

pygtkChart

API Documentation

July 11, 2009

Contents

Contents	1
1 Package pygtk_chart	2
1.1 Modules	2
1.2 Variables	2
2 Module pygtk_chart.bar_chart	3
2.1 Variables	3
2.2 Class Bar	3
2.2.1 Methods	3
2.2.2 Properties	5
2.2.3 Class Variables	5
2.3 Class BarChart	6
2.3.1 Methods	6
2.3.2 Properties	10
2.3.3 Class Variables	10
2.4 Class MultiBar	11
2.4.1 Methods	11
2.4.2 Properties	12
2.4.3 Class Variables	12
2.5 Class MultiBarChart	13
2.5.1 Methods	14
2.5.2 Properties	16
2.5.3 Class Variables	16
3 Module pygtk_chart.basics	17
3.1 Functions	17
3.2 Variables	18
4 Module pygtk_chart.chart	19
4.1 Class Chart	19
4.1.1 Methods	20
4.1.2 Properties	23
4.1.3 Class Variables	23
4.2 Class ChartObject	23
4.2.1 Methods	24
4.2.2 Properties	25

4.2.3	Class Variables	25
4.3	Class Background	25
4.3.1	Methods	26
4.3.2	Properties	27
4.3.3	Class Variables	27
4.4	Class Title	28
4.4.1	Methods	28
4.4.2	Properties	29
4.4.3	Class Variables	29
5	Module pygtk_chart.line_chart	30
5.1	Functions	30
5.2	Variables	31
5.3	Class RangeCalculator	32
5.3.1	Methods	32
5.4	Class LineChart	33
5.4.1	Methods	33
5.4.2	Properties	36
5.4.3	Class Variables	37
5.5	Class Axis	37
5.5.1	Methods	37
5.5.2	Properties	40
5.5.3	Class Variables	40
5.6	Class XAxis	40
5.6.1	Methods	40
5.6.2	Properties	41
5.6.3	Class Variables	41
5.7	Class YAxis	42
5.7.1	Methods	42
5.7.2	Properties	43
5.7.3	Class Variables	43
5.8	Class Grid	43
5.8.1	Methods	44
5.8.2	Properties	45
5.8.3	Class Variables	45
5.9	Class Graph	46
5.9.1	Methods	46
5.9.2	Properties	51
5.9.3	Class Variables	51
6	Module pygtk_chart.pie_chart	52
6.1	Variables	52
6.2	Class PieArea	52
6.2.1	Methods	52
6.2.2	Properties	54
6.2.3	Class Variables	54
6.3	Class PieChart	55
6.3.1	Methods	55
6.3.2	Properties	60
6.3.3	Class Variables	60

1 Package pygtk_chart

This package contains four pygtk widgets for drawing simple charts:

- `line_chart.LineChart` for line charts,
- `pie_chart.PieChart` for pie charts,
- `bar_chart.BarChart` for bar charts,
- `bar_chart.MultiBarChart` for charts with groups of bars.

Version: beta

Author: Sven Festersen, John Dickinson

License: GPL

1.1 Modules

- **bar_chart:** Contains the BarChart widget.
(Section 2, p. 3)
- **basics:** This module contains simple functions needed by all other modules.
(Section 3, p. 17)
- **chart:** This is the main module.
(Section 4, p. 19)
- **line_chart:** Contains the LineChart widget.
(Section 5, p. 30)
- **pie_chart:** Contains the PieChart widget.
(Section 6, p. 52)

1.2 Variables

Name	Description
<code>--url--</code>	Value: <code>'http://pygtkchart.sven-festersen.de'</code>

2 Module pygtk_chart.bar_chart

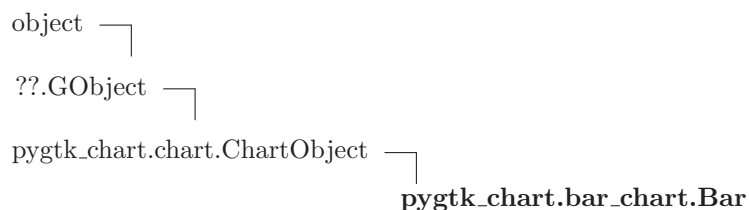
Contains the BarChart widget.

Author: John Dickinson (john@johnandkaren.com)

2.1 Variables

Name	Description
COLOR_AUTO	Value: 0
COLORS	Value: [(0.8, 0.0, 0.0), (0.203921568627, 0.396078431373, 0.6431...]

2.2 Class Bar



2.2.1 Methods

__init__(self, name, value, label='')

x.__init__(...) initializes x; see x.__class__.__doc__ for signature

Overrides: object.__init__ extit(inherited documentation)

do_get_property(self, property)

Overrides: pygtk_chart.chart.ChartObject.do_get_property

do_set_property(self, property, value)

Overrides: pygtk_chart.chart.ChartObject.do_set_property

set_value(self, value)

Set the value of the Bar.

Parameters

value: (type=float.)

get_value(*self*)

Returns the current value of the Bar.

Return Value

float.

set_color(*self*, *color*)

Set the color of the bar. Color has to either COLOR_AUTO or a tuple (r, g, b) with r, g, b in [0, 1].

Parameters**color:** (*type=a color.*)**get_color**(*self*)

Returns the current color of the bar or COLOR_AUTO.

Return Value

a color.

set_label(*self*, *label*)

Set the label for the bar chart bar.

Parameters**label:** the new label*(type=string.)***get_label**(*self*)

Returns the current label of the bar.

Return Value

string.

Inherited from pygtk_chart.chart.ChartObject(Section 4.2)

draw(), get_antialias(), get_visible(), set_antialias(), set_visible()

Inherited from ??GObject

__cmp__(), __copy__(), __deepcopy__(), __delattr__(), __gdoc__(), __gobject_init__(), __hash__(), __new__(), __repr__(), __setattr__(), chain(), connect(), connect_after(), connect_object(), connect_object_after(), disconnect(), disconnect_by_func(), emit(), emit_stop_by_name(), freeze_notify(), get_data(), get_properties(), get_property(), handler_block(), handler_block_by_func(), handler_disconnect(), handler_is_connected(), handler_unblock(), handler_unblock_by_func(), notify(), props(), set_data(), set_properties(), set_property(), stop_emission(), thaw_notify(), weak_ref()

Inherited from object

__getattr__(), __reduce__(), __reduce_ex__(), __str__()

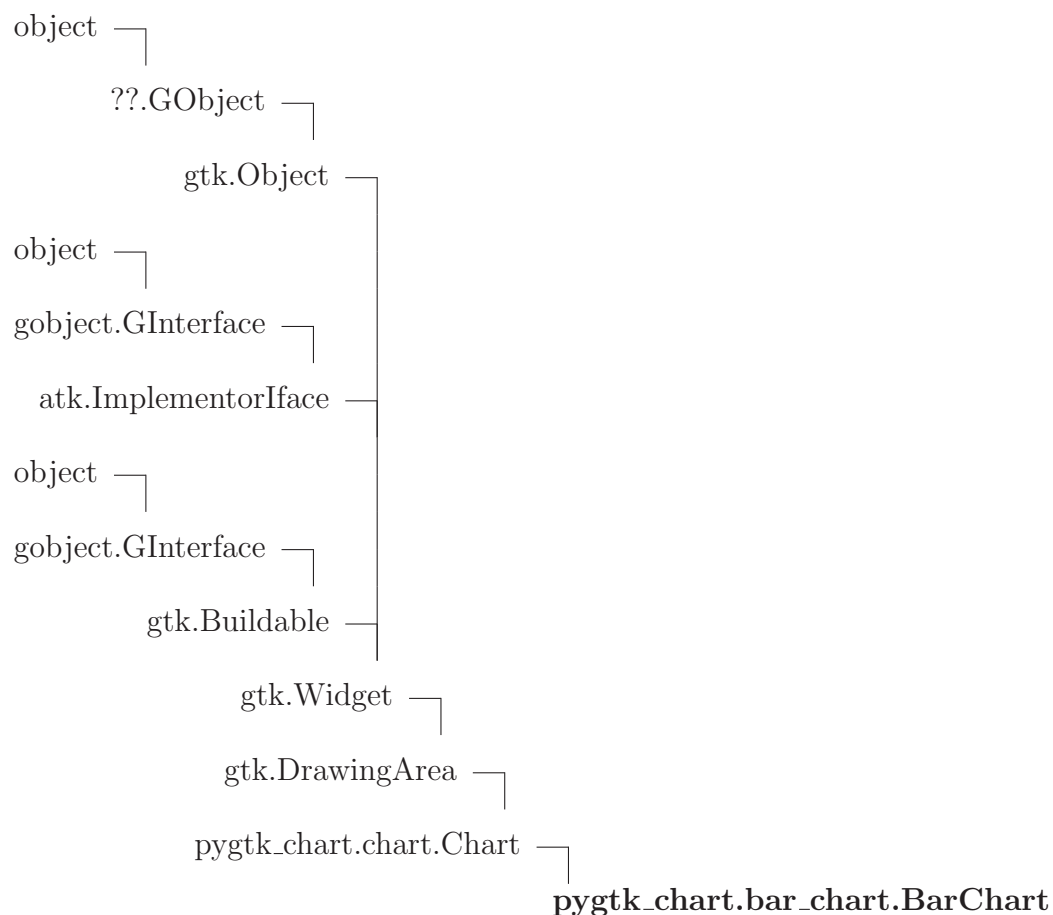
2.2.2 Properties

Name	Description
<i>Inherited from <code>??GObject</code></i>	
<code>--grefcount--</code>	
<i>Inherited from object</i>	
<code>--class--</code>	

2.2.3 Class Variables

Name	Description
<code>--gproperties--</code>	Value: <code>{"name":(gobject.TYPE_STRING, "bar name", "A unique name ...</code>
<code>--gtype--</code>	Value: <code><GType pygtk_chart+bar_chart+Bar (171192984)></code>
<i>Inherited from <code>pygtk_chart.chart.ChartObject</code> (Section 4.2)</i>	
<code>--gsignals--</code>	

2.3 Class **BarChart**



Known Subclasses: `pygtk_chart.bar_chart.MultiBarChart`

2.3.1 Methods

`__init__(self)`

`x.__init__(...)` initializes `x`; see `x.__class__.__doc__` for signature

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

`do_get_property(self, property)`

`do_set_property(self, property, value)`

draw(*self*, *context*)

Draw the widget. This method is called automatically. Don't call it yourself. If you want to force a redrawing of the widget, call the `queue_draw()` method.

Parameters

context: The context to draw on.

(*type=cairo.Context*)

Overrides: `gtk.Widget.draw`

add_bar(*self*, *bar*)**get_bar**(*self*, *name*)

Returns the Bar with the id 'name' if it exists, None otherwise.

Parameters

name: the id of a Bar

(*type=string*)

Return Value

Bar or None.

set_draw_labels(*self*, *draw*)

Set whether to draw the labels of the bars.

Parameters

draw: (*type=boolean.*)

get_draw_labels(*self*)

Returns True if bar labels are shown.

Return Value

boolean.

set_enable_mouseover(*self*, *mouseover*)

Set whether a mouseover effect should be shown when the pointer enters a bar.

Parameters

mouseover: (*type=boolean.*)

get_enable_mouseover (<i>self</i>)

Returns True if the mouseover effect is enabled.
--

Return Value boolean.

set_show_values (<i>self</i> , <i>show</i>)
--

Set whether the bar's value should be shown in its label.

Parameters <i>show</i> : (<i>type=boolean.</i>)

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

Returns True if the value of a bar is shown in its label.

Return Value boolean.

Inherited from pygtk_chart.chart.Chart(Section 4.1)

draw_basics(), export_png(), export_svg(), expose()

Inherited from gtk.DrawingArea

size()

Inherited from gtk.Widget

activate(), add_accelerator(), add_events(), add_mnemonic_label(), can_activate_accel(), child_focus(), child_notify(), class_path(), create_pango_context(), create_pango_layout(), destroy(), do_button_press_event(), do_button_release_event(), do_can_activate_accel(), do_client_event(), do_composited_changed(), do_configure_event(), do_delete_event(), do_destroy_event(), do_direction_changed(), do_drag_begin(), do_drag_data_delete(), do_drag_data_get(), do_drag_data_received(), do_drag_drop(), do_drag_end(), do_drag_leave(), do_drag_motion(), do_enter_notify_event(), do_event(), do_expose_event(), do_focus(), do_focus_in_event(), do_focus_out_event(), do_get_accessible(), do_grab_broken_event(), do_grab_focus(), do_grab_notify(), do_hide(), do_hide_all(), do_hierarchy_changed(), do_key_press_event(), do_key_release_event(), do_leave_notify_event(), do_map(), do_map_event(), do_mnemonic_activate(), do_motion_notify_event(), do_no_expose_event(), do_parent_set(), do_popup_menu(), do_property_notify_event(), do_proximity_in_event(), do_proximity_out_event(), do_realize(), do_screen_changed(), do_scroll_event(), do_selection_clear_event(), do_selection_get(), do_selection_notify_event(), do_selection_received(), do_selection_request_event(), do_show(), do_show_all(), do_show_help(), do_size_allocate(), do_size_request(), do_state_changed(), do_style_set(), do_unmap(), do_unmap_event(), do_unrealize(), do_visibility_notify_event(), do_window_state_event(), drag_begin(), drag_check_threshold(), drag_dest.add_image_targets(), drag_dest.add_text_targets(), drag_dest.add_uri_targets(), drag_dest.find_target(),

drag_dest_get_target_list(), drag_dest_get_track_motion(), drag_dest_set(), drag_dest_set_proxy(),
 drag_dest_set_target_list(), drag_dest_set_track_motion(), drag_dest_unset(), drag_get_data(),
 drag_highlight(), drag_source_add_image_targets(), drag_source_add_text_targets(),
 drag_source_add_uri_targets(), drag_source_get_target_list(), drag_source_set(), drag_source_set_icon(),
 drag_source_set_icon_name(), drag_source_set_icon_pixbuf(), drag_source_set_icon_stock(),
 drag_source_set_target_list(), drag_source_unset(), drag_unhighlight(), ensure_style(),
 error_bell(), event(), freeze_child_notify(), get_accessible(), get_action(), get_activate_signal(),
 get_allocation(), get_ancestor(), get_child_requisition(), get_child_visible(), get_clipboard(),
 get_colormap(), get_composite_name(), get_direction(), get_display(), get_events(),
 get_extension_events(), get_has_tooltip(), get_modifier_style(), get_name(), get_no_show_all(),
 get_pango_context(), get_parent(), get_parent_window(), get_pointer(), get_root_window(),
 get_screen(), get_settings(), get_size_request(), get_snapshot(), get_style(), get_tooltip_markup(),
 get_tooltip_text(), get_tooltip_window(), get_toplevel(), get_visual(), get_window(),
 grab_add(), grab_default(), grab_focus(), grab_remove(), has_screen(), hide(), hide_all(),
 hide_on_delete(), input_shape_combine_mask(), intersect(), is_ancestor(), is_composited(),
 is_focus(), keynav_failed(), list_mnemonic_labels(), map(), menu_get_for_attach_widget(),
 mnemonic_activate(), modify_base(), modify_bg(), modify_cursor(), modify_fg(),
 modify_font(), modify_style(), modify_text(), path(), queue_clear(), queue_clear_area(),
 queue_draw(), queue_draw_area(), queue_resize(), queue_resize_no_redraw(), rc_get_style(),
 realize(), region_intersect(), remove_accelerator(), remove_mnemonic_label(), ren-
 der_icon(), reparent(), reset_rc_styles(), reset_shapes(), selection_add_target(), se-
 lection_add_targets(), selection_clear_targets(), selection_convert(), selection_owner_set(),
 selection_remove_all(), send_expose(), set_accel_path(), set_activate_signal(), set_app_paintable(),
 set_child_visible(), set_colormap(), set_composite_name(), set_direction(), set_double_buffered(),
 set_events(), set_extension_events(), set_has_tooltip(), set_name(), set_no_show_all(),
 set_parent(), set_parent_window(), set_redraw_on_allocate(), set_scroll_adjustments(),
 set_sensitive(), set_set_scroll_adjustments_signal(), set_size_request(), set_state(), set_style(),
 set_tooltip_markup(), set_tooltip_text(), set_tooltip_window(), set_ufposition(), set_usize(),
 shape_combine_mask(), show(), show_all(), show_now(), size_allocate(), size_request(),
 style_get_property(), thaw_child_notify(), translate_coordinates(), trigger_tooltip_query(),
 unmap(), unparent(), unrealize()

Inherited from gtk.Object

do_destroy(), flags(), remove_data(), remove_no_notify(), set_flags(), unset_flags()

Inherited from ??GObject

__cmp__(), __copy__(), __deepcopy__(), __delattr__(), __gdoc__(), __gobject_init__(),
 __hash__(), __new__(), __repr__(), __setattr__(), chain(), connect(), connect_after(),
 connect_object(), connect_object_after(), disconnect(), disconnect_by_func(), emit(),
 emit_stop_by_name(), freeze_notify(), get_data(), get_properties(), get_property(),
 handler_block(), handler_block_by_func(), handler_disconnect(), handler_is_connected(),
 handler_unblock(), handler_unblock_by_func(), notify(), props(), set_data(), set_properties(),
 set_property(), stop_emission(), thaw_notify(), weak_ref()

Inherited from atk.ImplementorIface

ref_accessible()

Inherited from gtk.Buildableadd_child(), construct_child(), do_add_child(), do_construct_child(), do_get_internal_child(),
do_parser_finished(), do_set_name(), get_internal_child(), parser_finished()***Inherited from object***

__getattr__(), __reduce__(), __reduce_ex__(), __str__()

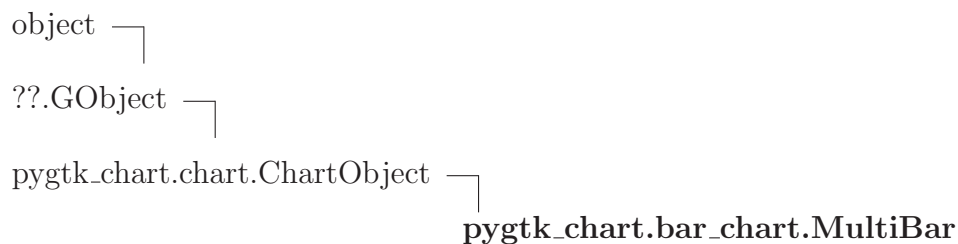
2.3.2 Properties

Name	Description
<i>Inherited from gtk.Widget</i>	allocation, name, parent, requisition, saved_state, state, style, window
<i>Inherited from ??GObject</i>	__grefcount__
<i>Inherited from object</i>	__class__

2.3.3 Class Variables

Name	Description
__gproperties__	Value: { "draw-labels": (gobject.TYPE_BOOLEAN, "draw bar labels", ...
__gsignals__	Value: { "bar-clicked": (gobject.SIGNAL_RUN_LAST, gobject.TYPE_NON...
__gtype__	Value: <GType pygtk_chart+bar_chart+BarChart (166919632)>

2.4 Class MultiBar



2.4.1 Methods

__init__(*self*, *name*, *label*='')

x.__init__(...) initializes *x*; see *x*.__class__.__doc__ for signature

Overrides: object.__init__ extit(inherited documentation)

do_get_property(*self*, *property*)

Overrides: pygtk_chart.chart.ChartObject.do_get_property

do_set_property(*self*, *property*, *value*)

Overrides: pygtk_chart.chart.ChartObject.do_set_property

get_value(*self*)

Returns the maximum value of the MultiBar.

Return Value

float.

set_label(*self*, *label*)

Set the label for the bar chart bar.

Parameters

label: the new label

(*type*=string.)

get_label(*self*)

Returns the current label of the bar.

Return Value

string.

add_bar (<i>self</i> , <i>bar</i>)

get_bar (<i>self</i> , <i>name</i>)
--

Returns the Bar with the id 'name' if it exists, None otherwise.

Parameters

name: the id of a Bar
(*type=string*)

Return Value

Bar or None.

Inherited from *pygtk_chart.chart.ChartObject*(Section 4.2)

draw(), get_antialias(), get_visible(), set_antialias(), set_visible()

Inherited from *??GObject*

__cmp__(), __copy__(), __deepcopy__(), __delattr__(), __gdoc__(), __gobject_init__(),
__hash__(), __new__(), __repr__(), __setattr__(), chain(), connect(), connect_after(),
connect_object(), connect_object_after(), disconnect(), disconnect_by_func(), emit(),
emit_stop_by_name(), freeze_notify(), get_data(), get_properties(), get_property(),
handler_block(), handler_block_by_func(), handler_disconnect(), handler_is_connected(),
handler_unblock(), handler_unblock_by_func(), notify(), props(), set_data(), set_properties(),
set_property(), stop_emission(), thaw_notify(), weak_ref()

Inherited from *object*

__getattribute__(), __reduce__(), __reduce_ex__(), __str__()

2.4.2 Properties

Name	Description
<i>Inherited from <i>??GObject</i></i>	
__grefcount__	
<i>Inherited from <i>object</i></i>	
__class__	

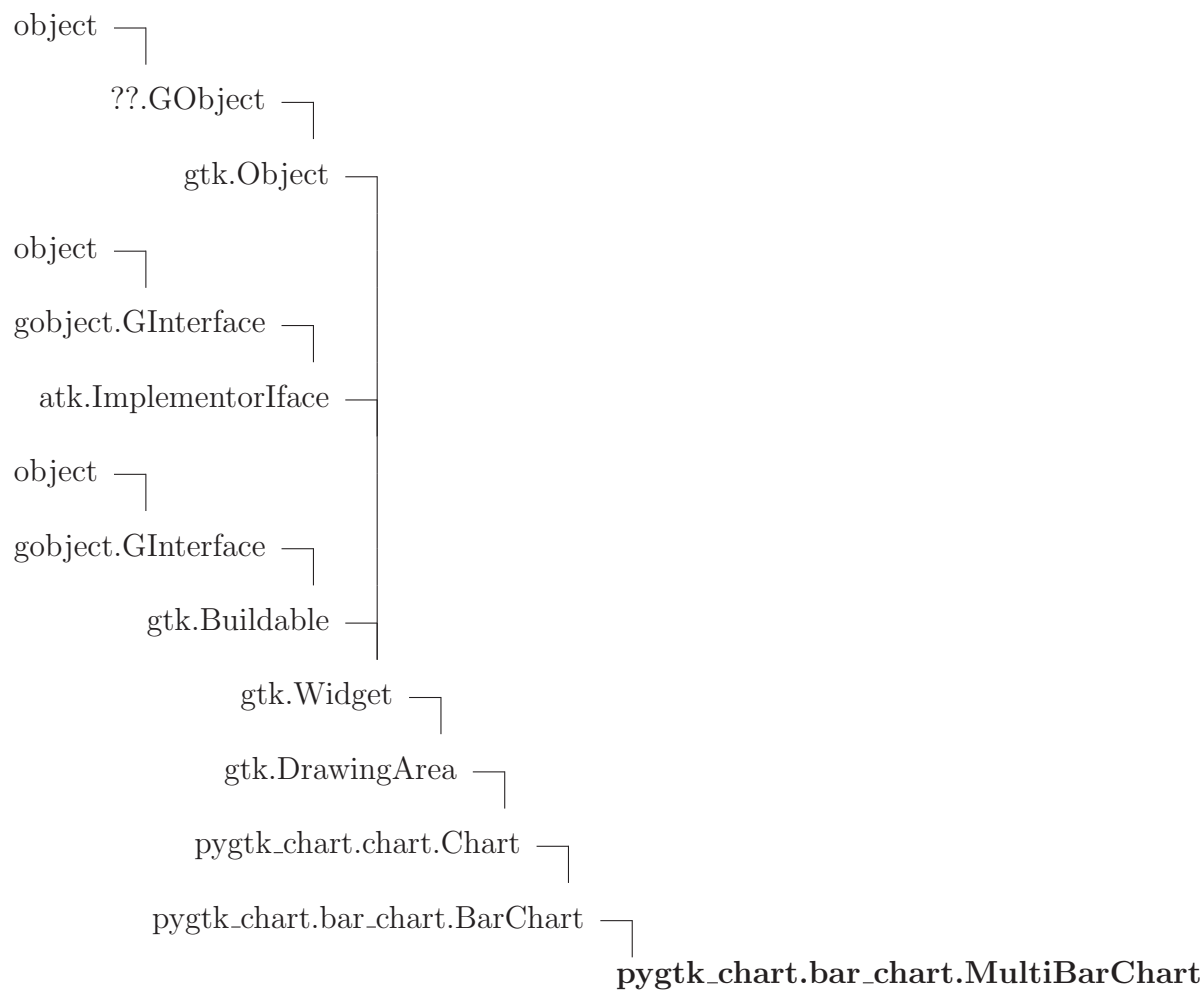
2.4.3 Class Variables

Name	Description
__gproperties__	Value: {"name":(gobject.TYPE_STRING, "bar name", "A unique name ...

continued on next page

Name	Description
--gtype--	Value: <GType pygtk_chart+bar_chart+MultiBar (171247656)>
<i>Inherited from pygtk_chart.chart.ChartObject (Section 4.2)</i>	
--gsignals--	

2.5 Class MultiBarChart



2.5.1 Methods

`__init__(self)`

`x.__init__(...)` initializes `x`; see `x.__class__.__doc__` for signature

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

`add_bar(self, bar)`

Overrides: `pygtk_chart.bar_chart.BarChart.add_bar`

Inherited from `pygtk_chart.bar_chart.BarChart` (Section 2.3)

`do_get_property()`, `do_set_property()`, `draw()`, `get_bar()`, `get_draw_labels()`, `get_enable_mouseover()`, `get_show_values()`, `set_draw_labels()`, `set_enable_mouseover()`, `set_show_values()`

Inherited from `pygtk_chart.chart.Chart` (Section 4.1)

`draw_basics()`, `export_png()`, `export_svg()`, `expose()`

Inherited from `gtk.DrawingArea`

`size()`

Inherited from `gtk.Widget`

`activate()`, `add_accelerator()`, `add_events()`, `add_mnemonic_label()`, `can_activate_accel()`, `child_focus()`, `child_notify()`, `class_path()`, `create_pango_context()`, `create_pango_layout()`, `destroy()`, `do_button_press_event()`, `do_button_release_event()`, `do_can_activate_accel()`, `do_client_event()`, `do_composited_changed()`, `do_configure_event()`, `do_delete_event()`, `do_destroy_event()`, `do_direction_changed()`, `do_drag_begin()`, `do_drag_data_delete()`, `do_drag_data_get()`, `do_drag_data_received()`, `do_drag_drop()`, `do_drag_end()`, `do_drag_leave()`, `do_drag_motion()`, `do_enter_notify_event()`, `do_event()`, `do_expose_event()`, `do_focus()`, `do_focus_in_event()`, `do_focus_out_event()`, `do_get_accessible()`, `do_grab_broken_event()`, `do_grab_focus()`, `do_grab_notify()`, `do_hide()`, `do_hide_all()`, `do_hierarchy_changed()`, `do_key_press_event()`, `do_key_release_event()`, `do_leave_notify_event()`, `do_map()`, `do_map_event()`, `do_mnemonic_activate()`, `do_motion_notify_event()`, `do_no_expose_event()`, `do_parent_set()`, `do_popup_menu()`, `do_property_notify_event()`, `do_proximity_in_event()`, `do_proximity_out_event()`, `do_realize()`, `do_screen_changed()`, `do_scroll_event()`, `do_selection_clear_event()`, `do_selection_get()`, `do_selection_notify_event()`, `do_selection_received()`, `do_selection_request_event()`, `do_show()`, `do_show_all()`, `do_show_help()`, `do_size_allocate()`, `do_size_request()`, `do_state_changed()`, `do_style_set()`, `do_unmap()`, `do_unmap_event()`, `do_unrealize()`, `do_visibility_notify_event()`, `do_window_state_event()`, `drag_begin()`, `drag_check_threshold()`, `drag_dest_add_image_targets()`, `drag_dest_add_text_targets()`, `drag_dest_add_uri_targets()`, `drag_dest_find_target()`, `drag_dest_get_target_list()`, `drag_dest_get_track_motion()`, `drag_dest_set()`, `drag_dest_set_proxy()`, `drag_dest_set_target_list()`, `drag_dest_set_track_motion()`, `drag_dest_unset()`, `drag_get_data()`, `drag_highlight()`, `drag_source_add_image_targets()`, `drag_source_add_text_targets()`,

`drag_source_add_uri_targets()`, `drag_source_get_target_list()`, `drag_source_set()`, `drag_source_set_icon()`,
`drag_source_set_icon_name()`, `drag_source_set_icon_pixmap()`, `drag_source_set_icon_stock()`,
`drag_source_set_target_list()`, `drag_source_unset()`, `drag_unhighlight()`, `ensure_style()`,
`error_bell()`, `event()`, `freeze_child_notify()`, `get_accessible()`, `get_action()`, `get_activate_signal()`,
`get_allocation()`, `get_ancestor()`, `get_child_requisition()`, `get_child_visible()`, `get_clipboard()`,
`get_colormap()`, `get_composite_name()`, `get_direction()`, `get_display()`, `get_events()`,
`get_extension_events()`, `get_has_tooltip()`, `get_modifier_style()`, `get_name()`, `get_no_show_all()`,
`get_pango_context()`, `get_parent()`, `get_parent_window()`, `get_pointer()`, `get_root_window()`,
`get_screen()`, `get_settings()`, `get_size_request()`, `get_snapshot()`, `get_style()`, `get_tooltip_markup()`,
`get_tooltip_text()`, `get_tooltip_window()`, `get_toplevel()`, `get_visual()`, `get_window()`,
`grab_add()`, `grab_default()`, `grab_focus()`, `grab_remove()`, `has_screen()`, `hide()`, `hide_all()`,
`hide_on_delete()`, `input_shape_combine_mask()`, `intersect()`, `is_ancestor()`, `is_composited()`,
`is_focus()`, `keynav_failed()`, `list_mnemonic_labels()`, `map()`, `menu_get_for_attach_widget()`,
`mnemonic_activate()`, `modify_base()`, `modify_bg()`, `modify_cursor()`, `modify_fg()`,
`modify_font()`, `modify_style()`, `modify_text()`, `path()`, `queue_clear()`, `queue_clear_area()`,
`queue_draw()`, `queue_draw_area()`, `queue_resize()`, `queue_resize_no_redraw()`, `rc_get_style()`,
`realize()`, `region_intersect()`, `remove_accelerator()`, `remove_mnemonic_label()`, `render_icon()`,
`reparent()`, `reset_rc_styles()`, `reset_shapes()`, `selection_add_target()`, `selection_add_targets()`,
`selection_clear_targets()`, `selection_convert()`, `selection_owner_set()`,
`selection_remove_all()`, `send_expose()`, `set_accel_path()`, `set_activate_signal()`, `set_app_paintable()`,
`set_child_visible()`, `set_colormap()`, `set_composite_name()`, `set_direction()`, `set_double_buffered()`,
`set_events()`, `set_extension_events()`, `set_has_tooltip()`, `set_name()`, `set_no_show_all()`,
`set_parent()`, `set_parent_window()`, `set_redraw_on_allocate()`, `set_scroll_adjustments()`,
`set_sensitive()`, `set_set_scroll_adjustments_signal()`, `set_size_request()`, `set_state()`, `set_style()`,
`set_tooltip_markup()`, `set_tooltip_text()`, `set_tooltip_window()`, `set_uposition()`, `set_usize()`,
`shape_combine_mask()`, `show()`, `show_all()`, `show_now()`, `size_allocate()`, `size_request()`,
`style_get_property()`, `thaw_child_notify()`, `translate_coordinates()`, `trigger_tooltip_query()`,
`unmap()`, `unparent()`, `unrealize()`

Inherited from `gtk.Object`

`do_destroy()`, `flags()`, `remove_data()`, `remove_no_notify()`, `set_flags()`, `unset_flags()`

Inherited from `??GObject`

`__cmp__()`, `__copy__()`, `__deepcopy__()`, `__delattr__()`, `__gdoc__()`, `__gobject_init__()`,
`__hash__()`, `__new__()`, `__repr__()`, `__setattr__()`, `chain()`, `connect()`, `connect_after()`,
`connect_object()`, `connect_object_after()`, `disconnect()`, `disconnect_by_func()`, `emit()`,
`emit_stop_by_name()`, `freeze_notify()`, `get_data()`, `get_properties()`, `get_property()`,
`handler_block()`, `handler_block_by_func()`, `handler_disconnect()`, `handler_is_connected()`,
`handler_unblock()`, `handler_unblock_by_func()`, `notify()`, `props()`, `set_data()`, `set_properties()`,
`set_property()`, `stop_emission()`, `thaw_notify()`, `weak_ref()`

Inherited from `atk.ImplementorInterface`

`ref_accessible()`

Inherited from gtk.Buildable

add_child(), construct_child(), do_add_child(), do_construct_child(), do_get_internal_child(),
do_parser_finished(), do_set_name(), get_internal_child(), parser_finished()

Inherited from object

__getattr__(), __reduce__(), __reduce_ex__(), __str__()

2.5.2 Properties

Name	Description
<i>Inherited from gtk.Widget</i>	allocation, name, parent, requisition, saved_state, state, style, window
<i>Inherited from ??GObject</i>	__grefcount__
<i>Inherited from object</i>	__class__

2.5.3 Class Variables

Name	Description
__gsignals__	Value: { "multibar-clicked": (gobject.SIGNAL_RUN_LAST, gobject.TYP...
__gtype__	Value: <GType pygtk_chart+bar_chart+MultiBarChart (171249280)>
<i>Inherited from pygtk_chart.bar_chart.BarChart (Section 2.3)</i>	__gproperties__

3 Module `pygtk_chart.basics`

This module contains simple functions needed by all other modules.

Author: Sven Festersen (sven@sven-festersen.de)

3.1 Functions

<code>is_in_range</code> (<i>x</i> , (<i>xmin</i> , <i>xmax</i>))
Use this method to test whether $xmin \leq x \leq xmax$.
Parameters
<i>xmin</i> : (<i>type=number</i>)
<i>x</i> : (<i>type=number</i>)
<i>xmax</i> : (<i>type=number</i>)

<code>intersect_ranges</code> (<i>range_a</i> , <i>range_b</i>)
--

<code>get_center</code> (<i>rect</i>)
Find the center point of a rectangle.
Parameters
<i>rect</i> : The rectangle.
(<i>type=gtk.gdk.Rectangle</i>)
Return Value
A (x, y) tuple specifying the center point.

<code>color_rgb_to_cairo</code> (<i>color</i>)
Convert a 8 bit RGB value to cairo color.
Parameters
<i>color</i> : The color to convert.
(<i>type=a triple of integers between 0 and 255</i>)
Return Value
A color in cairo format.

color_html_to_cairo(*color*)

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

Parameters

color: The color to convert.
(*type=html color string*)

Return Value

A color in cairo format.

color_list_from_file(*filename*)

Read a file with one html hex color per line and return a list of cairo colors.

show_text(*context, rect, x, y, text, font, size, slant=0, weight=0, underline=False, reference_point=0*)

3.2 Variables

Name	Description
REF_BOTTOM_LEFT	Value: 0
REF_TOP_LEFT	Value: 1
REF_TOP_RIGHT	Value: 2
REF_BOTTOM_RIGHT	Value: 4

4 Module `pygtk_chart.chart`

(section) Module Contents

This is the main module. It contains the base classes for chart widgets.

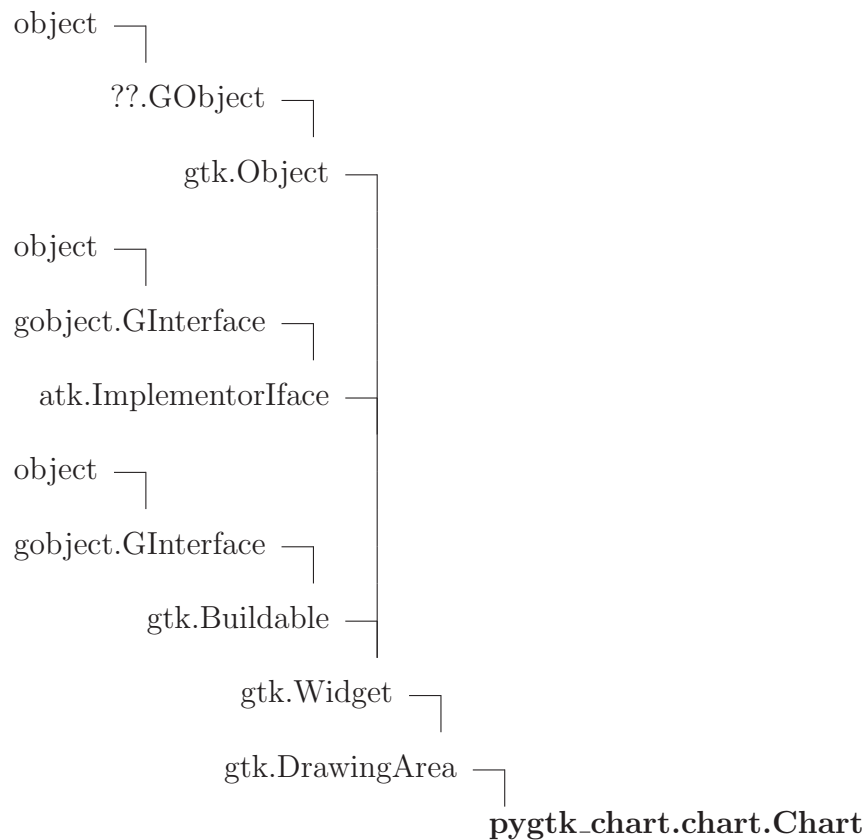
- class `Chart`: base class for all chart widgets.
- class `ChartObject`: base class for all things that can be drawn on a chart.
- class `Background`: background of a chart widget.
- class `Title`: title of a chart.

(section) Colors

All colors have to be `(r, g, b)` tuples. The value of `r`, `g` and `b` has to be between 0.0 and 1.0. For example `(0, 0, 0)` is black and `(1, 1, 1)` is white.

Author: Sven Festersen (sven@sven-festersen.de)

4.1 Class `Chart`



Known Subclasses: `pygtk_chart.pie_chart.PieChart`, `pygtk_chart.line_chart.LineChart`, `pygtk_chart.bar_c`

This is the base class for all chart widgets.

4.1.1 Methods

`__init__(self)`

`x.__init__(...)` initializes `x`; see `x.__class__.__doc__` for signature

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

`expose(self, widget, event)`

This method is called when an instance of `Chart` receives the `gtk.expose_event`.

Parameters

`widget`: The widget that received the event.

(*type=gtk.Widget*)

`event`: The event.

(*type=gtk.Event*)

`draw_basics(self, context, rect)`

Draw basic things that every plot has (background, title, ...).

Parameters

`context`: The context to draw on.

(*type=cairo.Context*)

`rect`: A rectangle representing the charts area.

(*type=gtk.gdk.Rectangle*)

`draw(self, context)`

Draw the widget. This method is called automatically. Don't call it yourself. If you want to force a redrawing of the widget, call the `queue_draw()` method.

Parameters

`context`: The context to draw on.

(*type=cairo.Context*)

Overrides: `gtk.Widget.draw`

export_svg(*self*, *filename*)

Saves the contents of the widget to svg file. The size of the image will be the size of the widget.

Parameters

filename: The path to the file where you want the chart to be saved.
(*type=string*)

export_png(*self*, *filename*)

Saves the contents of the widget to png file. The size of the image will be the size of the widget.

Parameters

filename: The path to the file where you want the chart to be saved.
(*type=string*)

Inherited from gtk.DrawingArea

size()

Inherited from gtk.Widget

activate(), add_accelerator(), add_events(), add_mnemonic_label(), can_activate_accel(), child_focus(), child_notify(), class_path(), create_pango_context(), create_pango_layout(), destroy(), do_button_press_event(), do_button_release_event(), do_can_activate_accel(), do_client_event(), do_composited_changed(), do_configure_event(), do_delete_event(), do_destroy_event(), do_direction_changed(), do_drag_begin(), do_drag_data_delete(), do_drag_data_get(), do_drag_data_received(), do_drag_drop(), do_drag_end(), do_drag_leave(), do_drag_motion(), do_enter_notify_event(), do_event(), do_expose_event(), do_focus(), do_focus_in_event(), do_focus_out_event(), do_get_accessible(), do_grab_broken_event(), do_grab_focus(), do_grab_notify(), do_hide(), do_hide_all(), do_hierarchy_changed(), do_key_press_event(), do_key_release_event(), do_leave_notify_event(), do_map(), do_map_event(), do_mnemonic_activate(), do_motion_notify_event(), do_no_expose_event(), do_parent_set(), do_popup_menu(), do_property_notify_event(), do_proximity_in_event(), do_proximity_out_event(), do_realize(), do_screen_changed(), do_scroll_event(), do_selection_clear_event(), do_selection_get(), do_selection_notify_event(), do_selection_received(), do_selection_request_event(), do_show(), do_show_all(), do_show_help(), do_size_allocate(), do_size_request(), do_state_changed(), do_style_set(), do_unmap(), do_unmap_event(), do_unrealize(), do_visibility_notify_event(), do_window_state_event(), drag_begin(), drag_check_threshold(), drag_dest_add_image_targets(), drag_dest_add_text_targets(), drag_dest_add_uri_targets(), drag_dest_find_target(), drag_dest_get_target_list(), drag_dest_get_track_motion(), drag_dest_set(), drag_dest_set_proxy(), drag_dest_set_target_list(), drag_dest_set_track_motion(), drag_dest_unset(), drag_get_data(), drag_highlight(), drag_source_add_image_targets(), drag_source_add_text_targets(),

drag_source_add_uri_targets(), drag_source_get_target_list(), drag_source_set(), drag_source_set_icon(),
 drag_source_set_icon_name(), drag_source_set_icon_pixmap(), drag_source_set_icon_stock(),
 drag_source_set_target_list(), drag_source_unset(), drag_unhighlight(), ensure_style(),
 error_bell(), event(), freeze_child_notify(), get_accessible(), get_action(), get_activate_signal(),
 get_allocation(), get_ancestor(), get_child_requisition(), get_child_visible(), get_clipboard(),
 get_colormap(), get_composite_name(), get_direction(), get_display(), get_events(),
 get_extension_events(), get_has_tooltip(), get_modifier_style(), get_name(), get_no_show_all(),
 get_pango_context(), get_parent(), get_parent_window(), get_pointer(), get_root_window(),
 get_screen(), get_settings(), get_size_request(), get_snapshot(), get_style(), get_tooltip_markup(),
 get_tooltip_text(), get_tooltip_window(), get_toplevel(), get_visual(), get_window(),
 grab_add(), grab_default(), grab_focus(), grab_remove(), has_screen(), hide(), hide_all(),
 hide_on_delete(), input_shape_combine_mask(), intersect(), is_ancestor(), is_composited(),
 is_focus(), keynav_failed(), list_mnemonic_labels(), map(), menu_get_for_attach_widget(),
 mnemonic_activate(), modify_base(), modify_bg(), modify_cursor(), modify_fg(),
 modify_font(), modify_style(), modify_text(), path(), queue_clear(), queue_clear_area(),
 queue_draw(), queue_draw_area(), queue_resize(), queue_resize_no_redraw(), rc_get_style(),
 realize(), region_intersect(), remove_accelerator(), remove_mnemonic_label(), ren-
 der_icon(), reparent(), reset_rc_styles(), reset_shapes(), selection_add_target(), se-
 lection_add_targets(), selection_clear_targets(), selection_convert(), selection_owner_set(),
 selection_remove_all(), send_expose(), set_accel_path(), set_activate_signal(), set_app_paintable(),
 set_child_visible(), set_colormap(), set_composite_name(), set_direction(), set_double_buffered(),
 set_events(), set_extension_events(), set_has_tooltip(), set_name(), set_no_show_all(),
 set_parent(), set_parent_window(), set_redraw_on_allocate(), set_scroll_adjustments(),
 set_sensitive(), set_set_scroll_adjustments_signal(), set_size_request(), set_state(), set_style(),
 set_tooltip_markup(), set_tooltip_text(), set_tooltip_window(), set_uposition(), set_usize(),
 shape_combine_mask(), show(), show_all(), show_now(), size_allocate(), size_request(),
 style_get_property(), thaw_child_notify(), translate_coordinates(), trigger_tooltip_query(),
 unmap(), unparent(), unrealize()

Inherited from gtk.Object

do_destroy(), flags(), remove_data(), remove_no_notify(), set_flags(), unset_flags()

Inherited from ??GObject

__cmp__(), __copy__(), __deepcopy__(), __delattr__(), __gdoc__(), __gobject_init__(),
 __hash__(), __new__(), __repr__(), __setattr__(), chain(), connect(), connect_after(),
 connect_object(), connect_object_after(), disconnect(), disconnect_by_func(), emit(),
 emit_stop_by_name(), freeze_notify(), get_data(), get_properties(), get_property(),
 handler_block(), handler_block_by_func(), handler_disconnect(), handler_is_connected(),
 handler_unblock(), handler_unblock_by_func(), notify(), props(), set_data(), set_properties(),
 set_property(), stop_emission(), thaw_notify(), weak_ref()

Inherited from atk.ImplementorIface

ref_accessible()

Inherited from gtk.Buildable

add_child(), construct_child(), do_add_child(), do_construct_child(), do_get_internal_child(),
do_parser_finished(), do_set_name(), get_internal_child(), parser_finished()

Inherited from object

__getattr__(), __reduce__(), __reduce_ex__(), __str__()

4.1.2 Properties

Name	Description
<i>Inherited from gtk.Widget</i>	allocation, name, parent, requisition, saved_state, state, style, window
<i>Inherited from <code>?.GObject</code></i>	__grefcount__
<i>Inherited from object</i>	__class__

4.1.3 Class Variables

Name	Description
<i>Inherited from gtk.DrawingArea</i>	__gtype__

4.2 Class ChartObject

Known Subclasses: pygtk_chart.chart.Background, pygtk_chart.chart.Title, pygtk_chart.pie_chart.PieArea, pygtk_chart.line_chart.Axis, pygtk_chart.line_chart.Graph, pygtk_chart.line_chart.Grid, pygtk_chart.bar_chart.Bar, pygtk_chart.bar_chart.MultiBar

This is the base class for all things that can be drawn in a chart, e.g. title, axes, graphs,...

4.2.1 Methods

`__init__(self)``x.__init__(...)` initializes `x`; see `x.__class__.__doc__` for signatureOverrides: `object.__init__` `extit`(inherited documentation)**`do_get_property(self, property)`****`do_set_property(self, property, value)`****`draw(self, context, rect)`**This method is called by the parent `Chart` instance. It calls `_do_draw`.**Parameters****`context`:** The context to draw on.*(type=cairo.Context)***`rect`:** A rectangle representing the charts area.*(type=gtk.gdk.Rectangle)***`set_antialias(self, antialias)`**This method sets the antialiasing mode of the `ChartObject`. Antialiasing is enabled by default.**Parameters****`antialias`:** If `False`, antialiasing is disabled for this `ChartObject`.*(type=boolean)***`get_antialias(self)`****`set_visible(self, visible)`**Use this method to set whether the `ChartObject` should be visible or not.**Parameters****`visible`:** If `False`, the `PlotObject` won't be drawn.*(type=boolean)***`get_visible(self)`***Inherited from `??GObject`*

`--cmp--()`, `--copy--()`, `--deepcopy--()`, `--delattr--()`, `--gdoc--()`, `--gobject_init--()`,
`--hash--()`, `--new--()`, `--repr--()`, `--setattr--()`, `chain()`, `connect()`, `connect_after()`,
`connect_object()`, `connect_object_after()`, `disconnect()`, `disconnect_by_func()`, `emit()`,
`emit_stop_by_name()`, `freeze_notify()`, `get_data()`, `get_properties()`, `get_property()`,
`handler_block()`, `handler_block_by_func()`, `handler_disconnect()`, `handler_is_connected()`,
`handler_unblock()`, `handler_unblock_by_func()`, `notify()`, `props()`, `set_data()`, `set_properties()`,
`set_property()`, `stop_emission()`, `thaw_notify()`, `weak_ref()`

Inherited from object

`--getattr__()`, `--reduce--()`, `--reduce_ex--()`, `--str--()`

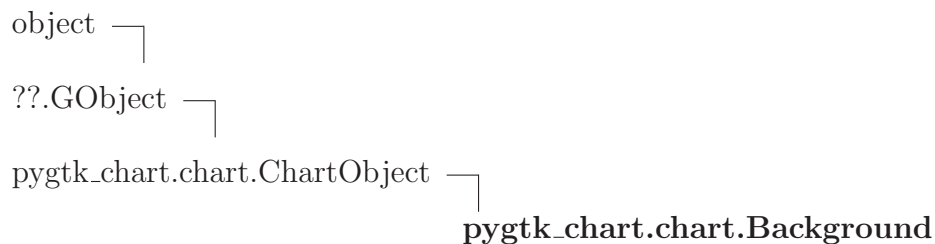
4.2.2 Properties

Name	Description
<i>Inherited from <code>??GObject</code></i>	
<code>--grefcount--</code>	
<i>Inherited from object</i>	
<code>--class--</code>	

4.2.3 Class Variables

Name	Description
<code>--gsignals--</code>	Value: {"appearance-changed": (gobject.SIGNAL_RUN_LAST, gobject.T...
<code>--gproperties--</code>	Value: {"visible": (gobject.TYPE_BOOLEAN, "visibilty of the objec...
<code>--gtype--</code>	Value: <GType pygtk_chart+chart+ChartObject (169550512)>

4.3 Class Background



The background of a chart.

4.3.1 Methods

`__init__(self)`

`x.__init__(...)` initializes `x`; see `x.__class__.__doc__` for signature

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

`do_get_property(self, property)`

Overrides: `pygtk_chart.chart.ChartObject.do_get_property`

`do_set_property(self, property, value)`

Overrides: `pygtk_chart.chart.ChartObject.do_set_property`

`set_color(self, color)`

The `set_color()` method can be used to change the color of the background.

Parameters

`color`: Set the background to be filled with this color.

(*type=a color*)

`get_color(self)`

`set_gradient(self, color_start, color_end)`

Use `set_gradient()` to define a vertical gradient as the background.

Parameters

`color_start`: The starting (top) color of the gradient.

(*type=a color*)

`color_end`: The ending (bottom) color of the gradient.

(*type=a color*)

`get_gradient(self)`

set_image(self, filename)

The `set_image()` method sets the background to be filled with a png image.

Parameters

filename: Path to the png file you want to use as background image. If the file does not exists, the background is set to white.
(*type=string*)

get_image(self)

Inherited from `pygtk_chart.chart.ChartObject` (Section 4.2)

`draw()`, `get_antialias()`, `get_visible()`, `set_antialias()`, `set_visible()`

Inherited from `??GObject`

`__cmp__()`, `__copy__()`, `__deepcopy__()`, `__delattr__()`, `__gdoc__()`, `__gobject_init__()`, `__hash__()`, `__new__()`, `__repr__()`, `__setattr__()`, `chain()`, `connect()`, `connect_after()`, `connect_object()`, `connect_object_after()`, `disconnect()`, `disconnect_by_func()`, `emit()`, `emit_stop_by_name()`, `freeze_notify()`, `get_data()`, `get_properties()`, `get_property()`, `handler_block()`, `handler_block_by_func()`, `handler_disconnect()`, `handler_is_connected()`, `handler_unblock()`, `handler_unblock_by_func()`, `notify()`, `props()`, `set_data()`, `set_properties()`, `set_property()`, `stop_emission()`, `thaw_notify()`, `weak_ref()`

Inherited from `object`

`__getattr__()`, `__reduce__()`, `__reduce_ex__()`, `__str__()`

4.3.2 Properties

Name	Description
<i>Inherited from <code>??GObject</code></i>	
<code>__grefcount__</code>	
<i>Inherited from <code>object</code></i>	
<code>__class__</code>	

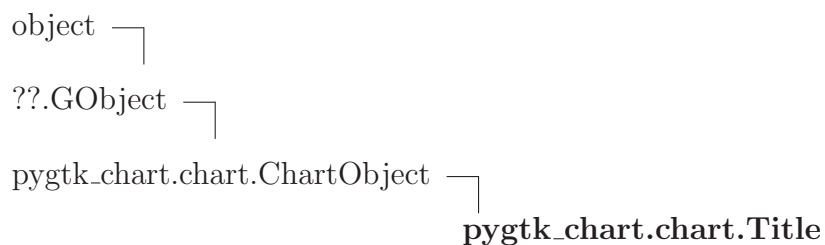
4.3.3 Class Variables

Name	Description
<code>__gproperties__</code>	Value: <code>{"color": (gobject.TYPE_PYOBJECT, "background color", "The...</code>

continued on next page

Name	Description
--gtype--	Value: <GType pygtk_chart+chart+Background (169553328)>
<i>Inherited from pygtk_chart.chart.ChartObject (Section 4.2)</i>	
--gsignals--	

4.4 Class Title



The title of a chart. The title will be drawn centered at the top of the chart.

4.4.1 Methods

__init__(self, text=None)

x.__init__(...) initializes x; see x.__class__.__doc__ for signature

Overrides: object.__init__ extit(inherited documentation)

do_get_property(self, property)

Overrides: pygtk_chart.chart.ChartObject.do_get_property

do_set_property(self, property, value)

Overrides: pygtk_chart.chart.ChartObject.do_set_property

set_color(self, color)

The set_color() method sets the color of the title text.

Parameters

color: The color of the title.

(type=a color)

get_color(self)

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

Use the set_text() method to set the title of the chart.

Parameters

text: The title of the chart.
(*type=string*)

```
get_text(self)
```

Inherited from pygtk_chart.chart.ChartObject (Section 4.2)

draw(), get_antialias(), get_visible(), set_antialias(), set_visible()

Inherited from ??GObject

__cmp__(), __copy__(), __deepcopy__(), __delattr__(), __gdoc__(), __gobject_init__(),
__hash__(), __new__(), __repr__(), __setattr__(), chain(), connect(), connect_after(),
connect_object(), connect_object_after(), disconnect(), disconnect_by_func(), emit(),
emit_stop_by_name(), freeze_notify(), get_data(), get_properties(), get_property(),
handler_block(), handler_block_by_func(), handler_disconnect(), handler_is_connected(),
handler_unblock(), handler_unblock_by_func(), notify(), props(), set_data(), set_properties(),
set_property(), stop_emission(), thaw_notify(), weak_ref()

Inherited from object

__getattr__(), __reduce__(), __reduce_ex__(), __str__()

4.4.2 Properties

Name	Description
<i>Inherited from ??GObject</i>	
__grefcount__	
<i>Inherited from object</i>	
__class__	

4.4.3 Class Variables

Name	Description
__gproperties__	Value: {"color": (gobject.TYPE_PYOBJECT, "title color", "The colo...
__gtype__	Value: <GType pygtk_chart+chart+Title (169554848)>
<i>Inherited from pygtk_chart.chart.ChartObject (Section 4.2)</i>	
__gsignals__	

5 Module `pygtk_chart.line_chart`

Contains the LineChart widget.

Author: Sven Festersen (sven@sven-festersen.de)

5.1 Functions

`graph_new_from_function`(*func*, *xmin*, *xmax*, *graph_name*, *samples*=100, *do_optimize_sampling*=True)

Returns a `line_chart.Graph` with data created from the function $y = \text{func}(x)$ with x in $[\text{xmin}, \text{xmax}]$. The id of the new graph is *graph_name*. The parameter *samples* gives the number of points that should be evaluated in $[\text{xmin}, \text{xmax}]$ (default: 100). If *do_optimize_sampling* is True (default) additional points will be evaluated to smoothen the curve.

Parameters

<code>func</code>:	the function to evaluate (<i>type=a function</i>)
<code>xmin</code>:	the minimum x value to evaluate (<i>type=float</i>)
<code>xmax</code>:	the maximum x value to evaluate (<i>type=float</i>)
<code>graph_name</code>:	a unique name for the new graph (<i>type=string</i>)
<code>samples</code>:	number of samples (<i>type=int</i>)
<code>do_optimize_sampling</code>:	set whether to add additional points (<i>type=boolean</i>)

Return Value

`line_chart.Graph`

`optimize_sampling`(*func*, *data*)

graph_new_from_file (<i>filename</i> , <i>graph_name</i> , <i>x_col</i> =0, <i>y_col</i> =1)	
Returns a <code>line_chart.Graph</code> with point taken from data file <i>filename</i> . The id of the new graph is <i>graph_name</i> .	
Data file format: The columns in the file have to be separated by tabs or one or more spaces. Everything after '#' is ignored (comment).	
Use the parameters <i>x_col</i> and <i>y_col</i> to control which columns to use for plotting. By default, the first column (<i>x_col</i> =0) is used for x values, the second (<i>y_col</i> =1) is used for y values.	
Parameters	
filename:	path to the data file (<i>type=string</i>)
graph_name:	a unique name for the graph (<i>type=string</i>)
x_col:	the number of the column to use for x values (<i>type=int</i>)
y_col:	the number of the column to use for y values (<i>type=int</i>)
Return Value	
<code>line_chart.Graph</code>	

5.2 Variables

Name	Description
RANGE_AUTO	Value: 0
GRAPH_PADDING	Value: 0.06666666666667
GRAPH_POINTS	Value: 1
GRAPH_LINES	Value: 2
GRAPH_BOTH	Value: 3
COLOR_AUTO	Value: 4
POSITION_AUTO	Value: 5
POSITION_LEFT	Value: 6
POSITION_RIGHT	Value: 7
POSITION_BOTTOM	Value: 6
POSITION_TOP	Value: 7
COLORS	Value: [(0.8, 0.0, 0.0), (0.203921568627, 0.396078431373, 0.6431...]

5.3 Class RangeCalculator

This helper class calculates ranges. It is used by the LineChart widget internally, there is no need to create an instance yourself.

5.3.1 Methods

```
__init__(self)
```

```
add_graph(self, graph)
```

```
get_ranges(self)
```

```
set_xrange(self, xrange)
```

```
set_yrange(self, yrange)
```

```
get_absolute_zero(self, rect)
```

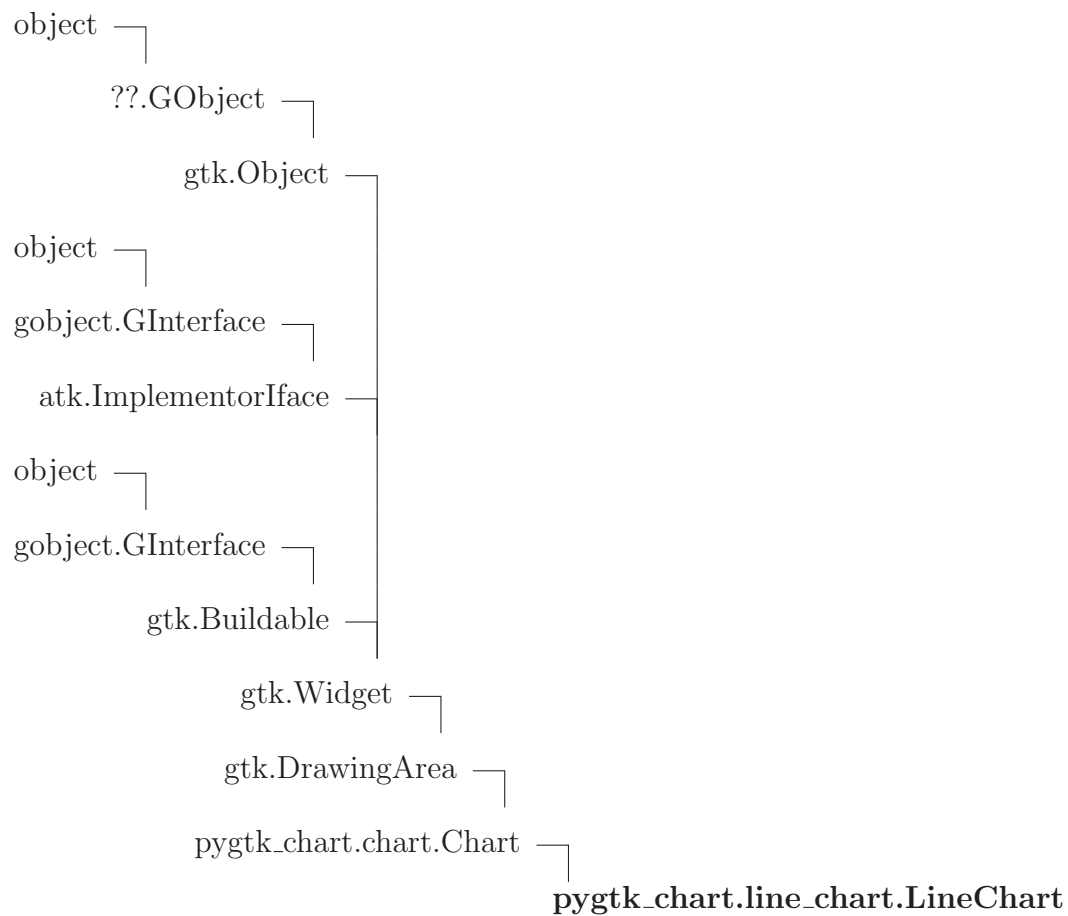
```
get_absolute_point(self, rect, x, y)
```

```
prepare_ticks(self, rect)
```

```
get_xticks(self, rect)
```

```
get_yticks(self, rect)
```

5.4 Class **LineChart**



A widget that shows a line chart. The following objects can be accessed:

- **LineChart.background** (inherited from **chart.Chart**)
- **LineChart.title** (inherited from **chart.Chart**)
- **LineChart.graphs**
- **LineChart.grid**
- **LineChart.xaxis**
- **LineChart.yaxis**

5.4.1 Methods

```

__init__(self)
x.__init__(...) initializes x; see x.__class__.__doc__ for signature
Overrides: object.__init__ extit(inherited documentation)

```

draw(*self*, *context*)

Draw the widget. This method is called automatically. Don't call it yourself. If you want to force a redrawing of the widget, call the `queue_draw()` method.

Parameters

context: The context to draw on.

(*type=cairo.Context*)

Overrides: `gtk.Widget.draw`

add_graph(*self*, *graph*)

Add a graph object to the plot.

Parameters

graph: The graph to add.

(*type=line_chart.Graph*)

remove_graph(*self*, *name*)

Remove a graph from the plot.

Parameters

name: The name of the graph to remove.

(*type=string*)

set_xrange(*self*, *xrange*)

Set the visible xrange. xrange has to be a pair: (xmin, xmax) or `RANGE_AUTO`. If you set it to `RANGE_AUTO`, the visible range will be calculated.

Parameters

xrange: The new xrange.

(*type=pair of numbers*)

set_yrange(*self*, *yrange*)

Set the visible yrange. yrange has to be a pair: (ymin, ymax) or `RANGE_AUTO`. If you set it to `RANGE_AUTO`, the visible range will be calculated.

Parameters

yrange: The new yrange.

(*type=pair of numbers*)

Inherited from `pygtk_chart.chart.Chart` (Section 4.1)

`draw_basics()`, `export_png()`, `export_svg()`, `expose()`

Inherited from `gtk.DrawingArea`

`size()`

Inherited from `gtk.Widget`

`activate()`, `add_accelerator()`, `add_events()`, `add_mnemonic_label()`, `can_activate_accel()`, `child_focus()`, `child_notify()`, `class_path()`, `create_pango_context()`, `create_pango_layout()`, `destroy()`, `do_button_press_event()`, `do_button_release_event()`, `do_can_activate_accel()`, `do_client_event()`, `do_composited_changed()`, `do_configure_event()`, `do_delete_event()`, `do_destroy_event()`, `do_direction_changed()`, `do_drag_begin()`, `do_drag_data_delete()`, `do_drag_data_get()`, `do_drag_data_received()`, `do_drag_drop()`, `do_drag_end()`, `do_drag_leave()`, `do_drag_motion()`, `do_enter_notify_event()`, `do_event()`, `do_expose_event()`, `do_focus()`, `do_focus_in_event()`, `do_focus_out_event()`, `do_get_accessible()`, `do_grab_broken_event()`, `do_grab_focus()`, `do_grab_notify()`, `do_hide()`, `do_hide_all()`, `do_hierarchy_changed()`, `do_key_press_event()`, `do_key_release_event()`, `do_leave_notify_event()`, `do_map()`, `do_map_event()`, `do_mnemonic_activate()`, `do_motion_notify_event()`, `do_no_expose_event()`, `do_parent_set()`, `do_popup_menu()`, `do_property_notify_event()`, `do_proximity_in_event()`, `do_proximity_out_event()`, `do_realize()`, `do_screen_changed()`, `do_scroll_event()`, `do_selection_clear_event()`, `do_selection_get()`, `do_selection_notify_event()`, `do_selection_received()`, `do_selection_request_event()`, `do_show()`, `do_show_all()`, `do_show_help()`, `do_size_allocate()`, `do_size_request()`, `do_state_changed()`, `do_style_set()`, `do_unmap()`, `do_unmap_event()`, `do_unrealize()`, `do_visibility_notify_event()`, `do_window_state_event()`, `drag_begin()`, `drag_check_threshold()`, `drag_dest_add_image_targets()`, `drag_dest_add_text_targets()`, `drag_dest_add_uri_targets()`, `drag_dest_find_target()`, `drag_dest_get_target_list()`, `drag_dest_get_track_motion()`, `drag_dest_set()`, `drag_dest_set_proxy()`, `drag_dest_set_target_list()`, `drag_dest_set_track_motion()`, `drag_dest_unset()`, `drag_get_data()`, `drag_highlight()`, `drag_source_add_image_targets()`, `drag_source_add_text_targets()`, `drag_source_add_uri_targets()`, `drag_source_get_target_list()`, `drag_source_set()`, `drag_source_set_icon()`, `drag_source_set_icon_name()`, `drag_source_set_icon_pixmap()`, `drag_source_set_icon_stock()`, `drag_source_set_target_list()`, `drag_source_unset()`, `drag_unhighlight()`, `ensure_style()`, `error_bell()`, `event()`, `freeze_child_notify()`, `get_accessible()`, `get_action()`, `get_activate_signal()`, `get_allocation()`, `get_ancestor()`, `get_child_requisition()`, `get_child_visible()`, `get_clipboard()`, `get_colormap()`, `get_composite_name()`, `get_direction()`, `get_display()`, `get_events()`, `get_extension_events()`, `get_has_tooltip()`, `get_modifier_style()`, `get_name()`, `get_no_show_all()`, `get_pango_context()`, `get_parent()`, `get_parent_window()`, `get_pointer()`, `get_root_window()`, `get_screen()`, `get_settings()`, `get_size_request()`, `get_snapshot()`, `get_style()`, `get_tooltip_markup()`, `get_tooltip_text()`, `get_tooltip_window()`, `get_toplevel()`, `get_visual()`, `get_window()`, `grab_add()`, `grab_default()`, `grab_focus()`, `grab_remove()`, `has_screen()`, `hide()`, `hide_all()`, `hide_on_delete()`, `input_shape_combine_mask()`, `intersect()`, `is_ancestor()`, `is_composited()`, `is_focus()`, `keynav_failed()`, `list_mnemonic_labels()`, `map()`, `menu_get_for_attach_widget()`, `mnemonic_activate()`, `modify_base()`, `modify_bg()`, `modify_cursor()`, `modify_fg()`, `modify_font()`, `modify_style()`, `modify_text()`, `path()`, `queue_clear()`, `queue_clear_area()`, `queue_draw()`, `queue_draw_area()`, `queue_resize()`, `queue_resize_no_redraw()`, `rc_get_style()`, `realize()`, `region_intersect()`, `remove_accelerator()`, `remove_mnemonic_label()`, `ren-`

`der_icon()`, `reparent()`, `reset_rc_styles()`, `reset_shapes()`, `selection_add_target()`, `selection_add_targets()`, `selection_clear_targets()`, `selection_convert()`, `selection_owner_set()`, `selection_remove_all()`, `send_expose()`, `set_accel_path()`, `set_activate_signal()`, `set_app_paintable()`, `set_child_visible()`, `set_colormap()`, `set_composite_name()`, `set_direction()`, `set_double_buffered()`, `set_events()`, `set_extension_events()`, `set_has_tooltip()`, `set_name()`, `set_no_show_all()`, `set_parent()`, `set_parent_window()`, `set_redraw_on_allocate()`, `set_scroll_adjustments()`, `set_sensitive()`, `set_set_scroll_adjustments_signal()`, `set_size_request()`, `set_state()`, `set_style()`, `set_tooltip_markup()`, `set_tooltip_text()`, `set_tooltip_window()`, `set_uposition()`, `set_usize()`, `shape_combine_mask()`, `show()`, `show_all()`, `show_now()`, `size_allocate()`, `size_request()`, `style_get_property()`, `thaw_child_notify()`, `translate_coordinates()`, `trigger_tooltip_query()`, `unmap()`, `unparent()`, `unrealize()`

Inherited from gtk.Object

`do_destroy()`, `flags()`, `remove_data()`, `remove_no_notify()`, `set_flags()`, `unset_flags()`

Inherited from ??GObject

`__cmp__()`, `__copy__()`, `__deepcopy__()`, `__delattr__()`, `__gdoc__()`, `__gobject_init__()`, `__hash__()`, `__new__()`, `__repr__()`, `__setattr__()`, `chain()`, `connect()`, `connect_after()`, `connect_object()`, `connect_object_after()`, `disconnect()`, `disconnect_by_func()`, `emit()`, `emit_stop_by_name()`, `freeze_notify()`, `get_data()`, `get_properties()`, `get_property()`, `handler_block()`, `handler_block_by_func()`, `handler_disconnect()`, `handler_is_connected()`, `handler_unblock()`, `handler_unblock_by_func()`, `notify()`, `props()`, `set_data()`, `set_properties()`, `set_property()`, `stop_emission()`, `thaw_notify()`, `weak_ref()`

Inherited from atk.ImplementorIface

`ref_accessible()`

Inherited from gtk.Buildable

`add_child()`, `construct_child()`, `do_add_child()`, `do_construct_child()`, `do_get_internal_child()`, `do_parser_finished()`, `do_set_name()`, `get_internal_child()`, `parser_finished()`

Inherited from object

`__getattr__()`, `__reduce__()`, `__reduce_ex__()`, `__str__()`

5.4.2 Properties

Name	Description
<i>Inherited from gtk.Widget</i>	allocation, name, parent, requisition, saved_state, state, style, window
<i>Inherited from ??GObject</i>	<code>__grefcount__</code>
<i>Inherited from object</i>	

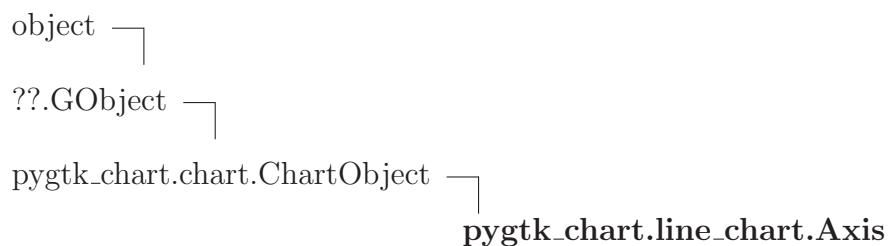
continued on next page

Name	Description
<code>__class__</code>	

5.4.3 Class Variables

Name	Description
<i>Inherited from <code>gtk.DrawingArea</code></i>	
<code>__gtype__</code>	

5.5 Class Axis



Known Subclasses: `pygtk_chart.line_chart.XAxis`, `pygtk_chart.line_chart.YAxis`

5.5.1 Methods

`__init__(self, range_calc, label)`
`x.__init__(...)` initializes `x`; see `x.__class__.__doc__` for signature
 Overrides: `object.__init__` extit(inherited documentation)

`do_get_property(self, property)`
 Overrides: `pygtk_chart.chart.ChartObject.do_get_property`

`do_set_property(self, property, value)`
 Overrides: `pygtk_chart.chart.ChartObject.do_set_property`

set_label(*self*, *label*)

Set the label of the axis.

Parameters

label: new label
(*type=string.*)

get_label(*self*)

Returns the current label of the axis.

Return Value

string.

set_show_label(*self*, *show*)

Set whether to show the axis' label.

Parameters

show: (*type=boolean.*)

get_show_label(*self*)

Returns True if the axis' label is shown.

Return Value

boolean.

set_position(*self*, *pos*)

Set the position of the axis. pos has to be one these constants:
POSITION_AUTO, POSITION_BOTTOM, POSITION_LEFT,
POSITION_RIGHT, POSITION_TOP.

get_position(*self*)

Returns the position of the axis. (see set_position for details).

set_show_tics(*self*, *show*)

Set whether to draw tics at the axis.

Parameters

show: (*type=boolean.*)

get_show_tics(*self*)

Returns True if tics are drawn.

Return Value

boolean.

set_show_tic_labels(*self*, *show*)

Set whether to draw tic labels. Labels are only drawn if tics are drawn.

Parameters

show: (*type=boolean.*)

get_show_tic_labels(*self*)

Returns True if tic labels are shown.

Return Value

boolean.

set_tic_format_function(*self*, *func*)

Use this to set the function that should be used to label the tics. The function should take a number as the only argument and return a string. Default: str

Parameters

func: (*type=function.*)

get_tic_format_function(*self*)

Returns the function currently used for labeling the tics.

Inherited from *pygtk_chart.chart.ChartObject*(Section 4.2)

draw(), get_antialias(), get_visible(), set_antialias(), set_visible()

Inherited from *??GObject*

__cmp__(), __copy__(), __deepcopy__(), __delattr__(), __gdoc__(), __gobject_init__(), __hash__(), __new__(), __repr__(), __setattr__(), chain(), connect(), connect_after(), connect_object(), connect_object_after(), disconnect(), disconnect_by_func(), emit(), emit_stop_by_name(), freeze_notify(), get_data(), get_properties(), get_property(), handler_block(), handler_block_by_func(), handler_disconnect(), handler_is_connected(), handler_unblock(), handler_unblock_by_func(), notify(), props(), set_data(), set_properties(), set_property(), stop_emission(), thaw_notify(), weak_ref()

Inherited from *object*

__getattr__(), __reduce__(), __reduce_ex__(), __str__()

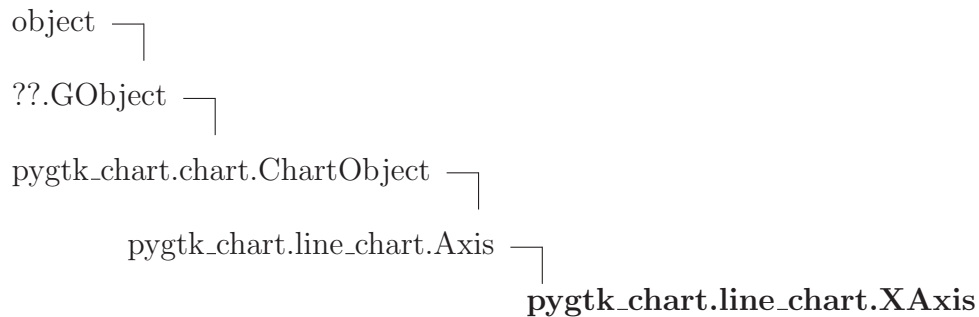
5.5.2 Properties

Name	Description
<i>Inherited from <code>??GObject</code></i>	
<code>--grefcount--</code>	
<i>Inherited from object</i>	
<code>--class--</code>	

5.5.3 Class Variables

Name	Description
<code>--gproperties--</code>	Value: <code>{"label":(gobject.TYPE_STRING, "axis label", "The label o...</code>
<code>--gtype--</code>	Value: <code><GType pygtk_chart+line_chart+Axis (170829088)></code>
<i>Inherited from <code>pygtk_chart.chart.ChartObject</code> (Section 4.2)</i>	
<code>--gsignals--</code>	

5.6 Class XAxis



This class represents the xaxis. It is used by the LineChart widget internally, there is no need to create an instance yourself.

5.6.1 Methods

<code>--init__(self, range_calc)</code>
<code>x.__init__(...)</code> initializes x; see <code>x.__class__.__doc__</code> for signature
Overrides: <code>object.__init__</code> <code>extit</code> (inherited documentation)

draw(*self, context, rect, yaxis*)

This method is called by the parent Plot instance. It calls `_do_draw`.

Parameters

context: The context to draw on.

rect: A rectangle representing the charts area.

Overrides: `pygtk_chart.chart.ChartObject.draw`

Inherited from *pygtk_chart.line_chart.Axis* (Section 5.5)

`do_get_property()`, `do_set_property()`, `get_label()`, `get_position()`, `get_show_label()`,
`get_show_tic_labels()`, `get_show_tics()`, `get_tic_format_function()`, `set_label()`, `set_position()`,
`set_show_label()`, `set_show_tic_labels()`, `set_show_tics()`, `set_tic_format_function()`

Inherited from *pygtk_chart.chart.ChartObject* (Section 4.2)

`get_antialias()`, `get_visible()`, `set_antialias()`, `set_visible()`

Inherited from *??GObject*

`__cmp__()`, `__copy__()`, `__deepcopy__()`, `__delattr__()`, `__gdoc__()`, `__gobject_init__()`,
`__hash__()`, `__new__()`, `__repr__()`, `__setattr__()`, `chain()`, `connect()`, `connect_after()`,
`connect_object()`, `connect_object_after()`, `disconnect()`, `disconnect_by_func()`, `emit()`,
`emit_stop_by_name()`, `freeze_notify()`, `get_data()`, `get_properties()`, `get_property()`,
`handler_block()`, `handler_block_by_func()`, `handler_disconnect()`, `handler_is_connected()`,
`handler_unblock()`, `handler_unblock_by_func()`, `notify()`, `props()`, `set_data()`, `set_properties()`,
`set_property()`, `stop_emission()`, `thaw_notify()`, `weak_ref()`

Inherited from *object*

`__getattr__()`, `__reduce__()`, `__reduce_ex__()`, `__str__()`

5.6.2 Properties

Name	Description
<i>Inherited from <code>??GObject</code></i>	
<code>__grefcount__</code>	
<i>Inherited from <code>object</code></i>	
<code>__class__</code>	

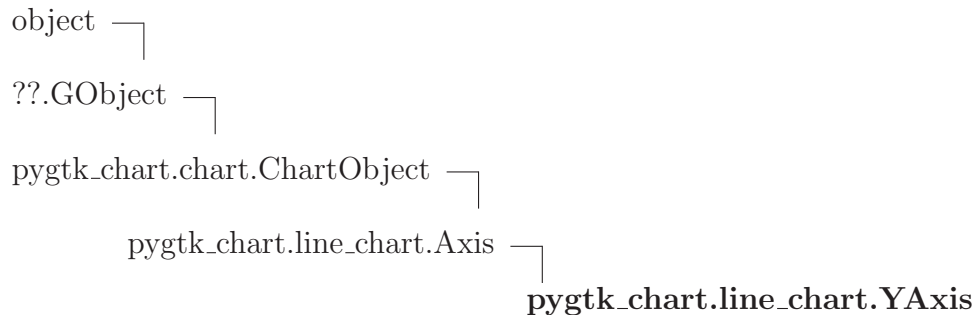
5.6.3 Class Variables

Name	Description
<i>Inherited from <code>pygtk_chart.line_chart.Axis</code> (Section 5.5)</i>	
<code>__gproperties__</code> , <code>__gtype__</code>	

continued on next page

Name	Description
<i>Inherited from <code>pygtk_chart.chart.ChartObject</code> (Section 4.2)</i>	
<code>--gsignals--</code>	

5.7 Class YAxis



This class represents the yaxis. It is used by the LineChart widget internally, there is no need to create an instance yourself.

5.7.1 Methods

```
__init__(self, range_calc)
```

`x.__init__(...)` initializes x; see `x.__class__.__doc__` for signature
 Overrides: `object.__init__` `extit`(inherited documentation)

```
draw(self, context, rect, xaxis)
```

This method is called by the parent Plot instance. It calls `_do_draw`.

Parameters

- `context`: The context to draw on.
- `rect`: A rectangle representing the charts area.

Overrides: `pygtk_chart.chart.ChartObject.draw`

Inherited from `pygtk_chart.line_chart.Axis` (Section 5.5)

`do_get_property()`, `do_set_property()`, `get_label()`, `get_position()`, `get_show_label()`,
`get_show_tic_labels()`, `get_show_tics()`, `get_tic_format_function()`, `set_label()`, `set_position()`,
`set_show_label()`, `set_show_tic_labels()`, `set_show_tics()`, `set_tic_format_function()`

Inherited from `pygtk_chart.chart.ChartObject` (Section 4.2)

`get_antialias()`, `get_visible()`, `set_antialias()`, `set_visible()`

Inherited from `??GObject`

`__cmp__()`, `__copy__()`, `__deepcopy__()`, `__delattr__()`, `__gdoc__()`, `__gobject_init__()`, `__hash__()`, `__new__()`, `__repr__()`, `__setattr__()`, `chain()`, `connect()`, `connect_after()`, `connect_object()`, `connect_object_after()`, `disconnect()`, `disconnect_by_func()`, `emit()`, `emit_stop_by_name()`, `freeze_notify()`, `get_data()`, `get_properties()`, `get_property()`, `handler_block()`, `handler_block_by_func()`, `handler_disconnect()`, `handler_is_connected()`, `handler_unblock()`, `handler_unblock_by_func()`, `notify()`, `props()`, `set_data()`, `set_properties()`, `set_property()`, `stop_emission()`, `thaw_notify()`, `weak_ref()`

Inherited from `object`

`__getattribute__()`, `__reduce__()`, `__reduce_ex__()`, `__str__()`

5.7.2 Properties

Name	Description
<i>Inherited from <code>??GObject</code></i>	
<code>__grefcount__</code>	
<i>Inherited from <code>object</code></i>	
<code>__class__</code>	

5.7.3 Class Variables

Name	Description
<i>Inherited from <code>pygtk_chart.line_chart.Axis</code> (Section 5.5)</i>	
<code>__gproperties__</code> , <code>__gtype__</code>	
<i>Inherited from <code>pygtk_chart.chart.ChartObject</code> (Section 4.2)</i>	
<code>__gsignals__</code>	

5.8 Class Grid

```

object └─
  ??GObject └─
    pygtk_chart.chart.ChartObject └─
      pygtk_chart.line_chart.Grid

```

A class representing the grid of the chart. It is used by the LineChart widget internally, there is no need to create an instance yourself.

5.8.1 Methods

`__init__(self, range_calc)`

`x.__init__(...)` initializes `x`; see `x.__class__.__doc__` for signature

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

`do_get_property(self, property)`

Overrides: `pygtk_chart.chart.ChartObject.do_get_property`

`do_set_property(self, property, value)`

Overrides: `pygtk_chart.chart.ChartObject.do_set_property`

`set_draw_horizontal_lines(self, draw)`

Set whether to draw horizontal grid lines.

Parameters

`draw`: (*type=boolean.*)

`get_draw_horizontal_lines(self)`

Returns True if horizontal grid lines are drawn.

Return Value

boolean.

`set_draw_vertical_lines(self, draw)`

Set whether to draw vertical grid lines.

Parameters

`draw`: (*type=boolean.*)

`get_draw_vertical_lines(self)`

Returns True if vertical grid lines are drawn.

Return Value

boolean.

set_color (<i>self</i> , <i>color</i>)

Set the color of the grid.

Parameters

<i>color</i> : The new color of the grid. <i>(type=a color)</i>
--

get_color (<i>self</i>)

Returns the color of the grid.

Return Value

a color.

Inherited from pygtk_chart.chart.ChartObject(Section 4.2)

draw(), get_antialias(), get_visible(), set_antialias(), set_visible()

Inherited from ??GObject

__cmp__(), __copy__(), __deepcopy__(), __delattr__(), __gdoc__(), __gobject_init__(),
 __hash__(), __new__(), __repr__(), __setattr__(), chain(), connect(), connect_after(),
 connect_object(), connect_object_after(), disconnect(), disconnect_by_func(), emit(),
 emit_stop_by_name(), freeze_notify(), get_data(), get_properties(), get_property(),
 handler_block(), handler_block_by_func(), handler_disconnect(), handler_is_connected(),
 handler_unblock(), handler_unblock_by_func(), notify(), props(), set_data(), set_properties(),
 set_property(), stop_emission(), thaw_notify(), weak_ref()

Inherited from object

__getattribute__(), __reduce__(), __reduce_ex__(), __str__()

5.8.2 Properties

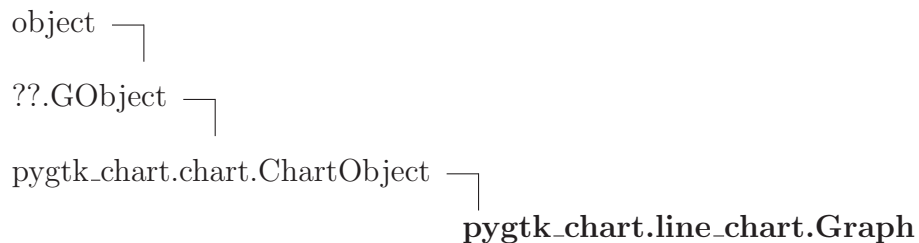
Name	Description
<i>Inherited from ??GObject</i>	
__grefcount__	
<i>Inherited from object</i>	
__class__	

5.8.3 Class Variables

continued on next page

Name	Description
<code>--gproperties--</code>	Value: { <code>"show-horizontal": (gobject.TYPE_BOOLEAN,</code> <code>"show horizonta...</code>
<code>--gtype--</code>	Value: <code><GType</code> <code>pygtk_chart+line.chart+Grid (170784408)></code>
<i>Inherited from <code>pygtk_chart.chart.ChartObject</code> (Section 4.2)</i>	
<code>--gsignals--</code>	

5.9 Class Graph



This class represents a graph or the data you want to plot on your LineChart widget.

5.9.1 Methods

<code>--init--(self, name, title, data)</code>
Create a new instance.
Parameters
name: A unique name for the graph. This could be everything. It's just a name used internally for identification. You need to know this if you want to access or delete a graph from a chart. <i>(type=string)</i>
title: The graphs title. This can be drawn on the chart. <i>(type=string)</i>
data: This is the data you want to be visualized. data has to be a list of (x, y) pairs. <i>(type=list of pairs of numbers)</i>
Overrides: <code>object.__init__</code>

do_get_property(*self*, *property*)Overrides: `pygtk_chart.chart.ChartObject.do_get_property`**do_set_property**(*self*, *property*, *value*)Overrides: `pygtk_chart.chart.ChartObject.do_set_property`**has_something_to_draw**(*self*)**get_x_range**(*self*)

Get the the endpoints of the x interval.

Return Value

pair of numbers

get_y_range(*self*)

Get the the endpoints of the y interval.

Return Value

pair of numbers

get_name(*self*)

Get the name of the graph.

Return Value

string

get_title(*self*)

Returns the title of the graph.

Return Value

string

set_title(*self*, *title*)

Set the title of the graph.

Parameters

title: The graph's new title.
(*type=string*)

set_range_calc(*self*, *range_calc*)

get_color(*self*)

Returns the current color of the graph or COLOR_AUTO.

Return Value

a color (see set_color() for details).

set_color(*self*, *color*)

Set the color of the graph. color has to be a (r, g, b) triple where r, g, b are between 0 and 1. If set to COLOR_AUTO, the color will be chosen dynamically.

Parameters

color: The new color of the graph.
(*type=a color*)

get_type(*self*)

Returns the type of the graph.

Return Value

a type constant (see set_type() for details)

set_type(*self*, *type*)

Set the type of the graph to one of these:

- GRAPH_POINTS: only show points
- GRAPH_LINES: only draw lines
- GRAPH_BOTH: draw points and lines, i.e. connect points with lines

Parameters

type: One of the constants above.

get_point_size(*self*)

Returns the radius of the data points.

Return Value

a positive integer

set_point_size(*self*, *size*)

Set the radius of the drawn points.

Parameters

size: The new radius of the points.
(*type=a positive integer in [1, 100]*)

get_fill_to(*self*)

The return value of this method depends on the filling under the graph. See `set_fill_to()` for details.

set_fill_to(*self*, *fill_to*)

Use this method to specify how the space under the graph should be filled. `fill_to` has to be one of these:

- None: don't fill the space under the graph.
- int or float: fill the space to the value specified (setting `fill_to=0` means filling the space between graph and xaxis).
- a Graph object: fill the space between this graph and the graph given as the argument.

The color of the filling is the graph's color with 30% opacity.

Parameters

`fill_to`: (*type=one of the possibilities listed above.*)

get_fill_color(*self*)

Returns the color that is used to fill space under the graph or `COLOR_AUTO`.

set_fill_color(*self*, *color*)

Set which color should be used when filling the space under a graph. If `color` is `COLOR_AUTO`, the graph's color will be used.

Parameters

`color`: (*type=a color or COLOR_AUTO.*)

get_fill_opacity(*self*)

Returns the opacity that is used to fill space under the graph.

set_fill_opacity(*self*, *opacity*)

Set which opacity should be used when filling the space under a graph. The default is 0.3.

Parameters

`opacity`: (*type=float in [0, 1].*)

get_show_values(*self*)

Returns True if y values are shown.

Return Value

boolean

set_show_values(*self*, *show*)

Set whether the y values should be shown (only if graph type is GRAPH_POINTS or GRAPH_BOTH).

Parameters

show: (*type=boolean*)

get_show_title(*self*)

Returns True if the title of the graph is shown.

Return Value

boolean.

set_show_title(*self*, *show*)

Set whether to show the graph's title or not.

Parameters

show: (*type=boolean.*)

add_data(*self*, *data_list*)

Add data to the graph.

Parameters

data_list: (*type=a list of pairs of numbers*)

get_data(*self*)

Returns the data of the graph.

Return Value

a list of x, y pairs.

Overrides: `??GObject.get_data`

Inherited from *pygtk_chart.chart.ChartObject*(Section 4.2)

`draw()`, `get_antialias()`, `get_visible()`, `set_antialias()`, `set_visible()`

Inherited from *??GObject*

`--cmp--()`, `--copy--()`, `--deepcopy--()`, `--delattr--()`, `--gdoc--()`, `--gobject_init--()`,

`__hash__()`, `__new__()`, `__repr__()`, `__setattr__()`, `chain()`, `connect()`, `connect_after()`, `connect_object()`, `connect_object_after()`, `disconnect()`, `disconnect_by_func()`, `emit()`, `emit_stop_by_name()`, `freeze_notify()`, `get_properties()`, `get_property()`, `handler_block()`, `handler_block_by_func()`, `handler_disconnect()`, `handler_is_connected()`, `handler_unblock()`, `handler_unblock_by_func()`, `notify()`, `props()`, `set_data()`, `set_properties()`, `set_property()`, `stop_emission()`, `thaw_notify()`, `weak_ref()`

Inherited from object

`__getattr__()`, `__reduce__()`, `__reduce_ex__()`, `__str__()`

5.9.2 Properties

Name	Description
<i>Inherited from <code>??GObject</code></i>	
<code>__grefcount__</code>	
<i>Inherited from object</i>	
<code>__class__</code>	

5.9.3 Class Variables

Name	Description
<code>__gproperties__</code>	Value: <code>{"name":(gobject.TYPE_STRING, "graph id", "The graph's un...</code>
<code>__gtype__</code>	Value: <code><GType pygtk_chart+line_chart+Graph (170836600)></code>
<i>Inherited from <code>pygtk_chart.chart.ChartObject</code> (Section 4.2)</i>	
<code>__gsignals__</code>	

6 Module pygtk_chart.pie_chart

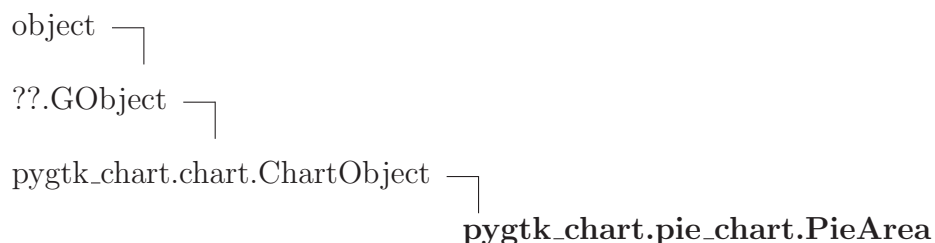
Contains the PieChart widget.

Author: Sven Festersen (sven@sven-festersen.de)

6.1 Variables

Name	Description
COLOR_AUTO	Value: 0
COLORS	Value: [(0.8, 0.0, 0.0), (0.203921568627, 0.396078431373, 0.6431...]

6.2 Class PieArea



6.2.1 Methods

`__init__(self, name, value, label='')`

`x.__init__(...)` initializes x; see `x.__class__.__doc__` for signature

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

`do_get_property(self, property)`

Overrides: `pygtk_chart.chart.ChartObject.do_get_property`

`do_set_property(self, property, value)`

Overrides: `pygtk_chart.chart.ChartObject.do_set_property`

set_value(*self*, *value*)

Set the value of the PieArea.
Parameters
value: (*type=float.*)

get_value(*self*)

Returns the current value of the PieArea.
Return Value

float.

set_color(*self*, *color*)

Set the color of the pie area. Color has to either COLOR_AUTO or a tuple (r, g, b) with r, g, b in [0, 1].
Parameters
color: (*type=a color.*)

get_color(*self*)

Returns the current color of the pie area or COLOR_AUTO.
Return Value

a color.

set_label(*self*, *label*)

Set the label for the pie chart area.
Parameters
label: the new label

(*type=string.*)

get_label(*self*)

Returns the current label of the area.
Return Value

string.

Inherited from *pygtk.chart.chart.ChartObject*(Section 4.2)

draw(), get_antialias(), get_visible(), set_antialias(), set_visible()

Inherited from *??GObject*

--cmp--(), --copy--(), --deepcopy--(), --delattr--(), --gdoc--(), --gobject_init--(),

__hash__(), __new__(), __repr__(), __setattr__(), chain(), connect(), connect_after(), connect_object(), connect_object_after(), disconnect(), disconnect_by_func(), emit(), emit_stop_by_name(), freeze_notify(), get_data(), get_properties(), get_property(), handler_block(), handler_block_by_func(), handler_disconnect(), handler_is_connected(), handler_unblock(), handler_unblock_by_func(), notify(), props(), set_data(), set_properties(), set_property(), stop_emission(), thaw_notify(), weak_ref()

Inherited from object

__getattr__(), __reduce__(), __reduce_ex__(), __str__()

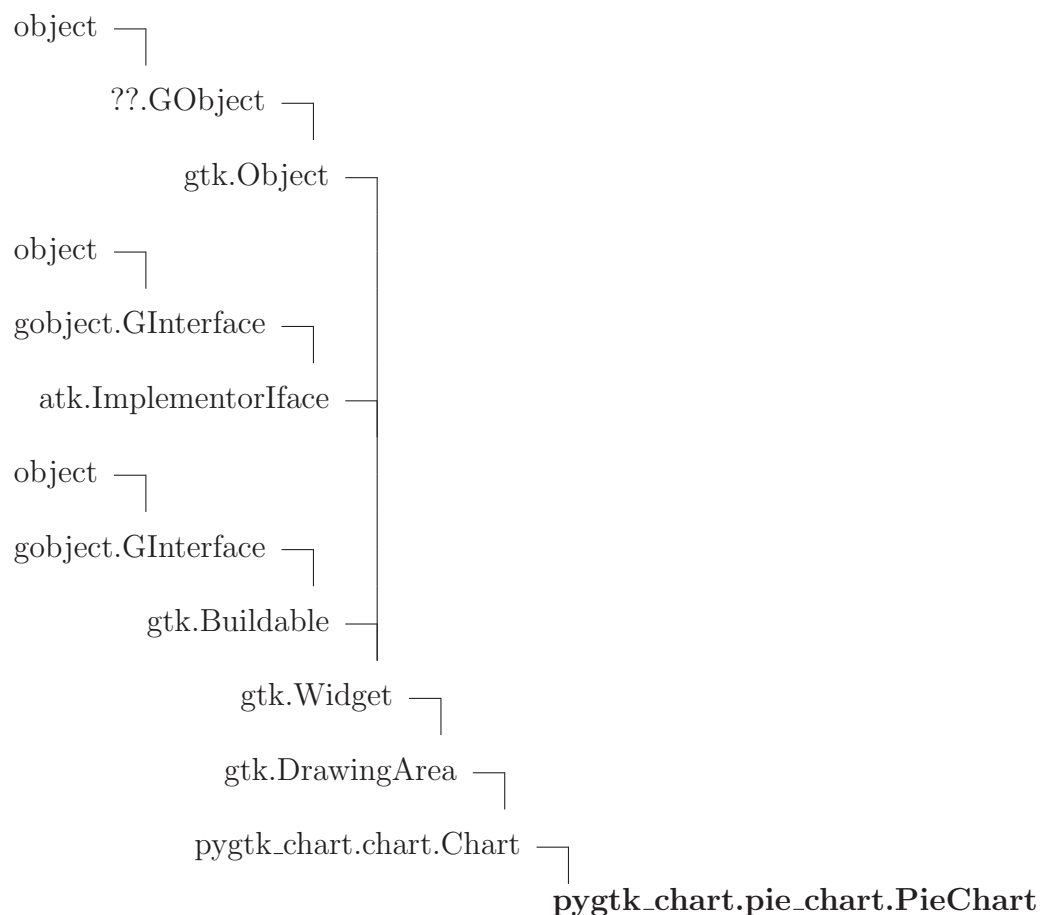
6.2.2 Properties

Name	Description
<i>Inherited from ?? GObject</i>	
__grefcount__	
<i>Inherited from object</i>	
__class__	

6.2.3 Class Variables

Name	Description
__gproperties__	Value: {"name":(gobject.TYPE_STRING, "pie are name", "A unique n...
__gtype__	Value: <GType pygtk.chart.pie.chart.PieArea (170264208)>
<i>Inherited from pygtk.chart.chart.ChartObject (Section 4.2)</i>	
__gsignals__	

6.3 Class **PieChart**



6.3.1 Methods

__init__(*self*)

x.__init__(...) initializes *x*; see *x.__class__.__doc__* for signature

Overrides: *object.__init__* extit(inherited documentation)

do_get_property(*self*, *property*)

do_set_property(*self*, *property*, *value*)

draw(*self*, *context*)

Draw the widget. This method is called automatically. Don't call it yourself. If you want to force a redrawing of the widget, call the `queue_draw()` method.

Parameters

context: The context to draw on.
(*type=cairo.Context*)

Overrides: `gtk.Widget.draw`

add_area(*self*, *area*)**get_pie_area**(*self*, *name*)

Returns the PieArea with the id 'name' if it exists, None otherwise.

Parameters

name: the id of a PieArea
(*type=string*)

Return Value

a PieArea or None.

set_rotate(*self*, *angle*)

Set the rotation angle of the PieChart in degrees.

Parameters

angle: angle in degrees 0 - 360
(*type=integer.*)

get_rotate(*self*)

Get the current rotation angle in degrees.

Return Value

integer.

set_draw_shadow(*self*, *draw*)

Set whether to draw the pie chart's shadow.

Parameters

draw: (*type=boolean.*)

get_draw_shadow(*self*)

Returns True if pie chart currently has a shadow.

Return Value

boolean.

set_draw_labels(*self*, *draw*)

Set whether to draw the labels of the pie areas.

Parameters

draw: (*type=boolean.*)

get_draw_labels(*self*)

Returns True if area labels are shown.

Return Value

boolean.

set_show_percentage(*self*, *show*)

Set whether to show the percentage an area has in its label.

Parameters

show: (*type=boolean.*)

get_show_percentage(*self*)

Returns True if percentages are shown.

Return Value

boolean.

set_enable_scroll(*self*, *scroll*)

Set whether the pie chart can be rotated by scrolling with the mouse wheel.

Parameters

scroll: (*type=boolean.*)

get_enable_scroll(*self*)

Returns True if the user can rotate the pie chart by scrolling.

Return Value

boolean.

set_enable_mouseover(*self*, *mouseover*)

Set whether a mouseover effect should be shown when the pointer enters a pie area.

Parameters

mouseover: (*type=boolean.*)

get_enable_mouseover(*self*)

Returns True if the mouseover effect is enabled.

Return Value

boolean.

set_show_values(*self*, *show*)

Set whether the area's value should be shown in its label.

Parameters

show: (*type=boolean.*)

get_show_values(*self*)

Returns True if the value of a pie area is shown in its label.

Return Value

boolean.

Inherited from *pygtk.chart.chart.Chart*(Section 4.1)

draw_basics(), export_png(), export_svg(), expose()

Inherited from *gtk.DrawingArea*

size()

Inherited from *gtk.Widget*

activate(), add_accelerator(), add_events(), add_mnemonic_label(), can_activate_accel(), child_focus(), child_notify(), class_path(), create_pango_context(), create_pango_layout(), destroy(), do_button_press_event(), do_button_release_event(), do_can_activate_accel(), do_client_event(), do_composited_changed(), do_configure_event(), do_delete_event(), do_destroy_event(), do_direction_changed(), do_drag_begin(), do_drag_data_delete(), do_drag_data_get(), do_drag_data_received(), do_drag_drop(), do_drag_end(), do_drag_leave(), do_drag_motion(), do_enter_notify_event(), do_event(), do_expose_event(), do_focus(), do_focus_in_event(), do_focus_out_event(), do_get_accessible(), do_grab_broken_event(), do_grab_focus(), do_grab_notify(), do_hide(), do_hide_all(), do_hierarchy_changed(), do_key_press_event(), do_key_release_event(), do_leave_notify_event(), do_map(), do_map_event(), do_mnemonic_activate(), do_motion_notify_event(), do_no_expose_event(), do_parent_set(),

do_popup_menu(), do_property_notify_event(), do_proximity_in_event(), do_proximity_out_event(), do_realize(), do_screen_changed(), do_scroll_event(), do_selection_clear_event(), do_selection_get(), do_selection_notify_event(), do_selection_received(), do_selection_request_event(), do_show(), do_show_all(), do_show_help(), do_size_allocate(), do_size_request(), do_state_changed(), do_style_set(), do_unmap(), do_unmap_event(), do_unrealize(), do_visibility_notify_event(), do_window_state_event(), drag_begin(), drag_check_threshold(), drag_dest_add_image_targets(), drag_dest_add_text_targets(), drag_dest_add_uri_targets(), drag_dest_find_target(), drag_dest_get_target_list(), drag_dest_get_track_motion(), drag_dest_set(), drag_dest_set_proxy(), drag_dest_set_target_list(), drag_dest_set_track_motion(), drag_dest_unset(), drag_get_data(), drag_highlight(), drag_source_add_image_targets(), drag_source_add_text_targets(), drag_source_add_uri_targets(), drag_source_get_target_list(), drag_source_set(), drag_source_set_icon(), drag_source_set_icon_name(), drag_source_set_icon_pixbuf(), drag_source_set_icon_stock(), drag_source_set_target_list(), drag_source_unset(), drag_unhighlight(), ensure_style(), error_bell(), event(), freeze_child_notify(), get_accessible(), get_action(), get_activate_signal(), get_allocation(), get_ancestor(), get_child_requisition(), get_child_visible(), get_clipboard(), get_colormap(), get_composite_name(), get_direction(), get_display(), get_events(), get_extension_events(), get_has_tooltip(), get_modifier_style(), get_name(), get_no_show_all(), get_pango_context(), get_parent(), get_parent_window(), get_pointer(), get_root_window(), get_screen(), get_settings(), get_size_request(), get_snapshot(), get_style(), get_tooltip_markup(), get_tooltip_text(), get_tooltip_window(), get_toplevel(), get_visual(), get_window(), grab_add(), grab_default(), grab_focus(), grab_remove(), has_screen(), hide(), hide_all(), hide_on_delete(), input_shape_combine_mask(), intersect(), is_ancestor(), is_composited(), is_focus(), keynav_failed(), list_mnemonic_labels(), map(), menu_get_for_attach_widget(), mnemonic_activate(), modify_base(), modify_bg(), modify_cursor(), modify_fg(), modify_font(), modify_style(), modify_text(), path(), queue_clear(), queue_clear_area(), queue_draw(), queue_draw_area(), queue_resize(), queue_resize_no_redraw(), rc_get_style(), realize(), region_intersect(), remove_accelerator(), remove_mnemonic_label(), render_icon(), reparent(), reset_rc_styles(), reset_shapes(), selection_add_target(), selection_add_targets(), selection_clear_targets(), selection_convert(), selection_owner_set(), selection_remove_all(), send_expose(), set_accel_path(), set_activate_signal(), set_app_paintable(), set_child_visible(), set_colormap(), set_composite_name(), set_direction(), set_double_buffered(), set_events(), set_extension_events(), set_has_tooltip(), set_name(), set_no_show_all(), set_parent(), set_parent_window(), set_redraw_on_allocate(), set_scroll_adjustments(), set_sensitive(), set_set_scroll_adjustments_signal(), set_size_request(), set_state(), set_style(), set_tooltip_markup(), set_tooltip_text(), set_tooltip_window(), set_uposition(), set_usize(), shape_combine_mask(), show(), show_all(), show_now(), size_allocate(), size_request(), style_get_property(), thaw_child_notify(), translate_coordinates(), trigger_tooltip_query(), unmap(), unparent(), unrealize()

Inherited from `gtk.Object`

do_destroy(), flags(), remove_data(), remove_no_notify(), set_flags(), unset_flags()

Inherited from `??GObject`

--cmp--, --copy--, --deepcopy--, --delattr--, --gdoc--, --gobject_init--(),

__hash__(), __new__(), __repr__(), __setattr__(), chain(), connect(), connect_after(), connect_object(), connect_object_after(), disconnect(), disconnect_by_func(), emit(), emit_stop_by_name(), freeze_notify(), get_data(), get_properties(), get_property(), handler_block(), handler_block_by_func(), handler_disconnect(), handler_is_connected(), handler_unblock(), handler_unblock_by_func(), notify(), props(), set_data(), set_properties(), set_property(), stop_emission(), thaw_notify(), weak_ref()

Inherited from atk.ImplementorIface

ref_accessible()

Inherited from gtk.Buildable

add_child(), construct_child(), do_add_child(), do_construct_child(), do_get_internal_child(), do_parser_finished(), do_set_name(), get_internal_child(), parser_finished()

Inherited from object

__getattr__(), __reduce__(), __reduce_ex__(), __str__()

6.3.2 Properties

Name	Description
<i>Inherited from gtk.Widget</i>	
allocation, name, parent, requisition, saved_state, state, style, window	
<i>Inherited from ??GObject</i>	
__grefcount__	
<i>Inherited from object</i>	
__class__	

6.3.3 Class Variables

Name	Description
__gproperties__	Value: {"rotate":(gobject.TYPE_INT, "rotation", "The angle to ro...
__gsignals__	Value: {"area-clicked":(gobject.SIGNAL_RUN_LAST, gobject.TYPE_NO...
__gtype__	Value: <GType pygtk_chart+pie_chart+PieChart (170256872)>

Index

- pygtk_chart (*package*), 2
 - pygtk_chart.bar_chart (*module*), 3–16
 - pygtk_chart.bar_chart.Bar (*class*), 3–5
 - pygtk_chart.bar_chart.BarChart (*class*), 5–10
 - pygtk_chart.bar_chart.MultiBar (*class*), 10–13
 - pygtk_chart.bar_chart.MultiBarChart (*class*), 13–16
 - pygtk_chart.basics (*module*), 17–18
 - pygtk_chart.basics.color_html_to_cairo (*function*), 17
 - pygtk_chart.basics.color_list_from_file (*function*), 18
 - pygtk_chart.basics.color_rgb_to_cairo (*function*), 17
 - pygtk_chart.basics.get_center (*function*), 17
 - pygtk_chart.basics.intersect_ranges (*function*), 17
 - pygtk_chart.basics.is_in_range (*function*), 17
 - pygtk_chart.basics.show_text (*function*), 18
 - pygtk_chart.chart (*module*), 19–29
 - pygtk_chart.chart.Background (*class*), 25–28
 - pygtk_chart.chart.Chart (*class*), 19–23
 - pygtk_chart.chart.ChartObject (*class*), 23–25
 - pygtk_chart.chart.Title (*class*), 28–29
 - pygtk_chart.line_chart (*module*), 30–51
 - pygtk_chart.line_chart.Axis (*class*), 37–40
 - pygtk_chart.line_chart.Graph (*class*), 46–51
 - pygtk_chart.line_chart.graph_new_from_file (*function*), 30
 - pygtk_chart.line_chart.graph_new_from_function (*function*), 30
 - pygtk_chart.line_chart.Grid (*class*), 43–46
 - pygtk_chart.line_chart.LineChart (*class*), 32–37
 - pygtk_chart.line_chart.optimize_sampling (*function*), 30
 - pygtk_chart.line_chart.RangeCalculator (*class*), 31–32
 - pygtk_chart.line_chart.XAxis (*class*), 40–42
 - pygtk_chart.line_chart.YAxis (*class*), 42–43
 - pygtk_chart.pie_chart (*module*), 52–60
 - pygtk_chart.pie_chart.PieArea (*class*), 52–54
 - pygtk_chart.pie_chart.PieChart (*class*), 54–60