CCP4 Molecular Graphics Documentation Contents

Documentation Contents | On-line Documentation | Tutorials | CCP4mg Home
Quick Reference | Print version | Contact | References

Documentation Contents

General

Molecular models

Other types of data

Applications

Other utilities

- Lighting
- Saving the program status
- Picture Definition Files (also Object attributes and Introduction to Python)
- Making presentations

Quick Reference

See **Display Table** for general overview of the display table. For information on specific type of data click the appropriate data type in the *File* list below.

File	Tools	View	Applications
Coordinate Files	General tools	General tools	Movies
Downloading Coordinate Files	..	..	Superpose
Read electron density map/MTZ	Save/restore view	Lighting	Picture Wizard
Read QM maps	Save/restore status	Transparency	Geometry
Add vectors/labels	Picture definition files	View from	
Add image	History and Scripting	Rotate,translate,align view	
Add legend	Presentation/Notebook	RocknRoll	
Add crystal	List monomer definition		
Model definition file	Preferences		
Open presentation			
Tutorials			
Output screen image			

Windows Bring a 'lost' window to the front.
Projects Edit projects or change projects.

Printer-friendly version

These help pages are also available as PDF files:
.../ccp4mg/help/documentation.pdf
.../ccp4mg/help/tutorials.pdf

On-line help

- http://www.ysbl.york.ac.uk/~ccp4mg/ccp4mg_help/index.html has the latest help pages which may have more information.

Contact Us

CCP4mg developers can be contacted at ccp4mg@ccp4.ac.uk

References

The CCP4 molecular-graphics project
E. Potterton, S. McNicholas, E. Krissinel, K. Cowtan and M. Noble
Acta Cryst. (2002). D**58**, 1955-1957

Developments in the CCP4 molecular-graphics project

L. Potterton, S. McNicholas, E. Krissinel, J. Gruber, K. Cowtan, P. Emsley, G. N. Murshudov, S. Cohen, A. Perrakis and M. Noble

Acta Cryst. (2004). D**60**, 2288-2294

CCP4 Molecular Graphics Documentation

The graphics window

Documentation Contents | On-line Documentation | Tutorials | CCP4mg Home

Contents

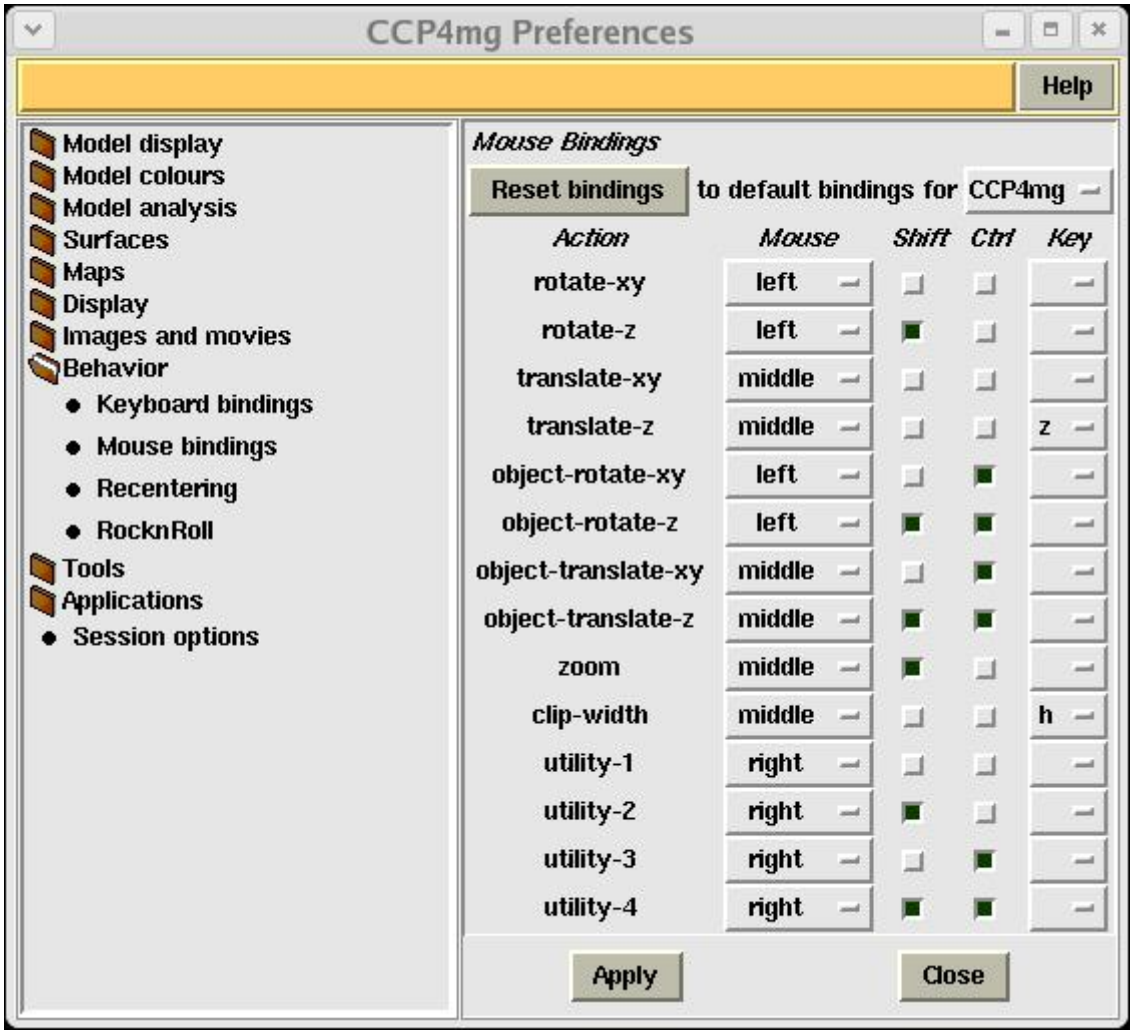
Mouse bindings

You really must have a three button mouse!!

By default clicking with the left mouse is used to identify an object, the middle mouse button is used to paste and the right mouse button is used to bring up a pop-up menu. To rotate the display hold down the left mouse button and move the cursor left-right or up-down; to translate hold down the middle mouse button and move the cursor in the direction that you want to drag the display. To rotate about the z-axis (i.e. the axis perpendicular to the screen) or translate in the z direction hold down the *shift* key and use the appropriate mouse button. Note that dragging downwards brings things forwards.

To rotate or translate a selected object use the same mouse buttons and movements but also hold down the *Ctrl* key. To select the moving object use the ***Move*** command on the objects icon menu. The moving object is indicated on the Display Table by highlighting the icon in gold colour.

The bindings can be changed from the **Mouse bindings** interface that is in the **Behaviour** folder of the **Preferences** window (accessed from the **Tools** menu).



This interface lists the actions and the mouse and key bindings for the action. The binding can be to left,right or middle mouse button, the *Shift* and/or the *Ctrl* keys can be depressed and any one other keyboard key might be depressed (this set in the right hand column of the interface). The interface also has the option to set the bindings to the default bindings of different molecular graphics programs. Note that other programs may have feature which is not supported in CCP4mg and we have needed to make slight changes. The convention in CCP4mg is that the transforming individual objects uses the same bindings as the equivalent view transformation but with the *Ctrl* key depressed. Maintaining this convention while using *Coot* bindings is difficult; the *f* key is used.

Keyboard shortcuts

If the cursor is in the main graphics window then some commonly used functions can be accessed rapidly by pressing a keyboard key. Currently only the number keys 1 to 10 are used for general functions (note these are the actual numbers and not the function keys). The character keys will be used by specific applications. The bindings can be changed from the **Keyboard bindings** interface that is in the **Behavior** folder of the **Preferences** window (accesed from the **Tools** pull-down menu.

Action	**Mouse/keyboard**	**Notes**
Rotate about vertical (y) axis	hold down left mouse button and drag left-right	Keyboard 'o'/'p' or (faster) 'O'/'P'

Rotate about horizontal (x) axis	hold down left mouse button and drag up-down	Keyboard 'a'/'q' or (faster) 'A'/'Q'
Rotate about out-of-plane (z) axis	hold down shift key and left mouse button and drag in circular motion	Keyboard 'w'/'e' or (faster) 'W'/'E'
Translate in x direction	hold down middle mouse button and drag left-right	
Translate in y direction	hold down middle mouse button and drag up-down	
Zoom	hold down Shift key and middle mouse button and drag up-down	mouse wheel
Translate in z direction	hold down 'z' key and middle mouse and drag (down for forward)	
Clipping planes separation	hold down 'h' and middle mouse button and drag up/down	Depress the '<' (shift key and '**,**') or '>' (shift key and '**.**')
Label an atom	click on atom with left mouse button	
Centre on atom	double click on atom with left mouse button	
Atom pop-up menu - centre on, select around,geometry	Right mouse click on an atom	
Quick pop-up menu - clear atom labels, show display table, recentre, centre on..	Right mouse click NOT on any displayed object	
Next residue	Space bar	You must have centered on a residue by double-click or using Find/Recentre
Find/Centre on	key 1	***Find/Recentre*** on ***Tools*** menu or quick pop-up menu.
Clear labels	key 2	Quick pop-up menu or ***Clear picked atom labels*** on ***Tools*** menu
Toggle stereo on/off	key 3	
Toggle fog on/off	key 4	
Hide/show all molecules	key 5	
Hide/show all maps	key 6	
Background black/white	key 7	
Recentre	***Recentre*** on ***Tools*** menu or quick pop-up menu.	***Recentre on*** in the model or display object icon menu
Pasting atom/residue name into a window.	Left mouse click on atom and middle mouse click in GUI box	
Toggle data object or display object visibility	Right mouse click on the object icon menu	

CCP4 Molecular Graphics Documentation

Tools Menu

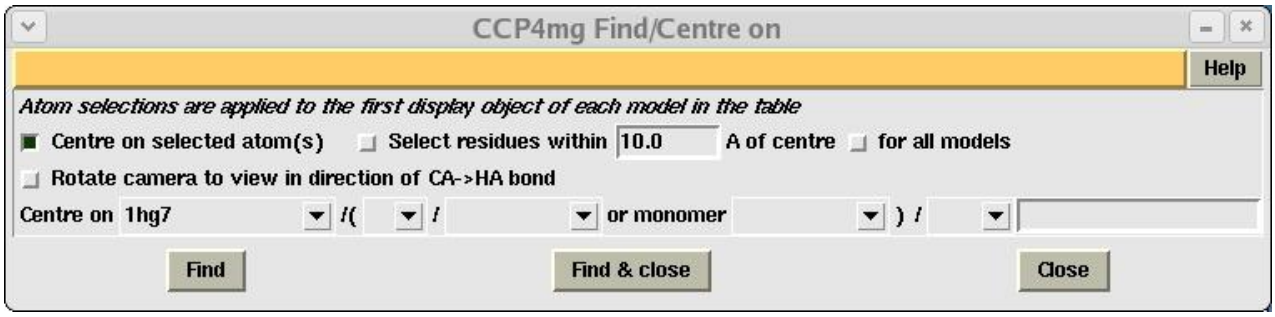
Documentation Contents | On-line Documentation | Tutorials | CCP4mg Home

Contents

Tools menu

Recentre
Zoom in or out to ensure that all graphical objects are on the screen and filling the screen.

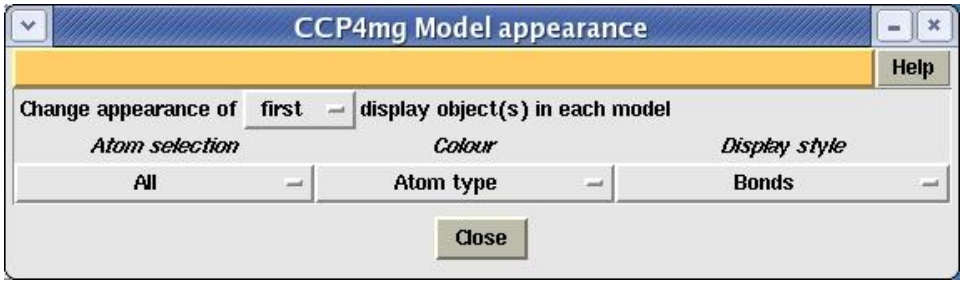
Find/Centre on..
Opens the window



The bottom line of this window is a molecule selection widget which you can use to select a chain, residue or atom. By default, when you click **Find** or **Find&close** the display will recentre on the selected atom(s) and will label them. If the **Centre on selected atom(s)** button is toggled off then the atom(s) will only be labelled. By default the orientation of the view is not changed but if **Rotate camera to view in direction of CA-HA bond** is toggled on then, if the selected residue is an amino acid, the orientation will be changed as specified. This interface can also be used to select residues around the specified atoms: toggle on the **Select residues within..** button and set the radius as required. If the **for all models** button is on then a sphere of atoms will be selected for all of the models. Note that the selection is applied to the first display object for each model; these selection options are probably most useful if there is one display object per model.

Clear picked atom labels
Clear all atom labels created by picking or **Find**ing an atom. Note the Preferences option **Picked atom labels** in the **Model display** folder controls the behaviour of picked atom labels.

For all models
Hide Hide (i.e. make invisible) all model data objects. The dot next to the model name will become a circle to indicate that the model is hidden.
Show Show (i.e. make visible) all model data objects. The dot next to the model names will be a filled-in dot. Note that if some of the model display objects are hidden (indicated by greyed-out icon) then they will remain hidden.
Appearance Opens the window



The selection, colour and display menus are similar to those for each model object on the display table. Any item picked from the menu will be applied to all models. The **Change appearance of** menu controls whether the change is applied to just the first or to all display objects in the models.

Delete Delete all loaded models. You will be asked to confirm.

For all data

Hide Hide (i.e. make invisible) all data objects. The dot next to the data object name will become a circle to indicate that the data is hidden.

Show Show (i.e. make visible) all data objects. The dot next to the data object names will be a filled-in dot. Note that if some of the display objects are hidden (indicated by greyed-out icon) then they will remain hidden.

Delete Delete all loaded data. You will be asked to confirm

Save/restore view

Save view You are prompted for a file name with the default extension *.view.pkl* and the current view is saved to that file.

Restore view You must select a file, default extension *.view.pkl*, and the view is restored from that file.

Save/restore status

See Program Status

Save status You are prompted for a file name with the default extension *.status.pkl* and the complete current program status is saved to that file.

Save as default status Save the complete program status to your *.CCP4MG/status/last_exit_status.pkl* file. This file is used to initialise the program on restart. This option is most useful when you fear the program may crash.

Restore status You must select a file, default extension *.status.pkl*, and the complete program status is restored from that file.

Restore default status Restore status from *.CCP4MG/status/last_exit_status.pkl* file. This is the status of the program when it was restarted or when you lasted did *Save as default status*.

Picture definition file

See Picture definition files

Read picture definition You must select a file, default extension *.mgpic.py*, and the selected data and display status are restored from the file. Note the options at the top of the file selection window to **Keep currently loaded data** (default off, so other data is deleted) and **Hide slow-to-draw objects such as surfaces** (default on, so loading is quicker and surfaces need to be toggled on when required).

Save picture definition Save the current selected data and display status to file with extension *.mgpic.py*.

Save options window controls which display and colour Preferences (as set in the Preferences window) are also written to the file. NB the option at top of the file selection window to **Only save non-default parameters** - this is useful to simplify the file.

History & Scripting

See History and Scripting

Replay script Select a script file (default extension *.py*). Note the replay mode options in the file selection window; see here.

Review and record will open a window showing the commands entered in this session of the program. There are options to save all or some limited selection of commands to a script file; see here for details.

Presentation/Notebook

See Presentations for details of the Presentation interface.

Edit You must select an existing presentation directory (default extension *.ccp4mg_presentation* to edit.

Create You are required to enter a name for a new presentation directory (default extension *.ccp4mg_presentation*

Note that **Open presentation** on the **File** menu is the usual way to open a Presentation in show mode.

List monomer definition

List a REFMAC monomer library definition. See Atom typing.

Preferences..

See below.

Preferences

Open the preferences window by selecting **Preferences** from the bottom of the **Tools** menu or from the **Windows** menu. The left panel of the window lists folders, click on the folder name to open the folder and click on the preference title to present the preferences on the right side of the window. Note that most preferences require you to hit the **Apply** button for changes to take effect.

All preferences are saved automatically to the program status file and so are saved and restored between sessions. The preferences are optionally saved to picture definition files.

For some preferences **Model display**->**Drawing style**, **Surfaces**->**Surface parameters** the global preference can be over-ridden by setting preferences for individual display objects.

The groups of preferences are described below:

Model display

Default appearance shows selection, colour and display style menus similar to those for each model display object. All new model display objects will initially be drawn with the parameters set here.
Render quality The **Solid object render quality** is**Smooth** by default. The drawing style is improved by drawing more and finer facets for each solid object surface so there is a trade-off between quality and speed and the appropriate setting will depend on the size of your computer versus the size of your models. When CCP4mg creates output images it automatically redraws with optimal quality - this is not affected by the render quality set here.
Drawing style shows parameters for display objects such are spheres and ribbons. See examples.
Atom labels By default when a model is loaded it is not labelled with atomIDs. The **Put labels on** option in these preferences can be set so models are automatically labelled. There is a list of atom properties which can be selected for inclusion in the label. The format of the atom label is the Coordinate ID. All model display objects have **Label atoms** and **Label text** options on the icon menu which will override the options set here.
Picked atom labels There is a list of atom properties which can be selected for inclusion in the label of a picked atom. The format of the atom label is the Coordinate ID. To avoid cluttering the display the program will automatically limit the number of picked labels to **Max picked labels** by deleting the oldest labels. You can change the number of labels or set it to 999 (effectively no limit).

Model colours

The various colour schemes are defined. It is possible, for example, to reset the default colour for a carbon atom. It is possible to override colour scheme defined here by selecting **Edit colour scheme..** from the Colour menu for each individually model display object.
For schemes where a colour is applied dependent of some floating point value (e.g. temperature factor) see Colouring by Atom/Residue Property.

Model analysis

Monomer types Specify the type of a monomer (e.g. '*amino_acid*' or '*solvent*') for use in the atom selection interface.
Hydrogen bonds Set the parameters for calculating hydrogen bonds.
Close contacts Set the parameters for calculating close contacts.
Secondary structure Set the parameters for calculating secondary structure.
Solvent accessible surface Parameters for colouring by solvent accessible surface area and buried area.
General listing/analysis has options:
List data file format for listing model data; can be PDB or mmCIF.
Show warning on loading If set on this will show warning messages on loading; otherwise the messages can be seen **Structure definition**->**Display load warnings** from the model icon menu.
Animation See Animation

Surfaces

See Surfaces.
Surface drawing style See Surface drawing style preferences
Electrostatic surface colours See Electrostatic Potential Surfaces and Colouring by Atom/Residue Property.

Maps

Map drawing style See Map drawing style preferences

Display

User interface Controls the appearance of the user interface:
Label display columns if on then each column of the display table is labelled for each display object type, e.g. '*Selection*' or '*Colour*'.
Show model number Each model is given a unique number. If this option is on the number is shown in the display table and can be cross-referenced to atom labels etc..
Show menu bar in graphics window Can be toggled off to remove menu bar from main graphics window.
Interface font size can be set small, medium or large. The full font details are defined in $CCP4MG/ccp4i/etc/linux/configure.def or $CCP4MG/ccp4i/etc/windows/configure.def
Colours See Colours.
Fonts There are four different types of text in the program: atom labels, picked atom labels, annotation and legend. Each type can have a different default font; annotation and legend objects have a **Font..** option on the icon menu to override the default. The font names listed in the interface are those fonts found on your system. Font names that begin with a capital letter are anti-aliased fonts which will usually be better quality but we have known them to be handled badly (no text is drawn) on some hardware.
View menu preferences See View Menu
Lighting See Lighting

Images and Movies

See Images.
Screenshots, Postscript See Screenshot preferences
Movie compile See Movie compile

Behaviour

Keyboard bindings See Keyboard shortcuts.
Mouse bindings See Mouse bindings.
Recentering You can recentre the display by picking an atom or choosing **Recentre** from the Tools menu or from the icon menu of an individual object.
Glide to new centre By default the view will glide rather than jump to the new centre.
Reset the radius of display When recentering on an atom or residue the program will usually adjust the zoom to focus in on the selected atom(s).
Recentering radius The recentering radius can be adjusted, note that the display is immediately set to the selected radius for you to see what it looks like.
RocknRoll The display can be set to continuously rock or roll using the **RocknRoll** option on the **View** menu.

Tools

Browser/web tools See Web Tools

Applications

Geometry See Geometry.
Run Coot Requires the full path name for the Coot setup script.

Session options

Restore status on restart This is on by default and the program status is restored to the last exit status. See Program Status.
Backup data files on exit By default off. If switched on then all currently used data files will be saved. See backing up data files.

CCP4 Molecular Graphics Documentation

View Menu

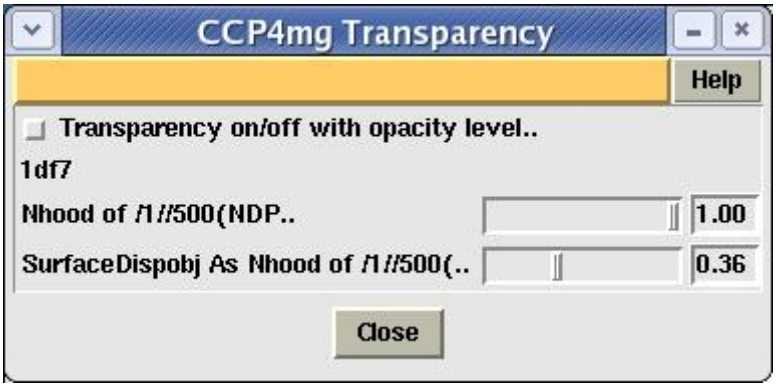
Documentation Contents | On-line Documentation | Tutorials | CCP4mg Home

Contents

The View pull-down menu options

Preferences for the tools on this menu can be found in the **Preferences** window (on the Tools pull-down menu), in the **Display** folder click **View menu preferences**.

Stereo can be toggled on/off. The stereo mode can be selected in the Preferences. The stereo angle can be set; the default value (for CCP4mg 1.1) of -6 degrees gives wall-eye stereo or a positive value gives cross-eye stereo.
Show axes Show the axes in top right corner of display window
Depth queue fog Fogging distant objects creates the illusion of depth. The fog density is graded between the front and back clipping planes and the density can be altered by opening the Clip plane/depth cue slider described below. Also the default of applying fogging to text labels can be switched off in the Preferences window.
Clip planes are two planes perpendicular to the z-direction (i.e. the planes are parallel to the screen); only objects between the front and back clipping plane are displayed. Limiting the display in the z-direction simplifies the picture when you are trying to focus on a small area. The clip plane separation is controlled by moving the cursor up-down with the *h*-key and th middle mouse button depressed. It can also be controlled by..
Clip plane/depth cue slider If toggled on a panel is opened at the bottom of the main graphics window with sliders controlling the clip plane separation and fog thickness.
Background colour can be selected from the defined colours.
Lighting style See lighting.
Set transparency A window like this is shown:



All of the display objects are listed and initially have opacity set to 1.0 in a range of 0.0-1.0 (i.e. they are totally opaque) - setting this to 0.1-0.4 will make objects transparent. Transparency is not applied until the **Transparency on/off** button at the top of the window is toggled on. Note

that drawing transparent objects is slow and proportional to the total displayed and not just the size of the transparent object.
Compact display table Toggling this on will remove the display objects from the display table so only the data objects are listed. Clicking on the data object line in the table will toggle the listing of its display objects.
View from Reorients the display to view from left,right,top,bottom or back.

Rotate/translate/align view

Save view or **Restore view** to file, similar to saving the program status.
Align view on click this button after selecting a plane from the menu.
Sliders to set specific translation, rotation and zoom. Click the **Apply** button to apply.

RocknRoll

The rock and roll parameters can be changed in the **Preferences** window in the **Behaviour** folder.

Rock Continuously rocks the display until you click the **Stop rocking** button at the top of the main graphics window or the display table. The rock axis, time period and angle can be set and the roll axis and roll rate can be set in the Preferences.
Roll Continuously rotates the display until you click the **Stop rolling** button at the top of the main graphics window or the display table. The rotation axis and the roll rate (the angle (in degrees) that the orientation is changed for each step can be set in the Preferences.

CCP4 Molecular Graphics Documentation

Projects

Documentation Contents | On-line Documentation | Tutorials | CCP4mg Home

Contents

Introduction

The interface to the CCP4 program suite, CCP4i, has the concept of projects to help users organise their data. It is expected that users will keep all data relating to one project in one directory. Each project in CCP4i has a short project name and an associated directory. CCP4i creates a sub-directory called CCP4_DATABASE which contains a database of all tasks performed via the CCP4i interface. There is one 'current open project' and the file browser looks here first and, by default, output files are written here.

There is a master file, *directories.def*, which keeps track of all of the projects; it is usually $HOME/.CCP4/unix/directories.def on Linux and on Windows.

CCP4mg uses the project concept and has access to the same *directories.def* file so it knows about any projects set up in CCP4i. CCP4mg also has a 'current open project' but this does not have to be the same as the CCP4i current open project. The **Project** pull-down menu has the option to **Edit Projects** (see below); shows the current open project at the top of the list in italics and lists all projects for you to select a new current open project. In the CCP4mg file browser looks first in the current project directory and has a line labelled **Go to directory** with a menu listing all other project directories and directory aliases (see below) to navigate straight to them.

Project Directory Editor

Clicking **Edit projects** on the **Projects** pull-down menu will open a window in which you can define new projects. There is a list of projects with an **Add project** button at the bottom - click on this to add another line to the list. In the new line you should enter a one-word name for the project and the name of the directory associated with the project. You can use the **Browse** button to open a file browser to select the directory. If the directory does not exist then it will be created. A sub-directory called CCP4_DATABASE will also be created in the project directory.

The **Edit list** button beneath the list of projects can be used to remove projects from the list. If you pick **Delete selected item** then you must click on a line in the list with the *Shift* key help down and using the right mouse button. Note that this will just remove the project from the list; it will not delete the project directory.

There is an option beneath the list of projects to select the current open project and also a list of directory aliases. These are one word aliases for commonly used directories that will appear in the file browser **Go to directory** menu but they are not projects.

CCP4 Molecular Graphics Documentation

Reading Coordinate Files and Atom Typing

Documentation Contents | On-line Documentation | Tutorials | CCP4mg Home

Contents

See also Coordinate Model Interface

Downloading Coordinate Files

CCP4mg will download and load coordinate files from the coordinate deposition sites (RCSB, EBI or PDBj). Access this via the **Download coordinate file** option on the **File** pull-down menu. The Download window allows you to choose a download site has an option to **View server web page**. You must enter the four-letter code for the file required. By default the downloaded file will be written to your current project directory,; alternatively you can enter a name for the new file in the Download window.

The choice of browser and URLs for web pages and downloads can be changed in the **Preferences** window: choose **Browser/web tools** from the **Tools** folder.

Picture Wizard representation styles

The coordinate file browser has options at the top of the window to select a representation style from a menu listing all of the Picture Wizard styles. The default is to display *All atoms* as bonds but the alternative styles can be viewed in the Picture Wizard interface. For most representation styles several display objects will be created as the model is loaded; some of these (for example an object displaying solvent) may be not visible. By default all of the model will be drawn in the selected style but if the ***Show Picture Wizard selection window*** option is set on then another window appears with options to select chains, monomers etc.. If no selections are entered in this window then all of the model will be drawn in the representation style. For a few styles the Picture Wizard selection window will always appear (for example for defining an interface) and then selections must be entered. It is possible to create your own representation styles.

Reading the Coordinate File

CCP4mg reads model coordinates from PDB or mmCIF format files. The program does some basic analysis of the structure and reports any problems. The report can later be accessed from the ***Structure definition*** sub-menu of the model icon menu. Most of the causes of

warnings will not prevent the molecule from being displayed.

Coordinate files should follow the file format guide from www.wwpdb.org/documentation/format30/index.html. CCP4mg also expects that each atom in the file will have a unique identifier where the identifier is composed of: chain id, residue sequence number, residue insertion code, atom name and alternate location indicator. CCP4mg does not use the atom serial number or the segment identifier. If the element symbol is not present then CCP4mg attempts to deduce the element type from the atom name.

CCP4mg attempts to cross-reference the residues and atoms in the structure to a library of monomers. The monomer library provides:

- The expected inter-atomic bonding - used to determine which atoms to connect
- The 'energy type' of the atom - used to lookup atomic radii and hydrogen bonding capability
- The ideal geometry of the monomer
- The residue type (e.g. nucleic acid or solvent)

The protocol for matching a residue in the structure to a monomer library residue is to first try to match the residue name and then to match the atoms.
To match the residue name..

- If the user has specified a synonym for the residue name and the synonym is in the monomer library
- or the residue name matches a monomer in the library
- or the residue name matches a synonym in the *ccp4mg/data/mon_lib_list* file
- or the residue is a nucleic acid and, dependent on the presence of O2' (or O2*) is matched to Xr (RNA) or Xd (DNA).

To match the atoms..

- Attempt to match the atom names of all atoms in the residue- it does not matter if some atoms in the monomer are not matched.
- or the bonding within the residue is deduced on the basis of inter-atomic distances and a graph matching algorithm is used to match against the monomer library residue on the basis of atom element type and bonding

If no match is found then then is a warning message is added to the file loading warning messages and the atoms are assigned some generic type based on their element type. This is generally not a problem - the most likely problem would be misinterpreting the hydrogen bonding capability of oxygen or nitrogen atoms.

Monomer library files

The ideal geometry information for a monomer can be taken from one of two sources:
1) The REFMAC5 monomer library (Vagin et.al.) which contains the structure definitions used in REFMAC5 refinement. The data is in the directory **ccp4mg/data/monomer_library**. Each monomer in the library is recorded in a separate file called *MON*.cif (where *MON* is the name of the monomer). These files are organised by the initial letter in their name into sub-directories named ***a*** to ***z***. You can view the content of a monomer library file using the **List monomer definition** option from the **Tools** pull-down menu.
2) The user can provide a monomer library file containing definitions of novel structures which are not in the database. The file can be generated using LibCheck via the CCP4i Sketcher interface (see $CCP4/ccp4i/help/modules/sketcher.html and $CCP4/html/intro_mon_lib.html) Novel monomer library files or modifications to the standard files should be placed in the

directory *your_home_directory*/.CCP4MG/data/monomer_library. CCP4mg will use monomer definitions in this directory in preference to those the standard distribution library.

PDB File Remediation - Summer 2007

The wwPDB has completed an exercise of cleaning up and standardising residue and atom names to IUPAC conventions. This creates a problem for programs such as CCP4mg supporting data files that follow two distinct conventions. For monomers that have suffered an atom name change in the remediation the REFMAC5 monomer library has been modified to include an *alt_atom_id* attribute for each atom. This is the OLD atom name and the *atom_id* is the remediated atom name. Note that these changes mean that pre-CCP4mg1.1 monomer library is incompatible with CCP4mg1.1 and subsequent program versions. The program will attempt to match atom names in a residue using both the *atom_id* and *alt_atom_id* but will not consider a mix of the two.

Note that for old style files with residue names 'A','C','G' the program checks for the presence of 'O2*' to decide whether a residue is DNA or RNA and then uses the monomer library files named 'AD','CD','GD' or 'AR','CR','GR' as appropriate.

The **List monomers** option on the **Tools** menu can be used to view the monomer library files.

Atom energy type and properties

CCP4mg uses the same atom energy types as REFMAC5. These have code names such as 'CT' or 'NR56' and are used to look up properties such as the atom hydrogen bonding capability or charge for surface potential calculations. The data is in the file
ccp4mg/data/ener_lib.cif
If you wish to modify this file you should make a copy to
your_home_directory/.CCP4MG/data/ener_lib.cif
If this file exists then CCP4mg will use it in preference to the standard distribution file. Beware that the ener_lib.cif used with CCP4mg may not be identical to that used by REFMAC5 - particularly, at the time of writing, it has additional information on charges for surface potential calculations.

Residue name synonyms

Sometimes the name of a residue used in a coordinate file does not match the name used in the REFMAC5 monomer library. To help handle this the monomer library has a list of commonly used synonyms (i.e. alternative names) for residues in the file *ccp4mg/data/mon_lib_list* (look for 'synonym' to find the list). If you have a coordinate file with an unrecognised residue name you can either edit the mon_lib_list file to include the alternative name or you can use the **Residue type assignment** interface (on the **Structure definition** sub-menu of the model icon menu) to enter a synonym for a residue name. Additions to the *mon_lib_list* file will apply to all models loaded into CCP4mg but synonyms entered in the **Residue type assignment** interface will apply to just the one model.

Other data taken from files in the monomer library are tabulated below.

Data	CIF catagory	File
Monomer name synonyms	_chem_comp_synonym	mon_lib_list
Inter-residue links	_chem_link	mon_lib_list
Properties of atoms	_lib_atom	ener_lib.cif

Residue types

The type of a residue (e.g. nucleic acid, solvent etc.) is necessary information for several program features, for example

- the atom selection menu
- drawing ribbons through either amino acid or nucleic acid chains

The residue type is a function of the residue name (e.g. residue name 'ALA' implies type amino acid and residue name 'H2O' implies type solvent). The residue type is included in the REFMAC5 monomer library definition (see above). CCP4mg recognises the types:

CC4mg residue type	REFMAC5 monomer library type(s)
amino acid	L-peptide,D-peptide,peptide
nucleic acid	DNA,RNA,DNA/RNA
saccharide	saccharide,pyranose,D-saccharid,L-saccharid
solvent	solvent
solute	non-polymer
monomer	non-polymer

If an imported coordinate file contains a residue which is not included in the REFMAC5 monomer library then it will not be properly recognised for atom selection. Another potential problem is that when protein or nucleic acid backbone is drawn as a ribbon there will be a gap in the ribbon at the site of an unknown residue type. One solution to this problem is to generate a monomer library file but a quicker option is to use the **Residue type assignment** option from the **Structure definition** sub-menu on the model icon menu. In the residue type assignment window you should type in the name of the residue and choose a residue type from the menu. The definitions entered here will override any definitions taken from the monomer library and will be saved in the *.ccp4mg* file to be restored in subsequent program runs.

The **Residue type assignment** interface can also be used to enter a residue name synonym.

.

Bonds

CCP4mg deems atoms to be bonded if:
The residue and atoms are matched to a monomer library file and the atoms are listed as bonded in that file.
The residue does not match any in the monomer library and the interatomic distance is less than the sum of the bonding radius for the two atoms multiplied by a safety factor of 1.2. The bonding radius is dependent on the atom element type and is taken from the file *ccp4mg/data/elements.cif*.
The atoms in different residues are closer than 2.4A and are listed as possible inter-residue links in the file *ccp4mg/data/mon_lib_list*.

Bonds can be added or deleted using the **Add/delete bonds** option on the submenu of the model icon menu (the dot next to the model name on the Display table). Note also that if you want to show bonds or other inter-atomic interactions the Vectors graphical object may be useful.

References

A. A. Vagin, R. A. Steiner, A. A. Lebedev, L. Potterton, S. McNicholas, F. Long and G. N. Murshudov, (2004). REFMAC5 dictionary: organization of prior chemical knowledge and guidelines for its use. *Acta Cryst.*,**D60**, 2184-2195

CCP4 Molecular Graphics Documentation

Coordinate Model Interface

Documentation Contents | On-line Documentation | Tutorials | CCP4mg Home

Contents

See also:

Loading model data

Coordinate data can be loaded from PDB or mmCIF files or it can be automatically downloaded from one of the deposition sites. As the data is loaded atom typing and secondary structure analysis are performed. The loaded model is shown in the Display Table with the interface described below.

Dot menu functionality

Hide/Show
: Hide all display objects for this data object or show all the display objects that are not themselves hidden. Note that you can quickly toggle hide/show by clicking on the dot with the right mouse button.

Centre on
: Put the currently displayed atoms in the centre of the screen and fill the screen with them.

File Save/Restore ->Save data to file
: Saves the current coordinates to a PDB or mmCIF file - you are prompted for the file name. Any changes such as addition or deletion of atoms or structure transformation will be reflected in the saved file.

File Save/Restore ->Restore data from file
: Reread the original data file - this effectively undoes any edit or transformation applied to the data.

Add model display object
: Add a display object that can represent a selected set of atoms in any colour scheme and a variety of display styles (e.g. spheres, ribbons).

Add display object
: Add a display object to represent **Surface**, **Hydrogen bonds**, **Close contacts**, **Annotation** (text labels on the model) or **Temp factor spheroids**.

Display same as

Set up one or more display objects to represent this model the same as either another loaded model or as a model in a status file. If you select a status file then you will be presented with a list of the models in that status file to choose one of them. Note that if, for example, the source data has particular ligand selected for one display object and this data does not contain an identically named ligand then nothing will be selected for the equivalent display object in this model. If you want to display many models in the same style the Picture Wizard and its user generated templates may be more useful.

List data

List either all atoms or just the currently visible atoms in either PDB or mmCIF format (set in the **Preferences**, **General listing/analysis** in the **Model analysis** folder). The coordinate data is prefaces by the file header information from the original data file and any editing or model transformation is reflected in the listing. The listing is in a window which has options to **Save to file** and **Find** in the text. Note that the combination of saving only visible atoms and saving to file can be used to write out a selected subset of atoms. Note also that clicking on an ATOM or HETATM line will cause the atom to be labelled in the main display, and double clicking will centre on the selected atom.

Structure definition - Display load warnings

By default less serious warning messages from data loading are not show (but see **Preferences**, **General listing/analysis** in the **Model analysis** folder). This will show the load warning - see atom typing for more background.

Structure definition - Residue type assignment

The information in the monomer library is used to categorise residue types (e.g. ALA is 'peptide', H2O is 'solvent') that is used in atom selection. This interface allows overriding the typing and providing a synonym if a monomer in your data has a different name to the definition in the monomer library. Note that a right mouse click on a line in the interface table will present a popup menu with additional table editing options.

Structure definition - List secondary structure

A new window is opened with a listing of the secondary structure assignment for the model. There is an option to save the data to a file.

Structure definition - Edit secondary structure

You may specify the secondary structure type for one residue or a range of residues to override the calculated secondary structure. Note:

This information is saved to the .ccp4mg file and will be reinstated on reloaded the same data file.

Right mouse clicking on a line in the interface table will present a popup menu with additional table editing options or to centre on a selected residue.

If you just want to customise colouring by secondary structure that can be achieved by using the **Edit colour scheme..** option from the colour menu (see tutorial).

Structure definition - (UN)Load DSSP secondary str.

Load the secondary structure assignment output by the DSSP program (http://swift.cmbi.ru.nl/gv/dssp/). The DSSP filename will be remembered and loaded automatically if the program is closed and restarted. While DSSP data is loaded there is an option to unload it.

Structure definition - Add/delete bonds

The displayed bonds are usually derived from the monomer library and distance criteria - this interface will override the assigned bonds and the information is saved to the .ccp4mg file and will be reinstated on reloaded the same data file. . The interface is a table with the option to **Add/Delete** and then select two atoms. Note:

You can paste an atom name into the interface by left-mouse clicking on a displayed atom and them middle-mouse clicking in the atom name text widget.

A right mouse click on a line in the interface table will present a popup menu with additional table editing options.

You could represent unusual bonds (e.g. metal coordination) using vectors.

Structure definition - List charge
List the currently assigned charges for electrostatic calculations.

Transform coordinates
See here.

Generate symmetry mates
See here.

Animation
See here

Clone
Load the same coordinate data file again.

Delete
Delete the data object. Note that even if you have edited the structure you will not be prompted to save it.

Model Display Object

A new model display object is created by the **Add model display object** option on the model dot menu. The display object icon menu (click ) has options:

Hide/Show
Hide or show the display object. If the object is hidden the icon is greyed. Toggle hide/show by clicking on the icon with the right mouse button.

Centre on
Put the display object in the centre of the screen and fill the screen with it.

Add colour legend
Create a legend object with the a legend for the current colouring scheme. Note that the legend object can be moved or edited.

Label atoms
There are options to label no atoms, all atoms, or one atom per one, five or ten residues or SSE or chain termini. There is also a **User defined..** option which gives access to the usual atom selection menu to make a customised selection of atoms to label. If the **One atom per res** option is on then only one atom in the customised selection of residues is labelled. The labelled atom is either a CA or the first atom in the residue. The default for this options set in the Preferences **Atom labels** in the **Model display** folder.

Label text
There is a list of items which may be included in the atom label. The default can be set in the Preferences window **Atom labels** in the **Model display** folder.

Clone
Create a new model display object with the same selection, colour and style as the current display object.

List selected atoms
List the atoms selected in the display object in PDB or mmCIF format (as chosen in the Preferences **General listing/analysis** in the **Model analysis** folder. From the listing window the data can be saved to file.

Molecule drawing style
By default the drawing parameters such as helix width or sphere radius are controlled by the Preferences **Drawing style** in the **Model display** folder. This opens a window with drawing style options to be customised for this display object only. See Drawing Styles for examples of different styles.

Delete
Delete this display object

Model Display Object Selection Menu

Clicking on the button in the *Atom selection* column gives the Atom Selection menu.

Colouring by Atom/Residue Property

The *Colour Scheme* menu has options for *Atom properties* and *Residue properties* including **Temperature factor**, **Occupancy**, **Charge** and **Solvent Accessibility**. There is a ***Colour legend*** option on the display object icon menu (the dot next to the model name) which displays a legend. The colour coding can be changed in the **Preferences** window, look in the **Model colours** folder. In this colour preferences window the defined parameters are

Below/Above normal range *colour*	Colour for atoms with data values below or above the defined ranges
Data value	The lower range value
Colour	Colour for atoms with the data value *Range value*
Add range/colour button	Add an extra line to the table.
Right mouse click on line in table	Options to *Delete range/colour* or *Insert above*
Interpolate	Three modes for colouring (see below)
direction (only for *around colour wheel* interpolation)	direction round the colour wheel

The colour wheel is shown left. By default colour interpolation is applied in *around colour wheel* mode. This means that an atom/residue with a data value between two of the range values in the table will be given a colour which is a mixture of the two appropriate colours with the path from one colour to the other following an arc around the colour wheel. The alternative interpolation mode is *between RGB values* which takes a straight line path across the colour wheel. This latter mode looks better for colour schemes include a white middle state e.g. red-white-blue.

If colour interpolation is switched off then all atoms/residues with data values between one range value and the next highest range value are given the same colour. In this mode the colour for *Above normal range* is not used - all atoms/residues with data values above the highest range value are given the colour for that range.

Model Display Object Colour Menu

Clicking on the button in the *Colour scheme* column gives the atom colouring menu.

Atom type
: Colour according to the atom element type. The colour scheme is defined in Preferences **Atom type** in the **Model colours** folder.

One colour
: A sub-menu lists possible colours and has a option **New colour** to open the interface to define a new colour to add to the list of options.

By chain
: Colour each peptide or nucleic acid chain a different colour. Solvent, monomers etc. are coloured grey.

Main/side chain
: Protein main chain and side chain are coloured differently. Non-protein is coloured grey.

Secondary structure
: The types of secondary structure are coloured differently. The colour scheme is defined

in Preferences **Secondary structure** in the **Model colours** folder.

Blend thru model

This opens a window with options to customise the colouring but just hitting the **Apply** button with give the default blend thru model colouring. You can choose whether to colour peptide, nucleic acid or both and whether to colour each residue or each SSE the same. The interface then consists of a table to select chains (or ranges of residues) and colours for the start and end of the chain (or range). Blending from start to end colour can go clockwise or anticlockwise around the colour wheel. The **Add range** button will add another row to the table and right-mouse clicking on any row in the table will give a menu with options to **Delete range**.

Residue property - Residue type

Colours protein and nucleic acid residues according to their type. The colour scheme is defined in Preferences **Residue type** in the **Model colours** folder.

Residue property - Solvent accessibility

>Colours according to the sum of the solvent accessibility over all non-hydrogen atoms in the residue. There is an option on the display icon menu to **List solvent accessibility**. The colour scheme is defined in Preferences **Residue solvent accessibility** in the **Model colours** folder.

Residue property - Area buried by..

On selecting this option another small window appears with a menu to select one other model display object; the surface area buried by contact with this object will be calculated and displayed using the colouring scheme defined in Preferences **Residue buried area** in the **Model colours** folder. There is an option on the display icon menu to **List buried area**.

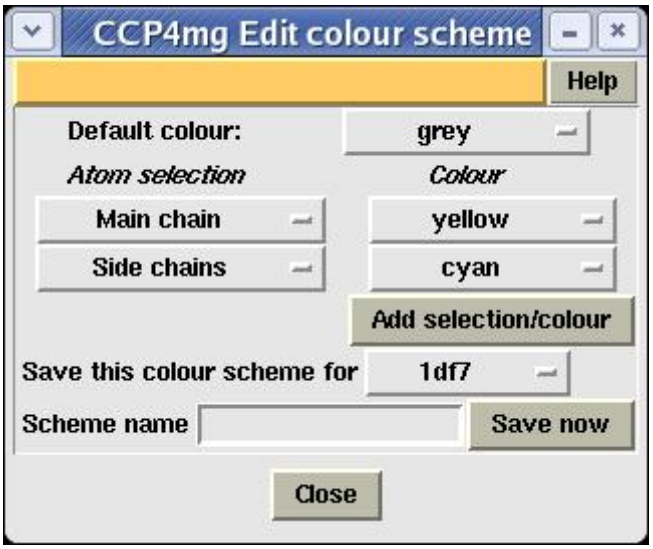
Atom property..

There is a submenu with options **Temperature factor**, **Occupancy**,**Alternate location**, **Charge** and **Solvent Accessibility** and **Buried area**. The colour scheme for each of these is defined in Preferences in the **Model colours** folder. Note that atoms can also be labelled with temperature factor, occupancy and change.

Non-carbon by atom type

If this option is checked on then carbon atoms will be coloured by the scheme selected from the rest of the interface and the non-carbon atoms will be coloured by their atom type.

Edit colour scheme



The rules for the current colour scheme are presented in a new window. If the current colour scheme is one of the colour by atom or residue property options or blend thru model then the rules can not be edited. The window (shown left) specifies a default colour and then a list of rules which are a selection criteria and a colour. The selection criteria button accesses the usual selection menu to change the selection. The colour menu has options to **Edit colour** or add a **New colour** which use the Colour Preferences interface. Clicking the **Add selection/colour** button will add a line to the bottom of the table; a right mouse click on a line in the table with give a popup menu with options to **Delete selection/colour** and **Insert above**. Note that the order of the rules may be significant as later rules override previous ones. The edited colour scheme

can be saved by entering a name for the scheme and hitting the **Save now** button. Note that the scheme can be saved so that it is accessible for only the current model or for all models. The information is saved to the model definition file.

Stick colour

Choose the colour of the stick in ball-and-stick representation. The default is the **Same as atom**.

Model Display Object Style Menu

It is quicker to try the options than to read about them - so do that!

The final option **Join display objects** is on by default and means that when a model is split into more than one display object, such as a main chain ribbon and side chain cylinders, the bonds connecting the two objects will be drawn.

Temperature Factor Spheroids

The temperature factor spheroid icon menu includes the options:

Scale

Enter a number to scale the size of the spheroids.

The selection menu lists the model display objects; when one of these is selected temperature factor spheroids will be drawn over all of the atoms in that object.

The display style menu has options to draw the spheroids as **Axes**, **Solid** or **Solid&Axes**. The quality of the rendering in Solid mode is controlled by the Preference **Render quality** in the Display folder.

CCP4 Molecular Graphics Documentation
Atom Selection

Documentation Contents | On-line Documentation | Tutorials | CCP4mg Home

Contents

See also the Selection tutorial with examples of using the selection interface.

The Interface

An atom selection menu is opened from the Display Table by clicking the *Atom selection* button for a variety of model display objects. The menu is also used elsewhere usually in a cut-down form. The input from the atom selection interface is converted into a text string selection command. The selection command can be editied directly via the **Enter selection** or **Interactive selection**->**Advanced** options. This command is interpreted by the program to select a set of atoms.

The options on the atom selection menu are customised for the data; for example the chains and monomers in the structure will be listed where appropriate. Several options on the selection menu have pull-right sub-menus for selecting one or more items. For example for *Monomers* the individual items are listed with check-boxes but there are also commands at the top of the menu ('All' and 'None' in this example) to select multiple items. Clicking the dashed line at the top of the menu will put the menu into a separate window (as shown left) that remains on screen until it is deleted.

The selection menu options are listed below; the menu is divided by separators which indicate how the menu options can be combined. For example if you select *CA trace* and *..of chain .. A* then the CA trace of chain A will be selected. But if you select *..of chain .. A* and then *Residue type..ALA* the residue type selection will override the chain selection and all ALA residues, whatever chain they are in, will be displayed. If you want to combine selections in a way thay is not accomodated on the menu then use the **Interactive selection**->**Advanced** option.

All atoms, All non-solvent atoms or No atoms

These options are only combined with the selection of models.

Exclude hydrogen

If set then hydrogen atoms are excluded from whatever else is selected.

Models

This is only available if the coordinate file contains multiple NMR models or if symmetry mates, which are created as separate models, have been generated. This selection will be combined with any of the other selections.

Not..

If this is set then all atoms which do not satisfy the other specified selection criteria are selected.

All peptide, Main chain, CA trace, Side chains

Select peptide by picking one of the radio buttons. These options are combined with the *..of chains..*, *..of residue range(s)..*, *..of secondary structure..* and *..and monomers..* selections.

..of residue range(s)..

This tool is for selecting multiple individual residues, ranges of residues, or secondary structure elements or loops.

The tool has an interface window that contains a table. Each line of the table defines a first and last residue to give a residue range. If only the first residue is specified then that single residue will be selected. To enter a range you can:

- Type the coordinate id of the first and last residue into the boxes - the format will be something like *A/27* for residue 27 of chain A. You need to click the **Apply** button for it to take effect.
- Type the whole range into the first box in a format such as *A/10-40*
- Use the pull-down menus to select the chain and residue - start by selecting the chain from the left-most menu and then the residue
- Use the left-most pull-down menu to select a secondary structure element or loop. The first and last residues in the SSE will be entered in the appropriate boxes.
- 'Cut-and-paste' from the display: click on an atom on the display and then click, with the middle mouse button, in the box for either the first or last residue.

To add or delete lines from the table

- Click on the **Add range** button to add a new line to the table.
- Click on a line with the right mouse button to get a menu with options to **Delete range** or **Insert above**.

..of secondary structure..

Select types of secondary structure. To select specific secondary structure elements use the *..of residue range(s)..* tool.

All nucleic acid and ..of strands..

Select the nucleic acid in the model. This may be limited by selection from the *..of strands* menu that lists the chains containing nucleic acid.

..and monomers..

Select from a submenu listing the monomers. Monomers are residues not recognised as peptide, nucleic acid, solvent or solute; the classification can be edited.

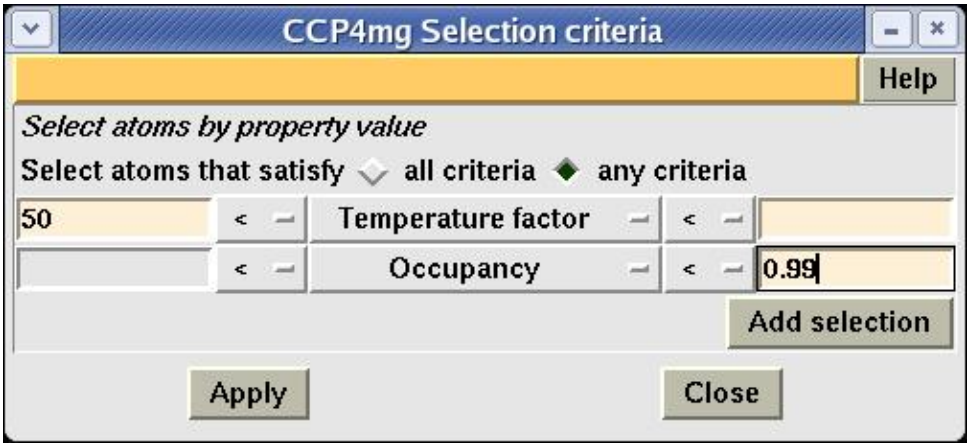
Residue types..

Individual residue types (e.g. ALA or GLN) or groups of residue types (e.g. large or hydrophobic) can be selected.

Atom element types..

Atom element types can be selected.

Property..



The Property selection window has one or more lines with a menu to select a property and text boxes to enter upper and lower limits. The currently supported properties are: temperature factor, occupancy, x, y, z, atom SAS, residue SAS, atom buried area and residue buried area. Note also the menu either side of the property menu to select the inequality. It is acceptable to provide only an upper or lower limit. Clicking the **Add selection** button will add another line to the table; a right mouse click on a line in the table will give a popup menu with options to **Delete selection** and **Insert above**. If you have more than one selection criteria note the option at the top of the window to select atoms that either satisfy all criteria or satisfy any criteria.

If the selection criteria is atom or residue contact area then you will be prompted to select a different model display object for the contact calculation.

Neighbourhood of..

With this tool you can select all atoms or residues within a given distance of some central group such as an atom, residue, chain or model display object.

The first two lines of the neighbourhood interface window are for selecting the central group by either:

- Type the coordinate id into the entry box
- Use the menus to successively select chain, residue and atom. An alternative to selecting the chain and residue is to select a 'monomer'. The monomer menu is the list of residues which are not recognised as amino acids or nucleic acids.
- 'Paste' an atom name into the entry box: click on an atom in the image and then click in the box with the middle mouse button.
- From the 'around selection' menu choose a model display object

Also in the window you can choose to select groups of atoms (e.g. a residue or a side chain) which have one or more atoms close to the central atoms and you can opt to select only those groups which have hydrogen bonds to the central atoms. The criteria for hydrogen bonding are set in the **Preferences** window under the **Model analysis** folder.

If some atom selection has been made before opening the neighbourhood window then then is an option **Only select neighbouring atom in *current selection***. If this is checked on then only the

atoms in the current selection close to the central atoms will be selected.

For coordinate files containing symmetry information there is an option to **find neighbouring atoms in symmetry mates**. If checked on then the program will select atoms in neighbouring symmetry models. Note that the program will extend the loaded molecule to include all atoms in symmetry mates that have atoms close to the central group. This option uses Generate symmetry mates tool on the model icon menu.

Finally you can choose not to include some groups of atoms: by default the central group of atoms is excluded; solvent, solute or peptide atoms may be excluded.

Buried surface..

This tool will find the atoms or residues on the surface of the first set of atoms that are buried by a second set of atoms. This tool is particularly useful for selecting part of a surface over the first set of atoms that contact the second set of atoms.

The first set of atoms are the currently selected set of atoms. If the current selection is 'all atoms' then the first set of atoms is all atoms excluding the second set of atoms. Solvent atoms are also automatically excluded from the selection.

The bottom two lines of the Buried surface window are for selecting the second group of atoms with options similar to those for the Neighbourhood window above.

Features

Select atoms that are part of a chemical feature. Currently supports **Disulphide bridges**.

Enter selection

You can use this tool to view or edit the selection command.

Save current selection..,Restore selection and Delete selection

Save current selection will save the current selection command with the option for it to be accessible for

- The present model: the selection is saved to a file called *model_name*.ccp4mg and is only available to use with present model.
- **All models**: the selection is saved to a file called $HOME/.CCP4MG/generic.ccp4mg and is available for all models.

You should enter a short one-word name for the selection. After you have saved a selection extra options appear at the bottom of the selection menu:**Restore selection** and **Delete selection**. These options have pull-right menus that list the saved selections. Selections which have been saved for all models appear in the list after a separator.

Simple interactive selection

Switch this tool on from **Interactive Selection** at the bottom of the **Atom Selection** menu.

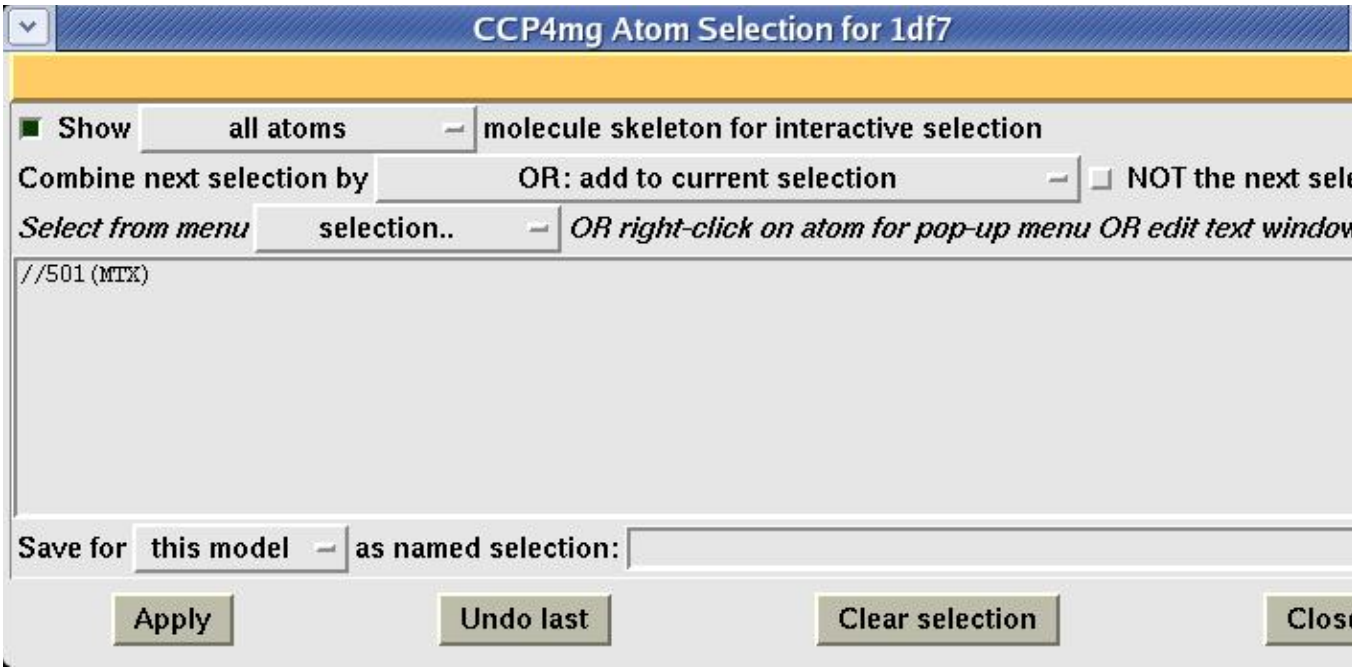
This tool does not have an interface window - you click on an atom in the model and, from a pop-up menu, choose to add or remove atoms, side/main chain, residues, SSEs, neighbourhoods or residue ranges.

When you select this tool the display changes to show all of the atoms in the model (otherwise you could not click on them). Those atoms which are not currently selected are shown with thin bonds. You might want to set the display of the selected atoms to **Fat bonds** or **Cylinders** to emphasise the difference from the unselected atoms. Note that the display of unselected atoms can be switched off or changed to displaying only a CA trace in the advanced interactive selection window.

To make a selection click on an atom with the right mouse button - this brings up a menu with various options. Note that there is an undo option if you make a mistake.

The Selection tutorial has an example of using the simple selection tool.

Advanced interactive selection



This tool has several different options for setting up a more complex selection including those described for the simple interactive selection.

In the centre of the window is a text window showing the current selection command. In an attempt to make this more readable it is automatically split over several lines. While the interactive selection window is open you can:

- Edit the selection command text - see Selection commands. Click the **Apply** button for changes to take effect.
- Select from the atom selection menu in the interactive selection window and the option will be appended to the selection command.
- Click on an atom with the right mouse to pop-up a menu which includes options to select the atom/residue/chain or select atoms in the neighbourhood or select a range of atoms/residues/chains. This is the same facility as in the simple interactive selection.

By default the selection commands that you enter are combined with an 'OR' operator. There is a menu to change this to an 'AND' or 'AND NOT' operator. For more complex combinations you will need to edit the text window to insert curly brace, that is { and }, which are used to group selection commands.

To save the selection you should enter a ***Selection name*** at the bottom of the interactive selection window and click the **Save** button. The name should be one word and less than 20 characters - if it is not then the program will force it to be. If you do not enter a name then the program will create a

name from the text of the selection command.

At the top of the window there is an option to ***Show molecule skeleton*** which will update the display to show all atoms in the model with thin bonds and the currently selected atoms with fat bonds.

The Selection tutorial has some examples of using the advanced selection tool.

Selection Commands

A selection command is a text string which defines a set of selected atoms. All user interface input is converted to a selection command and the command is interpreted to determine which atoms are displayed. By editing the selection command directly you can set up more complex selections than are accessible via the user interface. The selection command can be viewed or edited using the **Enter selection** or **Interactive selection** - **Advanced** tools.

The components of selection commands are

- The **coordinate id** (**CID**) specifies a particular atom, residue or chain and can also specify a range of residues, residue types, group of atoms etc.. For example 'A/27/CA' is the CID for the CA atom of residue 27 in chain A.
- **Aliases** are one word aliases for more complex CIDs. For example the alias 'catrace' will select all atoms called 'CA' in amino acid residues.
- A **Command** is a keyword followed by a variable number of arguments. For example 'neighb cid="A/27" maxd=5.0' will select all atoms in the neighbourhood, that is within 5 Angstrom, of residue 27 of chain A.
- The **Operators** combine CIDs, aliases and commands to create more complex selections. For example the rule '(GLU,ASP)' and 'neighb cid="A/27" maxd=20' combines the CID '(GLU,ASP)' which selects residue types GLU or ASP and the command 'neighb cid="A/27" maxd=20' which will select all residues within 20A of A/27. Combining these components with the 'and' operator will select atoms in GLU or ASP residues within 20A of A/27

Coordinate IDs

A coordinate ID is an text string that has the following format:

```
/mdl/chn/seq(res).ic/atm[elm]:aloc
```

where

Item	*Description*
`mdl`	the number of the model (usually an NMR model)
`chn`	chain ID
`seq`	residue sequence number
`res`	residue name
`ic`	residue insertion code
`atm`	atom name
`elm`	chemical element ID
`aloc`	alternative location indicator

The residue name and atom element type are usually redundant, that is they are not essential to uniquely identify an atom, and do not need to be included when entering a CID. But note that all spaces in the coordinate ID are ignored and this means that certain atom names will be interpreted correctly only if the chemical element name is supplied (compare Calcium `CA[CA]` and Carbon in alpha-position `CA[C]`).

Any item in the coordinate ID may be replaced by a wildcard `"*"`, which means an indefinite value for that item. The wildcard value is automatically implied for any missing item except the chain ID, insertion code and alternative location indicator:

- A missing chain ID is interpreted as an empty ID `""` if there is a slash `"/"` after it. In all other cases, the missing chain ID is interpreted as a wildcard `"*"`.
- A missing insertion code is interpreted as a wildcard `"*"` if the residue sequence number is not specified or set to a wildcard. In all other cases, a missing insertion code is interpreted as `""` ('no insertion code').
- A missing alternative location indicator is defaulted to a wildcard `"*"` if no atom name or chemical element name is given. If any value for atom and/or chemical element name is given , even though a wildcard, then a missing alternative location indicator is defaulted to `""` ('no alternative location indicator').

A coordinate ID may be incomplete. Below are the rules for interpretation of incomplete IDs. Curling brackets `{..}` denote parts of an ID string that may be omitted as a whole:

1. If the ID starts with slash `"/"` then it is interpreted as `/mdl { /chn { /seq(res).ic { /atm[elm]:aloc }}}`
2. If the ID starts with a letter: `chn { /seq(res).ic { /atm[elm]:aloc }}`
3. If ID starts with a (possibly negative) number, bracket `"("` or dot `"."`: `seq(res).ic { /atm[elm]:aloc }`
4. If none of the previous cases apply but the ID string contains square bracket `"["` or colon `":"`, it is interpreted as `atm[elm]:aloc`

Below are examples of valid coordinate IDs:

Coordinate ID	*Description*
`/1/A/33(SER).B/CA[C].A`	model 1, chain A, residue SER with sequence number 33 and insertion code B, C-alpha atom in alternative location A.
`/1/A/*(SER).*`	any SER residue in chain A, model 1.
`/1/A/(SER)`	any SER residue with no insertion code in chain A, model 1.
`/1//(SER)`	any SER residue with no insertion code in chain without a chain ID, model 1.
`/1/A/*.*/CA[C]`	any C-alpha atom with no alternative location indicator in chain A, model 1, in residues with any sequence number and insertion code.
`/1/A/*/CA[C]`	any C-alpha atom with no alternative location indicator in chain A, model 1, in residues with any sequence number and no insertion code.
`A/*/CA`	any C-alpha or Calcium atom with no alternative location indicator in chain A of any model, in residues with any sequence number and no insertion code.

`CA[C]`	any C-alpha atom with no alternative location indicator in any chain, any model, in residues with any sequence number and no insertion code.
`CA`	any atom of chain CA with no alternative location indicator, in any model, in residues with any sequence number and no insertion code.

The following coordinate IDs are incorrect:

```
1.   /A/23/CA[C]      stating with a slash implies that the model number will be given
2.   /1/CG[C]         the residue is not defined
3.   /-15             stating with a slash implies that the model number will be given
4.   */*(*).*/*[*]:*
```

Selection Aliases

Selection aliases are one word aliases for complex CIDs. The predefined selection aliases are specified in the file ccp4mg/python/ui/selection_protocols.py. This needs a nice GUI to allow customisation. Within the program the implementation of selection aliases is different from selection commands but the user need not be concerned by this.

amino_acid	All amino acid residues - defined as a list of residue types
nucleic	All nucleic acid residues - defined as a list of residue types
solvent	All solvent - defined as a list of residue types
solute	All solute - defined as a list of residue types
catrace	All atoms called 'CA' in amino acid residues
peptide_link	The main chain atoms N,C,O,H
main	All main chain atoms in amino acid residues
side	All not main chain atoms in amino acid residues
ca+side	All CA and side chain atoms in amino acid residues
backbone	Atoms C3*,O3*,H3*,C4*,C5*,1H5*,O5*,2H5*,H5T,P,O1P,O2P
sugar_ring	Atoms C1*,H1*,C2*,2H2*,1H2*,C3*,O3*,H3*,C4*,H4*,C5*,1H5*,2H5*,O5*,H5T,O4*,O2*
base	Defined as the atoms which are not backbone or sugar_ring
polar_atoms	Defined as the atoms of element type N,O,P

Selection Commands

More complex selections can be defined by a selection command which consists of a command keyword and a variable number of arguments. The format for commands is:
command [arg1 [arg2 .... [argn]]]
with the arguments in the order given in the descriptions below.
Currently supported commands and their arguments:

neighb	Select atoms, residues or chains within neighbourhood of given object

	cid	An ID for the central atom, residue or chain
	maxd	The maximum distance from the central atom to the neighbours
	group	Select all atoms in a group that have at least one atom within maxd distance. The currently supported groups are: model,chain,residue,main,side,catrace,main_side,CA+side,solvent_monomers.
	hbonded	If this has value 1 then only hydrogen bonded atoms are selected.
	excl	A comma separated list of atoms which will be excluded from the selection: central (the 'central' atoms defined by cid argument), solvent, monomer (all monomers), peptide (main chain atoms excluding CA).
sphere	Select atoms, residues or chains within radius of a given x,y,z coordinate	
	x	X coordinate
	y	Y coordinate
	z	Z coordinate
	radius	The maximum distance from the central point
	type	Should be 'atom','residue' or 'chain' to specify type of objects selected
range	Select all atoms in a range between two given CID	
	start	CID for one or more atoms at start of range
	end	CID for one or more atoms at end of range
all	Select all atoms in the model	
oneatom	Select one atom per residue in the model	
termini	Select the termini residues of chains	

Selection Operators

Complex selection commands are built up from CIDs, aliases and commands connected by the operators

a ***or*** *b*	select all atoms defined by *a* and *b*
a ***and*** *b*	select only the atoms in both *a* and *b*
not *a*	select atoms that are not in *a*
a ***xor*** *b*	select all atoms that are in *a* or *b* but not in both
a ***excl*** *b*	select the atoms that are in *a* but exclude those in *b*. This is equivalent to 'and not'.

The selection components can also be grouped by using curly brace. Note that curly brace are used to be distinct from the brackets used to denote residue types in the CID.

Examples of Selection Commands

amino_acid and not main
uses the aliases ***amino_acid*** and ***main*** to select atoms in amino acids that are not main chain (i.e. side chain)
(GLU,ASP)/(O) and neighb (HIS)/NE2,ND1 5
Select oxygen atoms in glutamic acid or aspartic acid residues within 5A of the ND1 or NE2 of a histidine residue
{ main and {A/10-25 or A/35-40} } or (MTX)
Select the main chain atoms for two given ranges of residues and any residue with residue name MTX

CCP4 Molecular Graphics Documentation

Structure analysis

Documentation Contents | On-line Documentation | Tutorials | CCP4mg Home

Contents

Secondary structure

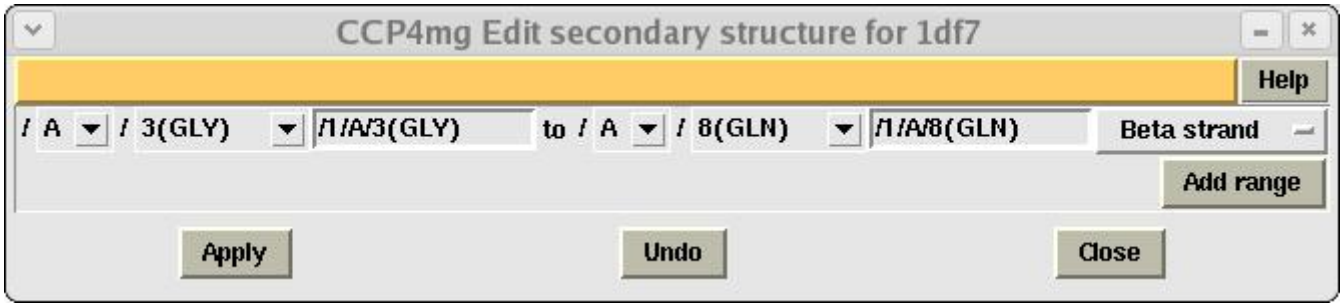
The option to colour by secondary structure is on the ***Colour scheme*** menu for any model containing protein. A ***Colour legend*** option is on the display object icon menu. Also on the display object icon menu, if the object is coloured by secondary structure, is an option to ***List secondary structure***; the listing can then be printed to a file.

The secondary structure is assigned by the method of Kabsch and Sanders(1983). The method first finds all hydrogen bonds between main chain amide and main chain carboxy groups. Then if hbond(*i*,*j*) is a hydrogen bond between the carboxy oxygen of the *i*th residue and the amide group of the *j*th residue the secondary structure types are then defined as follows:

3-turn	If hbond(*i*,*i*+3) exists then residue *i*+1,*i*+2,*i*+3 are a 3-turn
4-turn	If hbond(*i*,*i*+4) exists then residue *i*+1,*i*+2,*i*+3,*i*+4 are a 4-turn
5-turn	If hbond(*i*,*i*+5) exists then residue *i*+1,*i*+2,*i*+3,*i*+4,*i*+5 are a 5-turn
alpha helix	Two or more consecutive 4-turns
beta strand	Two consecutive residues in one strand have hydrogen bond bridges to two consecutive residues in a neighbouring strand.
beta bulge	In two adjascent beta strands, two residues (called 1 and 2) are opposite one residue (called chi) in the other strand

The beta bulge definition is taken from Richardson et al.(1978).

It is possible to override the automatic secondary structure definition: click on the model icon (the dot next to the name) and select **Structure definition** and **Edit secondary structure** to see the window:



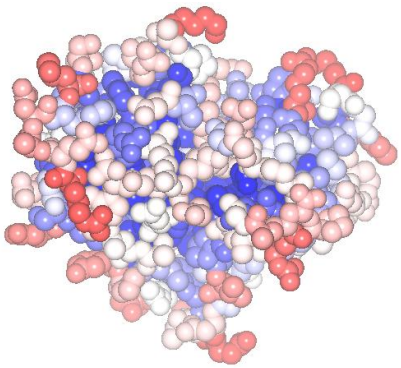
Each line in this window defines three parameters: the first residue in a range, the last residue in a range and the secondary structure to be assigned to that range of residues. Beware that when you run the program in future these changes will be applied automatically.

It is also possible to load the secondary structure assignment output by the DSSP program (http://swift.cmbi.ru.nl/gv/dssp/) using the **Load DSSP secondary str** option on the **Structure definition** menu.

Hints

- The criteria for hydrogen bonds and the option to display beta bulges as a separate class from beta strands can be set in the **Preferences** window (look under **Model analysis**).
- Finding hydrogen bonds is dependent upon recognising a residue as an amino acid and the residue must contain atoms named 'CA', 'N' and 'O'. The program currently needs to be informed that non-standard amino acids are amino acids (*sorry! - needs to be fixed*) using **Residue type assignment** from *Structure definition* on the model icon menu.
- The secondary structure type of a range of residues can be edited using ***Edit secondary structure*** from ***Structure definition*** on the model icon menu.
- To modify the colouring based on secondary structure: first colour by secondary structure and then use the **Edit colour scheme..** option.
- The default secondary structure colour assignment for the whole program can be changed in the **Preferences** window (look in the **Model colours** folder).
- There are **Secondary structure** preferences in the **Model analysis** folder of the Preferences window.

Solvent Accessible Surface Area and Buried Area



The option to colour by solvent accessible surface (SAS) area is on the *Colour scheme* menu (on both **Atom properties** and **Residue properties** sub-menus). A ***Colour legend*** option is on the display object icon menu. Also on the display object icon menu, if the object is coloured by SAS, is an option to **List solvent access** which will list the SAS for each atom and residue,

and totals for all atoms and sets of atoms specified in the Preferences. The listing can written to a file.

The SAS area is the surface area of an atom or residue which is on the surface of the molecule and therefore accessible to solvent. Conceptually the surface of the molecule is defined by rolling a sphere (notionally equivalent to a solvent molecule) over a structure made up atomic spheres with radii of van der Waals radius. The SAS is traced out by the centre of the rolling 'solvent' sphere and so this surface is further out than a surface defined by just van der Waals spheres about atoms. A macromolecular surface has many narrow indentations which are inaccessible to the rolling 'solvent' sphere and so the SAS surface tends to smooth over these narrow indentations.

CCP4mg uses the method of Lee and Richards (similar to the implementation in the CCP4 program areaimol). The SAS is calculated for the whole model with solvent molecules and hydrogen atoms excluded. Only the first of any alternate located atoms are included. The excluded atoms will be coloured in an 'out-of-range' colour.

Buried area is calculated for a set of atoms in the context of a second set of atoms. The buried area for an atom in the first set is the difference between its SAS in the presence of only the first set of atoms and in the presence of both sets of atoms.

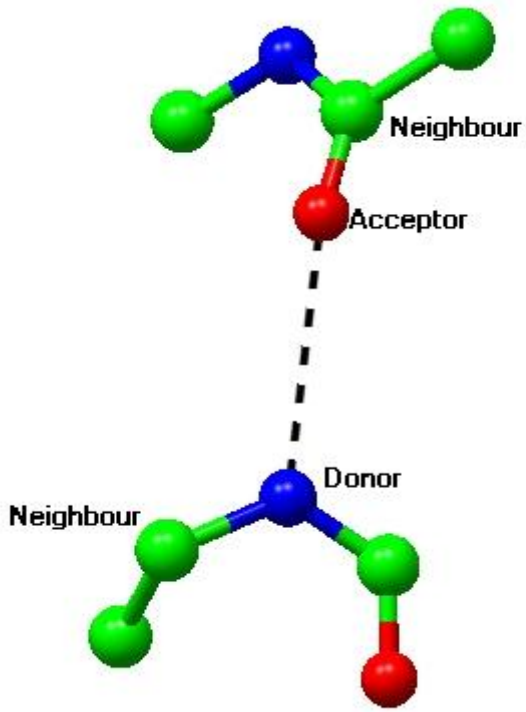
The parameters used in the calculation can be changed in the **Preferences** window (**Model analysis** folder):
The **water radius** is conventionally set to 1.4Å which is most appropriate for comparison of results
The **point density** parameter determines the resolution of the calculation,the default is 1Å, but decreasing this value should improve the result but increase the calculation time.
Exclude solvent By default solvent is automatically excluded from the calculation and will be coloured in the 'below normal range' colour.
List total SAS for selected atoms There are two options to enter a selection command that will be used to select a set of atoms (by default one is 'polar_atoms'). The total SAS for all atoms in these selected set will be written out by the **List solvent access** command.

The default colour scheme can be changed in the **Preferences** window (look under **Model colours**).

Hydrogen Bonds

Hydrogen bonds is one of the options on the ***Add display object..*** sub-menu of the model icon menu. Display Table and usually hydrogen bonds are displayed for all the currently displayed atoms in the model. The first and third column in the Display Table for hydrogen bonds have two menus for selecting two sets of atoms - only the hydrogen bonds between these sets of atoms will be displayed. Note that it is possible to find hydrogen bonds to a different model: the menu in the third column has the option **As display object..** which lists the display objects of the this model and all other loaded models. The hydrogen bond display object icon menu has options to change the style and labelling of the displayed hydrogen bond and options to list or print the hydrogen bonds. The hydrogen bonds can also be written out as a vectors file which can be imported and manipulated as vectors which provides more flexibility in editing and display style which may be useful when creating presentation graphics.

The parameters for calculating hydrogen bonds can be changed in the **Preferences** window (look under **Model analysis**).



There are two sets of criteria for with and without hydrogen atoms defined. The parameters are maximum and minimum donor atom or hydrogen atom to acceptor atom distance, and minimum angles between three atoms. The default criteria are generous and will show hydrogen bonds of poor geometry.

Close contacts

The interface to close contacts is essentially the same as for hydrogen bonds - see above.

The search for close contacts will work in either 'simple' mode, finding all inter-atomic distances between given minimum and maximum values, or in 'van der Waals' mode, finding inter-atomic distances which are between a given minimum and maximum fraction of the sum of the van der Waals radii of the two atoms. The choice of mode and the maximum and minimum values can be changed in the **Preferences** window (look under **Model analysis**). There are also options to label close contacts with the fraction of van der Waals contact distance (rather than absolute distance) and to exclude close hydrogen-bonded atoms.

References

Kabsch and Sanders(1983), Dictionary of Protein Secondary Structure: Pattern Recognition of Hydrogen Bonded and Geometrical Features, *Biopolymers*, **22**, 2577.
J.S. Richardson, E.D. Getzoff and D.C. Richardson (1978). The Beta Bulge: A Common Small Unit of Nonrepetitive Protein Structure. *PNAS*, **75**, 2574-2578.>
B.Lee and F.M.Richards, (1971). *J.Mol.Biol.*, **55**, 379 - 400
E.N. Baker and R.E. Hubbard. Hydrogen bonding in globular proteins. *Prog. Biophys Mol Biol*, **44**:97-179, 1984.

CCP4 Molecular Graphics Documentation

Molecular Surfaces

Documentation Contents | On-line Documentation | Tutorials | CCP4mg Home

Contents

How to Generate a Surface

To generate a molecular surface you must first load the molecule. The molecule will then be listed in the Display Table window with one model display object line showing the atom selection, colour and style for the displayed model. To create a surface you can either:

1) Click on the Display Style menu of model display object and select the ***Surface*** option. A surface will be draw over the selected atoms and the surface colour mirrors the colours of the atoms beneath the surface.
2) Create a new **Surface display object** which allows more options than the simple approach above. To do this click on the icon menu of the model object (the dot next to the molecule name) and select ***Add display object*** and then ***Surface***.

In the display table the first two columns for the surface display object are Atom Selection and Colour, the same as model display objects. The third column is for selection of context atoms. Note that the surface is not drawn immediately; you need to click on the surface icon and select ***Show*** from from menu.
The selection menu is identical to the selection menu for model display objects. The most useful options are probably *All peptide* to create a surface over the protein and *Neighbourhood* which allows you to select just the area of the protein around a given residue or ligand. Beware that if you select a limited number of atoms then the surface will probably wrap around the back of the selected atoms. To prevent this you should set the context selection (in the third column) appropriately. The most useful context selection option, to use when you have selected a limited set of atoms in the protein, is *All peptide*. The surface calculation algorithm will automatically exclude hydrogen atoms and atoms closer than 0.1A. Also if the input set has atoms with more than one alternate location then only the atoms with the first alternate location are used.
The Colour menu has the same options as the model colour menu with the additional options to colour by *Electrostatic potential* and by *Potential from map*. The electrostatic potential is calculated on demand, beware this is a slow process.
The colouring scheme for electrostatic potential surfaces and map surfaces can be changed using the option on the icon menu to ***Change colouring***.

The Surface Display Object Interface

The surface display object icon menu (click on ) has options:

Hide/Show
Hide or show the display object. If the object is hidden the icon is greyed. Toggle hide/show by clicking on the icon with the right mouse button.

Centre on
Put the display object in the centre of the screen and fill the screen with it.

Surface drawing style
Set the drawing style to **Solid**, **Dots** or **Mesh**.

Surface drawing options
See Preferences below.

List surface info
Open window giving some stats.

Add colour legend
Create a legend object with the a legend for the current colouring scheme. Note that the legend object can be moved or edited.

Clone
Create a new surface display object with the same selection, colour and style as the current surface object.

Delete
Delete this surface object

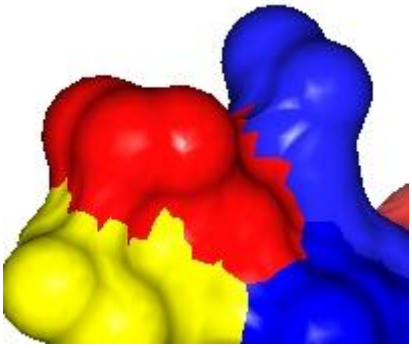
The Atom selection and Colour scheme menus are the same as for model display objects except there is an option for colouring by Electrostatic potential.

The **Display style** menu is actually a menu to select the atoms that provide the context for drawing the surface. By default this contaxt is **Save as selection**. To draw a surface that, for example, covers just the residues of a ligand binding site the selection should be set to the neighbourhood of the ligand and the context should be **All peptide**. This context menu has a subset of the usual selection menu.

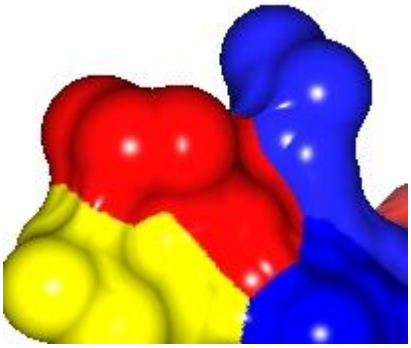
Surface drawing style preferences

The **Preferences** window can be opened from the **Tools** pull-down menu and the folder **Sufaces** contains a **Surface Parameters** option. Alternatively preferences to apply to only one surface can be accessed by clicking **Surface drawing options..** on the surface icon menu(). The variable parameters are:

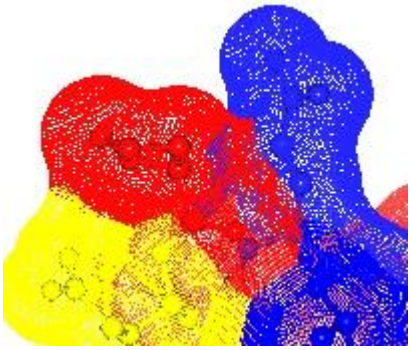
Surface probe radius (0-3) is the radius of the probe atom which is considered to be rolling over the surface of the molecule. A value of 1.5 Ångström, corresponding to a water molecule, is normally used. Note that reducing this value to zero is effectively the same (but less efficient!) as drawing the model with display style Spheres.
Cutoff distance for context atoms After the user has selected a set of atoms to be surfaced and, optionally, a set of context atoms, the algorithm excludes any context atoms more than this cutoff distance from the set of surface atoms. This can speed up the calculation and only in the case on a very large prone radius might it be necessary to increase this value.
Resolution in degrees This is the maximum arc angle for any triangular facet on the surface. The default value of 30degrees gives a reasonably fast calculation time. For quality output images, particularly close-up of the surface, it may help to reduce this to 5-10degrees.
Blend colour borders By default each triangular facet of the surface is assigned to one or other of the underlying atoms and is coloured the same as the underlying atom. If this option is switched on then a triangle may straddle between two atoms so that, if the two atoms are different colours, the colours are blended across the triangle. This helps to reduce the jaggedness of colour edges when seen close up but should be used in conjunction with small values for the resolution.



Resolution=30,
no blending



Resolution=10,
blending borders



Resolution=10,
blending borders
Surface drawn as dots

Maximum dot separation Set the maximum separation of dots on the surface.
Dot size is the size of the dot in pixels. Whether non-integer values are interpreted (by anti-aliasing the edge of the dot) seems to depend on the graphics capability of the machine.

Note that there are limits set on some of the above parameters: the range is indicated on the interface. If you want to change these values contact CCP4mg developers for advice.

Electrostatic Potential Surfaces

The electrostatic potential in CCP4mg is in units of Volts. This is different from GRASP for which potentials are in kT/e (25.6mV, 0.593 kcal/mole at 25Â°C). To convert CCP4mg values to GRASP values multiple by 0.0256.

Charge Assignment

By default the charges used in the calculation are taken from the file **ccp4mg/data/ener_lib.cif**. This file lists attributes such as hydrogen-bonding capability and van Der Waals radius for the atom energy types. The file is in mmCIF format with a loop for the atom energy types (called *_lib_atom*) which has an attribute *surface_potential_charge*. At the time of writing the loop definition and some of the data for nitrogen energy types looks like this

```
loop_
_lib_atom.type
_lib_atom.weight
_lib_atom.hb_type
_lib_atom.vdw_radius
_lib_atom.vdwh_radius
_lib_atom.ion_radius
_lib_atom.element
_lib_atom.valency
_lib_atom.sp
_lib_atom.surface_potential_charge
..
..
..
NC1     14.00670  D     1.550     1.600    1.32   N   3  2  0.0
NH1     14.00670  D     1.550     1.600    1.32   N   3  2  0.0
NC2     14.00670  D     1.550     1.600    1.32   N   3  2  0.5
NH2     14.00670  D     1.550     1.600    1.32   N   3  2  0.0
NC3     14.00670  D     1.550     1.600    1.32   N   3  2  0.0
```

So the energy type *NC2* has the surface potential charge *0.5* but all other listed energy types

have zero surface potential charge.

The atom energy type is defined in the monomer library files which are **ccp4mg/data/monomer_library/*x*/*XXX*.cif** for the residue type ***XXX***. See the atom typing documentation.

Charges specified in the PDB file will over-ride those assigned according to atom energy type. The charge should be in columns 79-80 of the file (i.e. immediately after the atom element type). Atoms in the PDB without assigned charged will be assigned charge according to the atom energy type. If you want to force the charge to be 0.0 then it is necessary to enter a value of **99** in the charge columns of the PDB file.

To see the charge values used in the calculation click on a model display icon menu  and

select **Label text** and click on the display of charges and click off the display of any other parameters. Then select **Label atoms** and **All atoms** from the model display icon menu. Only the atoms with non-zero charge values will be labelled. The charges are assigned to atoms the first time that an electrostatic potential is calculated and before that the labelled charges will reflect the charge values in the PDB file.

It is also possible to list the charged atoms. From the model icon menu select **Structure definition** and **List charges**.

Note that there may also be a partial charge labelled *_chem_comp_atom.partial_charge* in the monomer library files. This data is intended for use in empirical energy calculations (e.g. REFMAC5) and is not used by the electrostatic surface calculation.

Background Theory

The Surface

The surface is a Lee and Richards surface which is derived (conceptually at least) by rolling a ball which is equivalent to a solvent molecule over the surface of the protein and taking the path of the centre of the ball for the surface. The resultant surface glides over small invaginations in the van der Waals surface of the protein and so, more acturately, represents the volume that is inaccessible to a solvent molecule.

Charge

Atom charges are assigned based on a residue lookup table assuming physiological pH, without consideration to synergistic effects. Atom charge is represented as smooth charge density sampled at grid points. Initial grid charges are assigned based on atom charge and atom volume to each grid point inside the atom sphere. The initial discrete charge density is smoothed using charge anti aliasing to reduce grid position artefacts [Bruccoleri, Novotny, Davis, Sharp, 1996].

Dielectric

A solvent envelope of the protein is calculated based on a fast Fourier surface algorithm (insert blurp about the fft algorithm here....). The spatially varying dielectric grid is assigned values between 78 (bulk solvent) and 2 (fully inside the molecule) using volume filtering based this envelope and using harmonic averaging over the nearest nine grid points. Together with charge anti aliasing this approach has been shown to minimize positional artefacts [Bruccoleri,

Novotny, Davis, Sharp, 1996].

Solvent screening

Solvent screening is modelled as simple Debey-Huckel screening parameter grid, calculated as function of ionic strength of bulk solvent and temperature [Nicholls, Honig 1990]. Default values for ionic strength and temperature are given 150mM and RT respectively.

Numerics

Using charge, dielectric and Debey-Huckel grids defined in this way, the linearized Poisson boltzmann equation (LPBE) is solved using a finite difference approach. The PBE finite difference matrix is solved iteratively by optimized successive overrelaxation algorithm (SOR) taking advantage of Chebyshev acceleration and odd-even ordering [Davis, McCammon 1988].

The optimized overrelaxation factor is calculated using the spectral radius for LPB matrix approximated as a function of the matrix dimensions.

Reference

Computational analyses of the surface properties of protein-protein interfaces. J. Gruber, A. Zawaira, R. Saunders, C. P. Barrett and M. E. M. Noble. Acta Cryst. (2007). D63, 50-57

The interpretation of protein structures: estimation of static accessibility. Lee B, Richards FM. (1971). *J Mol Biol* 55(3):379-400.

CCP4 Molecular Graphics Documentation

Symmetry Models

Documentation Contents | On-line Documentation | Tutorials | CCP4mg Home

Contents

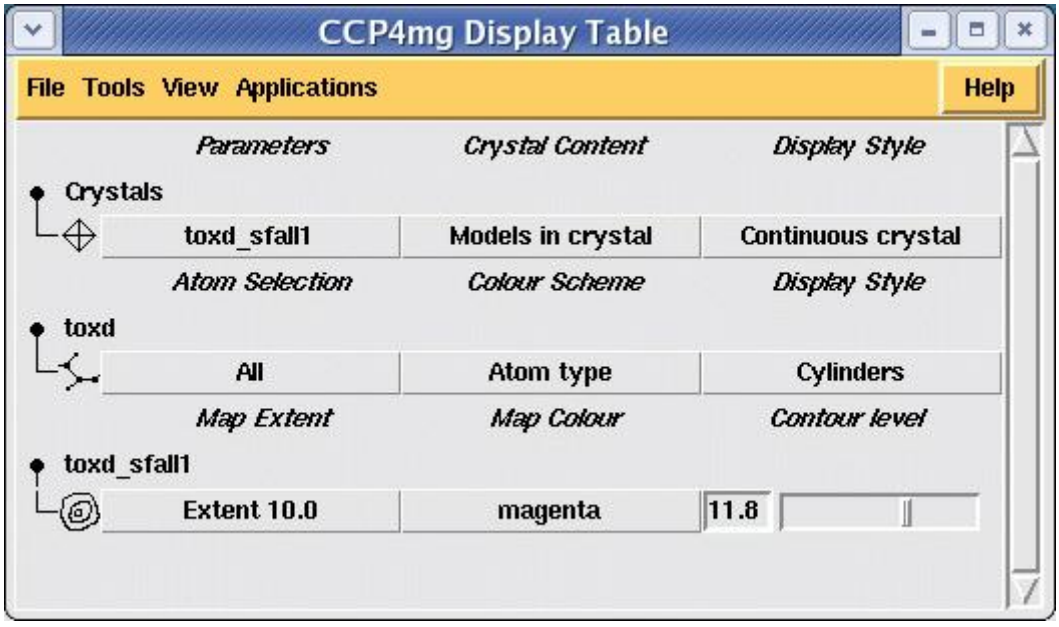
Introduction

It is often useful to display the symmetry equivalent molecules of a structure solved by crystallography. CCP4mg has two approaches to displaying symmetry related molecules
Using the *crystal* object This approach is more automated and is appropriate for crystallographic model building. One or more symmetry copies of the model are drawn to the display but only one copy of the model atomic data is stored in the program. If the model is changed (for example the structure is changed or the selection of displayed atoms is changed) then the changes will be reflected in all of the displayed copies.
or
Using the model tool *Generate symmetry mates*. This tool creates a copy of all the model atomic data, with only the atom coordinates changed (to the new symmetry related position). The original model and the symmetry copy can then be changed independently and any structure analysis or display tool can be applied to the two copies in a more flexible manner. This approach is useful for analysing or creating pictures of the crystal contact between symmetry related models.

Note that the cell and space group for the structure must be in the PDB file in the CRYST1 line. See the PDB Format Definition at wwPDB and the space group must match one of those defined in ccp4mg/ccp4_suite/lib/data/symop.lib. (Hint: the appropriate format for the space group name is the one in inverted commas at the end of the line.)

The *Crystal* Object

In CCP4mg a '*crystal*' is an object which has the usual crystal symmetry parameters (cell dimensions, space group and symmetry transformations) and which can 'contain' one or more of the loaded models. The crystal object controls the display of symmetry related molecules. A crystal object is created automatically when an electron density map is loaded and can also be created if model coordinate data containing complete crystal parameters is loaded. The crystal object is shown at the top of the CCP4mg Display Table so after a map and a model have been loaded the Display Table will look like this



In addition to the entries in the table for the model and the map there is an entry for a *Crystal* object. The crystal object interface provides tools to control appearance of symmetry related copies of the model and to show crystal feature such as the unit cell edges. The three items of the *crystal* row in the Display Table are

Parameters The crystal symmetry parameters (space group and cell parameters) are, by default, taken from the map or MTZ file (it is usually more reliable!) but the source of crystal parameters can be selected from the menu listing all of the map, MTZ or coordinate files.

Models in crystal A menu lists all of the loaded models and these can be toggled on or off to be included in the crystal.

Display Style The options are

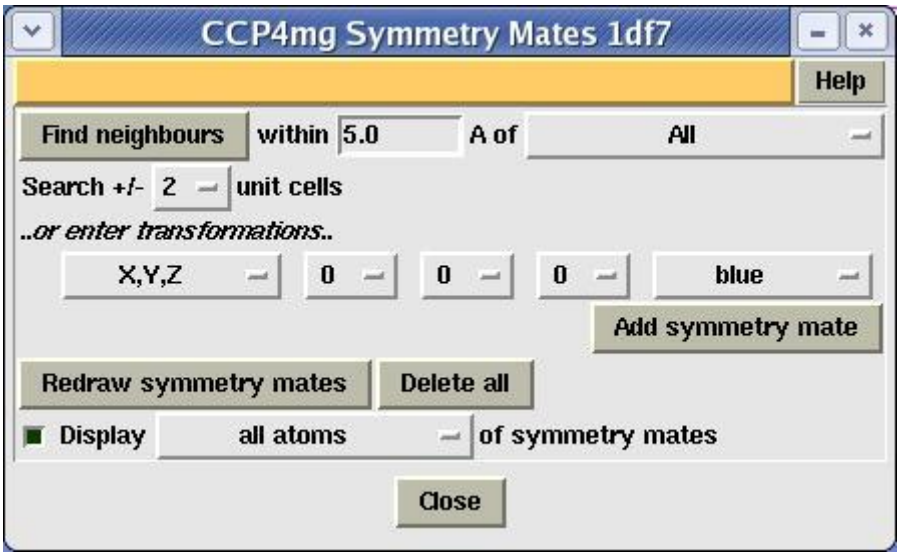
- **Continuous crystal** This is the default mode which is most useful when performing crystallographic model building. Symmetry related copies of the model are generated automatically if they occupy a region of the crystal which is currently in view in the display. So as the user translates or rotates the display the appropriate symmetry models are generated to give the impression of a infinite continuous crystal.
- **Contents one unit cell** This shows all of the symmetry related copies which have some atoms within the volume of the unit cell edges (see below).
- **Symmetry off** No symmetry related copies are shown.
- **Colour carbon by symmetry** This is on by default. The carbon atoms of the symmetry copies are coloured according to their symmetry operator.
- **Cell edges** Toggles the display of the unit cell edges

To remove a crystal, click on the icon  , and select **Delete** from the menu.

A crystal object can be created without loading a map. A coordinate file containing crystal symmetry parameters must already be loaded. From the **File** menu select **Add crystal** and select a coordinate file from which the crystal symmetry parameters will be read.

Generate Symmetry Mates

This tool can be found on the model icon menu, click on  next to the model name. The tool interface looks like this..



The top line of the window interfaces to a tool to find the symmetry mates which are close to either the whole of the original model or some part of the model. To use this: select a cutoff distance and, if required, use the selection menu to choose a part of the model; then click on the **Find neighbours** button. Searching for neighbouring symmetry mates can be quite slow but the progress is reported at the top of the window and there is a **Interrupt** button which will stop the task. By default, the search for neighbouring symmetry equivalent models is carried out over the range of 2 unit cells either side of the original model. If you suspect that this is not enough then there is an option to change the range.

The neighbouring symmetry mates are listed in a table in the interface and shown in the display. All of the newly generated atoms shown on the display are also added to the atomic data held in the program. The new, symmetry related, chains, residues and atoms are given the same names as the objects in the original structure but the separate symmetry mates are identified by a model number (this is the same as the numbering of models in NMR structures (see Coordinates IDs). Note that no change has been made to the coordinate file but if you wish to save the coordinates click on the model icon, , and select **File save/restore**.

The symmetry related copies are listed in a table showing the transformation that generated them and their colour on the display. The applied symmetry transformations can be changed by clicking on the button showing them and selecting from the menu. You can add new symmetry mates by clicking the **Add symmetry mate** button or remove existing ones from the table by clicking on a line in the table with the RIGHT mouse button and selecting **Delete symmetry mate**. These changes are only applied after you have clicked the **Redraw symmetry mates** button. Note that if you want to reapply the **Find neighbours** option that you will probably want to use the **Delete all** option to remove existing symmetry mates first.

The bottom line of the interface has options to control the display of the symmetry related models. By default a new model display object is created automatically to display the symmetry related models and all atoms of the new model are displayed.

syminfo.lib

The symmetry information in CCP4mg is taken from the file:
ccp4mg/ccp4_suite/lib/data/syminfo.lib
This is (or should be) the same as the CCP4 program suite distribution of this file. If your data file has a space group label that is not defined in this file then CCP4mg will not create

symmetry copies. You can fix problems by editing the syminfo.lib file particularly if you want to support an alternative space group name make a copy of the equivalent spacegroup entry, as shown here:

```
begin_spacegroup
number  8
basisop x,y,z
symbol ccp4 8
symbol Hall ' C -2y'
symbol xHM  'C 1 m 1'
symbol old  'C m'
symbol laue '-P 2y' '2/m'
symbol patt '-C 2y' '2/m'
symbol pgrp ' P -2y' 'm'
hklasu ccp4 'k>=0 and (l>0 or (l=0 and h>=0))'
mapasu ccp4 0<=x<-1; 0<=y<-1; 0<=z<-1
mapasu zero 0<=x<1; 0<=y<=1/4; 0<=z<1
mapasu nonz 0<=x<1; 0<=y<=1/4; 0<=z<1
cheshire 0<=x<=0; 0<=y<=1/2; 0<=z<=0
symop x,y,z
symop x,-y,z
cenop x,y,z
cenop x+1/2,y+1/2,z
end_spacegroup
```

and change the *symbol old* value to the spacegroup name that you have in your data.

CCP4 Molecular Graphics Documentation

Animations

Documentation Contents | On-line Documentation | Tutorials | CCP4mg Home

Contents

Introduction

CCP4mg will animate morphs imported from other morph-generating software and will also animate multiple NMR conformations. The structures can be displayed in any style available in CCP4mg and derived information, such as secondary structure and hydrogen bonds, is updated for each step. The animation can be made into a movie.

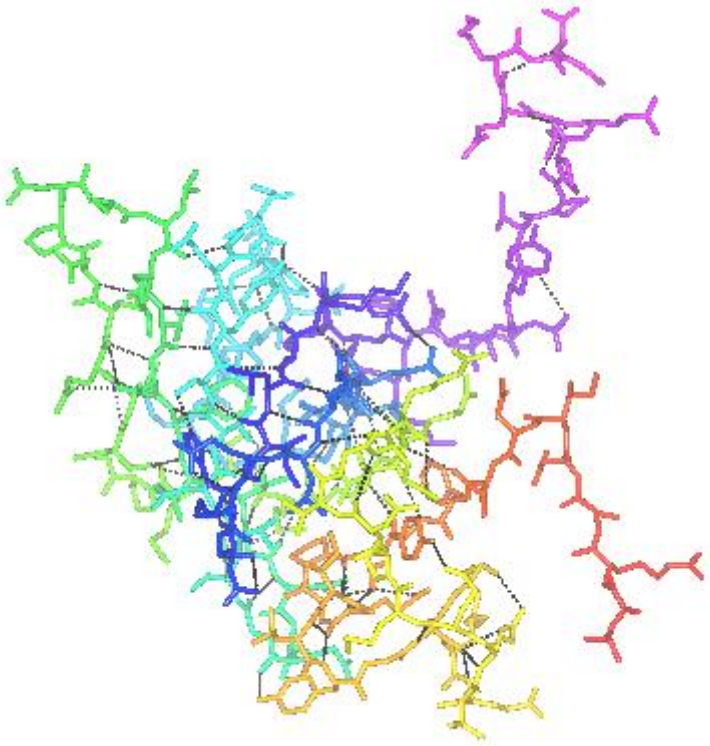
Morphs

Morphs are short animations which show a model inter-converting between two conformations. They are intended to help a viewer visualise the changes between the two conformations. They are not claimed to represent an actual conformational pathway. There are useful introductions at the sites linked below.

CCP4mg can not create morphs but it will display the morphs created by other means such as:
lsqman program - part of the Uppsala Software Factory package dejavu.
The Yale Database of Macromolecular Movements server.

These packages will export their results as PDB files - either as a series of PDB files or as one PDB file containing multiple 'NMR' models. CCP4mg can load the data in either of these forms. First it is necessary to load into CCP4mg a model which contains exactly the same atoms as the morph files - one of the PDB files output by the morphing package is most appropriate. Then go to the model icon menu (the dot next to the model name) and select the **Animation** option and then either **Select multiple PDB files** or **Select PDB with multiple models**. You will be asked to select a file - for the multiple PDB file case you should select just the first file of the sequence. The other files are deduced assuming that the filename has the form base-name*n*.pdb where *n* is a number between 1 and the number of files. (If you have data with filenames in a different form please inform the CCP4mg developers.)

Once the animation has been selected it will be saved and restored between program sessions. Selecting an animation does not immediately load the data so this has no significant computer memory requirements.



Morphing between two of the NMR models in 1liz.pdb done by MOVMOL.

Viewing Animations

After the data has been loaded the following options are available on the **Animation** menu:
Run animation Runs the animation continuously from first to last step and then backwards from last to first step. The animation can be halted by clicking the **Stop animation** button at the top of the window or hitting the keyboard Escape button (**Esc**). The start point of the animation can be set using the **Go to step..** option.
Go to step.. opens a window in which you can select an animation step to display and set the direction of the animation.
Show step legend Display a legend listing the current step. This is updated as an animation or movie is run.
Deselect animation The animation is 'forgotten'.

There are Animation **Preferences** in the **Model Analysis** folder of the Preferences window. It is optional whether properties such as secondary structure are updated for each frame of the animation and the minimum time between frame updates can be set.

Animation Movies

To make an animated movie you should first select the animation files and then open the **Movie** application from the **Application** menu. See the Movie documentation. As the movie options to Preview or Record a scene are run any selected animation will be automatically updated by one step for each frame of the movie.
Notes..
The Movie window button to **Fit Animation** will set the running time of the movie to accomodate one run through any selected animation. You may want to double this running time to accomodate a cyclical run of the animation from start to finish and back to the start. You may want to use the **Go to step..** option to reset the animation to the start before beginning recording.

You probably do not want the viewpoint of the movie to change during running so set the Final view to **Same as initial**.

Preferences

Animation preferences are in the **Model analysis** folder of the **Preferences** window. The options are:
Update properties (e.g. secondary structure) every step By default any properties which might change with the changing coordinates are recalculated.
Minimum time delay between updates This controls the speed of the animation replay.

CCP4 Molecular Graphics Documentation
Electron Density Maps

Documentation Contents | On-line Documentation | Tutorials | CCP4mg Home

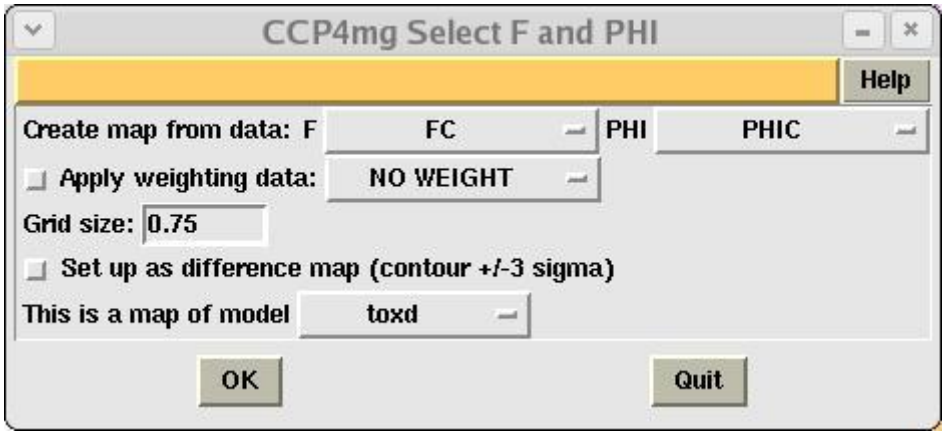
Contents

See also the tutorial on electron density maps.

Loading a Map or MTZ File

An electron density map can be read from a CCP4 map file or an MTZ experimental data file. Loading an MTZ file is slower as the map needs to be calculated but there is an option to recalculate the map with a different grid spacing which can be useful in creating the optimum image. Other experimental data formats can be converted to MTZ using the **Convert to MTZ and Standardise** task in the CCP4i interface; this can be found in the **Reflection Data Utilities** module.

This window appears when reading an MTZ file



You need to select structure factor (**F**) and phase (**PHI**) column data and optionally, **weight** column data, to calculate the map.
The default map **grid size** of 0.75Å is appropriate for model building work but a higher resolution (i.e. smaller grid size) may give a better picture but slower performance.
If the **Set up as difference map** option is on then the map will initially be drawn with two contour levels and +/- 3 sigma.
If there are model(s) loaded then there is an option to specify which model is associated with the map. This model will then be treated as part of the crystal and symmetry related models will be generated (see the Crystal documentation).

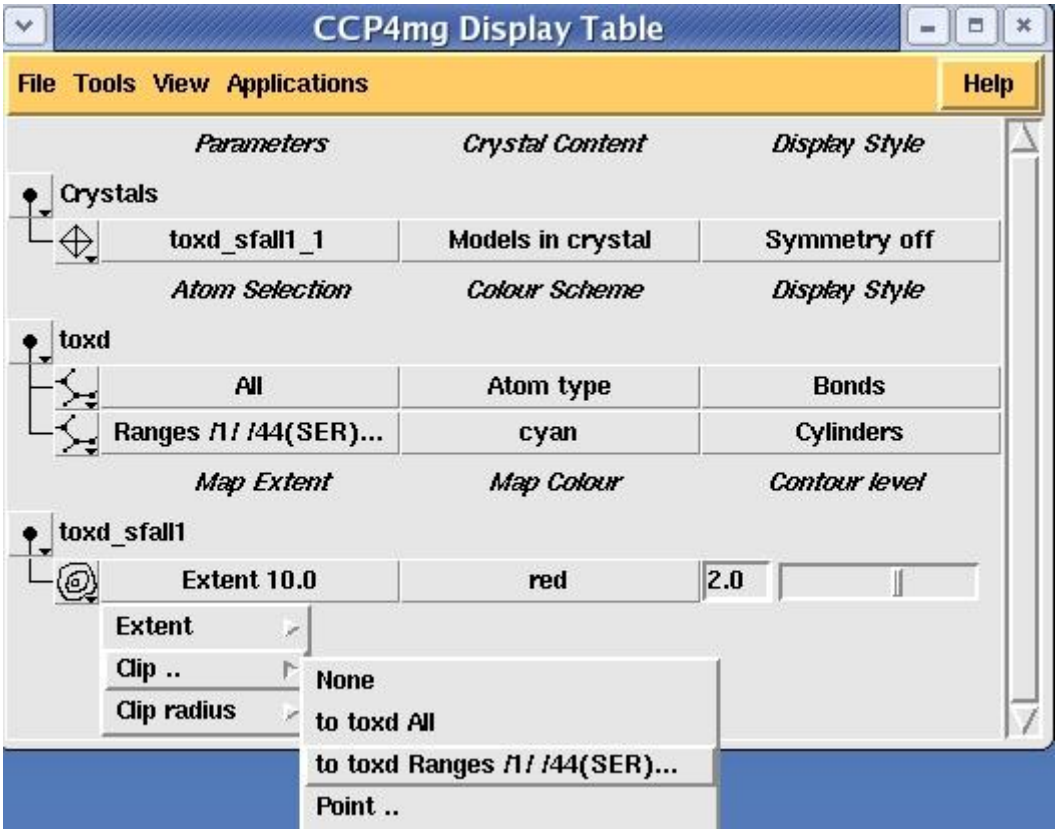
After the map is loaded the map data object is listed in the display table and a *Crystal* object is also listed. This object is mostly concerned with the display of the model(s) in the crystal and is described here.

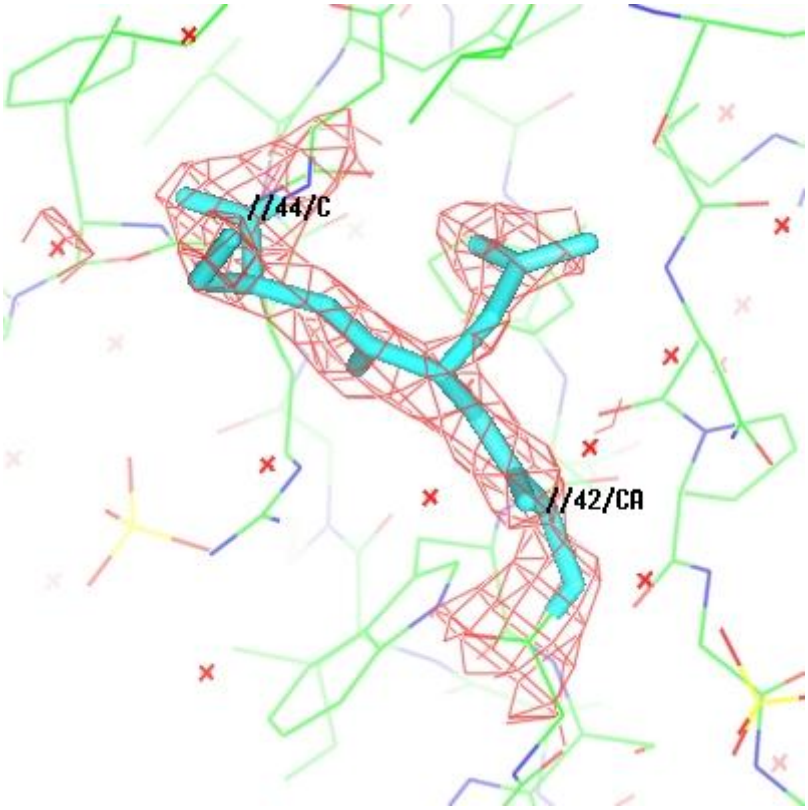
The Contour Display Object

The map display object is the contour level that is displayed as chicken-wire by default. In the display table the three main options for the map contours display are

Map Extent

There are various options to control how much of the contour surface is displayed. By default a 10Å box is shown at the centre of the display. The size of this box can be reset. The electron density can also be clipped to show only the density within a given distance of a selected set of atoms. The **Clip..** sub-menu lists all of the model display objects for models associated with the map and if you choose one of these then the map will be clipped to the atoms in that display object. An example of map clipping is shown below. There is a model display object which has a short range of residues selected and these are displayed as cyan cylinders. This display object can be selected from the **Clip..** sub-menu to give the picture shown below. The display object can be hidden if it is not required in the picture.





In this image there is pocket of density not associated with the the selected residue range. This can be removed by reducing the **Clip radius** using the final option on the Map Extent menu.

Map colour

There is a menu of all currently defined colours and options to add a new colour or edit the present colour.

Contour level

A slider bar controls the contour level. By default, the contour level value is reported in the absolute value of electrons/Å3.

The contour surface is shown as chicken-wire but this can be changed by clicking on the map display object icon and selecting **Surface style..** and then one of

- **Chicken-wire**
- **Cylinder** The chicken-wire is drawn in thin cylinders which may give a clearer image
- **Solid** the contour surface is shown as a solid object
- **Dots**
- **Contour Slice** see below
- **Mask** see below

Map Slices and Masks

On the map display object icon menu , the **Surface style..** option menu includes **Contour Slice** and **Mask** which draw a 2D plane through the map. The map slice shows contours and

the mask is a solid plane with grid squares coloured by the density value at that grid point. The extent of the plane is controlled by the options on the Map extent menu (left hand column of display table) and the maximum density cutoff for the contours or mask is controlled by the **Contour Level** slider bar in the right hand column of the display table. Other options are controlled by the **Contour slice details..** option on the icon menu.

The **Contour slice details..** window has the following options:
Contour spacing Set the slice contour spacing.
Define plane has a menu:
through selected atoms for which you choose three atoms
parallel to crystal face which gives a choice of crystal faces
specifying plane which requires three values to define a plane
For the last two cases the input defines the orientation of the plane but the position is set to the centre of the current display.

To get smoother contouring or a less coarse grid in the mask it may be helpful to reduce the **Map grid size** - this can be done from the map icon menu (the dot next to the map name). If a slice or mask is not displayed check that the Contour level on the display table is set fairly high and that the Contour spacing (in the Contour slice details window) is set fine enough.

Customising the Map Display

Map grid size on the map icon menu can be used to change the grid spacing if the data was loaded from and MTZ file.

The default initial appearance of the contour display object can be changed in the **Preferences** window (at the bottom of the **Tools** menu) - look in the **Maps** folder on the Preferences window.

If maps are loaded then an option on the **Tools** menu is **For all maps** has a sub-menu with options to **hide**, **show** or **delete** all maps. There is also an **Appearances** option which will change the appearance of all map objects.

Map drawing style preferences

Look in the **Maps** folder of the **Preferences** window for options to control the default appearance of electron density cntour objects.

The **sampling rate** is applied when generating maps from experimental data (MTZ files) and is the resolution of the map. The default value of 0.75 is quite coarse, values like 0.5 will give more attractive, but slower, pictures.
The **Map extent** is the size of the initially drawn cube of density; the extent can be changed for the individual contour objects in the left (selection) column of the display table.
Surface drawing style can be chicken wire, solid, dots or cylinders. This can be changed for individual contour objects via the icon menu.
Contour scale can be absolute (in electrons/Å) or sigma values. This controls the labelling of the contour level slider in the right column of the display table.
The **Max/min contouring level** for the slider is +/-6 sigma by default.
First/second contour colour set the initial colour of the first and the second contour objects for one map.
Maximum dot density and **Dot size** control the appearance contour level drawn in dots style.

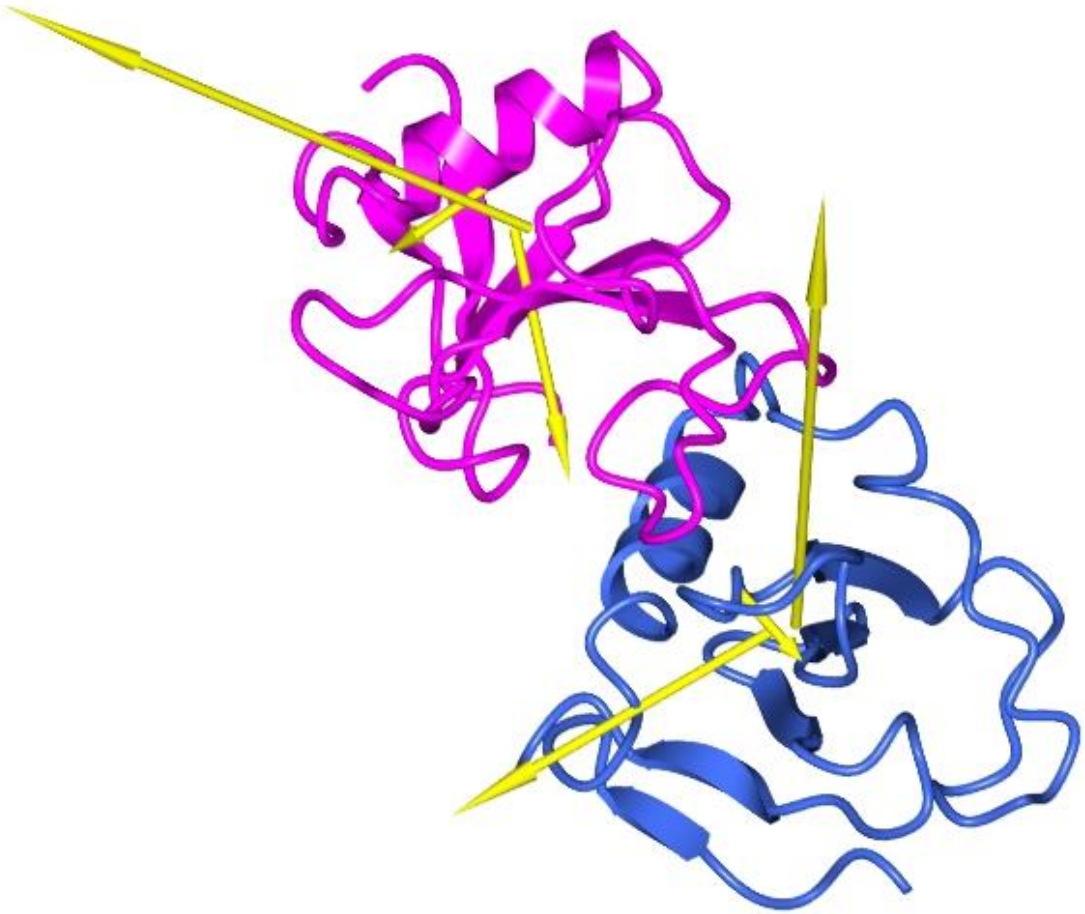
CCP4 Molecular Graphics Documentation
Vectors

Documentation Contents | On-line Documentation | Tutorials | CCP4mg Home

Contents

Introduction

CCP4mg can display arbitrary vectors (with the endpoints defined either by atom positions or xyz coordinates); these can be useful to show experimental results (such as the TLS axes output by REFMAC5) or to provide arrows or lines when creating images.



The vectors can come from

- a file written by another program
- created by the user in CCP4mg
- from a CCP4mg tool such as hydrogen bond analysis or Protein Superposition

On the CCP4mg display table each vector file is treated as a data object which can have one or more display objects. For each display object the user can change the selection of vectors, line appearance and label appearance. There are several options on display styles (cylinders, arrow heads etc.). A text label can be associated with each vector. The input file can define the appearance of the vectors (colour, position of labels etc.) but these parameters can also be changed by the user. The vectors can be given name tags so the user can select which vectors to display based on their tags.

Some CCP4mg tools have information that can be represented as inter-atomic vectors. If you want to put this information in an image but need to refine it by adding or deleting vectors, changing text labels or changing the appearance of the vectors then write out a vector file and use the Vector editing tools.

- Hydrogen bonds - use the **Export vectors** option on the hydrogen bond icon menu.
- Close contacts - use the **Export vectors** option on the close contacts icon menu.
- Protein Superposition - use the **Export vectors** option at the bottom of the window

Vector Appearance

On the display table the three main menus for the vector display object are

Selection

If vector tags have been defined then the selection menu will list the tags which can be toggled on or off to be displayed.

Line style/colour

There are various options to select the line style (e.g. dashed, cylinder, arrow, cone), colour, line width, cylinder radius, arrow head position, and cone radius.

Label style/colour

There are options to set the position,colour and font of the label

Scale vector length

The length of drawn vectors can be scaled using the **Scale vectors** option on the display icon menu. Note that this just changes the display and does not change the vector definition within the program.

Editing Vectors

Click on a vector with the right mouse button - the pop-up menu has the following options

Edit vector

Opens the Edit vector window which is described in the Creating Vectors section.

Move

This option will only work for vectors which are defined by xyz coordinates and not between atoms.

The sub-menu has options to move the whole **vector** or the **start** or **end** of the vector. After selecting one of these, move by holding down the *Control* key and the middle mouse button and moving the cursor in the main graphics window. The vector data icon (the dot next to the file name) is highlighted in gold to indicate that this object is the current active object. This tool will remain active until you either click again on the vector and select **OFF** from the **Move** sub-menu or some other object claims the active focus.

Delete vector

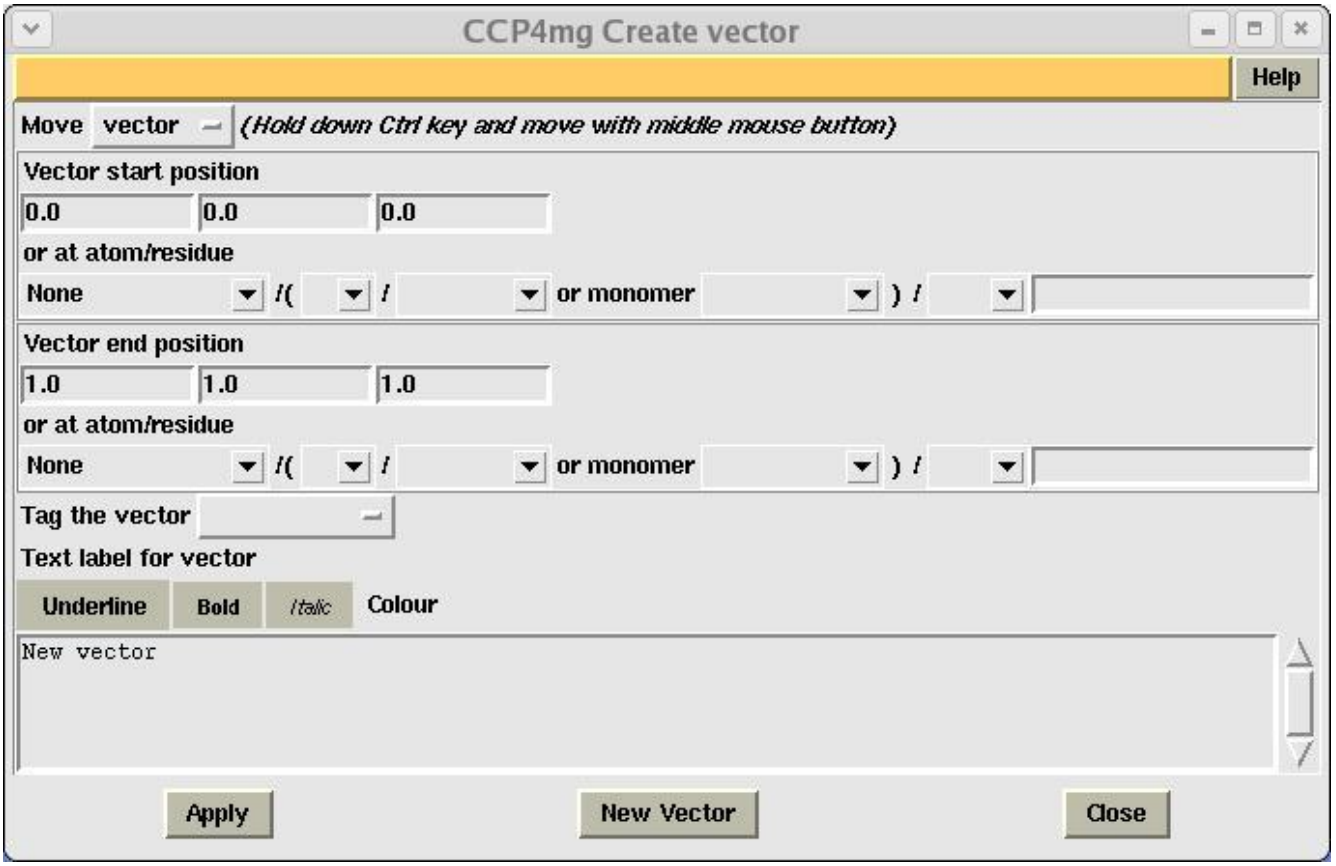
Delete the vector

Tag vector

This option only appears if tags have been defined for the vector data - see the **Edit tags** option on the vector icon menu. The defined tags are listed and one can be selected to apply to this vector.

Adding Vectors

Selecting the **Add vector** option from the vector icon menu (the dot next to the file name) or selecting **Add vectors/labels** and **Create** from the **File** pull-down menu will open the *Add vector* window that is shown below



The options in this window (dependent on if the vector is defined between atoms or at general xyz coordinates) are..
Move Select from a menu either to move the whole vector, or the start or end of the vector. Then hold down the *Control* key and the middle mouse button and move the cursor in the main graphics window to move the vector. This option is not available if the vector has been defined as being

between atoms.
Alternatively set the position of the cursor using the next two options
Vector start position can be set by entering the coordinates of choosing an atom position. If the vectors are not between atoms then any selection of atoms (e.g. a chain or residue) could be entered and the vector terminus would be placed at the centre of the specified atoms.
Vector end position As for the start position.
Text label for vector Enter text - there are options to set the text style and colour.
Tag the vector Select one tag to apply to the vector. Tag names can be created or edited using the **Edit tags** option from the vector icon menu (click on the dot next to the file name). The purpose of tagging vectors is to group them together so that they can be selected together to control their visibility or display style.
You will need to click on the **Apply** button at the bottom of the window to see the effect of your changes. Alternatively, clicking on **New vector** will apply changes to the current vector, create a new vector and initialise the interface.

If the *Add vector* window is open you can also select atoms using the atom popup menu: right mouse click on an atom to get the popup menu and then select the *Add vector* menu option.

Vector File Format

You only need to know about the file format if you plan to write a file from another program. If you have problems or suggestions please contact the CCP4mg developers *ccp4mg@ysbl.york.ac.uk*.

The file is in a CIF format (though currently there is no formal CIF dictionary file). The basic principles of CIF are described in the first chapter of *$CCP4/doc/ccifdoc.ps*. There is an example file for displaying the TLS axes vectors here.

The categories of data in the file are described below.

MODEL

This category is essential if the vectors are described in terms of atom positions. It provides a short identifying id and the name of a PDB/mmCIF file which is required to be loaded by CCP4mg in order to display the vectors. If the data in the file is associated with a particular PDB/mmCIF file then it is recommended that this category is provided.
Example

```
loop_
_model.id
_model.filename
"rnase"   "/xtal/ccp4/examples/rnase/rnase.pdb"
```

TAGS

This category is optional but recommended if the vectors have a tags data item. This category provides a fuller name for each tag which can be used in the CCP4mg GUI to provide a better description for the user. The data item name is a one word name for each possible tag. The annotation data item is a fuller description but since it will appear on a display table menu it should still be kept short.
Example

```
loop_
tags.name
_tags.annotation
"LIB" "Libration"
"SCREW" "Screw"
"TRAN" "Translation"
"RTRAN" "Reduced translation"
"SCREWROT" "Screw rotation"
```

DISPLAY_OBJECTS

This category is optional and can define the appearance of one or more data objects that are created when the vector data is loaded into CCP4mg. All of the items in this category are optional .

Data Item	Permissible Values	Interpretation
selection	A list of selected tags e.g. "LIB SCREW"	Only vectors with specified tags will be displayed. Default all vectors are displayed
line_style	"solid","dashed","arrow","dashed arrow","cylinder", "cylinder arrow"	Line style default is "dashed"
line_width	Positive integer	Not relevant for cylinder line styles.Default 2
cylinder_radius	float	Radius of cylinder
arrow_head	int	for *line_style* 'arrow' or 'dashed arrow'.Arrow head at 0=start, 1=end, 2=both
cone_mode	string	'fixed' or 'fraction'
cone_radius	float	For *cone_mode* 'fixed': the cone radius
cone_fraction	float	For *cone_mode* 'fraction': cone radius is *cone_fraction* of vector length
dash_length	float	Length of dashes for dashed vectors
dash_end	logical (0 or 1)	If 1 end vector with a gap. If 0 end with a dash.
line_colour	One of colours defined in $CCP4MG/python/etc/colour.py	Default "white"
label_mode	Position of label with respect to vector: 'no label','label centre','label start','label end'	Default is 'no label'
label_colour	One of colours defined in $CCP4MG/python/etc/colour.py	Default "white"
font_family	Text label drawn with this font. See any font menu for supported fonts.	Currently supported: new century schoolbook, helvetica,lucida, lucidabright,lucidatypewriter,times
font_size	The font size	Currently supported: 8,10,11,12,14,18,20,24,34
font_slant	Specifies if font is italic	Currently supported:o (=italic),r (=regular)
font_weight	Specifies it font is bold	Currently supported: bold or medium

Colours: blue, red, green, grey, yellow, magenta, royal blue, cyan,coral,pale green,pink,lemon,purple,tan,black
Example

```
loop_
_display_objects.line_style
_display_objects.line_colour
_display_objects.line_width
_display_objects.label_mode
_display_objects.label_colour
"cylinder arrow" "yellow" 2 "label centre" "yellow"
```

VECTOR

This category is compulsory. There are several reasonable combinations of data items.

Unreasonable combinations will be ignored.

Point-point vector requires the start and end x,y,z coordinate

```
loop_
_vector.start_x
_vector.start_y
_vector.start_z
_vector.end_x
_vector.end_y
_vector.end_z
```

Atom-atom vector requires the id of two atoms in the format described here. The MODEL category is compulsory to define the model containing the vectors.

```
loop_
_vector.start_atom
_vector.end_atom
```

If the atoms are in more than one different models then the appropriate model for each atom must be specified:

```
loop_
_vector.start_atom
_vector.start_atom_model
_vector.end_atom
_vector.end_atom_model
```

Atom-point and **point-atom** vectors are a combination of the above as:

```
loop_
_vector.start_atom
_vector.end_x
_vector.end_y
_vector.end_z
loop_
_vector.start_x
_vector.start_y
_vector.start_z
_vector.end_atom
```

Atom-displacement and **point-displacement** vectors define a start point and a displacement:

```
loop_
_vector.start_atom
_vector.end_d_x
_vector.end_d_y
_vector.end_d_z
loop_
_vector.start_x
_vector.start_y
_vector.start_z
_vector.end_d_x
_vector.end_d_y
_vector.end_d_z
```

There are additional optional data items:

Labels for the vector

```
loop_
_vector.start_atom
_vector.end_atom
_vector.label
"/A/27/CA" "/A/29/N" "Close contact 2.5A"
```

Tags for the vector

Tags enable the user to select a set of vectors for display. It is recommended that you also provide

a TAGS category.

```
loop_
_vector.tag
_vector.start_x
_vector.start_y
_vector.start_z
_vector.end_x
_vector.end_y
_vector.end_z
_vector.label
LIB 59.401     4.334     6.581
100.654    14.289     2.295 "Libration"
LIB 59.401     4.334     6.581
57.056    12.337     2.607 " "
```

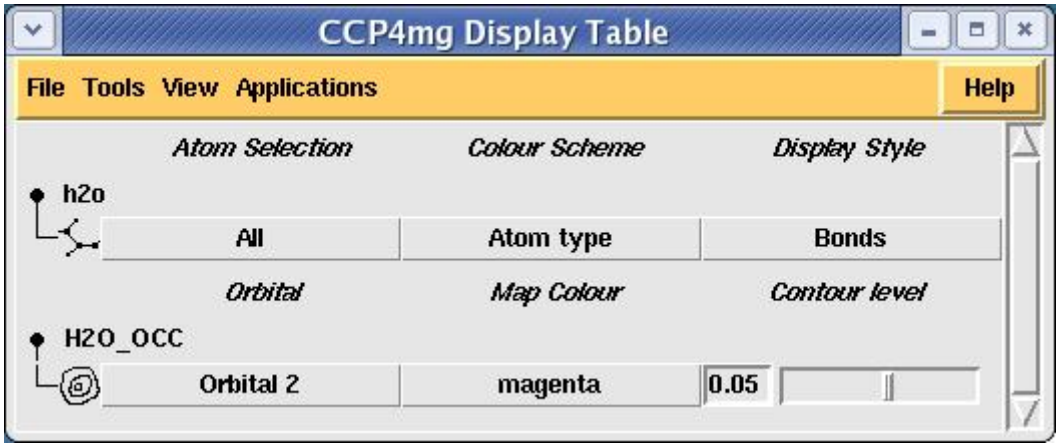
CCP4 Molecular Graphics Documentation
QM Fields

Documentation Contents | On-line Documentation | Tutorials | CCP4mg Home

Display of Quantum Mechanical Fields

If you are unfamiliar with CCP4mg see the tutorial for quick introduction to displaying atom coordinate files.

QM field (or other data) can be read in from the **File** menu option **Read map and MTZ**, sub-option **CUBE map file**. There is an example of a CUBE file in the tutorial data directory, *H2O_OCC.CUBE* and the corresponding atom coordinate file is *h2o.pdb*. By default, on loading the file, one display object, showing a single contour line for one orbital, is displayed. The display table after loading the coordinate file and the field map looks like this:



The three columns in the contour display object line control:

- Selection of orbital
- Colour of orbital for a single contour
- Slider control of contour level

Clicking on the  icon will open a menu with additional options:

- **Contour mode** By default a single contour, at the level controlled by the slider, is displayed. The alternative is to display plus/minus contours which will be coloured red and blue for plus and minus the contour level.
- **Set plot plane** Chose three atoms to define a plane and then set Surface style to **Contour slice** to show a slice through the plane of the chosen atoms.
- **Surface style** By default, the contour surface is shown as 'chicken-wire' lines but this can be changed to **Cylinders** which give a better appearance when making pictures or set to **Transparent** for a transparent surface.
- **Set opacity** Set the opacity of the transparent surface.

CCP4 Molecular Graphics Documentation

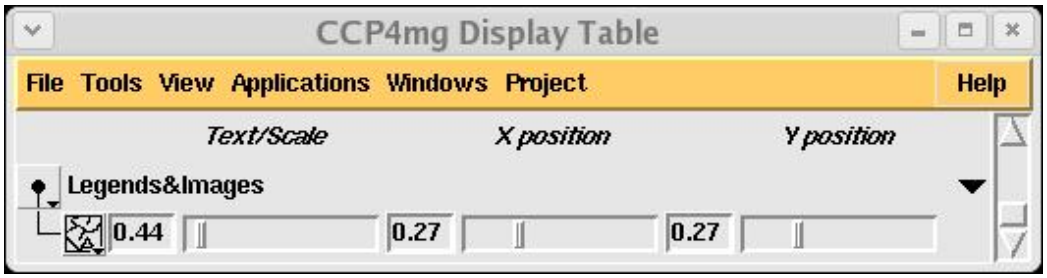
Text and imported images

Documentation Contents | On-line Documentation | Tutorials | CCP4mg Home

Contents

Imported images

CCP4mg can load and display images in PNG,GIF, PPM and JPG format. This option might be useful for displaying a logo or a graph. Load an image by selecting **Add image** from the **File** menu. The loaded images and legends are grouped under *Legends&Images* on the Display Table.

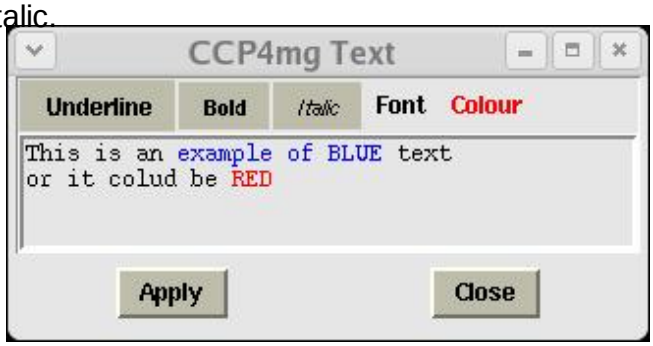


The row in the Display Table for each Image object has three sliders controling the scale, x position and y position of the object. The image icon menu (click on ) has option:

Aspect ratio opens window with slider cotrolling aspect ration of the image
Move object the icon is highlighted in gold and the image can be dragged around the screen by moving the cursor while holding down the *Ctrl* key and the middle mouse button (assuming standard mouse bindings).

Legends

A legend is text that can be placed anywhere on the 2D screen. It can be created by selecting either **Add legend** from the **File** pull-down menu or **Add legend object** from the *Legends&Images* dot menu. The three columns in the Display Table row for Legend object are a button that is initially labelled **Enter text** and two sliders to control x and y position. The **Enter text** button brings up a simple **Text** editor window in which you can enter the required legend text. Note that the font and colour or the text can be set,the text can be underlined or made bold or italic.



The Legend icon menu (click on A standard mouse bindings).

Notes on Fonts
Fonts with names beginning with capital letters are anti-aliased and so usually better looking. The fonts available within CCP4mg depend on what is available on your computer.
Text is not rescaled if you change the size of the display or output images at a different scale.
Greek symbols such as alpha or beta can usually done with a Symbol font.

Annotation

Annotation is text that is attached to a model object and which moves in 3D with the model object. Annotation can be created by selecting **Annotation** from the **Add display object** item on the model icon menu (the dot next to the model name). The easiest way to add annotation is using the Picture Wizard annotation tool.

The three items in the row for an Annotation object are:
Enter text .. opens a Text editor window as described above for Legends.
Colour menu - by default set to the complement of the background colour.
Position The x,y,x position (initially the centre of the model) is shown on a button which opens a window which has a standard molecule selection widget which can be used to select one atom, residue or chain. The annotation will be placed at the centre of the selected atom(s). While this window is open the annotation can also be dragged by moving the cursor while holding down the *Ctrl* key and the middle mouse button (assuming standard mouse bindings). This is the same as the icon menu **Move object** option.

The Annotation icon menu (click on A) has options:

Font.. (over-ridden by font specified in the Text editor)
Move object the icon will be highlighted in gold and the image can be dragged around the screen by holding down the *Ctrl* key and the middle mouse button (assuming standard mouse bindings).

Also see Notes on Fonts above.

CCP4 Molecular Graphics Documentation

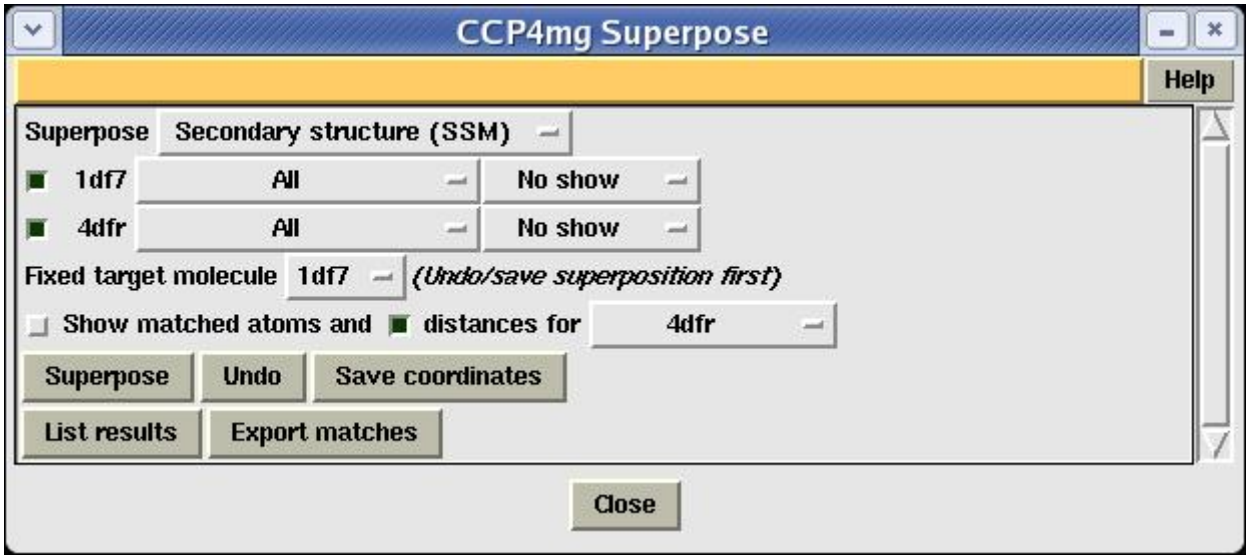
Superpose

Documentation Contents | On-line Documentation | Tutorials | CCP4mg Home

Contents

Introduction

Access to the Superpose application is from the *Application* menu. There are four different methods for superposing structures. They are explained in the folllowing sections. The method can be chosen from the menu at the top of the Superpose window.



All the loaded models are listed, one per line, in the window. On the line for each model is:
A radion button to indicate if the model is to be superposed (by default is on)
A selection menu to select a limited set of atoms to be used in the superposition (default 'All')
A menu to choose a colour for 'pyramid' highlighting of the selected atoms (by default 'No show')

Click the button labelled ***Superpose*** to superpose the structures. It is usually easier to see the resultant superposition if the options on the main ***Molecule*** pull-down menu are used to set ***Selection..*** to ***CA trace*** and ***Colour..*** to ***By molecule***.
All of the molecules are moved to the position of one 'target' molecule which does not move. The *Fixed target molecule* is selected in the window.
After superposition, if *Show matched atoms* is set on, the equivalent atoms in the target model

and one other selected model can be indicated by dashed lines between the atoms. The inter-atomic distances are also shown.
After the structures are superposed the window is updated to include buttons to ***Undo*** the superposition and ***Save coordinates*** to file.

The ***List results*** will give a listing of the matched SSEs (secondary structure elements) and residues, the inter-atomic distances and the transformation matrix. In the listing window there is an option to ***Save to file***.

It is possible to limit the region of the structure that is superposed using the selection menu which is, by default, labelled *All atoms*. For some cases it may only be necessary to select the limited set of atoms for the target model - this is discussed further for each method below.

Secondary Structure (SSM)

This method will superpose **protein** structures very quickly and without needing any input from the user to specify equivalent residues. The method works equally well for close homologs or for matching limited regions of similarity in very different structures. The user can select limited regions of the loaded models for matching but note that the method works by first matching equivalent secondary structure elements so is not applicable to very small fragments of protein or non-protein which do not include a á-helix or â-strand. If the user's selected atoms include one Cá from a SSE then the whole of the SSE will be used in the superposition.

For some structures there may be several possible superpositions which score similarly so there is a an option to show other possible matches for all superposed molecules.

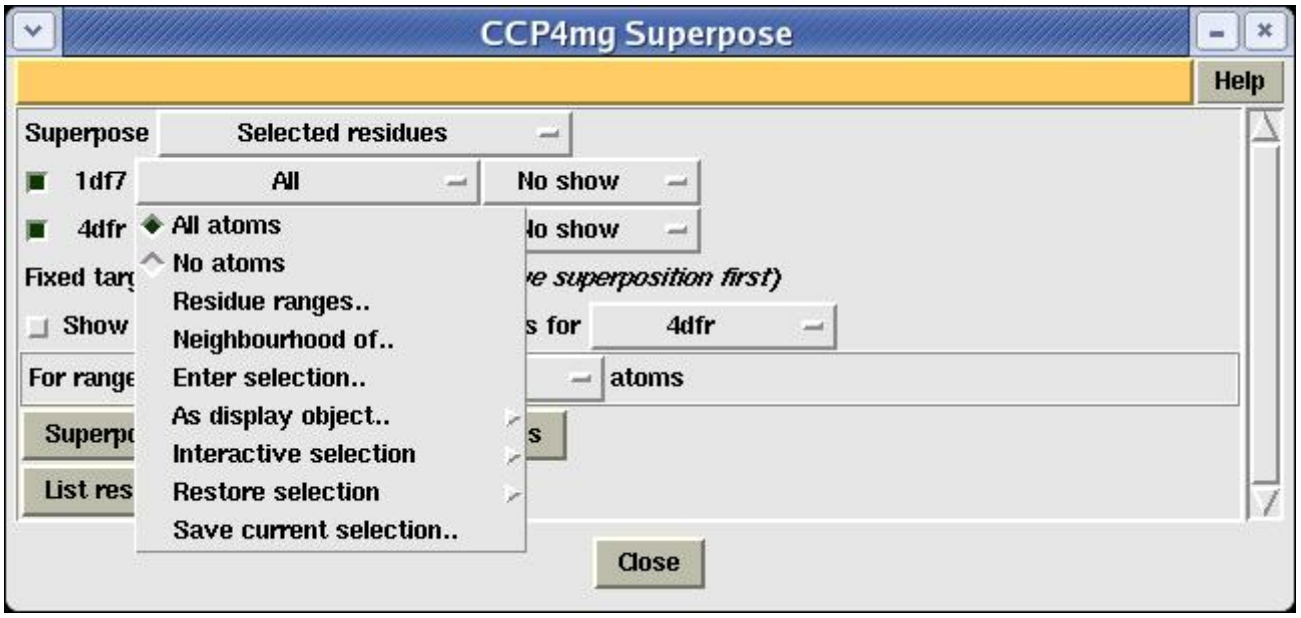
Close Atoms

This method is useful for performing locally optimised superposition after a global superposition by SSM. For example this method will optimise the superposition of the residues forming the binding site of a ligand. To do this you need to select the required local region of the 'fixed' model. This is probably best done using the *Neighbourhood of..* option from the 'fixed' model's selection menu and selecting the *Neighbourhood of* the ligand. It is not necessary to make any atom selection for the 'moving' models.

This method will, by default, do an SSM superposition to get the models reasonably superposed. It will then attempt to find the equivalent residues in the 'moving' model to the selected residues in the 'fixed' model. The criteria for equivalent residues is that the Cá atoms are within a given cutoff distance and the residues have similar orientation. The equivalent atoms within the equivalent residues must have the same name and be within a given cutoff distance. A least squares fit of all equivalent atoms is performed.

Selected residue

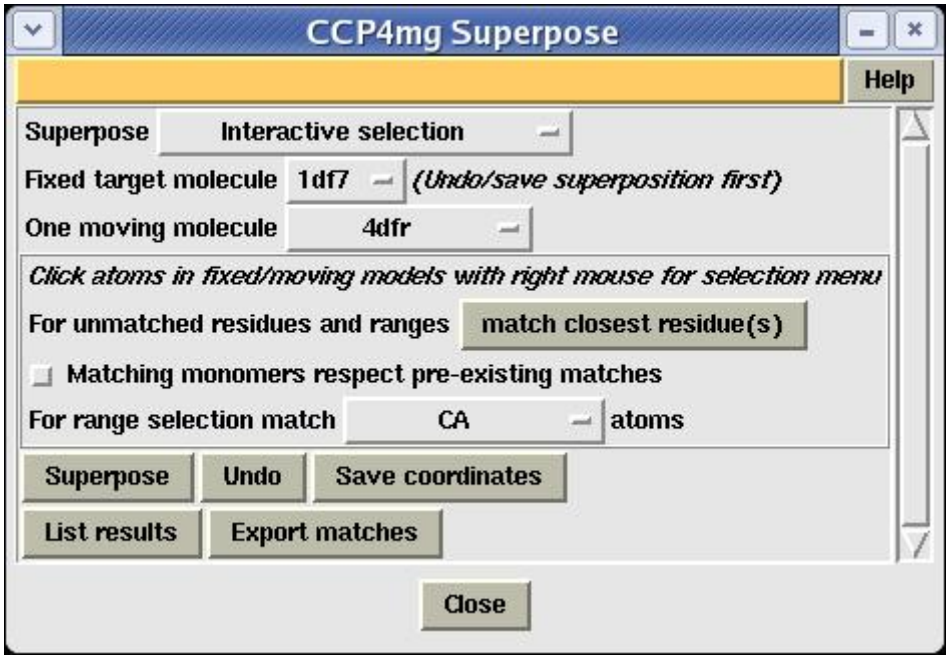
This method is closest in principle to the CCP4 program LSQKAB and will superpose proteins or nucleic acids.



You must select the residues to superpose for each of the structures using the selection menu by each of the model names. The *residue ranges* option on the selection menu is probably must useful in this context. You will be warned if you have not selected an equal number of residues for each structure. By default the CA atoms of selected residues are superposed but the menu labelled **For range selection match .. atoms** allows alternative selections.

Interactive selection

For this method the user must manually select atoms, residues or ranges of residues to superpose. The interface is significantly changed to



You must select two models to work on at any one time: a **Fixed target molecule** and **One moving molecule**. Selecting the matching atoms is done via tools on the popup menu that appears if you click on an atom in the fixed or moving molecule with the right mouse button. The popup menu will have a section labelled *Superpose* with the appropriate options from these:

Select this.. and a sub-menu of options atom,residue, start residue range etc.. After you have selected one of these, the required selection will be highlighted on the model with 'pyramids'. If you select 'Start of residue range' then you will need to click again on an atom to select 'End of residue range'. You can make several selections in either the fixed or the moving molecule before either selecting the matches in the other molecule (the **Match to..** option below) or automatically finding matches with the **find closest match** tool in the Superpose window.
Match to.. and a sub-menu listing selections is available after one or more selections have been made for the other molecule. If one of the selections is chosen then the apropriate atoms are matched and the matches are indicated by dashed lines between the atoms. Note that for matching residue ranges the the option in the Superpose window **For residue range match .. atoms** is applied.
Clear unmatched selections if you pick an atom with is selected but unmatched (indicated by 'pyramid' highlight) then this option allows you to clear the selections for that model.
Undo matches for atom/residue/all atoms if you pick a matched atom this will undo matches.

Note that if you have two molecules already closely superposed, for example by using SSM, then you may not need to select the matching groups in the second molecule. The **find closest match** option will match selected residues and ranges of residues to the closest equivalents in the other molecule. The criteria for 'close' are those used in 'Close atoms' mode. When the required atoms have been matched click the **Superpose** button in the Superpose window for a least-squares fit to be applied to the selected atom pairs.

Saving matches If you want to Superpose more that two molecules you can change the fixed and/or moving model and the display of any existing matched atoms will be hiden but the matches will be saved, and restored if you revert to the original molecules. The matches can be saved to file using the **Export matches** button; the matches are written to a vectors file which can be reloaded using the **Import matches** button.

Transforming Coordinates

You can enter a transformation matrix or move the model manually: from the model icon menu select **Transform coordinates** and then choose one of the options..

Enter transformation

In the window you can see the rotation matrix and translation vector that are currently applied to the coordinates and can reset these before clcking the **Apply** button to apply the transformation. Beware that you may need to click the **Undo** button to undo any existing transformation. The program checks that your input has the correct number of elements and that the rotation matrix has a determinant of one.

Move model

The model is made the 'active' moving object which is indicated by the model icon being highlighted in colour gold. To move the model you should hold down the **Ctrl** key and use either the left mouse button to rotate or the middle mouse button to translate the model. By default the rotation is about the centre of the model but you can reset the centre of rotation to an atom, residue, secondary structure element or chain by clicking on an atom in the model with the right mouse button and selecting **Rotate around this** from the pop-up menu.

Undo transform

This will return the model to its initial position.

Saving the transformed coordinates

The model will remain in the transformed position until the transformation is undone but the new coordinates will not be saved to file unless you explicitly do so. The transformation will be saved between program sessions and in status files and will be reapplied when the model is loaded. The new coordinates can be saved to file by selecting the **File save/restore** option from the model icon menu - this will invoke the multi-file saver window.

Methodology

The Secondary Structure (SSM) superposition method superposes pairs of structures by the steps:
Find the secondary structure elements (SSEs) and represent them as one simple vector spanning the length of the SSE
Find equivalent SSEs in the two structures using graph-theory matching by geometric criteria of distances and angles between the vectors
Superpose vectors representing equivalent SSEs
Find the most likely equivalent residues in the superposed SSEs
Superpose CA atoms of equivalent residues
Iterate the last two steps

The same procedure is also available on the EBI web server http://www.ebi.ac.uk/msd-srv/ssm

Krissinel & K. Hendrick (2004). Proceedings of CCP4 Study Weekend. Acta Cryst. D.

E.M. Mitchell et.al. (1989). J. Mol. Biol. 212 151-166

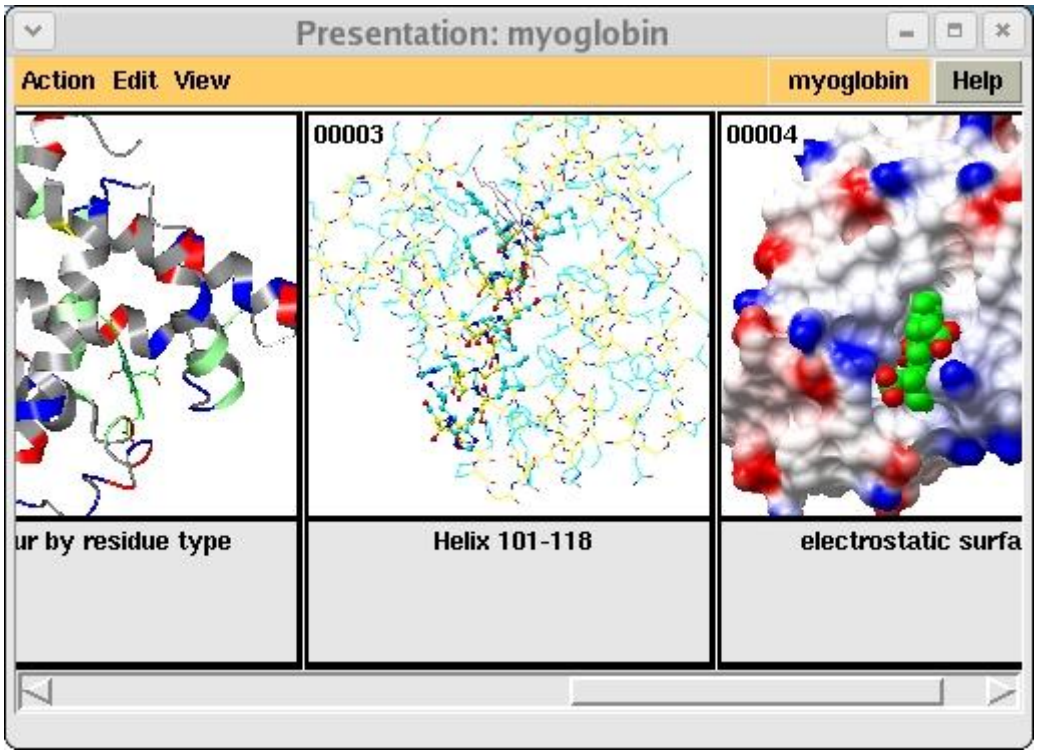
CCP4 Molecular Graphics Documentation
Presentations

Documentation Contents | On-line Documentation | Tutorials | CCP4mg Home

Contents

Introduction

This is nice interface to saving and restoring multiple program status. The Presentation application enables one user to set up multiple scenes within CCP4mg and save them so that other users (perhaps colleagues or students) can restore and examine exactly the same scene. The application could also be helpful preparing in advance for a presentation using CCP4mg 'live'. To save a scene the application makes a snapshot image of the scene and saves a CCP4mg status file. The snapshots are displayed in a window similar to the Movie interface window:



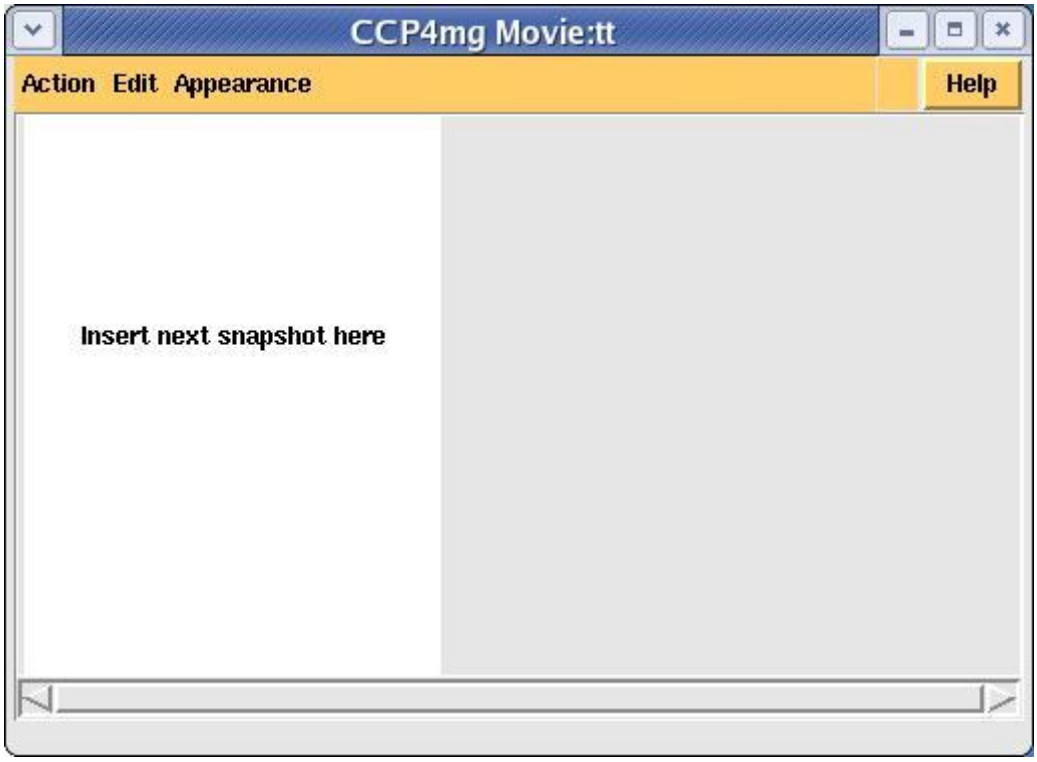
Clicking on one of the snapshots in interface caused the appropriate status file to be read and the program status shown in the snapshot is restored.

The snapshot and status files are saved in a directory which is given a name with the file extension *.ccp4mg_presentation*. To package up the presentation to send to someone else the application has a tool to copy all the data files used in the presentation to the *.ccp4mg_presentation* directory and to compress (zip) the directory.

Creating a Presentation

To begin pick **Presentation/Notebook** from the **Tools** pull-down menu. If you are creating a new presentation enter a name for it. For a presentation called *foo* CCP4mg creates a directory *foo.ccp4mg_presentation* in your current project directory. This directory will contain all of the snapshot and status files that are created.

The presentation interface initially looks like this



To load a snapshot, set up the display as you want it and then click on the **Insert next snapshot here** frame and select **Insert current view/display** from the popup menu. The current program status is saved to a status file and a small snapshot image is made and displayed in the presentation interface as shown in the Introduction.

Note that the first snapshot is labelled 00001 and subsequent snapshots will be labelled 00002, 00003 etc.. There is a text entry box under the snapshot in which you can enter a title for that snapshot.

A left mouse double-click on a snapshot picture will restore the status of that snapshot. A right mouse click on a snapshot picture will bring up a popup menu with options to
Restore the display and/or the view saved for that snapshot
Replace the snapshot with the current display status
Set insert to move the insert frame before or after the snapshot
Cut to remove the snapshot from the interface (but note that files are not deleted and can be restored using the **Load existing snapshot** option in the *Insert* frame).
Copy to save the snapshot to a buffer which can be **Paste**d in the Insert frame.

The Action pull-down menu

Next and **Previous** will load the status corresponding to the next or the previous snapshot.
Exit Exit the Presentation interface.

The Edit pull-down menu

Cleanup old files will delete any snapshot and status files that have been removed from the interface.

Package the presentation will collect all data files referenced in the presentations status files to the presentation directory and change the status files so that they now expect the data files to be in the presentation directory. Optionally compress the directory using a zip mechanism.

The **Preferences** for creating an html file are described above.

Viewing a Presentation

Note that the internal browser option idescribed below is only available on Linux systems.

Open a presentation using the **Open presentation** option on the **File** pull-down menu. If an *index.html* file exists for the presentation then it will be displayed in the internal browser; otherwise a similar Presentation interface to that show above is used. In the standard interface you can change the scene by:

- Selecting **Next** or **Previous** from the **Action** menu.
- Double clicking a snapshot with the left mouse button.
- Clicking a snapshot with the right mouse button and selecting a restore mode from the pop-up menu.

In the browser interface clicking on a snapshot in the html page will restore that status.

CCP4 Molecular Graphics Documentation

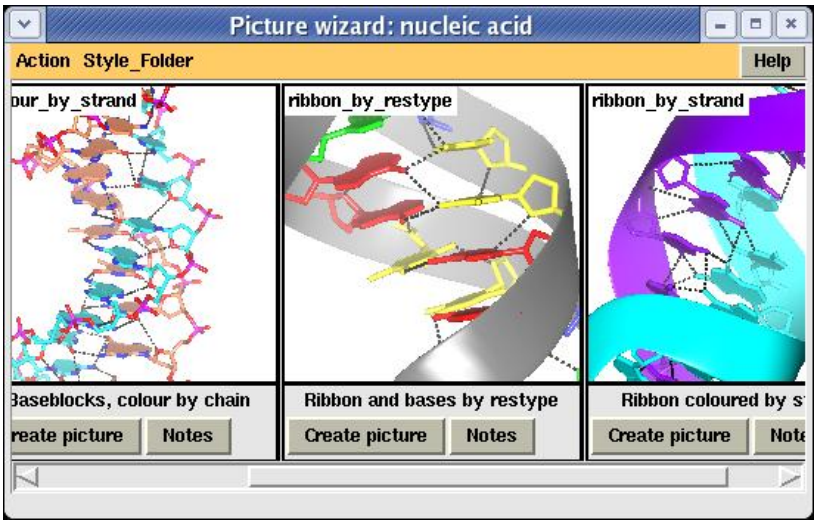
Picture Wizard

Documentation Contents | On-line Documentation | Tutorials | CCP4mg Home

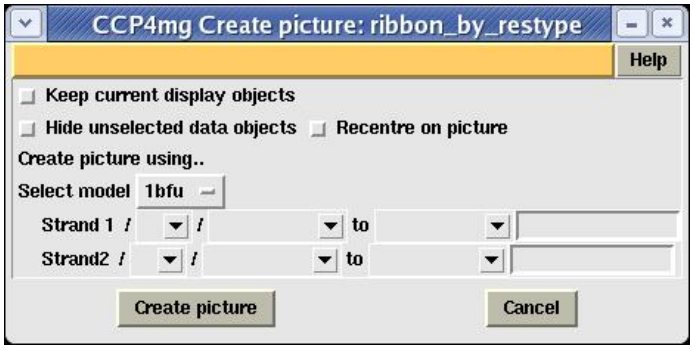
Contents

Introduction

The Picture Wizard is a tool to quickly set up complex pictures. The Picture Wizard interface shows pictures of various representations of molecular models - you can select a representation style and it will be applied to your loaded model. New display objects will be created for the model in the same style as those shown in the interface pictures. The interface looks similar to the Movies/Presentation interface:



There are a series of snapshots showing a representation style; beneath the snapshot are two buttons. The **Notes** button will bring up some notes describing the representation style with hints on how to customise it. The **Create picture** button brings up another window, for example:



This window has some standard options:
Keep current display objects By default this is off, existing display objects are deleted when the new display objects for the picture are created.
Hide unselected data objects Usually the picture wizard only changes the display of one model (or other data object); if this option is set on then any other loaded data objects will be hidden.
Recentre on picture Put the selected data object in the centre of the screen.

The remaining options depend on the selected representation style but they usually include options to select the data object that is to be changed and options to select ligands or chains etc. within the data object. Generally if you do not specify the ligand, chain or whatever the representation style will be applied to ALL ligands/chains etc.. Click the **Create picture** button at the bottom of the window to see your data object redrawn in the new representation style.

The representation styles are organised into **Style folders** such as 'nucleic acids' and 'ligand binding site'; you can change between

folders using the **Style folders** menu. The data files for the different representations are in the directory *ccp4mg/data/picture_wizard_files*.

It is possible to draw a model in a representation style immediately on loading. The coordinate file browser has options at the top of the window to select a representation style from a menu listing all of the Picture Wizard styles.

Hints on Customising Your Picture

After creating a picture with the Wizard you may want to change some aspect of the picture. It is generally easy to change the colour and display style of display objects so the representations available in the Wizard focus on examples of different atom selections and leave you to set the colour and display style. You may want to hide or delete some display object; use the options display object icon menu  to do this. In the **Create picture** window is the option to **Keep current display objects**; you could use this to set up display objects from more that one picture wizard representation; for example if your 'ligand' is nucleic acid you may want to use a *ligand binding site* representation and a *nucleic acid* representation.

Ligand binding site

Selecting the appropriate residues to display for an binding site is difficult to automate completely; it requires a human to know which residues are important biologically and to judge whether a picture is 'over complex'. The wizard ligand binding site pictures use up to three display objects with neighbourhood selections to define a binding site:
A selection of **main chain/side chain** close to the ligand (usually with at least one atom within 3 Å of the ligand). Note that using this mode rather than **residue** mode is usually effective in removing some main chain where only the side chain interacts with the ligand and vica versa.
A selection of **main chain/side chain** with hydrogen bonding capability to the ligand. The CCP4mg default cutoff for display of hydrogen bonds is usually 3.9Å which is very generous. To ensure no potential hydrogen bonding groups are lost the neighbourhood selection cutoff is usually 4.0Å.
To give some idea of the context of the binding site the whole structure can be displayed as *ribbons* or short stretches of main chain around the site can be shown (displayed as *worms* seems appropriate). A display object selecting neighbourhood **main chain** within 6-7Å of the ligand is used in some Wizard pictures.

If the automatic selection does not give the result you require you can try tweaking the cutoff distances but will probably have to use the Interactive Selection option to add or remove individual atoms or residues.

If you create a picture with a surface the surface will initailly be hidden - you need to click on the surface icon  with the right mouse button to make it visible. Calculating the surface may take some time. All surfaces will initially be drawn opaque; to make a surface transparent select **Set transparency** from the **View** menu and set an opacity level for the surface object (about 0.3 is usually good) and then toggle on the **Transparency on/off** checkbox. Beware that drawing transparent surfaces is very slow and you are best to toggle transparancy off before changing the opacity level, changing the view or doing anything else.

Nucleic acid

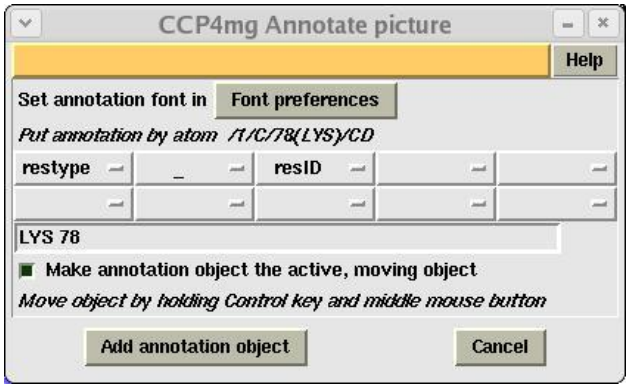
Most nucleic acid representations use 'base blocks' to fill in the bases with flat blocks. The base blocks are usually displayed in conjunction with a cylinders representation of the structure. Colouring the blocks different from the cylinders can be effective. The cylinders could also be changed to ball-n-stick though the stick thickness will probably need increasing. The backbone of the nucleic acid can be drawn as ribbon and in this case it often looks better if any accompanying cylinders exclude the backbone from the selection to minimise the bits of cylinder 'sticking out' of the ribbon.

Annotation: Labelling the Structure

Some Wizard pictures have the option to label binding site residues. These labels are in a fixed position relative to the residue Ca atom and have a limited range of formats. The labels can be toggled on or off, or the displayed information changed from the display object icon menu  options **Label atoms** and **Label text**. The font used for these labels cn be changed: in the **Preferences** window, in the **Display** folder, select **Fonts** and change the **Atom label** font.

An alternative approach to labelling the structure which gives more flexiblity in content and position of the text is Annotation. One annotation object needs to be created for each label and the object needs to be placed close to the appropriate piece of structure. The Wizard has a tool to create Annotation quickly: right mouse click on an atom that you want to place annotation nearby, select

Add annotation from the menu. In the window that appears



you can use the ten menus to select the syntax of a structure identification label using the identifiers (model, chain, resID, restype, atomID or altLoc) and separators. The identifier of the picked atom, in the selected syntax, is shown in a text input box below the menus. You can manually edit the text in the box if you like. After clicking the **Add annotation object** button a new Annotation object appears in the display window and the annotation is displayed on the model. If the option to **Make annotation object the active, moving object** is checked then the Annotation icon in the display table will be highlighted in gold and the annotation can be moved by holding down the Control key and the middle mouse button. The best placing of the text obviously depends on the intended view. Note also that the size of the text does not change as the display window size changes so you should choose a font size appropriate for the intended size of the output image.

Creating Representation Styles

You can set up a picture and save its representation style to use with other data objects. Be warned though, that you may need to do some editing of a script file to make something which is flexible and more generally useful. User created picture wizard files are saved in their home directory .CCP4MG/data/picture_wizard_files. Subdirectories corresponding to different style folders are created in this directory and each representation style needs two files: *style_name*.snapshot.ppm that is the picture that appears in the interface and *style_name*.mgpic.py that is a script to create the picture. The picture wizard interface has tools to create these folders and files. On the **Style Folders** menu is an option **New style folder** which will create a new directory in your picture_wizard_files directory. On the **Action** menu is an option to **Save current representation** which will create a snapshot file and a script in the current style folder directory.

Click on a snapshot with the the right mouse button to get a menu with tools to **Copy**,**Rename** and **Delete** the representation styles files. You can not rename or delete the style files in the CCP4mg distribution but copying them to your own folder may be a good starting point for creating your own styles.

Script File Format

The user-created picture wizard script files are called .CCP4MG/data/picture_wizard_files/*style_folder*/*style_name*.mgpic.py. The general format of the file has three sections:

```
#SECTION TITLE
The title that appears under the snapshot in the interface
(keep it short!)

#SECTION NOTES
The notes that the user can access from the interface

#SECTION CHOICES
A list of the options that will appear in the Create picture window (e.g. selecting data objects, ligands, chains).

#SECTION SCRIPT
A Python language script to create the picture
```

The options which appear in the **Create picture** interface are specified in the CHOICES section of the script file. This section specifies a list of variables which all have, at least, a name, a type and a label. The type and the label information are used to create an appropriate interface. A variable with the given name is initialised with the user's chosen value before the script is run so the script can 'see' the user's choices.

The SCRIPT section is a picture definition file with the display object attributes. The picture definition file is usually used to save the complete program status and includes definition of the data objects (particularly the name of the file that data should be loaded from). The picture wizard files do not have any definition of data objects as these are expected to be already loaded into CCP4mg. The picture wizard file defines the display objects - for example a MolDisp (model display object):

```
MolDisp (  parent = MolData1,
        colour = 'atomtype',
        style = 'BALLSTICK',
        selection_parameters = {
        'select' : 'nopeptide',
        'monomers' : monomers_list } )
```

This is a command to create a MolDisp (model display) object; the command has several arguments to define the object. The most important argument is *parent*, the name of its parent data object. The value for parent is in the variable *MolData1*; the value of this variable is initialised before the script is run from the users choices. *MolData1* is one of the variables which must be defined in the

CHOICES section. The other arguments determine the atom selection, colour and display style for the object and are explained in the object attributes documentation.

An example of a CHOICES section

```
CHOICES = [
    { 'name' : 'MolData1','type' : 'MolData','label' : 'Select model','multi' : 0 },
    { 'name' : 'ligand1', 'type' : 'monomer','label' : 'Ligand 1', 'parent':'MolData1' },
    { 'name' : 'ligand2', 'type' : 'monomer','label' : 'Ligand 2', 'parent':'MolData1' },
    { 'name' : 'label_site_residues', 'type' : 'logical','label' : 'Label binding site residues' , 'initial' : '0' }
      ]
```

This section is in Python syntax and defines a list called CHOICES. The items in the list are *dictionaries*. See the Python syntax page for an explanation of the Python syntax for lists and dictionaries.

An auto-generated script file created if you **Save current representation** includes the option to select the data object(s) in the CHOICES section but will probably include some commands in the SCRIPT section which are specific to the data objects that were used creating the files, for example the specific names of ligands or residue ranges. These specific selections probably will need replacing by variables that are specified in the CHOICES section.

CCP4 Molecular Graphics Documentation

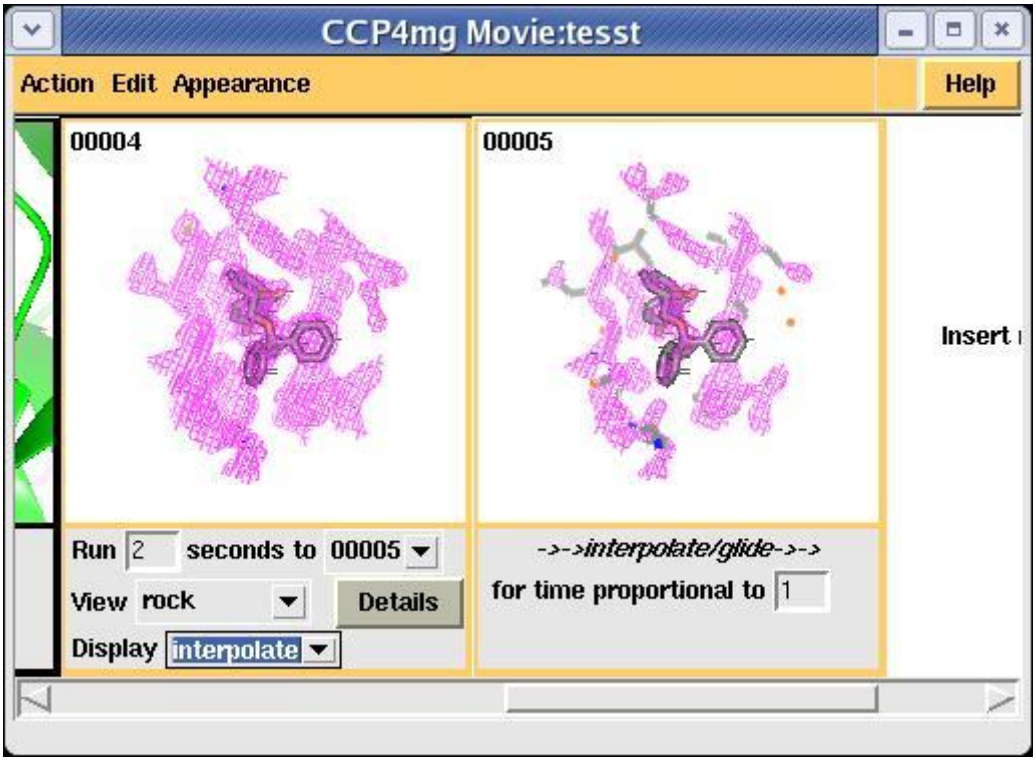
Movies

Documentation Contents | On-line Documentation | Tutorials | CCP4mg Home

Contents

Introduction

Molecular graphics movies are created by making many screenshots with one screenshot being equivalent to one frame of the movie. The screenshots are then compiled together into one movie file. It would be very tedious for you to set up and save every frame of a movie; instead, to make a movie in CCP4mg, you need to make some key 'snapshots' for each scene of the movie and the program will fill in the intermediate frames. The snapshots are displayed in the movie interface..



Below each snapshot picture in the interface are options to control the action when the program fills in the intermediate frames. This action can be a change in the view and/or a change in the displayed objects. The change in the view can be simple rotation or rocking about the vertical axis or some generic transformation that you can specify. It can also be a smooth glide to the viewpoint of the next snapshot. Various aspects of the display can be changed such as: the position of the objects, bond torsions (under development), electron density levels, parameters for calculations (such as maximum allowed length of a hydrogen bond) , colours. To specify the required change in the display you need to save a snapshot for both the beginning and end of movie scene with the display setup as you require and the program will smoothly interpolate the change through the movie scene.

CCP4mg will run the appropriate 'helper' programs (if they are installed on your computer) to convert the frames to the following movie formats :

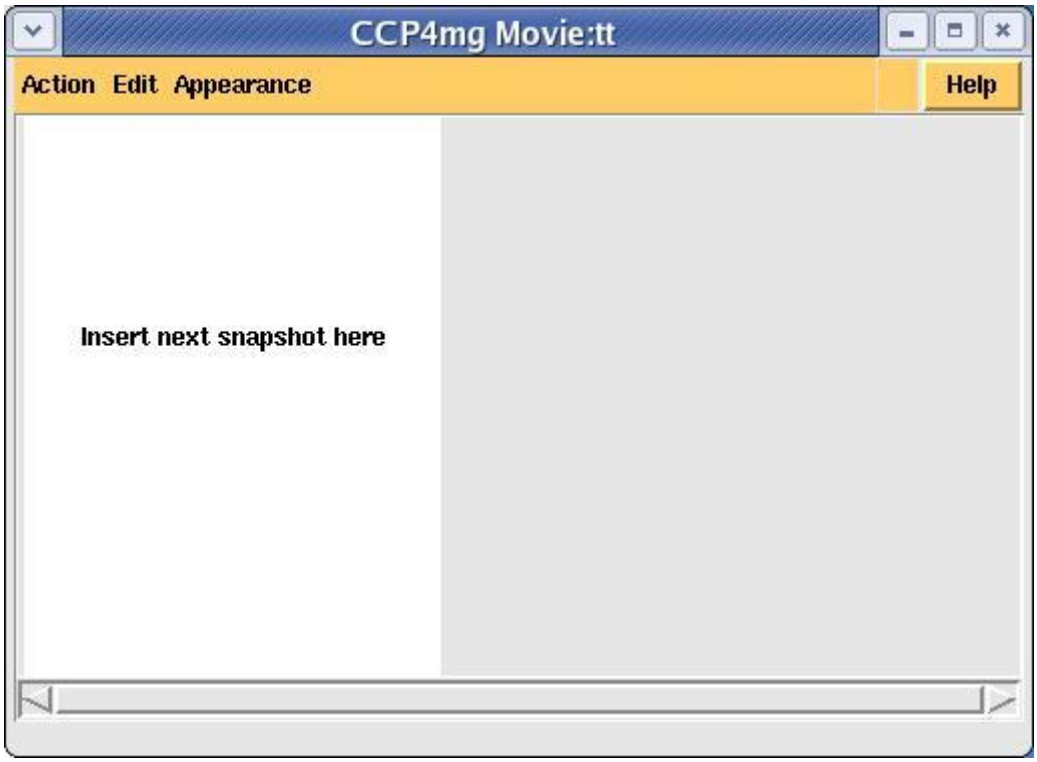
- an animated GIF file - can be displayed in a web browser or incorporated into a PowerPoint presentation.
- an mpeg file - can NOT be displayed in a web browser but can be incorporated into a PowerPoint presentation.
- an AVI file which can be displayed in most movie presentation programs

You could also use the saved frames as input to a movie generating program of your choice.

Overview of the Movie Interface

To begin a movie pick **New movie** from the **Movies** option on the **Applications** pull-down menu and enter a name for the movie. For a movie called 'foo' CCP4mg creates a directory 'foo.ccp4mg_presentation' in your project directory. This directory will contain all of the screenshot and movie files that are created.

The movie interface initially looks like this



To load a snapshot, set up the display as you want it and then click on the **Insert next snapshot here** frame and select **Insert current view/display** from the popup menu. There are other options to enter a new snapshot. The current program status is saved to a file and a small snapshot image is made and displayed in the movie interface as shown in the Introduction.

Note that the first snapshot is labelled 00001 and subsequent snapshots will be labelled 00002, 00003 etc..

One snapshot is enough to define a simple movie scene in which the view (controlled by a **View** menu underneath the snapshot picture) is either kept fixed, rotated (by 360 degrees about the vertical axis) or rocked (default 20 degrees about the vertical axis) or transformed by some rotation or translation that you should enter in the window accessed via the **Details** button. The time taken for the scene can be set in the first line underneath the snapshot picture. You can see the effect of these options using the **Preview** option on the **Action** pull-down menu - this will show what the movie would look like but does not actually make any screenshots.

More complex movie scenes can be defined by adding more snapshots in which either the view or some property of the display (such as model position or electron density contouring level) is changed and the movie scene will move smoothly between the different snapshots' view or display properties. To do this you will need to insert one or more snapshots and then, in the first line under the snapshot picture change the **Run..to** index number, which was initially 00001 to 00002 or the last snapshot required in the scene. Note that the second and subsequent snapshots in a scene are then labelled ->->*interpolate/glide*->->. The **View** should be set to **glide to** and/or the **Display** set to **interpolate to**. See Transforming Display Objects.

To create a 'fly-through' effect in the movie you can save a series of different views, set the 'Run .. to' option for the first snapshot to the index number of the final snapshot and set the **Run .. seconds** time to something appropriate. The length of time spent between each

snapshot is proportional to the value under each snapshot labelled **time proportional to** but the total time for the scene is always the **Run .. seconds** time for the first snapshot in the scene.

.

Animations can be recorded in a movie. To record an animation you need save a snapshot with the animation loaded (see the Animation documentation for details on loading an animation). If the display mode is set to **interpolate** then the scene will be created with one step of the animation per frame of movie. See Animation in Movies.

There are various tools to change the snapshots set up in the movie interface. A left mouse double-click on a snapshot picture will restore the status of that snapshot. A right mouse click on a snapshot picture will bring up a popup menu with options to
Restore the display and/or the view saved for that snapshot
Replace the snapshot with the current display status
Set insert to move the insert frame before or after the snapshot
Cut to remove the snapshot from the interface (but note that files are not deleted and can be restored using the **Load existing snapshot** option in the *Insert* frame).
Copy to save the snapshot to a buffer which can be **Paste**d in the Insert frame.

The Action pull-down menu

The following tools can be applied to just one selected snapshot or range of snapshots. A left mouse click on a snapshot will select that snapshot and highlight it with a gold border. Holding down the shift key and clicking on another snapshot will select and highlight the range of snapshots. Where a snapshot does a glide or interpolate to one or more other snapshots then you only need to select the first snapshot or the group for all to be used.
Preview plays the frames that would make up the movie in the main display window but does not record anything.
Make movie records the frames of the movie and then coverts them to a movie. This option performs all three steps described below for **Make movie in stages**.
Make movie in stages is a menu with sub-menu options..
Record frames plays the frames and saves them. Note that the display might not change if the screenshots are done in offscreen mode (see here). The *ccp4mg_presentation* directory contains subdirectories for the scenes which are called 'foo.ccp4mg_presentation/scene_00001' to 'foo.ccp4mg_presentation/scene_*n*'. After recording a scene the scene directory contains screenshot images which are named 'frame_000001.png' ,'frame_000002.png' etc.. Where a snapshot does a glide or interpolate to one or more following snapshots all the screenshots and movie files are saved into the scene directory of the first snapshot.
Convert frames to scene movie(s). compiles the frames into a movie for each of the selected scenes. The movie file is saved in the appropriate scene directory - so a typical movie file path name might be 'foo.ccp4mg_presentation/scene_00001/movie.gif'. Compiling a movie can take some time and is performed in the background - you can continue doing other things while it is running.
Merge scene movies merges the movie files from the selected scenes into one movie file. For example if you merge scenes 00001 to 00004 then the resultant movie file will be called 'foo.ccp4mg_presentation/movie_00001_00004.avi'.
Play will play the movie.

Loading a New Snapshot

A new snapshot will always be loaded into the position of the frame labelled *Insert next*

snapshot here. This frame can be repositioned by clicking on a snapshot picture with the right mouse button and selecting the **Set insert..before/after** option from the popup menu. On the same snapshot popup menu is an option to **Copy** the snapshot. This can then be **Paste**d into the *Insert* frame. The new, pasted, snapshot is given a new label. The other options on the *Insert* frame menu are to **Insert current/view display**, that will create a new snapshot from the current graphics window display, and **Load existing snapshot** which will list the existing snapshot files even if they are not currently loaded into the movie interface.

Changing the Display

Some of the attributes of the displayed objects can be changed during the course of a movie so, for example, models can move and electron density levels can change. (Note that a model is moved using the **Transform coordinates** options on the model icon menu - the dot next to the model name.)

Currently the transformable attributes are:

- Model object: position
- Map contours: contour level
- Legend: position
- Annotation: position
- Image: position and scale
- Vectors: scale
- All objects: transparency
- The definition of colours given in **Preferences**->**Display**->**Colours**

Some calculation parameters such as the maximum hydrogen bond distance can also be interpolated. Requests for other options are welcome.

Other changes between snapshots such as adding or removing data or display objects will be ignored - the composition of the initial snapshot will be retained throughout the movie shot.

Animation in Movies

See the Animation documentation for details on loading an animation. To make a movie with one step of the animation per frame of movie you must load the animation then save a snapshot; "Animation of xxxx" will be written over the snapshot picture in the movie window and the run time for that shot will be set automatically to the appropriate time to run through the animation once. To run the animation in the movie the display option must be set to **interpolate**. Note that the first frame will always have the animation step that was set when the snapshot was saved. If the display includes some feature such as hydrogen bonds which are dependent on the model conformation then they will be updated for each step of the animation. It is perfectly possible, but probably slow, to set the model display style to Surfaces which will be recalculated for each step in the animation.

Scene Details

There is a **Details** button for the first snapshot in each scene. The Details window has a collection of options for different view modes..
For all view modes..
There is an option to 'rock' back to the initial position - with this set on the specified actions for the scene will be performed and then they will be reversed back to the initial view and display. The action of the scene can be repeated multiple times.
Rock details
The angle and period of the 'rock' view mode can be changed.

Transform details
The rotation, translation and zoom components for any arbitrary transformation can be specified
Screenshot drawing quality
By default, before making a screenshot, the display is redrawn with the highest quality rendering. The high quality rendering takes significantly more time than the normal quality rendering and, if used making movies, will make recording the movie very slow. Since movie are usually done at low resolution (say 600 by 600 pixels) the high quality rendering is often unnecessary and by default is turned off. For a movie scene which is very close up so the rendering quality is more visible it may be worthwhile changing the quality from 'fast' to 'good'.
Movie details
There is an option to ray-trace the scene using POV-Ray. See below.

Replace .. Propagate: Editing multiple snapshots

This tool will enable you to make the same change to multiple snapshots. To use this **Restore** the first snapshot that requires changing and make the changes. Then select the range of snapshots that require the same changes and then select **Replace** and **propagate changes** from the right-mouse popup menu for the snapshot. When replacing the the snapshot the program saves the old snapshot status and can compute the changes between the old and new snapshots and apply the same changes to other snapshots. The program will go through all of the selected snapshots restoring the status, editing and saving. The sort of changes that can be handled in this way:

- the selection/colour/style of a display object
- parameters set via the **Preferences** interface
- colour schemes

Major changes such as adding/deleting a data object or display object can not be propagated.

Edit Tools

Delete movie files will delete any movie files for the scenes of currently selected snapshots.
Delete frame files will delete any frame files for the scenes of currently selected snapshots.
Cleanup old files will delete all old snapshot files which have been deleted from or replaced in the movie interface.
Collect data files will collect all data files used in the snapshots to the movie directory and change the snapshot status files so that, in future they will load the data from the files in the movie directory. There is an option to compress the movie directory. The compressed (zipped) file is called 'foo.ccp4mg_presentation.zip' and can be copied elsewhere. The file selection for opening movie or presentation files will list the zipped files and if one is selected it will be uncompressed and the movie or presentation will be opened.
Preferences are described below.

Preferences

The **Preferences** for the movie are on the **Edit** pull-down menu. You can set the intended frame rate and width and height for the movie - note that if you change these you will have to re-record any existing frames or movie files. The movie format can be chosen but not that the formats available are dependent on the installation of appropriate helper programs. See Compiling Movies.

Screenshot drawing quality see Scene Details.

Compiling Movies

Movie compilation is done by the script $CCP4_MG/bin/compile_movie. This script can be run external to CCP4mg,
.../ccp4mg/bin/compile_movie -help
will give more information. This script uses either:
convert program which is distributed with ImageMagick to generate animated GIF or MPEG1 format movies dependent on the file extension of the output file (should be .gif or .mpg). To get acceptable quality with the mpeg file the '-quality 100' option is necessary.
mencoder which is distributed with mplayer, can create movies in many different formats using a variety of codecs. The most useful is probably an AVI format file with MPEG4 encoding which should replay in PowerPoint. To get reasonable molecular graphics movies usually requires using the options for high quality and low compression. The default command line is:

```
-ovc lavc -lavcopts vcodec=msmpeg4v2:vbitrate=1800
```

Some web pages with appropriate advice on mencoder and ffmpeg:
http://electron.mit.edu/~gsteele/ffmpeg/
http://ffmpeg.mplayerhq.hu/compat.html for suggestions on using ffmpeg.

On **Windows** systems the **Movie Maker** utility can be used to convert either a collection of .png files or an animated gif file to some other movie format. To do this
1) from the **File** menu choose **Import into collections**
2) select the png or gif file and click **Import**
3) drag the frames or the movie from the middle view to the story board below
4) from the **File** menu choose **Save movie file**

Movie Compile Preferences

Find these in the **Preferences** window in the **Images and movies** folder. For the three programs which can be used to compile movies (*convert*, *ffmpeg* and *mencoder*) you can specify the full path name for the program executable (so the the CCP4mg compile script can find it) and specify command line argements. Note that *mencoder* also has specific command line arguments for merging two movies. Similarly the full path and command line arguments for the mpeg player *mplayer* can be set.

POV-Ray images in Movies

If you have POV-Ray installed then it can be used to perform ray-tracing for every image in a movie but, be warned, this will be very slow, both in writing out the POV-Ray input file for each frame and in compiling the movie which involves running POV-Ray for each frame. The time is proportional to the complexity of the picture and the number of frames.

It is advisable to try making a test movie of a few frames to see if you like the effect of the lighting scheme. The position of the lights used in the POV-Ray rendering can be set by the **Lighting** options, under **Display** in the **Preferences** window. See the Lighting help.

To use POV-Ray on a movie scene go the the snapshot Details window and toggle on the **Ray-trace scene with POV-Ray** button.

Hopefully users do not need to know about this but just in case: the program waits for a timeout period (default 20 secs) for each individual POV-Ray input file to be written. If you find files not been correctly created it may be worth increasing this time (see **Preferences** on **Tools** menu and **Movies** in the **Images and Movies** folder).

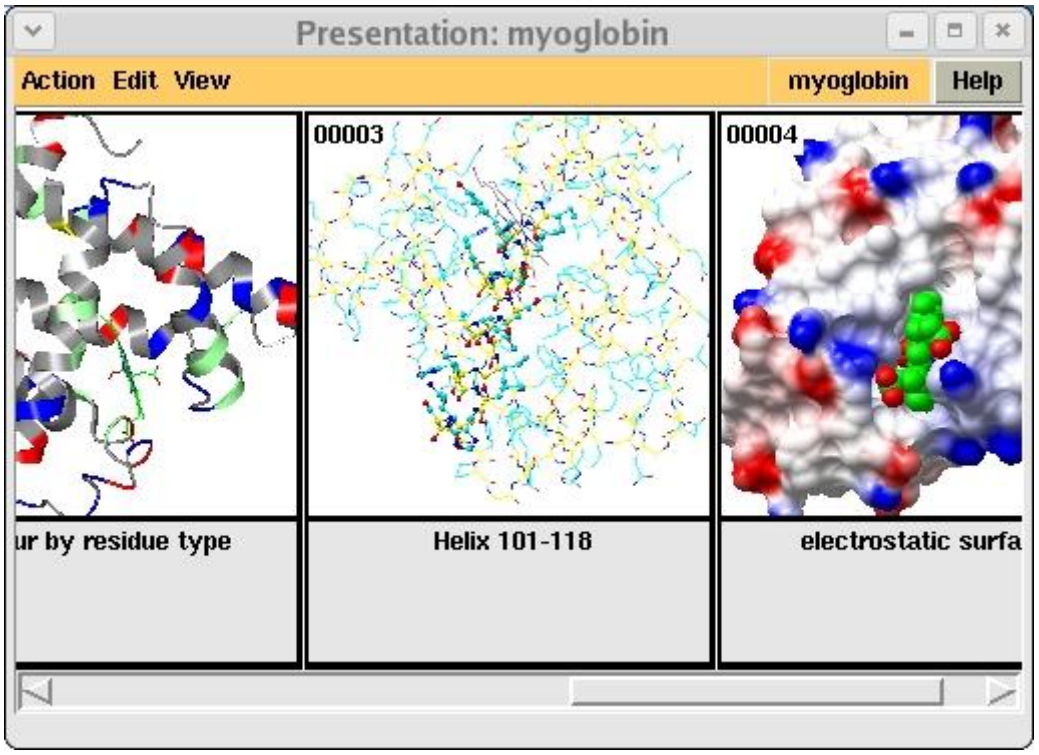
CCP4 Molecular Graphics Documentation
Presentations

Documentation Contents | On-line Documentation | Tutorials | CCP4mg Home

Contents

Introduction

This is nice interface to saving and restoring multiple program status. The Presentation application enables one user to set up multiple scenes within CCP4mg and save them so that other users (perhaps colleagues or students) can restore and examine exactly the same scene. The application could also be helpful preparing in advance for a presentation using CCP4mg 'live'. To save a scene the application makes a snapshot image of the scene and saves a CCP4mg status file. The snapshots are displayed in a window similar to the Movie interface window:



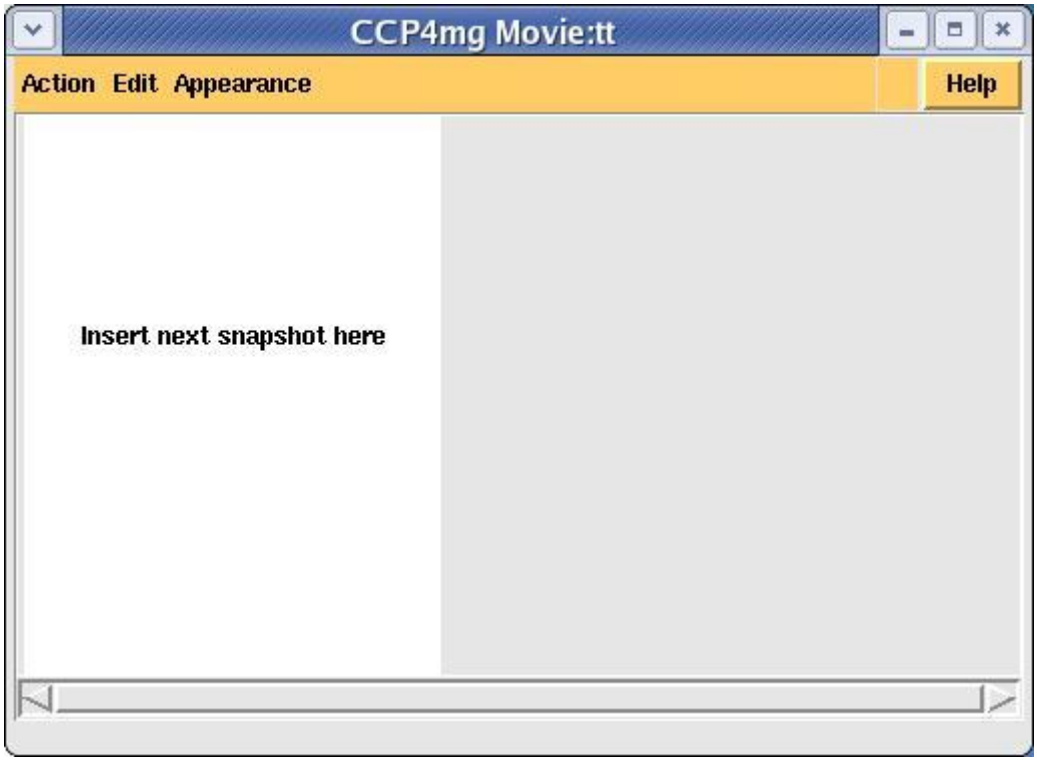
Clicking on one of the snapshots in interface caused the appropriate status file to be read and the program status shown in the snapshot is restored.

The snapshot and status files are saved in a directory which is given a name with the file extension *.ccp4mg_presentation*. To package up the presentation to send to someone else the application has a tool to copy all the data files used in the presentation to the *.ccp4mg_presentation* directory and to compress (zip) the directory.

Creating a Presentation

To begin pick **Presentation/Notebook** from the **Tools** pull-down menu. If you are creating a new presentation enter a name for it. For a presentation called *foo* CCP4mg creates a directory *foo.ccp4mg_presentation* in your current project directory. This directory will contain all of the snapshot and status files that are created.

The presentation interface initially looks like this



To load a snapshot, set up the display as you want it and then click on the **Insert next snapshot here** frame and select **Insert current view/display** from the popup menu. The current program status is saved to a status file and a small snapshot image is made and displayed in the presentation interface as shown in the Introduction.

Note that the first snapshot is labelled 00001 and subsequent snapshots will be labelled 00002, 00003 etc.. There is a text entry box under the snapshot in which you can enter a title for that snapshot.

A left mouse double-click on a snapshot picture will restore the status of that snapshot. A right mouse click on a snapshot picture will bring up a popup menu with options to
Restore the display and/or the view saved for that snapshot
Replace the snapshot with the current display status
Set insert to move the insert frame before or after the snapshot
Cut to remove the snapshot from the interface (but note that files are not deleted and can be restored using the **Load existing snapshot** option in the *Insert* frame).
Copy to save the snapshot to a buffer which can be **Paste**d in the Insert frame.

The Action pull-down menu

Next and **Previous** will load the status corresponding to the next or the previous snapshot.
Exit Exit the Presentation interface.

The Edit pull-down menu

Cleanup old files will delete any snapshot and status files that have been removed from the interface.

Package the presentation will collect all data files referenced in the presentations status files to the presentation directory and change the status files so that they now expect the data files to be in the presentation directory. Optionally compress the directory using a zip mechanism.

The **Preferences** for creating an html file are described above.

Viewing a Presentation

Note that the internal browser option idescribed below is only available on Linux systems.

Open a presentation using the **Open presentation** option on the **File** pull-down menu. If an *index.html* file exists for the presentation then it will be displayed in the internal browser; otherwise a similar Presentation interface to that show above is used. In the standard interface you can change the scene by:

- Selecting **Next** or **Previous** from the **Action** menu.
- Double clicking a snapshot with the left mouse button.
- Clicking a snapshot with the right mouse button and selecting a restore mode from the pop-up menu.

In the browser interface clicking on a snapshot in the html page will restore that status.

CCP4 Molecular Graphics Documentation

Geometry

Documentation Contents | On-line Documentation | Tutorials | CCP4mg Home

Contents

Introduction

This application contains tools for interactively querying the structure for distances, angles etc..

Generic options of side-panel applications

For all applications with a side-panel interface you can right-mouse click on the application name (*Geometry*) to give a pop-up menu with:
Hide/Show Hide or show the tools for the application. This can simplify the side-panel if you are not using Geometry for a while.
Close Remove the application completely from the side-panel.
Preferences Open a window with Preferences for the application.
Help Show the help page for the application.

Side-panel tools

When you click on a tool such as **Distances** it becomes the active tool and is highlighted in gold; clicking on it again or clicking on another tool will switch it off. While there is an active tool clicking on atoms and/or bonds with the left mouse button selects that atom or bond for the tool.

Distances
: Pick pairs of atoms for their inter-atomic distance to be shown on the display.

Distances/angles
: Click on a series of atoms for the distance, bond angle and torsion angle between atoms to be listed in the *Atom geometry* window. The distances between pairs of atoms are also shown on the display.

Clear contacts
: Clears all interatomic distances from the display.

Export Vectors
: Write the currently displayed interatomic distances to a *.vector* file.

Atom popup menu tools

While the geometry application is open the following tools are on the popup menu which appears when an atom is clicked with the right mouse button:

Geometry - first atom

Select the first atom for interatomic distance calculation

Geometry - next atom

Select the next atom and show the distance to the previous atom.

Contact around this atom/residue

Show close contacts around the picked atom or residue

Preferences

The Preferences window can be accesses by clicking on the side-panel title **Geometry** with the right-mouse button or it is in the **Applications** folder of the Preferences window.

You can set the **Maximum contact distance** and **Minimum contact distance** for the **Contact around this atom/residue** tool.
The **Line/label colour** can be changed

CCP4 Molecular Graphics Documentation

Web Tools

Documentation Contents | On-line Documentation | Tutorials | CCP4mg Home

Contents

Browser

A web browser is used to display help pages, screenshots after they have been created and Presentations. On Linux systems there is an internal browser based on HV3 or a default external browser is usually set up when first running the program:
Windows: the default hander for html files. This is found from the registry, looking for either 'HTTP','http' or 'https' in the 'HKEY_CLASSES_ROOT' part of the registry and extracting the 'shell/open/command' information.
Mac OSX: the 'open' command which will open the appropriate viewer for the file type
Linux: whichever of the browser: firefox,konqueror,mozilla or netscape, is found first

The **Preferences** option **Browser/web tools** (in the **Tools** folder) has a menu option listing possible browsers. If this does not work then it may be necessary to enter the full path name of the **Browser executable** which will then be used in preference to the menu selection.

The internal browser is always used for Presentations and external browsers are used for access to coordinate download sites. The Preferences window has a **Use internal browser** option which to use for help pages and screenshots. Note that the present version of the internal browser (CCP4mg 1.1) does not support Javascript and so is poor at presenting more sophisticated web sites.

The Preferences window also lists the URLs for the coordinate download sites - these are not expected to change but are there just in case.

CCP4 Molecular Graphics Documentation
History and Scripting

Documentation Contents | On-line Documentation | Tutorials | CCP4mg Home

Contents

Introduction

Recording and replaying commands can be useful to save time repeating a frequent command sequence and for making presentations. More sophisticated scripts can be used to extend the functionality of CCP4mg.
CCP4mg keeps a record of the commands that you enter and will display this history if you use the ***History and scripting*** utility on the ***Tools*** pull-down menu. You can select some or all of the command history to save to a file. This recording file can be replayed again later. If you need to modify the recording you do not need to manually edit it, there are options to delete commands or add new commands by picking them from the graphical interface.
The 'top' level of CCP4mg is written in the Python scripting language and you can replay Python scripts within CCP4mg. In many cases the easiest way to create a script will be to start from a recording made via the ***History and scripting*** utility. So, for example, if you want to apply some procedure to a range of molecules you should make a recording of applying the procedure to a single molecule and then edit the recording file to repeat for multiple molecules. Generally you will need some familiarity with Python to write or edit scripts but you may be able to do some simple things by following the examples below.

The ***History&Scripting***->***Replay script*** option on the ***Tools*** pull-down menu requires you to select a script file and has options:
Update display after each script line in this mode the script is executed one command at a time and after every command the display is updated. This will not work if the script contains control statements, loops etc.. If this is off the script is executed in one go and the display is not updated until the end of the script. If the***Update display..*** option is off it is possible to force a display update with an 'UpdateDisplay' command in the script.
Replay manually one line at a time The script is displayed in the interface window and replayed one line at a time as the user clicks the ***Do next*** button.

Recording

The ***History&scripting***->***Review & record*** command on the ***Tools*** pull-down menu will toggle open a window that, in the default **history** mode, lists all the commands you have entered in this session of the program with options to save some or all commands to a recording file. In the alternative **replay** mode a recording file is listed and can be replayed or edited.
For some options you will need to select one or more command lines from the window. Do this by clicking on a line with the left mouse button down. To select a range of lines first select one line and then click on the other end of the range using the left mouse button and with the Shift key down. The selected lines are highlighted by a gold background.

The options in history mode are

Clear selection	Clear any selection from the window
Replay Script	You will be prompted for the name of an existing script file which is replayed immediately and completely
Edit/replay stepwise	You will be prompted for the name of an existing script file and the History interface goes into replay mode
Begin recording	The button changes to ***End recording***. All commands that you pick from the GUI are marked to be recorded
End recording	You are prompted to enter the name of a file. All commands entered since ***Begin recording*** are saved to the file
Save selected lines	You are prompted to enter the name of a file. All selected command lines are saved to the file

Replaying and Editing

If you select the ***Edit/replay stepwise*** and enter the name of an existing recording file then the interface goes into replay mode and displays the contents of the recording file. One line of the recording is highlighted in bold lettering. It is the **next** line that will be replayed next by default. Lines in the recording can be selected and cleared the same as in history mode.

The options in replay mode are

Do next	Do the **next** command that is highlighted in the window
Go to line	Move the **next** line marker to the first selected line
Run continuously	Run the commands from the **next** line to the end of the recording and exit the replay mode
Quit	Exit from replay mode
Insert commands	Any commands picked from from anywhere on the interface will be entered into into recording before the **next** line.
Delete selected lines	Delete selected lines from the recording
Save to file	Save the current window contents to a file

CCP4 Molecular Graphics Documentation

Lighting

Documentation Contents | On-line Documentation | Tutorials | CCP4mg Home

Lighting

The position and nature of the light source(s) affect the appearance of solid objects such as surfaces, spheres, ribbons and cylinders. The lighting is particularly important if you are creating a ray-tracing picture using POV-Ray. When CCP4mg writes out an input script for POV-Ray it will specify the same light sources as are currently on in CCP4mg so roughly similar effects should be seen.

You can change between three pre-set lighting styles from the **View** menu.

The light source can be changed from the **Lighting** interface which is accessed via the **Display** folder of the **Preferences** window (on the **Tools** pull-down menu). By default there is one light source which is directly behind the user (don't ask how it manages to shine through your head!). Up to 8 light sources are possible. Each one has properties: ambient,specular and diffuse which can take values in the range 0 to 1. The light source can be positioned above, below, right, left or directly in front of the display. The effect of the different properties are probably best seen by changing the values but..
ambient light gives a completely flat, bright surface
specular light gives one bright patch on an otherwise unlit surface
diffuse light lightens all of the surface facing the light source
A combination of specular and diffuse light gives a good sense of depth and an attractive shiny surface.

The pre-set lighting styles use the first four or the eight light sources - the remaining four sources can be set to a customised style.

CCP4 Molecular Graphics Documentation

Saving program status

Documentation Contents | On-line Documentation | Tutorials | CCP4mg Home

Contents

Model Definition Files

CCP4mg saves additional information relating to a specific coordinate file to an extra file with the same name as the coordinate file but with the extension '.ccp4mg'. If you are reading a coordinate file from a directory for which you do not have write permission then no model definition file will be saved. The same information can also be saved to more generic files and this data can be applied to more than one coordinate file (see below).
Note that, for CCP4mg version 1.1 onwards the same information is also saved to a status file and the status file takes precedence over the model definition file. This is a change relative to earlier versions of CCP4mg to ensure that restoring a status file really does restore exactly what was originally saved. The model definition file is still effective if a model is loaded and some information is entered before the model is deleted. If the model is reloaded later the information will be restored.
The information currently saved is

- customised selection protocols saved by **Save current selection** on the Selection menu
- customised colour protocols saved from the **Edit Colour Scheme** window
- secondary structure definitions entered by the user from the **Structure definition** menu of the model icon menu (see Secondary Structure)
- residue type assignments entered by the user via **Residue type assignment** on the **Structure definition** menu see Reading Coordinate Files and Atom Typing

The file is in Python pickle format which is not really user readable. The file is read automatically when the coordinate file is loaded. If you move or rename the coordinate file then you need to make the equivalent changes to the *.ccp4mg* file.

When saving the selection or colour protocols you have the option to save for either the current coordinate file or for all models. If you save for all models the information is saved in the file *$HOME/.CCP4MG/generic.ccp4mg* and will always be available to restore for any model. A generic *.ccp4mg* file can be created from the **Model definition file** menu on the **File** pull-down menu. Choose to **Create new file** and enter a file name. When saving the selection or colour protocols you will also have the option to save to this file. This file is then loaded whenever CCP4mg is started unless you remove it with the **Remove** option on the **Model definition file** menu.

Program Status

There are currently three aspects of the program status which can be saved:

- User preferences as from the Preferences window
- The selected data and display objects as set in the Display Table

- The current view

Whenever the program status is saved then all three aspects of the status are saved to one file but restoring status may restore only some selected aspect.

The program status is saved automatically on program exit to a file:
$HOME/.CCP4MG/status/last_exit_status.pkl
and if there is an existing file of this name it is renamed:
$HOME/.CCP4MG/status/prev_exit_status.pkl

On restarting the program the user preferences are always restored from the status file but the selected data and view are only restored if the ***Restore data on restart*** option has been set in the ***Session options*** preferences.
Save status or ***Restore status*** options on the ***Tools*** pull-down menu will save the status to file or restore. You will be presented with a file browser to select a file.
Save view and ***Restore view*** on the ***Tools*** pull-down menu will similarly save the status and restore only the view from a status file.

In the ***Session options*** preferences there is also an option to ***Backup data files on exit***. If this is set on then on exit all loaded data files will copied into the directory *$HOME/.CCP4MG/saved_data_files*. Next time the program is run, if any of the data files required to restore status no longer exist, then they will be recreated from the saved data files and the user will be notified that this has been done.
Beware that on the next exit from the program the *saved_data_files* directory will be cleaned out before saving the currently loaded files.

Presentations

This tool will help if you want to use CCP4mg to make a presentation to an audience or show something in CCP4mg to colleagues. It is a nice interface to saving multiple program status.
See Presentations.

CCP4 Molecular Graphics Documentation

Picture Definition Files

Documentation Contents On-line Documentation Tutorials CCP4mg Home

Contents

Introduction

Picture definition files are human readable and editable files which define the scene displayed by CCP4mg; status files can also be used to save the scene information but these are not human readable. The information in the picture definition file corresponds to that presented in various interface windows:

Picture definition objects	**User interface equivalent**
Data objects (MolData, MapData etc.)	The display table
Display objects (MolDisp, MapDisp etc.)	The display table
SelectionScheme	Selection menu: Save current selection
ColourScheme	Colour menu: Edit colour scheme
ParamsManager	Preferences
Colours	Preferences: Display: Colours
View	The view - usually controlled by mouse input
Wizard	A picture wizard scene
Inline data	A coordinate file

An important use of picture definition files is by non-graphical programs which use a picture definition file to define some graphical information to present to users in CCP4mg. Another use may be if you have set up a complicated picture for one molecule and then want to do the same, or a similar picture, for another molecule; in this case it may be quicker to write out and edit a picture definition file than it would be to set up the picture again in the display table. But it may be even quicker to use the **Display same as** option on the model icon menu.

If you need to create a picture definition file the easiest way is to set up the display table as you would like it and then save a picture definition file which can be edited to the exact form that you require.

Picture definition files can be saved or read from the **Picture definition files** option on the **Tools** pull-down menu. The **Save picture definition** option will save the current display to a file. In the file selection window is an option **Only save non-default parameters** which is on by default and means that the minimum of information about each object in the display is saved. The appearance of the display is also affected by some of the program **Preferences** such as the Model Drawing Style - these will also be saved to the file if they are selected in the **Save options** interface (accessed from the **Picture definition files** sub-menu).

On reading a picture definition file there are options to **Keep currently loaded data** and to **Hide slow-to-draw objects such as surfaces**

CCP4mg can be started with a command line option which loads the picture definition file:

```
ccp4mg -picture filename.mgpic.py
```

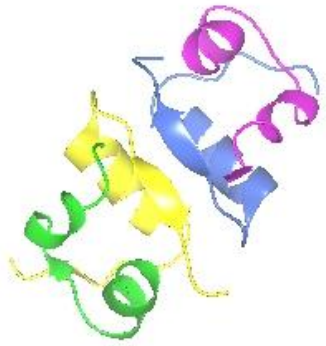
File Format

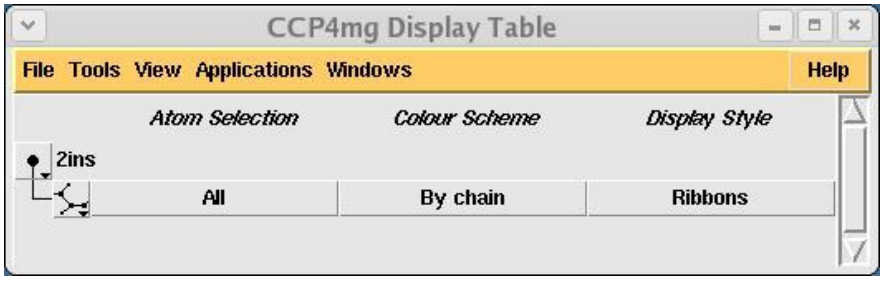
The picture definition file is a simple Python script. There are several benefits to using Python here:

- The Python parser is totally reliable and gives good error messages
- If you, the user, must learn a file format it might as well be something which could be more generally useful
- The scripting language has built-in flow control commands
- In future, additional functionality could be made accessible via the picture definition language

There is a quick introduction to Python here.

Here is a short example of a picture definition file which can recreate the picture and Display Table:






```
MolData (
     filename = ['demo', '2ins.pdb', '/home/lizp/demo_data/2ins.pdb']
      )

MolDisp (
          selection = 'all',
          colour = 'bychain',
          style = 'SPLINE'      )
```

Two objects are defined in this file (in strict Python terminology these are two *instances* of a *class*, see Python classes), and these correspond to the two lines in the display table. The two objects are:

- A **MolData** object is the data object corresponding to a model coordinate file. This has one attribute: the coordinate file **filename** (the syntax for this is explained later). It could have additional attributes, for example, to define symmetry mates or a transformation to apply to the coordinates.
- A **MolDisp** display object defines how the MolData object is displayed. This has attributes: **selection**, **colour** and **style** and could have additional attributes, for example, to define atom labels.

This file contains sufficient information to recreate the picture above. When the script is read it does not directly create the data and display objects but creates *shadow* objects which just hold the description of the object. If there are no problems reading the script then the information in the shadow objects is used to create the real MolData and MolDisp objects. This approach should help prevent faulty script files from corrupting the program.

This script relies on the program assuming that a display object belongs to the preceding data object. The connections between data objects and display objects can be made explicit as explained later.

Data Objects

The key information for any data object is the filename of the data file.

```
filename = ['demo', '2ins.pdb', '/home/lizp/demo_data/2ins.pdb']
```

The format for the file name is a list (Python list syntax) with three items:

- CCP4i project directory name (or the word 'FULLPATH' if the file is not in a project directory)
- file name (without the directory component)
- full pathname of the file

When reading a file CCP4mg first tries looking in the project directory for a file with the file name; if this does not exist it will try using the full pathname.

All data and display objects have a name which is a one-word text string that is used to reference the object. A data object name is usually derived from the name of the data file with the directory path and file extension stripped off. For the example above the data object name would be '2ins'. If there are multiple data objects with the same filename attribute then the second and subsequent objects are named *filename*1, *filename*2 etc.. The name of an object can be set explicitly, for example:

```
MolData (
     name = 'molecule_1',
     filename = ['demo', '2ins.pdb', '/home/lizp/demo_data/2ins.pdb']
      )
```

sets the name of the MolData object to 'molecule_1'.

Display Objects

If a data object is defined without any display objects then one display object with default attributes is created. Any display object attribute not defined in the script will be given a default value.

All display objects have three key attributes which are those controlled by the widgets in the three columns in the display table. The program always writes these attributes to the picture definition file. Additional attributes which are controlled from the icon menu in the display table are usually only written to the script if they have non-default values. Some properties of display objects, such as a **font**, are too complex to be described by a single parameter and so multiple parameters are grouped together in a Python data structure called a *dictionary*, for example in defining annotation:

```
Annotation (
             font =  {  'weight' : 'bold',
                        'slant' : 'i',
                        'family' : 'utopia',
                        'size' : '14'  },
             colour = 'complement',
             selection = 'This is annotation <br>',
             position = [11.52, 11.81, 15.56, '']          )
```

the font attribute is a dictionary with keys: weight,slant,family and size.

Some display objects have dictionary attributes such as *selection_parameters* that provides more details to complement the *selection* attribute. This feature is probably most important for display objects such as MolDisp and HBonds; for these objects the atom selection in the *selection* attribute is a selection rule which defines the required atoms but does not have sufficient information to initialise the user interface selection menu correctly. The additional information in *selection_parameters* will initialise the interface. For example a display object that displays the Calpha trace and two ligands

```
MolDisp (
          selection = 'catrace or //500(NDP) or //501(MTX)',
          colour = 'atomtype',
          style = 'BONDS',
          selection_parameters =  {
                         'select' : 'catrace',
                         'monomers' : ['//500(NDP)', '//501(MTX)']  } )
```

This has a *selection* attribute which is the selection rule which will select the Calpha atoms and the two named ligands. The *selection_parameters* attribute is a dictionary with two keys, 'select' and 'monomers', that contain the same information as the *selection* attribute but broken down in appropriate fashion to correctly initialise the selection menu on the display table. Beware that the *selection_parameters* will overrule the *selection* and so should be excluded from a script rather than be wrong. If the *selection* attribute is absent it will be recalculated from the *selection_parameters*.

SelectionScheme Objects

This creates a named selection which is equivalent to that created by the **Save current selection** on the display table selection menus. For example:

```
SelectionScheme (
          name = 'interesting_residues',
          context = '1abc',
          selection = '/1/A/10-20',
          selparams = { 'ranges' : [ '/1/A/10-20' ],
```

```
                        'select' : 'ranges'
                       })
```

The attributes are
name a one-word name for the scheme
context is either 'generic' or the name of a MolData object. If the attribute is not given then the SelectionScheme is assumed to be associated with the preceding MolData object.
selection which is a selection command
selparams which is a Python dictionary containing the status of the selection interface when this SelectionScheme was saved. Note that only the differences from the default initial status are saved. The selection interface will be restored to this status if the SelectionScheme is used.
When writing a picture definition file CCP4mg will write out generic SelectionSchemes that are currently used by a display object and all SelectionSchemes associated with a MolData object. Note that SelectionSchemes are also automatically saved to model definition files but the definition imported from the picture definition file will override any in the model definition file with the same name.

The pre-defined SelectionScheme can be used by a display object:

```
MolDisp (
          selection_parameters = {
                        'select' : 'selection_scheme',
                        'selection_scheme' : 'interesting_residues' },
          style = 'SPHERES',
          colour_parameters =  {
                       'colour_mode' : 'rules',
                       'user_scheme' : ['c_grey','1abc' ] )
```

In this example the *selection_parameters* for the display object have *select* set to 'selection_scheme' to indicate using a SelectionScheme and *selection_scheme* gives the name of the SelectionScheme.

ColourScheme Objects

ColourScheme objects work in similar fashion to the SelectionScheme objects but they create a named colour scheme which is equivalent to saving a colour scheme from the **Edit colour scheme** option on the model display colour menu. An example of defining a ColourScheme:

```
ColourScheme (
          name = 'c_grey',
          context = 'generic',
          colours = [['green', 'all'],
                     ['grey', '/*/*/*/*[C]:*'],
                     ['red', '/*/*/*/*[O]:*'],
                     ['blue', '/*/*/*/*[N]:*'],
                     ['yellow', '/*/*/*/*[S]:*'],
                     ['tan', '/*/*/*/*[H]:*'],
                     ['magenta', '/*/*/*/*[P]:*']] )
```

The ColourScheme attributes are:
name a one-word name for the scheme
context is either 'generic' or the name of a MolData object. If this attribute is not given then the ColourScheme is assumed to be associated with the preceding MolData object.
colours is a list of colouring rules; each rule is a list of two elements; the first element is a colour name and the second is a selection command. For each rule the atoms selected by a selection command are given the associated colour and the rules are applied in the order they are given - colouring by a later rule can override a preceding rule.

Note that ColourSchemes are also automatically saved to model definition files but the definition imported from a picture definition file will override any in the model definition file with the same name.

The pre-defined ColourScheme can be used by a display object:

```
MolDisp (
          selection_parameters = {
                        'select' : 'selection_scheme',
                        'selection_scheme' : 'interesting_residues' },
          style = 'SPHERES',
          colour_parameters =  {
                       'colour_mode' : 'rules',
                       'user_scheme' : ['c_grey','1abc' ] )
```

In this example the *colour_parameters* for the display object have *colour_mode* set to 'rules' to indicate using a ColourScheme and *user_scheme* gives the name of the ColourScheme which is a list of two elements; the first element is the ColourScheme name and the second is the name of the model data object that the ColourScheme is associated with.

ParamsManager Objects

The ParamsManager object is used to hold various user preferences. The most useful preferences are probably those for the model drawing style which corresponds to the preferences for **Drawing style** in the **Model display** folder or, for individual display objects, **Molecule drawing style..** on the display object icon menu. An example of the picture definition format:

```
ParamsManager(
          name = 'model_drawing_style',
          arrow_width = 1.2,
          flatten_loop = 1,
          helix_style = 1,
          worm_width = 0.3,
          ribbon_style = 1,
          alpha_helix_width = 1.0          )
```

Note that the ParamsManager does not need to have all the parameters associated with that type ParamsManager defined; any missing parameters will take default values. The ParamsManager must have a name parameter which indicates the type of data contained. The generic model drawing style is called 'model_drawing_style' and customised drawing styles for individual display objects should be called something like 'model_drawing_style_*xxxx*'. This is associated with the display object by the *drawing-style* parameter of the display object. For example:

```
MolDisp ( selection = '//A',
          colour = 'secstr',
          style = 'SPLINE',
          drawing_style = 'model_drawing_style_my_ribbon' )

ParamsManager(
          name = 'model_drawing_style_my_ribbon',
          arrow_width = 1.2,
          flatten_loop = 1,
          helix_style = 1,
          worm_width = 0.3,
          ribbon_style = 1,
          alpha_helix_width = 1.0          )
```

The Colours Object

This corresponds to the **Preferences** option **Colours** in the **Display** folder. An example of a picture definition Colours object:

```
Colours (
   colour_definitions = [
         ['white', [0.0, 0.0, 0.0, 1.0]],
         ['olive', [0.4, 0.6, 0, 1.0]]] ] )
```

This redefines white (to be black) and defines a new colour, 'olive'. The format of input is that Colours has an attribute *colour_definitions* which should be a list of colour definitions where each element of the list is another list with two elements. The first element is a text string which is the name of the colour and the second element is another list of four floating point numbers which are the RGB values (as values in the range 0.0 to 1.0) and an 'alpha' value which should be 1.0. For explanation of RGB see Wikipedia.

Wizard Objects

An alternative approach to specifying display objects is to specify a Picture Wizard script to be used to generate the display objects automatically. There is a library of Picture Wizard scripts or you can create your own. Most scripts will work without any additional input; some take optional input, usually of atom selections, and some scripts require atom selection input. The scripts absolutely requiring some selection input are those for displaying interfaces where two sets of atoms must be defined. An example of using A Wizard object:

```
MolData (
       filename = ['FULLPATH', '1ins.pdb', '/home/lizp/demo_data/1ins.pdb']
      )

Wizard ( drawing_style = 'interfaces:residues',
    range1 = '/1/A',
    range2 =  '/1/B or /1/C or /1/D' )
```

The Wizard object is immediately after the MolData object that it will be applied to. Wizard objects should always have a *drawing_style* parameter which is the Picture Wizard folder and the script name separated by a colon. The most common additional parameters are *range*n or *ligand*n which specify sets of atoms. The allowed parameters for each template are shown in the template file in the CHOICES section.

Inline Objects - Including Data in the File

This is a mechanism for including data in a picture definition file. It has currently only been tested for including PDB

coordinate file data. An example of defining an inline object:

```
Inline (name =  'pisa_t1.pdb' , data = '''
ATOM      3  N   VAL D   2      14.250  10.141  -6.104  1.00 45.00           N
ATOM      3  CA  VAL D   2      15.003  10.746  -7.205  1.00 45.00           C
ATOM      3  C   VAL D   2      14.112  10.940  -8.441  1.00 30.52           C
ATOM      3  O   VAL D   2      13.430  11.963  -8.554  1.00 45.00           O
ATOM      2  CB  VAL D   2      16.311  10.009  -7.532  1.00 32.25           C
ATOM      2  CG1 VAL D   2      16.034   8.920  -8.586  1.00 25.32           C
ATOM      3  CG2 VAL D   2      17.343  11.002  -8.105  1.00 45.00           C
ATOM      1  N   ASN D   3      14.095   9.983  -9.372  1.00 20.04           N
ATOM      3  CA  ASN D   3      13.227  10.230 -10.540  1.00 45.00           C
....

ATOM      6  ND2 ASN C  21      -2.497  21.308   3.598  1.00 36.48           N
ATOM      5  OXT ASN C  21       1.431  24.714   2.915  1.00 45.00           O
''' )
```

The data will be written to another data file called *name* and placed in the users current project directory. Any MolData object that needs to reference that file should have define the filename in the form:

```
MolData (
     filename = ['INLINE', 'pisa_t1.pdb', '']
      )
```

The word *INLINE* is used instead of any project or path.

The View Object

The view object defines the centre of the screen, the view orientation and the zoom, for example:

```
View (
     centre_xyz = [-6.78, -17.32, -0.62],
     zoom = 0.51,
     orientation = [0.572, 0.209, -0.517, 0.599]     )
```

The *centre_xyz* is the coordinates to appear at the centre of the screen. An alternative way to specify the screen centre is by selecting a model and, optionally, some atom(s) in the model. For example:

```
View (
     centre_MolData= '1abc',
     centre_selection = '//A/27',
     zoom = 0.51,
     orientation = [0.572, 0.209, -0.517, 0.599]     )
```

The *centre_MolData* attribute is the name of the model. *centre_selection* is optional and is a selection command.

The *orientation* is represented as a quaternion. The *zoom* and *orientation* are not easy to generate outside of molecular graphics.

CCP4 Molecular Graphics Documentation

Picture Definition File Object Attributes

Documentation Contents | On-line Documentation | Tutorials | CCP4mg Home

Contents

Introduction

This document lists the attributes associated with each type of data object and display object for the benefit of anyone writing or editing a picture definition file. Note that, rather than create a picture definition file from scratch, it is probably much easier to set up a display within CCP4mg, save a picture definition file and edit it.

There are some attributes which are common to all data objects and display objects and these are described first.

Data Object Attributes

attribute	key	data type	default	description
name		string	Derived from data filename	Must be one word and unique
filename		list of three strings	["","",""]	See below
visible		logical (0 or 1)	1	If 0 then all display objects are not displayed

Display Object Attributes

attribute	key	data type	default	description
name		string	Derived from name or parent data object	Must be one word and unique
visible		logical (0 or 1)	1	If 0 then objects is not displayed
transparent		logical (0 or 1)	0	If 1 object transparent with *opacity*
opacity		float range 0.0 to 1.0	1.0	0.0 is totally transparent, 1.0 is totally opaque

MolData

attribute	key	data type	default	description
transform_matrix		list of 16 floats	`[1.0,0.0,0.0,0.0, 0.0,1.0,0.0,0.0, 0.0,0.0,1.0,0.0, 0.0,0.0,0.0,1.0 ]`	coordinate transformation see here
load_animation		dictionary		Details of loaded animation see here
	filename	list of three strings	["","",""]	For *file_format* 'multiple_PDBs', first of PDB files defining trajectory.See below.
	file_format	string	'multiple_PDBs'	'multiple_PDBs' or 'NMR_models'
	frame_range	list of 2 integers		First and last frame; reset on loading trajectory
	current_frame	list of 2 integers	[-999,1]	The current frame number (or -999 for unset) and the increment for next frame (1 or -1)

	legend_name			
DSSP_file		list of three strings	["","",""]	A DSSP file containing secondary structure assignment
symmetry				

MolDisp

attribute	key	data type	default	description
selection		string	'all'	a selection command
selection_parameters		dictionary		Selection details for consistent GUI. See below
colour		string	'atomtype'	colour scheme see below
colour_parameters		dictionary		Colouring details.See below
style		string	'BONDS'	drawing style
style_parameters		dictionary		Drawing style details.See below
drawing_style	string	The name of a model drawing style parameters manager associated with this display object		

***selection_parameters* keys**

selection_parameters is a dictionary attribute of a MolDisp object which should provide sufficient information to correctly initialise the selection menu on the MolDisp GUI. The *selection* attribute, on its own, is enough to define which atoms are displayed. The most important item in the dictionary is the *select* key which indicates which selection mode is being applied. Dependent on the value if the select key, various other items will be necessary in the dictionary.

key	*select* value	data type	description
notselect	Any value	logical (0 or 1)	If 1 then display all atoms but those selected.
ranges	'peptide', 'main', 'side' or 'catrace'	list of strings	Each item is residue range e.g. '//A/25-47'.
secstr	'peptide', 'main', 'side' or 'catrace'	list of strings	Each item is secondary structure type ('SSE_None', 'SSE_Strand', 'SSE_Bulge', 'SSE_3Turn', 'SSE_4Turn', 'SSE_5Turn', 'SSE_Helix')
restype	'restype'	list of strings	Each item is a residue type (e.g. 'ALA' or 'T')
elementtypes	'elementtypes'	list of strings	Each item is an atomic element (e.g. 'C')
property	'property'	list of strings	Each item is a property ('B', 'OCC', 'X', 'Y', 'Z', 'ATOM_SAS', 'RES_SAS', 'ATOM_CONTACT', 'RES_CONTACT', 'SEC').
property_lt_value	'property'	list of floats	Lower limit to property value. List must be same length as *property* list.
property_lt_op	'property'	list of strings	Less than operator ('<' or '<='), must be same length as *property* list
property_gt_value	'property'	list of floats	Upper limit to property value. List must be same length as *property* list.
property_gt_op	'property'	list of strings	Greater than operator ('>', '>=' or '=='), must be same length as *property* list.
neighb_mol	'neighb'	string	MolData object >*name* for the central atoms - default is the parent MolData
neighb_sel	'neighb'	string	a selection commands command for the central atoms
neighb_rad	'neighb'	float	selection radius
neighb_hb	'neighb'	logical (0 or 1)	If 1 select only those groups which HBond to central atoms
neighb_group	'neighb'	string	Group of atoms to select: 'atom', 'residue', 'chain', 'model', 'catrance', 'main', 'side', 'main_side', 'solvent'
neighb_excl	'neighb'	logical (0 or 1)	If 1 then exclude the central atoms
neighb_excl_solvent	'neighb'	logical (0 or 1)	If 1 exclude solvent
neighb_excl_monomer	'neighb'	logical (0 or 1)	If 1 then exclude atoms in momomers
neighb_excl_peptide	'neighb'	logical (0 or 1)	If 1 exclude peptide bond atoms
neighb_selobj	'neighb'	string	A MolDisp *name*; alternative to *neighb_sel* for selecting central atoms - only used if *neighb_sel* is undefined.
neighb_and	'neighb'	logical (0 or 1)	If true combine the main neighbourhood selection by a logical operation 'and' with *neighb_and_selection* or *neighb_and_selobj*
neighb_and_selection	'neighb'	string	A selection command. If *neighb_and* is true then only the neighbourhood atoms that are also in this selection are included in the final selection.
neighb_and_label	'neighb'	string	A simple label describing the *neighb_and_selection* which will appear in the GUI.

neighb_and_selobj	'neighb'	string	>A MolDisp *name*; alternative to *neighb_and_selection* for selecting the central atoms.
buried_sel	'buried'	string	a selection commands command for a group of atoms that may exclude solvent (i.e. bury) the atoms in selection *buried_and_selection*
buried_and_selection	'buried'	string	A selection command. Find the atoms in this selection that are buried by the atoms in *buried_sel*. This must be defined.
buried_and_label	'buried'	string	A simple label describing the *buried_and_selection* which will appear in the GUI.
buried_and_selobj	'buried'	string	>A MolDisp *name*; alternative to *buried_and_selection*. Only used if *buried_and_selection* is undefined.
cid	'cid'	string	cid=coordinate id. A selection command that should be the same as the MolDisp *selection*
selection_scheme	'selection_scheme'	string	The name of a saved selection.

colour_parameters

colour_parameters is a dictionary attribute of a MolDisp object which contains information in addition to the *colour* attribute. No *colour_parameters* is necessary if *colour* has one of the following values: 'atomtype', 'restype', 'bychain', 'mainside', 'secstr', 'bvalue', 'occupancy', 'charge', 'res_sas','atom_sas', 'thru_chain', 'bymodel', 'bymol'.
Within the program there are two ways to customise these colour schemes and the customisation can be written to the picture definition file:
The details of these colour schemes (which will be applied to all the loaded models) can be changed in the **Preferences** window (see in the **Model colours** folder). By default the colour scheme details are not written out to the picture definition file by CCP4mg but the **Save options** option on the picture definition sub-menu can be used to select colour scheme to save.
To customise a colour scheme for an individual display object use the **Edit colour scheme** option on the colour menu. The customised scheme is saved as the *colour_rules* item of *colour_parameters*.

key	***select* value**	**data type**	**description**
colour_mode	Any value		Should have same value as *colour*
one_colour	'one_colour'	string	A recognised colour name
stick_colour	Any value	string	The stick colour in ball-n-stick. A recognised colour name or 'ATOM'
non_C_atomtype	Any value	logical (0 or 1)	If 1 colour non-C atoms by their element type
colour_rules	'colour_rules'	list of [colour,selection]	Each item is a list of two items: a recognised colour and a selection rule
colour_blend	'blend'	dictionary	See below for dictionary items
contact_MolDisp	'res_contact' or 'atom_contact'	string	The name of another MolDisp
user_scheme	'user_scheme'	string	Name of saved colour scheme

Below are the items in *colour_blend* dictionary. The items *ranges_first*, *ranges_last*, *colour* and *direction* are lists which must be the same length.

key	**data type**	**description**
type	string	Apply to type of model: 'amino_acid', 'nucleic' or 'all'
default_colour	string	A recognised colour name
group	string	all atoms in 'residue'/'SSE' are same colour
ranges_first	list of strings	Each item in list is coordinate id of first chain/residue in range
ranges_last	list of string	Each item in list is coordinate id of last chain/residue in range
colour	list	Each item is list of two recognised colour names for start/end of range
direction	list of strings	Each item is direction around colour wheel: 'clockwise' or 'anti-clockwise'

style_parameters

key	**data type**	**description**
inter_dispobj_bond	logical (0 or 1)	If 1 (the default) bonds will be drawn to atoms in other MolDisp objects

SurfaceDispObj

Displays a surface over selected atoms in the MolData object.

attribute	**data type**	**description**
selection	string	Same as MolDisp.selection. A selection rule
selection_parameters	dictionary	Same as MolDisp.selection_parameters
colour	string	Same as MolDisp.colour with extra options: 'electrostatic'
colour_parameters	dictionary	Same as MolDisp.colour_parameters
style	string	A selection command for the context atoms
style_parameters	dictionary	Same as MolDisp.selection_parameters with extra *select* option: 'same_as'.
surface_style	string	Surface style: 'solid','dots', or 'mesh'
drawing_style	string	The name of a Surface drawing style parameters manager associated with this display object

HBonds and Contacts

HBonds and Contacts are displayed for the parent MolData object. These object types have very similar attributes which are listed below.

attribute	data type	description
selection	string	Same as MolDisp.selection. A selection command
selection_parameters	dictionary	Same as MolDisp.selection_parameters
colour	string	a recognised colour name
style	string	A selection command for the context atoms
style_parameters	dictionary	Same as MolDisp.selection_parameters with extra *select* option: 'same_as'.
inter_model	logical (0 or 1)	If 0 (the default) do not draw bonds between NMR models
line_width	integer	Width of lines: 1,2,3,4, or -1 (=cylinders)
label_mode	logical (0 or 1)	If 1 label the bonds with bond length

Annotation

Annotation is arbitrary text positioned on the MolData object. See tutorial.

attribute	data type	description
text	string	Text string for annotation
colour	string	a recognised colour name
position	list [float,float,float,string]	Position x,y,z and a selection commands. Position is centre of atoms selected by selection commands (if it is set) with x,y,z offset.
font	dictionary	A font

AnisoU

This object display thermal ellipses. See tutorial.

attribute	data type	description
selection	string	The name of a MolDisp object. Thermal ellipses drawn for the selected atoms of the MolDisp.
colour	string	a recognised colour name
style	string	drawing style:'Solid', 'Axes' or 'Solid&Axes'
scale	float	Scale the ellipses (default=1.0)
quality	int	Drawing quality: 1 (default)=fast, 2=smooth

MapData

The data object for electron density maps loaded from MTZ structure factor file or a CCP4 map file. For MTZ files the map is calculated from the specified data columns. Also has general data object attributes.

attribute	data type	description
filetype	string	'MAP' or 'MTZ' filetype of loaded file
f	string	Structure factor column label in MTZ file
phi	string	phase column label in MTZ file
wt	string	Optional weight column label in MTZ file
rate	string	For MTZ file only. Sampling rate i.e. map resolution in Ångström.

MapDisp

The display object for electron density maps; displays one contour level in variety of possible styles. Also has general display object attributes.

attribute	data type	description
radius	float	Size of the displayed block of map
clip_mode	string	Map clipping mode. 'SELECTION': clip to selected display object. 'POINT':clip around point. 'OFF': no clipping.
clip_MolDisp	string	For *clip_mode* 'SELECTION'. The name of a MolDisp. The map is clipped to the atoms in the model display object.
atom_clip_radius	string	>For *clip_mode* 'SELECTION'. Clip map to this distance from atoms in *clip_atoms*.
clip_point	list of floats	For *clip_mode* 'POINT'. The x,y,z coordinate of centre point.
colour	string	A recognised colour name
contour_level	float	The contour level in absolute electron/Å**3
surface_style	string	The surface drawing style: 'chickenwire','solid','solid_chickenwire','dots','slice' or 'mask'

Crystal

A crystal object controls the crystal and symmetry display of the associated map and model(s). To enable the usual symmetry options a picture definition file should contain one crystal object for each MapData object.

attribute	data type	description
source	string	Data object name (MapData or MolData)
models_in_crystal	list of strings	List of MolData; the models which should be treated as part of the crystal.
style	string	The style of symmetry display: 'CONTINUOUS': continuous Crystal, 'UNITCELL': contents of unit cell, or 'OFF'
symm_diff_colour	logical (0 or 1)	If 1 colour carbon atoms of symmetry mates in different colours

cell_edges	logical (0 or 1)	If 1 display the cell edges
unit_cell_shift	list of three logicals(0 or 1)	If *style* is 'UNITCELL' show contents of adjacent unit cells in x,y, or z if first, second or third item in list is 1.

2D Objects

Images and legends are grouped together on the GUI Display Table and are treated as display objects.

Image

A 2D image loaded from an picture file such as *.png* or *.jpg*.

attribute	data type	description
filename	list of strings	picture file filename. See below
scale	float	Scale the picture size.
aspect	float	Aspect ratio x/y
x	float	Translation, as fraction of screen size, across screen from left side, of left side of image
y	float	Translation, as fraction of screen size, up screen from bottom, of bottom of image

Legend

A 2D text string displayed on the screen. See tutorial.

attribute	data type	description
selection	string	A text string to be displayed
x	float	Translation, as fraction of screen size, across screen from left side, of left side of text
y	float	Translation, as fraction of screen size, up screen from bottom, of bottom of text
text_colour	string	A recognised colour name
font	dictionary	>A font

VectorsData

The data object for generic vectors loaded from vectors file.

VectorsDispobj

A display object for generic vectors. The display object attributes can also be set in the vectors file.

attribute	data type	description
selection	list of strings	List of tags to be displayed
line_parameters	dictionary	elements of dictionary as described for vectors file
label_parameters	dictionary	elements of dictionary as described for vectors file
font	dictionary	>A font
scaling_mode	int	0=no scaling, 1=scale all vectors by 'UNTAGGED' scale, 2=scale by tags
scale	dictionary of floats	Dictionary elements are 'UNTAGGED' and tag names. Values are scale factors to apply to all or just the tagged vectors.

ParamsManager

A generic ParamsManager class is used in CCP4mg to handle many of the Preferences interfaces. The ParamsManagers are listed in the **Save options** interface to be selected for output to the picture definition file. An example of a *ParamsManager* in a picture definition file:

```
ParamsManager(
          name = 'surface_drawing_style',
          resolution_delta = 20.0,
          dot_spacing = 0.01,
          blend_edges = 1          )
```

The ParamsManager should always have a *name* attribute which identifies which set of preferences it handles. Some types of data object or display object can have a ParamsManager associated with then to handle the options specifically for that object. For example a SurfaceDispObj can have an associated 'surface_drawing_style' ParamsManager which has the same attributes as the global 'surface_drawing_style' ParamsManager and has a name which conventionally is 'surface_drawing_style_*SurfaceDispObj name*'. To tie the display object to the ParamsManager the SurfaceDispObj object needs to have an attribute *drawing_style* with the value of the ParamsManager name.

ColourSchemeManager

A generic ColourSchemeManager object is used to handle the colour schemes which can be accessed in the **Model colours** folder of the **Preferences** interface. The attributes of ColourSchemeManager are:

attribute	data type	description
name	string	name of colour scheme:'atomtype','restype','secstr','bvalue','occupancy','charge','atom_sas','res_sas','atom_contact','res_contact'
ranges	list	list of strings for *name* is 'atomtype','restype','secstr' or list of floats
colours	list of strings	List should be same length as *ranges* and values are recognised colour names

interpolate_mode	string	Relevant if *ranges* are floats. 'RGB' or 'HSV'
colour_wheel_direction	string	If *interpolate_mode* is 'HSV' then interpolation around colour wheel is 'clockwise' or 'anti-clockwise'

For the colour schemes that have float values in the ranges (everything but 'atomtype','restype','secstr') the first and last colour are the warning colours for any atom with property values below or above the normal range and the first and last value in the ranges list are not actually used. See the colour schemes documentation.

Colours

The standard colour names that are always defined within CCP4mg: blue, red, green, grey, yellow, magenta, royal blue, cyan,coral,pale green,pink,lemon,purple,tan,black. The colour name 'complement' will give the colour which complements the background colour (e.g. for black background gives white).

There is an option on the Picture Definition File **Save options** interface to save **Colour definitions**. To define a new colour in the picture definition file looks like this:

```
Colours ( yuck = [ 0.5 , 0.5, 0.5 , 1.0 ] )
```

The arguments to *Colours* are the names of the colours with values that are a list of four floats. The first three items in the list are the proportion of red, green and blue components in the colour in the range 0.0 to 1.0. The forth value is an 'alpha' value which should be set to 1.0 (or can be excluded). It is possible, but possibly anti-social, to redefine the standard colours is similar format:

```
Colours ( red = [ 0.0, 1.0, 0.0 , 1.0 ],
          green = [ 0.0, 0.0, 1.0, 1.0],
          blue =  [ 1.0, 0.0, 0.0 , 1.0 ]  )
```

View

The View attributes can determine the orientation of the display. The attributes are:

attribute	data type	description
centre_xyz	list of floats	The x,y,z coordinate to be placed at the centre of the screen
centre_MolData	string	The name of a MolData object to be placed in the centre of the screen
centre_selection	string	A cid (coordinate id) for a set of atoms to be placed in the centre of the screen (*centre_MolData* must also be used to specify model that cid applies to)
radius	float	Controls the 'zoom'

Notes on some common attributes

Filenames

The format for the file name is a list (Python list syntax) with three items:

- CCP4i project directory name (or the word 'FULLPATH' if the file is not in a project directory)
- file name (without the directory component)
- full pathname of the file

When loading the file CCP4mg first looks in the project directory (unless it is 'FULLPATH') for a file with the 'file name'; if this fails then it will try the full pathname.

Fonts

Several types of object have a font attribute. It is a dictionary and all the items have values which are strings. CCP4mg can only support the font families that are available on the computer system on which it is run. Beware that available fonts are very dependent on operating system. The dictionary keys are

key	description
family	Font family name
size	A string containing integer number (range 8-60 usually available in CCP4mg)
slant	'i' (italic) or 'r'
weight	'medium' or 'bold'

CCP4 Molecular Graphics Documentation

Python Scripting

Documentation Contents | On-line Documentation | Tutorials | CCP4mg Home

Contents

Introduction

The picture defintion file written out by CCP4mg uses a simple limited set of Python features which are described below in the Basic Python section. Some more features which might be required to create a more sophisticated script are described under Advanced Features. If you can not find what you need in the documentation or you need a little help with picture definition scripts please contact the developers ccp4mg@ccp4.ac.uk.

There is plenty of documentation and tutorials at python.org.

Python Essential Reference by David M. Beazley is a good, concise introduction. It may be too concise for novice programmers who might prefer...

File editors such as emacs can work in Python mode to colour text according to its role in the Python syntax. This can be very helpful when editing a file.

Basic Python

Left-hand indentation

For simple scripts all lines should begin at the left margin with no spaces or tabs etc.. The error message:

```
Invalid syntax in line
```

may be due to wrong indentation. In more complex scripts Python uses left-hand indentation to indicate the grouping of lines for loops and conditional statements.

Comments in lines

Everything on a line after a hash # is treated as a comment and ignored by the program. This does not apply to hashes within quotes.

Continuation lines

Commands may be split over two or more lines by terminating the first line with a line-continuation character ****. Beware any spaces after the continuation character will also result in a syntax error message. The line-continuation character is unnecessary if the break is between anything enclosed in parentheses (...), brackets [...] or braces {...}. Note that in the documentation examples and in the picture defintion files written by CCP4mg the items within parentheses or braces are ofter split one item per line without any line-continuation character.

Numbers

A number specified without a decimal point such as 789 will be treated as an integer and a number such as 789.12 will be treated as a floating point number. In most context in the picture defintion file it does not matter if integer and floating point numbers are interchanged but try to use the appropriate form.

Logical true or flase

True or or false are indicated by the integer 1 (true) or 0 (false).

Strings

Text strings can be enclosed in single (') or double (") quotes. Triple quotes (''' or """) can be used to enclose text that includes newlines and single or double quotes. For example

```
label = '''This is O2'
               or O2* if you prefer'''
```

Lists

A list is enclosed in square brackets with comma separated elements. The elements in a list can be a mixture of integer, float or strings for example:

```
mylist = [ 'hello', 789, 3.14 ]
```

Within the picture definition file the obvious use of a list is in specifying a file:

```
filename = ['demo', '2ins.pdb', '/home/lizp/demo_data/2ins.pdb']
```

The filename list has three string items. The order of the elements in a list is important. To access one particular element in a list use the syntax:

```
filename[1]
```

The index of the required element is given in square brackets immediately after the identity of the list. There is a big catch here: in many programming languages, inclding Python, the first item in a list is considered to be 0 (zero). So the items in the filename list have indices 0,1 and 2 and in the example above *filename[1]* has the value '2ins.pdb'.

Dictionaries

A dictionary is a group of items like a list but in a dictionary each item has a name (called a key) and the order of items is unimportant. For example, a dictionary is used to define a font:

```
font =  {  'weight' : 'bold',
           'slant' : 'i',
           'family' : 'utopia',
           'size' : '14'  }
```

The dictionary definition is enclosed in braces. Within the picture definition file the keys are always one-word strings which are enclosed in quotes (eg 'family' or 'weight'). The value of each item can be an integer, a float, a string or a list and these can be mixed within any one dictionary (note that the values in the font example all happen to be strings). Note that a colon (:) and not an equals (=) separate the key and the value.

Objects = Class instances

In CCP4mg documentation we usually talk about 'objects' and 'types of object' such as MolData or HBonds. In Python terminology the 'types of object' are called 'classes' and the individual objects of a given type are termed 'instances of a class'. For now I will continue to use the CCP4mg terminology of 'object' and 'type of object'. An example of the syntax to create an object in the picture defintion file:

```
Legend (  text = 'This is a legend <br>',
          y = 0.29,
          x = 0.3,
          font =  {      'weight' : 'bold',
                         'slant' : 'i',
                         'family' : 'utopia',
                         'size' : '14'  },
          text_colour = 'complement'          )
```

The new object will be a *Legend* object and the *text*, *x*, *y*, *font* and *text_colour* attributes are defined for the object. Note that the attribute names are not in quotes and the attribute name is separated from its value by an equals sign. This syntax is different from that of dictionaries. One of the attributes for this object is *font* which is a dictionary which follows the rules for dictionary syntax.

Advanced Features

Object pointers and methods

The identity of an object can be thought of as a pointer to the objects place in the computer memory. In the simple scripts we have looked at so far when an object is created its identity is not saved. In more complex scripts we may needed to ask the object questions about itself or give it commands and to do this we need to save its identity when it is created. For example you may want to find out what chains and monomers are in the data associated with a model data object:

```
model_id = MolData (
             filename = [ 'demo', '2ins.pdb', '/home/lizp/demo/2ins.pdb']
           )
chain_list = model_id.get(mode='chains')
monomers_list = model_id.get(mode='monomers')
```

In this example *model_id* is the name (or identity) of the MolData object, i.e. its place in memory. This name can be used with the appropriate methods for that type of object to get information or to do something to the object. In the example the name 'model_id' is used with the *get* method to get some information. It is used twice, first to get the list of chains in the model and, secondly, to get a list of monomers. Note the syntax used is the object name and the method ('get') separated by a dot. After the method are arguments in parentheses. The syntax for the arguments is the same as for specifying the object attributes (see Objects above) i.e. the name of the argument (not in quotes), an equals sign, and the value for each argument. The 'get' method has one argument called 'mode', the value of this determines what information is returned. The returned information is saved with the names 'chain_list' and 'monomers_list'. The names model_id, chain_list and monomers_list are arbitary; you can use any name that you like for the object or the information with a few limitations given in the next section.

Identifiers

There are a few limitations on the identifiers (i.e. names of objects or data etc.). Identifiers must be one word of either alphanumeric characters or underscores. The first character can not be a number.

Loops

Sometimes rather than repeat identical blocks of script multiple times in a file it is better to have one block of script and to 'loop' through it multiple times. For example to load three different pdb files but to write the appropriate commands only once:

```
pdbname_list = ['1xxx.pdb', '2xxx.pdb', '3xxx.pdb']

for pdbname in pdbname_list:
  MolData( filename = [ 'MYPROJECT' , pdbname, '' ] )
```

First in this script a list called 'pdbname_list' is defined; it is a list of three pdb files - note that these are text strings so they are in quotes. Then the statement 'for pdbname in pdbname_list:' says to do the subsequent script for each 'pdbname' in the list 'pdbname_list'. Note that the statement ends with a colon and the subsequent line(s) that will be repeated each time round the loop are indented (usually by 3 characters). In this example there is only one line inside the loop which creates a MolData object with a filename whose project directory is 'MYPROJECT' and whose file name is whichever pdbname is set for the time around the loop. Note that the pdbname used in specifying the filename is not in quotes!

Conditional statements

Sometimes what you want to do is dependent on some factor; for example when laoding a PDB file the required display objects may depend on whether there are any ligand monomers as in the following script. Note the use of hashes to start 'comment lines'.

```
# A short list of PDB files that can be found in the
# ccp4mg/tutorial/data directory
pdbname_list = ['1df7.pdb', 'rnase.pdb']

for pdbname in pdbname_list:

    model_id = MolData(filename = [ 'ccp4mg_tutorial' , pdbname, '' ] )

    #Create one display object to show the protein ribbons
    MolDisp ( colour = 'bychain',
              style = 'SPLINE',
              selection_parameters = {
                         'select' : 'amino_acid' }
            )

    # If there are any monomers display them as ballnstick
    monomers_list = model_id.get(mode='monomers')
    if len(monomers_list)>0:
       MolDisp (  colour = 'atomtype',
                  style = 'BALLSTICK',
                  selection_parameters = {
                     'select' : 'nopeptide',
                     'monomers' : monomers_list }
                )
```

In this script a short list of pdb files are defined as pdbname_list and then we loop over ever pdbname in that list. For each pdbname a MolData object is created and a MolDisp object

which shows the 'amino_acid' as ribbons. We then query if there are any monomer ligands using the *get* method. Then comes the conditional part of the script. We use the function *len* (short for length) to find out how long the monomers_list is. If the length of monomers_list is greater than zero then we create another MolDisp object to show the monomers.

Another trivial use of conditional statements that could be tagged on the end of the previous script to write the number of monomers found:

```
if len(monomers_list)< 1:
   print pdbname,"has no ligands"
elif len(monomers_list)== 1:
   print pdbname,"has one ligand:",monomers_list[0]
elif len(monomers_list)== 2:
   print pdbname,"has two ligands:",monomers_list[0],'and',monomers_list[1]
elif len(monomers_list)> 2:
   print pdbname,"has lots of ligands",monomers_list
```

In this this example the first test is if the length of the monomer list is less than 1; then *elif* (short for *else if*) is used make further tests (is the length of the monomer list equal to 1 or 2 or is it more than 2) Note that the *elif* test will only be applied if the preceding tests have failed. If a test is successful then the appropriate lines of script after the test are executed. In this example the *print* command is used to write out the name of the pdb file (*pdbname*) and information on the number of ligands.

print

The *print* statement can be used to write information which will appear ??????. The print statement is followed by a comma-separated list of objects which can of any type (integer,float,string,list or dictionary).