites_cartn_tran_matrix_11	Alternative to compute orth. coords
...	...	
_atom_sites.cartn_transf_vector[1]	_atom_sites_cartn_tran_vector_1	
...	...	
_atom_site.cartn_x	_atom_site_cartn_x	atomic coordinates
...	...	
or		
_atom_site.fract_x	_atom_site_fract_x	
...	...	
_struct_conn.id		bonds
...		
_geom_bond.atom_site_id_1	_geom_bond_atom_site_label_1	
...	...	
_struct_conf.id		helices, sheets, turns
_struct_sheet_range.id		
...		

A search is made through multiple data blocks for the desired tags, so a single dataset may be composed from multiple data blocks, but multiple data sets may not be stacked in the same file.

Machine-Specific SupportMachine-Specific SupportMachine-Specific SupportMachine-Specific Support

In the following sections, support for '**Monochrome X-Windows**', '**Tcl/Tk IPC**', '**UNIX sockets based IPC**', '**Compiling RasWin with Borland and MetroWerks**' are described.

Monochrome X-Windows Support

RasMol supports the many monochrome UNIX workstations typically found in academia, such as low-end SUN workstations and NCD X-terminals. The X11 version of RasMol (when compiled in 8 bit mode) now detects black and white X-Windows displays and enables dithering automatically. The use of run-time error diffusion dithering means that all display modes of RasMol are available when in monochrome mode. For best results, users should experiment with the set ambient command to ensure the maximum contrast in resulting images.

Tcl/Tk IPC support

Version 4 of Tk graphics library changed the protocol used to communicate between Tk applications. RasMol version 2.6 was modified such that it could communicate with both this new protocol and the previous version 3 protocol supported by RasMol v2.5. Although Tcl/Tk 3.x applications may only communicate with other 3.x applications and Tcl/Tk 4.x applications with other 4.x applications, these changes allow RasMol to communicate between processes with both protocols (potentially concurrently).

UNIX sockets based IPC

The UNIX implementation of RasMol supports BSD-style socket communication. An identical socket mechanism is also being developed for VMS, Apple Macintosh and Microsoft Windows systems. This should allow RasMol to interactively display results of a computation on a remote host. The current protocol acts as a TCP/IP server on port 21069 that executes command lines until either the command '**exit**' or the command '**quit**' is typed. The command exit from the RasMol server, the command '**quit**' both disconnects the current session and terminates RasMol. This functionality may be tested using the UNIX command '**telnet <hostname> 21069**'.

Compiling RasWin with Borland and MetroWerks

A number of changes were made to the source code in the transition from version 2.5 to 2.6 to allow the Microsoft Windows version of RasMol to compile using the Borland C/C++ compiler. These fixes include name changes for the standard library and special code to avoid a bug in _fmemset. Additional changes were made in the transition from 2.6 to 2.7 to allow compilation with the MetroWerks compilers.

Molecular Graphics

[1] Nelson Max, "**Computer Representation of Molecular Surfaces**", *IEEE Computer Graphics and Applications*, pp.21-29, August 1983.

[2] Arthur M. Lesk, "**Protein Architecture: A Practical Approach**", IRL Press Publishers, 1991.

Molecular Graphics Programs

[3] Per J. Kraulis, "**MOLSCRIPT: A Program to Produce both Detailed and Schematic Plots of Protein Structures**", *Journal of Applied Crystallography*, Vol.24, pp.946-950, 1991.

[4] David Bacon and Wayne F. Anderson, "**A Fast Algorithm for Rendering Space-Filling Molecule Pictures**", *Journal of Molecular Graphics*, Vol.6, No.4, pp.219-220, December 1988.

[5] David C. Richardson and Jane S. Richardson, "**The Kinemage: A tool for Scientific Communication**", *Protein Science*, Vol.1, No.1,pp.3-9, January 1992.

[6] Mike Carson, "**RIBBONS 2.0**", *Journal of Applied Crystallography*, Vol.24, pp.958-961, 1991.

[7] Conrad C. Huang, Eric F. Pettersen, Teri E. Klein, Thomas E. Ferrin and Robert Langridge, "**Conic: A Fast Renderer for Space-FillingMolecules with Shadows**", *Journal of Molecular Graphics*, Vol.9, No.4, pp.230-236, December 1991.

Molecular Biology Algorithms

[8] Wolfgang Kabsch and Christian Sander, "**Dictionary of Protein Secondary Structure: Pattern Recognition of Hydrogen-Bonded and Geometrical Features**", *Biopolymers*, Vol.22, pp.2577-2637, 1983.

[9] Michael L. Connolly, "**Solvent-Accessible Surfaces of Proteins and Nucleic Acids**", *Science*, Vol.221, No.4612, pp.709-713, August 1983.

[10] Khaled Belhadj-Mostefa, Ron Poet and E. James Milner-White, "**Displaying Inter-Main Chain Hydrogen Bond Patterns in Proteins**", *Journal of Molecular Graphics*, Vol.9, No.3, pp.194-197, September 1991.

[11] Mike Carson, "**Ribbon Models of Macromolecules**", *Journal of Molecular Graphics*, Vol.5, No.2, pp.103-106, June 1987.

[12] Mike Carson and Charles E. Bugg, "**Algorithm for Ribbon Models of Proteins**", *Journal of Molecular Graphics*, Vol.4, No.2, pp.121-122, June 1986.

[13] H. Iijima, J. B. Dunbar Jr. and G. Marshall, "**Calibration of Effective van der Waals Atomic Contact Radii for Proteins and Peptides**", *Proteins: Structure, Functions and Genetics*, Vol.2, pp.330-339,1987.

Graphics Algorithms

[14] J. Foley, A. van Dam, S. Feiner and J. Hughes, "**Computer Graphics: Principles and Practice**", 2nd Edition, Addison Wesley Publishers, 1990.

[15] J. Cleary and G. Wyvill, "**Analysis of an Algorithm for Fast Ray Tracing using Uniform Space Subdivision**", *The Visual Computer*, Vol.4, pp.65-83, 1988.

[16] Thomas Porter, "**Spherical Shading**", *Computer Graphics* Vol.12, ACM SIGGRAPH, pp.282-285, 1978.

[17] Jean-Michel Cense, "**Exact Visibility Calculation for Space-Filling Molecular Models**", *Journal of Molecular Graphics*, Vol.9, No.3, pp.191-193, September 1991.

[18] Chris Schafmeister, "**Fast Algorithm for Generating CPK Images on Graphics Workstations**", *Journal of Molecular Graphics*, Vol.8, No.4, pp.201-206, December 1990.

[19] Bruce A. Johnson, "**MSURF: A Rapid and General Program for the Representation of Molecular Surfaces**", *Journal of Molecular Graphics*, Vol.5, No.3, pp.167-169, September 1987.

File Formats

[20] Frances C. Bernstein et al., "**The Protein Data Bank: A Computer-Based Archival File for Macromolecular Structures**", *Journal of Molecular Biology*, Vol.112, pp.535-542, 1977.

[21] Arthur Dalby, James G. Nourse, W. Douglas Hounshell, Ann K. I. Gushurst, David L. Grier, Burton A. Leland and John Laufer, "**Description of Several Chemical File Formats Used by Computer Programs Developed at Molecular Design Limited**", *Journal of Chemical Information and Computer Sciences*, Vol.32, No.3, pp.244-255, 1992.

[22] Adobe Systems Inc., "**PostScript Language Reference Manual**", Addison-Wesley Publishers, Reading, Mass., 1985.

[23] Philip E. Bourne et al., "**The Macromolecular Crystallographic Information File (mmCIF)**", Meth. Enzymol. (1997) 277, 571-590.

[24] Sydney R. Hall, "**The STAR File: a New Format for Electronic Data Transfer and Archiving**", *Journal of Chemical Information and Computer Sciences*, Vol. 31, 326-333, 1991.

Updated 19 April 2005.

Herbert J. Bernstein
Bernstein + Sons, 5 Brewster Lane, Bellport, NY 11713-2803, USA
yaya@bernstein-plus-sons.com
+1-631-286-1339