klayout\_snapper

v0.0.1

# Overview

A technology agnostic collection of

1. Helper functions implementing actions that are not available for keybinding in the vanilla version of KLayout
2. Helper functions for drawing polygons and instances
3. Utilities for snapping objects by edges, to nearest neighbours, or cursors
4. Utilities to manipulate paths, conversion to path based PCells, and routing

and libraries supporting these utilities. These functions are assigned keybindings for ease of use.

## snapper

\_\_init\_\_.py

pcell.py

ports.py

snap.py

utils.py

## Helper Functions, Actions

Functions for navigating cells, switching between selections states, grid sizes, zoom, and instantiation of commonly used PCells/shapes that are otherwise not exposed for keybinding or not implemented in the vanilla version of KLayout

current\_coordinates.lym

duplicate\_nozoom.lym

edit\_top\_level\_selection\_off.lym

edit\_top\_level\_selection\_on.lym

enter\_instance.lym

exit\_instance.lym

flatten\_paths.lym

flatten\_selected\_cells.lym

grid\_add\_1.lym

grid\_add\_5.lym

grid\_add\_p001.lym

grid\_add\_p025.lym

grid\_add\_p1.lym

grid\_edit\_1.lym

grid\_edit\_100.lym

grid\_edit\_25.lym

grid\_edit\_5.lym

grid\_edit\_none.lym

grid\_edit\_p001.lym

grid\_edit\_p025.lym

grid\_edit\_p1.lym

instance\_replace.lym

merge\_cell\_variants\_keep\_new.lym

merge\_cell\_variants\_keep\_old.lym

remove\_$.lym

select\_all\_instances.lym

select\_containing\_instance.lym

to\_top.lym

zoom\_1.lym

zoom\_100.lym

zoom\_25.lym

zoom\_5.lym

zoom\_50dbu.lym

zoom\_p1.lym

## Helper Functions, Drawing

Helper functions to aid drawing/organize drawn components

add\_selected\_cells\_col.lym

add\_selected\_cells\_row.lym

add\_bounding\_box.lym

add\_bounding\_boxes.lym

bud\_port.lym

draw\_connecting\_poly.lym

space\_instances\_x.lym

space\_instances\_y.lym

instance\_array\_col.lym

instance\_array\_row.lym

instance\_bend180.lym

instance\_bend90.lym

## Snap Utilities

Utilities for snapping objects by edges, to nearest neighbours, or cursors

snap\_box\_down.lym

snap\_box\_left

snap\_box\_left.lym

snap\_box\_right.lym

snap\_box\_up.lym

snap\_instances\_to\_grid.lym

snap\_obj.lym

snap\_obj\_align\_bot\_left.lym

snap\_obj\_align\_top\_right.lym

snap\_path.lym

snap\_path\_align\_bot\_left.lym

snap\_path\_align\_top\_right.lym

## Path Utilities

Utilities to manipulate paths, conversion to path based PCells, and routing

add\_angle\_to\_path.lym

bud\_path.lym

bud\_path\_bend.lym

join\_or\_split\_paths.lym

revert\_to\_path.lym

path\_to\_clearancepath.lym

path\_to\_roundpath.lym

path\_to\_waveguide.lym

straighten\_path.lym

# Function Descriptions

### add\_angle\_to\_path

Turns the closest corner on a selected path to a 45 degree bend

Assumptions

----------

Inputs

----------

Selected path/guiding shape of a PCell

Behaviour

----------

Turns the closest corner on a selected path to a 45 degree bend

See also

----------

### add\_bounding\_box

Cover a selection of objects with a covering bounding box of the selected layers

Assumptions

----------

Inputs

----------

Selected objects

Behaviour

----------

Cover the selection of objects with a covering bounding box of the selected layers

See also

----------

add\_bounding\_boxes

### add\_bounding\_box

Cover a selection of objects each individually with a bounding box of the selected layers

Assumptions

----------

Inputs

----------

Selected objects

Behaviour

----------

Cover the selection of objects each individually with a covering bounding box of the selected layers

See also

----------

add\_bounding\_box

add\_bounding\_boxes

### add\_selected\_cells\_col

Add selected cells as instances in a column to the current layout

Assumptions

----------

Inputs

----------

Selected object (optional)

Selected cells in cell hierarchy

Behaviour

----------

Create instances in a column using the selected object's bounding box as a reference for spacing and position

See also

----------

add\_selected\_cells\_row

add\_selected\_cells\_col

### add\_selected\_cells\_col

Add selected cells as instances in a column to the current layout

Assumptions

----------

Inputs

----------

Selected object (optional)

Selected cells in cell hierarchy

Behaviour

----------

Create instances in a column using the selected object's bounding box as a reference for spacing and position

See also

----------

add\_selected\_cells\_row

add\_selected\_cells\_col

### bud\_path

Assumptions

----------

Inputs

----------

1)

a) Selected non-path object (including PCells)

b) Selected path (Path/Path guiding shape)

2)

Cursor location

Behaviour

----------

1a)

Create a new path centered at an edge closest to the current cursor position

1b)

Extend the end of the path towards the current cursor position, along diagonal/manhattan directions

### bud\_path\_bend

Assumptions

----------

Inputs

----------

1) Selected path (Path/Path guiding shape)

2)

Cursor location

Behaviour

----------

Append a 90 degree bend at the end of the path closest to the cursor. The side length of the bend is set according to the config parameter 'bend-radius'

### bud\_port

Assumptions

----------

Inputs

----------

1) Selected object

2) Cursor location

Behaviour

----------

Create a port, on the port gdslayer, on a previously unoccupied datatype on that cell

at the edge closest to the current cursor position

### clear\_all\_zero\_area\_shapes

Clears all shapes (excluding Text) having zero area.

### current\_coordinates

Display coordinates in copyable string form in a dialog box

Assumptions

----------

Inputs

----------

User moves cursor to desired location

Behaviour

----------

Displays dialog box with coordinates consistent with coordinates displayed on the bottom right of window.

### draw\_connecting\_poly

Assumptions

----------

Connecting edges are on Manhattan directions.

Inputs

----------

Selection of an object

and

1) Cursor location

or

2) A second object

Behaviour

----------

Draws a connecting polygon between the selection between the closest edge to 1) the current cursor location, or 2) the closest edge of a second selected object

See also

----------

### duplicate\_nozoom

Duplicate selected objects without adjusting zoom (quality of life)

Assumptions

----------

Inputs

----------

Selection (multiple) of shapes or instances

Behaviour

----------

Duplicates the selection without adjusting zoom,

in contrast to built-in duplicate function.

### edit\_top\_level\_selection\_off

Toggles Object Editor Options > Basic Editing > Objects > Snap to other objects

OFF

### edit\_top\_level\_selection\_on

Toggles Object Editor Options > Basic Editing > Objects > Snap to other objects

ON

### enter\_instance

Enters a selected instance, viewing it as a top cell

### exit\_instance

Exits the current instance to its parent, viewing it as a top cell

### flatten\_paths

In selection, if any paths present, convert them to polygons or boxes.

### flatten\_selected\_cells

Flatten all instances of cells selected in the Cells hiearchy

### grid\_add\_1

Keyboard shortcut to increment editing Object Editor Options > Basic Editing > Snapping > Grid to

Current value + 1 um

Assumptions

----------

Inputs

----------

Behaviour

----------

Keyboard shortcut to increment editing Object Editor Options > Basic Editing > Snapping > Grid to

Current value + 1 um

### grid\_add\_5

Keyboard shortcut to increment editing Object Editor Options > Basic Editing > Snapping > Grid to

Current value + 5 um

Assumptions

----------

Inputs

----------

Behaviour

----------

Keyboard shortcut to increment editing Object Editor Options > Basic Editing > Snapping > Grid to

Current value + 5 um

### grid\_add\_p001

Keyboard shortcut to increment editing Object Editor Options > Basic Editing > Snapping > Grid to

Current value + 0.001 um

Assumptions

----------

Inputs

----------

Behaviour

----------

Keyboard shortcut to increment editing Object Editor Options > Basic Editing > Snapping > Grid to

Current value + 0.001 um

### grid\_add\_p025

Keyboard shortcut to increment editing Object Editor Options > Basic Editing > Snapping > Grid to

Current value + 0.025 um

Assumptions

----------

Inputs

----------

Behaviour

----------

Keyboard shortcut to increment editing Object Editor Options > Basic Editing > Snapping > Grid to

Current value + 0.025 um

### grid\_add\_p1

Keyboard shortcut to increment editing Object Editor Options > Basic Editing > Snapping > Grid to

Current value + 0.1 um

Assumptions

----------

Inputs

----------

Behaviour

----------

Keyboard shortcut to increment editing Object Editor Options > Basic Editing > Snapping > Grid to

Current value + 0.1 um

### grid\_edit\_1

Keyboard shortcut to set editing Object Editor Options > Basic Editing > Snapping > Grid to

1 um

Assumptions

----------

Inputs

----------

Behaviour

----------

Keyboard shortcut to set editing Object Editor Options > Basic Editing > Snapping > Grid to

1 um

### grid\_edit\_100

Keyboard shortcut to set editing Object Editor Options > Basic Editing > Snapping > Grid to

100 um

Assumptions

----------

Inputs

----------

Behaviour

----------

Keyboard shortcut to set editing Object Editor Options > Basic Editing > Snapping > Grid to

100 um

### grid\_edit\_25

Keyboard shortcut to set editing Object Editor Options > Basic Editing > Snapping > Grid to

25 um

Assumptions

----------

Inputs

----------

Behaviour

----------

Keyboard shortcut to set editing Object Editor Options > Basic Editing > Snapping > Grid to

25 um

### grid\_edit\_5

Keyboard shortcut to set editing Object Editor Options > Basic Editing > Snapping > Grid to

5 um

Assumptions

----------

Inputs

----------

Behaviour

----------

Keyboard shortcut to set editing Object Editor Options > Basic Editing > Snapping > Grid to

5 um

### grid\_edit\_none

Keyboard shortcut to set editing Object Editor Options > Basic Editing > Snapping > Grid to

No grid

Assumptions

----------

Inputs

----------

Behaviour

----------

Keyboard shortcut to set editing Object Editor Options > Basic Editing > Snapping > Grid to

No grid

### grid\_edit\_p001

Keyboard shortcut to set editing Object Editor Options > Basic Editing > Snapping > Grid to

0.001 um

Assumptions

----------

Inputs

----------

Behaviour

----------

Keyboard shortcut to set editing Object Editor Options > Basic Editing > Snapping > Grid to

0.001 um

### grid\_edit\_p025

Keyboard shortcut to set editing Object Editor Options > Basic Editing > Snapping > Grid to

0.025 um

Assumptions

----------

Inputs

----------

Behaviour

----------

Keyboard shortcut to set editing Object Editor Options > Basic Editing > Snapping > Grid to

0.025 um

### grid\_edit\_p1

Keyboard shortcut to set editing Object Editor Options > Basic Editing > Snapping > Grid to

0.1 um

Assumptions

----------

Inputs

----------

Behaviour

----------

Keyboard shortcut to set editing Object Editor Options > Basic Editing > Snapping > Grid to

0.1 um

### instance\_annotation1

Add a Text pcell with "1" in text

### instance\_array\_col

Turn the selected instance into an array (vertical)

### instance\_array\_row

Turn the selected instance into an array (horizontal)

### instance\_bend180

Instance a circular 180 degree bend waveguide

### instance\_bend90

Instance a circular 90 degree bend waveguide

### instance\_replace

Replace the current selected instance(s) with cell selected in cell hierarchy

### join\_or\_split\_paths

Assumptions

----------

Path object includes Path, PCells with Path Guiding Shapes, Path Guiding Shape

Inputs

----------

Situation dependent:

1) 1 selected path object, and cursor location

2) 2 selected path objects

Behaviour

----------

1) Cleave the path at the location on the path closest to the grid point closest to cursor

1) Join the 2 selected paths

### merge\_cell\_variants\_keep\_new

Assumptions

----------

In the cell hierarchy, base cell exists

e.g.

Both CELL$1 and CELL are in the layout. CELL$1 is selected in the Cells hierarchy viewer.

Inputs

----------

Selection (multiple) of cell variants (cells with $ in the name) in cell hierarchy viewer

Behaviour

----------

Merge a selected cell variant ($) with the base variant, keeping the variant, replacing all instances in the layout

If no $ in the cell name, do nothing.

If no base variant of the cell name exists, do nothing.

If base variant of the cell exists, replace this current cell and all instances with the base variant.

See also

----------

merge\_cell\_variants\_keep\_old

remove\_$

### merge\_cell\_variants\_keep\_old

Assumptions

----------

In the cell hierarchy, base cell exists

e.g.

Both CELL$1 and CELL are in the layout. CELL$1 is selected in the Cells hierarchy viewer.

Inputs

----------

Selection (multiple) of cell variants (cells with $ in the name) in cell hierarchy viewer

Behaviour

----------

Merge a selected cell variant ($) with the base variant, keeping the old version, replacing all instances in the layout

If no $ in the cell name, do nothing.

If no base variant of the cell name exists, do nothing.

If base variant of the cell exists, replace this current cell and all instances with the base variant.

See also

----------

merge\_cell\_variants\_keep\_new

remove\_$

### path\_to\_clearancepath

Convert a selected path into a ClearancePath PCell

### path\_to\_roundpath

Convert a selected path into a ROUND\_PATH PCell

### path\_to\_waveguide

Converts a selected path into waveguide routing, with corners being a selected bend cell from the device library

Assumptions

----------

Bend cell has port marker polygons on its terminations. Used to calculate its orientation and positioning.

Inputs

----------

Selection of a path. Path is manhattan.

Parameters for routing (set in option editor)

Behaviour

----------

Corners of path with sufficient clearance are replaced with the library bend cell. Straight segments of the path are filled in with straight paths.

All other remaining non-manhattan paths are left as is.

See also

----------

path\_to\_roundpath

path\_to\_clearancepath

revert\_to\_path

### remove\_$

Assumptions

----------

In the cell hierarchy, no base cell variant exists.

e.g.

CELL$1 is in the layout, but not CELL

Inputs

----------

Selection (multiple) of cell in cell hierarchy viewer, with a cell name including $

Behaviour

----------

Remove $ from cell names where a base variant does not exist.

If no base variant of the cell name exists, renames the selected cell name to the base variant

See also

----------

merge\_cell\_variants\_keep\_new

merge\_cell\_variants\_keep\_old

### revert\_to\_path

Inputs

----------

1) Selection of a path. Path is manhattan.

OR

2) Selection of a PCell with a path guiding shape

Behaviour

----------

1)

If selection is a path with routing, clears the routing on the selected path.

2)

Reverts various PCells with path guiding shape back to Path.

### select\_all\_instances

Select all instances within current cellview currently selected in the cell hierarchy

Assumptions

----------

Cells selected in cell hierarchy exist on the current viewed cell

Inputs

----------

Cells selected in cell hierarchy

Behaviour

----------

All instances of cells selected in cell hierarchy are added to currently selected objects

### select\_containing\_instance

Changes the selection to the instance containing the currently selected object, if exists

### snap\_box\_down

Adjust the BOTTOM edge of the currently selected box to the nearest edge of the second selection

Inputs

----------

2 selected objects:

1) a Box

2) an object

Behaviour

----------

The BOTTOM edge of the currently selected box is

adjusted to overlap with the edge of the object closest to the box.

If the object is an instance, we consider its bounding box.

### snap\_box\_left

Adjust the LEFT edge of the currently selected box to the nearest edge of the second selection

Inputs

----------

2 selected objects:

1) a Box

2) an object

Behaviour

----------

The LEFT edge of the currently selected box is

adjusted to overlap with the edge of the object closest to the box.

If the object is an instance, we consider its bounding box.

### snap\_box\_right

Adjust the RIGHT edge of the currently selected box to the nearest edge of the second selection

Inputs

----------

2 selected objects:

1) a Box

2) an object

Behaviour

----------

The RIGHT edge of the currently selected box is

adjusted to overlap with the edge of the object closest to the box.

If the object is an instance, we consider its bounding box.

snap\_box\_top

Adjust the TOP edge of the currently selected box to the nearest edge of the second selection

Inputs

----------

2 selected objects:

1) a Box

2) an object

Behaviour

----------

The TOP edge of the currently selected box is

adjusted to overlap with the edge of the object closest to the box.

If the object is an instance, we consider its bounding box.

### snap\_instances\_to\_grid

Adjust the positions of selected instances to nearest grid point.

### snap\_obj

Snaps a selected object to 1) Cursor 2) an object 3) aligned to multiple objects

Align by center

Assumptions

----------

All snapping occurs on a Manhattan basis. All edges that are snapped are vertically or horizontally aligned. Behaviour of rotated edges not guaranteed.

If selected object is an instance, we assume it either has :ports: drawn at its ports, or we will assume we want to snap only to edges on the boundaries of its bounding box.

Inputs

----------

Situation 1: 1 selected object

Situation 2: 2 selected objects

Situation 3: 3 or more selected objects.

Behaviour

----------

Situation 1:

The edge of the selected object closest to the cursor will be moved to the grid point closest to the cursor by translation (no rotation or mirroring)

Situation 2:

The first selected object is brought to the second object, to join the closest edges.

Situation 3:

The first selected object is brought to the second object, for the axis traverse to its edge, and then the same to the third selected object.

e.g. For aligning a 90 degree bend to 2 waveguides in orthogonal directions.

See also

----------

snap\_obj

snap\_obj\_align\_bot\_left

snap\_obj\_align\_top\_right

### snap\_obj\_align\_bot\_left

Snaps a selected object to 1) Cursor 2) an object 3) aligned to multiple objects

Align by bottom/left

Assumptions

----------

All snapping occurs on a Manhattan basis. All edges that are snapped are vertically or horizontally aligned. Behaviour of rotated edges not guaranteed.

If selected object is an instance, we assume it either has :ports: drawn at its ports, or we will assume we want to snap only to edges on the boundaries of its bounding box.

Inputs

----------

Situation 1: 1 selected object

Situation 2: 2 selected objects

Situation 3: 3 or more selected objects.

Behaviour

----------

Situation 1:

The edge of the selected object closest to the cursor will be moved to the grid point closest to the cursor by translation (no rotation or mirroring)

Situation 2:

The first selected object is brought to the second object, to join the closest edges.

Situation 3:

The first selected object is brought to the second object, for the axis traverse to its edge, and then the same to the third selected object.

e.g. For aligning a 90 degree bend to 2 waveguides in orthogonal directions.

See also

----------

snap\_obj

snap\_obj\_align\_bot\_left

snap\_obj\_align\_top\_right

### snap\_obj\_align\_top\_right

Snaps a selected object to 1) Cursor 2) an object 3) aligned to multiple objects

Align top/right

Assumptions

----------

All snapping occurs on a Manhattan basis. All edges that are snapped are vertically or horizontally aligned. Behaviour of rotated edges not guaranteed.

If selected object is an instance, we assume it either has :ports: drawn at its ports, or we will assume we want to snap only to edges on the boundaries of its bounding box.

Inputs

----------

Situation 1: 1 selected object

Situation 2: 2 selected objects

Situation 3: 3 or more selected objects.

Behaviour

----------

Situation 1:

The edge of the selected object closest to the cursor will be moved to the grid point closest to the cursor by translation (no rotation or mirroring)

Situation 2:

The first selected object is brought to the second object, to join the closest edges.

Situation 3:

The first selected object is brought to the second object, for the axis traverse to its edge, and then the same to the third selected object.

e.g. For aligning a 90 degree bend to 2 waveguides in orthogonal directions.

See also

----------

snap\_obj

snap\_obj\_align\_bot\_left

snap\_obj\_align\_top\_right

### snap\_path

Snaps a selected path/PCell with path guiding shape/guiding shape path of a PCell

to 1) cursor a) at the end of the path, b) along the path 2) the end of the path to the [alignment] of an edge of a subsequently selected object(s)

[alignment] edge center

For simplicity, we refer to any path/PCell with path guiding shape/guiding shape path of a PCell generally as path

Assumptions

----------

All snapping occurs on a Manhattan basis. All edges that are snapped are vertically or horizontally aligned. Behaviour of rotated edges not guaranteed.

If selected object is an instance, we assume it either has :ports: drawn at its ports, or we will assume we want to snap only to edges on the boundaries of its bounding box.

Inputs

----------

Situation 1: 1 selected object (the path)

a) Cursor close to the ends

b) Cursor close to a segment along the path

Situation 2: 2 selected or more selected objects

Behaviour

----------

Situation 1:

a) The end of the path is extended/contracted such that the end of the path ends on the grid point closest to the cursor. Angles of the previous line segments are preserved in the translation of the end point.

b) The line segment of the path closest to the cursor is adjusted such that the segment now overlaps the grid point closest to the cursor. Angles of surrounding line segments are preserved.

Situation 2:

The selected path (first selection) has its end points translated to meet the edges of the subsequent selections.

See also

----------

snap\_path

snap\_path\_align\_bot\_left

snap\_path\_align\_top\_right

### snap\_path\_align\_bot\_left

Snaps a selected path/PCell with path guiding shape/guiding shape path of a PCell

to 1) cursor a) at the end of the path, b) along the path 2) the end of the path to the [alignment] of an edge of a subsequently selected object(s)

[alignment] bottom/left

For simplicity, we refer to any path/PCell with path guiding shape/guiding shape path of a PCell generally as path

Assumptions

----------

All snapping occurs on a Manhattan basis. All edges that are snapped are vertically or horizontally aligned. Behaviour of rotated edges not guaranteed.

If selected object is an instance, we assume it either has :ports: drawn at its ports, or we will assume we want to snap only to edges on the boundaries of its bounding box.

Inputs

----------

Situation 1: 1 selected object (the path)

a) Cursor close to the ends

b) Cursor close to a segment along the path

Situation 2: 2 selected or more selected objects

Behaviour

----------

Situation 1:

a) The end of the path is extended/contracted such that the end of the path ends on the grid point closest to the cursor. Angles of the previous line segments are preserved in the translation of the end point.

b) The line segment of the path closest to the cursor is adjusted such that the segment now overlaps the grid point closest to the cursor. Angles of surrounding line segments are preserved.

Situation 2:

The selected path (first selection) has its end points translated to meet the edges of the subsequent selections.

See also

----------

snap\_path

snap\_path\_align\_bot\_left

snap\_path\_align\_top\_right

### snap\_path\_align\_top\_right

Snaps a selected path/PCell with path guiding shape/guiding shape path of a PCell

to 1) cursor a) at the end of the path, b) along the path 2) the end of the path to the [alignment] of an edge of a subsequently selected object(s)

[alignment] top/right

For simplicity, we refer to any path/PCell with path guiding shape/guiding shape path of a PCell generally as path

Assumptions

----------

All snapping occurs on a Manhattan basis. All edges that are snapped are vertically or horizontally aligned. Behaviour of rotated edges not guaranteed.

If selected object is an instance, we assume it either has :ports: drawn at its ports, or we will assume we want to snap only to edges on the boundaries of its bounding box.

Inputs

----------

Situation 1: 1 selected object (the path)

a) Cursor close to the ends

b) Cursor close to a segment along the path

Situation 2: 2 selected or more selected objects

Behaviour

----------

Situation 1:

a) The end of the path is extended/contracted such that the end of the path ends on the grid point closest to the cursor. Angles of the previous line segments are preserved in the translation of the end point.

b) The line segment of the path closest to the cursor is adjusted such that the segment now overlaps the grid point closest to the cursor. Angles of surrounding line segments are preserved.

Situation 2:

The selected path (first selection) has its end points translated to meet the edges of the subsequent selections.

See also

----------

snap\_path

snap\_path\_align\_bot\_left

snap\_path\_align\_top\_right

### space\_instances\_x

Space out instances horizontally

### space\_instances\_y

Space out instances vertically

### straighten\_path

Ensures only Manhattan translations are within the path.

### to\_top

View top cell containing the current instance

### zoom\_1

Zooms current cellview to 1 um span box

### zoom\_100

Zooms current cellview to 100 um span box

### zoom\_25

Zooms current cellview to 25 um span box

### zoom\_5

Zooms current cellview to 5 um span box

### zoom\_50dbu

Zooms current cellview to 50 dbu span box

### zoom\_p1

Zooms current cellview to 0.1 um span box