

---

# KiCad Documentation

*Release 0.1*

Thomas Pointhuber

Sep 09, 2018



<b>1</b>	<b>kicad.pcbnew</b>	<b>3</b>
1.1	Board . . . . .	3
1.2	Dimension . . . . .	5
1.3	Drawsegment . . . . .	6
1.4	Arc . . . . .	6
1.5	Circle . . . . .	7
1.6	Line . . . . .	8
1.7	Polygon . . . . .	9
1.8	Layer . . . . .	9
1.9	Module . . . . .	10
1.10	Net . . . . .	11
1.11	Pad . . . . .	11
1.12	PcbTarget . . . . .	12
1.13	Text . . . . .	13
1.14	Track . . . . .	14
1.15	Via . . . . .	15
1.16	Zone . . . . .	15
<b>2</b>	<b>kicad.primitives</b>	<b>17</b>
2.1	Polygon . . . . .	17
<b>3</b>	<b>kicad.util</b>	<b>19</b>
3.1	Point2D . . . . .	19
<b>4</b>	<b>kicad.plotter</b>	<b>21</b>
<b>5</b>	<b>List PCB Entities</b>	<b>23</b>
<b>6</b>	<b>Indices and tables</b>	<b>25</b>
	<b>Python Module Index</b>	<b>27</b>



This is the initial proposal for a high level [KiCad](#). Python API.

Main goals are:

- easy to understand
- everything is documented and tested
- stable

Please note this in initial development. Everything can change, and this is at the moment an unofficial realisation of this idea (thus unstable 3rd-party software).



### 1.1 Board

**class** `kicad.pcbnew.Board` (*board=None*)

Create a new Board object

**Parameters** `board` (`pcbnew.BOARD`) – already existing board object

**Example**

```
>>> from kicad.pcbnew import Board
>>> b = Board()
```

**aux\_origin**

Aux origin of Board

**Returns** `kicad.util.Point2D`

**Example**

```
>>> from kicad.pcbnew import Board
>>> b = Board()
>>> b.aux_origin = [1, 2]
>>> b.aux_origin
kicad.util.point.Point2D(1.0, 2.0)
```

**filepath**

Filepath of the Board

**Returns** unicode

**Example**

```
>>> from kicad.pcbnew import Board
>>> b = Board()
>>> b.filepath = "path/to/board.kicad_mod"
```

(continues on next page)

(continued from previous page)

```
>>> print(b.filepath)
path/to/board.kicad_mod
```

**static from\_editor()**

Get the current board visible in pcbnew

**Returns** *kicad.pcbnew.Board***Example**

```
>>> from kicad.pcbnew import Board
>>> b = Board.from_editor()
```

**static from\_file(path)**

Load a board from a given filepath

**Parameters** *path* (str, unicode) – path to the “.kicad\_mod” file**Returns** *kicad.pcbnew.Board***Example**

```
>>> from kicad.pcbnew import Board
>>> b = Board.from_file("path/to/board.kicad_mod")
```

**get\_native()**

Get native object from the low level API

**Returns** *pcbnew.BOARD***grid\_origin**

Grid origin of Board

**Returns** *kicad.util.Point2D***Example**

```
>>> from kicad.pcbnew import Board
>>> b = Board()
>>> b.grid_origin = [1, 2]
>>> b.grid_origin
kicad.util.point.Point2D(1.0, 2.0)
```

**is\_highlighted**

is highlighted?

**Returns** *bool***is\_locked**

is locked?

**Returns** *bool***is\_selected**

is selected?

**Returns** *bool***layer**

primary layer of the item

**Returns** *kicad.pcbnew.Layer*



**layers**

All layers where the item is present on

**Returns** `kicad.pcbnew.LayerSet`

**modules**

List of Modules present in the Board

**Returns** Iterator over `kicad.pcbnew.Module`

**tracks**

List of Tracks present in the Board

**Returns** Iterator over `kicad.pcbnew.Track`

**vias**

List of Vias present in the Board

**Returns** Iterator over `kicad.pcbnew.Via`

**zones**

List of Zones present in the Board

**Returns** Iterator over `kicad.pcbnew.Zone`

## 1.2 Dimension

```
class kicad.pcbnew.Dimension (dimension)
```

**get\_native()**

Get native object from the low level API

**Returns** `pcbnew.EDA_TEXT`

**is\_highlighted**

is highlighted?

**Returns** `bool`

**is\_locked**

is locked?

**Returns** `bool`

**is\_selected**

is selected?

**Returns** `bool`

**layer**

layer of the drawsegment

**Returns** `kicad.pcbnew.Layer`

**layers**

All layers where the item is present on

**Returns** `kicad.pcbnew.LayerSet`

**text**

Text

**Returns** `unicode`

**value**  
value in mm  
**Returns** float

## 1.3 Drawsegment

**class** kicad.pcbnew.**Drawsegment** (*drawsegment*)

**get\_native**()  
Get native object from the low level API  
**Returns** pcbnew.DRAWSEGMENT

**is\_highlighted**  
is highlighted?  
**Returns** bool

**is\_locked**  
is locked?  
**Returns** bool

**is\_selected**  
is selected?  
**Returns** bool

**layer**  
layer of the drawsegment  
**Returns** *kicad.pcbnew.Layer*

**layers**  
All layers where the item is present on  
**Returns** kicad.pcbnew.LayerSet

**width**  
Width of line in mm  
**Returns** float

## 1.4 Arc

**class** kicad.pcbnew.**Arc** (*arc*)

**angle**  
angle of arc in degree  
**Returns** float

**center**  
Center point of arc  
**Returns** *kicad.util.Point2D*

**get\_native()**  
Get native object from the low level API  
**Returns** `pcbnew.DRAWSEGMENT`

**is\_highlighted**  
is highlighted?  
**Returns** `bool`

**is\_locked**  
is locked?  
**Returns** `bool`

**is\_selected**  
is selected?  
**Returns** `bool`

**layer**  
layer of the drawsegment  
**Returns** `kicad.pcbnew.Layer`

**layers**  
All layers where the item is present on  
**Returns** `kicad.pcbnew.LayerSet`

**start**  
Start point of arc  
**Returns** `kicad.util.Point2D`

**width**  
Width of line in mm  
**Returns** `float`

## 1.5 Circle

**class** `kicad.pcbnew.Circle` (*circle*)

**center**  
Center point of circle  
**Returns** `kicad.util.Point2D`

**diameter**  
Diameter of circle  
**Returns** `float`

**get\_native()**  
Get native object from the low level API  
**Returns** `pcbnew.DRAWSEGMENT`

**is\_highlighted**  
is highlighted?  
**Returns** `bool`

**is\_locked**  
is locked?  
**Returns** bool

**is\_selected**  
is selected?  
**Returns** bool

**layer**  
layer of the drawsegment  
**Returns** *kicad.pcbnew.Layer*

**layers**  
All layers where the item is present on  
**Returns** *kicad.pcbnew.LayerSet*

**radius**  
Radius of circle  
**Returns** float

**width**  
Width of line in mm  
**Returns** float

## 1.6 Line

```
class kicad.pcbnew.Line(line)

    end
    End point of line
    Returns kicad.util.Point2D

    get_native()
    Get native object from the low level API
    Returns pcbnew.DRAWSEGMENT

    is_highlighted
    is highlighted?
    Returns bool

    is_locked
    is locked?
    Returns bool

    is_selected
    is selected?
    Returns bool

    layer
    layer of the drawsegment
    Returns kicad.pcbnew.Layer
```

**layers**

All layers where the item is present on

**Returns** `kicad.pcbnew.LayerSet`

**start**

Start point of line

**Returns** `kicad.util.Point2D`

**width**

Width of line in mm

**Returns** `float`

## 1.7 Polygon

```
class kicad.pcbnew.Polygon(polygon)
```

**get\_native()**

Get native object from the low level API

**Returns** `pcbnew.DRAWSEGMENT`

**is\_highlighted**

is highlighted?

**Returns** `bool`

**is\_locked**

is locked?

**Returns** `bool`

**is\_selected**

is selected?

**Returns** `bool`

**layer**

layer of the drawsegment

**Returns** `kicad.pcbnew.Layer`

**layers**

All layers where the item is present on

**Returns** `kicad.pcbnew.LayerSet`

**width**

Width of line in mm

**Returns** `float`

## 1.8 Layer

```
class kicad.pcbnew.Layer(id)
```

```
static from_id(id)
    Get Layer object from id
    Parameters id(int) –
    Returns kicad.pcbnew.Layer

static from_name(name)
    Get Layer object from name
    Parameters name –
    Returns kicad.pcbnew.Layer

id
    internal ID of the layer
    Returns int

name
    name of the layer
    Returns unicode

class kicad.pcbnew.LayerSet(layer_set)

get_native()
    Get native object from the low level API :return: pcbnew.LSET
```

## 1.9 Module

```
class kicad.pcbnew.Module(module)

description
    Description of the Module
    Returns unicode

static from_editor()
    Get the current module

get_native()
    Get native object from the low level API
    Returns pcbnew.MODULE

is_highlighted
    is highlighted?
    Returns bool

is_locked
    is locked?
    Returns bool

is_selected
    is selected?
    Returns bool
```

**keywords**

Keywords of the Module

**Returns** unicode

**layer**

primary layer of the item

**Returns** `kicad.pcbnew.Layer`

**layers**

All layers where the item is present on

**Returns** `kicad.pcbnew.LayerSet`

**pads**

List of Pads present in the Module

**Returns** Iterator over `kicad.pcbnew.Pad`

**position**

Position of the Module

**Returns** `kicad.util.Point2D`

**reference**

Reference of the Module

**Returns** unicode

**value**

Value of the Module

**Returns** unicode

## 1.10 Net

```
class kicad.pcbnew.Net (netinfo)
```

**get\_native()**

Get native object from the low level API

**Returns** `pcbnew.NETINFO_ITEM`

**name**

Name of Net

**Returns** unicode

## 1.11 Pad

```
class kicad.pcbnew.Pad (pad)
```

**get\_native()**

Get native object from the low level API

**Returns** `pcbnew.PAD`

**is\_highlighted**  
is highlighted?  
**Returns** bool

**is\_locked**  
is locked?  
**Returns** bool

**is\_selected**  
is selected?  
**Returns** bool

**layer**  
primary layer of the item  
**Returns** kicad.pcbnew.Layer

**layers**  
All layers where the item is present on  
**Returns** kicad.pcbnew.LayerSet

**name**  
Name of the Pad  
**Returns** unicode

**net**  
Net of the Zone  
**Returns** *kicad.pcbnew.Net*

## 1.12 PcbTarget

**class** kicad.pcbnew.PcbTarget (*target*)

**get\_native()**  
Get native object from the low level API  
**Returns** pcbnew.EDA\_TEXT

**is\_highlighted**  
is highlighted?  
**Returns** bool

**is\_locked**  
is locked?  
**Returns** bool

**is\_selected**  
is selected?  
**Returns** bool

**layer**  
primary layer of the item  
**Returns** kicad.pcbnew.Layer



**layers**

All layers where the item is present on

**Returns** `kicad.pcbnew.LayerSet`

**position**

position of the PcbTarget

**Returns** `kicad.util.Point2D`

**width**

Width of line in mm

**Returns** `float`

## 1.13 Text

**class** `kicad.pcbnew.Text` (*text*)

**get\_native()**

Get native object from the low level API

**Returns** `pcbnew.EDA_TEXT`

**is\_highlighted**

is highlighted?

**Returns** `bool`

**is\_locked**

is locked?

**Returns** `bool`

**is\_selected**

is selected?

**Returns** `bool`

**layer**

layer of the drawsegment

**Returns** `kicad.pcbnew.Layer`

**layers**

All layers where the item is present on

**Returns** `kicad.pcbnew.LayerSet`

**position**

Position of the Text

**Returns** `kicad.util.Point2D`

**text**

Text

**Returns** `unicode`

**text\_size**

Text Size

**Returns** `kicad.util.Point2D`

**thickness**  
Thickness  
**Returns** float

## 1.14 Track

**class** kicad.pcbnew.**Track** (*track*)

**end**  
End of the Track  
**Returns** *kicad.util.Point2D*

**get\_native()**  
Get native object from the low level API  
**Returns** pcbnew.TRACK

**is\_highlighted**  
is highlighted?  
**Returns** bool

**is\_locked**  
is locked?  
**Returns** bool

**is\_selected**  
is selected?  
**Returns** bool

**layer**  
primary layer of the item  
**Returns** kicad.pcbnew.Layer

**layers**  
All layers where the item is present on  
**Returns** kicad.pcbnew.LayerSet

**net**  
Net of the Track  
**Returns** *kicad.pcbnew.Net*

**start**  
Start of the Track  
**Returns** *kicad.util.Point2D*

**width**  
Width of Track in mm  
**Returns** float

## 1.15 Via

```
class kicad.pcbnew.Via(via)
```

**drill**

Drill size of Via in mm

**Returns** float

**get\_native()**

Get native object from the low level API

**Returns** pcbnew.VIA

**is\_highlighted**

is highlighted?

**Returns** bool

**is\_locked**

is locked?

**Returns** bool

**is\_selected**

is selected?

**Returns** bool

**layer**

primary layer of the item

**Returns** kicad.pcbnew.Layer

**layers**

All layers where the item is present on

**Returns** kicad.pcbnew.LayerSet

**net**

Net of the Via

**Returns** *kicad.pcbnew.Net*

**position**

Position of the Via

**Returns** *kicad.util.Point2D*

**width**

Width of Via in mm

**Returns** float

## 1.16 Zone

```
class kicad.pcbnew.Zone(zone)
```

**get\_native()**

Get native object from the low level API

**Returns** `pcbnew.ZONE`

**is\_highlighted**

is highlighted?

**Returns** `bool`

**is\_locked**

is locked?

**Returns** `bool`

**is\_selected**

is selected?

**Returns** `bool`

**layer**

primary layer of the item

**Returns** `kicad.pcbnew.Layer`

**layers**

All layers where the item is present on

**Returns** `kicad.pcbnew.LayerSet`

**net**

Net of the Zone

**Returns** `kicad.pcbnew.Net`

**priority**

Priority of the Zone

**Returns** `int`

## 2.1 Polygon

**class** kicad.primitives.Polygon

**class** kicad.primitives.PolygonSet (*poly\_set=None*)

**difference** (*other*)

Performs boolean PolygonSet difference

**Parameters** **other** (*kicad.primitives.PolygonSet*) – second operand of difference operation

**fracture** ()

Converts a set of polygons with holes to a single outline with slits/fractures connecting the outer ring to the inner holes

**get\_native** ()

Get native object from the low level API :return: pcbnew.SHAPE\_POLY\_SET

**intersection** (*other*)

Performs boolean PolygonSet intersection

**Parameters** **other** (*kicad.primitives.PolygonSet*) – second operand of intersection operation

**unfracture** ()

Converts a single outline slitted ('fractured') polygon into a set of outlines with holes

**union** (*other*)

Performs boolean PolygonSet union

**Parameters** **other** (*kicad.primitives.PolygonSet*) – second operand of union operation



## 3.1 Point2D

**class** kicad.util.**Point2D** (*coordinates=None, y=None*)  
Representation of a 2D Point in space

### Example

```
>>> from kicad.util.point import Point2D
>>> Point2D(0, 1)
kicad.util.point.Point2D(0.0, 1.0)
>>> Point2D([2, 3])
kicad.util.point.Point2D(2.0, 3.0)
>>> Point2D((4, 5))
kicad.util.point.Point2D(4.0, 5.0)
>>> Point2D({'x': 6, 'y': 7})
kicad.util.point.Point2D(6.0, 7.0)
>>> Point2D(Point2D(8, 9))
kicad.util.point.Point2D(8.0, 9.0)
```

**static from\_wxPoint** (*wxobj*)  
Convert a wxPoint to a Point2D

**Parameters** *wxobj* (pcbnew.wxPoint) – point to convert

**Returns** *kicad.util.Point2D*

**static from\_wxSize** (*wxobj*)  
Convert a wxSize to a Point2D

**Parameters** *wxobj* (pcbnew.wxSize) – point to convert

**Returns** *kicad.util.Point2D*

**round\_to** (*base, prec=10*)  
Round to a specific base (like it's required for a grid)

### Parameters

- **base** (float) – base we want to round to
- **prec** (int) – precision of rounding operation

**Returns** *kicad.util.Point2D*

### Example

```
>>> from kicad.util.point import Point2D
>>> Point2D(0.1234, 0.5678).round_to(0.01)
kicad.util.point.Point2D(0.12, 0.57)
```

### **to\_wxPoint()**

Convert coordinate to internal coordinate

**Returns** *pcbnew.wxPoint*

### **to\_wxSize()**

Convert size given as Point2D to internal size

**Returns** *pcbnew.wxSize*



**class** kicad.plotter.**Plotter** (*board, layer=None, color\_mode=None*)

**close()**

Close a plotfile after writing

**Example**

```
>>> from kicad.pcbnew import Board, Layer
>>> from kicad.plotter import Plotter
>>> b = Board.from_editor()
>>> p = Plotter(b, layer=Layer.from_id(0))
>>> p.open('test', Plotter.PLOT_FORMAT_SVG)
kicad.plotter.Plotter(board="")
>>> p.plot_layer()
>>> p.close()
```

**color\_mode**

is color mode enabled?

**Returns** bool

**is\_open**

is plotfile open?

**Returns** bool

**layer**

layer to plot on

**Returns** *kicad.pcbnew.Layer*

**open** (*filename, format, sheet\_description=None*)

Open a new plotfile for writing

**Parameters**

- **filename** (str) – Name of the file to plot

- **format** – format of the output file
- **sheet\_description** – some description

Returns *kicad.plotter.Plotter*

#### Example

```
>>> from kicad.pcbnew import Board, Layer
>>> from kicad.plotter import Plotter
>>> b = Board.from_editor()
>>> p = Plotter(b, layer=Layer.from_id(0))
>>> with p.open('test', Plotter.PLOT_FORMAT_SVG):
...     p.plot_layer()
... 
```

**plot\_layer()**

plot layer to opened file

## CHAPTER 5

---

### List PCB Entities

---

This examples is a reimplementation of the `listPcb.py` script found in the official KiCad repository. It basically loads a board and then prints a short representation of all vias, tracks, drawings modules and zones.

```
#!/usr/bin/env python

from __future__ import print_function

import argparse

from kicad.pcbnew import Board
from kicad.pcbnew import Text

def list_pcb(board):
    print()
    print("LIST VIAS:")
    for via in board.vias:
        print(" * Via:   {} - {}/{}".format(via.position, via.drill, via.width))

    print()
    print("LIST TRACKS:")
    for track in board.tracks:
        print(" * Track: {} to {}, width {}".format(track.start, track.end, track.
↪width))

    print()
    print("LIST DRAWINGS:")
    for drawing in board.drawings:
        if type(drawing) is Text:
            print(" * Text:   '{}' at {}".format(drawing.text, drawing.position))
        else:
            print(" * Drawing: {}".format(drawing))

    print()
```

(continues on next page)

(continued from previous page)

```
print("LIST MODULES:")
for module in board.modules:
    print("* Module: {} at {}".format(module.reference, module.position))

print()
print("LIST ZONES:")
for zone in board.zones:
    print("* Zone: '{}' with priority {}".format(zone.net.name, zone.priority))

# reimplementation of pcbnew/python/examples/listPcb.py script using our abstraction_
↪layer
if __name__ == "__main__":
    parser = argparse.ArgumentParser()

    parser.add_argument('board', help='board file to list elements', action='store')

    args = parser.parse_args()

    board = Board.from_file(args.board)

    list_pcb(board)
```

This script can now simply be executed from the commandline, and outputs some nice informations about the board file:

```
$ python ./examples/list_pcb.py ./tests/pcbnew/testproject/testproject.kicad_pcb
```

## CHAPTER 6

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`



### k

- `kicad.pcbnew`, 3
- `kicad.pcbnew.board`, 3
- `kicad.pcbnew.dimension`, 5
- `kicad.pcbnew.drawsegment`, 6
- `kicad.pcbnew.layer`, 9
- `kicad.pcbnew.module`, 10
- `kicad.pcbnew.net`, 11
- `kicad.pcbnew.pad`, 11
- `kicad.pcbnew.pcbtarget`, 12
- `kicad.pcbnew.text`, 13
- `kicad.pcbnew.track`, 14
- `kicad.pcbnew.via`, 15
- `kicad.pcbnew.zone`, 15
- `kicad.plotter`, 21
- `kicad.primitives`, 17
- `kicad.primitives.polygon`, 17
- `kicad.util`, 19
- `kicad.util.point`, 19





## A

angle (kicad.pcbnew.Arc attribute), 6  
Arc (class in kicad.pcbnew), 6  
aux\_origin (kicad.pcbnew.Board attribute), 3

## B

Board (class in kicad.pcbnew), 3

## C

center (kicad.pcbnew.Arc attribute), 6  
center (kicad.pcbnew.Circle attribute), 7  
Circle (class in kicad.pcbnew), 7  
close() (kicad.plotter.Plotter method), 21  
color\_mode (kicad.plotter.Plotter attribute), 21

## D

description (kicad.pcbnew.Module attribute), 10  
diameter (kicad.pcbnew.Circle attribute), 7  
difference() (kicad.primitives.PolygonSet method), 17  
Dimension (class in kicad.pcbnew), 5  
Drawsegment (class in kicad.pcbnew), 6  
drill (kicad.pcbnew.Via attribute), 15

## E

end (kicad.pcbnew.Line attribute), 8  
end (kicad.pcbnew.Track attribute), 14

## F

filepath (kicad.pcbnew.Board attribute), 3  
fracture() (kicad.primitives.PolygonSet method), 17  
from\_editor() (kicad.pcbnew.Board static method), 4  
from\_editor() (kicad.pcbnew.Module static method), 10  
from\_file() (kicad.pcbnew.Board static method), 4  
from\_id() (kicad.pcbnew.Layer static method), 9  
from\_name() (kicad.pcbnew.Layer static method), 10  
from\_wxPoint() (kicad.util.Point2D static method), 19  
from\_wxSize() (kicad.util.Point2D static method), 19

## G

get\_native() (kicad.pcbnew.Arc method), 6  
get\_native() (kicad.pcbnew.Board method), 4  
get\_native() (kicad.pcbnew.Circle method), 7  
get\_native() (kicad.pcbnew.Dimension method), 5  
get\_native() (kicad.pcbnew.Drawsegment method), 6  
get\_native() (kicad.pcbnew.LayerSet method), 10  
get\_native() (kicad.pcbnew.Line method), 8  
get\_native() (kicad.pcbnew.Module method), 10  
get\_native() (kicad.pcbnew.Net method), 11  
get\_native() (kicad.pcbnew.Pad method), 11  
get\_native() (kicad.pcbnew.PcbTarget method), 12  
get\_native() (kicad.pcbnew.Polygon method), 9  
get\_native() (kicad.pcbnew.Text method), 13  
get\_native() (kicad.pcbnew.Track method), 14  
get\_native() (kicad.pcbnew.Via method), 15  
get\_native() (kicad.pcbnew.Zone method), 15  
get\_native() (kicad.primitives.PolygonSet method), 17  
grid\_origin (kicad.pcbnew.Board attribute), 4

## I

id (kicad.pcbnew.Layer attribute), 10  
intersection() (kicad.primitives.PolygonSet method), 17  
is\_highlighted (kicad.pcbnew.Arc attribute), 7  
is\_highlighted (kicad.pcbnew.Board attribute), 4  
is\_highlighted (kicad.pcbnew.Circle attribute), 7  
is\_highlighted (kicad.pcbnew.Dimension attribute), 5  
is\_highlighted (kicad.pcbnew.Drawsegment attribute), 6  
is\_highlighted (kicad.pcbnew.Line attribute), 8  
is\_highlighted (kicad.pcbnew.Module attribute), 10  
is\_highlighted (kicad.pcbnew.Pad attribute), 11  
is\_highlighted (kicad.pcbnew.PcbTarget attribute), 12  
is\_highlighted (kicad.pcbnew.Polygon attribute), 9  
is\_highlighted (kicad.pcbnew.Text attribute), 13  
is\_highlighted (kicad.pcbnew.Track attribute), 14  
is\_highlighted (kicad.pcbnew.Via attribute), 15  
is\_highlighted (kicad.pcbnew.Zone attribute), 16  
is\_locked (kicad.pcbnew.Arc attribute), 7  
is\_locked (kicad.pcbnew.Board attribute), 4

is\_locked (kicad.pcbnew.Circle attribute), 7  
is\_locked (kicad.pcbnew.Dimension attribute), 5  
is\_locked (kicad.pcbnew.Drawsegment attribute), 6  
is\_locked (kicad.pcbnew.Line attribute), 8  
is\_locked (kicad.pcbnew.Module attribute), 10  
is\_locked (kicad.pcbnew.Pad attribute), 12  
is\_locked (kicad.pcbnew.PcbTarget attribute), 12  
is\_locked (kicad.pcbnew.Polygon attribute), 9  
is\_locked (kicad.pcbnew.Text attribute), 13  
is\_locked (kicad.pcbnew.Track attribute), 14  
is\_locked (kicad.pcbnew.Via attribute), 15  
is\_locked (kicad.pcbnew.Zone attribute), 16  
is\_open (kicad.plotter.Plotter attribute), 21  
is\_selected (kicad.pcbnew.Arc attribute), 7  
is\_selected (kicad.pcbnew.Board attribute), 4  
is\_selected (kicad.pcbnew.Circle attribute), 8  
is\_selected (kicad.pcbnew.Dimension attribute), 5  
is\_selected (kicad.pcbnew.Drawsegment attribute), 6  
is\_selected (kicad.pcbnew.Line attribute), 8  
is\_selected (kicad.pcbnew.Module attribute), 10  
is\_selected (kicad.pcbnew.Pad attribute), 12  
is\_selected (kicad.pcbnew.PcbTarget attribute), 12  
is\_selected (kicad.pcbnew.Polygon attribute), 9  
is\_selected (kicad.pcbnew.Text attribute), 13  
is\_selected (kicad.pcbnew.Track attribute), 14  
is\_selected (kicad.pcbnew.Via attribute), 15  
is\_selected (kicad.pcbnew.Zone attribute), 16

## K

keywords (kicad.pcbnew.Module attribute), 10  
kicad.pcbnew (module), 3  
kicad.pcbnew.board (module), 3  
kicad.pcbnew.dimension (module), 5  
kicad.pcbnew.drawsegment (module), 6  
kicad.pcbnew.layer (module), 9  
kicad.pcbnew.module (module), 10  
kicad.pcbnew.net (module), 11  
kicad.pcbnew.pad (module), 11  
kicad.pcbnew.pcbtarget (module), 12  
kicad.pcbnew.text (module), 13  
kicad.pcbnew.track (module), 14  
kicad.pcbnew.via (module), 15  
kicad.pcbnew.zone (module), 15  
kicad.plotter (module), 21  
kicad.primitives (module), 17  
kicad.primitives.polygon (module), 17  
kicad.util (module), 19  
kicad.util.point (module), 19

## L

Layer (class in kicad.pcbnew), 9  
layer (kicad.pcbnew.Arc attribute), 7  
layer (kicad.pcbnew.Board attribute), 4  
layer (kicad.pcbnew.Circle attribute), 8

layer (kicad.pcbnew.Dimension attribute), 5  
layer (kicad.pcbnew.Drawsegment attribute), 6  
layer (kicad.pcbnew.Line attribute), 8  
layer (kicad.pcbnew.Module attribute), 11  
layer (kicad.pcbnew.Pad attribute), 12  
layer (kicad.pcbnew.PcbTarget attribute), 12  
layer (kicad.pcbnew.Polygon attribute), 9  
layer (kicad.pcbnew.Text attribute), 13  
layer (kicad.pcbnew.Track attribute), 14  
layer (kicad.pcbnew.Via attribute), 15  
layer (kicad.pcbnew.Zone attribute), 16  
layer (kicad.plotter.Plotter attribute), 21  
layers (kicad.pcbnew.Arc attribute), 7  
layers (kicad.pcbnew.Board attribute), 4  
layers (kicad.pcbnew.Circle attribute), 8  
layers (kicad.pcbnew.Dimension attribute), 5  
layers (kicad.pcbnew.Drawsegment attribute), 6  
layers (kicad.pcbnew.Line attribute), 8  
layers (kicad.pcbnew.Module attribute), 11  
layers (kicad.pcbnew.Pad attribute), 12  
layers (kicad.pcbnew.PcbTarget attribute), 12  
layers (kicad.pcbnew.Polygon attribute), 9  
layers (kicad.pcbnew.Text attribute), 13  
layers (kicad.pcbnew.Track attribute), 14  
layers (kicad.pcbnew.Via attribute), 15  
layers (kicad.pcbnew.Zone attribute), 16  
LayerSet (class in kicad.pcbnew), 10  
Line (class in kicad.pcbnew), 8

## M

Module (class in kicad.pcbnew), 10  
modules (kicad.pcbnew.Board attribute), 5

## N

name (kicad.pcbnew.Layer attribute), 10  
name (kicad.pcbnew.Net attribute), 11  
name (kicad.pcbnew.Pad attribute), 12  
Net (class in kicad.pcbnew), 11  
net (kicad.pcbnew.Pad attribute), 12  
net (kicad.pcbnew.Track attribute), 14  
net (kicad.pcbnew.Via attribute), 15  
net (kicad.pcbnew.Zone attribute), 16

## O

open() (kicad.plotter.Plotter method), 21

## P

Pad (class in kicad.pcbnew), 11  
pads (kicad.pcbnew.Module attribute), 11  
PcbTarget (class in kicad.pcbnew), 12  
plot\_layer() (kicad.plotter.Plotter method), 22  
Plotter (class in kicad.plotter), 21  
Point2D (class in kicad.util), 19

Polygon (class in kicad.pcbnew), 9  
Polygon (class in kicad.primitives), 17  
PolygonSet (class in kicad.primitives), 17  
position (kicad.pcbnew.Module attribute), 11  
position (kicad.pcbnew.PcbTarget attribute), 13  
position (kicad.pcbnew.Text attribute), 13  
position (kicad.pcbnew.Via attribute), 15  
priority (kicad.pcbnew.Zone attribute), 16

## R

radius (kicad.pcbnew.Circle attribute), 8  
reference (kicad.pcbnew.Module attribute), 11  
round\_to() (kicad.util.Point2D method), 19

## S

start (kicad.pcbnew.Arc attribute), 7  
start (kicad.pcbnew.Line attribute), 9  
start (kicad.pcbnew.Track attribute), 14

## T

Text (class in kicad.pcbnew), 13  
text (kicad.pcbnew.Dimension attribute), 5  
text (kicad.pcbnew.Text attribute), 13  
text\_size (kicad.pcbnew.Text attribute), 13  
thickness (kicad.pcbnew.Text attribute), 13  
to\_wxPoint() (kicad.util.Point2D method), 20  
to\_wxSize() (kicad.util.Point2D method), 20  
Track (class in kicad.pcbnew), 14  
tracks (kicad.pcbnew.Board attribute), 5

## U

unfracture() (kicad.primitives.PolygonSet method), 17  
union() (kicad.primitives.PolygonSet method), 17

## V

value (kicad.pcbnew.Dimension attribute), 5  
value (kicad.pcbnew.Module attribute), 11  
Via (class in kicad.pcbnew), 15  
vias (kicad.pcbnew.Board attribute), 5

## W

width (kicad.pcbnew.Arc attribute), 7  
width (kicad.pcbnew.Circle attribute), 8  
width (kicad.pcbnew.Drawsegment attribute), 6  
width (kicad.pcbnew.Line attribute), 9  
width (kicad.pcbnew.PcbTarget attribute), 13  
width (kicad.pcbnew.Polygon attribute), 9  
width (kicad.pcbnew.Track attribute), 14  
width (kicad.pcbnew.Via attribute), 15

## Z

Zone (class in kicad.pcbnew), 15  
zones (kicad.pcbnew.Board attribute), 5