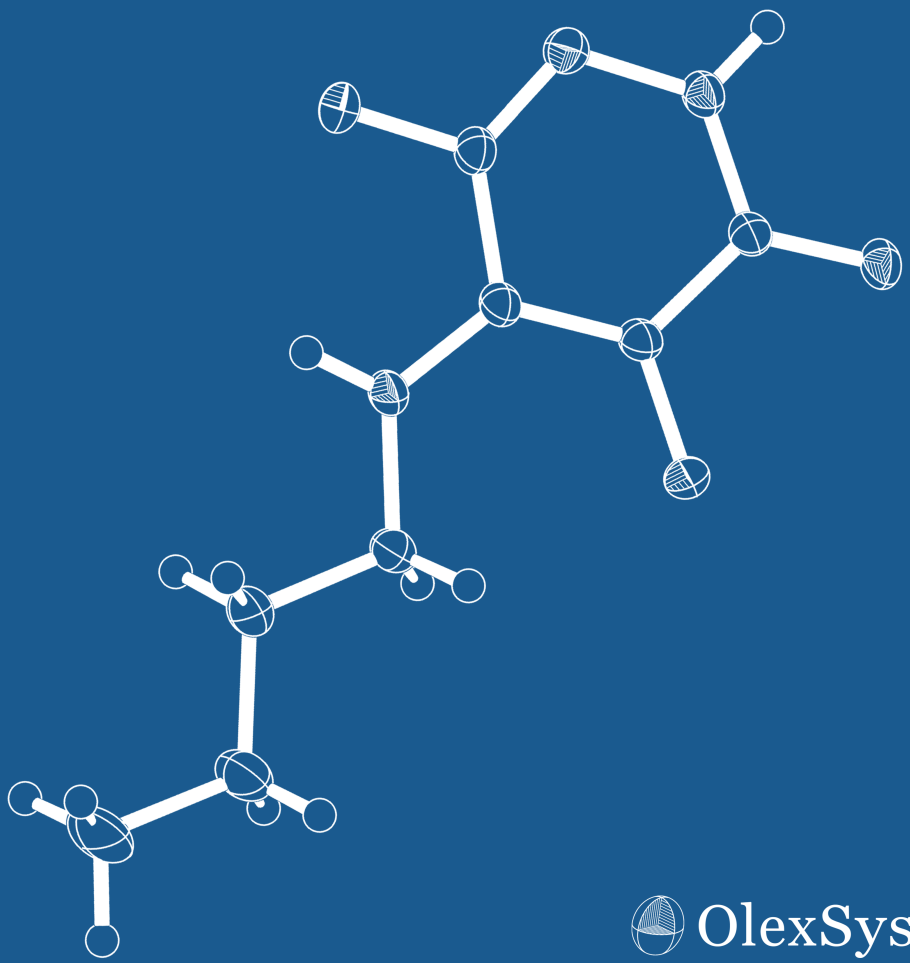
2014
international year of crystallography

OLEX2

Getting Started



OlexSys

Olex2

Getting Started

OlexSys

This booklet has been produced for the 23rd IUCr Congress and General Assembly in Montreal 2014. It is an abriged version of the full official Olex2 Manual, which is available from OlexSys Ltd on request.

The manual was written by Hazel A. Sparkes, Oleg Dolomanov and Horst Puschmann. We thank Natasha Chetina for editing this text.

OlexSys Ltd are:

Dr. Horst Puschmann (*Chief Scientific Officer*): horst@olexsys.org
Dr. Oleg Dolomanov (*Chief Technical Officer*): oleg@olexsys.org
Beatrice Leigh (*Chief Executive Officer*): bea@olexsys.org
Prof. Judith A.K Howard (*Chairperson*): judith@olexsys.org

Department of Chemistry
Durham University
Durham
DH1 3LE
U.K.

Printed and bound by **printed.com**, UK

Getting Started with Olex2

This abridged version of the full official Olex2 manual has been produced especially for the IUCr Meeting in Montreal 2014. This is the International Year of Crystallography, and we are not only celebrating 100 years of crystallography, but also 10 years of Olex2 - the standard for crystallographic software. The full official Olex2 manual, is available on request.

Olex2 Crystallographic Software

Olex2 is an immensely powerful program that handles very complex crystallographic tasks through an intuitive GUI. The graphical user interface is the result of 10 years of striving to provide the best possible experience to modelling even the most challenging structures with ease. The software originated at Durham University and has been in active development since 2004. Currently in development are extensions to improve productivity, some of which are already available to try out.

OlexSys Ltd

Founded in 2010 and still based at Durham University, OlexSys Ltd has one aim: To develop and improve Olex2 far into the future, whilst ensuring the core software remains open source and free. We understand what is involved in determining good crystal structures and we benefit enormously from being firmly embedded in the academic environment; understanding all steps of the crystallography process, from crystallisation, crystal selection, data collection and reduction through to structure solution, refinement and presentation.
OlexSys Ltd offers extensive crystallographic services:

- Crystal growth and data collection
- Preparation of crystallographic texts
- Generation of publication-quality crystallographic reports
- Pre-publication structure and manuscript checking
- Replies to Referee's comments
- Database curation of crystallographic data
- Crystallographic cloud services
- Olex2 Workshops

Visit us at www.olex2.org for downloads, information, requests, feedback and suggestions. We are always happy to help and your feedback will help us to improve Olex2!

When using Olex2 please always cite the following reference:

O. V. Dolomanov, L. J. Bourhis, R. J. Gildea, J. A. K. Howard and H. Puschmann. Olex2: A complete structure solution, refinement and analysis program. (2009) *J. Appl. Cryst.*, **42**, 339-341.

Olex2

Installation of Olex2

Olex2 is available free of charge for Windows, Mac OS X and Linux and the installers and zip archives can be obtained from *www.olexsys.org*. The Windows installer is a small application which downloads the appropriate Olex2 installation files for your system. In most cases, installation will be very straightforward, but various manual options available – including compiling your own version from source code repositories. For Mac OS X there is a DMG image for the release version and zip files for alpha and beta releases. Currently we only provide 32-bit builds for Mac. For Linux there are zip files for 32 and 64 bit versions of the operating system.
For a standard installation, Olex2 must be installed and launched at least once in an administrator mode in order for all of the features to function. Alternatively, the software can also be installed in a custom location, for example your home directory.

Variants of Olex2

The current release version of Olex2 is version 1.2.5. Of this version, we distribute three variants: the alpha, beta and release version. The alpha version will be updated rather more often than the others and you can expect to see more bugs. The beta version will be updated once we are close to making an official release and the release version will only be updated after we have had feedback from our beta testers. You can install and run all three versions independently on one computer. If you want to help us by testing the alpha and beta versions, please give them a go and feedback any problems to us. If something goes wrong, you can always fall back on the release version to complete your work.

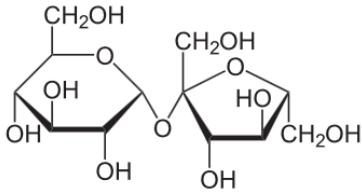
Updating Olex2

Unless you change the settings in the menu bar (Help > Update Options), Olex2 will automatically check for updates on start-up if you are connected to the internet. If updates are found, these will be downloaded in the background and applied on the next start-up of the software. Updates only happen within the same series and variant (see above) – eventually the current 1.2.5 version of Olex2 will update to the 1.2.6 version, but it will not automatically update to Olex2 2.0 – instead a notice regarding the new release will appear.

Chapter 1

Structure Determination of Sucrose

This section provides a guide through the basic structure solution and refinement of sucrose. With these step-by-step instructions you will be able to repeat this structure solution and refinement process for yourself and this will help you become familiar with the way in which Olex2 works.



1.1 Selecting Data

1. Click on the HOME tab in the GUI panel in Olex2. Under the START header one of the sample structures is **Sucrose**. Click on this and the model/instructions *.ins* file will be loaded.

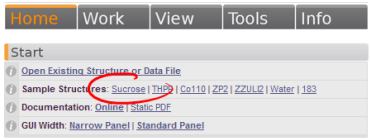


Figure 1.1: Home Tab in GUI Panel

Since this sample structure has already been solved the sucrose molecule appears on your screen. Normally, if you are loading an unsolved structure the screen would be blank as none of the atom positions have been determined.

1.2 Solving the Structure

1. Click on the WORK tab in the GUI panel. Like all active GUI elements, it will change colour and become orange.
2. Click on the arrow next to the SOLVE button so that it is turns orange and points upwards. The *structure solution* options are displayed.
3. Select a **solution program** from the drop-down menu. Depending on your computer set-up, you may only see the built-in olex2.solve program, but you may also see other solution programs in the list, if they are installed.
4. Click SOLVE (or type `solve`) and Olex2 will attempt to solve the structure. Once the structure is solved, *electron density peaks* (*Q-Peaks*) are shown as brown spheres. Hover over these to see their peak height, or look at INFO | ELECTRON DENSITY PEAKS for a click-able list of the top 10 peaks. The colour intensity of these Q-peaks represents the relative intensity. The molecule of sucrose is clearly visible, along with some much weaker, artificial peaks.

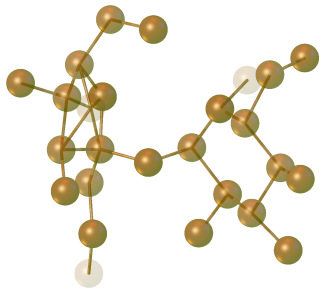


Figure 1.2: Sucrose Q-Peaks. The structure can easily be guessed from the position of the electron density peaks.

> Q-peaks represent maxima in the *electron density map i.e.* places where the atoms are likely to be located. If the structure solution has been successful, you will probably see the whole molecule, but this is not always the case: sometimes you will only see parts of the molecule and these might be fragmented. The intensity of the spheres is scaled to the largest peak present. So the weakest peaks appear faded out. The scaling used here can be adjusted in INFO | ELECTRON DENSITY PEAKS.

5. In the WORK section there is another header tab TOOLBOX WORK. Select *Q-Peak Intensities* from the LABELS menu. The relative intensities of the Q-peaks will now be displayed as labels with each atom.

> The Q-peak names also indicate the relative intensity order *i.e.* the most intense peak is Q1 with Q2, Q3 ... etc. having less relative intensity. Select LABELS ON/OFF from the drop-down menu in WORK | TOOLBOX WORK | LABELS (or press F3). At this stage it is normally more useful to see their relative intensities.

6. Still in the TOOLBOX WORK click on the Q icon to toggle between *Q-peaks*, *Q-peaks with bonds* and *No Q-peaks* (Or press CTRL+Q repeatedly). Some of the Q-peaks may not represent real atoms at this stage so do not worry if there are unexpected bonds.

Since X-rays are diffracted by the electrons surrounding atomic nuclei, the larger Q-peaks will relate to heavier atom types.

7. The 'structure' can be rotated by holding down the LEFT MOUSE button in the background of the main screen and moving the mouse.

8. The next stage is to assign atom types to the Q-peaks. Use the LEFT MOUSE button to click on Q-peaks that you believe correspond to a particular atom type (e.g. oxygen). Selected atoms turn green.

   If the Q-peaks don't appear to form a sensible structure, try typing `compaq -a`, which will pull Q-peaks together. Clicking on the centre button  (top right of GUI) has the same effect.

9. In TOOLBOX WORK click on the relevant atom type e.g. O. These Q-peaks will change colour depending on the atom type. Carry on assigning as many Q-peaks as possible to the correct atom type; it should be possible to find all C and O atoms at this stage (although this is not always the case for every crystal structure). With atoms selected, you can also type `name C` to assign all selected peaks as carbon atoms. By using the UP key, it is easy to repeat this command.

   Hovering over the atom with the mouse pointer shows the atom label e.g. O1, which takes the format of an atom type followed by a number. There are often some artificial peaks at this stage, so identify recognisable fragments rather than trying to assign an atom type to every Q-peak. Hydrogen atoms are not usually identified at this stage, since they have a very low scattering power.

1.3 Refinement

Now that initial atom assignments have been made, the next stage is to refine the structure to see how well your assignments fit the data and improve the model.

1.3 Initial Atom Type Assignement

1. Under the WORK tab click on the arrow next to the REFINE button so that it is pointing upwards and highlighted orange. The refinement options are now displayed.

2. Click on the REFINE button and Olex2 will refine your structure. Alternatively, you can type `refine` or press CTRL+R.

   During a *refinement* Olex2 is calculating *structure factors* based on your current structural model and comparing them to the experimentally measured values. Adjustments are then made to the atomic parameters to try and improve the fit.

3. If one of the atoms you assigned is now represented as a comparatively huge sphere (not just slightly larger than the others) then it is not real. Select it and press the DELETE key on your keyboard (or right-click and select *Delete* from the context menu). If you delete an atom by mistake, use CTRL+ *Z* to undo.

4. If there are peaks larger than ~3 units, then there are missing C or O atoms. Select these peaks and assign them to either C or O and refine again. Repeat this step until there are no more large Q-peaks left.

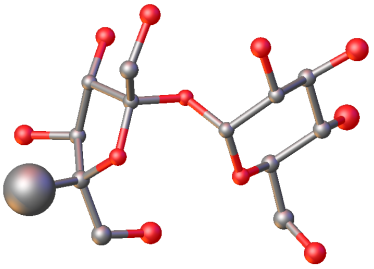


Figure 1.3: Large incorrect atom. The real atom – Hydrogen – is much lighter than what has been assigned: Carbon

5. At this point, you can hide all remaining Q-peaks: in TOOLBOX WORK click on the hide Q icon (or use CTRL+Q) to hide them.

6. Rotate your structure and examine the size of the atom spheres, they should all be approximately the same size. If this is not the case some of your atoms have been incorrectly assigned. In this image, atoms 3 are correctly assigned atom types with similar sized spheres. Atom 1 is too light an atom type (should be heavier) and atom 2 is too heavy an atom type (should be lighter).

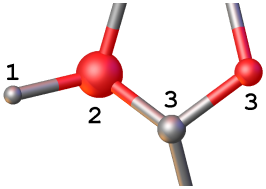


Figure 1.4: Atom spheres of different size: this indicates 1 must be heavier than the currently assigned atom type and 2 must be lighter.

> The atoms are displayed as spheres when they are modelled *isotropically* (*i.e.* one parameter to define the atomic displacement parameter). The size of all atom spheres is approximately the same but there are exceptions to the rule, for example if there are long floppy chains (e.g. propyl or butyl) in your structure, the size of the spheres generally increases along the chain to the distal atom due to the increased libration. Atomic disorder can also cause changes in the size of the spheres.

7. If a sphere is significantly *smaller* than the others, the atom type is too *light* e.g. assigned as a carbon atom but it should be an oxygen atom. If it is distinctively *larger*, it has been assigned as too *heavy* an atom. Select the offending spheres and change their atom types. If the sphere representing the atom has become too small to see, draw a rectangle (SHIFT+ LEFT MOUSE) around the atom to select it.

> If an assigned atom type is too light, there is not enough electron density available to fit the experimental data so the refinement pulls the available electron density into a smaller volume to try and improve the fit to the experimental data. If the assigned atom type is too heavy it has more electron density than expected and the refinement increases the volume over which the electron density is spread to improve the fit to the experimental data.

8. Carry out further refinement cycles (and adjust atom types) until all atoms are approximately represented by spheres of the same size. If the structure had become unrecognisable, you will need to click SOLVE to return to the original structure and start the solving and refining processes again.

At this stage you should have $R1$ ~ 8–9%. The biggest Q peak should be ~ 1.2.

1.3 Anisotropic Refinement

1. So far, we have limited the shape in which we can place electron density to a sphere. The next stage is to model the atoms as ellipsoids: using *anisotropic displacement parameters* , **ADP**, requires 6 parameters to define the volume occupied by the electrons during this *anisotropic refinement*, rather than the single one that was used to define the sphere during *isotropic* .

2. Click on the rugby-ball shaped blue icon  on TOOLBOX WORK at the top right. Olex2 will then automatically carry out an *anisotropic refinement*.

> Atoms will now be displayed more like rugby balls. If they appear like pancakes, tetrahedra or are comparatively much larger than other atoms in the structure, this must always be investigated. Common reasons for odd shaped ellipsoids include incorrect atom assignments, atomic disorder, incorrect space group assignment or poor data quality.

3. Again, the atoms should be of roughly similar size and shape in similar environments. If you have incorrect atom assignments at this stage, Olex2 might display tetrahedra instead of the expected ellisoids. Such atoms have become *non-positive definite* , *N.P.D*, and are in reality heavier (*i.e.* O not C). Very large ellipsoids indicate the atom type should be lighter (*i.e.* C not O). In the illustration: atoms 3 are correctly assigned atom types with similar sized spheres. Atom 1 is non-positive definite (atom type is too light) and 2 is a large ellipsoid (atom type is too heavy).

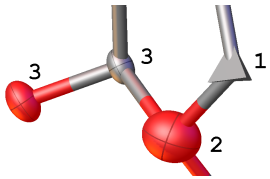


Figure 1.5: Incorrect ellipsoids. Atom 2 is too large (and should be lighter) and atom 1 has become non-positive definite, which means that the real atom is heavier than the assigned type.

1.3 Add Hydrogen Atoms

1. The next stage is to add hydrogen atoms to your structure. In TOOLBOX WORK click on ADD H in the top right hand corner (or type `hadd`). Olex2 automatically adds hydrogen atoms and includes them in the subsequent refinement.

> The hydrogen atoms are automatically added using a riding model with appropriate AFIX instructions.

2. In TOOLBOX WORK, clicking on the H square cycles through *Show H* / *Show H with H-bonds* / *Hide H* (alternatively use CTRL+H). Stop at *Show H with hydrogen bonds*.

   Hydrogen bonding interactions are shown as red dashed lines between a hydrogen atom and the acceptor atom.

3. Check that there have been no extra hydrogen atoms generated. If there are any, delete them. Select atoms by clicking on them with the LEFT MOUSE button and then press the DELETE key. REFINE the structure.

   If you select some hydrogen atoms and use the Delete All H button or type `kill $H` in the command-line, only the selected atoms will be deleted.

4. Repeat hydrogen atom addition as before. This time Olex2 will only add the missing hydrogen atoms. If for any reason, they are still incorrect, it will be necessary to add them manually. Also if hydrogen atoms were missing after ADD H, you will have to add them manually. Refer to page 5 how to do this.

5. The first assignment of hydrogen atom positions is not always fully optimised, particularly for less well defined groups such as –OH. If there are Q-peaks close to oxygen atoms that are not where the hydrogen is currently located, the hydrogen position may need adjusting. The image shows an incorrectly positioned hydrogen atom 1 which should be located at the Q-peak 2.

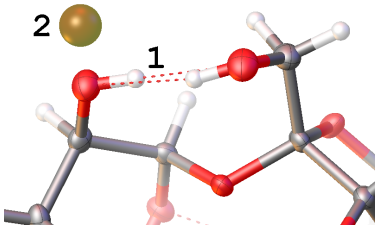


Figure 1.6: Incorrectly Positioned Hydrogen

6. To optimise the hydrogen atom positions there are two options: multiple refinement cycles using the REFINE button may see the hydrogen atom(s) rotate onto the Q-peak (sometimes updating the weighting scheme, see below, helps the refinement escape the false minimum and to rotate the H-atom(s) into the correct position). Alternatively, select a hydrogen atom and then in TOOLBOX WORK choose FIT option from the SELECT GROUP OR ATOM(S)AND THEN... tool. Once in FIT mode, holding the SHIFT key and LEFT MOUSE button down whilst dragging the relevant hydrogen atom allows you to move the hydrogen atom into the desired position. Press ESC when finished. Repeat if more than one atoms needs to be relocated.

   When in the FIT mode it is possible to rotate the structure to see if the relocated hydrogen atom is now on the Q-peak. Remove your finger from the mouse and SHIFT key, hold the LEFT MOUSE button down and move the mouse.

7. REFINE the structure.

> The refinement engine will automatically adjust the position to maintain the appropriate geometric constraints dictated by the AFIX instruction.

8. Rotate the structure and examine the hydrogen positions to make sure that the structure looks sensible. If required, repeat adjustments to the hydrogen atom positions.

9. Adjust the weighting scheme. Under the refinement options, there is a WEIGHT option. Tick the box next to AUTO UPDATE WEIGHTS. Click REFINE and repeat until the weights are no longer changing.

10. The structure is now solved and refined.

$R1$ will now be between 4–5%, wR_2 around 10% and the largest Q peak will be about 0.3.

1.4 Tidying the Structure

1. The key numbers to look at are $R1$, wR_2, *max* and *min* Q peak values (also called highest peak and deepest hole), *Goof*, *Hooft* and *Flack* which should be examined after each refinement cycle.

   - The values of $R1$ and wR_2 are an indication of the agreement between F_{calc} and F_{obs} and should decrease as the refinement proceeds. Very high values of $R1$/ wR_2 (e.g. > 40/70 respectively) suggest that the structure solution is incorrect.
   - **Q peaks**: *max* and *min* gives information on the highest peaks and deepest holes in the electron density map.
   - $GooF$ is the goodness of fit of a structure and should converge towards 1 at the end of the refinement.
   - **Hooft** y and **Flack** x are applicable only to structures in non-centrosymmetric space groups (*i.e. not* containing an inversion centre). They are displayed with an error and should be close to zero if the structure solution is correct, while ~1 implies the structure should be inverted and values significantly different from 0 or 1 indicate a racemic twin.

2. Check that the chemical composition is correct and there are no extra or missing atoms. If there is an error, in TOOLBOX WORK the atom types will not appear green. To update the chemical composition to agree with the structure displayed under WORK | TOOLBOX WORK click on OK and then REFINE again.

3. Go to the INFO tab and open the BAD REFLECTIONS header tab if there are one or two reflections that have very large $|F^2_{calc} - F^2_{obs}|/esd$. These values they may need to be omitted. Such reflections will be obvious as their values will be significantly out of line with the rest of the reflections. If it seems necessary to omit more than a handful of reflections, there is probably a reason that should be investigated.

> Do not omit reflections until all of the atoms have been located as the poorly fitting reflections are affected by missing atoms. For example low angle reflections often appear badly fitting prior to the addition of hydrogen atoms.

1.5 Checking the Structure

During a refinement there are a number of checks that should be made after each refinement cycle.

- Check that all of the *ellipsoids* are of a similar size throughout the structure and look sensible (the ellipsoid axes should not appear significantly different).
- Check that the correct number of *hydrogen atoms* are present and that they are appropriately positioned.
- Go to the INFO tab and open the *REFINEMENT INDICATORS* header tab. If there are any errors, some or all of the parameters will be orange (potential moderate problem) or red (potential serious problem). These should improve as the refinement proceeds. Attempts should always be made to correct these or be able to explain why they are not ideal.
- Check that the *bond lengths* and *angles* are sensible. There are several ways to do this:
  - Hover over a bond and the bond length will be displayed. Left click on the bond to select it.
  - Select 2 atoms for a bond length, 3 atoms for a bonded angle or 4 atoms to get a torsion angle. Then go to the VIEW tab, under the GEOMETRY header tab click on DISTANCE AND ANGLES (OF SELECTION).
  - Select all bond that you are interested in, then type `labels`. This will display the bond distances as labels on the bonds. You can also type `sel -l` to print a list of the distances or `sel -l -c` to then place this list on the clipboard, which you can then paste elsewhere.
  - The Cambridge Structural Database is an excellent resource to use to check that your bond lengths and angles are sensible and within expected ranges. If they are not, investigate why not.

> The `sel` command can be always used in the command-line to print information regarding the current selection.

1.5.0 What To Look For In a Good Refinement

A summary of the refinement indicators can be found under INFO | REFINEMENT INDICATORS. These can be used to check the refinement of your structure. Red colour indicates there is a problem with the refinement, green colour indicates a good refinement.

All of the ADPs are chemically sensible The thermal ellipsoids of the atoms should all look similar and of approximately the same dimensions. If there are some really large or really small ones, then there is something wrong. If there are some that are very elongated or very 'squashed', then that is also not right.

No residual least squares (L.S.) shift Check that the *Max Shift* is very close to 0 in the top panel, if it is not, the refinement has not converged and further cycles of refinement should be carried out. If convergence cannot be achieved using further cycles for refinement, then the reasons for this need to be investigated, for example overuse of restraints. Often you find that the largest shifts are associated with hydrogen atoms that have not been appropriately constrained. Click on the MAX SHIFT label that tells you about the largest shift and the 'culprit' will be selected.

Good R factors: $R1$ and wR_2 $R1$ and wR_2 should be as small as possible. wR_2 is always larger than $R1$ due to the way it is calculated. The expected values depend both on the type of structure that is being solved and the quality of the data e.g. for a molecular organic compound an $R1$ ~ 5–7% and wR_2 ~ 10–15% is reasonable. If heavy elements are involved, the final R-factors will be considerably lower. This is because a larger proportion of the electrons in the molecule are in the metal, and we can typically pinpoint the position of these very accurately. Associated with wR_2 (and also the weighting scheme) is the Goodness of Fit ($GooF$) parameter, which should be close to 1. If it deviates a lot (say, larger than 1.2 or smaller than 0.8), there is a problem.

The Highest Peak and Deepest Hole The *Highest Electron Density Peak* and *Deepest Electron Density Hole* should be small and approximately equal for a molecular organic compound ~0.5 electrons per Å^3. If the structure contains a heavy atom e.g. a transition metal, Cl or Br, slightly larger peaks might be expected particularly in the vicinity of the heavy element. You can click on the Q-peak and type `envi`, or right-click on it and then choose BANG (which stands for **B**onds **ANG**gles) to calculate the distance of any suspect residual peak to the heavy element.

Data/Parameter Ratio This one is a tricky one because there really is not much you can do to fix it if this should become a problem. The ratio of the number of reflections (observed parameters) to refinement variables (model parameters) should be no less than 8 to 1. If enough data have not been recorded, e.g. a very weakly diffracting crystal, very small sample or high pressure data – it may be necessary to reduce the number of refinement parameters using constraints.

1.6 Finalising the Structure

Your structure is now finished and completed from the refinement point of view. If the structure is to be included in a thesis or scientific publication, then you probably need to come up with a naming scheme for its atoms. If it is not your own structure, it is worth liaising with the chemist to agree on a naming scheme – structures are often part of a series which should have consistent naming throughout.

1. The next stage is to label your structure to a sensible naming scheme . In TOOLBOX WORK | LABELS select LABELS ON/OFF until the atom labels are displayed (alternatively press F3). Hide the Q-peaks either using the Q square in TOOLBOX WORK or use CTRL+Q.

   > Generally, it is ideal to name atoms in order e.g. all of the carbon atoms around a benzene ring 1-6 with substituents numbered 7 onwards down a chain, then all of the oxygen atoms etc. Alternatively, simply number atoms in order irrespective of atom type. If there are a series of similar structures it is sensible to use the same numbering scheme so that comparisons between bond lengths can be easily made.

2. Click on the NAMING header tab which is one of the options further down the GUI panel below TOOLBOX WORK.

3. Choose a starting number for your naming sequence, usually 1, and type it into the START box. Click on NAME on the right hand side of the Naming box. An orange box appears to tell you that you are now in mode Name.

4. Now with the LEFT MOUSE button click on the atoms in a sensible order to name them sequentially from your chosen start number. The atom labels of the atoms that have been clicked on change from green to black.

5. When finished press ESC. If you are labelling by atom types repeat the processes until all atoms are numbered.

   > A number should only be used once in association with a particular atom type. If one appears more than once, the two atoms with the same name will be labelled in red.

   By default, hydrogen atoms automatically take their name from the atom to which they are attached, if WORK | NAMING | AUTOMATIC HYDROGEN NAMING is ticked, any changes to atom labels are reflected in changes to the attached hydrogen atoms (changes take place after the refinement). If not, the hydrogen atoms may need relabeling separately, use `FixHL` to give you a good start – this will rename H-atoms to fit a commonly acceptable scheme.

6. Sort the atoms in the *.ins* file. Go to WORK | SORTING, which is one of the header options further down the GUI panel. Click on **Mass & Label**. Click on REFINE to update the model files.

7. Identify and tabulate intermolecular interactions if present.

   The `HTAB` instruction (type `help htab` for details) is very helpful here!

8. Generate a CIF and report for publication, and validate the CIF.

9. Produce relevant Figures.

1.6 Structures to Try Next

The next structure you should try now is sucrose (again). Despite the seemingly large number of steps required to obtain the finished structure, it is actually not all that hard. The key here is repetition: we normally recommend solving and refining sucrose several times before proceeding to another structure. Having followed this procedure through it should be possible to solve the other sample structures. Most of these are relatively easy, but the disorder in structure '183' is a real challenge.

Chapter 2

Olex2 Commands

This section describes some of the commands that are used 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.

2.1 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!

2.2 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.

2.3 Changing the Model View

• matr [1,2,3 or abc] or [abc a1b1c1] or [x11 x12 x13 y11 y12 y13 z11 z12 z13] *Orientates the model along; a (1 or `1 0 0`), b (2 or `0 1 0`), c (3 or `0 0 1`) or any other crystallographic direction e.g. `123`, which sets the current normal along the* $(1 \times a + 2 \times b + 3 \times c)$ *vector. Two crystallographic directions (from and to) may be specified to align the 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 `1 2 0 1 0 1` or `-1 +1 +1` (same as `-1 0 1 0 1`). If no arguments are given, the current Cartesian orientation matrix is printed.* **Options: -r**: uses the reciprocal lattice instead of the direct. **Examples:** `matr 1` or `matr a` or `matr 100` - sets the current normal along the crystallographic a direction. `matr 100 011` sets the current normal along the (`0 1 1 -- 1 0 0`) 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. It makes a continuous rotation around give axis when 5 arguments are provided. Note that the 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 the x axis. `rota 0 0 1 90 1` rotates the 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] [-rings] *Finds the best plane*

through the current selection or given atoms, or out of all visible atoms if none are given. **OPTIONS: -n**: sets the view along the normal of the plane. **-r**: creates a regular plane. **-rings**: creates planes for all rings given by a template like NC5.

• **zoom** *To get the current value of the scene zoom use:* `echo zoom()` *To set current zoom to a certain value use:* `zoom(eval(Value-zoom()))` *This can be used to put different structures on the same scale. Note that a value of 1 corresponds to the scale where the smallest dimension of the screen view is 1 Angstrom. To reset zoom to default for the current model use:* `gl.zoom`

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 or Alt key+left mouse button); shifted in the viewing plane by pressing Ctrl+Shift and holding the right mouse button down or by holding left and right mouse buttons down. The default mouse behaviour can be overridden in some modes (look at mode split) also some objects, like cell, basis and text boxes can override some mouse operations (like zooming on the cell basis) or extend it (moving the basis while holding Shift key down).

2.4 Fixed and Refined Parameters

• **fix {occu, xyz, U_{iso}} [atoms]** *Fixes the specified refinement parameter, i.e. these parameters will not be refined in subsequent refinement cycles.* **OPTIONS: -occu**: will fix the occupancy **-xyz**: will fix the xyz coordinates **-U_{iso}**: 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 (*i.e.* not refine them).

• **free {occu, xyz, U_{iso}} [atoms]** *The opposite of FIX - makes the specified parameters for the given atoms refineable. Freeing the occupancy is also available from the context menu.*

• **mode fixu** *Fixes U_{iso} or ADP for subsequently clicked atoms.*

• **mode fixxyx** *Fixes the coordinates for subsequently clicked atoms.*

• **mode occu occupancy_to_set** *Sets atom occupancy to the provided 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

2.5 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] [–m]** *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.* **OPTIONS: -a**: assembles broken fragments -c: similar to the default behaviour but considers atom-to-atom distances. It will move all atoms as close as possible to the largest fragment in the structure. **-q**: moves the electron density peaks closer to the atoms. **-m**: disconnects metals, then does compaq -a and then reattaches the metals.

• **addbond A1, A2 or atoms** *Adds a bond to the con-*

nectivity 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. Use this command to permanently remove bonds from the display too.*

• **sort [m] [l] [p] [h] [z] [n] [s] atoms [s] [h] [m] moiety** *The sorting of atoms in the atom list is very powerful but also quite complex.* **Options: -m**: atomic weight **-z**: atomic number **-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. **-s**: non-numerical label suffix **-n**: number after the atom symbol Sorting of moieties **-s**: by size **-h**: by heaviest atom **-m**: by molecular weight **Examples:** *sort [+atom_sort_type], sort [Atoms]- [moiety [+moiety_sort_type]- [moiety_atoms]]*. If just *moiety* is provided - the atoms will be split into the moieties without sorting. `sort +m1 F2 F1 moiety +s` will sort atoms by atomic mass and label, put F1 after F2 and form moieties sorted by size. Note that when sorting atoms, any subsequent sort type operates inside the groups created by the preceding sort types.

• **name [selection/atom names] [–c] [–s=]** *The command allows the atom names to be changed.* **Options: c**: checks if the generated names are unique. s: changes the suffix only (no value removes the suffix, *i.e.* the part of the label after the element symbol and numerical value) **Examples:** `name O1 O2` : renames O1 to O2 name 1: (some atoms selected). Sequentially names the atoms in order of the selection by adding 1,2, etc to the element symbol. Note that in this case if any generated name is not unique (and the -c option is not given), a random name will be generated.

`name $q C` : changes the element type of Q to C - all the electron density peaks will become carbons.

`name sel -s=a` : changes suffix of the selected atoms to 'a', replacing any existing suffix. Note that sel is a required keyword in this case (but may be removed in the future.

`name Q? C?` : changes the type for all electron density peaks with single number labels to carbon atoms, preserving the number

• **mode name [–p] [–s] [–t] [–a=0]** *Puts the program into the naming mode.* **Options: -p**: label prefix **-s**: label suffix **-t**: element symbol **-a**: autocomplete, turned off by default. Value 1 switches the autocompleting on, with value 2 stopping the procedure when an atom of a different type is encountered on the way. Value 4 - when an atom with a different part is encountered on the way and value 6 is selected when combining the cases of 2 and 4. A special value, 8, does automatic naming.

• **FixHL** *Updates H-atom labels according to the labels of the bearing atoms*

• **mode grow [–s] [–v] [–b] [–shells]** *Displays the directions in which the molecule can be grown.* **Options: -s**: also shows the short interaction directions. **-v**: [2.0 Å] shows directions to the molecules within the 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. If an atom is selected before entering this mode, the environment of only this atom(s) can be grown. **-r**: shows 'growing bonds' to symmetry equivalent atoms of the selected one(s) within 15 Å. Shortcut Ctrl+G is used to enter the 'mode grow' **-shells**: only applicable in `mode grow -shells` - allows growing atom by atom. If a 'grow' bond is clicked, only the immediate attached to that bond atom is grow, if the atom with outgoing 'grow' bonds is clicked - atoms for all bonds are grown

Olex2 will display the altered connectivity table in the case if structure is grown or packed.

2.6 Symmetry Operations

• **lstsymm** *Prints symmetry operations and their codes for the current structure.*
• **envi [r=2.7 Å] A1 or selected atom [–h] [–q]** *Prints a list of those atoms within a sphere of radius r around the specified atom. If more than one atom is selected, the one that was selected first is used.* **Options: -h**: adds hydrogen atoms to the list **-q**: adds Q-peaks to the list
• **mode grow [–s] [–v] [–b] [–shells]** *Displays the directions in which the molecule can be grown.* **Options: -s**: also shows the short interaction directions. **-v**: [2.0 Å] shows directions to the molecules within the 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. If an atom is selected before entering this mode, the environment of only this atom(s) can be grown. **-r**: shows 'growing bonds' to symmetry equivalent atoms of the selected one(s) within 15 Å. Shortcut Ctrl+G is used to enter the 'mode grow' **-shells**: only applicable in `mode grow -shells` - allows growing atom by atom. If a 'grow' bond is clicked, only the immediate attached to that bond atom is grow, if the atom with outgoing 'grow' bonds is clicked - atoms for all bonds are grown
• **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** *Generates symmetry equivalents of the selected atoms (or all atoms, if there is no selection) using the provided symmetry operation. Note: For symmetry operations starting with – and a letter, a leading zero must be added or the expression has to be quoted (for example, `0-x,-y,-z`), otherwise Olex2 confuses this with an option. The Symmetry operation is represented as `1_555`, `1555` or `-1+X,Y, \ensuremath{ Z}` and atoms as a selection or a names list. As a special case, twelve numbers can be provided to specify any matrix operating on the fractional coordinates (e.g. see the match)*
• **pack a_from a_to b_from b_to c_from c_to [atoms]** *Packs all or specified atoms within given dimensions.* **Options: -c**: prevents clearing existing model. **Examples:** `pack $O` will pack all O atoms with the default of -1.5 to 1.5 cells range. **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`* **Cell** *Shows content of the unit cell. In conjunction with `grow -w`, it allows the creation of views where all asymmetric units contributing to the unit cell are shown.* **Wbox** *Packs the volume of the structure inside a 3D selection box. You can select 3 atoms and type `sel wbox` to create a box around just that part of the structure. To keep already shown box around atoms and work on another part of the structure, use the −c option.* **r** *Packs fragments within radius r of the selected atom(s) or the centre of gravity of the asymmetric unit.*
• **grow [atoms] [–w] [–s]** *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.* **Options: -w**: permits the application of previously used symmetry operations to other fragments of the asymmetric unit. In other words: if parts of the structure have been grown, this command will also generate symmetry equivalent atoms that are not connected to the already grown fragment, *i.e.* solvent molecules. **Examples:** 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.

2.7 Constraints and Restraints

• **exyz atom types (to add for the selected atom) [–EADP] [–lo]** *Makes the selected site be shared by atoms of different types.* **OPTIONS: -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 restraints 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 a given set of bond pairs, sharing an atom or atom triplet generates two `DFIX` commands and one `DANG` command.* **EXAMPLES:** `tria 1 1 180 C1 C2 C3` will generate `DFIX 1 0.02 C1 C2 C2 C3`.
`DANG 2 0.04 C1 C3` will calculate the distance for DANG from d1 d2 and the angle.

• **rrings [d=1.39] [esd=0.01] ring_content or selection** *Finds rings using the selection or rings content (like C6). It also sets DFIX restraint for the bond lengths using the d parameter and FLAT with e.s.d. of 0.1 restraint for the ring. It also adds SADI restraints for the 1-3 distances. If d is negative, the SADI restraint is used instead.*

• **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 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 list of given atoms to be similar, taking the provided esd into account.*

• **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 .* {ins} file, however mixing rigid group constraints and the SAME instructions might lead to an erroneous instruction file. Note that if only two atoms are selected in two fragments with identical connectivity, Olex2 employs the matching procedure and sets SAME for the two fragments to which the atoms belong.

• **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.* **OPTIONS: -r**: adds specific restraints/constraints (EADP, ISOR or SIMU) for the generated atoms

• **afix number{mn} [–n]** *If no atoms are 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. In 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 as dependent atoms. The AFIX command can also be used to generate missing atoms to complete rings or fragments. For example, the following command generates three missing atoms in positions 4,5 and 6 for the Ph ring when applied to*

a selection of 3 atoms (assumed to be in positions 1, 2 and 3): `AFIX 66 1,2,3` *Note that there are no white spaces between the identification of the selected positions.* **Options: -n**: consider *n*-atoms as parts of rings

• **part [part=new_part] [occupancy] [atoms] [–p=1]** *Changes the part number/occupancy for selected atom* **Options: -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. If the number of parts is greater than 2 and -lo option is given, a new SUMP restrain will be automatically added.

• **fvar [part=new_part] [occupancy] [atoms] [–p=1]** *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] [part=new_part] [occupancy] [atoms] [–p=1]** *Creates a new linear equation. If any of the selected atoms has refinable or fixed occupancy, a new variable is added with the value 1/(number of given atoms). Otherwise, an already used variable is used with weight of 1.0. Also look at* `part` *command.* **Examples:** If 3 atoms (A1, A2, A3) are selected, this command will generate three free variables (var1, var2 and var3) and insert the `1 1 var1 1 var2 1 var3` instruction (equivalent to

$$1 = 1 \times occ(A1) + 1 \times occ(A2) + 1 \times occ(A3)$$

• **mode split [–r={eadp, isor, simu}]** *Splits subsequently clicked atoms into parts. While in this mode, the newly generated atoms can be selected and moved as a group with Shift key down, or rotated when dragging the selection. The original and generated atoms will be placed into different parts.* **Options: -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

• **mode fit [–s]** *Allows fitting of selected group (moving and rotating in 3D).* **Options: -s**: a new group is created at the fitted location. The occupancy of this and the original group is constrained to be 1.

• **ImportFrag [–a] [–d] [–p]** *Import and . {*xyz} file into the current model. Mode fit is automatically executed to help with fitting the imported molecule. **Options: -a**: sets AFIX to the imported molecule **-d**: generates DFIX for 1,2 and 1,3 distance for the imported molecule **-p**: sets given part to the imported molecule

• **restrain ADP [Ueq] {Ueq, volume}1** *This is a generic macro to generate restraints. By default, the Ueq/volume similarity restraint is generated. If the ADP Ueq/volume is unknown then the restraint value will be randomly generated.* **Bond [d atoms] Angle value [atoms] Dihedral value [atoms]** **Note:** Only available in Olex2.refine

• **constrain U [atoms] A Site or xyz [atoms] A Same group [n=2 atoms]** *This is a generic macro to generate constraints.* The same group or non-crystallographic symmetry constraint makes two or more groups identical and linked through a transformation matrix, refined as a shift and 3 Euler angles. If two atoms are given, they must belong to two identical fragments; Olex2 will then try to match the fragments containing the atoms and automatically generate the constraint. In more generic/complex cases the user has to provide the number of groups to generate the constraint for and also the selection which matches atoms in the fragments.

• **xf.rm.ShareADP1 [atoms]** *Generates the shared, rotated ADP constraint. For 3 atoms, it generates an ADP rotated around the bond e.g. around X-C bond in X-CF_3. For more atoms, it creates an ADP rotated around the normal of the plane formed by the atoms.*

2.8 Selection Syntax

sel

- **155**: all symmetry generated atoms currently shown, which were generated by the given symmetry operation.
- **rings NC5**: all pyridine rings
- **fvar -2**: atoms where a parameter is linked to FVAR 2
- **part 1**: all atoms in PART 1
- **isot**: all isotropic atoms
- **frag C5**: the whole fragment containing C5
- **$E**: all atoms of the given type. E is a chemical element symbol or one of the following: * - all types, M - all metals, X - all halogens
- $,***E****: all atoms,* but* of the E type.
- $*,**H**: all non-H atoms
- **wbox**: Shows the 3D selection box constructed for all/selected atoms. This box can be used to pack the structure or to crop the voids display/electron density maps. If the third argument (cell) is provided, the frame gets the dimensions of the unit cell rather than being rectangular (this box can be used for packing and 3D maps trimming/extending).
- **ofile** *n*: slects all atoms in the overlayed file *n*; if *n* is 0, elements of the currently focused file are selected

sel atoms where

- **xatom.bai.mw** > 20: Select all atoms where the atomic mass is larger than 20
- **xatom.bai.z** > 2: Select all atoms where the atomic number is greater than 2
- `atoms` : all atoms

sel bonds where

- **xbond.length > 2**: all bonds longer than 2 Å
- **xbond.b.bai.z == 1**: all bonds where the lightest atoms is H
- `atoms` : all atoms

2.9 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. Use* `delIns omit` *to remove all the OMITs from the INS file header.* `omit val` *Inserts* `OMIT h k l` *for all reflections with* $|F_o^2 - F_c^2| > val$ **omit s 2theta** *Inserts* `OMIT s 2theta` *instruction in the ins file*
• **hkledit [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 contrast to the* `OMIT h k l` *instruction, which excludes the reflection and all it equivalents, this dialogue allows the exclusion of those equivalents that are actually outliers. If a particular reflection is specified, this particular reflection and all its constituent equivalents can be viewed.*
• **hklexclude [–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). **Options: -c**: option controls how the given indices are treated. If no **-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.
• **hklappend -h=h1;h2;.. -k=k1;k2.. -l=l1;l2..** *Acts in the opposite way to* `excludehkl` .
• **hklview [–h=h1;h2;.. -k=k1;k2.. -l=l1;l2.. [–c]** *Shows the reflection currently used in the refinement (use Ctrl+T a few times to centre on the reflection view).*

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.

2.10 Customising Olex2

• **setfont {Console, Picture_labels}** *Brings up the dialog to choose a font for the Console or Labels, which end up on the picture. Use the built in function 'choosefont([olex2])' to choose a system. Alternatively, a specially prepared/portable font can be used to specify the font.*
• **editmaterial {helpcmd, helptxt, execout, error, exception, any object name available with lstgo}** *Brings up a dialog to change properties of the specified text section or graphical object.* **Options: helpcmd**: the command name in the help window **helptxt**: the body of the help item **execout**: the output text printed in the console of external programs **error**: reporting errors in the console **exception**: reporting exception in the console This command can be used to edit properties of any objects printed by `lstgo` macro. An example of that could be editing material of the console text: **Examples:** `EditMaterial Console`. The the object name is case sensitive.
• **save {scene, style, view, model} [file_name]** *If the file name is not provided, the* Save as*... dialog will be shown which allows you to save current settings to file. The scene saves current font names/sizes as well as the materials for the specific console output e.g. external programs output, error and exception reporting. The style saves information about the appearance of objects in the scene. The view saves current zoom and the scene orientation. The model saves current view including the crystallographic mode and style.*
• **load {scene, style, view, model, radii} [file_name]** *Load one of the previously saved items. If no file name is provided, the 'Open file...' dialog will appear. If just a file name is provided (the extension will be guessed by Olex2); for styles and scene the last used folders will be used by default, the current folder will be used for the views and models. Loading radii (vdw, pers, sfil) allows the user to change the radii Olex2 uses for various calculations/display.*
• **grad [C1 C2 C3 C4] [–p]** *Choose the colour of the four corners of the graduated background.* **Options: -p**: a file name for the picture to be placed in the background
• **brad *r* [hbonds]** *Adjusts the bond radii in the display. If* hbonds *is provided as the second argument, the given radius* r *is applied to all hydrogen bonds. This operates on all or selected bonds.*
• **ads {elp, sph, ort, std}** *A function for drawing styles development. Changes atom drawing style for all/selected atoms.* **elp**: represents atoms as ellipsoids (if ellipsoids are available) **sph**: represents atoms as spheres **ort**: same as elp, but spheres have one of the quadrants cut out **std**: a stand-alone atom (*i.e.* shown 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.* **Options: sfil**: sphere packing radii (as in ShelXTL ***XP***) **pers**: a fixed radii for model viewing **isot**: each atom has its own radius depending on the value of the U_{iso} or ADP **isoth**: same as isot, but the H atoms are also displayed with their real U_{iso}'s **bond**: all atoms get the 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 as a percentage.*

2.11 Tables, Reports and Images

• **pictps filename.ps** *Generates a post-script file of what is visible in the molecule display. The bond width is taken from the display. This can be changed with* `brad`. **-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 of stripes in the octant [4] **-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]

• **pict filename.ext [n=2] [–pq] [–dpi]** *Generates a bitmap image of what is visible on the molecule display. n refers to the size of the output image. If n<10, it refers to a multiple of the current display size. If* `n>100`*, it refers to the width of the image in pixels. ext {png, jpg, bmp}. png is best.* **Options: -pq**: print quality. **-nbg**: removes the background from the picture (making it transparent with the alpha channel). **-dpi**: physical resolution of the image.

• **picta filename.ext [n=1] [–pq] [–dpi]** *A portable version of* `pict` *with limited resolution (see explanation for* n *in* `Pict`*), which is OS and graphics card dependent. This function will also use the graphics card settings like antialiasing when producing the picture.* **Options: -pq**: print quality **-nbg**, **-dpi**: same as for `pict`

• **pictpr filename** *Creates PovRay file for current view*

• **picts filename.ext [n=1] [–a=6] [–s=10] [**$-h = n \times (screenheight)$**]** *Creates a 'stereo' picture with two views taken with the* $\pm$ *a option. Value is rotated around y axis and placed into one picture separated by s % of a single projection width.* **-a**: half of the view angle **-s**: separator width in % **-h**: the height of the output, by default equal to current screen height multiplied by the given resolution.

• **label [atoms]** *Adds labels to all/given/selected atoms and bonds. These labels can be moved by pressing the Shift key while holding down the Left Mouse button, and edited by double-clicking on them.* **Options:** -type: {subscript, brackets, default -symm: { [$], #, full} - if an atom is generated by non-identity symmetry operation, it will be added as a superscript. Note that if # is used as the symmetry identifier then the labels of every type of new atom generated (in growing or packing) should be recalculated. Olex2 will then print current mapping of the symmetry numbers to the symmetry operators.

• **pim** *Helps with creating pictures when metal - 'π interactions need special drawing (e.g. in the case of Cp-Me, a single bond will be rendered from the ring centroid to the metal vs 5 bonds from every C atom to the metal). Note that currently, these stippled bonds do not appear in the PostScript rendered pictures and thus for PS pictures, a workaround with creating centroids is needed. To remove all of the lines created by this command - right click on one of them and choose Graphics->Select the group, then hit the DEL key.*

2.12 Structure Analysis

• **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 a 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 instruction will be added.* **-t**: adds extra elements (comma separated like in -t=Se,I) to the donor list. Defaults are [N,O,F,Cl,S,Br] **-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] [–r=C6,NC5]** *The command analyses the π– 'π 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] Åand the intercentroid shift is smaller than [3] Å.* **Options: -g**: if any of the rings is fully or partially constructed of symmetry generated atoms, it grows the structure using those symmetry operators. **-r**: ring content, the defaults are C6 and NC5 rings, the rings

are tested for being flat and regular.

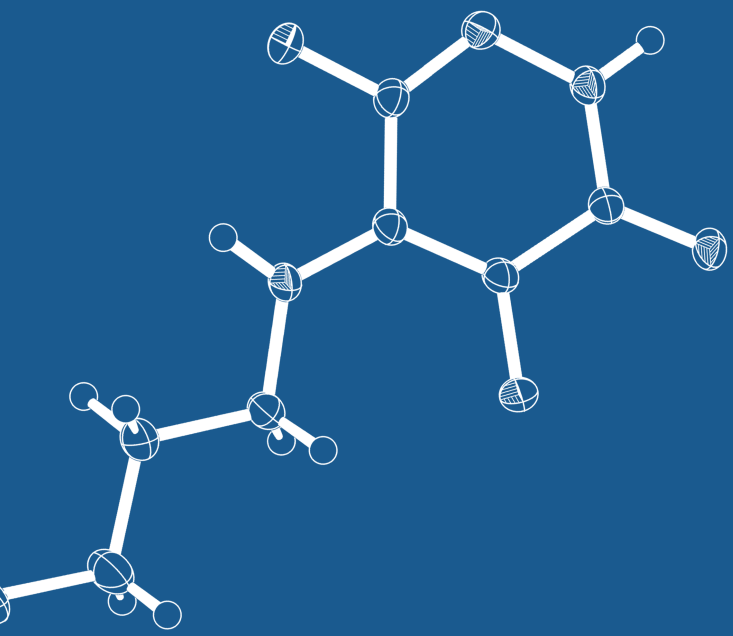
• **calcvoid [radii file name] [all atoms/selected atoms] [–d=0] [–p] [–r=0.2 Å]** *Calculates and displays the structure map. Also calculates the largest channels along crystallographic directions and the packing index.* **Options: -d**: extra distance from the surface (added to the atomic radii) **-p**: precise calculation where each map voxel is tested. The default quick algorithm uses the atom masks to find the volume occupied by the molecule. The precise calculation is vectorised. **-r**: resolution, a resolution of at least 0.1 Åand -p option 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 which is to provide a file name with radii ([element radius] a line format). The other 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=o]** *Calculates molecular volume and surface area for all/selected atoms.* **Options: -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 by the same factor.

• **calcfourier {-calc,- diff, -obs, -tomc} [–r=0.25 Å] [–i] [–scale=simple] [–fcf]** *Calculates Fourier for current model.* **-r**: the resulting map resolution in Å-**i**: integrates the calculated map **-scale**: when Olex2 calculates structure factors, it uses the linear scale as a $sum(F_o^2)/sum(F_c^2)$ 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).

• **kill Atom names or selection or labels** *Deletes given or selected atoms, bonds or labels. Note that if the bonds are deleted this way they will reappear the next time the structure connectivity is updated. Use* `delbond` *to remove bonds permanently.*

• **match [atoms] [–a] [–w] [–i] [–n] [–u] [–esd] [–h] [–cm] [–o]** *This procedure finds the relationship between the connectivity graphs of molecular fragments of loaded structure and aligns the fragments. If no arguments are given, the procedure analyses all fragments and in the case when fragments with matching connectivity are found, it aligns Acta A45 (1989), 208 and prints corresponding root mean square distance (RMSD) in angstroms. If two atoms are provided (explicitly by name or through the selection) the graph relation information - orientation matrix and the matching atoms - is printed. Use* `-a` *option to align the fragments.* **-a**: align the fragments (used when a pair of atoms are provided) **-w**: specifies weight for the atomic positions - by default the unit weights are used. If this option is given, the atomic position are weighted by the element mass. **-i**: try to invert one of the fragments. **-n**: transfers labels from one fragment to another (two atoms should be provided as 'to' and 'from' fragments). If the value is a symbol (or a set thereof) this is appended to the label. `$xx` replaces the symbol after the atom type symbol with xx, leaving the ending. `-xx` changes the ending of the label TO xx. Note that if the molecules match with `-i` options, this should also be provided for the label transfer. **-u**: restores the coordinates of the matched fragments - this is useful if the grown structure is matched. **-esd**: if the variance-covariance matrix can be located (after refinement with the MORE negative option in the xl), the esd on the RMSD can be calculated using this option. **-h**: calculates the final matching and RMSD calculation without taking H-atoms into account. **-o**: when overlaying molecules from different structures whole lattices (if packed/grown) are overlayed, not only the two fragments. To use, select an atom in a fragment of one lattice and an atom in a matching fragment of the other lattice. When a selection of two atoms is given the command prints the alignment matrix. This matrix alongside the `sgen` command can be used to generate new atoms. Use the `-cm` option to copy the matrix to the clipboard.



Olex2 is a powerful tool for small molecule crystallography and is designed to be simple to use for novice users whilst providing complex functionality and tools for experts. Olex2 provides features for a complete crystallographic structure determination from structure solution and refinement, through report writing and CIF generation to producing images.

OlexSys