

Deepin-UI

API Documentation

August 8, 2012

Contents

Contents	1
1 Package dtk.ui	2
1.1 Modules	2
2 Module dtk.ui.animation	4
2.1 Functions	4
2.2 Class Animation	4
2.2.1 Methods	5
3 Module dtk.ui.application	7
3.1 Class Application	7
3.1.1 Methods	7
3.1.2 Properties	9
4 Module dtk.ui.box	10
4.1 Class EventBox	10
4.1.1 Methods	10
4.2 Class ImageBox	10
4.2.1 Methods	10
4.3 Class BackgroundBox	10
4.3.1 Methods	11
5 Module dtk.ui.browser	12
5.1 Class WebView	12
5.1.1 Methods	12
6 Module dtk.ui.button	13
6.1 Class Button	13
6.1.1 Methods	13
6.2 Class ImageButton	13
6.2.1 Methods	14
6.3 Class ThemeButton	14
6.3.1 Methods	14
6.4 Class MenuButton	14
6.4.1 Methods	14
6.5 Class MinButton	15
6.5.1 Methods	15

6.6	Class CloseButton	15
6.6.1	Methods	15
6.7	Class MaxButton	15
6.7.1	Methods	16
6.8	Class ToggleButton	16
6.8.1	Methods	17
6.9	Class ActionButton	17
6.9.1	Methods	18
6.10	Class CheckButton	18
6.10.1	Methods	18
6.11	Class RadioButton	19
6.11.1	Methods	19
6.12	Class DisableButton	19
6.12.1	Methods	19
6.13	Class LinkButton	20
6.13.1	Methods	20
7	Module dtk.ui.cache_pixbuf	21
7.1	Class CachePixbuf	21
7.1.1	Methods	21
7.1.2	Properties	22
8	Module dtk.ui.categorybar	23
8.1	Class Categorybar	23
8.1.1	Methods	23
8.2	Class CategoryItem	24
8.2.1	Methods	24
9	Module dtk.ui.color_selection	25
9.1	Class HSV	25
9.1.1	Methods	25
9.2	Class ColorSelectDialog	25
9.2.1	Methods	26
9.2.2	Class Variables	26
9.3	Class ColorItem	26
9.3.1	Methods	27
9.3.2	Class Variables	29
9.4	Class ColorButton	29
9.4.1	Methods	29
9.4.2	Class Variables	29
10	Module dtk.ui.combo	30
10.1	Class ComboBox	30
10.1.1	Methods	30
10.1.2	Class Variables	31
11	Module dtk.ui.config	32
11.1	Class Config	32
11.1.1	Methods	32
11.1.2	Class Variables	33
12	Module dtk.ui.constant	34

12.1 Variables	34
13 Module dtk.ui.dialog	36
13.1 Variables	36
13.2 Class DialogLeftButtonBox	36
13.2.1 Methods	36
13.3 Class DialogRightButtonBox	36
13.3.1 Methods	37
13.4 Class DialogBox	37
13.4.1 Methods	38
13.5 Class ConfirmDialog	39
13.5.1 Methods	39
13.6 Class InputDialog	40
13.6.1 Methods	40
13.7 Class OpenFileDialog	41
13.7.1 Methods	41
13.8 Class SaveFileDialog	41
13.8.1 Methods	42
14 Module dtk.ui.dominant_color	43
14.1 Functions	43
15 Module dtk.ui.draw	44
15.1 Functions	44
16 Module dtk.ui.droplist	51
16.1 Class DroplistScrolledWindow	51
16.1.1 Methods	51
16.2 Class Droplist	51
16.2.1 Methods	52
16.2.2 Class Variables	54
16.3 Class DroplistItem	55
16.3.1 Methods	55
16.3.2 Properties	56
17 Module dtk.ui.entry	57
17.1 Class Entry	57
17.1.1 Methods	57
17.1.2 Class Variables	59
17.2 Class TextEntry	60
17.2.1 Methods	60
17.2.2 Class Variables	61
17.3 Class InputEntry	61
17.3.1 Methods	62
17.3.2 Class Variables	63
17.4 Class ShortcutKeyEntry	63
17.4.1 Methods	64
17.4.2 Class Variables	65
18 Module dtk.ui.frame	66
18.1 Class HorizontalFrame	66
18.1.1 Methods	66

18.2 Class VerticalFrame	66
18.2.1 Methods	67
19 Module dtk.ui.global_key	68
19.1 Functions	68
19.2 Variables	68
19.3 Class GlobalKey	68
19.3.1 Methods	68
19.3.2 Properties	70
20 Module dtk.ui.group	71
20.1 Class ImageButtonGroup	71
20.1.1 Methods	71
20.2 Class ToggleButtonGroup	71
20.2.1 Methods	71
20.3 Class ImageButtonItem	72
20.3.1 Methods	72
20.4 Class ToggleButtonItem	72
20.4.1 Methods	73
21 Module dtk.ui.iconview	74
21.1 Class IconView	74
21.1.1 Methods	74
21.1.2 Class Variables	76
21.2 Class IconItem	76
21.2.1 Methods	76
21.2.2 Class Variables	78
22 Module dtk.ui.keymap	79
22.1 Functions	79
23 Module dtk.ui.label	81
23.1 Class Label	81
23.1.1 Methods	81
24 Module dtk.ui.line	83
24.1 Class HSeparator	83
24.1.1 Methods	83
24.2 Class VSeparator	83
24.2.1 Methods	83
25 Module dtk.ui.listview	84
25.1 Functions	84
25.2 Class ListView	84
25.2.1 Methods	85
25.2.2 Class Variables	91
25.3 Class ListItem	92
25.3.1 Methods	92
25.3.2 Class Variables	94
26 Module dtk.ui.listview_preview_pixbuf	95
26.1 Functions	95

27 Module dtk.ui.locales	96
27.1 Variables	96
28 Module dtk.ui.mask	97
28.1 Functions	97
29 Module dtk.ui.menu	99
29.1 Class Menu	99
29.1.1 Methods	99
29.2 Class MenuItem	100
29.2.1 Methods	101
29.2.2 Properties	101
30 Module dtk.ui.mplayer_view	103
30.1 Class MplayerView	103
30.1.1 Methods	103
30.1.2 Class Variables	103
31 Module dtk.ui.mplayer_window	104
31.1 Class MplayerWindow	104
31.1.1 Methods	104
32 Module dtk.ui.navigatebar	107
32.1 Class Navigatebar	107
32.1.1 Methods	107
32.2 Class NavItem	108
32.2.1 Methods	108
32.2.2 Properties	109
33 Module dtk.ui.new_treeview	110
33.1 Class TreeView	110
33.1.1 Methods	110
33.2 Class TreeItem	110
33.2.1 Methods	111
34 Module dtk.ui.notebook	112
34.1 Class Notebook	112
34.1.1 Methods	112
35 Module dtk.ui.osd_tooltip	114
35.1 Class OSDTooltip	114
35.1.1 Methods	114
36 Module dtk.ui.paned	116
36.1 Class Paned	116
36.1.1 Methods	116
36.2 Class HPaned	116
36.2.1 Methods	116
36.3 Class VPaned	116
36.3.1 Methods	117
37 Module dtk.ui.panel	118
37.1 Class Panel	118

37.1.1 Methods	118
38 Module dtk.ui.progressbar	119
38.1 Class ProgressBar	119
38.1.1 Methods	119
39 Module dtk.ui.pseudo_skin	120
39.1 Variables	120
40 Module dtk.ui.scalebar	121
40.1 Class HScalebar	121
40.1.1 Methods	121
40.2 Class VScalebar	121
40.2.1 Methods	122
41 Module dtk.ui.scrolled_window	123
41.1 Class ScrolledWindow	123
41.1.1 Methods	123
42 Module dtk.ui.skin	125
42.1 Class SkinWindow	125
42.1.1 Methods	125
43 Module dtk.ui.skin_config	126
43.1 Variables	126
43.2 Class SkinConfig	126
43.2.1 Methods	126
44 Module dtk.ui.slider	128
44.1 Class Slider	128
44.1.1 Methods	128
44.1.2 Class Variables	129
44.2 Class Wizard	129
44.2.1 Methods	130
45 Module dtk.ui.spin	131
45.1 Class SpinBox	131
45.1.1 Methods	131
45.1.2 Class Variables	132
46 Module dtk.ui.statusbar	133
46.1 Class StatusBar	133
46.1.1 Methods	133
47 Module dtk.ui.tab_window	134
47.1 Class TabBox	134
47.1.1 Methods	134
47.2 Class TabWindow	134
47.2.1 Methods	135
48 Module dtk.ui.talk_view	136
48.1 Class TalkView	136
48.1.1 Methods	136

48.2 Class TalkItem	136
48.2.1 Methods	136
49 Module dtk.ui.theme	138
49.1 Variables	138
49.2 Class DynamicColor	138
49.2.1 Methods	138
49.2.2 Properties	139
49.3 Class DynamicAlphaColor	139
49.3.1 Methods	139
49.3.2 Properties	140
49.4 Class DynamicShadowColor	140
49.4.1 Methods	140
49.4.2 Properties	141
49.5 Class DynamicPixbuf	141
49.5.1 Methods	142
49.5.2 Properties	142
49.6 Class Theme	142
49.6.1 Methods	143
49.6.2 Properties	144
50 Module dtk.ui.thread_pool	145
50.1 Functions	145
50.2 Class MissionThreadPool	145
50.2.1 Methods	145
50.2.2 Properties	146
50.2.3 Class Variables	146
50.3 Class MissionThread	147
50.3.1 Methods	147
50.3.2 Properties	148
50.4 Class TestMissionThread	148
50.4.1 Methods	148
50.4.2 Properties	149
51 Module dtk.ui.threads	150
51.1 Functions	150
51.2 Class AnonymityThread	150
51.2.1 Methods	150
51.2.2 Properties	151
52 Module dtk.ui.timeline	152
52.1 Variables	152
52.2 Class Timeline	152
52.2.1 Methods	152
52.2.2 Class Variables	152
53 Module dtk.ui.titlebar	154
53.1 Class Titlebar	154
53.1.1 Methods	154
54 Module dtk.ui.tooltip	156
54.1 Functions	156

55	Module dtk.ui.tooltip_test	159
56	Module dtk.ui.treeview	160
56.1	Class TreeView	160
56.1.1	Methods	160
56.1.2	Class Variables	162
56.2	Class Tree	162
56.2.1	Methods	162
56.2.2	Properties	162
56.3	Class TreeViewItem	163
56.3.1	Methods	163
56.3.2	Properties	163
57	Module dtk.ui.unique_service	164
57.1	Functions	164
57.2	Class UniqueService	164
57.2.1	Methods	164
57.2.2	Class Variables	165
58	Module dtk.ui.utils	166
58.1	Functions	166
59	Module dtk.ui.volume_button	185
59.1	Variables	185
59.2	Class VolumeButton	185
59.2.1	Methods	186
59.2.2	Class Variables	187
59.2.3	Instance Variables	187
60	Module dtk.ui.window	188
60.1	Class Window	188
60.1.1	Methods	188

1 Package dtk.ui

1.1 Modules

- **animation** (Section 2, p. 4)
- **application** (Section 3, p. 7)
- **box** (Section 4, p. 10)
- **browser** (Section 5, p. 12)
- **button** (Section 6, p. 13)
- **cache_pixbuf** (Section 7, p. 21)
- **categorybar** (Section 8, p. 23)
- **color_selection** (Section 9, p. 25)
- **combo** (Section 10, p. 30)
- **config** (Section 11, p. 32)
- **constant** (Section 12, p. 34)
- **dialog** (Section 13, p. 36)
- **dominant_color** (Section 14, p. 43)
- **draw** (Section 15, p. 44)
- **droplist** (Section 16, p. 51)
- **entry** (Section 17, p. 57)
- **frame** (Section 18, p. 66)
- **global_key** (Section 19, p. 68)
- **group** (Section 20, p. 71)
- **iconview** (Section 21, p. 74)
- **keymap** (Section 22, p. 79)
- **label** (Section 23, p. 81)
- **line** (Section 24, p. 83)
- **listview** (Section 25, p. 84)
- **listview_preview_pixbuf** (Section 26, p. 95)
- **locales** (Section 27, p. 96)
- **mask** (Section 28, p. 97)
- **menu** (Section 29, p. 99)
- **mplayer_view** (Section 30, p. 103)
- **mplayer_window** (Section 31, p. 104)
- **navigatebar** (Section 32, p. 107)
- **new_treeview** (Section 33, p. 110)
- **notebook** (Section 34, p. 112)
- **osd_tooltip** (Section 35, p. 114)
- **paned** (Section 36, p. 116)
- **panel** (Section 37, p. 118)
- **progressbar** (Section 38, p. 119)
- **pseudo_skin** (Section 39, p. 120)
- **scalebar** (Section 40, p. 121)
- **scrolled_window** (Section 41, p. 123)
- **skin** (Section 42, p. 125)
- **skin_config** (Section 43, p. 126)
- **slider** (Section 44, p. 128)
- **spin** (Section 45, p. 131)
- **statusbar** (Section 46, p. 133)
- **tab_window** (Section 47, p. 134)
- **talk_view** (Section 48, p. 136)

- **theme** (*Section 49, p. 138*)
- **thread_pool** (*Section 50, p. 145*)
- **threads** (*Section 51, p. 150*)
- **timeline** (*Section 52, p. 152*)
- **titlebar** (*Section 53, p. 154*)
- **tooltip** (*Section 54, p. 156*)
- **tooltip_test** (*Section 55, p. 159*)
- **treeview** (*Section 56, p. 160*)
- **unique_service** (*Section 57, p. 164*)
- **utils** (*Section 58, p. 166*)
- **volume_button** (*Section 59, p. 185*)
- **window** (*Section 60, p. 188*)

2 Module *dtk.ui.animation*

2.1 Functions

LinerInterpolator(*factor, lower, upper*)

Linear interpolator

Parameters

factor: the current factor
lower: the init lower value
upper: the init upper value

Return Value

the calculated value

RandomInterpolator(*base, offset, *args*)

Random interpolator

Parameters

base: the base value used to calculate result value
offset: the offset apply to base.

Return Value

the random value based on 'base' and 'offset'

2.2 Class Animation

The animation class used to convenient production special effects.

2.2.1 Methods

`__init__`(*self*, *widgets*, *property*, *duration*, *ranges*, *interpolator=LinerInterpolator*, *stop_callback=None*)

Initialize Animation class.

Parameters

- widgets:** the widgets apply to this animation. the type of this param is an `gtk.Widget` or an list of `gtk.Widget`.
- property:** the `gtk.Widget`'s property used to do effect or an function to change the actual effect.
- duration:** the time of this effect to continued, the unit of time is millisecond
- ranges:** the range of the property's value. the type of this param is an `[lower,upper]` or `([lower, upper], [lower,upper])`, this is decision by the parameter of the 'widget' or 'widgets'.
- interpolator:** this is an function used to calculate the property value by the current time and value range.
- stop_callback:** the callback when this animation stop.

`set_delay`(*self*, *delay*)

Set the delay time of before the start do effect.

Parameters

- delay:** the time of dealy, unit of time is millisecond

`start_after`(*self*, *time*)

Start the animation after the dealy time. or you can use `Animation.set_delay` function.

Parameters

- time:** the time of dealy, unit of time is millisecond

`start`(*self*)

Start the animation object.

`stop`(*self*)

stop immediatly the animation object

`mul`____(*self*, *other*)

Overload the '*' operator to link two or more animation object. the animation's effect is happend parallel.

Parameters

- other:** the right hand side animation class.

Return Value

the new animation class with the two operator animation's effect.

<code>__add__(self, other)</code>

3 Module *gtk.ui.application*

3.1 Class Application

object 
gtk.ui.application.Application

This is the base class of every program based on *deepin-ui*. Every program should realize it.

3.1.1 Methods

<p>__init__(<i>self</i>, <i>app_support_colormap</i>=True)</p> <hr/> <p>Initialize the Application class.</p> <p>Parameters</p> <p>app_support_colormap: Set False if your program don't allow manipulate colormap, such as <i>mplayer</i>, otherwise you should keep this option as True.</p> <p>Overrides: object.__init__</p>
--

<p>init(<i>self</i>)</p> <hr/> <p>This do the remain initialize step.</p> <p>It Initializes the window and some important signal such as "destroy".</p>
--

<p>add_titlebar(<i>self</i>, <i>button_mask</i>=["theme", "menu", "max", "min", "close"], <i>icon_dpixbuf</i>=None, <i>app_name</i>=None, <i>title</i>=None, <i>add_separator</i>=False, <i>show_title</i>=True)</p> <hr/> <p>Add titlebar to the application.</p> <p>Connect click signal of the standard button to default callback.</p> <p>Parameters</p> <p>button_mask: A list of string, each of which stands for a standard button on top right of the window. By default, it's ["theme", "menu", "max", "min", "close"].</p> <p>icon_dpixbuf: The icon pixbuf of type <i>gtk.ui.theme.DynamicPixbuf</i>. By default, it is None.</p> <p>app_name: The name string of the application, which will be displayed just next to the <i>icon_dpixbuf</i>. By default, it is None.</p> <p>title: The title string of the window, which will be displayed on the center of the titlebar. By default, it is None.</p> <p>add_separator: If True, add a line between the titlebar and the body of the window. By default, it's False.</p> <p>show_title: If False, the titlebar will not be displayed. By default, it's True.</p>

close_window(*self*, *widget*)

Close the window when the close button is clicked.

Parameters

widget: A widget of Gtk.Widget. Passed by gtk.

show_titlebar(*self*)

Show title bar of the window.

By default, it is invoked at the last step of add_titlebar.

hide_titlebar(*self*)

Hide the title bar.

raise_to_top(*self*)

Raise the window to the top of the window stack.

set_title(*self*, *title*)

Set the application title.

Parameters

title: The title string of the application.

set_default_size(*self*, *default_width*, *default_height*)

Set the default size of the window.

Parameters

default_width: Default width in pixels of the application, once set, application don't allow smaller than width.

default_height: Default height in pixels of the application, once set, application don't allow smaller than height.

set_icon(*self*, *icon_dpixbuf*)

Set the icon of the application.

This icon is used by the window manager or the dock.

Parameters

icon_dpixbuf: The icon pixbuf of dtk.ui.theme.DynamicPixbuf.

destroy(*self*, *widget*, *data=None*)

Destroy the window and quit the program.

Parameters

widget: Not used.

data: Not used.

run(*self*)

Show the window and start the mainloop.

You must use this function at last of program, otherwise program will run in loop too early that all code after application.run won't execute until program exit.

set_skin_preview(*self*, *preview_pixbuf*)

Set the skin preview of the application.

Parameters

preview_pixbuf: A pixbuf of type dtk.ui.theme.DynamicPixbuf.

Note: The size of preview_pixbuf must be proportional to the size of program, otherwise adjust skin will got wrong coordinate.

theme_callback(*self*, *widget*)

Invoked when the theme button is clicked.

Parameters

widget: Not used.

Return Value

Always return False

menu_callback(*self*, *widget*)

Invoked when the menu button is clicked.

Parameters

widget: Not used.

Return Value

Always return False

set_menu_callback(*self*, *callback*)

Set the menu_button_callback function.

Parameters

callback: A function which is invoked when the menu button is clicked.

Inherited from object

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`,
`__reduce__()`, `__reduce_ex__()`, `__repr__()`, `__setattr__()`, `__sizeof__()`,
`__str__()`, `__subclasshook__()`

3.1.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

4 Module `gtk.ui.box`

4.1 Class `EventBox`



Event box, not like `Gtk.EventBox`, it don't show visible window default.

4.1.1 Methods

<code>__init__</code> (<i>self</i>) <hr/> Initialize the <code>EventBox</code> class.

4.2 Class `ImageBox`



`ImageBox`.

4.2.1 Methods

<code>__init__</code> (<i>self</i> , <i>image_dpixbuf</i>) <hr/> Initialize the <code>ImageBox</code> class. Parameters <i>image_dpixbuf</i> : Image dynamic pixbuf.

4.3 Class `BackgroundBox`



`BackgroundBox` is container for clip background.

4.3.1 Methods

<code>__init__</code> (<i>self</i>)
Initialize the <code>BackgroundBox</code> class.
<code>draw_mask</code> (<i>self</i> , <i>cr</i> , <i>x</i> , <i>y</i> , <i>w</i> , <i>h</i>)
Mask render function.
Parameters
<i>cr</i> : Cairo context.
<i>x</i> : X coordinate of draw area.
<i>y</i> : Y coordinate of draw area.
<i>w</i> : Width of draw area.
<i>h</i> : Height of draw area.

5 Module `dtk.ui.browser`

5.1 Class `WebView`

`webkit.WebView` —
`dtk.ui.browser.WebView`

5.1.1 Methods

<code>__init__(self, cookie_filepath=None)</code>
Init for <code>WebView</code> .
Parameters
<code>cookie_filepath</code> : Filepath to save cookie.

6 Module `gtk.ui.button`

6.1 Class `Button`



Button with Deepin UI style.

6.1.1 Methods

<code>__init__</code> (<i>self</i> , <i>label</i> , <i>font_size</i> =DEFAULT_FONT_SIZE)
Initialize Button class.
Parameters
<i>label</i> : Button label.
<i>font_size</i> : Button label font size.

<code>set_label</code> (<i>self</i> , <i>label</i> , <i>font_size</i> =DEFAULT_FONT_SIZE)
Set label of Button.
Parameters
<i>label</i> : Button label.
<i>font_size</i> : Button label font size.

6.2 Class `ImageButton`



`ImageButton` class.

6.2.1 Methods

<code>__init__</code> (<i>self</i> , <i>normal_dpixbuf</i> , <i>hover_dpixbuf</i> , <i>press_dpixbuf</i> , <i>scale_x</i> =False, <i>content</i> =None)
--

Initialize ImageButton class.

Parameters

<code>normal_dpixbuf</code> :	DynamicPixbuf for button normal status.
<code>hover_dpixbuf</code> :	DynamicPixbuf for button hover status.
<code>press_dpixbuf</code> :	DynamicPixbuf for button press status.
<code>scale_x</code> :	Whether scale horticulturally, default is False.
<code>content</code> :	Button label content.

6.3 Class ThemeButton

gtk.Button —
dtk.ui.button.ThemeButton

ThemeButton class.

6.3.1 Methods

<code>__init__</code> (<i>self</i>)
--

Initialize ThemeButton class.

6.4 Class MenuButton

gtk.Button —
dtk.ui.button.MenuButton

MenuButton class.

6.4.1 Methods

<code>__init__</code> (<i>self</i>)
--

Initialize MenuButton class.

6.5 Class MinButton

gtk.Button —
 dtk.ui.button.MinButton

MinButton.

6.5.1 Methods

<code>__init__</code> (<i>self</i>)
Initialize MinButton class.

6.6 Class CloseButton

gtk.Button —
 dtk.ui.button.CloseButton

CloseButton class.

6.6.1 Methods

<code>__init__</code> (<i>self</i>)
Initialize CloseButton class.

6.7 Class MaxButton

gtk.Button —
 dtk.ui.button.MaxButton

MaxButton class.

6.7.1 Methods

```
__init__(self, sub_dir="button", max_path_prefix="window_max",  
unmax_path_prefix="window_unmax")
```

Initialize `MaxButton` class.

Parameters

`sub_dir`: Subdirectory of button images.
`max_path_prefix`: Image path prefix for maximise status.
`unmax_path_prefix`: Image path prefix for un-maximise status.

6.8 Class `ToggleButton`

`gtk.ToggleButton` — `gtk.ui.button.ToggleButton`

`ToggleButton` class.

6.8.1 Methods

<pre>__init__(self, inactive_normal_dpixbuf, active_normal_dpixbuf, inactive_hover_dpixbuf=None, active_hover_dpixbuf=None, inactive_press_dpixbuf=None, active_press_dpixbuf=None, inactive_disable_dpixbuf=None, active_disable_dpixbuf=None, button_label=None, padding_x=0)</pre>	
Initialize <code>ToggleButton</code> class.	
Parameters	
<code>inactive_normal_dpixbuf</code> :	DynamicPixbuf for inactive normal status.
<code>active_normal_dpixbuf</code> :	DynamicPixbuf for active normal status.
<code>inactive_hover_dpixbuf</code> :	DynamicPixbuf for inactive hover status, default is None.
<code>active_hover_dpixbuf</code> :	DynamicPixbuf for active hover status, default is None.
<code>inactive_press_dpixbuf</code> :	DynamicPixbuf for inactive press status, default is None.
<code>active_press_dpixbuf</code> :	DynamicPixbuf for active press status, default is None.
<code>inactive_disable_dpixbuf</code> :	DynamicPixbuf for inactive disable status, default is None.
<code>active_disable_dpixbuf</code> :	DynamicPixbuf for active disable status, default is None.
<code>button_label</code> :	Button label, default is None.
<code>padding_x</code> :	Padding x, default is 0.

6.9 Class `ActionButton`



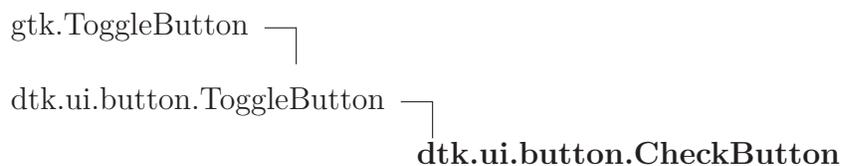
`ActionButton` class.

6.9.1 Methods

<code>__init__</code> (<i>self</i> , <i>actions</i> , <i>index</i> =0)
Initialize for <code>ActionButton</code> class.
Parameters
<i>actions</i> : Actions for button.
<i>index</i> : Index default is 0.

<code>update_action_index</code> (<i>self</i> , <i>widget</i>)
Update action index of <code>ActionButton</code> .
Parameters
<i>widget</i> : <code>ActionButton</code> widget.

6.10 Class `CheckButton`

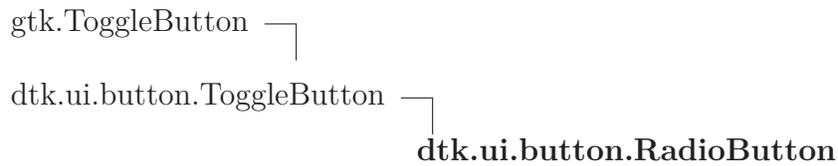


`CheckButton` class.

6.10.1 Methods

<code>__init__</code> (<i>self</i> , <i>label_text</i> =None, <i>padding_x</i> =8)
Initialize <code>CheckButton</code> class.
Parameters
<i>label_text</i> : Label text.
<i>padding_x</i> : Horticultural padding value, default is 8.
Overrides: <code>dtk.ui.button.ToggleButton.__init__</code>

6.11 Class `RadioButton`



`RadioButton` class.

6.11.1 Methods

<pre><code>__init__(self, label_text=None, padding_x=8)</code></pre>
Initialize <code>RadioButton</code> class.
Parameters
<code>label_text</code> : Label text.
<code>padding_x</code> : Horticultural padding value, default is 8.
Overrides: <code>dtk.ui.button.ToggleButton.__init__</code>

6.12 Class `DisableButton`



`DisableButton` class.

6.12.1 Methods

<pre><code>__init__(self, dpixbufs)</code></pre>
Initialize <code>DisableButton</code> class.
Parameters
<code>dpixbufs</code> : DyanmicPixbuf.

6.13 Class `LinkButton`



`LinkButton` click to open browser.

6.13.1 Methods

```
__init__(self, text, link, enable_gaussian=True,
          text_color=ui_theme.get_color("link_text"))
```

Initialize `LinkButton` class.

Parameters

`text`: Link content.

`link`: Link address.

`enable_gaussian`: To enable gaussian effect on link, default is `True`.

`text_color`: Link color, just use when option `enable_gaussian` is `False`.

Overrides: `dtk.ui.label.Label.__init__`

Inherited from `dtk.ui.label.Label` (Section 23.1)

`copy_to_clipboard()`, `get_text()`, `select_all()`, `set_text()`

7 Module `gtk.gdk.pixbuf`

7.1 Class `CachePixbuf`

object —
 `gtk.gdk.pixbuf.CachePixbuf`

Cache pixbuf use to cache pixbuf to avoid new pixbuf generate by `scale_simple`.

`gtk.gdk.pixbuf.scale_simple` is function will make application very slow,

We use `CachePixbuf` increase the call times of `gtk.gdk.pixbuf.scale_simple`.

7.1.1 Methods

<code>__init__(self)</code> <hr/> Init cache pixbuf. Overrides: <code>object.__init__</code>

<code>scale(self, pixbuf, scale_width, scale_height, vertical_mirror=False, horizontal_mirror=False)</code> <hr/> Scale with given size and return new pixbuf. Parameters <code>pixbuf</code> : Original pixbuf. <code>scale_width</code> : Scale width of pixbuf. <code>scale_height</code> : Scale height of pixbuf. <code>vertical_mirror</code> : Whether pixbuf mirror vertically. <code>horizontal_mirror</code> : Whether pixbuf mirror horizontally.

<code>get_cache(self)</code> <hr/> Get pixbuf cache. Return Value Return cache pixbuf.

Inherited from object

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`,
`__reduce__()`, `__reduce_ex__()`, `__repr__()`, `__setattr__()`, `__sizeof__()`,
`__str__()`, `__subclasshook__()`

7.1.2 Properties

Name	Description
<i>Inherited from object</i> __class__	

8 Module `dtk.ui.categorybar`

8.1 Class `Categorybar`



`Categorybar`.

8.1.1 Methods

`__init__`(*self*, *items*, *font_size*=DEFAULT_FONT_SIZE, *padding_left*=20, *padding_middle*=10, *padding_right*=25)

Initialize `Categorybar` class.

Parameters

`items`: A list of category item, format: (`icon_dpixbuf`, `content`, `click_callback`)

Overrides: `dtk.ui.box.EventBox.__init__`

`set_index`(*self*, *index*)

Set selected item index.

Parameters

`index`: Index of selected item.

`get_index`(*self*)

Get selected index.

Return Value

Return selected item index.

`get_icon_width`(*self*, *items*)

Get icon width.

Parameters

`items`: A list of category item, format: (`icon_dpixbuf`, `content`, `click_callback`)

8.2 Class *CategoryItem*

gtk.Button —
 dtk.ui.categorybar.CategoryItem

CategoryItem class for use in *CategoryBar*.

8.2.1 Methods

```
__init__(self, item, index, font_size, icon_width, padding_left,  
padding_middle, padding_right, set_index, get_index)
```

Initialize *CategoryItem* class.

Parameters

item:	Category item, format: (<i>item_dpixbuf</i> , <i>content</i> , <i>click_callback</i>)
index:	Category item index.
font_size:	Font size.
icon_width:	Icon width.
padding_left:	Padding at left of item.
padding_middle:	Padding between icon and font.
padding_right:	Padding at right of item.
set_index:	Set index callback.
get_index:	Get index callback.

9 Module `gtk.ui.color_selection`

9.1 Class `HSV`

`gtk.ColorSelection` └─
 `gtk.ui.color_selection.HSV`

HSV widget that use `deepin-ui` widget instead `Gtk.ColorSelection`'s child widget.

9.1.1 Methods

<code>__init__(self)</code>
Initialize HSV class.

<code>get_hsv_widget(self)</code>
Get hsv widget in <code>Gtk.ColorSelection</code> widget.

<code>get_color_string(self)</code>
Get color string.

<code>get_rgb_color(self)</code>
Get RGB color.

9.2 Class `ColorSelectDialog`

`gtk.Window` └─
`gtk.ui.window.Window` └─
 `gtk.ui.dialog.DialogBox` └─
 `gtk.ui.color_selection.ColorSelectDialog`

`ColorSelectDialog` widget.

9.2.1 Methods

__init__ (<i>self</i> , <i>confirm_callback</i> =None, <i>cancel_callback</i> =None)
Initialize ColorSelectDialog class.
Parameters
<i>confirm_callback</i> : Callback when user click OK, this callback accept one argument, color string.
<i>cancel_callback</i> : Callback when user click cancel, this callback don't accept any argument.
Overrides: <i>dtk.ui.window.Window.__init__</i>

update_color_info (<i>self</i> , <i>color_string</i> , <i>clear_highlight</i> =True)
Update color information.
Parameters
<i>color_string</i> : Hex color string.
<i>clear_highlight</i> : Whether clear color select view's highlight status, default is True.

Inherited from dtk.ui.dialog.DialogBox(Section 13.4)

`get_mask_func()`

Inherited from dtk.ui.window.Window(Section 60.1)

`add_move_event()`, `add_toggle_event()`, `close_window()`, `draw_mask()`, `get_edge()`, `get_shadow_size()`, `hide_shadow()`, `is_disable_window_maximized()`, `min_window()`, `resize_window()`, `show_shadow()`, `show_window()`, `toggle_fullscreen_window()`, `toggle_max_window()`

9.2.2 Class Variables

Name	Description
DEFAULT_COLOR_LIST	Value: ["#000000", "#808080", "#E20417", "#F29300", "#FFEC00", "..."]

9.3 Class *ColorItem*



ColorItem class for use in *ColorSelectDialog*.

9.3.1 Methods

<code>__init__(self, color)</code>
Initialize <i>ColorItem</i> class.
Parameters
color : Hex color string.

<code>emit_redraw_request(self)</code>
IconView interface function.
Emit 'redraw-request' signal.

<code>get_width(self)</code>
IconView interface function.
Get item width.
Return Value
Return item width, in pixel.

<code>get_height(self)</code>
IconView interface function.
Get item height.
Return Value
Return item height, in pixel.

<code>render(self, cr, rect)</code>
IconView interface function.
Render item.
Parameters
cr : Cairo context.
rect : Render rectangle area.

icon_item_motion_notify(*self*, *x*, *y*)

IconView interface function.

Handle 'motion-notify-event' signal.

Parameters**x**: X coordinate that user motion on item.**y**: Y coordinate that user motion on item.**icon_item_lost_focus**(*self*)

IconView interface function.

Handle 'lost-focus' signal.

icon_item_highlight(*self*)

IconView interface function.

Handle 'highlight' signal.

icon_item_normal(*self*)

Normal icon item.

icon_item_button_press(*self*, *x*, *y*)

IconView interface function.

Handle 'button-press' signal.

icon_item_button_release(*self*, *x*, *y*)

IconView interface function.

Handle 'button-release' signal.

icon_item_single_click(*self*, *x*, *y*)

IconView interface function.

Handle 'click' signal.

icon_item_double_click(*self*, *x*, *y*)

IconView interface function.

Handle 'double-click' signal.

10 Module *gtk.ui.combo*

10.1 Class *gtk.ui.combo.ComboBox*



gtk.ui.combo.ComboBox class.

10.1.1 Methods

<code>__init__</code> (<i>self</i> , <i>items</i> , <i>droplist_height</i> =None, <i>select_index</i> =0, <i>max_width</i> =None)	
Initialize <i>gtk.ui.combo.ComboBox</i> class.	
Parameters	
<code>items</code> :	ComboBox item, item format: (<i>item_label</i> , <i>item_value</i>)
<code>droplist_height</code> :	You can set maximum height of droplist, default is None.
<code>select_index</code> :	Initialize selected index, default is 0.
<code>max_width</code> :	Maximum width of <i>gtk.ui.combo.ComboBox</i> , default is None that width along with content.

<code>select_first_item</code> (<i>self</i>)	
Select first item.	

<code>select_last_item</code> (<i>self</i>)	
Select last item.	

<code>select_prev_item</code> (<i>self</i>)	
Select preview item.	

<code>select_next_item</code> (<i>self</i>)	
Select next item.	

set_select_index (<i>self</i> , <i>item_index</i>)
Set select index.
Parameters
<i>item_index</i> : The index of selected item.

get_item_with_index (<i>self</i> , <i>item_index</i>)
Get item with given index.
Return Value
Return item that match given index, or return None if haven't special index.

get_current_item (<i>self</i>)
Get current item.
Return Value
Return current item.

10.1.2 Class Variables

Name	Description
<code>__gsignals__</code>	Value: {"item-selected": (gobject.SIGNAL_RUN_LAST, gobject.TYPE_N...

11 Module *dtk.ui.config*

11.1 Class Config

gobject.GObject —
dtk.ui.config.Config

Config module to read *.ini file.

11.1.1 Methods

<p>__init__(<i>self</i>, <i>config_file</i>, <i>default_config</i>=None)</p> <hr/> <p>Init config module.</p> <p>Parameters</p> <p><i>config_file</i>: Config filepath.</p> <p><i>default_config</i>: Default config value use when config file is empty.</p>
<p>load_default(<i>self</i>)</p> <hr/> <p>Load config items with default setting.</p>
<p>load(<i>self</i>)</p> <hr/> <p>Load config items from the file.</p>
<p>get(<i>self</i>, <i>section</i>, <i>option</i>, <i>default</i>=None)</p> <hr/> <p>Get specified the section for read the option value.</p> <p>Parameters</p> <p><i>section</i>: Section to index item.</p> <p><i>option</i>: Option to index item.</p> <p><i>default</i>: Default value if item is not exist.</p> <p>Return Value</p> <p>Return item value with match in config file.</p>

set (<i>self</i> , <i>section</i> , <i>option</i> , <i>value</i>)
Set item given value.
Parameters
<i>section</i> : Section to setting.
<i>option</i> : Option to setting.
<i>value</i> : Item value to save.

write (<i>self</i> , <i>given_filepath</i> =None)
Save configure to file.
Parameters
<i>given_filepath</i> : If <i>given_filepath</i> is None, save to default filepath, otherwise save to given filepath.

get_default (<i>self</i>)
Get default config value.
Return Value
Return default config value.

set_default (<i>self</i> , <i>default_config</i>)
Set default config value and load it.
Parameters
<i>default_config</i> : Default config value.

convert_from_list (<i>self</i> , <i>config_list</i>)
Convert to dict from list format.
Parameters
<i>config_list</i> : Config value as List format.
Return Value
Return config value as Dict format.

11.1.2 Class Variables

Name	Description
<code>__gsignals__</code>	Value: { <code>"config-changed"</code> : (<code>gobject.SIGNAL_RUN_LAST</code> , <code>gobject.TYPE_...</code>)

12 Module `gtk.ui.constant`

12.1 Variables

Name	Description
<code>MENU_ITEM_RADIUS</code>	Value: 2
<code>DEFAULT_WINDOW_WIDTH</code>	Value: 890
<code>DEFAULT_WINDOW_HEIGHT</code>	Value: 629
<code>EDGE_DICT</code>	Value: { <code>gtk.gdk.TOP_LEFT_CORNER</code> : <code>gtk.gdk.WINDOW_EDGE_NORTH_WEST</code> , ...}
<code>WIDGET_POS_TOP_LEFT</code>	Value: 0
<code>WIDGET_POS_TOP_RIGHT</code>	Value: 1
<code>WIDGET_POS_TOP_CENTER</code>	Value: 2
<code>WIDGET_POS_BOTTOM_LEFT</code>	Value: 3
<code>WIDGET_POS_BOTTOM_RIGHT</code>	Value: 4
<code>WIDGET_POS_BOTTOM_CENTER</code>	Value: 5
<code>WIDGET_POS_LEFT_CENTER</code>	Value: 6
<code>WIDGET_POS_RIGHT_CENTER</code>	Value: 7
<code>WIDGET_POS_CENTER</code>	Value: 8
<code>font_test_window</code>	Value: <code>gtk.Window(gtk.WINDOW_POPUP)</code>
<code>DEFAULT_FONT</code>	Value: <code>'</code> <code>'.join(str(font_test_window.get_pango_context().get_fon...</code>
<code>ALIGN_START</code>	Value: <code>pango.ALIGN_LEFT</code>
<code>ALIGN_MIDDLE</code>	Value: <code>pango.ALIGN_CENTER</code>
<code>ALIGN_END</code>	Value: <code>pango.ALIGN_RIGHT</code>
<code>BUTTON_NORMAL</code>	Value: 0
<code>BUTTON_PRESS</code>	Value: 1
<code>BUTTON_HOVER</code>	Value: 2
<code>DEFAULT_FONT_SIZE</code>	Value: 9
<code>SHADE_SIZE</code>	Value: 200
<code>COLOR_NAME_DICT</code>	Value: {"dark_grey": "#333333", "red": "#FF0000", "orange": "#FF...

continued on next page

Name	Description
BLACK_COLOR_MAPPED	Value: "dark_grey"
WHITE_COLOR_MAPPED	Value: "dodger_blue"
COLOR_SEQUENCE	Value: ["red", "orange", "gold", "yellow", "green_yellow", "char...
SIMILAR_COLOR_SEQUENCE	Value: ["red", "orange", "gold", "yellow", "green_yellow", "char...

13 Module dtk.ui.dialog

13.1 Variables

Name	Description
DIALOG_MASK_SINGLE_PAGE	Value: 0
DIALOG_MASK_GLASS_PAGE	Value: 1
DIALOG_MASK_MULTIPLE_PAGE	Value: 2
DIALOG_MASK_TAB_PAGE	Value: 3

13.2 Class DialogLeftButtonBox



HBox to handle left side buttons in DialogBox.

13.2.1 Methods

__init__ (<i>self</i>)
Initialize DialogLeftButtonBox class.

set_buttons (<i>self</i> , <i>buttons</i>)
Set buttons in box.
Parameters
<i>buttons</i> : A list of Gtk.Widget instance.
Note: This functin will use new buttons instead old buttons in button box.

13.3 Class DialogRightButtonBox

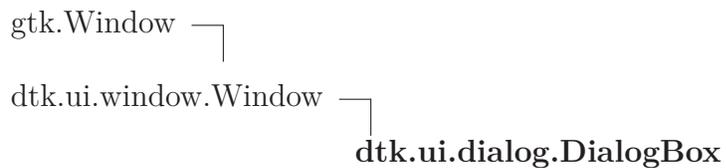


HBox to handle right side buttons in DialogBox.

13.3.1 Methods

__init__ (<i>self</i>) <hr/> Initialize DialogRightButtonBox class.
set_buttons (<i>self</i> , <i>buttons</i>) <hr/> Set buttons in box. Parameters <i>buttons</i> : A list of Gtk.Widget instance. Note: This functin will use new buttons instead old buttons in button box.

13.4 Class DialogBox



Dialog box to standard dialog layout and ui detail.

If you want build a dialog, you should use this standard.

13.4.1 Methods

```
__init__(self, title, default_width=None, default_height=None,
mask_type=None, close_callback=None, modal=True,
window_hint=gtk.gdk.WINDOW_TYPE_HINT_DIALOG, window_pos=None,
skip_taskbar_hint=True, resizable=False)
```

Initialize `DialogBox` class.

Parameters

- title:** Dialog title.
- default_width:** Width of dialog, default is None.
- default_height:** Height of dialog, default is None.
- mask_type:** Background mask type, it allow use below type:
- `DIALOG_MASK_SINGLE_PAGE` single mask style, use in single page that background mask include dialog button area.
 - `DIALOG_MASK_GLASS_PAGE` glass mask style, similar `DIALOG_MASK_SINGLE_PAGE` but with different color.
 - `DIALOG_MASK_MULTIPLE_PAGE` multiple mask style, use in multiple page that background mask not include dialog button area.
 - `DIALOG_MASK_TAB_PAGE` tab mask style, use in preference page that background mask not include button area.

Overrides: `dtk.ui.window.Window.__init__`

```
get_mask_func(self, widget, padding_left=0, padding_right=0,
padding_top=0, padding_bottom=0)
```

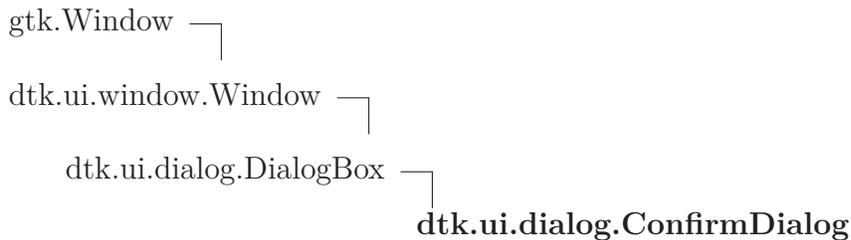
Get mask function to render background, you can use this function to return "render function" to draw your ui to keep same style.

Parameters

- widget:** `DialogBox` widget.
- padding_left:** Padding at left side.
- padding_right:** Padding at right side.
- padding_top:** Padding at top side.
- padding_bottom:** Padding at bottom side.

Inherited from `gtk.ui.window.Window`(Section 60.1)

`add_move_event()`, `add_toggle_event()`, `close_window()`, `draw_mask()`, `get_edge()`,
`get_shadow_size()`, `hide_shadow()`, `is_disable_window_maximized()`, `min_window()`,
`resize_window()`, `show_shadow()`, `show_window()`, `toggle_fullscreen_window()`,
`toggle_max_window()`

13.5 Class `ConfirmDialog`

Simple message confirm dialog.

13.5.1 Methods

__init__ (<i>self</i> , <i>title</i> , <i>message</i> , <i>default_width</i> =330, <i>default_height</i> =145, <i>confirm_callback</i> =None, <i>cancel_callback</i> =None)	
Initialize <code>ConfirmDialog</code> class.	
Parameters	
<code>title</code> :	Title for confirm dialog.
<code>message</code> :	Confirm message.
<code>default_width</code> :	Dialog width, default is 330 pixel.
<code>default_height</code> :	Dialog height, default is 145 pixel.
<code>confirm_callback</code> :	Callback when user click confirm button.
<code>cancel_callback</code> :	Callback when user click cancel button.
Overrides: <code>gtk.ui.window.Window.__init__</code>	

Inherited from `gtk.ui.dialog.DialogBox`(Section 13.4)

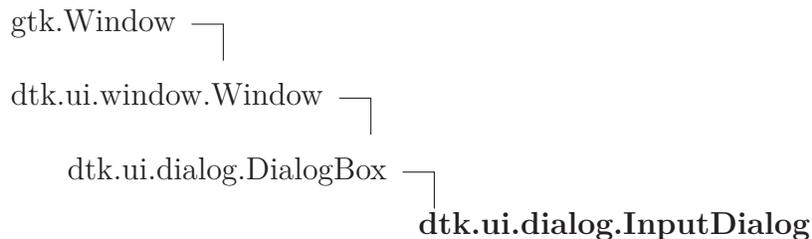
`get_mask_func()`

Inherited from `gtk.ui.window.Window`(Section 60.1)

`add_move_event()`, `add_toggle_event()`, `close_window()`, `draw_mask()`, `get_edge()`,
`get_shadow_size()`, `hide_shadow()`, `is_disable_window_maximized()`, `min_window()`,

`resize_window()`, `show_shadow()`, `show_window()`, `toggle_fullscreen_window()`,
`toggle_max_window()`

13.6 Class `InputDialog`



Simple input dialog.

13.6.1 Methods

<code>__init__(self, title, init_text, default_width=330, default_height=145, confirm_callback=None, cancel_callback=None)</code>	
Initialize <code>InputDialog</code> class.	
Parameters	
<code>title</code> :	Input dialog title.
<code>init_text</code> :	Initialize input text.
<code>default_width</code> :	Width of dialog, default is 330 pixel.
<code>default_height</code> :	Height of dialog, default is 330 pixel.
<code>confirm_callback</code> :	Callback when user click confirm button, this callback accept one argument that return by user input text.
<code>cancel_callback</code> :	Callback when user click cancel button, this callback not need argument.
Overrides: <code>dtk.ui.window.Window.__init__</code>	

<code>focus_input(self, widget)</code>	
Grab focus on input entry.	
Parameters	
<code>widget</code> :	<code>InputDialog</code> widget.

Inherited from `dtk.ui.dialog.DialogBox`(Section 13.4)

`get_mask_func()`

Inherited from `gtk.ui.window.Window` (Section 60.1)

`add_move_event()`, `add_toggle_event()`, `close_window()`, `draw_mask()`, `get_edge()`,
`get_shadow_size()`, `hide_shadow()`, `is_disable_window_maximized()`, `min_window()`,
`resize_window()`, `show_shadow()`, `show_window()`, `toggle_fullscreen_window()`,
`toggle_max_window()`

13.7 Class `OpenFileDialog`

`gtk.FileChooserDialog` —
↳ **`gtk.ui.dialog.OpenFileDialog`**

Simple dialog to open file.

13.7.1 Methods

<code>__init__</code> (<i>self</i> , <i>title</i> , <i>parent</i> , <i>ok_callback</i> =None, <i>cancel_callback</i> =None)	
Initialize <code>OpenFileDialog</code> class.	
Parameters	
<code>title</code> :	Dialog title.
<code>parent</code> :	Parent widget to call open file dialog.
<code>ok_callback</code> :	Callback when user click ok button, this function accept one argument: filename.
<code>cancel_callback</code> :	Callback when user click cancel button, this function accept one argument: filename.

13.8 Class `SaveFileDialog`

`gtk.FileChooserDialog` —
↳ **`gtk.ui.dialog.SaveFileDialog`**

Simple dialog to save file.

13.8.1 Methods

<code>__init__</code> (<i>self</i> , <i>title</i> , <i>parent</i> , <i>ok_callback</i> =None, <i>cancel_callback</i> =None)	
Initialize SaveFileDialog class.	
Parameters	
<code>title</code> :	Dialog title.
<code>parent</code> :	Parent widget to call open file dialog.
<code>ok_callback</code> :	Callback when user click ok button, this function accept one argument: filename.
<code>cancel_callback</code> :	Callback when user click cancel button, this function accept one argument: filename.

14 Module `dtk.ui.dominant_color`

14.1 Functions

`get_dominant_color(image_path)`

Parse image and return dominant color in image.

Parameters

`image_path`: Image path to parse.

Return Value

Return dominant color, format as hexadecimal number.

15 Module `gtk.ui.draw`

15.1 Functions

`draw_radial_ring`(*cr, x, y, outer_radius, inner_radius, color_infos*)

Draw radial ring.

Parameters

cr: Cairo context.
x: X coordinate of draw area.
y: Y coordinate of draw area.
outer_radius: Radius for outer ring.
inner_radius: Radius for inner ring.
color_infos: A list of `ColorInfo`, `ColorInfo` format as [(`color_pos`, (`color_hex_value`, `color_alpha`))].

`get_desktop_pixbuf`()

Get screenshot of desktop.

Return Value

Return desktop screenshot as `gtk.gdk.Pixbuf`.

`draw_round_rectangle`(*cr, x, y, width, height, r*)

Draw round rectangle.

Parameters

cr: Cairo context.
x: X coordinate of rectangle area.
y: Y coordinate of rectangle area.
width: Width of rectangle area.
height: Height of rectangle area.
r: Radius of rectangle corner.

draw_pixbuf(*cr, pixbuf, x=0, y=0, alpha=1.0*)

Draw pixbuf on cairo context, this function use frequently for image render.

Parameters

cr: Cairo context.
pixbuf: `gtk.gdk.Pixbuf`
x: X coordiante of draw area.
y: Y coordiante of draw area.
alpha: Alpha value to render pixbuf, float value between 0 and 1.0

draw_window_frame(*cr, x, y, w, h, color_frame_outside_1, color_frame_outside_2, color_frame_outside_3, color_frame_inside_1, color_frame_inside_2*)

Draw window frame.

Parameters

cr: Cairo context.
x: X coordiante of draw area.
y: Y coordiante of draw area.
w: Width of draw area.
h: Height of draw area.
color_frame_outside_1: Use for draw outside 8 points.
color_frame_outside_2: Use for draw middle 4 points.
color_frame_outside_3: Use for draw inside 4 points.
color_frame_inside_1: Use for draw outside frame.
color_frame_inside_2: Use for draw inner frame and inside 4 points.

```
draw_window_rectangle(cr, sx, sy, ex, ey, r)
```

Draw window rectangle.

Parameters

cr: Cairo context.
sx: Source x coordinate.
sy: Source y coordinate.
ex: Target x coordinate.
ey: Target x coordinate.
r: Window frame radius.

```
draw_text(cr, markup, x, y, w, h, text_size=DEFAULT_FONT_SIZE,  

text_color="#000000", text_font=DEFAULT_FONT,  

alignment=pango.ALIGN_LEFT, gaussian_radius=None,  

gaussian_color=None, border_radius=None, border_color=None,  

wrap_width=None)
```

Standard function for draw text.

Parameters

cr: Cairo context.
markup: Pango markup string.
x: X coordinate of draw area.
y: Y coordinate of draw area.
w: Width of draw area.
h: Height of draw area.
text_size: Text size, default is `DEFAULT_FONT_SIZE`.
text_color: Text color, default is `"#000000"`.
text_font: Text font, default is `DEFAULT_FONT`.
alignment: Font alignment option, default is `pango.ALIGN_LEFT`. You can set `pango.ALIGN_MIDDLE` or `pango.ALIGN_RIGHT`.
gaussian_radius: Gaussian radius, default is `None`.
gaussian_color: Gaussian color, default is `None`.
border_radius: Border radius, default is `None`.
border_color: Border color, default is `None`.
wrap_width: Wrap width of text, default is `None`.

```
render_text(cr, markup, x, y, w, h, text_size=DEFAULT_FONT_SIZE,  
text_color="#000000", text_font=DEFAULT_FONT,  
alignment=pango.ALIGN_LEFT, wrap_width=None)
```

Render text for function `draw_text`, you can use this function individually.

Parameters

cr: Cairo context.

markup: Pango markup string.

x: X coordinate of draw area.

y: Y coordinate of draw area.

w: Width of draw area.

h: Height of draw area.

text_size: Text size, default is `DEFAULT_FONT_SIZE`.

text_color: Text color, default is `"#000000"`.

text_font: Text font, default is `DEFAULT_FONT`.

alignment: Font alignment option, default is `pango.ALIGN_LEFT`. You can set `pango.ALIGN_MIDDLE` or `pango.ALIGN_RIGHT`.

wrap_width: Wrap width of text, default is `None`.

```
draw_line(cr, sx, sy, ex, ey, line_width=1,  
antialias_status=cairo.ANTIALIAS_NONE)
```

Draw line.

Parameters

cr: Cairo context.

sx: Source X coordinate.

sy: Source Y coordinate.

ex: Target X coordinate.

ey: Target Y coordinate.

line_width: Line width, default is 1 pixel.

antialias_status: Antialias status, default is `cairo.ANTIALIAS_NONE`.

`draw_vlinear`(*cr, x, y, w, h, color_infos, radius=0, top_to_bottom=True*)

Draw linear area vertically.

Parameters

`cr`: Cairo context.

`x`: X coordinate of draw area.

`y`: Y coordinate of draw area.

`w`: Width of draw area.

`h`: Height of draw area.

`color_infos`: A list of `ColorInfo`, `ColorInfo` format:
(`color_stop_position`, (`color_hex_value`,
`color_alpha`))

`radius`: Rectangle corner radius.

`top_to_bottom`: Draw direction, default is from top to bottom,
function will draw from bottom to top if set option
as `False`.

`draw_hlinear`(*cr, x, y, w, h, color_infos, radius=0, left_to_right=True*)

Draw linear area horticulturally.

Parameters

`cr`: Cairo context.

`x`: X coordinate of draw area.

`y`: Y coordinate of draw area.

`w`: Width of draw area.

`h`: Height of draw area.

`color_infos`: A list of `ColorInfo`, `ColorInfo` format:
(`color_stop_position`, (`color_hex_value`,
`color_alpha`))

`radius`: Rectangle corner radius.

`left_to_right`: Draw direction, default is from left to right,
function will draw from right to left if set option as
`False`.

expose_linear_background(*widget, event, color_infos*)

Expose linear background.

Parameters

widget: Gtk.Widget instance.
event: Expose event.
color_infos: A list of ColorInfo, ColorInfo format:
(color_stop_position, (color_hex_value,
color_alpha))

draw_window_shadow(*cr, x, y, w, h, r, p, color_window_shadow*)

Draw window shadow.

Parameters

cr: Cairo context.
x: X coordinate of draw area.
y: Y coordinate of draw area.
w: Width of draw area.
h: Height of draw area.
r: Radius of window shadow corner.
p: Padding between window shadow and
window frame.
color_window_shadow: theme.DynamicShadowColor.

draw_radial_round(*cr, x, y, r, color_infos*)

Draw radial round.

Parameters

cr: Cairo context.
x: X coordinate of draw area.
y: Y coordinate of draw area.
r: Radius of radial round.
color_infos: A list of ColorInfo, ColorInfo format:
(color_stop_position, (color_hex_value,
color_alpha))

`draw_blank_mask`(*cr, x, y, w, h*)

Draw blank mask, use for default mask function.

Parameters

cr: Cairo context.

x: X coordiante of rectangle area.

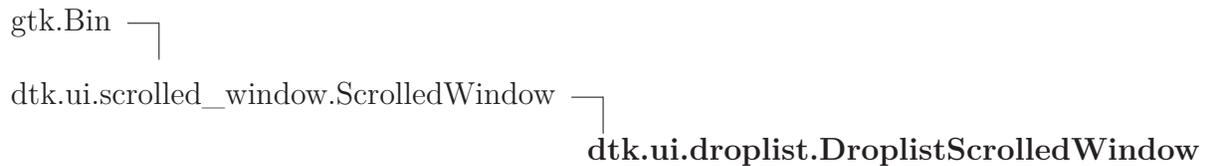
y: Y coordiante of rectangle area.

w: Width of rectangle area.

h: Width of rectangle area.

16 Module `gtk.ui.droplist`

16.1 Class `DroplistScrolledWindow`



ScrolledWindow for droplist.

16.1.1 Methods

__init__ (<i>self</i> , <i>right_space</i> =2, <i>top_bottom_space</i> =3)	
Initialize <code>DroplistScrolledWindow</code> class.	
Parameters	
<code>right_space</code> :	the space between right border and the vertical scrollbar.
<code>top_bottom_space</code> :	the space between top border and the vertical scrollbar.
Overrides: <code>dtk.ui.scrolled_window.ScrolledWindow.__init__</code>	

Inherited from `dtk.ui.scrolled_window.ScrolledWindow` (Section 41.1)

`add_child()`, `add_with_viewport()`, `get_hadjustment()`, `get_vadjustment()`, `set_hadjustment()`, `set_vadjustment()`

16.2 Class `Droplist`



Droplist.

16.2.1 Methods

```
__init__(self, items, x_align=ALIGN_START, y_align=ALIGN_START,
font_size=DEFAULT_FONT_SIZE, opacity=1.0, padding_x=0, padding_y=0,
item_padding_left=6, item_padding_right=32, item_padding_y=3,
max_width=None)
```

Initialize Droplist class.

Parameters

items:	A list of item, item format: (item_content, item_value).
x_align:	Horticultural alignment.
y_align:	Vertical alignment.
font_size:	Font size of droplist, default is DEFAULT_FONT_SIZE
opacity:	Opacity of droplist window, default is 1.0.
padding_x:	Padding x, default is 0.
padding_y:	Padding y, default is 0.
item_padding_left:	Padding at left of item, default is 6.
item_padding_right:	Padding at right of item, default is 32.
item_padding_y:	Padding of item vertically, default is 3.
max_width:	Maximum width of droplist, default is None.

```
get_droplist_width(self)
```

Get droplist width.

```
get_first_index(self)
```

Get index of first item.

Return Value

Return index of first item, or return None if haven't item in droplist.

```
get_last_index(self)
```

Get index of last item.

Return Value

Return index of last item, or return None if haven't item in droplist.

get_prev_index(*self*)

Get index of previous item.

Return Value

Return index of previous item, or return None if haven't item in droplist.

get_next_index(*self*)

Get index of next item.

Return Value

Return index of next item, or return None if haven't item in droplist.

get_select_item_rect(*self*, *item_index*=None)

Get item rectangle with given index.

Parameters*item_index*: If *item_index* is None, use select index.**Return Value**

Return (x, y, w, h) rectangle for match item.

active_item(*self*, *item_index*=None)

Select item with given index.

Parameters*item_index*: If *item_index* is None, use select index.**select_first_item**(*self*)

Select first item.

select_last_item(*self*)

Select last item.

select_prev_item(*self*)

Select previous item.

select_next_item(*self*)

Select next item.

scroll_page_to_select_item(*self*)

Scroll page to select item.

scroll_page_up(*self*)

Scroll page up.

scroll_page_down(*self*)

Scroll page down.

press_select_item(*self*)

Press select item.

get_droplist_item_at_coordinate(*self*, (*x*, *y*))

Get droplist item at coordinate, return None if haven't any droplist item at given coordinate.

Parameters

x: X coordiante.

y: Y coordiante.

Return Value

Return match item with given coordinate, return None if haven't any item match coordinate.

show(*self*, (*x*, *y*), (*offset_x*, *offset_y*)=(0,0))

Show droplist.

Parameters

x: Show x coordinate.

y: Show y coordinate.

offset_x: Offset x value when droplist haven't space to show in origin coordinate, default is 0.

offset_y: Offset y value when droplist haven't space to show in origin coordinate, default is 0.

hide(*self*)

Hide droplist.

16.2.2 Class Variables

Name	Description
<code>__gsignals__</code>	Value: { <code>"item-selected"</code> : (<code>gobject.SIGNAL_RUN_LAST</code> , <code>gobject.TYPE_N...</code>)}

16.3 Class DroplistItem

object —
 dtk.ui.droplist.DroplistItem

DroplistItem for *Droplist*.

16.3.1 Methods

<code>__init__(self, droplist, index, item, font_size, droplist_padding_x, droplist_padding_y, item_padding_left, item_padding_right, item_padding_y, max_width)</code>	
Initialize DroplistItem class.	
Parameters	
<code>droplist:</code>	Droplist.
<code>index:</code>	Drop item index.
<code>item:</code>	Drop item, format (item_content, item_value)
<code>font_size:</code>	Drop item font size.
<code>droplist_padding_x:</code>	Padding x of droplist.
<code>droplist_padding_y:</code>	Padding y of droplist.
<code>item_padding_left:</code>	Padding at left of item.
<code>item_padding_right:</code>	Padding at right of item.
<code>item_padding_y:</code>	Padding at top or bottom of item.
<code>max_width:</code>	Maximum width of droplist item.
Overrides: object. <code>__init__</code>	

Inherited from object

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`,
`__reduce__()`, `__reduce_ex__()`, `__repr__()`, `__setattr__()`, `__sizeof__()`,
`__str__()`, `__subclasshook__()`

16.3.2 Properties

Name	Description
<i>Inherited from object</i> __class__	

17 Module `gtk.ui.entry`

17.1 Class `Entry`

`gtk.EventBox` —
 `gtk.ui.entry.Entry`

`Entry`.

17.1.1 Methods

```
__init__(self, content="", padding_x=5, padding_y=2,
text_color=ui_theme.get_color("entry_text"),
text_select_color=ui_theme.get_color("entry_select_text"), back-
ground_select_color=ui_theme.get_shadow_color("entry_select_background"),
font_size=DEFAULT_FONT_SIZE)
```

Initialize `Entry` class.

Parameters

<code>content</code> :	Entry initialize content, default is "".
<code>padding_x</code> :	Horizontal padding value, default is 5 pixel.
<code>padding_y</code> :	Vertical padding value, default is 2 pixel.
<code>text_color</code> :	Color of text in normal status.
<code>text_select_color</code> :	Color of text in select status.
<code>background_select_color</code> :	Color of background in select status.
<code>font_size</code> :	Entry font size, default is <code>DEFAULT_FONT_SIZE</code> .

```
set_editable(self, editable)
```

Set entry editable status.

Parameters

`editable`: If it is `True`, entry can edit, else entry not allow edit.

is_editable(*self*)

Whether entry is editable.

Return Value

Return True if entry editable, else return False.

set_text(*self*, *text*)

Set entry text.

Parameters

text: Entry text string.

get_text(*self*)

Get entry text.

Return Value

Return entry text string.

clear_select_status(*self*)

Clear entry select status.

move_to_start(*self*)

Move cursor to start position of entry.

move_to_end(*self*)

Move cursor to end position of entry.

move_to_left(*self*)

Backward cursor one char.

move_to_right(*self*)

Forward cursor one char.

backspace(*self*)

Do backspace action.

select_all(*self*)

Select all text of entry.

cut_to_clipboard (<i>self</i>)

Cut selected text to clipboard.

copy_to_clipboard (<i>self</i>)
--

Copy selected text to clipboard.

paste_from_clipboard (<i>self</i>)

Paste text to entry from clipboard.

press_return (<i>self</i>)

Do return action.

select_to_left (<i>self</i>)

Select text to left char.

select_to_right (<i>self</i>)
--

Select text to right char.

select_to_start (<i>self</i>)
--

Select text to start position.

select_to_end (<i>self</i>)

Select text to end position.

delete (<i>self</i>)

Delete selected text.

17.1.2 Class Variables

Name	Description
MOVE_LEFT	Value: 1
MOVE_RIGHT	Value: 2
MOVE_NONE	Value: 3
__gsignals__	Value: {"edit-alarm": (gobject.SIGNAL_RUN_LAST, gobject.TYPE_NONE...

17.2 Class `TextEntry`

```

gtk.VBox ┌
         │
         └─ dtk.ui.entry.TextEntry

```

Text entry.

17.2.1 Methods

```

__init__(self, content="", action_button=None, back-
ground_color=ui_theme.get_alpha_color("text_entry_background"),
acme_color=ui_theme.get_alpha_color("text_entry_acme"),
point_color=ui_theme.get_alpha_color("text_entry_point"),
frame_point_color=ui_theme.get_alpha_color("text_entry_frame_point"),
frame_color=ui_theme.get_alpha_color("text_entry_frame"))

```

Initialize `InputEntry` class.

Parameters

<code>content</code> :	Initialize entry text, default is "".
<code>action_button</code> :	Extra button add at right side of text entry, default is <code>None</code> .
<code>background_color</code> :	Color of text entry background.
<code>acme_color</code> :	Acme point color of text entry.
<code>point_color</code> :	Pointer color of text entry.
<code>frame_point_color</code> :	Frame pointer color of text entry.
<code>frame_color</code> :	Frame color of text entry.

```

set_size(self, width, height)

```

Set text entry size with given value.

Parameters

<code>width</code> :	New width of text entry.
<code>height</code> :	New height of text entry.

```

set_editable(self, editable)

```

Set editable status of text entry.

Parameters

<code>editable</code> :	Text entry can editable if option is <code>True</code> , else can't edit.
-------------------------	---

set_text (<i>self</i> , <i>text</i>)
Set text of text entry.
Parameters
<i>text</i> : Text entry string.

get_text (<i>self</i>)
Get text of text entry.
Return Value
Return text of text entry.

focus_input (<i>self</i>)
Focus input cursor.

17.2.2 Class Variables

Name	Description
<code>__gsignals__</code>	Value: { <code>"action-active"</code> : (<code>gobject.SIGNAL_RUN_LAST</code> , <code>gobject.TYPE_N...</code>)

17.3 Class `InputEntry`

gtk.VBox —
 gtk.ui.entry.InputEntry

Text entry.

Generically speaking, `InputEntry` is similar `TextEntry`, only difference between two class is ui style, internal logic is same.

17.3.1 Methods

```
__init__(self, content="", action_button=None, background_color=ui_theme.get_alpha_color("text_entry_background"),
         acme_color=ui_theme.get_alpha_color("text_entry_acme"),
         point_color=ui_theme.get_alpha_color("text_entry_point"),
         frame_point_color=ui_theme.get_alpha_color("text_entry_frame_point"),
         frame_color=ui_theme.get_alpha_color("text_entry_frame"))
```

Initialize `InputEntry` class.

Parameters

<code>content:</code>	Initialize entry text, default is "".
<code>action_button:</code>	Extra button add at right side of input entry, default is <code>None</code> .
<code>background_color:</code>	Color of input entry background.
<code>acme_color:</code>	Acme point color of input entry.
<code>point_color:</code>	Pointer color of input entry.
<code>frame_point_color:</code>	Frame pointer color of input entry.
<code>frame_color:</code>	Frame color of input entry.

```
set_size(self, width, height)
```

Set input entry size with given value.

Parameters

<code>width:</code>	New width of input entry.
<code>height:</code>	New height of input entry.

```
set_editable(self, editable)
```

Set editable status of input entry.

Parameters

<code>editable:</code>	input entry can editable if option is <code>True</code> , else can't edit.
------------------------	--

```
set_text(self, text)
```

Set text of input entry.

Parameters

<code>text:</code>	input entry string.
--------------------	---------------------

<code>get_text(self)</code>
Get text of input entry.
Return Value
Return text of input entry.

<code>focus_input(self)</code>
Focus input cursor.

17.3.2 Class Variables

Name	Description
<code>__gsignals__</code>	Value: { <code>"action-active"</code> : (<code>gobject.SIGNAL_RUN_LAST</code> , <code>gobject.TYPE_N...</code>)

17.4 Class `ShortcutKeyEntry`



Shortcut key entry.

17.4.1 Methods

```
__init__(self, content="", action_button=None, background_color=ui_theme.get_alpha_color("text_entry_background"),
acme_color=ui_theme.get_alpha_color("text_entry_acme"),
point_color=ui_theme.get_alpha_color("text_entry_point"),
frame_point_color=ui_theme.get_alpha_color("text_entry_frame_point"),
frame_color=ui_theme.get_alpha_color("text_entry_frame"))
```

Initialize `ShortcutKeyEntry` class.

Parameters

<code>content</code> :	Initialize entry text, default is "".
<code>action_button</code> :	Extra button add at right side of shortcutkey entry, default is None.
<code>background_color</code> :	Color of shortcutkey entry background.
<code>acme_color</code> :	Acme point color of shortcutkey entry.
<code>point_color</code> :	Pointer color of shortcutkey entry.
<code>frame_point_color</code> :	Frame pointer color of shortcutkey entry.
<code>frame_color</code> :	Frame color of shortcutkey entry.

```
handle_button_press(self, widget, event)
```

Internal callback for 'action-active' signal.

```
set_shortcut_key(self, shortcut_key)
```

Set shortcut key.

Parameters

<code>shortcut_key</code> :	Key string that return by function 'dtk.ui.keymap.get_keyevent_name'.
-----------------------------	---

```
get_shortcut_key(self)
```

Get shortcut key.

Return Value

	Return shortcut key string, string format look function 'dtk.ui.keymap.get_keyevent_name'.
--	--

set_size (<i>self</i> , <i>width</i> , <i>height</i>)
Set shortcutkey entry size with given value.
Parameters
<i>width</i> : New width of shortcutkey entry.
<i>height</i> : New height of shortcutkey entry.

set_editable (<i>self</i> , <i>editable</i>)
Set editable status of shortcutkey entry.
Parameters
<i>editable</i> : shortcutkey entry can be edited if option is True, else can't edit.

set_text (<i>self</i> , <i>text</i>)
Set text of shortcutkey entry.
Parameters
<i>text</i> : shortcutkey entry string.

get_text (<i>self</i>)
Get text of shortcutkey entry.
Return Value
Return text of shortcutkey entry.

focus_input (<i>self</i>)
Focus input cursor.

17.4.2 Class Variables

Name	Description
<code>__gsignals__</code>	Value: { <code>"action-active"</code> : (<code>gobject.SIGNAL_RUN_LAST</code> , <code>gobject.TYPE_N...</code>)

18 Module `gtk.ui.frame`

18.1 Class `HorizontalFrame`

`gtk.Alignment` — `gtk.ui.frame.HorizontalFrame`

Horizontal frame to padding 1 pixel round child.

18.1.1 Methods

<p><code>__init__</code>(<i>self</i>, <i>padding</i>=1, <i>xalign</i>=0.0, <i>yalign</i>=0.0, <i>xscale</i>=1.0, <i>yscale</i>=1.0)</p> <p>Initialize <code>HorizontalFrame</code> class.</p> <p>Parameters</p> <p><code>padding</code>: Padding value.</p> <p><code>xalign</code>: The fraction of horizontal free space to the left of the child widget. Ranges from 0.0 to 1.0.</p> <p><code>yalign</code>: The fraction of vertical free space above the child widget. Ranges from 0.0 to 1.0.</p> <p><code>xscale</code>: The fraction of horizontal free space that the child widget absorbs, from 0.0 to 1.0.</p> <p><code>yscale</code>: The fraction of vertical free space that the child widget absorbs, from 0.0 to 1.0.</p>

18.2 Class `VerticalFrame`

`gtk.Alignment` — `gtk.ui.frame.VerticalFrame`

Vertical frame to padding 1 pixel round child.

18.2.1 Methods

`__init__`(*self*, *padding*=1, *xalign*=0.0, *yalign*=0.0, *xscale*=1.0, *yscale*=1.0)

Initialize *VerticalFrame* class.

Parameters

`padding`: Padding value.

`xalign`: The fraction of horizontal free space to the left of the child widget. Ranges from 0.0 to 1.0.

`yalign`: The fraction of vertical free space above the child widget. Ranges from 0.0 to 1.0.

`xscale`: The fraction of horizontal free space that the child widget absorbs, from 0.0 to 1.0.

`yscale`: The fraction of vertical free space that the child widget absorbs, from 0.0 to 1.0.

19 Module dtk.ui.global_key

19.1 Functions

<code>enable_global_key()</code>

Enable global key.

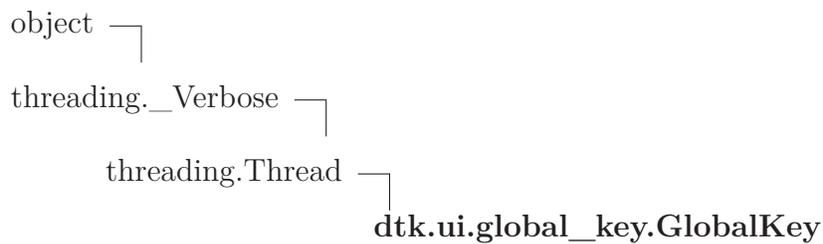
<code>disable_global_key()</code>

Disable global key.

19.2 Variables

Name	Description
<code>global_key_running</code>	Value: True
<code>global_key_lock</code>	Value: Lock()

19.3 Class GlobalKey



Class to handle global key.

19.3.1 Methods

<code>__init__(self)</code>

Init for global key.

Overrides: object.__init__

bind(*self*, *binding_string*, *action*)

Binding keymap with given action.

Parameters

binding_string: Keymap string, return by function
‘get_keyevent_name’ of module dtk.ui.keymap.
action: Callback.

unbind(*self*, *binding_string*)

Unbind keymap.

Parameters

binding_string: Keymap string that return by function
‘get_keyevent_name’ of module dtk.ui.keymap.

grab(*self*)

Grab key.

ungrab(*self*)

Ungrab key.

regrab(*self*)

Regrab key.

run(*self*)

GlobalKey thread loop.

Overrides: threading.Thread.run

exit(*self*)

Exit global key.

Inherited from threading.Thread

__repr__(), daemon(), getName(), ident(), isAlive(), isDaemon(), is_alive(),
join(), name(), setDaemon(), setName(), start()

Inherited from object

__delattr__(), __format__(), __getattr__(), __hash__(), __new__(),
__reduce__(), __reduce_ex__(), __setattr__(), __sizeof__(), __str__(),
__subclasshook__()

19.3.2 Properties

Name	Description
<i>Inherited from object</i> __class__	

20 Module *gtk.ui.group*

20.1 Class *ImageButtonGroup*



Group container for *ImageButton*.

20.1.1 Methods

<code>__init__</code> (<i>self</i> , <i>items</i> , <i>spacing</i> =5)
Initialize for <i>ImageButtonGroup</i> class.
Parameters
<i>items</i> : A list of <i>ImageButton</i> item, item format: (<i>normal_dpixbuf</i> , <i>hover_dpixbuf</i> , <i>press_dpixbuf</i> , <i>clicked_callback</i>)
<i>spacing</i> : Spacing value between items, default is 5 pixel.

20.2 Class *ToggleButtonGroup*



Group container for *ToggleButton*.

20.2.1 Methods

<code>__init__</code> (<i>self</i> , <i>items</i> , <i>spacing</i> =5)
Initialize for <i>ToggleButtonGroup</i> class.
Parameters
<i>items</i> : A list of <i>ImageButton</i> item, item format: (<i>inactive_dpixbuf</i> , <i>active_dpixbuf</i> , <i>inactive_hover_dpixbuf</i> , <i>active_hover_dpixbuf</i> , <i>toggled_callback</i>)
<i>spacing</i> : Spacing value between items, default is 5 pixel.

is_active (<i>self</i>)

Whether at toggle button group is active.

Return Value

Return True if active, else return False.

20.3 Class *ImageButtonItem*

```

gtk.Button ┌
           │
           └─ gtk.ui.group.ImageButtonItem

```

Image button item for *ImageButtonGroup*.

20.3.1 Methods

__init__ (<i>self, item, index, set_index, get_index</i>)
--

Initialize <i>ImageButtonItem</i> class.
--

Parameters

<i>item</i> :	Image button item, item format: (normal_dpixbuf, hover_dpixbuf, press_dpixbuf, clicked_callback)
---------------	--

<i>index</i> :	Item index.
----------------	-------------

<i>set_index</i> :	Set index callback.
--------------------	---------------------

<i>get_index</i> :	Get index callback.
--------------------	---------------------

20.4 Class *ToggleButtonItem*

```

gtk.ToggleButton ┌
                  │
                  └─ gtk.ui.group.ToggleButtonItem

```

Toggle button item for *ToggleButtonGroup*.

20.4.1 Methods

<hr/> <code>__init__</code> (<i>self</i> , <i>item</i> , <i>index</i> , <i>set_index</i> , <i>get_index</i>) <hr/>
Initialize <code>ToggleButtonItem</code> class.
Parameters
<i>item</i> : Toggle button item, item format: (<i>inactive_dpixbuf</i> , <i>active_dpixbuf</i> , <i>inactive_hover_dpixbuf</i> , <i>active_hover_dpixbuf</i> , <i>toggled_callback</i>)
<i>index</i> : Item index.
<i>set_index</i> : Set index callback.
<i>get_index</i> : Get index callback.

<hr/> <code>wrap_toggle_button_clicked_action</code> (<i>self</i>) <hr/>
Internal function to wrap clicked action.

select_right_item(*self*)

Select right item.

scroll_page_up(*self*)

Scroll page up of iconview.

scroll_page_down(*self*)

Scroll page down of iconview.

add_items(*self*, *items*, *insert_pos=None*)

Add items to iconview.

Parameters

- items**: A list of item that follow the rule of *IconItem*.
- insert_pos**: Insert position, default is *None* to insert new item at **end** position.

delete_items(*self*, *items*)

Delete items.

Parameters

- items**: Items need to remove.

clear(*self*)

Clear all items.

draw_mask(*self*, *cr*, *x*, *y*, *w*, *h*)

Draw mask interface.

Parameters

- cr**: Cairo context.
- x**: X coordiante of draw area.
- y**: Y coordiante of draw area.
- w**: Width of draw area.
- h**: Height of draw area.

clear_focus_item(*self*)

Clear item's focus status.

get_width(*self*)

Get item width.

This is *IconView* interface, you should implement it.

get_height(*self*)

Get item height.

This is *IconView* interface, you should implement it.

render(*self*, *cr*, *rect*)

Render item.

This is *IconView* interface, you should implement it.

icon_item_motion_notify(*self*, *x*, *y*)

Handle 'motion-notify-event' signal.

This is *IconView* interface, you should implement it.

icon_item_lost_focus(*self*)

Lost focus.

This is *IconView* interface, you should implement it.

icon_item_highlight(*self*)

Highlight item.

This is *IconView* interface, you should implement it.

icon_item_normal(*self*)

Set item with normal status.

This is *IconView* interface, you should implement it.

icon_item_button_press(*self*, *x*, *y*)

Handle button-press event.

This is *IconView* interface, you should implement it.

icon_item_button_release (<i>self</i> , <i>x</i> , <i>y</i>)

Handle button-release event.

This is IconView interface, you should implement it.
--

icon_item_single_click (<i>self</i> , <i>x</i> , <i>y</i>)

Handle single click event.

This is IconView interface, you should implement it.
--

icon_item_double_click (<i>self</i> , <i>x</i> , <i>y</i>)

Handle double click event.

This is IconView interface, you should implement it.
--

21.2.2 Class Variables

Name	Description
<code>__gsignals__</code>	Value: { <code>"redraw-request"</code> : (<code>gobject.SIGNAL_RUN_LAST</code> , <code>gobject.TYPE_...</code>)

22 Module *dtk.ui.keymap*

22.1 Functions

get_key_name(*keyval*)

Get key name with given key value.

Parameters

keyval: Key value.

Return Value

Return key name with given key value.

get_key_event_modifiers(*key_event*)

Get key modifiers with given key event.

Parameters

key_event: Key event.

Return Value

Return key modifier list with given key event.

get_keyevent_name(*key_event*)

Get key event name.

Parameters

key_event: Key event.

Return Value

Return key event string.

parse_keyevent_name(*keyevent_name*)

Parse keyevent name.

Parameters

keyevent_name: Key event name that return by function
get_keyevent_name.

Return Value

Return tuple that contain key value and modifier mask, (*keyval*,
modifier_mask).

has_ctrl_mask(*key_event*)

Whether has ctrl mask in key event.

Parameters

key_event: Key event.

Return Value

Return true if key event has ctrl mask.

has_shift_mask(*key_event*)

Whether has shift mask in key event.

Parameters

key_event: Key event.

Return Value

Return true if key event has shift mask.

23 Module `gtk.ui.label`

23.1 Class `Label`



`Label`.

23.1.1 Methods

```

__init__(self, text, text_color=None, text_size=DEFAULT_FONT_SIZE,
text_x_align=ALIGN_START, label_width=None, enable_gaussian=False,
enable_select=True, enable_double_click=True, gaussian_radious=2,
border_radious=1, wrap_width=None)
  
```

Initialize `Label` class.

Parameters

<code>text</code> :	Label text.
<code>text_color</code> :	Label text color, default is <code>None</code> .
<code>text_size</code> :	Label text size, default is <code>DEFAULT_FONT_SIZE</code> .
<code>text_x_align</code> :	Horizontal align option, default is <code>ALIGN_START</code> .
<code>label_width</code> :	Label maximum width, default is <code>None</code> .
<code>enable_gaussian</code> :	Default is <code>False</code> , if it is <code>True</code> , color option no effect, default gaussian effect is white text and black shadow.
<code>enable_select</code> :	Default is <code>True</code> , label content can't select if it is <code>False</code> .
<code>gaussian_radious</code> :	Radious of gaussian.
<code>border_radious</code> :	Radious of border.
<code>wrap_width</code> :	Wrap width.

```

copy_to_clipboard(self)
  
```

Copy select text to clipboard.

select_all(*self*)

Select all.

get_text(*self*)

Get text of label.

set_text(*self*, *text*)

Set text with given value.

Parameters

text: Label string.

24 Module *gtk.ui.line*

24.1 Class *HSeparator*

gtk.Alignment —
gtk.ui.line.HSeparator

Horizontal separator.

24.1.1 Methods

<code>__init__</code> (<i>self</i> , <i>color_infos</i> , <i>padding_x</i> =0, <i>padding_y</i> =0)
Init horizontal separator.
Parameters
<i>color_infos</i> : A list of color info, [(position, (hex_color, alpha_value))]
<i>padding_x</i> : Padding value in horizontally.
<i>padding_y</i> : Padding value in vertically.

24.2 Class *VSeparator*

gtk.Alignment —
gtk.ui.line.VSeparator

Vertically separator.

24.2.1 Methods

<code>__init__</code> (<i>self</i> , <i>color_infos</i> , <i>padding_x</i> =0, <i>padding_y</i> =0)
Init vertically separator.
Parameters
<i>color_infos</i> : A list of color info, [(position, (hex_color, alpha_value))]
<i>padding_x</i> : Padding value in horizontally.
<i>padding_y</i> : Padding value in vertically.

25 Module `gtk.ui.listview`

25.1 Functions

```
render_text(cr, rect, content, in_select, in_highlight, align=ALIGN_START, font_size=DEFAULT_FONT_SIZE)
```

Helper render text function for `Listitem`, you should implement your own.

Parameters

<code>cr</code> :	Cairo context.
<code>rect</code> :	Draw area.
<code>content</code> :	Content.
<code>in_select</code> :	Whether item is selected.
<code>in_highlight</code> :	Whether item is highlighted.
<code>align</code> :	Render alignment option, default is <code>ALIGN_START</code> .
<code>font_size</code> :	Render font size, default is <code>DEFAULT_FONT_SIZE</code> .

```
render_image(cr, rect, image_path, x, y)
```

Helper render image function for `Listitem`, you should implement your own.

Parameters

<code>cr</code> :	Cairo context.
<code>rect</code> :	Draw area.
<code>image_path</code> :	Image path.
<code>x</code> :	X coordiante of draw position.
<code>y</code> :	Y coordiante of draw position.

25.2 Class `Listview`

```
gtk.DrawingArea ┌
                 │
                 └─ gtk.ui.listview.ListView
```

Powerful listview widget.

25.2.1 Methods

```
__init__(self, sorts=[], drag_data=None, enable_multiple_select=True,
enable_drag_drop=True,
drag_icon_pixbuf=ui_theme.get_pixbuf("listview/drag_preview.png"),
drag_out_offset=50)
```

Initialize ListView widget.

Parameters

sorts:	Sort function for column of listview.
drag_data:	Drag data for drag data from listview, format: (targets, actions, button_masks)
enable_multiple_select:	Whether allow user select multiple item, default is True.
enable_drag_drop:	Whether allow user drag drop on listview, default is True.
drag_icon_pixbuf:	Drag icon.
drag_out_offset:	Out offset value to trigger drag action on listview, default is 50 pixel, if cursor not drag more than 50 pixel, listview won't think it is drag out action .

```
set_expand_column(self, column)
```

Set expand column.

Parameters

column: Column index to expand space.

```
add_titles(self, titles, title_height=24)
```

Add titles.

Parameters

titles: A list of title.
title_height: Height of title.

```
get_title_sizes(self)
```

Get title sizes.

Return Value

Return title size, as format (title_width, title_height).

add_items(*self*, *items*, *insert_pos*=None, *sort_list*=False)

Add items in listview.**Parameters**

items: A list of item.

insert_pos: The position to insert, default is None will insert new item at end of list.

sort_list: Whether sort list after insert, default is False.

sort_items(*self*, *compare_method*, *sort_reverse*=False)

Sort items with given method.**Parameters**

compare_method: Compare method to sort.

sort_reverse: Whether sort reverse, default is False.

redraw_item(*self*, *list_item*)

Redraw item.**Parameters**

list_item: List item need to redraw.

update_item_index(*self*)

Update index of items.

set_title_height(*self*, *title_height*)

Set title height.

get_column_sort_type(*self*, *column*)

Get sort type with given column index.**Parameters**

column: Column index.

Return Value

Return sort type with given column index, return None if haven't found match column index.

set_column_sort_type(*self*, *column*, *sort_type*)

Set sort type with given value.

Parameters

column: Column index.
sort_type: Sort type.

get_cell_widths(*self*)

Get cell width of columns.

set_cell_width(*self*, *column*, *width*)

Set cell width with given value.

Parameters

column: Column index.
width: Column width.

reset_cursor(*self*)

Reset cursor type.

get_offset_coordinate(*self*, *widget*)

Get viewport offset coordinate and viewport.

Parameters

widget: ListView widget.

Return Value

Return viewport offset and viewport: (*offset_x*, *offset_y*, *viewport*).

draw_shadow_mask(*self*, *cr*, *x*, *y*, *w*, *h*)

Shadow mask interface for overwrite.

Parameters

cr: Cairo context.
x: X coordiante of draw area.
y: Y coordiante of draw area.
w: Width of draw area.
h: Height of draw area.

draw_mask(*self*, *cr*, *x*, *y*, *w*, *h*)

Draw mask interface.

Parameters

- cr**: Cairo context.
- x**: X coordiante of draw area.
- y**: Y coordiante of draw area.
- w**: Width of draw area.
- h**: Height of draw area.

draw_item_hover(*self*, *cr*, *x*, *y*, *w*, *h*)

Draw item hover interface.

Parameters

- cr**: Cairo context.
- x**: X coordiante of draw area.
- y**: Y coordiante of draw area.
- w**: Width of draw area.
- h**: Height of draw area.

draw_item_select(*self*, *cr*, *x*, *y*, *w*, *h*)

Draw item select interface.

Parameters

- cr**: Cairo context.
- x**: X coordiante of draw area.
- y**: Y coordiante of draw area.
- w**: Width of draw area.
- h**: Height of draw area.

draw_item_highlight(*self, cr, x, y, w, h*)

Draw item highlight interface.

Parameters

- cr**: Cairo context.
- x**: X coordiante of draw area.
- y**: Y coordiante of draw area.
- w**: Width of draw area.
- h**: Height of draw area.

keep_select_status(*self*)

Handy function that change listview and keep select status not change.

is_in_visible_area(*self, event*)

Is event coordinate in visible area.

Parameters

- event**: gtk.gdk.Event.

Return Value

Return True if event coordiante in visible area.

emit_item_event(*self, event_name, event*)

Wrap method for emit event signal.

Parameters

- event_name**: Event name.
- event**: Event.

get_coordinate_row(*self, y*)

Get row with given y coordinate.

Parameters

- y**: Y coordinate.

Return Value

Return row that match given y coordinate, return None if haven't any row match y coordiante.

get_event_row(*self*, *event*, *offset_index=0*)

Get row at event.

Parameters

event: gtk.gdk.Event instance.
offset_index: Offset index base on event row.

Return Value

Return row at event coordinate, return None if haven't any row match event coordiante.

select_first_item(*self*)

Select first item.

select_last_item(*self*)

Select last item.

scroll_page_up(*self*)

Scroll page up.

scroll_page_down(*self*)

Scroll page down.

select_prev_item(*self*)

Select preview item.

select_next_item(*self*)

Select next item.

select_to_prev_item(*self*)

Select to preview item.

select_to_next_item(*self*)

Select to next item.

select_to_first_item(*self*)

Select to first item.

select_to_last_item (<i>self</i>)
Select to last item.

select_all_items (<i>self</i>)
Select all items.

delete_select_items (<i>self</i>)
Delete select items.

update_vadjustment (<i>self</i>)
Update vertical adjustment.

double_click_item (<i>self</i>)
Double click item.

clear (<i>self</i>)
Clear all list.

get_current_item (<i>self</i>)
Get current item.
Return Value
Return select row, or return None if not any item selected.

set_highlight (<i>self</i> , <i>item</i>)
Set highlight with given item.

clear_highlight (<i>self</i>)
Clear highlight status.

visible_highlight (<i>self</i>)
Visible highlight item.

25.2.2 Class Variables

Name	Description
SORT_DESCENDING	Value: False
SORT_ASCENDING	Value: True

continued on next page

Name	Description
<code>SORT_PADDING_X</code>	Value: 5
<code>TITLE_PADDING</code>	Value: 5
<code>__gsignals__</code>	Value: {"delete-select-items": (gobject.SIGNAL_RUN_LAST, gobject....

25.3 Class *ListItem*



ListItem template to build your own item for *ListView*.

Note: This class just template to build list item, you should build new item with same interface.

25.3.1 Methods

<code>__init__(self, title, artist, length)</code>
Initialize <i>ListItem</i> class.
Parameters
title: Title.
artist: Artist.
length: Length.

<code>set_index(self, index)</code>
Update index.
This is <i>ListView</i> interface, you should implement it.
Parameters
index: Index.

<code>get_index(self)</code>
Get index.
This is <i>ListView</i> interface, you should implement it.

emit_redraw_request(*self*)

Emit redraw-request signal.

This is ListView interface, you should implement it.

update(*self*, *title*, *artist*, *length*)

Update.

This is ListView interface, you should implement it.

Parameters

title: Title.

artist: Artist.

length: Length.

render_title(*self*, *cr*, *rect*, *in_select*, *in_highlight*)

Render title.

Parameters

cr: Cairo context.

rect: Redraw rectangle.

in_select: Whether current item is selected, this value pass from ListView.

in_highlight: Whether current item is highlighted, this value pass from ListView.

render_artist(*self*, *cr*, *rect*, *in_select*, *in_highlight*)

Render artist.

Parameters

cr: Cairo context.

rect: Redraw rectangle.

in_select: Whether current item is selected, this value pass from ListView.

in_highlight: Whether current item is highlighted, this value pass from ListView.

render_length (<i>self</i> , <i>cr</i> , <i>rect</i> , <i>in_select</i> , <i>in_highlight</i>)
Render length.
Parameters
<i>cr</i> : Cairo context.
<i>rect</i> : Redraw rectangle.
<i>in_select</i> : Whether current item is selected, this value pass from ListView.
<i>in_highlight</i> : Whether current item is highlighted, this value pass from ListView.

get_column_sizes (<i>self</i>)
Get column sizes.
This is ListView interface, you should implement it.
Return Value
Return column size tuple.

get_renders (<i>self</i>)
Get render callbacks.
This is ListView interface, you should implement it.
Return Value
Return render functions.

25.3.2 Class Variables

Name	Description
<code>__gsignals__</code>	Value: { <code>"redraw-request"</code> : (<code>gobject.SIGNAL_RUN_LAST</code> , <code>gobject.TYPE_...</code>)

26 Module *gtk.ui.listview__preview__pixbuf*

26.1 Functions

render__pixbuf(*widget, event, input_args*)

Render and save pixbuf.

Parameters

widget: Gtk.Widget instance.

event: Expose event.

input_args: Input arguments as format: (select_num, vlinear_color, text_color, filepath).

27 Module *dtk.ui.locales*

27.1 Variables

Name	Description
LOCALE_DIR	Value: <code>"/usr/share/locale"</code>
—	Value: <code>lambda i:</code>

28 Module `gtk.ui.mask`

28.1 Functions

`draw_mask`(*widget, x, y, w, h, render_callback*)

Draw mask with given render method.

Parameters

`widget`: Target widget.
`x`: X coordinate of draw area.
`y`: Y coordinate of draw area.
`w`: Width of draw area.
`h`: Height of draw area.
`render_callback`: Render callback.

`draw_window_mask`(*widget, x, y, w, h, render_callback*)

Draw window mask with given render method.

Parameters

`widget`: Target widget.
`x`: X coordinate of draw area.
`y`: Y coordinate of draw area.
`w`: Width of draw area.
`h`: Height of draw area.
`render_callback`: Render callback.

`draw_scrolled_window_mask`(*widget, x, y, w, h, render_callback*)

Draw scrolled window mask with given render method.

Parameters

`widget`: Target widget.
`x`: X coordinate of draw area.
`y`: Y coordinate of draw area.
`w`: Width of draw area.
`h`: Height of draw area.
`render_callback`: Render callback.

`draw_icon_view_mask(widget, x, y, w, h, render_callback)`

Draw icon view mask with given render method.

Parameters

`widget`: Target widget.
`x`: X coordinate of draw area.
`y`: Y coordinate of draw area.
`w`: Width of draw area.
`h`: Height of draw area.
`render_callback`: Render callback.

`draw_list_view_mask(widget, x, y, w, h, render_callback)`

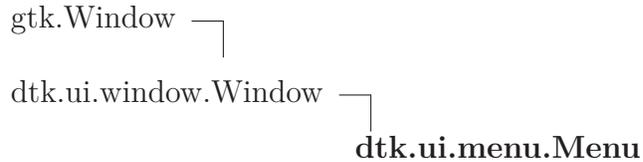
Draw list view mask with given render method.

Parameters

`widget`: Target widget.
`x`: X coordinate of draw area.
`y`: Y coordinate of draw area.
`w`: Width of draw area.
`h`: Height of draw area.
`render_callback`: Render callback.

29 Module `gtk.ui.menu`

29.1 Class `Menu`



`Menu`.

29.1.1 Methods

```

__init__(self, items, is_root_menu=False, select_scale=False,
x_align=ALIGN_START, y_align=ALIGN_START,
font_size=DEFAULT_FONT_SIZE, padding_x=3, padding_y=3,
item_padding_x=6, item_padding_y=3, shadow_visible=True,
menu_min_width=130)
  
```

Initialize `Menu` class.

Parameters

- `items`: A list of item, item format: (item_icon, item_name, item_node).
- `is_root_menu`: Default is False for submenu, you should set it as True if you build root menu.
- `select_scale`: Default is False, it will use parent's width if it set True.
- `x_align`: Horizontal alignment value.
- `y_align`: Vertical alignment value.
- `font_size`: Menu font size, default is `DEFAULT_FONT_SIZE`
- `padding_x`: Horizontal padding value, default is 3 pixel.
- `padding_y`: Vertical padding value, default is 3 pixel.
- `item_padding_x`: Horizontal item padding value, default is 6 pixel.
- `item_padding_y`: Vertical item padding value, default is 3 pixel.
- `shadow_visible`: Whether show window shadow, default is True.
- `menu_min_width`: Minimum width of menu.

Overrides: `gtk.ui.window.Window.__init__`

draw_menu_mask (<i>self</i> , <i>cr</i> , <i>x</i> , <i>y</i> , <i>w</i> , <i>h</i>)
Draw mask interface.
Parameters
<i>cr</i> : Cairo context.
<i>x</i> : X coordinate of draw area.
<i>y</i> : Y coordinate of draw area.
<i>w</i> : Width of draw area.
<i>h</i> : Height of draw area.

show (<i>self</i> , (<i>x</i> , <i>y</i>), (<i>offset_x</i> , <i>offset_y</i>)=(0,0))
Show menu with given position.
Parameters
<i>x</i> : X coordinate of menu.
<i>y</i> : Y coordinate of menu.
<i>offset_x</i> : Offset x when haven't enough space to show menu, default is 0.
<i>offset_y</i> : Offset y when haven't enough space to show menu, default is 0.

hide (<i>self</i>)
Hide menu.

Inherited from dtk.ui.window.Window(Section 60.1)

add_move_event(), add_toggle_event(), close_window(), draw_mask(), get_edge(), get_shadow_size(), hide_shadow(), is_disable_window_maximized(), min_window(), resize_window(), show_shadow(), show_window(), toggle_fullscreen_window(), toggle_max_window()

29.2 Class MenuItem



Menu item for *Menu*.

29.2.1 Methods

```
__init__(self, item, font_size, select_scale, show_submenu_callback,
hide_submenu_callback, get_root_menu_callback, get_menu_items_callback,
icon_width, icon_height, have_submenu, submenu_width, submenu_height,
menu_padding_x, menu_padding_y, item_padding_x, item_padding_y,
min_width)
```

Initialize MenuItem class.

Parameters

<code>item</code> :	item format: (item_icon, itemName, item_node).
<code>font_size</code> :	Menu font size.
<code>select_scale</code> :	Default is False, it will use parant's width if it set True.
<code>show_submenu_callback</code> :	Callback when show submenus.
<code>hide_submenu_callback</code> :	Callback when hide submenus.
<code>get_root_menu_callback</code> :	Callback to get root menu.
<code>get_menu_items_callback</code> :	Callback to get menu items.
<code>icon_width</code> :	Icon width.
<code>icon_height</code> :	Icon height.
<code>have_submenu</code> :	Whether have submenu.
<code>submenu_width</code> :	Width of submenu.
<code>submenu_height</code> :	Height of submenu.
<code>menu_padding_x</code> :	Horizontal padding of menu.
<code>menu_padding_y</code> :	Vertical padding of menu.
<code>item_padding_x</code> :	Horizontal padding of item.
<code>item_padding_y</code> :	Vertical padding of item.
<code>min_width</code> :	Minimum width.

Overrides: object.__init__

Inherited from object

```
__delattr__(), __format__(), __getattr__(), __hash__(), __new__(),
__reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(),
__str__(), __subclasshook__()
```

29.2.2 Properties

Name	Description
<i>Inherited from object</i> __class__	

31 Module `dtk.ui.mplayer_window`

31.1 Class `MplayerWindow`

gtk.Window —
 `dtk.ui.mplayer_window.MplayerWindow`

Special Window class for mplayer.

Generally speaking, compared with Window class, it uses a different shadow mechanism.

31.1.1 Methods

<pre>__init__(self, enable_resize=False, shadow_radius=6, window_type=gtk.WINDOW_TOPLEVEL)</pre>
Initialise the Window class.
Parameters
<pre>enable_resize: If True, the window will be set resizable. By default, it's False.</pre>
<pre>shadow_radius: The radius of the shadow.</pre>
<pre>window_type: A flag of type gtk._gtk.WindowType, which indicates the type of the window. By default, it's gtk.WINDOW_TOPLEVEL.</pre>

<pre>show_window(self)</pre>
Show the window.

<pre>draw_mask(self, cr, x, y, w, h)</pre>
Draw mask interface, you should implement it you own.
Parameters
<pre>cr: Cairo context.</pre>
<pre>x: X coordinate of draw area.</pre>
<pre>y: Y coordinate of draw area.</pre>
<pre>w: Width of draw area.</pre>
<pre>h: Height of draw area.</pre>

set_window_shape(*self*, *shape_flag*)

Enable window shape.

Parameters

shape_flag: The flag that indicates the shape.

hide_shadow(*self*)

Hide the window shadow.

show_shadow(*self*)

Show the window shadow.

is_disable_window_maximized(*self*)

An interface which indicates whether the window could be maximized, you should implement this function you own.

Return Value

Always return False.

min_window(*self*)

Minimize the window. Make it iconified.

toggle_max_window(*self*)

Toggle the window size between maximized size and normal size.

toggle_fullscreen_window(*self*)

Toggle the window between fullscreen mode and normal size.

close_window(*self*)

Close the window. Send the destroy signal to the program.

Return Value

Always return False.

resize_window(*self*, *widget*, *event*)

Resize the window.

Parameters

widget: The window of type gtk.Widget.

event: A signal of type gtk.gdk.Event.

add_move_event(*self*, *widget*)

Add move event callback.

Parameters

widget: A widget of type gtk.Widget.

add_toggle_event(*self*, *widget*)

Add toggle event callback.

Parameters

widget: A widget of type gtk.Widget.

get_edge(*self*)

Get the edge which the cursor is on, according to the cursor type.

Return Value

If there is a corresponding cursor type, return an instance of gtk.gdk.WindowEdge, else return None.

get_shadow_size(*self*)

Get the shadow size.

Return Value

Always return (0, 0)

32 Module `dtk.ui.navigatebar`

32.1 Class `Navigatebar`



`Navigatebar`.

32.1.1 Methods

```

__init__(self, items, add_separator=False,
font_size=DEFAULT_FONT_SIZE, padding_x=10, padding_y=10,
vertical=True,
item_hover_pixbuf=ui_theme.get_pixbuf("navigatebar/nav_item_hover.png"),
item_press_pixbuf=ui_theme.get_pixbuf("navigatebar/nav_item_press.png"))

```

Initialize `Navigatebar` class.

Parameters

<code>items</code> :	A list of navigate item, item format: (<code>item_icon_dpixbuf</code> , <code>item_content</code> , <code>clicked_callback</code>)
<code>add_separator</code> :	Whether add separator between <code>navigatebar</code> and body, default is <code>False</code> .
<code>font_size</code> :	Font size, default is <code>DEFAULT_FONT_SIZE</code> .
<code>padding_x</code> :	Padding value horizontal.
<code>padding_y</code> :	Padding value vertical.
<code>vertical</code> :	Draw direction, default is vertical.
<code>item_hover_pixbuf</code> :	Item hover <code>dpixbuf</code> .
<code>item_press_pixbuf</code> :	Item press <code>dpixbuf</code> .

Overrides: `dtk.ui.box.EventBox.__init__`

```

set_index(self, index)

```

Set selected item with given index.

Parameters

<code>index</code> :	Item index.
----------------------	-------------

get_index (<i>self</i>)
Get selected index.
Return Value
Return selected item index.
expose_nav_separator (<i>self, widget, event</i>)
Internal callback for 'expose-event' signal.

32.2 Class NavItem

object —
 dtk.ui.navigatebar.NavItem

Navigate item.

32.2.1 Methods

__init__ (<i>self, element, index, font_size, padding_x, padding_y, vertical, set_index, get_index, item_hover_pixbuf, item_press_pixbuf</i>)
Initialize NavItem class.
Parameters
<i>element</i> : Item format: (item_icon_dpixbuf, item_content, clicked_callback)
<i>index</i> : Item index.
<i>font_size</i> : Font size.
<i>padding_x</i> : Padding value horizontal.
<i>padding_y</i> : Padding value vertical.
<i>vertical</i> : Draw direction.
<i>set_index</i> : Set index callback.
<i>get_index</i> : Get index callback.
<i>item_hover_pixbuf</i> : Item hover pixbuf.
<i>item_press_pixbuf</i> : Item press pixbuf.
Overrides: object.__init__

Inherited from object

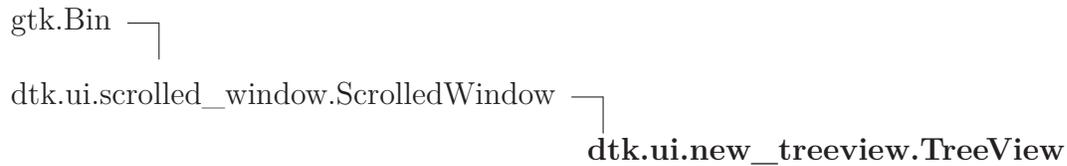
`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`,
`__reduce__()`, `__reduce_ex__()`, `__repr__()`, `__setattr__()`, `__sizeof__()`,
`__str__()`, `__subclasshook__()`

32.2.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

33 Module `gtk.ui.new_treeview`

33.1 Class `TreeView`



`TreeView` widget.

33.1.1 Methods

```

__init__(self, items=[], sort_methods=[], row_normal_height=None,
row_select_height=None, drag_data=None, enable_multiple_select=True,
enable_drag_drop=True, drag_icon_pixbuf=None, start_drag_offset=50,
right_space=2, top_bottom_space=3)
  
```

Initialize `TreeView` class.

Parameters

`right_space`: the space between right border and the vertical scrollbar.

`top_bottom_space`: the space between top border and the vertical scrollbar.

Overrides: `dtk.ui.scrolled_window.ScrolledWindow.__init__`

Inherited from `dtk.ui.scrolled_window.ScrolledWindow` (Section 41.1)

`add_child()`, `add_with_viewport()`, `get_hadjustment()`, `get_vadjustment()`, `set_hadjustment()`, `set_vadjustment()`

33.2 Class `TreeItem`



Tree item template use for `TreeView`.

33.2.1 Methods

<code>__init__(<i>self</i>)</code>

Initialize <i>TreeItem</i> class.

34 Module `gtk.ui.notebook`

34.1 Class `Notebook`

```
gtk.EventBox ┌
              │
              └─ dtk.ui.notebook.Notebook
```

`Notebook`.

34.1.1 Methods

```
__init__(self, items, fore-
ground_left_pixbuf=ui_theme.get_pixbuf("notebook/foreground_left.png"),
fore-
ground_middle_pixbuf=ui_theme.get_pixbuf("notebook/foreground_middle.png"),
fore-
ground_right_pixbuf=ui_theme.get_pixbuf("notebook/foreground_right.png"),
back-
ground_left_pixbuf=ui_theme.get_pixbuf("notebook/background_left.png"),
back-
ground_middle_pixbuf=ui_theme.get_pixbuf("notebook/background_middle.png"),
back-
ground_right_pixbuf=ui_theme.get_pixbuf("notebook/background_right.png"))
```

Initialize `Notebook` class.

Parameters

<code>items</code> :	Notebook item, format (<code>item_icon</code> , <code>item_content</code> , <code>item_callback</code>).
<code>foreground_left_pixbuf</code> :	Left foreground pixbuf.
<code>foreground_middle_pixbuf</code> :	Middle foreground pixbuf.
<code>foreground_right_pixbuf</code> :	Right foreground pixbuf.
<code>background_left_pixbuf</code> :	Left background pixbuf.
<code>background_middle_pixbuf</code> :	Middle background pixbuf.
<code>background_right_pixbuf</code> :	Right background pixbuf.

button_press_notebook(*self*, *widget*, *event*)

Internal callback for 'button-press-event' signal.

Parameters

widget: Notebook widget.

event: Button press event.

35 Module `gtk.ui.osd_tooltip`

35.1 Class `OSDTooltip`

`gtk.Window` —
 `gtk.ui.osd_tooltip.OSDTooltip`

OSD tooltip.

35.1.1 Methods

```
__init__(self, monitor_widget, text_font=DEFAULT_FONT, text_size=18,
offset_x=0, offset_y=0,
text_color=ui_theme.get_color("osd_tooltip_text"),
border_color=ui_theme.get_color("osd_tooltip_border"),
border_radious=1)
```

Initialize `OSDTooltip` class.

Parameters

`monitor_widget`: Widget to monitor event.
`text_font`: Text font, default is `DEFAULT_FONT`.
`text_size`: Text size, default is 18.
`offset_x`: Offset X coordinate relative to monitor widget.
`offset_y`: Offset Y coordinate relative to monitor widget.
`text_color`: Text color.
`border_color`: Border color.
`border_radious`: Border radious.

```
show(self, text)
```

Show.

Parameters

`text`: OSD tooltip text.

```
hide_immediately(self)
```

Hide immediately.

change_style(*self*, *text_font*, *text_size*)

Change OSD tooltip style.

Parameters

text_font: OSD tooltip text font.

text_size: OSD tooltip text size.

36 Module *gtk.ui.paned*

36.1 Class *Paned*

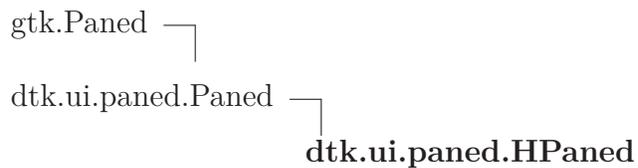


Paned.

36.1.1 Methods

<code>__init__</code> (<i>self</i>) <hr/> Initialize <i>Paned</i> class.
--

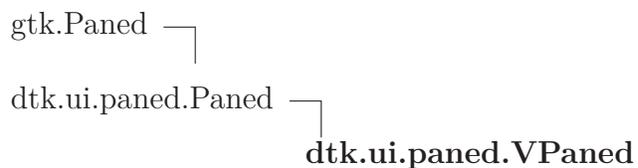
36.2 Class *HPaned*



36.2.1 Methods

<code>__init__</code> (<i>self</i>) Initialize <i>Paned</i> class. Overrides: <code>dtk.ui.paned.Paned.__init__</code> <code>exitit</code> (inherited documentation)

36.3 Class *VPaned*



36.3.1 Methods

`__init__(self)`

Initialize Paned class.

Overrides: `dtk.ui.paned.Paned.__init__` extit(inherited documentation)

37 Module `gtk.ui.panel`

37.1 Class `Panel`



`Panel`.

37.1.1 Methods

<p><code>__init__</code>(<i>self</i>, <i>width</i>, <i>height</i>, <i>window_type</i>=<code>gtk.WINDOW_TOPLEVEL</code>)</p> <hr/> <p>Initialize <code>Panel</code> class.</p> <p>Parameters</p> <p><code>width</code>: Initialize panel width.</p> <p><code>height</code>: Initialize panel height.</p> <p><code>window_type</code>: Window type, default is <code>gtk.WINDOW_TOPLEVEL</code>.</p>
<p><code>show_panel</code>(<i>self</i>)</p> <hr/> <p>Show panel.</p>
<p><code>hide_panel</code>(<i>self</i>)</p> <hr/> <p>Hide panel.</p>
<p><code>resize_panel</code>(<i>self</i>, <i>w</i>, <i>h</i>)</p> <hr/> <p>Resize panel.</p> <p>Parameters</p> <p><code>w</code>: Resize width.</p> <p><code>h</code>: Resize height.</p>

38 Module `gtk.ui.progressbar`

38.1 Class `ProgressBar`

`gtk.Button` —
`gtk.ui.progressbar.ProgressBar`

Progress bar.

38.1.1 Methods

<code>__init__(self)</code>
Initialize progress bar.

<code>test_progressbar(self)</code>
Test progressbar.

39 Module `dtk.ui.pseudo_skin`

39.1 Variables

Name	Description
app_theme	Value: <code>Theme(os.path.join(get_parent_dir(__file__</code> <code>3), "app_them...</code>

40 Module `gtk.ui.scalebar`

40.1 Class `HScalebar`



`HScalebar`.

40.1.1 Methods

```
__init__(self, left_fg_dpixbuf, left_bg_dpixbuf, middle_fg_dpixbuf,
middle_bg_dpixbuf, right_fg_dpixbuf, right_bg_dpixbuf, point_dpixbuf)
```

Init `HScalebar` class.

Parameters

`left_fg_dpixbuf`: Left foreground pixbuf.
`left_bg_dpixbuf`: Left background pixbuf.
`middle_fg_dpixbuf`: Middle foreground pixbuf.
`middle_bg_dpixbuf`: Middle background pixbuf.
`right_fg_dpixbuf`: Right foreground pixbuf.
`right_bg_dpixbuf`: Right background pixbuf.
`point_dpixbuf`: Pointer pixbuf.

40.2 Class `VScalebar`



`VScalebar`.

40.2.1 Methods

__init__(*self*, *upper_fg_dpixbuf*, *upper_bg_dpixbuf*, *middle_fg_dpixbuf*, *middle_bg_dpixbuf*, *bottom_fg_dpixbuf*, *bottom_bg_dpixbuf*, *point_dpixbuf*)

Initialize VScalebar class.

Parameters

upper_fg_dpixbuf: Upper foreground pixbuf.

upper_bg_dpixbuf: Upper background pixbuf.

middle_fg_dpixbuf: Middle foreground pixbuf.

middle_bg_dpixbuf: Middle background pixbuf.

bottom_fg_dpixbuf: Bottom foreground pixbuf.

bottom_bg_dpixbuf: Bottom background pixbuf.

point_dpixbuf: Pointer pixbuf.

set_has_point(*self*, *value*)

Set has point.

get_has_point(*self*)

Get has point.

41 Module `gtk.ui.scrolled_window`

41.1 Class `ScrolledWindow`

`gtk.Bin` —
 `gtk.ui.scrolled_window.ScrolledWindow`

The scrolled window with deepin's custom scrollbar.

41.1.1 Methods

<p><code>__init__</code>(<i>self</i>, <i>right_space</i>=2, <i>top_bottom_space</i>=3)</p> <hr/> <p>Init scrolled window.</p> <p>Parameters</p> <p> <i>right_space</i>: the space between right border and the vertical scrollbar.</p> <p> <i>top_bottom_space</i>: the space between top border and the vertical scrollbar.</p>
--

<p><code>add_with_viewport</code>(<i>self</i>, <i>child</i>)</p> <hr/> <p>Used to add children without native scrolling capabilities.</p> <p>If a child has native scrolling, use <code>ScrolledWindow.add()</code> instead of this function.</p> <p>Parameters</p> <p> <i>child</i>: the child without native scrolling.</p>
--

<p><code>add_child</code>(<i>self</i>, <i>child</i>)</p> <hr/> <p>Add the child to this <code>ScrolledWindow</code>. The child should have native scrolling capabilities.</p> <p>Parameters</p> <p> <i>child</i>: the child with native scrolling.</p>

<p><code>get_vadjustment</code>(<i>self</i>)</p> <hr/> <p>Returns the vertical scrollbar's adjustment, used to connect the vertical scrollbar to the child widget's vertical scroll functionality.</p>

get_hadjustment(*self*)

Returns the horizontal scrollbar's adjustment, used to connect the horizontal scrollbar to the child widget's horizontal scroll functionality.

set_hadjustment(*self*, *adj*)

Sets the `gtk.Adjustment` for the horizontal scrollbar.

Parameters

adj: horizontal scroll adjustment

set_vadjustment(*self*, *adj*)

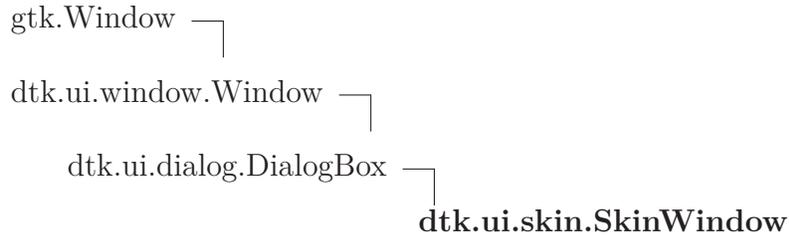
Sets the `gtk.Adjustment` for the vertical scrollbar.

Parameters

adj: vertical scroll adjustment

42 Module *gtk.ui.skin*

42.1 Class *gtk.ui.window.Window*



gtk.ui.window.Window.

42.1.1 Methods

<code>__init__</code> (<i>self</i> , <i>app_frame_pixbuf</i> , <i>preview_width</i> =450, <i>preview_height</i> =500)
--

Initialize *gtk.ui.window.Window* class.

Parameters

app_frame_pixbuf: Application's pixbuf for frame.

preview_width: Preview width, default is 450.

preview_height: Preview height, default is 500.

Overrides: *gtk.ui.window.Window.__init__*

Inherited from dtk.ui.dialog.DialogBox(Section 13.4)

get_mask_func()

Inherited from dtk.ui.window.Window(Section 60.1)

add_move_event(), *add_toggle_event*(), *close_window*(), *draw_mask*(), *get_edge*(),
get_shadow_size(), *hide_shadow*(), *is_disable_window_maximized*(), *min_window*(),
resize_window(), *show_shadow*(), *show_window*(), *toggle_fullscreen_window*(),
toggle_max_window()

43 Module *dtk.ui.skin_config*

43.1 Variables

Name	Description
<code>skin_config</code>	Value: <code>SkinConfig()</code>

43.2 Class *SkinConfig*

gobject.GObject —
 dtk.ui.skin_config.SkinConfig

SkinConfig.

43.2.1 Methods

<code>__init__</code> (<i>self</i>)
Initialize <i>SkinConfig</i> class.

<code>set_application_window_size</code> (<i>self</i> , <i>app_window_width</i> , <i>app_window_height</i>)
Set application window with given size.
Parameters
<i>app_window_width</i> : Application window width.
<i>app_window_height</i> : Application window height.

```
init_skin(self, skin_name, system_skin_dir, user_skin_dir,  
skin_config_file, app_given_id, app_given_version)
```

Init skin.

Parameters

skin_name: Skin name.
system_skin_dir: Default skin directory.
user_skin_dir: User's skin directory, generic use
~/.config/project-name/skin
skin_config_file: Skin's config filepath, generic use
~/.config/project-name/skin_config.ini
app_given_id: Project name.
app_given_version: Project version.

```
load_themes(self, ui_theme, app_theme)
```

Load theme from given directories.

Parameters

ui_theme: `dtk.ui.theme.ui_theme`.
app_theme: Theme instance, build it like below:

```
>>> app_theme = Theme(  
>>>     os.path.join(get_parent_dir(__file__), "app_theme"),  
>>>     os.path.expanduser("~/config/project-name/theme")  
>>> )
```

44 Module `gtk.ui.slider`

44.1 Class `Slider`

```

gtk.Viewport ┌
              │
              └─ gtk.ui.slider.Slider

```

`Slider`.

44.1.1 Methods

<p><code>__init__</code>(<i>self</i>, <i>slide_callback</i>=None)</p> <hr/> <p>Initialize <code>Slider</code> class.</p> <p>Parameters</p> <p><code>slide_callback</code>: Callback when slider change image, arguments: (index, widget), default is None.</p>
--

<p><code>slide_to</code>(<i>self</i>, <i>widget</i>)</p> <hr/> <p>Slide to given widget.</p> <p>Parameters</p> <p><code>widget</code>: Widget in slider.</p>
--

<p><code>append_widget</code>(<i>self</i>, <i>widget</i>)</p> <hr/> <p>Append widget.</p> <p>Parameters</p> <p><code>widget</code>: Widget to append in slider.</p>

<p><code>add_slide_timeout</code>(<i>self</i>, <i>widget</i>, <i>milliseconds</i>)</p> <hr/> <p>Adds a timeout for “<code>widget</code>” to slide in after “<code>seconds</code>”.</p> <p>Parameters</p> <p><code>widget</code>: Add widget.</p> <p><code>milliseconds</code>: Delay time.</p>
--

<p><code>remove_slide_timeout</code>(<i>self</i>, <i>widget</i>)</p> <hr/> <p>Removes a timeout previously added by “<code>add_slide_timeout</code>”.</p> <p>Parameters</p> <p><code>widget</code>: Remove widget from slider.</p>
--

```
reset_slide_timeout(self, widget, milliseconds=None)
```

Shorthand to “`remove_slide_timeout`” plus “`add_slide_timeout`”.

Parameters

`widget`: Slider widget.
`milliseconds`: New delay value.

```
try_remove_slide_timeout(self, widget)
```

Try remove slide timeout that match given widget.

Parameters

`widget`: Match widget.

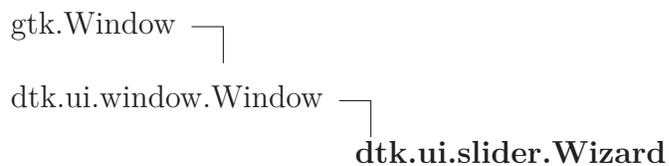
```
try_reset_slide_timeout(self, widget, *args, **kwargs)
```

Like “`reset_slide_timeout`”, but fails silently if the timeout for “`widget`” does not exist.

44.1.2 Class Variables

Name	Description
<code>active_widget</code>	Value: None

44.2 Class Wizard



Wizard.

44.2.1 Methods

```
__init__(self, slider_files, navigate_files, finish_callback=None,
window_width=548, window_height=373, navigatebar_height=58,
slide_delay=4000, close_area_width=32, close_area_height=32)
```

Initialize Wizard class.

Parameters

`slider_files`: Big slide image file list.
`navigate_files`: Small navigate image file list.
`finish_callback`: Callback when finish show.
`window_width`: Default window width.
`window_height`: Default window height.
`navigatebar_height`: Navigatebar height.
`slide_delay`: Delay of slide, in milliseconds.
`close_area_width`: Width of top-right close area.
`close_area_height`: Height of top-right close area.

Overrides: `dtk.ui.window.Window.__init__`

```
set_slide_page(self, index, widget)
```

Set slide page.

Parameters

`index`: Index to set.
`widget`: The widget you want to set.

Inherited from `dtk.ui.window.Window` (Section 60.1)

`add_move_event()`, `add_toggle_event()`, `close_window()`, `draw_mask()`, `get_edge()`,
`get_shadow_size()`, `hide_shadow()`, `is_disable_window_maximized()`, `min_window()`,
`resize_window()`, `show_shadow()`, `show_window()`, `toggle_fullscreen_window()`,
`toggle_max_window()`

45 Module *gtk.ui.spin*

45.1 Class *SpinBox*

gtk.VBox —
 gtk.ui.spin.SpinBox

SpinBox.

45.1.1 Methods

<code>__init__(self, value=0, lower=0, upper=100, step=10, default_width=55)</code>	
Initialize <i>SpinBox</i> class.	
Parameters	
value:	Initialize value, default is 0.
lower:	Lower value, default is 0.
upper:	Upper value, default is 100.
step:	Step value, default is 10.
default_width:	Default width, default is 55 pixel.

<code>get_value(self)</code>	
Get current value.	
Return Value	
Return current value.	

<code>set_value(self, value)</code>	
Set value with given value.	
Parameters	
value:	New value.

<code>value_changed(self)</code>	
Emit 'value-changed' signal.	

<code>get_lower(self)</code>	
Get minimum value.	

set_lower (<i>self</i> , <i>value</i>)
Set lower with given value.
Parameters
<i>value</i> : New lower value.

get_upper (<i>self</i>)
Get upper value.

set_upper (<i>self</i> , <i>value</i>)
Set upper with given value.
Parameters
<i>value</i> : New upper value.

get_step (<i>self</i>)
Get step.

set_step (<i>self</i> , <i>value</i>)
Set step with given value.
Parameters
<i>value</i> : New step value.

45.1.2 Class Variables

Name	Description
<code>__signals__</code>	Value: { <code>"value-changed"</code> : (<code>gobject.SIGNAL_RUN_LAST</code> , <code>gobject.TYPE_N...</code> ...

46 Module *gtk.ui.statusbar*

46.1 Class *Statusbar*



Statusbar.

46.1.1 Methods

<code>__init__</code> (<i>self</i> , <i>height</i> , <i>add_separator=False</i>) <hr/> Initialize <i>Statusbar</i> class. Parameters <i>height</i> : <i>Statusbar</i> height. <i>add_separator</i> : Whether add separator between <i>statusbar</i> and window body, default is <code>False</code> . Overrides: <i>dtk.ui.box.EventBox.__init__</i>

47.2.1 Methods

```
__init__(self, title, items, confirm_callback=None, cancel_callback=None,  
window_width=458, window_height=472)
```

Initialize `TabWindow` clas.

Parameters

title: Tab window title.

items: A list of tab item, tab item format:
(tab_name, tab_widget)

confirm_callback: Callback when user click ok button.

cancel_callback: Callback when user click cancel button.

window_width: Default window width.

window_height: Default window height.

Overrides: `dtk.ui.window.Window.__init__`

Inherited from `dtk.ui.dialog.DialogBox`(Section 13.4)

`get_mask_func()`

Inherited from `dtk.ui.window.Window`(Section 60.1)

`add_move_event()`, `add_toggle_event()`, `close_window()`, `draw_mask()`, `get_edge()`,
`get_shadow_size()`, `hide_shadow()`, `is_disable_window_maximized()`, `min_window()`,
`resize_window()`, `show_shadow()`, `show_window()`, `toggle_fullscreen_window()`,
`toggle_max_window()`

48 Module `gtk.ui.talk_view`

48.1 Class `TalkView`



View widget for Deepin Talk.

48.1.1 Methods

<code>__init__(self, right_space=2, top_bottom_space=3)</code>	
Initialize <code>TalkView</code> class.	
Parameters	
<code>right_space</code> :	the space between right border and the vertical scrollbar.
<code>top_bottom_space</code> :	the space between top border and the vertical scrollbar.
Overrides: <code>dtk.ui.scrolled_window.ScrolledWindow.__init__</code>	

Inherited from `dtk.ui.scrolled_window.ScrolledWindow` (Section 41.1)

`add_child()`, `add_with_viewport()`, `get_hadjustment()`, `get_vadjustment()`, `set_hadjustment()`, `set_vadjustment()`

48.2 Class `TalkItem`



Talk item for *TalkView*.

48.2.1 Methods

<code>__init__(self)</code>	
Initialize <code>TalkItem</code> class.	

get_size(*self*)

Get size of talk item.

render(*self*, *cr*, *rect*)

Render talk item.

49 Module *dtk.ui.theme*

49.1 Variables

Name	Description
<code>ui_theme</code>	Value: <code>Theme(os.path.join(get_parent_dir(__file__ 2), "theme"),...</code>

49.2 Class *DynamicColor*



Dynamic color.

49.2.1 Methods

<code>__init__(self, color)</code>
Initialize <i>DynamicColor</i> .
Parameters
color : Initialize color.
Overrides: <code>object.__init__</code>

<code>update(self, color)</code>
Update color.
Parameters
color : Color value.

<code>get_color(self)</code>
Get color.
Return Value
Return current color value.

Inherited from object

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`,
`__reduce__()`, `__reduce_ex__()`, `__repr__()`, `__setattr__()`, `__sizeof__()`,

`__str__()`, `__subclasshook__()`

49.2.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

49.3 Class `DynamicAlphaColor`



Dynamic alpha color.

49.3.1 Methods

<code>__init__(self, color_info)</code>
Initialize <code>DynamicAlphaColor</code> class.
Parameters
<code>color_info</code> : Color information, format as (<code>hex_color</code> , <code>alpha</code>)
Overrides: <code>object.__init__</code>

<code>update(self, color_info)</code>
Update <code>color_info</code> with given value.

<code>get_color_info(self)</code>
Get color info.
Return Value
Return color information, format as (<code>hex_color</code> , <code>alpha</code>)

<code>get_color(self)</code>
Get color.
Return Value
Return hex color string.

get_alpha (<i>self</i>)
Get alpha value.
Return Value Return alpha value.

Inherited from object

```
__delattr__(), __format__(), __getattr__(), __hash__(), __new__(),
__reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(),
__str__(), __subclasshook__()
```

49.3.2 Properties

Name	Description
<i>Inherited from object</i> __class__	

49.4 Class *DynamicShadowColor*

```
object
└─ dtk.ui.theme.DynamicShadowColor
```

Dynamic shadow color.

49.4.1 Methods

__init__ (<i>self</i> , <i>color_info</i>)
Initialize <i>DynamicShadowColor</i> class.
Parameters <i>color_info</i> : Color information, format as:
<pre> >>> [(color_position_1, (hex_color_1, color_alpha_1), >>> (color_position_2, (hex_color_2, color_alpha_2), >>> (color_position_3, (hex_color_3, color_alpha_3))]</pre>
Overrides: object. __init__

49.5.1 Methods

<code>__init__</code> (<i>self</i> , <i>filepath</i>)
Initialize DynamicPixbuf class.
Parameters
<i>filepath</i> : Dynamic pixbuf filepath.
Overrides: <code>object.__init__</code>

<code>update</code> (<i>self</i> , <i>filepath</i>)
Update filepath with given value.
Parameters
<i>filepath</i> : Dynamic pixbuf filepath.

<code>get_pixbuf</code> (<i>self</i>)
Get pixbuf.

Inherited from object

`__delattr__`(), `__format__`(), `__getattr__`(), `__hash__`(), `__new__`(),
`__reduce__`(), `__reduce_ex__`(), `__repr__`(), `__setattr__`(), `__sizeof__`(),
`__str__`(), `__subclasshook__`()

49.5.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

49.6 Class Theme

object —
 `dtk.ui.theme.Theme`

Theme.

49.6.1 Methods

__init__(*self*, *system_theme_dir*, *user_theme_dir*)

Initialize Theme class.

Parameters

- system_theme_dir*: Default theme directory.
- user_theme_dir*: User's theme save directory, generic is
~/.config/project-name/theme

Overrides: object.__init__

load_theme(*self*)

Load theme.

get_theme_file_path(*self*, *filename*)

Get theme file path with given theme name.

Return Value

Return filepath of theme.

get_pixbuf(*self*, *path*)

Get pixbuf with given relative path.

Parameters

- path*: Image relative filepath to theme.

Return Value

Return pixbuf with given relative path.

get_color(*self*, *color_name*)

Get color with given dynamic color.

Parameters

- color_name*: DynamicColor name from theme.txt.

Return Value

Return color with given dynamic color.

get_alpha_color (<i>self</i> , <i>color_name</i>)
Get color with given dynmaic alpha color.
Parameters <i>color_name</i> : DynamicAlphaColor name from theme.txt.
Return Value Return color with given dynamic alpha color.

get_shadow_color (<i>self</i> , <i>color_name</i>)
Get color with given dynmaic shadow color.
Parameters <i>color_name</i> : DynamicShadowColor name from theme.txt.
Return Value Return color with given dynamic shadow color.

change_theme (<i>self</i> , <i>new_theme_name</i>)
Change theme with given new theme name.
Parameters <i>new_theme_name</i> : New theme name.

Inherited from object

__delattr__(), __format__(), __getattr__(), __hash__(), __new__(),
 __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(),
 __str__(), __subclasshook__()

49.6.2 Properties

Name	Description
<i>Inherited from object</i>	
__class__	

50 Module `dtk.ui.thread_pool`

50.1 Functions

<code>clean_cover(filepath)</code>

Clean cover.

50.2 Class `MissionThreadPool`



A class of thread pool.

50.2.1 Methods

<code>__init__(self, concurrent_thread_num=5, clean_delay=0, clean_callback=None, exit_when_finish=False)</code>
--

Initialise the thread pool.

Parameters

<code>concurrent_thread_num</code> :	Max concurrent thread number.
<code>clean_delay</code> :	The time between the finish of the thread and the invocation of thread clean up function.
<code>clean_callback</code> :	The clean up function, which is invoked after the thread is finished.
<code>exit_when_finish</code> :	Indicates whether the thread pool should be destroyed after all mission is finished. By default, it's False.

Overrides: `object.__init__`

run (<i>self</i>)
The thread function.
Overrides: <code>threading.Thread.run</code>

add_missions (<i>self</i> , <i>missions</i>)
Add missions to the thread pool.
Parameters
<i>missions</i> : A list of mission which is of type class <code>MissionThread</code> .

start_mission (<i>self</i> , <i>mission</i>)
Start a specific mission in the thread pool.
Parameters
<i>mission</i> : a mission which is of type class <code>MissionThread</code> .

Inherited from `threading.Thread`

`__repr__()`, `daemon()`, `getName()`, `ident()`, `isAlive()`, `isDaemon()`, `is_alive()`, `join()`, `name()`, `setDaemon()`, `setName()`, `start()`

Inherited from object

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

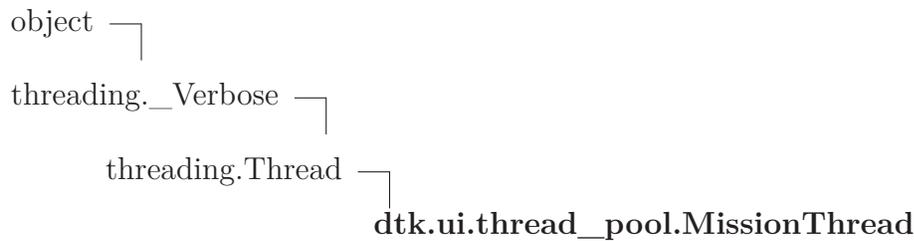
50.2.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

50.2.3 Class Variables

Name	Description
<code>FINISH_SIGNAL</code>	Value: "Finish"

50.3 Class `MissionThread`



This class stands for a single mission in the thread pool.

50.3.1 Methods

<code>__init__(self)</code> <hr/> Initialise the <code>MissionThread</code> . Overrides: <code>object.__init__</code>
--

<code>run(self)</code> <hr/> The thread function. Overrides: <code>threading.Thread.run</code>

<code>start_mission(self)</code> <hr/> The mission thread function of <code>MissionThread</code> . This function is <code>MissionThread</code> template, you should write your own implementation.

<code>get_mission_result(self)</code> <hr/> Return the mission result. This function is <code>MissionThread</code> template, you should write your own implementation. Return Value If you don't want handle result, just return <code>None</code> .
--

Inherited from `threading.Thread`

`__repr__()`, `daemon()`, `getName()`, `ident()`, `isAlive()`, `isDaemon()`, `is_alive()`, `join()`, `name()`, `setDaemon()`, `setName()`, `start()`

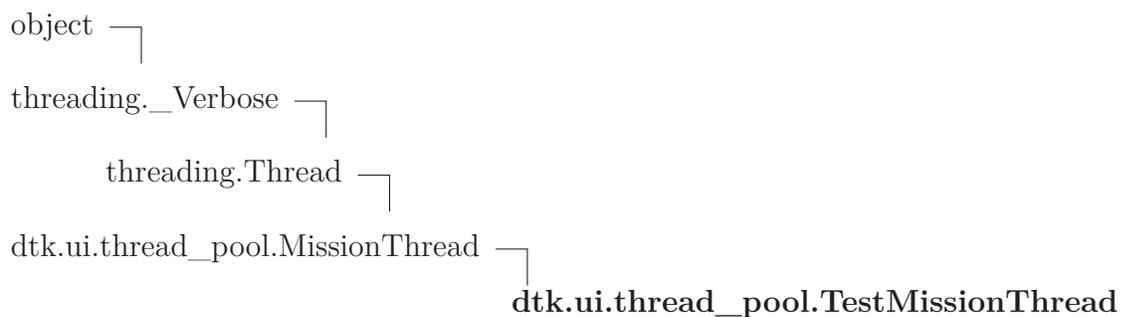
Inherited from `object`

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`,
`__reduce__()`, `__reduce_ex__()`, `__setattr__()`, `__sizeof__()`, `__str__()`,
`__subclasshook__()`

50.3.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

50.4 Class *TestMissionThread*



Test mission thread.

50.4.1 Methods

<code>__init__(self, artist)</code>
Init test mission thread.
Overrides: <code>object.__init__</code>

<code>start_mission(self)</code>
Start mission.
Overrides: <code>dtk.ui.thread_pool.MissionThread.start_mission</code>

get_mission_result (<i>self</i>)
Get mission result.
Return Value
If you don't want handle result, just return None.
Overrides: dtk.ui.thread_pool.MissionThread.get_mission_result

Inherited from dtk.ui.thread_pool.MissionThread(Section 50.3)

run()

Inherited from threading.Thread

__repr__(), daemon(), getName(), ident(), isAlive(), isDaemon(), is_alive(), join(), name(), setDaemon(), setName(), start()

Inherited from object

__delattr__(), __format__(), __getattr__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

50.4.2 Properties

Name	Description
<i>Inherited from object</i>	
__class__	

51 Module `dtk.ui.threads`

51.1 Functions

`post_gui(func)`

Post GUI code in main thread.

You should use `post_gui` wrap graphics function if function call from other threads.

Usage:

```
>>> @post_gui
>>> def graphics_fun():
>>>     ....
```

51.2 Class `AnonymityThread`



Anonymity thread.

51.2.1 Methods

`__init__(self, callback)`

Initialize `AnonymityThread` class.

Parameters

`callback`: Callback run in thread.

Overrides: `object.__init__`

`run(self)`

Run.

Overrides: `threading.Thread.run`

Inherited from threading.Thread

__repr__(), daemon(), getName(), ident(), isAlive(), isDaemon(), is_alive(),
join(), name(), setDaemon(), setName(), start()

Inherited from object

__delattr__(), __format__(), __getattr__(), __hash__(), __new__(),
__reduce__(), __reduce_ex__(), __setattr__(), __sizeof__(), __str__(),
__subclasshook__()

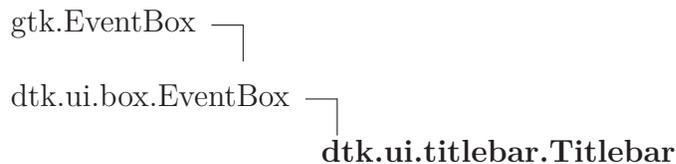
51.2.2 Properties

Name	Description
<i>Inherited from object</i> __class__	

Name	Description
__gsignals__	Value: {'update': (gobject.SIGNAL_RUN_LAST, gobject.TYPE_NONE, (g...

53 Module `gtk.ui.titlebar`

53.1 Class `Titlebar`



`Titlebar` defines every thing of a title bar of a application based on `deepin ui`.

53.1.1 Methods

```

__init__(self, button_mask=["theme", "menu", "max", "min", "close"],
         icon_dpixbuf=None, app_name=None, title=None, add_separator=False,
         height=26, show_title=True)
  
```

Initialize the title bar.

Parameters

- button_mask:** A string list. Each item of it indicates that there is a corresponding button on the title bar. By default, it's ["theme", "menu", "max", "min", "close"], which means theme button, menu button, max button, min button and close button, respectively.
- icon_dpixbuf:** A pixbuf of type `gtk.ui.theme.DynamicPixbuf`. It will be displayed at the top left of the window. By default, it's None.
- app_name:** Application name string. It will be displayed just next to the `icon_dpixbuf`. By default, it's None.
- title:** Title string of the application. It will be displayed on the center of the title bar. By default, it's None.
- add_separator:** If True, add a separation line between the title bar and the body of the window. By default, it's False.
- height:** The hight of the title bar. By default, it's 26 pixels.
- show_title:** If False, the title bar will not be displayed. By default, it's True.

Overrides: `gtk.ui.box.EventBox.__init__`

change_title(*self*, *title*)

Change the title of the application, which is displayed on the center of the title bar.

Parameters

title: New title string that want to set.

54 Module `gtk.ui.tooltip`

54.1 Functions

text(*widget*, *content*, **args*, ***kargs*)

set the tooltip's text content. the "content", "**args*" and "***kargs*" are pass to the `gtk.ui.Label`, so you can change the text's color and some other property.

Parameters

- widget**: the widget of you want to change.
- content**: the text which you want show.
- args**: pass to the `gtk.ui.Label`
- kargs**: pass to the `gtk.ui.Label`

custom(*widget*, *cb*, **args*, ***kargs*)

Set the custom tooltip content.

Parameters

- widget**: the widget of you want to change.
- cb**: the function used to generate the content widget. this function should return an `gtk.Widget`. Be careful: if this function generate it's content affected by other runtime factor, you also should use "always_update" to disable the internal cache mechanism
- args**: pass to the `cb`
- kargs**: pass to the `cb`

show_delay(*widget*, *delay*)

set the time of the tooltip's begin show after pointer stay on the widget.

Parameters

- widget**: the widget of you want to change.
- delay**: the time of start begin show.

hide_delay(*widget*, *delay*)

set the time of the tooltip's start to hide.

Parameters

- widget**: the widget of you want to change.
- delay**: the time of start begin hide.

hide_duration(*widget, delay*)

set the duration of the tooltip's hide effect duration.

Parameters

- widget**: the widget of you want to change.
- delay**: the time of the effect duration.

background(*widget, color*)

set the background of the tooltip's content.

Parameters

- widget**: the widget of you want to change.
- color**: the `gtk.Color` of background.

padding(*widget, t, l, b, r*)

set the padding of the tooltip's content.

Parameters

- widget**: the widget of you want to change.
- t**: the top space
- l**: the left space
- b**: the bottom space
- r**: the right space

has_shadow(*widget, need*)

whether this widget's tooltip need shadow.

Parameters

- widget**: the widget of you want disable tooltip.
- need**: wheter need shadow .

disable(*widget, is_disable*)

disable this widget's tooltip

Parameters

- widget**: the widget of you want disable tooltip.
- is_disable**: wheter disable tooltip.

always_update(*widget*, *need*)

Always create the new tooltip's content, used to show the custom tooltip content generate by function and the function's return widget is different every time be invoked.

Parameters

widget: Gtk.Widget instance.

need: whether always update.

disable_all(*is_disable*)

55 Module `dtk.ui.tooltip_test`

56 Module `gtk.ui.treeview`

56.1 Class `TreeView`

```
gtk.DrawingArea └─ dtk.ui.treeview.TreeView
```

56.1.1 Methods

```
__init__(self, width=20, height=30, font_size=10, font_x_padding=5,
font_width=120, font_height=0, font_align=pango.ALIGN_LEFT,
arrow_x_padding=10,
normal_pixbuf=ui_theme.get_pixbuf("treeview/arrow_right.png"),
press_pixbuf=ui_theme.get_pixbuf("treeview/arrow_down.png"), nor-
mal_hover_pixbuf=ui_theme.get_pixbuf("treeview/arrow_right_hover.png"),
press_hover_pixbuf=ui_theme.get_pixbuf("treeview/arrow_down_hover.png"))
```

```
tree_view_press_event(self, widget, event)
```

```
press_notify_function(self, event)
```

```
set_highlight_index(self, index)
```

```
get_highlight_index(self)
```

```
get_highlight_item(self)
```

```
tree_view_motion_event(self, widget, event)
```

```
get_offset_coordinate(self, widget)
```

Get offset coordinate.

```
draw_mask(self, cr, x, y, w, h)
```

```
tree_view_expose_event(self, widget, event)
```

```
tree_view_key_press_event(self, widget, event)
```

```
tree_view_leave_notify_event(self, widget, event)
```

```
add_items(self, parent_id, child_items)
```

```
add_item(self, parent_id, child_item)
```

```
create_tree(self, child_item)
```

```
scan_item(self, item_id, node)
```

```
clear_scan_save_item(self)
```

```
set_index_text(self, index, text)
```

```
del_item_index(self)
```

```
del_item_from_index(self, index)
```

```
del_item(self, item_id)
```

```
get_other_item(self, index)
```

```
get_item_from_index(self, index)
```

```
get_items(self, parent_id)
```

```
set_text(self, item)
```

```
get_text(self, item)
```

```
clear(self)
```

```
sort_all_nodes(self, nodes)
```

```
get_all_items(self)
```

```
sort(self)
```

```
sort2(self, node, width)
```

56.1.2 Class Variables

Name	Description
<code>__gsignals__</code>	Value: {"single-click-item": (gobject.SIGNAL_RUN_LAST, gobject.TY...

56.2 Class Tree



56.2.1 Methods

<code>__init__(self)</code> x. <code>__init__</code> (...) initializes x; see <code>help(type(x))</code> for signature Overrides: <code>object.__init__</code> <code>exitit</code> (inherited documentation)
--

<code>add_node(self, root_id, node_id, node_item)</code>
--

<code>scan_node(self, root_node, root_id, node_id, node_item)</code>
--

Inherited from object

`__delattr__`() , `__format__`() , `__getattr__`() , `__hash__`() , `__new__`() ,
`__reduce__`() , `__reduce_ex__`() , `__repr__`() , `__setattr__`() , `__sizeof__`() ,
`__str__`() , `__subclasshook__`()

56.2.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

56.3 Class `TreeViewItem`



56.3.1 Methods

```
__init__(self, item_title, has_arrow=True)
```

`x.__init__(...)` initializes `x`; see `help(type(x))` for signature

Overrides: `object.__init__` `exitit`(inherited documentation)

```
get_title(self)
```

```
get_has_arrow(self)
```

```
set_item_id(self, new_id)
```

```
get_item_id(self)
```

Inherited from object

```
__delattr__(), __format__(), __getattr__(), __hash__(), __new__(),
__reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(),
__str__(), __subclasshook__()
```

56.3.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

57 Module `dtk.ui.unique_service`

57.1 Functions

`is_exists`(*app_dbus_name*, *app_object_name*)

Check the program or service is already started by its `app_dbus_name` and `app_object_name`.

Parameters

`app_dbus_name`: the public service name of the service.

`app_object_name`: the public service path of the service.

Return Value

If the service is already on, `True` is returned. Otherwise return `False`.

57.2 Class `UniqueService`

`InterfaceType('Interface', (object,), {})` └─

`dbus.service.Object` └─

`dtk.ui.unique_service.UniqueService`

This class implement a dbus interface, which is used to ensure that the program or service is unique in the system.

57.2.1 Methods

`__init__`(*self*, *bus_name*, *app_dbus_name*, *app_object_name*, *unique_callback=None*)

Initialise the class.

Parameters

`bus_name`: the public service name of the service.

`app_dbus_name`: the public service name of the service.

`app_object_name`: the public service path of the service.

`unique_callback`: the callback which is invoked when the service is found already start. By default, it's `None`.

Overrides: `dbus.service.Object.__init__`

Inherited from `dbus.service.Object`

Introspect(), __dbus_object_path__(), __repr__(), __str__(), add_to_connection(), connection(), locations(), remove_from_connection()

57.2.2 Class Variables

Name	Description
<i>Inherited from dbus.service.Object</i>	SUPPORTS_MULTIPLE_CONNECTIONS, SUPPORTS_MULTIPLE_OBJECT_PATHS

58 Module `gtk.ui.utils`

58.1 Functions

`tree_view_get_toplevel_node_count(treeview)`

Get node count number of `TreeView`.

Parameters

`treeview`: `Gtk.TreeView` instance.

Return Value

Return number of node.

Return 0 if `treeview` haven't model.

`tree_view_get_selected_path(treeview)`

Get selected path of `TreeView`.

Parameters

`treeview`: `Gtk.TreeView` instance.

Return Value

Return selected path of `treeview`.

Return `None` if haven't any path selected.

`tree_view_focus_first_toplevel_node(treeview)`

Focus first toplevel node of `TreeView`.

Parameters

`treeview`: `Gtk.TreeView` instance.

`tree_view_focus_last_toplevel_node(treeview)`

Focus last toplevel node of `TreeView`.

Parameters

`treeview`: `Gtk.TreeView` instance.

`tree_view_scroll_vertical(treeview, scroll_up=True)`

Scroll `TreeView` vertically.

Parameters

`treeview`: `Gtk.TreeView` instance.

`scroll_up`: Defalut value is `True`, set as `False` if you want scroll down.

tree_view_focus_next_toplevel_node(*treeview*)

Focus next toplevel node of `TreeView`.

Parameters

`treeview`: `Gtk.TreeView` instance.

tree_view_focus_prev_toplevel_node(*treeview*)

Focus previous toplevel node of `TreeView`.

Parameters

`treeview`: `Gtk.TreeView` instance.

get_entry_text(*entry*)

Get text of entry.

Parameters

`entry`: `Gtk.Entry` instance.

Return Value

Return text of entry.

set_cursor(*cursor_widget*, *cursor_type=None*)

Set cursor type with given widget.

Parameters

`cursor_widget`: `Gtk.Widget` or `Gdk.Window` instance.

`cursor_type`: The cursor type of `gtk.gdk.Cursor`, please set with `None` if you want reset widget's cursor as default.

Return Value

Always return `False`

set_clickable_cursor(*widget*)

Show `gtk.gdk.HAND2` cursor when mouse hover widget.

Parameters

`widget`: `Gtk.Widget` instance.

set_hover_cursor(*widget*, *cursor_type*)

Set cursor type when mouse hover widget.

Parameters

`widget`: `Gtk.Widget` instance.

`cursor_type`: The cursor type of `gtk.gdk.Cursor`.

get_widget_root_coordinate(*widget*,
pos_type=WIDGET_POS_BOTTOM_CENTER)

Get root coordinate with given widget.

Parameters

widget: Gtk.Widget instance.

pos_type: The position of widget's area, you can set with below constants:

- WIDGET_POS_TOP_LEFT
- WIDGET_POS_TOP_RIGHT
- WIDGET_POS_TOP_CENTER
- WIDGET_POS_BOTTOM_LEFT
- WIDGET_POS_BOTTOM_RIGHT
- WIDGET_POS_BOTTOM_CENTER
- WIDGET_POS_LEFT_CENTER
- WIDGET_POS_RIGHT_CENTER
- WIDGET_POS_CENTER

Return Value

Return (x, y) as root coordination.

get_event_root_coords(*event*)

Get root coordinate with given event.

Parameters

event: Gdk.Event instance, general, we get event instance from gtk signal callback.

Return Value

Return (x, y) as event's root coordination.

get_event_coords(*event*)

Get coordinate with given event.

Parameters

event: Gdk.Event instance, general, we get event instance from gtk signal callback.

Return Value

Return (x, y) as event's coordination.

propagate_expose(*widget, event*)

Propagate expose to children.

General, this function use at last position of 'expose_event' callback to make child redraw after parent widget.

And you must put "return True" after "propagate_expose(widget, event)".

Example:

```
>>> def expose_event_callback(widget, event):
>>>     # Do something.
>>>
>>>     propagate_expose(widget, event)
>>>     return True
```

Parameters

widget: Gtk.Container instance.

This function do nothing if widget is not Gtk.Container instance or haven't any child widget.

event: Gdk.Event instance.

move_window(*widget, event, window*)

Move window with given widget and event.

This function generic use for move window when mouse drag on target widget.

Parameters

widget: Gtk.Widget instance to drag.

event: Gdk.Event instance, generic, event come from gtk signal callback.

window: Gtk.Window instance.

resize_window(*widget, event, window, edge*)

Resize window with given widget and event.

This function generic use for resize window when mouse drag on target widget.

Parameters

widget: Gtk.Widget instance to drag.

event: Gdk.Event instance, generic, event come from gtk signal callback.

window: Gtk.Window instance.

add_in_scrolled_window(*scrolled_window*, *widget*,
shadow_type=`gtk.SHADOW_NONE`)

Add widget in `scrolled_window`.

Wrap function 'add_with_viewport' with shadow type of `Gtk.Viewport`.

Parameters

`scrolled_window`: `Gtk.ScrolledWindow` instance.
`widget`: `Gtk.Widget` instance.
`shadow_type`: Shadow type of `Viewport`, default is `gtk.SHADOW_NONE`.

is_single_click(*event*)

Whether an event is single click event.

Parameters

`event`: `gtk.gdk.BUTTON_PRESS` event.

Return Value

Return True if event is single click event.

is_double_click(*event*)

Whether an event is double click event.

Parameters

`event`: `gtk.gdk.BUTTON_PRESS` event.

Return Value

Return True if event is double click event.

is_left_button(*event*)

Whether event is left button event.

Parameters

`event`: `gtk.gdk.BUTTON_PRESS` event.

Return Value

Return True if event is left button event.

is_right_button(*event*)

Whether event is right button event.

Parameters

event: gtk.gdk.BUTTON_PRESS event.

Return Value

Return True if event is right button event.

is_middle_button(*event*)

Whether event is middle button event.

Parameters

event: gtk.gdk.BUTTON_PRESS event.

Return Value

Return True if event is middle button event.

foreach_container(*widget, callback*)

Make callback call for all children of widget.

Parameters

widget: Gtk.Container instance.

callback: Callback.

foreach_recursive(*container, callback*)

Helper function for *foreach_container*.

Parameters

container: Gtk.Container instance.

callback: Callback.

container_remove_all(*container*)

Handy function to remove all children widget from container.

Parameters

container: Gtk.Container instance.

get_screen_size(*widget*)

Get screen size from the toplevel window associated with widget.

Parameters

widget: Gtk.Widget instance.

Return Value

Return screen size as (screen_width, screen_height)

is_in_rect(*tx, ty, (x, y, w, h)*)

Whether target coordinate in given rectangle.

Parameters

tx: Target x coordinate.

ty: Target y coordinate.

x: X coordinate of rectangle area.

y: Y coordinate of rectangle area.

w: Width of rectangle area.

h: Height of rectangle area.

Return Value

Return True if target coordinate in given rectangle.

scroll_to_top(*scrolled_window*)

Scroll `scrolled_window` to top position.

Parameters

scrolled_window: `Gtk.ScrolledWindow` instance.

scroll_to_bottom(*scrolled_window*)

Scroll `scrolled_window` to bottom position.

Parameters

scrolled_window: `Gtk.ScrolledWindow` instance.

get_content_size(*text, text_size=DEFAULT_FONT_SIZE, text_font=DEFAULT_FONT, wrap_width=None*)

Get text size, in pixel.

Parameters

text: String or markup string.

text_size: Text size, in pixel.

text_font: Text font.

wrap_width: The width of wrap rule, default don't wrap.

Return Value

Return text size as (`text_width`, `text_height`), return (0, 0) if occur error.

create_directory(*directory*, *remove_first=False*)

Create directory.

Parameters

- directory:** Target directory to create.
- remove_first:** If you want remove directory when directory has exist, set it as True.

remove_file(*path*)

Remove file if file exist.

Parameters

- path:** Target path to remove.

remove_directory(*path*)

Remove directory recursively, equivalent to command 'rm -rf path'.

Parameters

- path:** Target directory to remove.

touch_file(*filepath*)

Touch file, equivalent to command 'touch filepath'.

If filepath's parent directory is not exist, this function will create parent directory first.

Parameters

- filepath:** Target path to touch.

read_file(*filepath*, *check_exists=False*)

Read file content.

Parameters

- filepath:** Target filepath.
- check_exists:** Whether check file is exist, default is False.

Return Value

- Return "" if *check_exists* is True and *filepath* not exist.
- Otherwise return file's content.

read_first_line(*filepath*, *check_exists=False*)

Read first line of file.

Parameters

filepath: Target filepath.

check_exists: Whether check file is exist, default is False.

Return Value

Return "" if *check_exists* is True and *filepath* not exist.

Otherwise return file's first line.

eval_file(*filepath*, *check_exists=False*)

Eval file content.

Parameters

filepath: Target filepath.

check_exists: Whether check file is exist, default is False.

Return Value

Return None if *check_exists* is True and file not exist.

Return None if occur error when eval file.

Otherwise return file content as python structure.

write_file(*filepath*, *content*)

Write file with given content.

Parameters

filepath: Target filepath to write.

content: File content to write.

kill_process(*proc*)

Kill process.

Parameters

proc: Subprocess instance.

get_command_output_first_line(*commands*)

Run command and return first line of output.

Parameters

commands: Input commands.

Return Value

Return first line of command output.

get_command_output(*commands*)

Run command and return output.

Parameters

commands: Input commands.

Return Value

Return command output.

run_command(*command*)

Run command silently.

Parameters

command: Input command.

get_os_version()

Get OS version with command 'lsb_release -i'.

Return Value

Return OS version string.

get_current_time(*time_format*="%Y-%m-%d %H:%M:%S")

Get current time with given time format.

Parameters

time_format: Time format, default is %Y-%m-%d %H:%M:%S

Return Value

Return current time with given time format.

add_in_list(*e_list*, *element*)

Add element in list, don't add if element has in list.

Parameters

e_list: List to insert.

element: Element will insert to list.

remove_from_list(*e_list*, *element*)

Try remove element from list, do nothing if element not in list.

Parameters

e_list: List to remove.

element: Element try to remove from list.

sort_alpha(*e_list*)

Sort list with alpha order.

Parameters

`e_list`: List to sort.

get_dir_size(*dirname*)

Get size of given directory.

Parameters

`dirname`: Directory path.

Return Value

Return total size of directory.

print_env()

Print environment variable.

print_exec_time(*func*)

Print execute time of function.

Parameters

`func`: Function name.

Usage:

```
>>> @print_exec_time
>>> def function_to_test():
>>>     ...
```

get_font_families()

Get all font families in system.

Return Value

Return font families list in current system.

format_file_size(*bytes, precision=2*)

Returns a humanized string for a given amount of bytes.

Parameters

`bytes`: Bytes number to format.

`precision`: Number precision.

Return Value

Return a humanized string for a given amount of bytes.

add_color_stop_rgba(*pat, pos, color_info*)

Add color stop as rgba format.

Parameters

pat: Pattern.
pos: Stop position.
color_info: (color, alpha), color is hex value, alpha value range: [0, 1]

alpha_color_hex_to_cairo(*(color, alpha)*)

Convert alpha color (color, alpha) to cairo color (r, g, b, alpha).

Parameters

color: Hex color.
alpha: Alpha value.

Return Value

Return cairo value (red, green, blue, alpha).

color_hex_to_rgb(*color*)

Convert hex color to cairo color (r, g, b).

Parameters

color: Hex color value.

Return Value

Return cairo value, (red, green, blue)

color_hex_to_cairo(*color*)

Convert a html (hex) RGB value to cairo color.

Parameters

color: The color to convert.

Return Value

A color in cairo format, (red, green, blue).

color_rgb_to_hex(*rgb_color*)

Convert cairo color to hex color.

Parameters

rgb_color: (red, green, blue)

Return Value

Return hex color.

color_rgb_to_cairo(*color*)

Convert a 8 bit RGB value to cairo color.

Parameters

`color`: The color to convert.

(*type=a triple of integers between 0 and 255*)

Return Value

A color in cairo format.

get_match_parent(*widget, match_types*)

Get parent widget match given type.

Parameters

`widget`: Gtk.Widget instance.

`match_types`: A list gtk widget types.

Return Value

Return first parent widget match with given types.

Return None if nothing match.

widget_fix_cycle_destroy_bug(*widget*)

Fix bug that PyGtk destroys cycle too early.

Parameters

`widget`: Gtk.Widget instance.

map_value(*value_list, get_value_callback*)

Return value with map list.

Parameters

`value_list`: A list to loop.

`get_value_callback`: Callback for element in list.

Return Value

Return a new list that every element is result of `get_value_callback`.

get_same_level_widgets(*widget*)

Get same type widgets that in same hierarchy level.

Parameters

`widget`: Gtk.Widget instance to search.

Return Value

Return a list that type match given widget at same hierarchy level.

mix_list_max(*list_a*, *list_b*)

Return new list that element is max value between *list_a* and *list_b*.

Parameters

list_a: List a.

list_b: List b.

Return Value

Return new list that element is max value between two list.

Return empty list if any input list is empty or two list's length is not same.

unzip(*unzip_list*)

Unzip [(1, 'a'), (2, 'b'), (3, 'c')] to ([1, 2, 3], ['a', 'b', 'c']).

Parameters

unzip_list: List to unzip.

Return Value

Return new unzip list.

is_seriate_list(*test_list*)

Whether is seriate list.

Parameters

test_list: Test list.

Return Value

Return True is test list is seriate list.

get_disperse_index(*disperse_list*, *value*)

Get index in disperse list.

Parameters

disperse_list: Disperse list.

value: Match value.

Return Value

Return index in disperse list.

window_is_max(*widget*)

Whether window is maximized.

Parameters

`widget`: `Gtk.Widget` instance.

Return Value

Return True if widget's toplevel window is maximized.

last_index(*test_list*)

Return last index of list.

Parameters

`test_list`: Test list.

Return Value

Return last index of list.

cairo_state(*cr*)

Protected cairo context state for operate cairo safety.

Parameters

`cr`: Cairo context.

cairo_disable_antialias(*cr*)

Disable cairo antialias temporary.

Parameters

`cr`: Cairo context.

exec_time()

Print execute time with given code block.

Usage:

```
>>> with exec_time():
>>>     # Write any code at here.
>>>     # ...
```

remove_timeout_id(*callback_id*)

Remove callback id.

Parameters

`callback_id`: Callback id.

remove_signal_id(*signal_id*)

Remove signal id.

Parameters

`signal_id`: Signal id that return by function `gobject.connect`.

print_callback_args(**args*)

Print callback arguments.

Usage:

```
>>> some_widget.connect("signal", print_callback_args)
```

enable_shadow(*widget*)

Whether widget is support composited.

Parameters

`widget`: `Gtk.Widget` instance.

Return Value

Return True if widget is support composited.

rgb2hsb(*r_value*, *g_value*, *b_value*)

Convert color from RGB to HSB format.

Parameters

`r_value`: Red.

`g_value`: Green.

`b_value`: Blue.

Return Value

Return color with HSB (h, s, b) format.

find_similar_color(*search_color*)

Find simliar color match `search_color`.

Parameters

`search_color`: Color to search.

Return Value

Return similar color name and value, (`color_name`, `color_value`).

end_with_suffixs(*filepath, suffixs*)

Whether file ends with given suffixs.

Parameters

filepath: Filepath to test.
suffixs: A list suffix to match.

Return Value

Return True if filepath ends with with given suffixs.

place_center(*refer_window, place_window*)

Place *place_window* in center of *refer_window*.

Parameters

refer_window: Reference window.
place_window: Place window.

get_pixbuf_support_formats()

Get formats that support by pixbuf.

Return Value

Return formats that support by pixbuf.

get_parent_dir(*filepath, level=1*)

Get parent directory with given return level.

Parameters

filepath: Filepath.
level: Return level, default is 1

Return Value

Return parent directory with given return level.

gdkcolor_to_string(*gdkcolor*)

Gdk color to string.

Parameters

gdkcolor: Gdk.Color

Return Value

Return string of gdk color.

is_long(*string*)

Is string can convert to long type.

Parameters

`string`: Test string.

Return Value

Return True if string can convert to long type.

is_int(*string*)

Is string can convert to int type.

Parameters

`string`: Test string.

Return Value

Return True if string can convert to int type.

is_float(*string*)

Is string can convert to float type.

Parameters

`string`: Test string.

Return Value

Return True if string can convert to float type.

is_hex_color(*string*)

Is string can convert to hex color type.

Parameters

`string`: Test string.

Return Value

Return True if string can convert to hex color type.

get_window_shadow_size(*window*)

Get window shadow size.

Parameters

`window`: Test window.

Return Value

Return shadow size as (width, height), or return (0, 0) if window haven't shadow.

layout_set_markup(*layout, markup*)

Set layout markup.

Parameters

- `layout`: Pango layout.
- `markup`: Markup string.

get_optimum_pixbuf_from_file(*filepath, expect_width, expect_height, cut_middle_area=True*)

Get optimum size pixbuf from file.

Parameters

- `filepath`: Filepath to contain image.
- `expect_width`: Expect width.
- `expect_height`: Expect height.
- `cut_middle_area`: Default cut image with middle area.

Return Value

Return optimum size pixbuf with expect size.

unique_print(*text*)

Unique print, generic for test code.

Parameters

- `text`: Test text.

check_connect_by_port(*port, retry_times=6, sleep_time=0.5*)

Check connect has active with given port.

Parameters

- `port`: Test port.
- `retry_times`: Retry times.
- `sleep_time`: Sleep time between retry, in seconds.

Return Value

Return True if given port is active.

is_network_connected()

Is network connected, if nothing in file `/proc/net/arp`, network is disconnected.

Return Value

Return True if network is connected.

59 Module `gtk.ui.volume_button`

59.1 Variables

Name	Description
<code>ZERO_STATE</code>	Value: 0
<code>MIN_STATE</code>	Value: 1
<code>MID_STATE</code>	Value: 2
<code>MAX_STATE</code>	Value: 3
<code>MUTE_STATE</code>	Value: -1
<code>MOUSE_VOLUME_STATE_PRESS</code>	Value: 1
<code>MOUSE_VOLUME_STATE_HOVER</code>	Value: 2
<code>MOUSE_VOLUME_STATE_NORMAL</code>	Value: -1
<code>VOLUME_RIGHT</code>	Value: "right"
<code>VOLUME_LEFT</code>	Value: "left"

59.2 Class `VolumeButton`

`gtk.Button` —
`gtk.ui.volume_button.VolumeButton`

Volume button.

59.2.1 Methods

```
__init__(self, volume_max_value=100, volume_width=52, volume_x=0,
volume_y=15, line_width=3, volume_left_right_padding_x=5,
volume_left_show_value=[(1,33), (34,66), (67,100)], scroll_bool=False,
press_emit_bool=False,
bg_pixbuf=ui_theme.get_pixbuf("volumebutton/bg.png"),
fg_pixbuf=ui_theme.get_pixbuf("volumebutton/fg.png"),
zero_volume_normal_pixbuf=ui_theme.get_pixbuf("volumebutton/zero_normal.png"),
zero_volume_hover_pixbuf=ui_theme.get_pixbuf("volumebutton/zero_hover.png"),
zero_volume_press_pixbuf=ui_theme.get_pixbuf("volumebutton/zero_press.png"),
min_volume_normal_pixbuf=ui_theme.get_pixbuf("volumebutton/lower_normal.png"),
min_volume_hover_pixbuf=ui_theme.get_pixbuf("volumebutton/lower_hover.png"),
min_volume_press_pixbuf=ui_theme.get_pixbuf("volumebutton/lower_press.png"),
mid_volume_normal_pixbuf=ui_theme.get_pixbuf("volumebutton/middle_normal.png"),
mid_volume_hover_pixbuf=ui_theme.get_pixbuf("volumebutton/middle_hover.png"),
mid_volume_press_pixbuf=ui_theme.get_pixbuf("volumebutton/middle_press.png"),
max_volume_normal_pixbuf=ui_theme.get_pixbuf("volumebutton/high_normal.png"),
max_volume_hover_pixbuf=ui_theme.get_pixbuf("volumebutton/high_hover.png"),
max_volume_press_pixbuf=ui_theme.get_pixbuf("volumebutton/high_press.png"),
mute_volume_normal_pixbuf=ui_theme.get_pixbuf("volumebutton/mute_normal.png"),
mute_volume_hover_pixbuf=ui_theme.get_pixbuf("volumebutton/mute_hover.png"),
mute_volume_press_pixbuf=ui_theme.get_pixbuf("volumebutton/mute_press.png"),
point_volume_pixbuf=ui_theme.get_pixbuf("volumebutton/point_normal.png"),
inc_value=5)
```

```
set_press_emit_bool(self, emit_bool)
```

```
volume_other_set_value(self, volume_type)
```

```
volume_state(self)
```

```
set_volume_left_show_value(self, show_value)
```

```
set_volume_mute(self, mute_flag=True)
```

```
line_width(self)
```

```
value(self)
```

```
set_volume_position(self, x, y)
```

<code>max_value(self)</code>

59.2.2 Class Variables

Name	Description
<code>__gsignals__</code>	Value: {"volume-state-changed": (gobject.SIGNAL_RUN_LAST, gobject...

59.2.3 Instance Variables

Name	Description
<code>inc_value</code>	Init VolumeButton event.

60 Module `gtk.ui.window`

60.1 Class `Window`

`gtk.Window` —
 `gtk.ui.window.Window`

The `Window` class is a subclass of `gtk.Window`. It adds some features that `deepin-ui` have to `gtk.Window`.

60.1.1 Methods

<pre>__init__(<i>self</i>, <i>enable_resize</i>=False, <i>shadow_radius</i>=6, <i>window_type</i>=gtk.WINDOW_TOPLEVEL, <i>shadow_visible</i>=True)</pre>
Initialise the <code>Window</code> class.
Parameters
<p><code>enable_resize</code>: If True, the window will be set resizable. By default, it's False.</p> <p><code>shadow_radius</code>: The radius of the shadow. By default, it's 6.</p> <p><code>window_type</code>: A flag of type <code>gtk.WindowType</code>, which indicates the type of the window. By default, it's <code>gtk.WINDOW_TOPLEVEL</code>.</p> <p><code>shadow_visible</code>: If True, the shadow is visible. By default, it's True, just disable when your program not allow manipulate colormap, such as <code>mplayer</code>.</p>

<pre>show__window(<i>self</i>)</pre>
Show the window.

<pre>draw__mask(<i>self</i>, <i>cr</i>, <i>x</i>, <i>y</i>, <i>w</i>, <i>h</i>)</pre>
Draw mask interface, you should implement this function own.
Parameters
<p><code>cr</code>: Cairo context.</p> <p><code>x</code>: X coordinate of draw area.</p> <p><code>y</code>: Y coordinate of draw area.</p> <p><code>w</code>: Width of draw area.</p> <p><code>h</code>: Height of draw area.</p>

hide_shadow(*self*)

Hide the window shadow.

show_shadow(*self*)

Show the window shadow.

min_window(*self*)

Minimize the window. Make it iconified.

toggle_max_window(*self*)

Toggle the window size between maximized size and normal size.

toggle_fullscreen_window(*self*)

Toggle the window between fullscreen mode and normal size.

close_window(*self*)

Close the window. Send the destroy signal to the program.

Return Value

Always return False.

resize_window(*self*, *widget*, *event*)

Resize the window.

Parameters

widget: The window of type `gtk.Widget`.

event: A signal of type `gtk.gdk.Event`.

is_disable_window_maximized(*self*)

An interface which indicates whether the window could be maximized, you should implement this function you own.

Return Value

Always return False.

add_move_event(*self*, *widget*)

Add move event callback.

Parameters

widget: A widget of type `gtk.Widget`.

add_toggle_event(*self*, *widget*)

Add toggle event callback.

Parameters

widget: A widget of type `gtk.Widget`.

get_edge(*self*)

Get the edge which the cursor is on, according to the cursor type.

Return Value

If there is a corresponding cursor type, an instance of `gtk.gdk.WindowEdge` is returned, else `None` is returned.

get_shadow_size(*self*)

Get the shadow size.

Return Value

return the shadow size or (0, 0)

Index

- dtk (*package*)
 - dtk.ui (*package*), 2–3
 - dtk.ui.animation (*module*), 4–6
 - dtk.ui.application (*module*), 7–9
 - dtk.ui.box (*module*), 10–11
 - dtk.ui.browser (*module*), 12
 - dtk.ui.button (*module*), 13–20
 - dtk.ui.cache_pixbuf (*module*), 21–22
 - dtk.ui.categorybar (*module*), 23–24
 - dtk.ui.color_selection (*module*), 25–29
 - dtk.ui.combo (*module*), 30–31
 - dtk.ui.config (*module*), 32–33
 - dtk.ui.constant (*module*), 34–35
 - dtk.ui.dialog (*module*), 36–42
 - dtk.ui.dominant_color (*module*), 43
 - dtk.ui.draw (*module*), 44–50
 - dtk.ui.droplist (*module*), 51–56
 - dtk.ui.entry (*module*), 57–65
 - dtk.ui.frame (*module*), 66–67
 - dtk.ui.global_key (*module*), 68–70
 - dtk.ui.group (*module*), 71–73
 - dtk.ui.iconview (*module*), 74–78
 - dtk.ui.keymap (*module*), 79–80
 - dtk.ui.label (*module*), 81–82
 - dtk.ui.line (*module*), 83
 - dtk.ui.listview (*module*), 84–94
 - dtk.ui.listview_preview_pixbuf (*module*), 95
 - dtk.ui.locales (*module*), 96
 - dtk.ui.mask (*module*), 97–98
 - dtk.ui.menu (*module*), 99–102
 - dtk.ui.mplayer_view (*module*), 103
 - dtk.ui.mplayer_window (*module*), 104–106
 - dtk.ui.navigatebar (*module*), 107–109
 - dtk.ui.new_treeview (*module*), 110–111
 - dtk.ui.notebook (*module*), 112–113
 - dtk.ui.osd_tooltip (*module*), 114–115
 - dtk.ui.paned (*module*), 116–117
 - dtk.ui.panel (*module*), 118
 - dtk.ui.progressbar (*module*), 119
 - dtk.ui.pseudo_skin (*module*), 120
 - dtk.ui.scalebar (*module*), 121–122
 - dtk.ui.scrolled_window (*module*), 123–124
 - dtk.ui.skin (*module*), 125
 - dtk.ui.skin_config (*module*), 126–127
 - dtk.ui.slider (*module*), 128–130
 - dtk.ui.spin (*module*), 131–132
 - dtk.ui.statusbar (*module*), 133
 - dtk.ui.tab_window (*module*), 134–135
 - dtk.ui.talk_view (*module*), 136–137
 - dtk.ui.theme (*module*), 138–144
 - dtk.ui.thread_pool (*module*), 145–149
 - dtk.ui.threads (*module*), 150–151
 - dtk.ui.timeline (*module*), 152–153
 - dtk.ui.titlebar (*module*), 154–155
 - dtk.ui.tooltip (*module*), 156–158
 - dtk.ui.tooltip_test (*module*), 159
 - dtk.ui.treeview (*module*), 160–163
 - dtk.ui.unique_service (*module*), 164–165
 - dtk.ui.utils (*module*), 166–184
 - dtk.ui.volume_button (*module*), 185–187
 - dtk.ui.window (*module*), 188–190