Olex2 hackers guide

The easiest way to get into the code of Olex2 and to program extensions to Olex2 is to create a custom script in "[Olex2-base-directory]\etc\scripts".
You can program any Python-2.7 module by creating a file with the ending .py and start programming. In order to get the script working in Olex2 you have to import your script in the "customScripts.py" file like

```
import myscript.py
```

Your first code might look like the following:

```
def hello_world_example():
    '''This prints a Hello World"
    print("Hello world!!")
```

Finally you have to register the function so that Olex can run it:

```
OV.registerFunction(hello_world_example)
```

Now you can run the script with **spy.hello_world_example** in the Olex2 command line:

```
≫ spy.hello_world_example
Hello world!!!
```

You can get the source code for Olex2 development from:

```
svn checkout svn://svn.code.sf.net/p/olex2/code/trunk olex2-code
```

and

```
svn checkout svn://svn.olex2.org/olex2-gui/trunk olex2-gui
```

Useful Olex2 commands (much more on commands in Ilias brilliant "Notes on Olex2" http://xray.chem.wisc.edu/Resources/Manuals/Ilia_Guzei_notes_on_OLEX2.pdf):

```
shell BaseDir()          Opens the base install directory of Olex2 in a file
                         manager. Place plugin scripts into
                         "BaseDir()/etc/scripts/".
shell DataDir()          Opens the data directory of Olex2 in a file manager.
Restart()                Restarts Olex2.
echo sel()               print selected atoms
echo GetUserInput(2, 'some text', 'some SHELX command')
                         insert special user input lines into the res file.
                         1 means one line, 2 means multiline
```

All of the regular Olex2 commands for user interaction could also be used in Python scripts. A complete list follows:

Packages provided by Olex2:

Index

olx. (package) 5
  olx.**app** (package) 5
  olx.**cell** (package) 5
  olx.**console** (package) 5
    olx.console.**cursor** (package) 5
  olx.**fader** (package) 6
  olx.**file** (package) 6
  olx.**fs** (package) 7
  olx.**gl** (package) 7
    olx.gl.**lm** (package) 7
      olx.gl.lm.**light1** (package) 7
      olx.gl. lm.**light2** (package) 8
      olx.gl. lm.**light3** (package) 8
      olx.gl. lm.**light4** (package) 8
      olx.gl. lm.**light5** (package) 8
      olx.gl. lm.**light6** (package) 8
      olx.gl. lm.**light7** (package) 8
      olx.gl.lm.**light8** (package) 8
    olx.gl.**scene** (package) 8
  olx.**html** (package) 9
  olx.**lcells** (package) 11
  olx.**math** (package) 11
  olx.**mouse** (package) 11
  olx.**py** (package) 11
  olx.**time** (package) 12
  olx.**wbox** (package) 12
  olx.**xf** (package) 12
    olx.xf.**analysis** (package) 12
    olx.xf.**au** (package) 12
    olx.xf.**exptl** (package) 14
    olx.xf.**latt** (package) 14
    olx.xf.**rm** (package) 14
    olx.xf.**uc** (package) 15
  olx.**xgrid** (package) 15
    olx.xgrid.**label** (package) 15
    olx.xgrid.**legend** (package) 16
__init__(self) 17
AddIns(self, instruction, quiet=False) 17
AtReap(self, path) 17
BaseDir(self) 17
CifMerge(self, filepath, update_atoms_loop=None, report=True) 17
CopyVFSFile(self, copy_from, copy_to, isPersistent=0) 17
CurrentLanguageEncoding(self) 17
DataDir(self) 17
DelIns(self, instruction) 17
File(self, filename=None) 17
FileDrive(self, FileDrive=None) 17

FileExt(self, FileExt=None) 17
FileFull(self) 17
FileName(self, FileName=None) 17
FilePath(self, FilePath=None) 17
FindObject(self, variable) 17
FindValue(self, variable, default=u'') 17
GetCellVolume(self) 17
GetCompilationInfo(self) 17
GetComputername(self) 17
GetCrystalData(self) 17
GetCurrentSelection(self, calculate_connectivity=False) 17
GetDampingParams(self) 17
GetExtinction(self) 17
GetFVar(self, i) 17
GetFormula(self) 17
GetHtmlPanelX(self) 17
GetKeyname(self) 17
GetMacAddress(self) 17
GetOSF(self) 17
GetParam(self, variable, default=None) 17
GetParam_as_string(self, variable, default=None) 17
GetRefinementModel(self, calculate_connectivity=False) 17
GetSVNVersion(self) 17
GetTag(self) 17
GetUserComputerName(self) 17
GetUsername(self) 17
GetValue(self, control_name) 17
GuiParams(self) 17
HKLSrc(self, new_HKLSrc='') 17
HasGUI(self) 17
IsFileType(self, fileType) 17
IsPluginInstalled(self, plugin) 17
IsVar(self, variable) 17
ListFiles(self, dir_name, mask=None) 18
Lst(self, string) 18
Params(self) 18
Reap(self, path) 18
Reset(self) 18
SetExtinction(self, v, e=None) 18
SetFVar(self, i, v) 18
SetHtmlFontSize(self) 18
SetHtmlFontSizeControls(self) 18
SetMaxCycles(self, max_cycles) 18
SetMaxPeaks(self, max_peaks) 18
SetOSF(self, v) 18
SetParam(self, variable, value) 18
SetVar(self, variable, value) 18
StoreParameter(self, var='', save=False) 18

Translate(self, text) 18
TranslatePhrase(self, text) 18
XfAuGetValue(self, var='') 18
cmd(self, command) 18
external_edit(self, filePath) 18
file_ChangeExt(self, path, newExt) 18
func_wrap(self, f) 18
getCompatibleProgramName(self, name) 18
get_cif_item(self, key, default='', output_format=False) 18
get_txt_from_vfs(self, item) 18
htmlPanelWidth(self) 18
htmlUpdate(self) 18
makeGeneralHtmlPop(self, phil_path, htm='htm', number_of_lines=0) 18
olex_function(self, str) 18
registerCallback(self, event, function, profiling=False) 18
registerFunction(self, function, profiling=False, namespace='') 18
registerMacro(self, function, options, profiling=False) 18
reloadStructureAtreap(self, path, file, fader=True, sg_changed=False) 18
reset_file_in_OFS(self, fileName, txt=' ', copyToDisk=False) 18
setAllMainToolbarTabButtons(self) 18
set_cif_item(self, key, value) 18
set_refinement_program(self, program, method=None, scope='snum') 18
set_solution_program(self, program, method=None, scope='snum') 18
standardizeListOfPaths(self, list_of_paths) 18
standardizePath(self, path) 18
timer_wrap(self, f, *args, **kwds) 18
unregisterCallback(self, event, function, profiling=False) 18
unregisterFunction(self, function, profiling=False) 18
unregisterMacro(self, function, options, profiling=False) 18
write_to_olex(self, fileName, text, copyToDisk=False) 18
Alert(self, title, text, buttons=None, tickboxText=None) 18
CreateBitmap(self, bitmap) 19
Cursor(self, state='', text='') 19
DeleteBitmap(self, bitmap) 19
GetFormulaDisplay(self) 19
GetHtmlPanelwidth(self) 19
GetUserInput(self, arg, title, contentText) 19
HtmlDefineControl(self, d) 19
HtmlLoad(self, path) 19
IsControl(self, ctrl_name) 19
Listen(self, listenFile) 19
Refresh(self) 19
SetGrad(self, f=None) 19
SetImage(self, zimg_name, image_file) 19
UpdateHtml(self, html_name='') 19
setDisplayQuality(self, q=None) 19
setItemstate(self, txt) 19

olx. (package)
PACKAGE CONTENTS

Handles most functions of Olex2.

olx.**app** (package)
PACKAGE CONTENTS
FUNCTIONS
ArgCount(*args)
Returns number of arguments passed to the application.
BaseDir(*args)
Returns the directory from which the application is launched.
ConfigDir(*args)
Returns the configuration directory. If it is not set, the InstanceDir is returned.
GetArg(*args)
Returns application argument value by index.
GetLogName(*args)
Returns current log file name.
GetOpt(*args)
Returns application 'option=value' value by index. '=' only added if the values is not empty.
InstanceDir(*args)
Returns the instance specific, writable directory.
IsBaseDirWritable(*args)
Returns true if the application can write to the BaseDir().
IsDebugBuild(*args)
Returns true if the application is built with debug info.
ModuleHash(*args)
Returns current module MD5 hash.
OptCount(*args)
Returns number of options passed to the application.
OptValue(*args)
Returns value of the given option (default may be provided).
Platform(*args)
Returns current platform like WIN, MAC, Linux 32/64.
Profiling(*args)
Sets/Returns current procedure profiling status.
SaveOptions(*args)
Saves options to ConfigDir/.options file.
SharedDir(*args)
Returns a generic writable directory.

olx.**cell** (package)
PACKAGE CONTENTS
FUNCTIONS olx.cell.
DrawStyle(*args)
Returns or sets current cell drawing style [line,cone].
Thickness(*args)
Returns or sets current cell thickness.

olx.**console** (package)
PACKAGE CONTENTS

olx.console.**cursor** (package)
PACKAGE CONTENTS
FUNCTIONS olx.console.cursor.
Symbol(*args)
Returns or sets current symbol used to draw the cursor.

FUNCTIONS olx.console.

Clear(*args)
Clears the content of the output buffer.

Command(*args)
Changes/returns current command.

GetName(*args)
Returns object collection name.

IsGrouped(*args)
Returns true if the object is in a group.

IsSelected(*args)
Returns true if the object is selected.

LineSpacing(*args)
Changes/returns current line spacing.

Lines(*args)
Sets/returns the number of lines to display.

Post(*args)
Adds provided text to the output buffer.

PromptString(*args)
Changes/returns current prompt string.

ShowBuffer(*args)
Shows/hides the output buffer or returns current status.

Visible(*args)
Changes/returns object visibility.

olx.**fader** (package)

PACKAGE CONTENTS

FUNCTIONS olx.fader.

BG2FG(*args)
Copies current background frame to foreground frame.

GetName(*args)
Returns object collection name.

InitBG(*args)
Initialises xfader background frame.

InitFG(*args)
Initialises xfader foreground frame.

IsGrouped(*args)
Returns true if the object is in a group.

IsSelected(*args)
Returns true if the object is selected.

Position(*args)
Sets/returns current xfader position.

Step(*args)
Sets/returns xfader increment.

Visible(*args)
Changes/returns object visibility.

olx.**file** (package)

PACKAGE CONTENTS

FUNCTIONS olx.file.

AbsolutePath(*args)
Returns an absolute path to a folder relative to the basedir; arguments are (base=basedir,path).

Age(*args)
Returns file age for provided file using formatting string (if provided).

ChDir(*args)
Changes current folder to provided folder.

ChangeExt(*args)
Returns file name with changed extension.
Copy(*args)
Copies file provided as first argument into the file provided as second argument.
CurDir(*args)
Returns current folder.
Delete(*args)
Deletes specified file.
Exists(*args)
Returns true if specified file exists.
GetDrive(*args)
Returns drive component of the full file name.
GetExt(*args)
Returns file extension.
GetName(*args)
Returns name part of the full/partial file name.
GetPath(*args)
Returns path component of the full file name.
ListDirForGUI(*args)
Returns a ready to use in GUI list of files, matching provided mask(s) separated by semicolon. The third, optional argument [f,d,fd] specifies what should be included into the list.
MkDir(*args)
Creates specified folder.
OSPath(*args)
Returns OS specific path for provided path.
RelativePath(*args)
Returns a path to a folder relative to basedir; arguments are (base=basedir,path).
Rename(*args)
Renames specified file.
Which(*args)
Tries to find a particular file looking at current folder, PATH and program folder.

olx.**fs** (package)

PACKAGE CONTENTS

FUNCTIONS olx.fs.

Clear(*args, **kwds)
Clear the content of the VFS. A mask [-1] can be used to remove only items with particular persistence mask.
Dump(*args, **kwds)
Saves a file in the VFS to the disk file
Exists(*args)
Returns true if the specified file exists on the virtual file system.

olx.**gl** (package)

PACKAGE CONTENTS

olx.gl.**lm** (package)

PACKAGE CONTENTS

olx.gl.lm.**light1** (package)

PACKAGE CONTENTS

FUNCTIONS olx.gl.lm.light1.

olx.gl.lm.light1.
Ambient(*args)
Returns/sets enabled property of the light.
Attenuation(*args)
Returns/sets enabled property of the light.
Diffuse(*args)
Returns/sets enabled property of the light.
Enabled(*args)
Returns/sets enabled property of the light.
Position(*args)
Returns/sets enabled property of the light.
Specular(*args)
Returns/sets enabled property of the light.
SpotCutoff(*args)
Returns/sets spot cutoff property of the light.
SpotDirection(*args)
Returns/sets enabled property of the light.
SpotExponent(*args)
Returns/sets enabled property of the light.

olx.gl. lm.**light2** (package)

olx.gl. lm.**light3** (package)

olx.gl. lm.**light4** (package)

olx.gl. lm.**light5** (package)

olx.gl. lm.**light6** (package)

olx.gl. lm.**light7** (package)

olx.gl.lm.**light8** (package)

FUNCTIONS olx.gl.lm.
AmbientColor(*args)
Returns/sets ambient color of the model.
ClearColor(*args)
Returns/sets background color of the model.
LocalViewer(*args)
Returns/sets local viewer property of the model.
SmoothShade(*args)
Returns/sets smooth shading of the model.
TwoSides(*args)
Returns/sets two sides coloring of the model.

olx.gl.**scene** (package)
PACKAGE CONTENTS
FUNCTIONS olx.gl.scene.
MakeCurrent(*args, **kwds)
Make scene for rendering/updates.

FUNCTIONS olx.gl.
Basis(*args)
Returns/sets view basis.
CalcZoom(*args)
Returns optimal zoom value.
Compile(*args)

Compiles or decompiles the model according to the boolean parameter.

`Fog(*args, **kwds)`
Sets fog color, fog without arguments removes fog.

`GetZoom(*args)`
Returns current zoom value.

`LineWidth(*args)`
Returns/sets width of the raster OpenGl line.

`Perspective(*args, **kwds)`
Un/Sets perspective view.

`RasterZ(*args)`
Returns/sets maximum value of the raster Z 1 or -1 is typically expected.

`Stereo(*args)`
Returns/sets color/cross/anaglyph/hardware stereo mode and optionally stereo angle [3].

`StereoColor(*args)`
Returns/sets colors for left/right color stereo mode glasses.

`Zoom(*args, **kwds)`
If no arguments provided - resets zoom to fit to screen, otherwise increments/decrements current zoom by provided value.

olx.**html** (package)

PACKAGE CONTENTS

FUNCTIONS olx.html.

`Call(*args)`
Calls event of specified control, expects [popup.]control.event.

`ClientHeight(*args)`
Returns/sets client height of an HTML window (use 'self' to address the window itself).

`ClientWidth(*args)`
Returns/sets client width of an HTML window (use 'self' to address the window itself).

`ContainerHeight(*args)`
Returns/sets client height of an HTML window (use 'self' to address the window itself).

`ContainerWidth(*args)`
Returns/sets width of a popup window.

`DefineControl(*args, **kwds)`
v – value
i - tems
c - checked/down
bg - background color
fg - foreground color
min - min value
max - max value
Defines a managed control properties

`Dump(*args, **kwds)`
Saves content of the main or given page into the file.

`EndModal(*args)`
Ends a modal popup and sets the return code.

`GetBorders(*args)`
Returns borders width between HTML content and window boundaries.

`GetData(*args)`
Returns data associated with specified object.

`GetFontName(*args)`
Returns current font name.

`GetImage(*args)`
Returns image source for a button or zimg.

`GetItemState(*args)`
Returns item state of the given switch.

`GetItems(*args)`
Returns items of a combobox or list.

`GetLabel(*args)`
Returns labels of specified object. Applicable to labels, buttons and checkboxes.

`GetState(*args)`
Returns state of the checkbox or a button. For example: echo getstate(button, enabled/down) or echo getstate(checkbox).

`GetValue(*args)`
Returns value of specified object.

`Group(*args, **kwds)`
Creates an exclusive group of buttons.

`Height(*args)`
Returns/sets height of an HTML object window (use 'self' to address the window itself).

`Hide(*args, **kwds)`
Hides an Html popup window.

`Home(*args, **kwds)`
Reloads the page.

`IsEnabled`
Returns true if specified control is enabled

`IsItem(*args)`
Returns true if specified switch exists.

`IsPopup(*args)`
Returns true if specified popup window exists and visible.

`ItemState(*args, **kwds)`
u - does not update the html.
Changes state of the HTML switch, accepts masks like '*-picture-*'

`Load(*args, **kwds)`
Loads content into main or given HTML page. Example: load name file_name.

`LoadData(*args)`
Saves state, data, label and value of all objects to a file.

`LstObj(*args, **kwds)`
Prints the list of available HTML objects.

`Select(*args)`
Selects a treeview item by label (default) or data (third argument should be False).

`SetBG(*args)`
Sets background of specified object.

`SetBorders(*args, **kwds)`
Sets borders between HTML content and window edges.

`SetData(*args)`
Sets data for specified object.

`SetEnabled`
Enables/disables the control

`SetFG(*args)`
Sets foreground of specified object.

`SetFocus(*args)`
Sets input focus to the specified HTML control.

`SetFonts(*args, **kwds)`
Sets normal and fixed fonts to display HTML content [html normal_face fixed_face].

`SetImage(*args)`
Sets image location for a button or a zimg.

`SetItems(*args)`
Sets items for comboboxes and lists.

`SetLabel(*args)`
Sets labels for a label, button or checkbox.

`SetState(*args)`

Sets state of a checkbox or a button.

SetValue(*args)
Sets value of specified object".

ShowModal(*args)
Shows a previously created popup window as a modal dialog.

Snippet(*args)
Loades a file (first arg), replaces #name from name=val for following params and returns the result.

Tooltips(*args, **kwds)
Enables or disables tooltip for HTML. If no arguments is given the state of tooltops for the main HTML is inverted. If a single boolean argument is given - the state of the tooltips is set to the given value, if the argument is not a boolean the state of the tooltips for HTML windows with given name is inverted. If two arguments are given - the first should be an HTML window name and the second - a boolean value. This function executes the htmltt state change event.

Update(*args, **kwds)
Reloads the content of the main or given named HTML window.

Width(*args)
Returns/sets width of an HTML object window (use 'self' to address the window itself).

olx.**lcells** (package)

PACKAGE CONTENTS

FUNCTIONS olx.lcells.

Search(*args, **kwds)
d - deviation [1 A^3].
Searches current cell, cell from given file, or given cell.

Update(*args, **kwds)
Updates/creates indices using default/given configuration. To create an index, pass a folder name.

olx.**math** (package)

PACKAGE CONTENTS

FUNCTIONS olx.math.

eval(*args)
Evaluates given expression.

olx.**mouse** (package)

PACKAGE CONTENTS

FUNCTIONS olx.mouse.

Disable(*args, **kwds)
Disables one of the following operations: rotation, zooming, translation, selection

Enable(*args, **kwds)
Enables one of the following operations: rotation, zooming, translation, selection

IsEnabled(*args)
Returns current status for rotation, zooming, translation or selection.

Lock(*args, **kwds)
[Disables]/enables rotation, zooming and translation.

olx.**py** (package)

PACKAGE CONTENTS

FUNCTIONS olx.py.

Export(*args)
Exports the library to a folder.

LogLevel(*args)
Sets log level - default is macro, look at LogLevel for more information.

Reset(*args, **kwds)

???

Run(*args, **kwds)

Runs provided file.

olx.**time** (package)

PACKAGE CONTENTS

FUNCTIONS olx.time.

DF(*args)

Returns default date format.

FormatDateTime(*args)

Formats datatime using default 26 char format or using provided string. Valid formats are y(y(yy) - year like 7, 07 or 2007; M(M(M(M - month like 7, 07, Jul, July; d(d(d(d - day like 1, 01, Wed or Wednesday; h(h - for hours, m(m - minutes, s(s - seconds like 1 or 01.

Now(*args)

Returns current date and time as a long number if no format is provided. If a format string is provided it return a formatted string. The DF() function can be used for default formatting.

olx.**wbox** (package)

PACKAGE CONTENTS

FUNCTIONS olx.wbox.

GetName(*args)

Returns object collection name.

IsGrouped(*args)

Returns true if the object is in a group.

IsSelected(*args)

Returns true if the object is selected.

Type(*args)

Sets the frame type - sphere/box.

Visible(*args)

Changes/returns object visibility.

olx.**xf** (package)

PACKAGE CONTENTS

olx.xf.**analysis** (package)

PACKAGE CONTENTS

FUNCTIONS olx.xf.analysis.

AnalyseUeq(*args)

???

Scale(*args)

Scales the Q-peaks according to found fragments. Returns the scale or 0.

Trim(*args)

Trims the size of the assymetric unit according to the 18 A^3 rule. Returns true if any atoms were deleted.

olx.xf.**au** (package)

PACKAGE CONTENTS

Handles properties of the asymmetric unit.

FUNCTIONS olx.xf.au.

Fractionalise(x, y, z)

Returns fractional coordinates.

x, y, z = olx.xf.au.Fractionalise(i[2],i[3],i[4]).split(',')

GetAtomAfix(*args)

Returns atom AFIX.

GetAtomCount(*args)

Returns the atom count in the asymmetric unit.

`GetAtomCrd(*args)`
Returns a comma separated list of fractional coordinates for the specified atom.

`GetAtomName(*args)`
Returns atom label.

`GetAtomOccu(*args)`
Returns atom occupancy.

`GetAtomPart(*args)`
Returns part of the specified atom.

`GetAtomType(*args)`
Returns atom type (element).

`GetAtomU(*args)`
Returns a single number or six, comma separated values

`GetAtomUiso(*args)`
Returns a single number or six, comma separated values.

`GetAtomlabel(*args)`
The takes two arguments - the atom ID and increment. The increment is used to navigate through the periodic table, so increment +1 will return next element and -1 the previous element in the periodic table.

`GetCell(*args)`
Returns six comma separated values for a, b, c and alpha, beta gamma.

`GetCellSymm(*args)`
Returns space group of currently loaded file as name: 'C2', 'I41/amd', etc. Optionally, Hall symbol may be returned if 'hall' is provided as an argument.

`GetCellVolume(*args)`
Returns volume of the unit cell.

`GetFormula(*args)`
Returns chemical formula of the asymmetric unit

`GetPeak(*args)`
Returns peak intensity.

`GetVolume(*args)`
Returns volume of the unit cell divided by the number of symmetry elements.

`GetWeight(*args)`
Returns molecular mass of the asymmetric unit.

`GetZ(*args)`
Returns current Z.

`GetZprime(*args)`
Returns current Z divided by the number of matrices of current spacegroup.

`IsAtomDeleted(*args)`
Checks status of specified atom.

`IsPeak(*args)`
Checks if specified atom is peak.

`NPDCount(*args)`
Returns number of the NPD atoms.

`NewAtom(label, x, y, z, [True/False])`
Adds a new atom to the asymmetric unit and return its ID, by which it can be referred. The function takes the atom name and coordinates, the optional 5th parameter specifies if the position has to be tested for an existing atoms. If -1 is returned, the atom is not created.
e.g. id = olx.xf.au.NewAtom(label, x, y, z, 'false')

`Orthogonalise(*args)`
Returns orthogonalised coordinates

`SetAtomCrd(*args)`

Sets atom coordinates to specified values, first parameters is the atom "ID".

SetAtomOccu(*args)
Sets atom's occupancy; first parameter is the atom ID followed by occupancy.
olx.xf.au.SetAtomOccu(id, 1)

SetAtomPart(*args)
Sets part of the atom specified atom.

SetAtomU(*args)
Sets atoms Uiso/anis first paramater is the atom ID followed by 1 or six parameters.
olx.xf.au.SetAtomU(id, 0.04)

SetAtomlabel(*args)
Sets atom labels to provided value. The first parameter is the atom ID.

SetZ(*args)
Sets Z for the structure.

SetZprime(*args)
Sets Z' for the structure.

olx.xf.**exptl** (package)

PACKAGE CONTENTS

Handles experimental details.

FUNCTIONS olx.xf.exptl

Radiation(*args)
Returns/sets experiment wavelength in Ångström.

Size(*args)
Returns/sets crystal size. Returns/accepts strings line 0.5x0.5x0.5 in mm.

Temperature(*args)
Returns/sets experiment temperature. Returns value in C, accepts strings like 120K, 10F. Default scale is C.

olx.xf.**latt** (package)

PACKAGE CONTENTS

FUNCTIONS olx.xf.latt

GetFragmentAtoms(*args)
Returns a comma separated list of atoms in specified fragment.

GetFragmentCount(*args)
Returns number of fragments in the lattice.

GetMoiety(*args)
Returns molecular moiety.

IsGrown()
Returns true if the structure is grown.

olx.xf.**rm** (package)

PACKAGE CONTENTS

FUNCTIONS olx.xf.rm

BASF(*args)
Returns/sets BASF referred by index.

Completeness(*args)
Calculates completeness to the given 2 theta value.

Exti(*args)
Returns/sets EXTI.

FVar(*args)
Returns/sets FVAR referred by index.

HasOccu(*args)
Returns true if occupancy of any of the atoms is refined or deviates from 1.

MaxIndex(*args)

Calculates largest Miller index for the given 2 theta value.

NewAfixGroup(*args)

d-distance when applicable, sof-occupancy [11], u-default U value for atoms.

NewRestraint(*args)

s1 - standard deviation 1

s2 - standard deviation 2

Creates a new restraint expects restraint name, parameters if required and atom ids.

OSF(*args)

Returns/sets OSF

ShareADP(*args, **kwds)

Creates a rotated ADP constraint for given atoms. Currently works only for T-X3 groups (X-CMe3, X-CF3 etc) and for rings.

UpdateCR(*args)

Updates constraint or restraint parameters (name, index, {values})

olx.xf.**uc** (package)

PACKAGE CONTENTS

FUNCTIONS olx.xf.uc.

CellEx(*args)

Returns unit cell side/angle with esd

MatrixCount(*args)

Returns the number of matrices in the unit cell

VolumeEx(*args)

Returns unit cell volume with esd

FUNCTIONS olx.xf.

CurrentData(*args)

Returns current data index or changes current data block within the CIF.

DataCount(*args)

Returns number of available data sets.

DataName(*args)

Returns data name for given CIF block.

EndUpdate(*args, **kwds)

Must be called after the content of the asymmetric unit has changed

- this function will update the program state. If true is passed as an argument.

- the loader related metainformation (like LST for INS) will be cleared too.

GetFormula(arg) [none, html or list]

Returns a string for content of the asymmetric unit. Takes single or none parameters. If parameter equals 'html' and html formatted string is returned, for 'list' parameter a string like 'C:26,N:45' is returned. If no parameter is specified, just formula is returned.

GetMu(*args)

Returns absorption coefficient for current model.

RefinementInfo(*args)

Sets/returns refinement information.

SaveSolution(*args)

Saves current Q-peak model to provided file (res-file).

SetFormula(*args)

Sets formula for current file, takes a string of the following form 'C:25,N:4'.

olx.**xgrid** (package)

PACKAGE CONTENTS

olx.xgrid.**label** (package)

PACKAGE CONTENTS

FUNCTIONS olx.xgrid.label.

GetName(*args)
Returns object collection name.
IsGrouped(*args)
Returns true if the object is in a group.
IsSelected(*args)
Returns true if the object is selected.
Visible(*args)
Changes/returns object visibility.

olx.xgrid.**legend** (package)

PACKAGE CONTENTS

FUNCTIONS olx.xgrid.legend.

GetName(*args)
Returns object collection name.
IsGrouped(*args)
Returns true if the object is in a group.
IsSelected(*args)
Returns true if the object is selected.
Visible(*args)
Changes/returns object visibility.

FUNCTIONS olx.xgrid.

Contours(*args)
Returns/sets number of contour levels.
Depth(*args)
Returns/sets current depth.
Extended(*args)
Returns/sets extended size of the grid.
GetMax(*args)
Returns maximum value of the map.
GetMin(*args)
Returns minimum value of the map.
GetName(*args)
Returns object collection name.
IsGrouped(*args)
Returns true if the object is in a group.
IsSelected(*args)
Returns true if the object is selected.
IsValid(*args)
Returns true if grid data is initialised.
MaxDepth(*args)
Returns maximum available depth.
PlaneSize(*args)
Returns/sets current size.
RenderMode(*args)
Returns/sets grid rendering mode. Supported values: point, line, fill, plane, contour.
Scale(*args)
Returns/sets current scale.
Size(*args)
Returns/sets current size.
Visible(*args)
Changes/returns object visibility.

class OlexFunctions(guiFunctions.GuiFunctions)

Method resolution order:
OlexFunctions
guiFunctions.GuiFunctions
__builtin__.object

Methods defined here:

__init__(self)
AddIns(self, instruction, quiet=False)
AtReap(self, path)
BaseDir(self)
CifMerge(self, filepath, update_atoms_loop=None, report=True)
CopyVFSFile(self, copy_from, copy_to, isPersistent=0)
CurrentLanguageEncoding(self)
DataDir(self)
DelIns(self, instruction)
File(self, filename=None)
FileDrive(self, FileDrive=None)
FileExt(self, FileExt=None)
FileFull(self)
FileName(self, FileName=None)
FilePath(self, FilePath=None)
FindObject(self, variable)
FindValue(self, variable, default=u'')
GetCellVolume(self)
GetCompilationInfo(self)
GetComputername(self)
GetCrystalData(self)
GetCurrentSelection(self, calculate_connectivity=False)
GetDampingParams(self)
GetExtinction(self)
GetFVar(self, i)
GetFormula(self)
GetHtmlPanelX(self)
GetKeyname(self)
GetMacAddress(self)
GetOSF(self)
GetParam(self, variable, default=None)
GetParam_as_string(self, variable, default=None)
GetRefinementModel(self, calculate_connectivity=False)
GetSVNVersion(self)
GetTag(self)
GetUserComputerName(self)
GetUsername(self)
GetValue(self, control_name)
GuiParams(self)
HKLSrc(self, new_HKLSrc='')
HasGUI(self)
IsFileType(self, fileType)
IsPluginInstalled(self, plugin)
IsVar(self, variable)

ListFiles(self, dir_name, mask=None)
Lst(self, string)
Params(self)
Reap(self, path)
Reset(self)
SetExtinction(self, v, e=None)
SetFVar(self, i, v)
SetHtmlFontSize(self)
SetHtmlFontSizeControls(self)
SetMaxCycles(self, max_cycles)
SetMaxPeaks(self, max_peaks)
SetOSF(self, v)
SetParam(self, variable, value)
SetVar(self, variable, value)
StoreParameter(self, var='', save=False)
Translate(self, text)
TranslatePhrase(self, text)
XfAuGetValue(self, var='')
cmd(self, command)
external_edit(self, filePath)
file_ChangeExt(self, path, newExt)
func_wrap(self, f)
getCompatibleProgramName(self, name)
get_cif_item(self, key, default='', output_format=False)
get_txt_from_vfs(self, item)
htmlPanelWidth(self)
htmlUpdate(self)
makeGeneralHtmlPop(self, phil_path, htm='htm', number_of_lines=0)
olex_function(self, str)
registerCallback(self, event, function, profiling=False)
registerFunction(self, function, profiling=False, namespace='')
registerMacro(self, function, options, profiling=False)
reloadStructureAtreap(self, path, file, fader=True, sg_changed=False)
reset_file_in_OFS(self, fileName, txt=' ', copyToDisk=False)
setAllMainToolbarTabButtons(self)
set_cif_item(self, key, value)
set_refinement_program(self, program, method=None, scope='snum')
set_solution_program(self, program, method=None, scope='snum')
standardizeListOfPaths(self, list_of_paths)
standardizePath(self, path)
timer_wrap(self, f, *args, **kwds)
unregisterCallback(self, event, function, profiling=False)
unregisterFunction(self, function, profiling=False)
unregisterMacro(self, function, options, profiling=False)
write_to_olex(self, fileName, text, copyToDisk=False)

Methods inherited from guiFunctions.GuiFunctions:

Alert(self, title, text, buttons=None, tickboxText=None)

CreateBitmap(self, bitmap)
Cursor(self, state='', text='')
DeleteBitmap(self, bitmap)
GetFormulaDisplay(self)
GetHtmlPanelwidth(self)
GetUserInput(self, arg, title, contentText)
HtmlDefineControl(self, d)
HtmlLoad(self, path)
IsControl(self, ctrl_name)
Listen(self, listenFile)
Refresh(self)
SetGrad(self, f=None)
SetImage(self, zimg_name, image_file)
UpdateHtml(self, html_name='')
setDisplayQuality(self, q=None)
setItemstate(self, txt)

Data descriptors inherited from guiFunctions.GuiFunctions:

__dict__
dictionary for instance variables (if defined)

__weakref__
list of weak references to the object (if defined)

FUNCTIONS olx.

ADPDisp(*args, **kwds)
Signature: ADPDisp arguments [2]
Description: Compares two structures in the terms of atomsic dispacement after the structure optimisation and the experimental ADP. First structure is the X-Ray experimental structure and the second is the optimised one. The structures are expected to have identical labelling scheme.

ADS(*args, **kwds)
Signature: ADS arguments [any except none]
Description: Changes atom draw style [sph,elp,std]

ARad(*args, **kwds)
Signature: ARad arguments [any except none]
Description: Changes how the atoms are drawn [sfil - sphere packing, pers - static radii, isot - radii proportional to Ueq, isoth - as isot, but applied
to H atoms as well]

ASR(*args, **kwds)
Signature: ASR arguments [none] states - [Loaded file is expected]
Description: Absolute structure refinement: adds TWIN and BASF to current model in the case of non-centrosymmetric structure

ATA(*args)
Signature: ATA arguments [any] states - [Loaded file is expected]
Description: Test current structure against database. (Atom Type Assignment). Returns true if any atom type changed

AZoom(*args, **kwds)
Signature: AZoom arguments [any except none]
Description: Modifies given atoms [all] radius. The first argument is the new radius in %

Abort()
Signature: Abort arguments [none]
Description: 'abort' statement to terminate a macro execution

AddBond(*args, **kwds)
Signature: AddBond arguments [any]
Description: Adds specified bond to the connectivity table

AddIns(*args, **kwds)
Signature: AddIns arguments [any except none] states - [INS file is expected]; valid options - q;
Description: Adds an instruction to the INS file
Options:
q - quiet if has not added

```
AddLabel(*args, **kwds)
```
Signature: AddLabel arguments [3 or 5]
```
AddObject(*args, **kwds)
```
Signature: AddObject arguments [any except none or 1]
Description: Adds a new user defined object to the graphical scene
```
AddSE(*args, **kwds)
```
Signature: AddSE arguments [any except none] states - [Loaded file is expected]
Description: Tries to add a new symmetry element to current space group to form a new one. [-1] is for center of symmetry
```
Afix(*args, **kwds)
```
Alert(*args)
Signature: Alert arguments [2, 3 or 4]
Description: title message [flags YNCO-yes,no,cancel,ok XHEIQ-icon:exclamation,hand,eror,information,question R-show checkbox] [checkbox message]
```
Anis(*args, **kwds)
```
Signature: Anis arguments [any] states - [Loaded file is expected]; valid options - h;
Description: Makes provided atoms anisotropic if no arguments provided current selection or all atoms are considered
Options:
h - adds hydrogen atoms
```
AtClose(*args, **kwds)
AtomInfo(*args, **kwds)
```
Signature: AtomInfo arguments [any] states - [Loaded file is expected]
Description: Searches information for given atoms in the database
```
Atoms(*args)
```
Signature: Atoms arguments [1] states - [Loaded file is expected]
```
Atpy(*args, **kwds)
```
???
```
Atreap(*args, **kwds)
```
???
```
Atupdate(*args, **kwds)
```
???
```
BRad(*args, **kwds)
```
Signature: BRad arguments [any except none]; valid options - a;
Description: Multiplies provided [all] bonds default radius by given number. The default radius for covalent bonds is 0.1A and for H-bonds is 0.02A. To set radius for H-bonds use:
brad R hbonds
Any particula bond type can also be specified like:
brad 0.5 C-H
Note that the heavier atom type is always first
Options:
a - specified value is absolute, in A
```
Bang(*args, **kwds)
```
Signature: Bang arguments [any] states - [Loaded file is expected]; valid options - c;
Description: Prints bonds and angles table for selected/given atoms
Options:
c - copy info to the clipboard
```
BaseDir()
```
Signature: BaseDir arguments [none]
Description: Returns the startup folder
```
Basis(*args, **kwds)
```
Signature: Basis arguments [none or 1]
Description: Shows/hides the orientation basis

```
Bind(*args, **kwds)
```
Signature: Bind arguments [2]
```
CCrd(*args)
```
Signature: CCrd arguments [any] states - [Loaded file is expected]
Description: Returns center of given (selected) atoms in fractional coordinates
```
CONF(*args, **kwds)
```
Signature: CONF arguments [any] states - [Loaded file is expected]; valid options - a;
Description: Adds dihedral angle calculation instructions to create corresponding tables in the CIF
Options:
a - finds angles which made up of all given/selected atoms [true]
```
CalcCHN(*args, **kwds)
```
Signature: CalcCHN arguments [none or 1]
Description: Calculates CHN composition of current structure or for provided formula
```
CalcFourier(*args, **kwds)
```
Signature: CalcFourier arguments [none] states - [Loaded file is expected]; valid options - calc;diff;fcf;i;m;obs;r;scale;tomc;
Description: Calculates fourier map
Options:
calc - calculates calculated map
diff - calculates difference map
fcf - reads structure factors from a fcf file
i - integrates the map
m - mask the structure
obs - calculates observed map
r - resolution in Angstrems
scale - scale to use for difference maps, currently available simple(s) sum(Fo^2)/sum(Fc^2)) and regression(r)
tomc - calculates 2Fo-Fc map
```
CalcMass(*args, **kwds)
```
Signature: CalcMass arguments [none or 1]
Description: Calculates Mass spectrum of current structure or for provided formula
```
CalcPatt()
```
Signature: CalcPatt arguments [none] states - [Loaded file is expected]
Description: Calculates Patterson map
```
CalcR(*args)
```
Signature: CalcR arguments [none or 1] states - [Loaded file is expected]
Description: Calculates R1, R1 for I/sig >2 and wR2. If 'print' is provided - prints
```
CalcVars(*args, **kwds)
```
Signature: CalcVars arguments [any] states - [Loaded file is expected]
Description: Calculates previously defined variables and stores the named values in
```
CalcVoid(*args, **kwds)
```
Signature: CalcVoid arguments [none or 1] states - [Loaded file is expected]; valid options - d;i;p;r;
Description: Calculates solvent accessible void and packing parameters; optionally accepts a file with space separated values of Atom Type and radius, an entry a line
Options:
d - distance from Van der Waals surface [0]
i - invert the map for rendering
p - precise calculation
r - resolution[0.2]
```
CalcVol(*args, **kwds)
```
Signature: CalcVol arguments [none or 1]; valid options - cs;n;

Description: Calculates tetrahedron or bipyramidal shape volume for given (selected) atom

Options:

cs - do not clear the selection

n - normalises bonds before the calculation

`Capitalise(*args, **kwds)`

Signature: Capitalise arguments [any except none] states - [Loaded file is expected]

Description: Changes atom labels capitalisation for all/given/selected atoms. The first argument is the template like Aaaa

`Ceiling(*args, **kwds)`

Signature: Ceiling arguments [1]

`Cell(*args, **kwds)`

Built in macro Cell

Signature: Cell arguments [none or 1] states - [Loaded file is expected]; valid options - r;

Description: If no arguments provided inverts visibility of unit cell, otherwise sets it to the boolean value of the parameter

Options:

r - shows reciprocal cell

`Built in function Cell`

Signature: Cell arguments [1] states - [Loaded file is expected]

Description: Returns value of teh given parameter: a, b, c, alpha, beta, gamma, volume

`Cent(*args, **kwds)`

Signature: Cent arguments [any]; valid options - rings;

Description: Creates a centroid for given/selected/all atoms

Options:

rings - finds rings specified by template and add centroids for each of them. For example cent -rings=C6

`Center(*args, **kwds)`

Signature: Center arguments [any] states - [Loaded file is expected]; valid options - z;

Description: Sets the centre of rotation to given point

Options:

z - also recalculates the scene zoom

`ChangeSG(*args, **kwds)`

Signature: ChangeSG arguments [any except none] states - [Loaded file is expected]; valid options - c;

Description: [shift] SG. Changes space group of current structure, applying given shit prior (if provided) to the change of symmetry of the unit cell

Options:

c - apply cell change according to the centering change (experimental!)

`CheckMenu(*args, **kwds)`

Signature: CheckMenu arguments [1]

`CheckState(*args)`

Signature: CheckState arguments [1 or 2]

Description: Returns if true if given program state is active

`ChemDraw(*args, **kwds)`

Signature: ChemDraw arguments [any] states - [Loaded file is expected]

Description: Changes the view to show aromatic rings and double/tripple bonds.

`Chiv(*args, **kwds)`

Signature: Chiv arguments [any] states - [INS file is expected]; valid options - cs;

Description: Restrains chiral volume of atom(s) to '0' or provided value

Options:

cs - do not clear selection

`ChooseDir(*args)`

Signature: ChooseDir arguments [none, 1 or 2]

Description: Shows a dialog to pick a folder. Arguments [title=Choose directory], [default path=current directory].

ChooseElement(*args)

Signature: ChooseElement arguments [none]

ChooseFont(*args)

Signature: ChooseFont arguments [none, 1 or 2]

Description: Brings up a font dialog. If font information provided, initialises the dialog with that font; the first argument may be just 'olex2' or
'system' to enforce choosing the Olex2/System font (the font information can be provided in the second argument then)

ChooseMaterial(*args)

Signature: ChooseMaterial arguments [none or 1]

Description: Brings up a dialog to edit default or provided material

Cif(*args)

Signature: Cif arguments [1] states - [CIF file is expected]

Description: Returns instruction value (all data after the instruction). In case the instruction does not exist it return 'n/a' string

Cif2Doc(*args, **kwds)

Signature: Cif2Doc arguments [none or 1] states - [Loaded file is expected]; valid options - n;

Description: converts cif to a document

Options:

n - output file name

Cif2Tab(*args, **kwds)

Signature: Cif2Tab arguments [any] states - [Loaded file is expected]; valid options - l;n;t;

Description: creates a table from a cif

Options:

l - label option flag: 0 - as is, 1 - round brackets, 2 - subscript, 4 - superscript, this can be combined

n - output file name

t - table definition file

CifCreate(*args, **kwds)

Signature: CifCreate arguments [none] states - [Loaded file is expected]

Description: Creates cif from current file, variance-covariance matrix should be available

CifExtract(*args, **kwds)

Signature: CifExtract arguments [1 or 2] states - [Loaded file is expected]; valid options - i;

Description: extract a list of items from one cif to another

Options:

i - a custom CIF with items to extract [etc/CIF/extract.cif]

CifMerge(*args, **kwds)

Signature: CifMerge arguments [any] states - [Loaded file is expected]; valid options - f;u;

Description: Merges loaded or provided as first argument cif with other cif(s)

Options:

f - creates final CIF with embedded RES file and HKL loop

u - updates atom treatment if the asymmetric units of currently loaded file and of the

Clean(*args, **kwds)

Signature: Clean arguments [none]; valid options - aq;at;d;f;npd;

Description: Tidies up current model

Options:

aq - disables analysis of the Q-peaks based on thresholds

at - disables lonely atom types assignment to O and Cl

d - before 'blown up' atoms, a possibility to demote will be checked

f - does not run 'fuse' after the completion

npd - promotes at maximum given number of atoms a call [0]

Clear()
Signature: Clear arguments [none]
Description: Clears console buffer (text)
Collectivise(*args, **kwds)
Signature: Collectivise arguments [any]
Description: Does the opposite to the Individialise. If provided atoms are unique to the lattice a call to this function makes them uniq to the asymmetric
unit, the following call makes the uniq to the element type
Color(*args)
Signature: Color arguments [none, 1 or 2]
Compaq(*args, **kwds)
Signature: Compaq arguments [none] states - [Loaded file is expected]; valid options - a;c;m;q;
Description: Moves all atoms or fragments of the asymmetric unit as close to each other as possible. If no options provided, all fragments are assembled
around the largest one.
Options:
a - assembles broken fragments
c - similar as with no options, but considers atom-to-atom distances
m - assembles non-metallic parts of the structure first, then moves metals to the closest atom
q - moves Q-peaks to the atoms, atoms are not affected
Conn(*args, **kwds)
Signature: Conn arguments [any except none]
Description: Changes provided atom(s) connectivity (only until next connectivity modifying operation for now).
Usage: conn max_bond bonding_radius [selection/atom(s)/$type]
Usage: conn max_bond [selection/atom(s)/$type]
Usage: conn bonding_radius [selection/atom(s)/$type] - note the radius should have floating point
Constrain(*args, **kwds)
Signature: Constrain arguments [any except none] states - [Loaded file is expected]
Description: Creates a constraint
Crd(*args)
Built in function Crd
Signature: Crd arguments [any] states - [Loaded file is expected]
Description: Returns center of given (selected) atoms in cartesian coordinates
CreateBitmap(*args, **kwds)
Signature: CreateBitmap arguments [2]; valid options - r;
Description:
Options:
r -
CreateMenu(*args, **kwds)
Signature: CreateMenu arguments [1, 2 or 3]; valid options - c;m;r;s;
Description:
Options:
c -
m -
r -
s -
CreateShortcut(*args, **kwds)
Signature: CreateShortcut arguments [2]
Crs(*args)
Built in function Crs
Signature: Crs arguments [1] states - [CRS file is expected]

Description: Returns instruction value (all data after the instruction). In case the instruction does not exist it return 'n/a' string

`CurrentLanguage(*args)`

Built in function CurrentLanguage

Signature: CurrentLanguage arguments [none or 1]

Description: Returns/sets current language

`CurrentLanguageEncoding(*args)`

Built in function CurrentLanguageEncoding

Signature: CurrentLanguageEncoding arguments [none]

Description: Returns current language encoding, like: ISO8859-1

`Cursor(*args)`

Built in function Cursor

Signature: Cursor arguments [none, 1, 2 or 3]

`D2CG(*args, **kwds)`

Signature: D2CG arguments [any] states - [Loaded file is expected]; valid options - c;

Description: Calculates distacne from first atom to the unit weight-centroid formed by the rest. If the variance-covariance matrix exists m also calculates the esd.

Options:

c - copies the values to the Clipboard

`DELU(*args, **kwds)`

Signature: DELU arguments [any] states - [INS file is expected]; valid options - cs;

Description: Rigid bond constraint. If no atoms provided, all non-H atoms considered

Options:

cs - do not clear selection

`Dang(*args, **kwds)`

Signature: Dang arguments [any] states - [INS file is expected]; valid options - cs;

Description: Adds a ShelX compatible angle restraint

Options:

cs - do not clear selection

`DataDir()`

Built in function DataDir

Signature: DataDir arguments [none]

Description: Returns the location of user data

`DefineVar(*args, **kwds)`

Built in macro DefineVar

Signature: DefineVar arguments [1] states - [Loaded file is expected]

Description: Defines a variable to be clalculated with CalcVars. The argument is the variable name.

`Degen(*args, **kwds)`

Built in macro Degen

Signature: Degen arguments [any] states - [Loaded file is expected]; valid options - cs;

Description: Prints how many symmetry operators put given atom to the same site

Options:

cs - clear selection

`DelBond(*args, **kwds)`

Built in macro DelBond

Signature: DelBond arguments [any]

Description: Removes specified bond from the connectivity table

`DelIns(*args, **kwds)`

Built in macro DelIns

Signature: DelIns arguments [1] states - [INS file is expected]

Description: A number or the name (will remove all accurances) can be provided

DelObject(*args, **kwds)
Built in macro DelObject
Signature: DelObject arguments [1]
Description: Deletes graphical object by name
DeleteBitmap(*args, **kwds)
Built in macro DeleteBitmap
Signature: DeleteBitmap arguments [1]
DeleteMenu(*args, **kwds)
Built in macro DeleteMenu
Signature: DeleteMenu arguments [1]
Delta(*args, **kwds)
Built in macro Delta
Signature: Delta arguments [none or 1]
Description: Prints/sets current delta fir the covalent bonds
DeltaI(*args, **kwds)
Built in macro DeltaI
Signature: DeltaI arguments [none or 1]
Description: Prints/sets current delta for short interactions
Describe(*args, **kwds)
Built in macro Describe
Signature: Describe arguments [none] states - [Loaded file is expected]
Description: Describes current refinement in a human readable form
Dfix(*args, **kwds)
Built in macro Dfix
Signature: Dfix arguments [any]; valid options - cs;e;
Description: Restrains distances to the given value
Options:
cs - do not clear selection
e -
Dir(*args, **kwds)
Built in macro Dir
Signature: Dir arguments [none or 1]
Description: Lists current folder. A file name mask may be provided
Direction(*args, **kwds)
Built in macro Direction
Signature: Direction arguments [none]
Description: Prints current orientation of the model in factional coordinates
DisableMenu(*args, **kwds)
Built in macro DisableMenu
Signature: DisableMenu arguments [1]
EADP(*args, **kwds)
Built in macro EADP
Signature: EADP arguments [any] states - [INS file is expected]
Description: Forces EADP/Uiso of provided atoms to be constrained the same
EXYZ(*args, **kwds)
Built in macro EXYZ
Signature: EXYZ arguments [any] states - [INS file is expected]; valid options - eadp;
Description: Adds a new element to the given/selected site. Takes one selected atom and element types as any subsequent argument. Alternatively can take a few selected atoms of different type to be modelled as the type swapping disorder or a set of atoms of the same type and new element type on the command line.
Options:

eadp - does not set the equivalent ADP constraint for the shared site

`Echo(*args, **kwds)`

Built in macro Echo

Signature: Echo arguments [any]; valid options - c;m;

Description: Prints provided string, functions are evaluated before printing

Options:

c - copy printed information to clipboard

m - the printing color (info, warning, error or exceptiion)

`EditAtom(*args, **kwds)`

Built in macro EditAtom

Signature: EditAtom arguments [any] states - [Loaded file is expected]; valid options - cs;

Description: Shows information for the given atom and all of its dependents

Options:

cs - do not clear the selection

`EditIns(*args, **kwds)`

Built in macro EditIns

Signature: EditIns arguments [none] states - [INS file is expected]

`EditMaterial(*args, **kwds)`

Built in macro EditMaterial

Signature: EditMaterial arguments [1]

Description: Brings up material properties dialog for specified object

`Elevate(*args, **kwds)`

Built in macro Elevate

Signature: Elevate arguments [none or 1]

Description: Runs Olex2 in elevated/desktop mode [true]/false- only available on Windows

`EnableMenu(*args, **kwds)`

Built in macro EnableMenu

Signature: EnableMenu arguments [1]

`Env(*args)`

Built in function Env

Signature: Env arguments [1] states - [Loaded file is expected]

Description: Returns immediate atom environment

`Envi(*args, **kwds)`

Built in macro Envi

Signature: Envi arguments [none, 1 or 2]; valid options - c;cs;h;p;q;

Description: This macro prints environment of any particular atom. Default search radius is 2.7A.

Options:

c - prints just the connectivity information

cs - leaves selection unchanged

h - adds hydrogen atoms to the list

p - print out precision [2], the angle printing precision will be half of that for the distances

q - adds Q-peaks to the list

`Esd(*args, **kwds)`

Built in macro Esd

Signature: Esd arguments [any] states - [Loaded file is expected]; valid options - c;l;label;

Description: This procedure calculates possible parameters for the selection and evaluates their esd using the variance-covariance matrix coming from the ShelXL refinement with negative 'MORE' like 'MORE -1' option or from the olex2.refine

Options:

c - copies printed values to the clipboard

l - consider the list of bonds as independent

label - creates a graphics label

Exec(*args, **kwds)

Built in macro Exec

Signature: Exec arguments [any except none]; valid options - d;o;q;s;t;

Description: Executes external process

Options:

d - output dub file name

o - detached

q - do not post output to console

s - synchronise

t - a list of commands to be run when the process is terminated

Exit()

Built in macro Exit

Signature: Exit arguments [none]

Description: Exits Olex2

Export(*args, **kwds)

Built in macro Export

Signature: Export arguments [none or 1] states - [CIF file is expected]

Description: Exports reflections file and RES if present in the loaded CIF

ExportFont(*args, **kwds)

Built in macro ExportFont

Signature: ExportFont arguments [2]

Description: Exports given fonts into Olex2 portable format. At maximum two fonts a file are supported: a fixed and a proportional font. Example:

ExportFont ChooseFont()&ChooseFont test.fnt

ExportFrag()

Built in macro ExportFrag

Signature: ExportFrag arguments [none] states - [Loaded file is expected]

Description: Exports selected fragment to an external file

ExtraZoom(*args)

Built in function ExtraZoom

Signature: ExtraZoom arguments [none or 1]

Description: Sets/reads current extra zoom (default zoom correction)

FATA(*args)

Built in function FATA

Signature: FATA arguments [any] states - [Loaded file is expected]

Description: Calculates the diff Fourier map and integrates it to find artifacts around atoms. (Fourier Atom Type Analysis). Returns true if any atom type changed.

FPS()

Built in function FPS

Signature: FPS arguments [none]

Fade(*args, **kwds)

Built in macro Fade

Signature: Fade arguments [3]

FcfCreate(*args, **kwds)

Built in macro FcfCreate

Signature: FcfCreate arguments [any except none] states - [Loaded file is expected]; valid options - scale;

Description: Creates fcf from current file. Expects a number as in the shelx list number as the first argument, the second argument is the output file name filename().fcf is default

Options:

scale - [external],simple or regression

File(*args, **kwds)

Built in macro File

Signature: File arguments [none or 1] states - [Loaded file is expected]; valid options - s;

Description: Saves current model to a file. By default an ins file is saved and loaded

Options:

s - sort the main residue of the asymmetric unit

`FileDrive(*args)`

Built in function FileDrive

Signature: FileDrive arguments [none or 1]

Description: Returns file drive. If no arguments provided - of currently loaded file

`FileExt(*args)`

Built in function FileExt

Signature: FileExt arguments [none or 1]

Description: Returns file extension. If no arguments provided - of currently loaded file

`FileFull(*args)`

Built in function FileFull

Signature: FileFull arguments [none]

Description: Returns full path of currently loaded file

`FileLast(*args)`

Built in function FileLast

Signature: FileLast arguments [none or 1]

`FileName(*args)`

Built in function FileName

Signature: FileName arguments [none or 1]

Description: If no arguments provided, returns file name of currently loaded file, for one argument returns extracted file name

`FileOpen(*args)`

Built in function FileOpen

Signature: FileOpen arguments [3 or 4]

`FilePath(*args)`

Built in function FilePath

Signature: FilePath arguments [none or 1]

Description: Returns file path. If no arguments provided - of currently loaded file

`FileSave(*args)`

Built in function FileSave

Signature: FileSave arguments [3 or 4]

`FitCHN(*args, **kwds)`

Built in macro FitCHN

Signature: FitCHN arguments [any except none or 1]

Description: Fits CHN analysis for given formula and observed data given a lits of possible solvents. A mixture of up to 3 solvents only considered,

however any number of observed elements can be provided. Example: FitCHN C12H22O11 C:40.1 H:6 N:0 H2O CCl3H

`Fix(*args, **kwds)`

Built in macro Fix

Signature: Fix arguments [any except none] states - [INS file is expected]

Description: Fixes specified parameters of atoms: XYZ, Uiso, Occu

`FixHL(*args, **kwds)`

Built in macro FixHL

Signature: FixHL arguments [none] states - [Loaded file is expected]

Description: Fixes hydrogen atom labels.

`FixUnit(*args, **kwds)`

Built in macro FixUnit

Signature: FixUnit arguments [none or 1] states - [Loaded file is expected]
Description: Sets SFAc and UNIT to current content of the asymmetric unit. Takes Z', with default value of 1.

```
Flat(*args, **kwds)
```
Built in macro Flat
Signature: Flat arguments [any] states - [INS file is expected]; valid options - cs;
Description: Flat group restraint for at least 4 provided atoms
Options:
cs - do not clear selection

```
Flush(*args, **kwds)
```
Built in macro Flush
Signature: Flush arguments [none, 1 or 2]
Description:
#1 Flushes log streams
#2 An extension to 'flush log' to 'flush output' to flush console buffer into DataDir()/[output.txt] file, the file name can follow the command Built in function Flush
Signature: Flush arguments [2 or 3]
Description: Saves variables to a file. The second argument is a mask in the form '*' 'settings.*' etc, if the 3rd argument [0] is specified the variable names used as substringFrom(3rd arg); note that if the name is shorter that the value of the 3rd argument - the value is not saved.

```
FlushFS(*args, **kwds)
```
Built in macro FlushFS
Signature: FlushFS arguments [none or 1]
Description: Saves current content of the virtual file system. If no parameters is given - the global state is saved. Possible arguments: global, structure

```
Fmol(*args, **kwds)
```
Built in macro Fmol
Signature: Fmol arguments [none] states - [Loaded file is expected]
Description: Shows all fragments (as opposite to uniq)

```
Focus(*args, **kwds)
```
Built in macro Focus
Signature: Focus arguments [none]
Description: Sets input focus to the console

```
Free(*args, **kwds)
```
Built in macro Free
Signature: Free arguments [any except none] states - [INS file is expected]
Description: Frees specified parameters of atoms: XYZ, Uiso, Occu

```
Freeze(*args)
```
Built in function Freeze
Signature: Freeze arguments [none or 1]
Description: Gets/Sets display update status

```
FullScreen(*args)
```
Built in function FullScreen
Signature: FullScreen arguments [none or 1]
Description: Returns/sets full screen mode (true/false/swap)

```
Fuse(*args, **kwds)
```
Built in macro Fuse
Signature: Fuse arguments [none or 1] states - [Loaded file is expected]
Description: Re-initialises the connectivity list. If a number is provided, atoms of the same type connected by bonds shorter than the provided number are merged into one atom with center at the centroid formed by all removed atoms

```
Fvar(*args, **kwds)
```

Built in macro Fvar
Signature: Fvar arguments [any] states - [INS file is expected]
Description: Assigns/release occupancy for given atoms. Examples:
-'fvar' if nothing is selected will print current values of the variables. For a selection of even number atoms, will create a new variable and link
occupancies of the first half of the selection to occupancy the other half of the selection.
-'fvar 0' - makes occupancy of provided atoms refineable
-'fvar 1' - fixes occupancy of provided atoms at current value
-'fvar 1 1' - fixes occupancy of provided atoms at chemical occupancy of 1
-'fvar 2' will link occupancy of the given atoms to the value of the 2nd FVAR multiplied by current value of the occupancy of the given atoms, or, if occupancy already linked to a variable - it will replace the variable index.
-'fvar 2 0.5' will link occupancy of the given atoms to the value of the 2nd FVAR multiplied by 0.5.

`GenDisp(*args, **kwds)`
Built in macro GenDisp
Signature: GenDisp arguments [none or 1] states - [Loaded file is expected]; valid options - f;n;
Description: Generates anisotropic dispertion parameters for current radiation wavelength
Options:
f - generates full SFAC instructions
n - for neutron data

`GetCompilationInfo(*args)`
Built in function GetCompilationInfo
Signature: GetCompilationInfo arguments [any]
Description: Returns compilation info

`GetEnv(*args)`
Built in function GetEnv
Signature: GetEnv arguments [none or 1]
Description: Prints all variables if no arguments is given or returns the given veariable value

`GetFont(*args)`
Built in function GetFont
Signature: GetFont arguments [1]
Description: Returns specified font

`GetMAC(*args)`
Built in function GetMAC
Signature: GetMAC arguments [none or 1]
Description: Returns simicolon separated list of computer MAC addresses. If 'full' is provided as argument, the adoptor names are also returned as
adapter=MAC;..

`GetMaterial(*args)`
Built in function GetMaterial
Signature: GetMaterial arguments [1 or 2]
Description:
#1 Returns material of specified object
#2 Returns material of specified object. Special materials are: helpcmd, helptxt, execout, error, exception

`GetMouseX(*args)`
Built in function GetMouseX
Signature: GetMouseX arguments [none]
Description: Returns current mouse X position

`GetMouseY(*args)`
Built in function GetMouseY
Signature: GetMouseY arguments [none]
Description: Returns current mouse Y position

```
GetUserInput(*args)
```
Built in function GetUserInput

Signature: GetUserInput arguments [3]

```
GetVar(*args)
```
Built in function GetVar

Signature: GetVar arguments [1 or 2]

Description: Gets the value of the specified variable. If the variable does not exist and no default value is provided - an error occurs

```
GetWindowSize(*args)
```
Built in function GetWindowSize

Signature: GetWindowSize arguments [none, 1 or 3]

Description: Returns size of the requested window, main window by default

```
GlTooltip(*args)
```
Built in function GlTooltip

Signature: GlTooltip arguments [none or 1]

Description: Returns state of/sets OpenGL tooltip implementation for the main window (some old platforms do not have proper implementation of tooltips)

```
Grad(*args, **kwds)
Built in macro Grad
```
Signature: Grad arguments [none, 1 or 4]; valid options - i;p;

Description: Sets options for the background gradient. No options - showsn the gradient dialog where the user can choose the gradient colors. One
parameter is expected to be a boolean - shows/hides the gradient. Four parameters specify the gradient colours explicetly.

Options:

i - toggles gradient mode and the user/white background

p - sets/removes the gradient picture, the picture is assumed to have power of 2 dimensions (like 512x256), it is stretched if needed

```
GraphPD(*args, **kwds)
```
Built in macro GraphPD

Signature: GraphPD arguments [none] states - [Loaded file is expected]; valid options - fcf;r;s;

Description: Prints a intensity vs. 2 theta graph

Options:

fcf - take structure factors from the FCF file, otherwise calculate from current model

r - resolution in degrees [0.5]

s - use simple scale when calculating structure factors from the mode, otherwise regression scaling will be used

```
GraphSR(*args, **kwds)
```
Built in macro GraphSR

Signature: GraphSR arguments [none or 1] states - [Loaded file is expected]; valid options - b;

Description: Prints a scale vs resolution graph for current file (fcf file must exist in current folder)

Options:

b - number of bins

```
Group(*args, **kwds)
```
Built in macro Group

Signature: Group arguments [none or 1] states - [Loaded file is expected]; valid options - n;u;

Description: Groups current visible objects or selection

Options:

n - a custom name can be provided

u - ungroups given group

```
Grow(*args, **kwds)
```
Built in macro Grow

Signature: Grow arguments [any] states - [Loaded file is expected]; valid options - b;s;t;w;
Description:
#1 Grows whole structure or provided atoms only
#2 Options:
b - grows all visible grow bonds (when in a grow mode)
s - grow shells vs fragments
t - grows only provided atoms/atom types
w - grows the rest of the structure, using already applied generators

HAdd(*args, **kwds)

Built in macro HAdd
Signature: HAdd arguments [any]; valid options - r;
Description: Adds hydrogen atoms to all or provided atoms, however the ring atoms are treated separately and added all the time
Options:
r - use restraints vs constraints for water molecules [False]

HImp(*args, **kwds)

Built in macro HImp
Signature: HImp arguments [any except none] states - [Loaded file is expected]
Description: Increases, decreases length of H-bonds. Arguments: value [H atoms]. Value might be +/- to specify to increase/decrease current value

HKLF(*args)

Built in function HKLF
Signature: HKLF arguments [any] states - [Loaded file is expected]
Description: If no arguments given - returns current HKLF value, otherwise if the given value 0 - sets HKLF to 4 else - sets HKLF to 5 and adds the given
number of BASF parameters

HKLSrc(*args)

Built in function HKLSrc
Signature: HKLSrc arguments [none or 1] states - [Loaded file is expected]
Description: Returns/sets hkl source for currently loaded file

HasGUI(*args)

Built in function HasGUI
Signature: HasGUI arguments [none]
Description: Returns if true if Olex2 is built with GUI

Help(*args, **kwds)

Built in macro Help
Signature: Help arguments [any]; valid options - c;
Description: Prints available information. If no arguments provided prints available commands
Options:
c - specifies commands category

Hide(*args, **kwds)

Built in macro Hide
Signature: Hide arguments [any]; valid options - b;
Description: Hides selected objects or provided atom names (no atom related objects as bonds are hidden automatically)
Options:
b - also hides all bonds atatched to the selected atoms

HklAppend(*args, **kwds)

Built in macro HklAppend
Signature: HklAppend arguments [any]; valid options - c;h;k;l;
Description: moves reflection back into the refinement list. See excludeHkl for more details
Options:

c -
h -
k -
l -

HklBrush(*args, **kwds)

Built in macro HklBrush

Signature: HklBrush arguments [any]; valid options - f;

Description: for high redundancy data sets, removes equivalents with high sigma

Options:

f - consider Friedel law

HklEdit(*args, **kwds)

Built in macro HklEdit

Signature: HklEdit arguments [none, 1 or 3]

HklExclude(*args, **kwds)

Built in macro HklExclude

Signature: HklExclude arguments [any]; valid options - c;h;k;l;

Description: Excludes reflections with give indexes from the hkl file -h=1;2 : all reflections where h=1 or 2

Options:

c - true/false to use provided indexes in any reflection. The default is in any one reflection

h - semicolon separated list of indexes

k -

l -

HklExtract(*args, **kwds)

Built in macro HklExtract

Signature: HklExtract arguments [1] states - [Loaded file is expected]

HklImport(*args, **kwds)

Built in macro HklImport

Signature: HklImport arguments [any except none, 1, 2 or 3]; valid options - batch;

Description: Creates a Shelx compatible 44488(4) file format from given source. Valid arguments: fixed and separator. For example:

'HklImport in.hkl fixed 7 7 7 9 9 out.hkl' or 'HklImport in.hkl separator ' '' out.hkl

Options:

batch - for separator formatted file specifies that there is a batch number

HklMerge(*args, **kwds)

Built in macro HklMerge

Signature: HklMerge arguments [any] states - [Loaded file is expected]; valid options - z;

Description: Merges current HKL file (ehco HKLSrc()) to given file name. Warning: if no arguments provided, the current file is overwritten

Options:

z - zero negative intensity

HklSplit(*args, **kwds)

Built in macro HklSplit

Signature: HklSplit arguments [2]; valid options - b;

Description: Split an HKL file according to the Fc^2/esd or the value of |Fc^2-Fo^2|/esd. The threshold value is the first argument. If it ends with '%' - the pecentage of the merged reflections is taken into the account. The second argument is the splitting criterion - 'i' for intensity or 'a' for the agreeability. Unless -b option is provided, the 'agreeable' reflections end up in the *_a.hkl file, 'disagreeable' - in the *_d.hkl file; weaker reflections end up in *_w.hkl and stronger reflections - in the *_s.hkl files.

Options:

b - creates an HKLF 5 file (*_h5) with batches 1 and -2; only applicable when second parameter is 'a'

HklStat(*args, **kwds)

Built in macro HklStat

Signature: HklStat arguments [any] states - [Loaded file is expected]; valid options - l;m;
Description: If no arguments provided, prints the statistics on all reflections as well as the ones used in the refinement. If an expressions (condition) is given in the following form: x[ahbkcl], meaning that x=ah+bk+cl for x equals 0, ((ah+bk+cl) mod x) equals 0 for positove x and not equals 0 for negative x; the subsequent expressions are combined using logical 'and' operator. For instance 2[l] expression means: to find all reflections where l is even, the expression -2[l] means to find all reflections with odd l, 0[h-l] - reflections where h equals to l etc. The function operates on all P1 merged reflections after filtering by SHEL and OMIT, -m option merges the reflections in current space group
Options:
l - list the reflections
m - merge reflection in current space group

`HklView(*args, **kwds)`
Built in macro HklView
Signature: HklView arguments [none or 1]

`Htab(*args, **kwds)`
Built in macro Htab
Signature: Htab arguments [none, 1 or 2] states - [INS file is expected]; valid options - g;t;
Description: Adds HTAB instructions to the ins file, maximum bond length [2.9] and minimal angle [150] might be provided
Options:
g - generates found interactions
t - adds extra elements (comma separated -t=Se,I) to the donor list. Defaults are [N,O,F,Cl,S,Br]

`HtmlPanelSwap(*args, **kwds)`
Built in macro HtmlPanelSwap
Signature: HtmlPanelSwap arguments [none or 1]
Description: Swaps or sets the position of the HTML GUI panel. If no arguments given - swaps the panel position. The position can be specified by left and right

`HtmlPanelVisible(*args, **kwds)`
Built in macro HtmlPanelVisible
Signature: HtmlPanelVisible arguments [none, 1 or 2]
Description: Swaps visibility of the HTML GUI panel or sets it to a given state (true/false)

`HtmlPanelWidth(*args, **kwds)`
Built in macro HtmlPanelWidth
Signature: HtmlPanelWidth arguments [none or 1]

`ISOR(*args, **kwds)`
Built in macro ISOR
Signature: ISOR arguments [any] states - [INS file is expected]; valid options - cs;
Description: Forses Uij of provided atoms to behave in isotropic manner. If no atoms provided, all non-H atoms considered
Options:
cs - do not clear selection

`IT(*args, **kwds)`
Built in macro IT
Signature: IT arguments [any]; valid options - o;
Description: Calculates tensor of inertia
Options:
o - orients basis according to principle axes of inertia

`ImportFont(*args, **kwds)`
Built in macro ImportFont
Signature: ImportFont arguments [2]

`ImportFrag(*args, **kwds)`
Built in macro ImportFrag
Signature: ImportFrag arguments [none or 1] states - [Loaded file is expected]; valid options - a;c;d;o;p;

Description: Import a fragment into current structure
Options:
a - set specified AFIX to the imported fragment
c - take the content from the clipboard
d - generate DFIX for 1-2 and 1-3 distances
o - set specified occupancy to the imported fragment atoms
p - part to assign

`Individualise(*args, **kwds)`

Built in macro Individualise
Signature: Individualise arguments [any]
Description: Moves provided atoms to individual collections, so that the atom properties, such as draw style and appearance can be changed separately of the group. The first call to this macro creates a group unique to the asymmetric unit, the second call makes the atom unique to the lattice

`Info(*args, **kwds)`

Built in macro Info
Signature: Info arguments [any]; valid options - c;f;p;s;
Description: Prints out information for gived [all] atoms
Options:
c - copy the printed information ito the clipboard
f - print fractional coordinates vs Cartesian [true]
p - coordinate precision [3]
s - sorts the atom list

`Ins(*args)`

Built in function Ins
Signature: Ins arguments [1] states - [INS file is expected]
Description: Returns instruction value (all data after the instruction). In case the instruction does not exist it return 'n/a' string

`InstallPlugin(*args, **kwds)`

Built in macro InstallPlugin
Signature: InstallPlugin arguments [1]; valid options - "l;
Description:
Options:
l - local installation from a zip file, which must contains index.ind.

`Inv(*args, **kwds)`

Built in macro Inv
Signature: Inv arguments [any] states - [Loaded file is expected]; valid options - f;
Description: Inverts whole structure or provided fragments of the structure
Options:
f - force inversion for non-centrosymmetric space groups

`IsCurrentLanguage(*args)`

Built in function IsCurrentLanguage
Signature: IsCurrentLanguage arguments [1]
Description: Checks current language

`IsFileLoaded(*args)`

Built in function IsFileLoaded
Signature: IsFileLoaded arguments [none]
Description: Returns true/false

`IsFileType(*args)`

Built in function IsFileType
Signature: IsFileType arguments [1]
Description: Checks type of currently loaded file [ins,res,ires,cif,cmf,mol,xyz]

`IsOS(*args)`

Built in function IsOS

Signature: IsOS arguments [1]

Description: Returns true if current system Windows [win], Linux/GTK [linux], Mac [mac]

`IsPluginInstalled(*args)`

Built in function IsPluginInstalled

Signature: IsPluginInstalled arguments [1]

`IsVar(*args)`

Built in function IsVar

Signature: IsVar arguments [1]

Description: Checks if the specified variable exists

`Isot(*args, **kwds)`

Built in macro Isot

Signature: Isot arguments [any] states - [Loaded file is expected]; valid options - npd;

Description: Makes provided atoms isotropic, if no arguments provided, current selection or all atoms become isotropic

Options:

npd - makes all NPD atoms isotropic

`Kill(*args, **kwds)`

Built in macro Kill

Signature: Kill arguments [any]; valid options - au;

Description: Deletes provided/selected atoms, bonds and other objects. 'kill labels' hides all atom and bond labels

Options:

au - kill atoms in the asymmetric unit (disregarding visibility/availability). This option is intended for the internal use - the model is not rebuilt after its execution.

`LS(*args, **kwds)`

Built in macro LS

Signature: LS arguments [1 or 2] states - [INS file is expected]

Description: Sets refinement method and/or the number of iterations.

`LSM(*args)`

Built in function LSM

Signature: LSM arguments [none or 1] states - [INS file is expected]

Description: Return/sets current refinement method, L.S. or CGLS currently. The method can also be set using

`Label(*args, **kwds)`

Built in macro Label

Signature: Label arguments [any]; valid options - cif;symm;type;

Description: Creates moveable labels for provided atoms/bonds/angles (selection)

Options:

cif - creates labels for CIF data a combination of {b,a,t,h}

symm - symmetry dependent tag type {[$], #, full}

type - type of labels to make - subscript, brackers, default

`Labels(*args, **kwds)`

Built in macro Labels

Signature: Labels arguments [any]; valid options - a;ao;b;co;f;h;i;l;o;p;qi;r;u;v;

Description: Shows/hides atom labels. Takes no argument is given to invert current labels visibility or a boolean value

Options:

a - afix

ao - actual occupancy (as in the ins file)

b - bond lengths

co - chemical occupancy

f - fixed parameters

h - show hydrogen atom labels
i - display labels for identity atoms only
l - label
o - occupancy
p - part
qi - Q peak intensity
r - Uiso multiplier for riding atoms
u - Uiso
v - variables

LastError(*args)

Built in function LastError
Signature: LastError arguments [none]
Description: Returns last error

Line(*args, **kwds)

Built in macro Line
Signature: Line arguments [any]; valid options - f;n;
Description: Creates a line or best line for provided atoms
Options:
f - consider input in fractional coordinates vs Cartesian
n - just sets current view normal to the line without creating the object

Lines(*args, **kwds)

Built in macro Lines
Signature: Lines arguments [1]
Description: Sets the number of visible text lines in the console. Use -1 to display all lines Link link_parts_occupancies_with

Link(*args, **kwds)

Built in macro Link
Signature: Link arguments [none or 1]

Listen(*args, **kwds)

Built in macro Listen
Signature: Listen arguments [any]
Description: Listens for changes in a file provided as argument. If the file content changes it is automatically reloaded in Olex2. If no arguments provided prints current status of the mode

Load(*args, **kwds)

Built in macro Load
Signature: Load arguments [any except none]; valid options - c;
Description:
#1 Arguments - textures
#2 Loads style/scene/view/gview/model/radii. For radii accepts sfil, vdw, pers
Options:
c - when loading style clears current model customisation [false]

LoadDll(*args)

Built in function LoadDll
Signature: LoadDll arguments [1]

LogLevel(*args)

Built in function LogLevel
Signature: LogLevel arguments [none or 1]
Description: Returns/sets log level, default is 'm' - for macro, accepts/returns 'm', 'mf' or 'f'
LS LstFS LstFun LstGO LstIns LstMac LstSymm LstVar LSM Lst
LstFS LstFun LstGO LstIns LstMac LstSymm LstVar Lst

Lst(*args)

Built in function Lst

Signature: Lst arguments [1] states - [INS file is expected]

Description: returns a value from the Lst file

`LstFS(*args, **kwds)`

Built in macro LstFS

Signature: LstFS arguments [any]

Description: Prints out detailed content of virtual file system. Accepts * based masks

`LstFun(*args, **kwds)`

Built in macro LstFun

Signature: LstFun arguments [any]; valid options - h;

Description: Lists all defined functions. Accepts * based masks

Options:

h - Shows help

`LstGO(*args, **kwds)`

Built in macro LstGO

Signature: LstGO arguments [none]

Description: List current graphical objects

`LstIns(*args, **kwds)`

Built in macro LstIns

Signature: LstIns arguments [none] states - [INS file is expected]

Description: Lists all instructions of currently loaded Ins file

`LstMac(*args, **kwds)`

Built in macro LstMac

Signature: LstMac arguments [any]; valid options - h;

Description: Lists all defined macros. Accepts * based masks

Options:

h - Shows help

`LstSymm(*args, **kwds)`

Built in macro LstSymm

Signature: LstSymm arguments [none] states - [Loaded file is expected]

Description: Prints symmetry codes for current structure

`LstVar(*args, **kwds)`

Built in macro LstVar

Signature: LstVar arguments [any]

Description: Lists all defined variables. Accepts * based masks

`Mask(*args, **kwds)`

Built in macro Mask

Signature: Mask arguments [any except none]

Description: Sets primitives for atoms or bonds according to provided mask. Accepts atoms, bonds, hbonds or a name (like from LstGO).

Example: 'mask hbonds 2048' - this resets hydrogen bond style to default.

`Match(*args, **kwds)`

Built in macro Match

Signature: Match arguments [none, 1 or 2]; valid options - a;cm;esd;h;i;n;o;s;u;w;

Description: Fragment matching, alignment and label transfer routine

Options:

a - align

cm - copies the transformation matrix suitable for sgen to clipboard

esd - calculate esd (works for pairs only)

h - excludes H atoms from matching and the RMSD calculation

i - try inversion

n - naming. If the value a symbol [or set of] this is appended to the label, '$xx' replaces the symbols after the atom type symbol with xx, leaving the ending, '-xx' - changes the ending of the label with xx

o - matches overlayed lattices
s - subgraph match
u - unmatch
w - use non-unit weights [ZO - atomic number X occupancy], Z - atomic number, EM - atomic mass, AM - atomic mass X occupancy, O - occupancy

`MatchFiles(*args)`

Built in function MatchFiles
Signature: MatchFiles arguments [2 or 3]
Description: Matches given files

`Matr(*args, **kwds)`

Built in macro Matr
Signature: Matr arguments [none, 1, 2, 3 or 9]; valid options - r;
Description: Displays or sets current orientation matrix. For single argument, 1,2,3 001, 111, etc values are acceptable, two values taken are of the klm
form, which specify a view from k1*a+l1*b+m1*c to k2*a+l2*b+m2*c, three values pecify the view normal and nine values provide a full matrix
Options:
r - used reciprocal cell instead
Help Mode

`Mode(*args, **kwds)`

Built in macro Mode
Signature: Mode arguments [any except none] states - [Loaded file is expected]; valid options - a;c;l;p;r;s;shells;t;v;
Description: Turns specified mode on. Valid mode: fixu, fixc, grow, himp, match, move, name, occu, pack, part, split, fit
Options:
a - [name] autocomplete; [grow] grow (rebuild) asymmetric unit only; [fit] afix
c - [grow] covalent bonds; [move] copy fragments instead of moving
l - [name] lock atom types after naming
p - [name] prefix; [grow] inserts the new atoms into the AU with given [-1] part value
r - [split] a restraint/constraint for split atoms; [grow] show radial bonds between the same atoms; [fit] rotation angle increment (smooth rotation by default); [name] synchronise names in the residues
s - [grow] short interactions; [name] suffix; [fit] split, atoms to split offset [0]
shells - [grow] grow atom shells vs fragments
t - [name] type
v - [grow] use user provided delta for connectivity analysis, default 2A

`MolInfo(*args, **kwds)`

Built in macro MolInfo
Signature: MolInfo arguments [any] states - [Loaded file is expected]; valid options - g;o;s;
Description: Prints molecular volume, surface area and other information for visible/selected atoms
Options:
g - generation of the triangluation [5]
o - use occupancy of the atoms in the integration
s - source ([o]ctahedron, (t)etrahedron)

`Move(*args, **kwds)`

Built in macro Move
Signature: Move arguments [none or 2]; valid options - c;cs;
Description: Moves two atoms as close to each other as possible; if no atoms given, moves all fragments as close to the cell center as possible
Options:
c - copy moved atom

cs - leaves selection unchanged

Mpln(*args, **kwds)

Built in macro Mpln

Signature: Mpln arguments [any]; valid options - n;r;rings;we;

Description: Sets current view along the normal of the best plane

Options:

n - just orient, do not create plane

r - create regular plane

rings - creates planes for rings template, like NC5

we - use weights proportional to the (atomic mass)^we

Name(*args, **kwds)

Built in macro Name

Signature: Name arguments [none, 1 or 2]; valid options - c;cs;r;s;

Description: Names atoms. If the 'sel' keyword is used and a number (or just the number) is provided as second argument the numbering will happen in the order the atoms were selected

Options:

c - enables checking labels for duplications

cs - leaves current selection unchanged

r - synchronise names in the residues

s - simply changes suffix of provided atoms to the provided one (or none)

NextSolution(*args, **kwds)

Built in macro NextSolution

Signature: NextSolution arguments [none]

OFileDel(*args, **kwds)

Built in macro OFileDel

Signature: OFileDel arguments [1]

Description: Deletes overlayed file specified by index

OFileSwap(*args, **kwds)

Built in macro OFileSwap

Signature: OFileSwap arguments [none or 1]

Description: Makes overlayed file, given by index the current file to which all commands are applied

Omit(*args, **kwds)

Built in macro Omit

Signature: Omit arguments [1, 2 or 3] states - [INS file is expected]

Description: Removes any particular reflection from the refinement list. If a single number is provided, all reflections with delta(F^2)/esd greater than given number are omitted

OnRefine(*args, **kwds)

Built in macro OnRefine

Signature: OnRefine arguments [any]

Description: Internal procedure

P4p(*args)

Built in function P4p

Signature: P4p arguments [1] states - [P4P file is expected]

Description: Returns instruction value (all data after the instruction). In case the instruction does not exist it return 'n/a' string

Pack(*args, **kwds)

Built in macro Pack

Signature: Pack arguments [any] states - [Loaded file is expected]; valid options - c;

Description:
#1 Packs structure within default or given volume(6 or 2 values for parallelepiped or 1 for sphere). If atom names/types are provided it only packs the provided atoms.
#2 Extends the default macro by keyword 'wbox'
Options:
c - specifies if current lattice content should not be deleted

```
Part(*args, **kwds)
```

Built in macro Part
Signature: Part arguments [any] states - [Loaded file is expected]; valid options - c;lo;p;
Description: Sets part(s) to given atoms, also if -lo is given and -p > 1 allows linking occupancy of given atoms throw FVAR and/or SUMP in cases when –p > 2
Options:
c - creates a copy of all grown atoms to which applied in the asymmetric unit and automatically links occupancies with the original atoms
lo - link ocupancy of given atoms through FVAR's
p - number of parts

```
Patt(*args, **kwds)
```

Built in macro Patt
Signature: Patt arguments [none] states - [INS file is expected]
Description:

```
PiM(*args, **kwds)
```

Built in macro PiM
Signature: PiM arguments [any] states - [Loaded file is expected]; valid options - l;
Description: Creates an illustration of a pi-system to metal bonds
Options:
l - display labels for the created lines

```
PiPi(*args, **kwds)
```

Built in macro PiPi
Signature: PiPi arguments [none or 2] states - [Loaded file is expected]; valid options - g;r;
Description: Analysis of the pi-pi interactions. The procedure searches for flat requlars C6 or NC5 rings and prints information for the ones where the
centroid-centroid distance is smaller than [4] A and the shift is smaller than [3] A. These two parameters can be customised.
Options:
g - generates using found symmetry operations
r - ring content [C6,NC5]

```
PiSig(*args, **kwds)
```

Built in macro PiSig
Signature: PiSig arguments [none or 2] states - [Loaded file is expected]; valid options - g;r;
Description: Analysis of the pi-sigma interactions (experimental). The procedure searches for flat requlars 4 and 6 membored rings. Then it searches for
atoms within [4] A from the ring centre and with less than [25] angle between the plane normal and the (plane center-atom) vectors.
Options:
g - generates using found symmetry operations
r - ring content [C6,NC5]

```
Pict(*args, **kwds)
```

Built in macro Pict
Signature: Pict arguments [1 or 2]; valid options - bw;c;dpi;nbg;pq;
Description: Outputs a raster picture. Output file name is required, if a second numerical parameter is provided, it is considered to be image resolution

in range [0.1-10] in screen sizes, anythin greater than 10 is considered as the desired picture width.
Options:
bw - embossed output in b&w
c - embossed output in color
dpi - the physical resolution to be written to the file
nbg - mask the background with 0 alpha (an rgb colour may be gives)
pq - highest (picture) quality

```
PictPR(*args, **kwds)
```

Built in macro PictPR
Signature: PictPR arguments [1] states - [Loaded file is expected]
Description: PovRay output

```
PictPS(*args, **kwds)
```

Built in macro PictPS
Signature: PictPS arguments [1] states - [Loaded file is expected]; valid options - atom_outline_color;atom_outline_oversize;bond_outline_color;bond_outline_oversize;color_bond;color_fill;color_line;div_ellipse;div_pie;lw_ellipse;lw_font;lw_octant;lw_pie;multiple_bond_width;octants;p;scale_hb;stipple_disorder;
Description: Postscript rendering
Options:
atom_outline_color - atom outline color[0xffffff]
atom_outline_oversize - the extra size of outline in percents[5]
bond_outline_color - bond outline color[0xffffff]
bond_outline_oversize - the extra size of the outline in percents [10]
color_bond - bonds are colored
color_fill - ellipses are filled
color_line - lines
div_ellipse - the number of ellipse segments [36]
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
multiple_bond_width - if 0 double and triple bonds are rendered as a fraction of their real width. If a values is not 0 - it specifies the width of the strips
octants - comma separated atom types/names ADP's of which to be rendered with octants [-$C]
p - perspective
scale_hb - scale for H-bonds [0.5]
stipple_disorder - render stippled bonds for atoms in non 0 part [true]

```
PictS(*args, **kwds)
```

Built in macro PictS
Signature: PictS arguments [1 or 2] states - [Loaded file is expected]; valid options - a;h;s;
Description: Stereoscopic picture rendering
Options:
a - view angle [6]
h - output image height [screen*resolution]
s - separation between the images in % [10]

```
PictTEX(*args, **kwds)
```

Built in macro PictTEX
Signature: PictTEX arguments [1] states - [Loaded file is expected]; valid options - color_fill;color_line;
Description: Experimental tex/pgf rendering

Options:
color_fill - ellipses are filled
color_line - lines

Picta(*args, **kwds)
Built in macro Picta
Signature: Picta arguments [1 or 2]; valid options - dpi;nbg;pq;
Description: A portable version of pict with limited resolution (OS/graphics card dependent). Not stable on some graphics cards
Options:
dpi - the physical resolution to be written to the file
nbg - mask the background with 0 alpha (an rgb colour may be gives)
pq - picture quality

Plan(*args, **kwds)
Built in macro Plan
Signature: Plan arguments [1] states - [INS file is expected]
Description: Sets the number of Fourier peaks to be found from the difference map

Poly(*args, **kwds)
Built in macro Poly
Signature: Poly arguments [any except none] states - [Loaded file is expected]
Description: Sets polyhedra type for all/selected/given atoms. Last argument specifies the type - none, auto, regular

Popup(*args, **kwds)
Built in macro Popup
Signature: Popup arguments [2]; valid options - b;h;ondblclick;onsize;s;t;w;x;y;
Description: Creates a popup HTML window. Usage: popup popup_name html_source
Options:
b - border[trscaip],t-caption bar, r-sizeable border, s-system menu, c-close box, a-maximise box, i-minimise box, p-window should stay on the top of
others
h - height
ondblclick - a macro or commands to execute when window is double clicked
onsize - a macro to be executed when the popup is resized
s - do show the window after the creation
t - title
w - width
x - left position
y - top position

Post(*args, **kwds)
Built in macro Post
Signature: Post arguments [any]
Description: Prints a string, but only after a new line character is encountered

Process(*args)
Built in function Process
Signature: Process arguments [1]
Description: Processes a function passed as the argument and returns the result

ProcessCmd(*args, **kwds)
Built in macro ProcessCmd
Signature: ProcessCmd arguments [any except none or 1]
Description: Send a command to current process. 'nl' is translated to the new line char and 'sp' to the white space char

ProjSph(*args, **kwds)
Built in macro ProjSph

Signature: ProjSph arguments [any] states - [Loaded file is expected]; valid options - a;e;g;group;
Description: Creates a projection from the selected atom onto a sphere, coloring each point on the sphere with a unique color corresponding to fragments.
For referece see Guzei, I.A., Wendt, M.Dalton Trans., 2006, 3991−3999.
Options:
a - transparency level [0x9c] 0 - 255
e - emboss the sphere
g - sphere quality [6]
group - group the ligands into same-colour groups [false]

`Push(*args, **kwds)`
Built in macro Push
Signature: Push arguments [any except none, 1 or 2] states - [Loaded file is expected]
Description: Shifts the sctructure (or provided fragments) by the provided translation

`QPeakScale(*args, **kwds)`
Built in macro QPeakScale
Signature: QPeakScale arguments [none or 1]
Description: Prints/sets the scale of dependency of the Q-peak transparency vs height.

`QPeakSizeScale(*args, **kwds)`
Built in macro QPeakSizeScale
Signature: QPeakSizeScale arguments [none or 1]
Description: Prints/sets the scale the Q-peak size relative to other atoms, default is 1

`Qual(*args, **kwds)`
Built in macro Qual
Signature: Qual arguments [1]
Description: Sets drawings quality, 1 - low, 2 - medium, 3 - high

`RESI(*args, **kwds)`
Built in macro RESI
Signature: RESI arguments [any except none] states - [Loaded file is expected]; valid options - a;
Description: Creates residue with given class name and optionally number and adds selected or provided atoms into the residue. If provided residue class
name is 'none', provided atoms are removed from their residues
Options:
a - alias

`RGB(*args)`
Built in function RGB
Signature: RGB arguments [3 or 4]
Description:

`RIGU(*args, **kwds)`
Built in macro RIGU
Signature: RIGU arguments [any] states - [Loaded file is expected]
Description: Creates rigid bond (RIGU) restraint for a group of provided atoms (or selection)

`RRings(*args, **kwds)`
Built in macro RRings
Signature: RRings arguments [any]; valid options - cs;
Description: Makes all provided rings [like C6 or NC5] regular (flat and all distances similar). If a selection is given - the whole rings must be selected
Options:
cs - do not clear selection

`RSA(*args, **kwds)`
Built in macro RSA
Signature: RSA arguments [any] states - [Loaded file is expected]
Description: Identifies chiral centres and prints R/S their stereo configuration

RTab(*args, **kwds)
Built in macro RTab
Signature: RTab arguments [any except none] states - [INS file is expected]
Description: Adds RTAB with given name (first argument) for provided atoms/selection
Refresh(*args, **kwds)
Built in macro Refresh
Signature: Refresh arguments [none]
Description: Refreshes the GUI
Reload(*args, **kwds)
Built in macro Reload
Signature: Reload arguments [1]
Description:
RemoveSE(*args)
Built in function RemoveSE
Signature: RemoveSE arguments [1] states - [Loaded file is expected]
Description: Returns a new space group name without provided element
Reset(*args, **kwds)
Built in macro Reset
Signature: Reset arguments [any] states - [Loaded file is expected]; valid options - atoms;c;f;rem;s;
Description: Resets current structure for the solution with ShelX
Options:
atoms - saves the atom list alongside
c - content
f - alternative file name
rem - exclude remarks
s - space group
Restart()
Built in macro Restart
Signature: Restart arguments [none]
Description: Restarts Olex2
Restrain(*args, **kwds)
Built in macro Restrain
Signature: Restrain arguments [any] states - [Loaded file is expected]
Description: Creates a restraint
Rota(*args, **kwds)
Built in macro Rota
Signature: Rota arguments [2 or 5]
Description: For two arguments the first one specifies axis of rotation (1,2,3 or x,y,z) and the second one the rotation angle in degrees. For five
arguments the first three arguments specify the rotation vector [x,y,z] the forth parameter is the rotation angle and the fifth one is the increment - the
rotation will be continuous
Run(*args, **kwds)
Built in macro Run
Signature: Run arguments [any except none]
Description: Runs provided macros (combined by '>>')
SAInfo(*args, **kwds)
Built in macro SAInfo
Signature: SAInfo arguments [any]
Description: Finds and prints space groups which include any of the provided systematic absences in the form 'b~~', '~b~' or '~~b'
SG(*args, **kwds)

Built in macro SG
Signature: SG arguments [none or 1]; valid options - a;
Description: suggest space group
Options:
a -

SGE(*args, **kwds)
Built in macro SGE
Signature: SGE arguments [none or 1] states - [Loaded file is expected]
Description: Extended spacegroup determination. Internal use

SGInfo(*args, **kwds)
Built in macro SGInfo
Signature: SGInfo arguments [none or 1]; valid options - c;i;
Description: Prints space group information.
Options:
c - include lattice centering matrices
i - include inversion generated matrices if any

SGList(*args)
Built in function SGList
Signature: SGList arguments [none] states - [Loaded file is expected]
Description: Returns result of the last call to the space group determination procedure

SGS(*args, **kwds)
Built in macro SGS
Signature: SGS arguments [any except none] states - [Loaded file is expected]
Description: Changes current space group settings using provided cell setting (if applicable) and axis, or 9 transformation matrix elements and the space group symbol. If the transformed HKL file is required, it should be provided as the last argument (like test.hkl)

SIMU(*args, **kwds)
Built in macro SIMU
Signature: SIMU arguments [any] states - [INS file is expected]; valid options - cs;
Description: Similarity restraint for Uij of provided atoms. If no atoms provided, all non-H atoms considered
Options:
cs - do not clear selection

SSM(*args)
Built in function SSM
Signature: SSM arguments [none or 1]
Description: Return current structure solution method, TREF or PATT currently. If current method is unknown and an argument is provided, that argument is returned.

Sadi(*args, **kwds)
Built in macro Sadi
Signature: Sadi arguments [any] states - [INS file is expected]
Description: Similar distances restraint

Same(*args, **kwds)
Built in macro Same
Signature: Same arguments [any] states - [Loaded file is expected]; valid options - e;i;
Description: Creates SAME instruction for two fragments (two selected atoms or two atoms provided) or number_of_groups and groups following each another
(or selection)
Options:
e - expand SAME into the list of SADI
i - invert the graphs

Save(*args, **kwds)
Built in macro Save

Signature: Save arguments [any except none]
Description: Saves style/scene/view/gview/model

```
Scene(*args, **kwds)
```

Built in macro Scene
Signature: Scene arguments [none or 1]; valid options - s;
Description: Prints default scene parameters or sets it (none resets)
Options:
s - shows a file open dialog

```
Schedule(*args, **kwds)
```

Built in macro Schedule
Signature: Schedule arguments [any except none]; valid options - g;r;
Description: Schedules a particular macro (second argument) to be executed within provided interval (first argument). If the interval is not specified the
requested macro is called when the program is idle
Options:
g - requires GUI
r - repeatable

```
Sel(*args, **kwds)
```

Built in macro Sel
Signature: Sel arguments [any]; valid options - a;c;i;l;u;
Description: If no arguments provided, prints current selection. This includes distances, angles and torsion angles and other geometrical parameters. Selects atoms fulfilling provided conditions, like sel $type – selects any particular atom type; type can be one of the following shortcuts - * - for all atoms, M - for metals, X - for halogens
sel $*,type - selects all but type atoms
An extended syntax include keyword 'where' and 'rings' which allow selecting atoms and bonds according to their properties, like type and length or rings of particular connectivity like C6 or NC5. If the 'where' keyword is used, logical operators, like and (&&), and or (||) can be used to refine the selection. For example:
sel atoms where xatom.bai.z > 2 - to select all atoms heavier after H
sel bonds where xbond.length > 2 - to select all bonds longer than 2 A
sel bonds where xbond.b.bai.z == 1 - to select all bonds where the lightest atoms is H
Options:
a - select all
c - copies printed values to the clipboard
i - invert selection
l - consider the list of bonds as independent
u - unselect all
Built in function Sel
Signature: Sel arguments [none or 1] states - [Loaded file is expected]
Description: Returns current selection. By default expands bonds and planes into the list of atoms. If the 'a' argument is given, returns only selected atoms.

```
SelBack(*args, **kwds)
```

Built in macro SelBack
Signature: SelBack arguments [none]; valid options - a;o;x;
Description:
Options:
a -
o -
x -

```
SelName(*args)
```

Built in function SelName

Signature: SelName arguments [none]

Description: Returns name for the selected object group

SetCmd(*args, **kwds)

Built in macro SetCmd

Signature: SetCmd arguments [any]

Description:

SetEnv(*args, **kwds)

Built in macro SetEnv

Signature: SetEnv arguments [2]

Description: Sets an environmental variable

SetFont(*args, **kwds)

Built in macro SetFont

Signature: SetFont arguments [any except none or 1]; valid options - b;i;ps;

Description: Sets font for specified control

Options:

b - bold

i - italic

ps - point size

SetMaterial(*args, **kwds)

Built in macro SetMaterial

Signature: SetMaterial arguments [2 or 3]

Description:

#1 Assigns provided value to specified material

#2 Assigns provided value to specified material. Special materials are: helpcmd, helptxt, execout, error, exception

SetVar(*args)

Built in function SetVar

Signature: SetVar arguments [2]

Description: Sets the value of the specified variable

SetView(*args, **kwds)

Built in macro SetView

Signature: SetView arguments [any]; valid options - c;

Description: Sets view normal to the normal of the selected plane, to a bond or mean line

Options:

c - center

Sgen(*args, **kwds)

Built in macro Sgen

Signature: Sgen arguments [any except none] states - [Loaded file is expected]

Description: Grows the structure using provided atoms (all if none provided) and symmetry code.

Shell(*args, **kwds)

Built in macro Shell

Signature: Shell arguments [any]

Description: If no arguments launches a new interactive shell, otherwise runs provided file in the interactive shell (on windows ShellExecute is used to avoid flickering console).

ShowH(*args, **kwds)

Built in macro ShowH

Signature: ShowH arguments [none or 2] states - [Loaded file is expected]

Description: Changes the H-atom and H-bonds visibility

ShowP(*args, **kwds)

Built in macro ShowP

Signature: ShowP arguments [any]; valid options - m;v;

Description: Shows specified or all parts of the structure

Options:

m - do not modify the display view

v - operate only on currently visible atoms/fragments;

`ShowQ(*args, **kwds)`

Built in macro ShowQ

Signature: ShowQ arguments [none, 1 or 2]; valid options - wheel;

Description: Traverses the three states - peaks and peak bonds are visible, only peaks visible, no peaks or peak bonds. One numeric argument is taken to increment/decrement the numegbr of visibe peaks. Two aruments are taken to control the visibility of atoms or bonds, like in: 'showq a true' or 'showq b false'

Options:

wheel - number of peaks to hide (if negative) or to show

`ShowR(*args, **kwds)`

Built in macro ShowR

Signature: ShowR arguments [any]; valid options - m;v;

Description: Shows residues by number or name

Options:

m - do not modify the display view

v - operate only on currently visible atoms/fragments;

`ShowStr(*args, **kwds)`

Built in macro ShowStr

Signature: ShowStr arguments [none or 1] states - [Loaded file is expected]

Description: Shows/hides structure and console buffer

`ShowSymm(*args, **kwds)`

Built in macro ShowSymm

Signature: ShowSymm arguments [none or 1]

Description: Shows symmetry elements of the unitcell

`ShowWindow(*args, **kwds)`

Built in macro ShowWindow

Signature: ShowWindow arguments [1 or 2]

Description:

`SignPlugin(*args, **kwds)`

Built in macro SignPlugin

Signature: SignPlugin arguments [any except none or 1]

Description:

`Silent(*args, **kwds)`

Built in macro Silent

Signature: Silent arguments [none or 1]

Description: If no argument is provided, prints out current mode status. Takes 'on' and 'off' values to turn Olex2 log on and off

`Sort(*args, **kwds)`

Built in macro Sort

Signature: Sort arguments [any] states - [Loaded file is expected]

Description: Sorts atoms of the default residue. Atom sort arguments:

m - atomic mass

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 labels suffix

n - number after the atom symbol

x - atom moiety size

f - when a list of atoms provided the atoms will be ordered after the first given name - otherwise at the position of the firstly found atom in current list
w - rather than ordering atoms by given sequence this will swap their positions
Moiety sort arguments:
l - label
s - size
h - by heaviest atom
m - molecular mass
Usage: sort [+atom_sort_type] or [Atoms] [moiety [+moety sort type] [moiety atoms]]. If just 'moiety' is provided - the atoms will be split into the
moieties without sorting.
Example: sort +ml 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 preceeding sort types.

`Split(*args, **kwds)`

Built in macro Split
Signature: Split arguments [any] states - [INS file is expected]; valid options - r;
Description: Splits provided atoms along the longest axis of the ADP
Options:
r - EADP,ISOR or SIMU to be placed for the split atoms
Standardise starter StartLogging startupPanel

`Standardise(*args, **kwds)`

Built in macro Standardise
Signature: Standardise arguments [none or 1] states - [Loaded file is expected]
Description: Standardises atom coordinates (similar to HKL standardisation procedure). If '0' is provided as argument, the asymmetric unit content is
arranged as close to (0,0,0), while being inside the unit cell as possible
starter StartLogging startupPanel

`StartLogging(*args, **kwds)`

Built in macro StartLogging
Signature: StartLogging arguments [1]; valid options - c;
Description: Creates/opens for appending a log file, where all screen output is saved
Options:
c - empties the file if exists

`Stop(*args, **kwds)`

Built in macro Stop
Signature: Stop arguments [1]
Description: Switches specified mode off

`StoreParam(*args, **kwds)`

Built in macro StoreParam
Signature: StoreParam arguments [2 or 3]
Description:

`StrDir(*args)`

Built in function StrDir
Signature: StrDir arguments [none] states - [Loaded file is expected]
Description: Returns location of the folder, where Olex2 stores structure related data

`Strcat(*args)`

Built in function Strcat
Signature: Strcat arguments [2]
Description:

`Strcmp(*args)`

Built in function Strcmp

Signature: Strcmp arguments [2]

Description:

```
Style(*args, **kwds)
```

Built in macro Style

Signature: Style arguments [none or 1]; valid options - s;

Description: Prints default style or sets it (none resets)

Options:

s - shows a file open dialog

```
Sump(*args, **kwds)
```

Built in macro Sump

Signature: Sump arguments [any except none] states - [INS file is expected]

Description: Adds a linear equaltion into the refinement

```
SwapBg(*args, **kwds)
```

Built in macro SwapBg

Signature: SwapBg arguments [none]

Description: Swaps current background to white or vice-versa

```
TLS(*args, **kwds)
```

Built in macro TLS

Signature: TLS arguments [any] states - [Loaded file is expected]; valid options - a;

Description: TLS procedure. The TLS is calculated for the given atoms and then the matrices rotated to the L axes (making L diagonal) and shifted to make

S symmetric. The printed R1 is calculated for ADPs in the L axes and is:

R1=sum(i=1..3,j=i..3)(|Uobs_ij-Utls_ij|)/sum(i=1..3,j=i..3)(|Uobs_ij|)

R2' is invariant under the rotation and is calculated as

R2'=sum(i=1..3,j=1..3)((Uobs_ij-Utls_ij)^2)/sum(i=1..3,j=1..3)(Uobs_ij^2)

Options:

a - apply the TLS ADP to the atoms

```
TelpV(*args, **kwds)
```

Built in macro TelpV

Signature: TelpV arguments [1]

Description: Calculates ADP scale for given thermal probability (in %)

```
Test(*args, **kwds)
```

Built in macro Test

Signature: Test arguments [any]

Description:

```
TestBinding(*args, **kwds)
```

Built in macro TestBinding

Signature: TestBinding arguments [any]

Description: Internal tests

```
TestMT(*args, **kwds)
```

Built in macro TestMT

Signature: TestMT arguments [any]

Description: Testing multithreading

```
TestR(*args, **kwds)
```

Built in macro TestR

Signature: TestR arguments [any] states - [Loaded file is expected]

Description: Under development.

```
TestStat(*args, **kwds)
```

Built in macro TestStat

Signature: TestStat arguments [1]

Description: Test: runs statistical tests on structures in current folder. Expects a file name

```
TestSymm(*args, **kwds)
```

Built in macro TestSymm
Signature: TestSymm arguments [none] states - [Loaded file is expected]; valid options - e;
Description: Tests current structure for missing symmetry
Options:
e - tolerance limit

```
Textm(*args, **kwds)
```

???

```
ThreadCount(*args)
```

Built in function ThreadCount
Signature: ThreadCount arguments [none or 1]
Description: Returns/sets the number of simultaneous tasks

```
Title(*args)
```

Built in function Title
Signature: Title arguments [none or 1]
Description: If the file is loaded, returns it's title else if a parameter passed, it is returned

```
Tolman(*args, **kwds)
```

Built in macro Tolman
Signature: Tolman arguments [none or 5] states - [Loaded file is expected]; valid options - mpd;
Description: Calculates Tolamn code angle for the selection (M P S1 S2 S3)
Options:
mpd - M to P distance

```
Transform(*args, **kwds)
```

Built in macro Transform
Signature: Transform arguments [any] states - [Loaded file is expected]
Description: Transforms the structure or provided fragments according to the given matrix (a11, a12, a13, a21, a22, a23, a31, a32, a33, t1, t2, t3)

```
TranslatePhrase(*args)
```

Built in function TranslatePhrase
Signature: TranslatePhrase arguments [1]
Description: Translates provided phrase into current language

```
Tref(*args, **kwds)
```

Built in macro Tref
Signature: Tref arguments [1 or 2] states - [INS file is expected]
Description:

```
Tria(*args, **kwds)
```

Built in macro Tria
Signature: Tria arguments [any]; valid options - cs;
Description: Adds a distance restraint for bonds and 'angle' restraint for the angle.Takes bond pairs or atom triplets.
Options:
cs - do not clear selection

```
UncheckMenu(*args, **kwds)
```

Built in macro UncheckMenu
Signature: UncheckMenu arguments [1]
Description:

```
Undo(*args, **kwds)
```

Built in macro Undo
Signature: Undo arguments [none]
Description: Reverts some of the previous operations

```
UninstallPlugin(*args, **kwds)
```

Built in macro UninstallPlugin
Signature: UninstallPlugin arguments [1]

Description:

Uniq(*args, **kwds)

Built in macro Uniq

Signature: Uniq arguments [any] states - [Loaded file is expected]

Description: Shows only fragments specified by atom name(s) or selection

UnsetVar(*args)

Built in function UnsetVar

Signature: UnsetVar arguments [1]

Description: Removes the specified variable

Update UpdateFile UpdateOptions UpdateQPeakTable UpdateWght

Update(*args, **kwds)

Built in macro Update

Signature: Update arguments [none]; valid options - f;

Description: Does check the for the updates

Options:

f - force [true]

UpdateFile(*args, **kwds)

Built in macro UpdateFile

Signature: UpdateFile arguments [1]; valid options - f;

Description:

Options:

f -

UpdateOptions(*args, **kwds)

Built in macro UpdateOptions

Signature: UpdateOptions arguments [none]

Description: Shows the Update Options dialog

UpdateQPeakTable(*args, **kwds)

Built in macro UpdateQPeakTable

Signature: UpdateQPeakTable arguments [none] states - [Loaded file is expected]

Description: Internal routine for synchronisation

UpdateWght(*args, **kwds)

Built in macro UpdateWght

Signature: UpdateWght arguments [any] states - [INS file is expected]

Description: Copies proposed weight to current

User(*args, **kwds)

Built in macro User

Signature: User arguments [none or 1]

Description: Changes current folder

VATA(*args, **kwds)

Built in macro VATA

Signature: VATA arguments [any] states - [Loaded file is expected]

Description: Compares current model with the cif file and write the report to provided file (appending)

VSS(*args)

Built in function VSS

Signature: VSS arguments [1] states - [Loaded file is expected]

Description: Validate Structure or Solution. Takes a boolean value. If value is true, the number of tested atoms is limited by the 18A rule. Returns proportion of known atom types to the all atoms number.

VVol(*args)

Built in function VVol

Signature: VVol arguments [none or 1] states - [Loaded file is expected]

Description: A simplistic procedure to calculate molecular volume - only pairwise overlapping of atoms is considered.

ValidatePlugin(*args)

Built in function ValidatePlugin

Signature: ValidatePlugin arguments [1]

Description:

ViewGrid(*args, **kwds)

Built in macro ViewGrid

Signature: ViewGrid arguments [none or 1]

Description:

ViewLattice(*args, **kwds)

Built in macro ViewLattice

Signature: ViewLattice arguments [1]

Description: Loads cell information from provided file and displays it on screen as lattice points/grid

VoidE(*args, **kwds)

Built in macro VoidE

Signature: VoidE arguments [none] states - [Loaded file is expected]

Description: calculates number of electrons in the voids area

WBox(*args, **kwds)

Built in macro WBox

Signature: WBox arguments [any] states - [Loaded file is expected]; valid options - s;w;

Description: Calculates wrapping box around provided box using the set of best, intermidiate and worst planes

Options:

s - create separate boxes for fragments

w - use atomic mass instead of unit weights for atoms

Wait(*args, **kwds)

Built in macro Wait

Signature: Wait arguments [1]; valid options - r;

Description: Forces Olex2 and calling process to sleep for provided number of milliseconds

Options:

r - during wait all events are processed

WaitFor(*args, **kwds)

Built in macro WaitFor

Signature: WaitFor arguments [1]

Description:

Wilson(*args, **kwds)

Built in macro Wilson

Signature: Wilson arguments [none or 1] states - [Loaded file is expected]; valid options - b;p;

Description: Prints Wilson plot data

Options:

b - number of bins

p - uses linear bins for picture, otherwise uses spherical bins

WindowCmd(*args, **kwds)

Built in macro WindowCmd

Signature: WindowCmd arguments [any except none or 1]

Description: Windows specific command which send a command to a process with GUI window. First argument is the window name, the second is the command. 'nl' is considered as a new line char and 'sp' as white space char.

quit(*args, **kwds)

Built in macro quit

Signature: quit arguments [none]

Description: Exits Olex2