Olex2 Manual

Draft: June, 2010

> This document describes some of the commands that are available in Olex2. Many of these commands are also available directly from the Olex2 Graphical User Interface. Most items on the GUI have a small 'info' symbol next to them, where you can find out more about any of these items.

1. Introduction

There is no special console window in Olex2 - the commands described in this document can be typed where ever you are in Olex2 and the text you type (as well as the program response) will

appear in the bottom left hand corner of the main window. The text will then scroll up behind the displayed molecule. The number of lines of text that are visible can be set with the command **lines *n*.** You can also toggle between showing the **molecule only**, showing the **text only** and showing both at the same time (default) using **Ctr+T**. You can always examine the text output in your default text editor by typing **text**.

Many commands in Olex2 are modelled on the syntax that may be familiar from SHELX: four letter commands, where the letters often provide a hint about the function of the command. Many commands that are available in ShelXP, for example, can be used in Olex2. Also, all commands of the ShelXL and ShelXS syntax are interpreted by Olex2 and used to construct the internal Olex2 structure model. This model is then used directly to carry out a smtbx-refine refinement, whereas a shelx.ins file is generated on the fly if ShelXL/XH is chosen for the refinement.

All commands in Olex2 will **auto-complete** when pressing the TAB key. If the completion is not possible, because there is more than one command starting with the letters that have been typed, a list of these commands will be printed. It is good practice to use the auto-complete feature!

2. Understanding the Syntax

Selection: If one or more atoms are selected on the screen, then any command that acts on a selection will apply to the ***selected atoms only***. If there is no selection, it will apply to ***all*** atoms. Instead of making a selection on the screen, a list of atom names can also be supplied. *If a command has been successful, the selection will disappear. (Although there are a couple of exceptions to this rule)*

Mode: If Olex2 is in a **Mode**, the chosen action will be applied to all subsequently clicked atoms. The mouse pointer will change from the default arrow symbol to signify that Olex2 is in a mode. To get out of a mode, simply press the **Esc** key.

Syntax used in this document:

{a, b, c}: choice of a, b or c. For example: fix {occu, xyz, Uiso} [atoms] means 'fix occu [atoms]', 'fix xyz [atoms]', 'fix Uiso [atoms]'.

[val=2]: optional parameter. This parameter is not required for the command to work, and if it is not supplied, the default value will be used.

-k: This is an option switch.

i: Italic characters are used for variables.

[atoms] means an optional list of atoms. Any atoms that are selected will automatically be present in this list. If there are no selected atoms, **all** atoms will be in this list. Alternatively, the atom names of the atoms that should appear in this list can be typed by hand.

atoms means a compulsory list of atoms. Any atoms that are selected will automatically be present in this list. Alternatively, the atom names of the atoms that should appear in this list can be typed by hand.

Capital Letters are used for commands that will directly affect the structure model in the refinement. These commands will become part of the structure model and will appear in the ShelX input file. Please note that these commands can be typed either in upper or lower case.

Example Commands are represented in this format: **refine 4 20** and can be typed exactly as they are given. In this example, the structure will be refined with 4 refinement cycles and 20 electron density peaks will be returned from the electron density map integration.

3. Tables of Olex2 Commands

3. 1 Changing the Model View

matr	[1,2,3 or abc] or [abc a1b1c1] or [x11 x12 x13 y11 y12 y13 z11 z12 z13]	Orients the model along a (1 or 100), b (2 or 010), c (3 or 001) or any other crystallographic direction, like 123, which sets current normal along (1*a+2*b+3*c) vector. Two crystallographic directions (from and to) may be specified align current view normal along the (to-from) vector. Also a full Cartesian matrix can be specified. If the directions are signed or consist of multiple digits all components should be of the same length like in 120101 or -1+1+1 (same as -10101). If no arguments given, prints current Cartesian orientation matrix. **Examples**: • **matr 1** or **matr *a*** or **matr 100** - sets current normal along the crystallographic ***a*** direction • **matr 100 011** sets current normal along (011-100) direction (the normal direction changes if from and to are swapped)
rota	[axis angle] or [x y z angle increment]	Changes current view by rotating around given axis (x, y or z) when two arguments are provided and makes a continuous rotation around give axis when 5 arguments are provided. Note that X axis is aligned horizontally, Y - vertically and Z is out of the screen plane.

		Examples: • **rota x 90** rotates the structure 90 degrees around X axis • **rota 0 0 1 90 1** rotates model in the screen plane (around Z) 90 degrees with 1 degree increment.
direction		The command prints current normal in crystallographic coordinates and tries to match it to a crystallographic direction.
mpln	[atoms]] [**-n**] [**-r**]	Finds the best plane through the current selection or given atoms, or out of all visible atoms if none are given. • **-n** sets the view along the normal of the plane • -r: creates a regular plane

The model can be ***rotated*** using by moving the mouse pointer while holding the left mouse button down (also Shift+arrow keys); ***rotated around Z*** by pressing the **CTRL** key down while rotating; ***zoomed*** using the right mouse button (also Shift+Home/End). The default mouse behaviour can be overridden in some modes (look at mode split) also some objects, like cell basis or text boxes can override some mouse operations (like zooming on the cell basis) or extend it (moving the basis while holding Shift key down).

3. 2 Keyboard Shortcuts

CTRL+Q	**ShowQ**	Toggles between three states: • show electron density peaks • show electron density peaks with bonds • hides electron density peaks
CTRL+H	**ShowH**	Toggles between three states: • show hydrogen atoms • show hydrogens with internal h-bonds • hides hydrogen atoms
CTRL+T	**ShowStr**	Toggles between three states: • show structure only • show show structure and text • show text only
CTRL+I	**sel -i**	Inverts the current selection.

Ctrl+A	**sel -a**	Selects all atoms currently visible, however if labels are active (i.e. one or more label is selected) then this selects all labels.
Ctrl+U	**sel -u**	Deselects all of current selection.
Ctrl+G	**mode grow**	Enters mode grow. See also symmetry operations.
Ctrl+O	**reap**	Brings up the Open File dialogue.
F2	**swapbg**	Swaps the background between white and coloured.
F3	**labels**	Toggles labels on/off.
F4	**grad -i**	Toggles gradient background on/off.
F5		Go to the **work** menu.
F6		Go to the **view** menu.
F7		Go to the **tools** menu.
F8		Go to the **info** menu.
F11	**Fullscreen(true/false)**	Toggles full screen mode on/off.
Shift+F11	**HtmlPanelVisible**	Toggles html panel on/off.
Esc		Exits current mode (some modes, like mode *match*, can override this), clears current selection and text in the command line
Break		Interrupts the solution/refinement after the current cycle.

3. 3 Fixed/Refined Parameters

fix	{occu, xyz, Uiso} [atoms]	Fixes the specified refinement parameter, ie these parameters will not be refined in subsequent refinement cycles. • **occu**: will fix the occupancy • **xyz**: will fix the xyz coordinates • **Uiso**: will fix the whole ADP **Examples:** • **fix occu 0.5**: will set and fix the occupancy of the current selection to 0.5 • **fix xyz**: will fix the x, y and z co-ordinates of the currently selected atoms, ie not refine them.
free	{occu, xyz, Uiso} [atoms]	The opposite of fix - makes the specified parameters for the given atoms refineable. Feeing the occupancy is also available from the context menu.
mode	fixu	Fixes Uiso or ADP for subsequently clicked atoms.

mode	fixxyz	Fixes coordinates for subsequently clicked atoms.
mode	occu occupancy_to_set	Sets atoms occupancy to the povided value for subsequently clicked atoms.
labels -f show currently fixed atomic parameters, **labels -f -r** show labels for fixed atoms and also the number at which the occupancy of riding atoms is fixed		

3. 4 Atom Connectivity Table Manipulation

conn	***n*** [***r***] atoms	Sets the maximum number of bonds for the specified atoms to ***n*** and changes the default bond radius for the given atom type to ***r***. **Examples:** • **conn 5 $C** sets the maximum number of bonds all C atoms can have to 5, • **conn 1.3 $C** changes the bonding radius for C atoms to 1.3 (the floating point is used to distinguish between ***n*** and ***r*** in this case!), • **conn 5 1.3 $C** combines the two commands above
compaq	[**-a**] [**-c**] [**-q**]	Moves all atoms or fragments of the asymmetric unit as close to each other as possible. If no options are provided, all fragments are assembled around the largest one. • **-a:** assembles broken fragments • **-c:** similar to the default behaviour, but considers atom-to-atom distances and will move all atoms to the closest possible position to the largest fragment in the structure. • **-q:** moves the electron density peaks close to the atoms.
addbond	**A1 A2** *or* atoms	Adds a bond to the connectivity list for the specified atoms. This operation will also be successful if symmetry equivalent atoms are specified.
delbond	**A1 A2** *or* Selected bond(s)	Removes selected bonds from the connectivity list.
sort	[**m**] [**l**] [**p**] [**h**] atoms [**s**] [**h**] [**m**] moiety	The sorting of atoms in the atom list is very powerful, but also quite complex . • **-m**: atomic weight • **-l**: label, considering numbers

		• **-p**: part, 0 is first followed by all positive parts in ascending order and then negative ones • **-h**: to treat hydrogen atoms independent of the pivot atom. Sorting of moieties • **-s**: by size • **-h**: by heaviest atom • **-m**: by molecular weight **Usage:** • sort [+atom_sort_type] TBA sort [Atoms] [moiety [+moiety_sort_type] [moiety_atoms]] If just 'moiety' is provided - the atoms will be split into the moieties without sorting. **Examples:** • **sort +m1 F2 F1 moiety +s** will sort atoms by atomic **m**ass and **l**abel, put F1 after F2 and form moieties sorted by **s**ize. Note that when sorting atoms, any subsequent sort type operates inside the groups created by the preceeding sort types.
Olex2 will display the altered connectivity table in the case if structure is grown or packed		

3. 5 Symmetry Operations

lstsymm		Prints symmetry operations and their codes for current structure.
envi	*[**r**=2.7 Å]* **A1** *or* one selected atom [**-h**] [**-q**] **Note:** if more than one atom is selected the first one is used	Prints a list of those atoms within a sphere of radius ***r*** around the specified atom. • **-h**: adds hydrogen atoms to the list • **-q**: option adds Q-peaks to the list
mode	**grow** [**-s**] [**-v**] [**-b**]	Displays the directions in which the molecule can be grown • **-s:** also shows the short interaction directions

		• ***-v:*** [2.0 Å] shows directions to the molecules within ***v*** value of the Van der Waals radii of the selected atoms which can be generated by clicking on the direction representations, only unique symmetry operations (producing shortest contacts are displayed) • ***-r:*** shows directions to all symmetry equivalent atoms atoms of the selected one(s) within 15 Å • shortcut **CTRL+G** is used to enter the '**mode grow**'
mode	**pack**	Displays the position of symmetry equivalent asymmetric units as tetrahedra. These asymmetric units can be generated by clicking on the corresponding tetrahedron.
sgen	atoms The Symmetry operation is represented as 1_555, 1555 or -1+X,Y,Z and atoms as a selection or a names list	Generates symmetry equivalents of the provided (or all atoms, if there is no selection) using the provided symmetry operation. **Note:** For symmetry operations starting with '-' and letter, a leading zero must be added, for example, **0-x,-y,-z**, otherwise Olex2 confuses this with an option.
pack	**a_from a_to b_from b_to c_from c_to** [atoms]	Packs all or specified atoms within given dimensions • **-c:** prevents clearing existing model **Example:** **pack $O** will pack all O atoms with the default of -1.5 to 1.5 cells range.
pack	**from to**	Equivalent to '**pack from to from to from to**', like '**pack 0 1**' is expanded to '**pack 0 1 0 1 0 1**'
pack	**cell**	Shows content of the unit cell. In conjunction with '**grow -w**' allows the creation of views where all asymmetric units contributing to the unit cell are shown.
pack	***r***	Packs fragments within radius ***r*** of the selected atom(s) or the centre of gravity of the asymmetric unit.
grow	[atoms] [**-w**]	Grows all possible/given atoms; for polymeric structures or structures that require to be grown several times Olex2 will continue grow until the operation results in a symmetry element that has been used previously. • **-w:** permits the application of symmetry previously used operations to other fragments of the asymmetric unit

		Example: If the main molecule is grown, but only one solvent molecule is shown, using '**grow -w**' will produce other solvent molecules using symmetry operators used to grow the main molecule
If some atoms are deleted after growing operations, Olex2 will use existing unique atoms as the asymmetric unit atoms; this can be helpful to avoid a sequence of sgen/kill commands. **labels -l -i**: Adds labels only to the 'original' - i.e. not created by symmetry - molecule. In a packed structure: **Right-click on a bond > Graphics > Select the Groups(s)**: Will select all bonds (or atoms) of that type in the grown structure.		

3. 6 Disorder Modelling: Constraints and Restraints

EXYZ	atom types (to add for the selected atom) [-**EADP**] [-**lo**]	Makes the selected site shared by atoms of several atom types. • **-EADP:** adds the equivalent ADPs command for all atoms sharing one site. • **-lo:** links the occupancy of the atoms sharing the site through a free variable.
EADP	atoms	Makes the ADP of the specified atoms equivalent.
SADI	atoms *or* bonds [**esd**=0.02]	For selected bonds or atom pairs **SADI** makes the distances specified by selecting bonds or atom pairs similar within the esd. If only **one** atom is selected it is considered to belong to a regular molecule (like PF_6) and adds similarity restraints for P-F and F-F distances. For **three** selected atoms (**A1**,**A2**,**A3**) it creates similarity restraint for **A1-A2** and **A2-A3** distances.
DFIX	***d*** atom pairs or pairwise selection in order [**esd**=0.02]	For selected bonds or atom pairs **DFIX** will generate length fixing restraint with the given esd. If only **one** atom is selected, all outgoing bonds of that atom will be fixed to the given length with provided esd. For **three** selected atoms (**A1**,**A2**,**A3**) the A1-A2 and A2-A3 restraints will be generated.

DANG	***d*** atom pairs *or* pairwise selection in order [**esd**=0.04]	For selected bonds or atom pairs, distance restraints similar to dfix will be generated.
tria	***d1 d2 angle*** [**esd**=0.02]	For given set of bond pairs sharing an atom or atom triplets generates two dfix commands and one dang command. **Example: tria 1 1 180 C1 C2 C3** will generate 'DFIX 1 0.02 C1 C2 C2 C3' and 'DANG 2 0.04 C1 C3' it will calculate the distance for dang from d1 d2 and the angle.
FLAT	[atoms][**esd**=0.1]	Restrains given fragment to be flat (can be used on the grown structure) within given esd.
CHIV	[atoms][***val***=0] [**esd**=0.1]	Restrains the chiral volume of the provided group to be ***val*** within given esd
SIMU	[***d***=1.7] [**esd12**=0.04] [**esd13**=0.08]	Restrains the ADPs of all 1,2 and 1,3 pairs within the given atoms to be similar with the given esd.
DELU	[**esd12**=0.01] [**esd13**=0.01]	'rigid bond' restraint
ISOR	[**esd**=0.1] [**esd_terminal**=0.2]	Restrains the ADP of the given atom(s) to be approximately isotropic
SAME	N	Splits the selected atoms into the N groups and applies the SAME restraint to them. Olex2 will manage the order of atoms within the in file, however mixing rigid group constraints and the 'same' instructions might lead to an erroneous instruction file.
showp	[any]; space separated part number(s)	Shows only the parts requested: **showp 0 1** will show parts 0 and 1, **showp 0** just part 0. **showp** by itself will display all parts.
split	[-**r**={eadp, isor, simu}]	Splits selected atom(s) along the longest ADP axis into two groups and links their occupancy through a free variable. • **-r:** adds specific restraints/constraints (EADP, ISOR or SIMU) for the generated atoms
AFIX	shelx afix number{mn} [-**n**]	If no are atoms provided and afix corresponds to a fitted group where n is 6 or 9 (such as 106 or 79), all the rings which satisfy the given afix will be automatically made rigid

		(this is useful in the case of many PPh3 fragments); alternatively a single ring atom can be selected to make that ring rigid. In other cases, depending on afix either 5,6 or 10 atoms will be expected. Special cases of afix 0, 1 and 2 can be used to remove afix, fix all parameters or leave just the coordinates refinable, all other afix instructions will consider the first atom as a pivot atom and the rest - dependent atom. • **-n:** consider N-atoms as parts of rings
part	[**part**=new_part] [atoms] [-**p**=1]	Changes part number for given/selected atom; • **-lo**: links occupancies of the atoms through a +/-variable or linear equation (SUMP) depending on the -**p**[=1] • **-p**: specifies how many parts to create. If -**p**=1, -**lo** is ignored and the given or new part is assigned to the provided atoms.
fvar	[***value***] [atoms]	This command links two or more atoms through a free variable. • If **no *atoms*** are given the current free variables are printed. • If **no *value*** is given but two atom names are give, the occupancies of those atoms are linked through a new free variable. • If a **value of 0** is given, the occupancy of the specified atoms will be refined freely • if the **value is not 0**, the occupancy value of the specified atoms is set to the given value.
sump	[***val***=1] [**esd**=0.01]	Creates a new linear equation. If any of the selected atoms has refinable or fixed occupancy, a new variable is added with value 1/(number of given atoms), otherwise already used variable is used with weight of 1.0. **Example:** If 3 atoms (A1, A2, A3) are selected this command will generate three free variables and insert the **r2 1.0 var 3** instruction (equivalent to 1.0 = 1.0*occu(A1) + 1.0*occu(A2) + 1.0*occu(A3).
mode	split [-**r**={eadp, isor, simu}]	Splits subsequently clicked atoms into parts, or in combination with the Shift key can be used to drag an atom to change its position. While in the mode the newly generated atoms can be selected and moved as a group with Shift down or rotated when dragging the selection. The

		original and generated atoms will be placed into different parts. • **-r**: can be used to generate extra restraints or constraints for original and generated atoms (see also the '***split***' command); values EADP, ISOR or SIMU are allowed

3. 7 Selection Syntax

sel	sel atoms where xatom.bai.mw > 20	Will select all atoms where the atomic mass is larger than 20
sel	Symmetry operation (represented by 1_555 or 1555)	Will select all currently shown symmetry generated atoms which were generated by the symmetry operation given.
sel	sel rings NC5	Will select all NC5 rings in the structure
sel	sel part 1	Will select part 1 of the structure

3. 8 HKL file Operations

hklstat		Prints detailed information about reflections used in the refinement.
omit	h k l	Inserts 'OMIT h k l' instruction in the ins file
omit	val	Inserts 'OMIT h k l' for all reflections with $\|Fo^2 - Fc^2\| > val$.
omit	s 2theta	Inserts 'OMIT s 2theta' instruction in the ins file
edithkl	[h k l]	Brings up a dialogue, where 'bad' reflections from the Shelx lst file and all its constituent symmetry equivalents can be inspected and flagged to be excluded from the refinement. In constrast to the OMIT h k l instruction, which excludes the reflection and *all it equivalents*, this dialogue allows to exclude those equivalents that are actually outliers. If a particular reflection is specified, this particular reflection and all its constituent equivalents can be viewed.

excludehkl	-h=h1;h2;.. -k=k1;k2.. -l=l1;l2.. [-c]	This function provides a mechanism to reversibly exclude some reflections from refinement (these reflections will be moved to the end of the hkl file so they appear after the 0 0 0 reflection). • **-c:** option controls how the given indices are treated, if not -c option is provided, then any reflection having any of the given h, k or l indices will be excluded, otherwise only reflections with indices within provided h, k and l will be excluded.
appendhkl	-h=h1;h2;.. -k=k1;k2.. -l=l1;l2..	Acts in the opposite way to excludehkl
For more advanced HKL processing, a Python script may be used. A sample hklf5.py script is provided in {Olex2 folder}/etc/scripts. The script can be copied and modified to accommodate any particular twinning law and run inside Olex2. The script allows creating an HKLF 5 file where reflections which belong to different twin components are assigned different batch numbers. To run a python script in Olex2 use the following command to load the script: >>@py -l This command shows a 'File Open' dialog, a python script can be selected. After loading the script can be modified and executed by pressing OK.		

3. 9 Customising the Olex2 GUI

setfont	{Console, Picture_labels}	Brings up the dialog to choose font for the Console or Labels which end up on the picture. Built in function choosefont([olex2]) to choose system or specially prepared/portable font can be used to specify the font.
grad	[C1 C2 C3 C4] [-**p**]	Choose the colour of the four corners of the graduated background. • **-p**: a picture file name to be placed at the background
brad	***r*** [hbonds] operates on all or selected bonds	Adjust the bond radii in the display. If the 'hbonds' is provided the second argument, the given radius is applied to all hydrogen bonds.
ads	{elp, sph, ort, std}	A function for drawing styles development. Changes atom draw style for all/selected atoms. • elp - represents atoms as ellipsoids (if ADP is available) • sph - represents atoms as spheres • ort - same as elp, but atoms have one of the quadrants cut out

		• std - a standalone atom (i.e. grown as a cross in wire-frame mode)
arad	{sfil, pers, isot, isoth, bond, vdw}	A function for drawing styles development; applies different radii to all/selected atoms. • sfil - sphere packing radii (as in ShelXTL XP) • pers - a fixed radii for model viewing • isot - each atom has it's own radius depending on the value of the Uiso or ADP • isoth - same as isot, but the H atoms are also displayed with their real Uiso's • bond - all atoms get same radii as default bond radius • vdw - the default/loaded Van der Waals radii used in most of the calculations
azoom	% [atoms]	Changes the radii of all/given atoms, the change is given in percents.

3. 10 Output: Tables, Reports and Images

pictPS	filename.ps	Generates a post-script file of what is visible in the molecule display. • **-atom_outline_color** - the colour of the atom outline, used for extra 3D effect for the intersecting objects [0xFFFFFF] • **-atom_outline_oversize** - the size of the outline [5]% • **-bond_outline_color** - same as for the atom, can be changed to black to highlight bond boundaries • **-bond_outline_oversize** - the size of the outline [10]% • **-color_fill:** Fills the ellipses with colour. • **-color_bond:** Bonds will be in colour. • **-color_line:** Lines representing the ellipses will be in colour. • **-div_pie:** number [4] of stripes in the octant • **-lw_ellipse:** line width [0.5] of the ellipse • **-lw_font:** line width [1] for the vector font • **-lw_octant:** line width [0.5] of the octant arcs • **-lw_pie:** line width [0.5] of the octant stripes • **-p:** perspective • **-scale_hb:** scale for H-bonds [0.5] The bond width is taken from the display. This can be changed with **brad**
pict	filename.*ext [**n**=2]* [-**pq**]	Generates a bitmap image of what is visible on the molecule display. ***n*** Refers to the size of the output image. If ***n*** is smaller than 10, it refers to a multiple of the current display

		size, if it is larger than 100, it refers to the width of the image in pixels. • ext {png, jpg, bmp}. png is best. • **-pq:** print quality
picta	filename.*ext* [***n***=1] [**-pq**]	A portable version of **pict** with limited resolution (see explanation for ***n*** above), which is OS and graphics card dependent. This may not be stable on some graphics cards • **-pq:** print quality • **-n**: as for 'pict'
picts	filename.ext [***n***=1] [**-a**=6] [**-s**=10] [**-h**=n*(screen height)]	Creates a 'stereo' picture with two views taken with the +/- **a** option rotation around y axis and placed onto one picture separated by s pixels. • **-a**: half of the view angle • **-s**: separator width in % • **-h**: the height of the output, by default equals to current screen height multiplied by the given resolution
label	**label** [atoms]	Adds labels to the selected atoms. These labels can be moved by pressing the SHIFT key while holding down the left mouse button • **-type:** {subscript, brackets, default}, the type only affects the PostScript labels and not applicable to the raster pictures

3. 11 Structure Analysis

There are various tools available for the analysis of structures.

htab	[minimal angle=150°] [maximum bond length 2.9 Å] [**-t**] [**-g**]	Searches and adds found hydrogen bonds (like HTAB and RTAB in Shelx) into the list for the refinement program to add to the CIF. Equivalent symmetry positions are automatically inserted and merged with the existing ones. The command can be executed several times with different parameter values, only one unique instructions will be added. • **-t:** adds extra elements (comma separated like in -t=Br,I) to the donor list. Defaults are [N,O,F,Cl,S] • **-g**: if any of the found bonds are generated by symmetry transformations, the structure is grown using those symmetry transformation
pipi	[centroid-to-centroid distance 4 Å] [centroid-to-centroid shift 3 Å] [**-g**]	The command analyses the p-p interactions (only stacking interactions for now) for flat regular C6 or NC5 rings and prints information for the ones where the intercentroid

		distance is smaller than [4] Å and the intercentroid shift is smaller than [3] Å. • **-g:** if any of the rings is fully or partially constructued of symmetry generated atoms it grow the structure using those symmetry operators
calcvoid	[radii file name] [all atoms/selected atoms] [-**d**=0] [-**p**] [-**r**=0.2Å]	Calculates and displays the structure map. Also calculates the largest channels along crystallographic directions and the packing index. • -**d**: extra distance from the surface • -**p**: precise calculation, each map voxel is tested, the default quick algorithm, uses the atom masks to find volume occupied by the molecule. The precise calculation is vectorised • **-r**: resolution, a resolution of at least 0.1Å and **-p** options is required to get values for publishing **Note:** The radii used in the calculation are currently coming from the CSD website: http://www.ccdc.cam.ac.uk/products/csd/radii However there are several ways how the radii can be changed, one of the ways is to provide a file name with radii ([element radius] a line format), the other one is to load the radii from the same kind of the file using 'load radii vdw' command.
molinfo	[radii file name] [atoms] [-**g**=5] [-**s**=0]	Calculates molecular volume and the surface area for all/ selected atoms. • -g: generation of the triangulation process • -s: source of the triangles for the sphere triangulation, [o]ctahedron or [t]etrahedron are available Generation 5 for octahedron approximate sphere by 8192 triangles, for tetrahedron by 4096 triangles, each generation up increases the number of triangles by factor of 4, generation down - decreases it by the same factor.
calcfourier	{-**calc**,- **diff**, -**obs**, -**tomc**} [-**r**=0.25Å] [-**i**] [-**scale**=simple] [-**fcf**]	Calculates Fourier for current model • -**r**: the resulting map resolution in angstrems • -**i**: integrate the calculated map • -**scale**: when Olex2 calculates structure factors, it uses the linear scale as a sum(Fo^2)/sum(Fc^2) by default, however a linear regression scale can be also used (use -**scale**=regression) • -**fcf**: Olex2 will use an FCF with LIST 3 structure factors as a source of the structure factors. If this option is not specified, Olex2 will calculate the structure factors using the the reflection used in the refinement (use the 'hklstat' command to see more information on reflections).
calcpatt		Calculates and displays Patterson map
Notes etc about Structure Analysis		

4. Appendix

4. 1 About Versions and Tags

The Olex2 distribution system has undergone many changes since the project was started in 2004. We have always aimed at providing program updates as soon as possible to the Olex2 user community. We think that one of the best ways to encourage bug reports and suggestions is to translate this user feedback as soon as possible into real improvements in the software.

For a while - up to about December 2009 - we have made updates available on a very frequent basis. This has met with a warm welcome from many of our users, but has also caused some problems: Not all updates did *only* do what they were supposed to do! At that point, we have decided to change the policy somewhat, and have come up with the following system for the distribution of Olex2.

There are now distinct versions of Olex2. Before Version 1.0, everything consisted of continuously updated files. At some point, this became no longer supportable, and we decided to introduce proper versions into the Olex2 distribution system. Any new version requires a complete re-install. However different versions of Olex2 can exist next to each other without causing any interference. For each version of Olex2, there are three 'tags', referring to different source repositories. For example, for Version 1.1 there are the following tags:

- 1.1-alpha
- 1.1-beta
- 1.1

Alpha: Whenever we made some changes, we 'make' an *alpha* distribution of Olex2. We use this version for in-house testing (although you are very welcome to use this version too, as long as you are aware of the fact that this version is typically very experimental and will very likely cause some problems. However, if you have suggested a new feature, or reported a bug fix, you may well find that we have implemented your suggestions already!

Beta: Once we've done some testing of this alpha version, we 'promote' it to the *beta* distribution. This version is tested by a wider group of testers - these tend to be those users with whom we have a lot of contact.

Release: Once a distribution has been tested in the *beta* stage, a proper release is made. This can be expected to be stable and if you encounter any problems with release version, please tell us about this! It doesn't matter how small the problem is, we'd like to know.

4. 2 Installing Olex2

4. 2. 1 Windows:

- Please download the Olex2 installer from the Windows tab and run it. Select the destination folder to which to install Olex2 (typically **C:\Program Files\Olex2**). If you do not have administrator privileges, please select a folder where you have full access rights.
- Make sure you select the latest version of Olex2—Version 1.1—from the download repository.
- Click on **Install.** This will install Olex2 on your computer. When it is done, there will be a 'Run' button on the installer form. Click this to run Olex2. The first time Olex2 runs on your computer, it will take some time to start up (up to one minute!).

Olex2 should now be opened, there should be no red (error) lines in the main window and there should be a molecule of sucrose displayed on the screen. Olex2 does not require any third party programs to perform structure analyses—Structure Solution as well as Structure Refinement—but, if you have a ShelX licence, you may want to make sure that Olex2 can interact with the ShelXS, ShelXL and ShelXM. If you do not have a ShelX licence, and would like to obtain one, please go to the ShelX Pages for more information.

Please note that the ShelX executables that are shipped with WinGX do not work with Olex2.

You can either copy your ShelX executables into the Olex2 installation folder, or—better—you can copy your executables into a folder which you then add to the PATH variable of Windows. For example, create a folder **C:\Program Filex\Shelx**, then Right-Click on 'My Computer' (XP) or 'Computer' (Vista and 7) and select *Properties*. Then select *Advanced*. There you can add the location of your ShelX executables to the PATH variable.

4. 3 External Programs

4. 3. 2 SHELX

All programs of the SHELX family can interact seamlessly with Olex2. There is no need for registering any of these programs with Olex2, it is enough if the folder containing the ShelX programs are on the system PATH. This is normally the case if ShelXTL has been installed on a system. Otherwise, you will need to set you system PATH variable to include the folder where you keep your SHELX executables.

Please note that the SHELX executables that are shipped with WinGX do not work with Olex2. These executables have been modified in such a way that they will only work properly with WinGX. Since WinGX puts the folder that contains these executables on the system PATH, you might find that SHELX appears in Olex2 - and then doesn't work. In this case, you will need to get new SHELX executables and put them in the same folder where Olex2 is installed - executables found there will be used by Olex2 preferentially.

4. 3. 3 Platon

John Warren has provided an interface to PLATON.

4. 3. 4 SuperFlip

Arie van der Lee has provided an interface to SuperFlip.

4. 4 About Macros and Scripting in Olex2

Olex2 supports two different tpes of external scripting: Macros and Python scripts.