

GLUONIX DESIGNER

CONTENTS

Root/Popup	9
Nothing.....	9
Maximize.....	9
Restore.....	9
Minimize	9
Restart.....	9
Close.....	9
Hide	10
Show.....	10
Grab.....	10
After	10
Screen	10
Bind	10
Config.....	10
Config_Get.....	11
Add_Manu.....	11
Add_Sub_Menu.....	11
Add_Separator.....	11
Folder	11
File.....	11
Position.....	11
Size	12
Ratio	12
Locate.....	12
Locate_Reverse	12
Light_Mode.....	12
Dark_Mode.....	12
Update_Colors.....	12
Update_Color	13
Common	13
Copy.....	13
Delete.....	13
Hide	13

Show	13
Animate	13
Animate_Cancel	14
Focus	14
Grab	14
Bind	14
Config.....	14
Config_Get	14
Position.....	14
Size	15
Box	15
Locate.....	15
Locate_Reverse	15
Update_Color	15
Canvas/Scroll	16
Clear.....	16
Refresh.....	16
Bind_Item.....	16
Hide_Item.....	16
Show_Item.....	16
Delete_Item	16
Delete_All	17
Find_Near.....	17
Find_Overlap.....	17
Line.....	17
Polyline.....	17
Add	17
Remove.....	17
Pie.....	18
Arc	18
Circle	18
Rectangle	18
Rectangle2	18
Oval	19
Polygon.....	19
Add	19
Remove.....	19
Text.....	19

Set.....	19
Image.....	20
Set.....	20
Set.....	20
Top	20
Reset	20
Update	20
Update_All	20
Frame	21
Clear.....	21
Bar.....	21
Set.....	21
Get	21
Button	22
Set.....	22
Check	22
Set.....	22
Get	22
Compound.....	23
Set.....	23
Entry	23
Set.....	23
Get	23
Image	24
Set.....	24
Initial.....	24
Rotate	24
Label	24
Set.....	24
Roubel.....	25
Set.....	25
Line.....	25
List	25
Add	25
Remove.....	25
Set.....	26
Reset.....	26
Get	26

Clear.....	26
Top.....	26
Variable.....	27
Create	27
Get	27
Radio.....	27
Set.....	27
Reset.....	27
Scale.....	28
Set.....	28
Get	28
Spinner.....	28
Set.....	28
Get	28
Select.....	29
Add	29
Remove.....	29
Set.....	29
Get	29
Clear.....	29
Sort	29
Separator.....	30
Add	30
Switch	30
Set.....	30
Get	30
Text.....	31
Tag	31
Add	31
Set.....	31
Get	31
Tree	32
Add	32
Edit.....	32
Get	32
Get_All	32
Remove.....	32
Remove_All.....	32

Remove_Selected	33
Current	33
Selected	33
Child	33
Parent	33
Index	33
Expand	33
Select	34
Export	34
Editor	34
Callback	34
Load	34
Add_Rectangle	34
Update_Rectangle	35
Remove_Rectangle	35
Get_Rectangle	35
Player	35
Set	35
Playing	35
State	35
Stop	36
Pause	36
Play	36
Length	36
Time_Get	36
Time_Set	36
Position_Get	36
Position_Set	37
Rate_Get	37
Rate_Set	37
Audio_Get	37
Audio_Set	37
Audio_Delay_Get	37
Audio_Delay_Set	37
Audio_Track	38
Mute	38
Volume_Get	38
Volume_Set	38

Frame_Step	38
Loop	38
Subtitle	38
Subtitle_On	39
Subtitle_Off	39
Subtitle_Tracks	39
Subtitle_Set	39
Subtitle_Delay_Get	39
Subtitle_Delay_Set	39
Queue	39
Next	40
Video	40
Set	40
Playing	40
State	40
Stop	40
Pause	40
Play	40
Length	41
Time_Get	41
Time_Set	41
Position_Get	41
Position_Set	41
Rate_Get	41
Rate_Set	41
Frame_Step	42
Loop	42
Queue	42
Next	42
Bind / Bind_Item	43
Bind Specific	43
For All	43
Config	48
Auto_Dark	48
Background	48
Light_Background	48
Dark_Background	48
Foreground	48

Light_Foreground.....	49
Dark_Foreground	49
Border_Color.....	49
Light_Border_Color	49
Dark_Border_Color	49
Border_Size	50
Resize	50
Resize_Width	50
Resize_Height	50
Move.....	50
Move_Top	50
Move_Left.....	50
Top	50
Left	51
Width.....	51
Height.....	51
Font_Size.....	51
Font_Weight.....	51
Font_Family	51
Scrollbar	52
Vertical	52
Horizontal.....	52
Last	52
Value	52
Ridge	52
Disable.....	53
Path.....	53
URL	53
Array.....	53
Pil	53
Photo.....	53
Rotate	54
Transparent.....	54
Aspect_Ratio	54
Compound	54
Align.....	54
Secure	54
Variable	55

Minimum.....	55
Maximum	55
Increment.....	55
Orient.....	55
Height_List.....	56
Progress	56
Zero	56
Radius.....	56
Shadow_Size	56
Shadow_Color	56
Light_Shadow_Color	57
Dark_Shadow_Color	57
Background_Selected.....	57
Foreground_Selected.....	57
Translucent.....	57
Multiple	58
Select_Background.....	58
Select_Foreground.....	58
Disable_Background	58
Disable_Foreground	58
Hover_Background.....	58
Hover_Foreground	59
Hover_Border_Color.....	59
Hover_Shadow_Color	59
Light_Hover_Background.....	59
Light_Hover_Foreground.....	59
Light_Hover_Border_Color	59
Light_Hover_Shadow_Color.....	60
Dark_Hover_Background.....	60
Dark_Hover_Foreground.....	60
Dark_Hover_Border_Color	60
Dark_Hover_Shadow_Color	60

ROOT/POPUP

NOTHING

Root.Nothing()

Return → False

MAXIMIZE

Root.Maximize()

Return → Maximized Window

RESTORE

Root.Restore()

Return → Restore Window Size

MINIMIZE

Root.Minimize()

Return → Minimize Window To Taskbar

RESTART

Root.Restart()

Return → Restart Application

CLOSE

Root.Close()

Return → Close Application

HIDE

`Root.Hide()`

Return → Hide Application

SHOW

`Root.Show()`

Return → Show Application

GRAB

`Root.Grab(Path=False)`

Return → Returns and Saves screenshot of Application

AFTER

`Root.After(Delay=1000, Function=lambda : Root.Nothing())`

Return → Runs the provided function after milliseconds of delay

SCREEN

`Root.Screen()`

Return → Returns screen size: {'Width':1920, 'Height': 1080}

BIND

`Root.Bind(On_Click=lambda E: Root.Nothing())`

Return → Runs on click function and list of other binds

CONFIG

`Root.Config(Background='#F5F5F5')`

Return → Sets background color and list of configs

CONFIG_GET

Root.Config_Get('Background')

Return → Background color and list of configs

ADD_MANU

Root.Add_Manu(Main=False, Name, Command=False)

Return → Adds Menu Item. If Main is False, will Add to Top Bar, if command is given, No submenu can be added

ADD_SUB_MENU

Root.Add_Sub_Manu(Main, Name, Command=False)

Return → Adds Sub Menu Item. If command is given, No submenu can be added

ADD_SEPARATOR

Root.Add_Separator(Main)

Return → Adds horizontal separator in given step of menu

FOLDER

Root.Folder(Initial="", Title="", Persistent=True)

Return → Request folder input from user.

FILE

Root.File(Initial="", Title="", Multiple=False, Default='.txt', Type=[["Text files", "*.txt"], ["All files", "*.*"]])

Return → Request file input from user

POSITION

Root.Position()

Return → Current position of Application [Left, Top]

SIZE

Root.Size()

Return → Current size of Application [Width, Height]

RATIO

Root.Ratio()

Return → Current size ratio of Application [Width_Ratio, Height_Ratio]

LOCATE

Root.Locate(Width, Height, Left, Top)

Return → Converts the values from percentage to pixels

LOCATE_REVERSE

Root.Locate_Reverse(Width, Height, Left, Top)

Return → Converts the values from pixels to percentage

LIGHT_MODE

Root.Light_Mode()

Return → Changes full application to Light colors

DARK_MODE

Root.Dark_Mode()

Return → Changes full application to Dark colors

UPDATE_COLORS

Root.Update_Colors()

Return → Updates all widgets for automatic light and dark mode colors

UPDATE_COLOR

Root.Update_Color()

Return → Updates automatic light and dark mode colors for current widget

COMMON

Config List: 'Background', 'Resize', 'Resize_Width', 'Resize_Height', 'Move', 'Move_Left', 'Move_Top', 'Left', 'Top', 'Width', 'Height', 'Animate_Left', 'Animate_Top', 'Animate_Width', 'Animate_Height', 'Animate_Radius', 'Animate_Time'

COPY

Widget.Copy(Name=False, Main=False)

Return → Creates a copy of widget and inside widgets with a new name, For Canvas items, no arg Name, only Main

DELETE

Widget.Delete()

Return → Delete widget and all inside widgets

HIDE

Widget.Hide()

Return → Hide Widget

SHOW

Widget.Show()

Return → Show Widget

ANIMATE

Widget.Animate()

Return → Start Widget Animation & Show Widget

ANIMATE_CANCEL

Widget.Animate_Cancel()

Return → Cancel Ongoing Animation

FOCUS

Widget.Focus()

Return → Bring Widget To Focus {Works on Entry, Button, List & Select}

GRAB

Widget.Grab(Path=False)

Return → Returns and Saves screenshot of Widget

BIND

Widget.Bind(On_Click=lambda E: Root.Nothing())

Return → Runs on click function and list of other binds

CONFIG

Widget.Config(Background='#F5F5F5')

Return → Sets background color and list of configs

CONFIG_GET

Widget.Config_Get('Background')

Return → Background color and list of configs

POSITION

Widget.Position(Left=False, Top=False)

Return → Current position of Application [Left, Top]

SIZE

Widget.Size(Width=False, Height=False)

Return → Current size of Application [Width, Height]

BOX

Widget.Box()

Return → Current BBox of Application [Left, Top, Width, Height]

LOCATE

Widget.Locate(Width, Height, Left, Top)

Return → Converts the values from percentage to pixels

LOCATE_REVERSE

Widget.Locate_Reverse(Width, Height, Left, Top)

Return → Converts the values from pixels to percentage

UPDATE_COLOR

Root.Update_Color()

Return → Updates automatic light and dark mode colors for current widget

CANVAS/SCROLL

Config List: 'Background', 'Border_Color', 'Border_Size', 'Resize', 'Resize_Width', 'Resize_Height', 'Move', 'Move_Left', 'Move_Top', 'Left', 'Top', 'Width', 'Height', 'Radius'

Canvas Config Addon: 'Shadow_Size', 'Shadow_Color'

Scroll Config Addon: 'Scrollbar', 'Vertical', 'Horizontal', 'Last'

CLEAR

Canvas.Clear()

Return → Clears all widgets inside canvas or scroll

REFRESH

Canvas.Refresh()

Return → Update all idle tasks

BIND_ITEM

Canvas.Bind_Item(Item, On_Click=lambda E: Root.Nothing())

Return → Runs on click function and list of other binds on an item in canvas

HIDE_ITEM

Canvas.Hide_Item(Item)

Return → Hides an Item

SHOW_ITEM

Canvas.Show_Item(Item)

Return → Show an Item

DELETE_ITEM

Canvas.Delete_Item(Item)

Return → Delete an Item

DELETE_ALL

Canvas.Delete_All()

Return → Delete all Items

FIND_NEAR

Canvas.Find_Near(X, Y)

Return → List of all items close to this pixel

FIND_OVERLAP

Canvas.Find_Overlap(X1, Y1, X2, Y2)

Return → List of all the items overlapping provided rectangle

LINE

Config List: 'Outline', 'Width', 'Height', 'Left', 'Top', 'Thickness', 'Resize'

Canvas.Line()

Return → Draw a line with provided parameters and return item

POLYLINE

Config List: 'Outline', 'Thickness', 'Resize'

Canvas.Polyline()

Return → Draw a line with provided parameters and return item

ADD

Canvas.Add(X, Y)

Return → Adds a new point to polyline

REMOVE

Canvas.Remove(Index)

Return → Removes a point from polyline

PIE

Config List: 'Outline', 'Fill', 'Left', 'Top', 'Radius', 'Thickness', 'Resize', 'Start', 'Extent', 'Translucent'

Canvas.Pie()

Return → Draw a pie chart with provided parameters and return item

ARC

Config List: 'Outline', 'Left', 'Top', 'Radius', 'Thickness', 'Resize', 'Start', 'Extent', 'Translucent'

Canvas.Arc()

Return → Draw an arc with provided parameters and return item

CIRCLE

Config List: 'Outline', 'Fill', 'Left', 'Top', 'Radius', 'Thickness', 'Resize', 'Translucent'

Canvas.Circle()

Return → Draw a circle with provided parameters and return item

RECTANGLE

Config List: 'Outline', 'Fill', 'Width', 'Height', 'Left', 'Top', 'Thickness', 'Resize', 'Translucent'

Canvas.Rectangle()

Return → Draw a rectangle with provided parameters and return item

RECTANGLE2

Config List: 'Outline', 'Fill', 'Width', 'Height', 'Left', 'Top', 'Angle', 'Thickness', 'Resize', 'Translucent'

Canvas.Rectangle2()

Return → Draw a rotated rectangle with provided parameters and return item

OVAL

Config List: 'Outline', 'Fill', 'Width', 'Height', 'Left', 'Top', 'Thickness', 'Resize', 'Translucent'

Canvas.Oval()

Return → Draw an oval with provided parameters and return item

POLYGON

Config List: 'Outline', 'Fill', 'Thickness', 'Resize', 'Translucent'

Canvas.Polygon()

Return → List of points to create polygon, Draw a polygon with provided parameters and return item

ADD

Canvas.Add(X, Y)

Return → Adds a new point to polygon

REMOVE

Canvas.Remove(Index)

Return → Removes a point from polygon

TEXT

Config List: 'Width', 'Height', 'Left', 'Top', 'Color', 'Size', 'Value', 'Weight', 'Font', 'Anchor', 'Justify', 'Resize'

Canvas.Text()

Return → Draws a text and return the item

SET

Widget.Set(Value)

Return → Sets a new text to item

IMAGE

Config List: 'Width', 'Height', 'Left', 'Top', 'Anchor', 'Url', 'Array', 'Pil', 'Photo', 'Resize'

`Canvas.Image()`

Return → Draws an image and return item

SET

`Widget.Set(Path)`

Return → Sets a new image to item

SET

`Widget.Initial()`

Return → Reset image to initial image loaded at design time

TOP

`Scroll.Top()`

Return → Moves scrollbar to top

RESET

`Scroll.Reset()`

Return → Resets the size of scroll frame and moves or hides scrollbars

UPDATE

`Scroll.Update(Widget)`

Return → Updates the size of Scroll region based on provided widget. It is recommended to provide the widget that id most to right and bottom of scroll. You can add an hidden widget so the size given properly to scroll the frame

UPDATE_ALL

`Scroll.Update_All()`

Return → Will update the size of scroll frame based on all the widgets in scroll.

FRAME

Config List: 'Background', 'Border_Color', 'Border_Size', 'Resize', 'Resize_Width', 'Resize_Height', 'Move', 'Move_Left', 'Move_Top', 'Left', 'Top', 'Width', 'Height'

CLEAR

Frame.Clear()

Return → Clears all widgets inside frame

BAR

Config List: 'Background', 'Foreground', 'Border_Color', 'Border_Size', 'Resize', 'Resize_Width', 'Resize_Height', 'Move', 'Move_Left', 'Move_Top', 'Left', 'Top', 'Width', 'Height', 'Progress', 'Zero'

SET

Bar.Set(Value=10)

Return → Sets the bar to specific position

GET

Bar.Set()

Return → The current position of bar

BUTTON

Config List: 'Background', 'Foreground', 'Border_Color', 'Border_Size', 'Resize_Font', 'Resize', 'Resize_Width', 'Resize_Height', 'Move', 'Move_Left', 'Move_Top', 'Left', 'Top', 'Width', 'Height', 'Font_Size', 'Font_Weight', 'Font_Family', 'Value', 'Ridge', 'Disable', 'Disable_Foreground', 'Active_Background', 'Active_Foreground'

SET

```
Button.Set(Value='Click Me')
```

Return → Sets the name of the button

CHECK

Config List: 'Background', 'Foreground', 'Border_Color', 'Border_Size', 'Resize', 'Resize_Width', 'Resize_Height', 'Move', 'Move_Left', 'Move_Top', 'Left', 'Top', 'Width', 'Height'

SET

```
Check.Set(Check=True)
```

Return → Sets the current state of check box

GET

```
Check.Set()
```

Return → The current state of check button

COMPOUND

Config List: 'Background', 'Foreground', 'Border_Color', 'Border_Size', 'Resize_Font', 'Resize', 'Resize_Width', 'Resize_Height', 'Move', 'Move_Left', 'Move_Top', 'Left', 'Top', 'Width', 'Height', 'Font_Size', 'Font_Weight', 'Font_Family', 'Value', 'Path', 'Url', 'Array', 'Pil', 'Rotate', 'Transparent', 'Compound', 'Aspect_Ratio'

SET

Compound.Set(Path, Value)

Return → Sets the image path or name of compound button

ENTRY

Config List: 'Background', 'Foreground', 'Border_Color', 'Border_Size', 'Resize_Font', 'Resize', 'Resize_Width', 'Resize_Height', 'Move', 'Move_Left', 'Move_Top', 'Left', 'Top', 'Width', 'Height', 'Font_Size', 'Font_Weight', 'Font_Family', 'Align', 'Disable', 'Secure', 'Disable_Background', 'Disable_Foreground', 'Select_Background', 'Select_Foreground'

SET

Entry.Set(Value)

Return → Sets the value to entry

GET

Entry.Get()

Return → The value to entry

IMAGE

Config List: 'Background', 'Border_Color', 'Border_Size', 'Resize', 'Resize_Width', 'Resize_Height', 'Move', 'Move_Left', 'Move_Top', 'Left', 'Top', 'Width', 'Height', 'Path', 'Url', 'Array', 'Pil', 'Rotate', 'Transparent', 'Aspect_Ratio'

SET

Image.Set(Path)

Return → Sets the image path and reloads image. If Url is True, Path is Url & if Array is True, Path is cv2 Frame Array (RGB)

INITIAL

Image.Initial()

Return → Reset image to initial image loaded at design time

ROTATE

Image.Rotate(Value=10)

Return → Rotates image in given angle degree values

LABEL

Config List: 'Background', 'Foreground', 'Resize_Font', 'Resize', 'Resize_Width', 'Resize_Height', 'Move', 'Move_Left', 'Move_Top', 'Left', 'Top', 'Width', 'Height', 'Font_Size', 'Font_Weight', 'Font_Family', 'Align', 'Value'

SET

Label.Set(Value)

Return → Sets text value of label

ROUBEL

Config List: 'Background', 'Foreground', 'Resize_Font', 'Resize', 'Resize_Width', 'Resize_Height', 'Move', 'Move_Left', 'Move_Top', 'Left', 'Top', 'Width', 'Height', 'Font_Size', 'Font_Weight', 'Font_Family', 'Value', 'Radius', 'Shadow_Size', 'Shadow_Color'

SET

Label.Set(Value)

Return → Sets text value of label

LINE

Config List: 'Background', 'Border_Color', 'Border_Size', 'Resize', 'Resize_Width', 'Resize_Height', 'Move', 'Move_Left', 'Move_Top', 'Left', 'Top', 'Width', 'Height'

LIST

Config List: 'Background', 'Foreground', 'Border_Color', 'Border_Size', 'Resize_Font', 'Resize', 'Resize_Width', 'Resize_Height', 'Move', 'Move_Left', 'Move_Top', 'Left', 'Top', 'Width', 'Height', 'Font_Size', 'Font_Weight', 'Font_Family', 'Disable', 'Scrollbar', 'Vertical', 'Select_Foreground', 'Select_Background', 'Multiple'

ADD

List.Add(Value)

Return → Adds the value to list

REMOVE

List.Remove(Value)

Return → Removes the value from list

SET

List.Set(Value)

Return → Sets the value to list

RESET

List.Reset()

Return → Resets List Selection

GET

List.Get()

Return → The current value to list

CLEAR

List.Clear()

Return → Delete all the values in list

TOP

List.Top()

Return → Scroll list to top

VARIABLE

CREATE

Variable = Gluonix.Variable()

Return → Creates a variable object for Radio Button

GET

Variable.Get()

Return → Returns current values of variable set by radio buttons

RADIO

Config List: 'Background', 'Foreground', 'Border_Color', 'Border_Size', 'Resize', 'Resize_Width', 'Resize_Height', 'Move', 'Move_Left', 'Move_Top', 'Left', 'Top', 'Width', 'Height', 'Value', 'Variable'

SET

Radio.Set()

Return → Activates current radio and unset all the other radios for same variable

RESET

Radio.Reset()

Return → Unset radio

SCALE

Config List: 'Background', 'Border_Color', 'Border_Size', 'Resize', 'Resize_Width', 'Resize_Height', 'Move', 'Move_Left', 'Move_Top', 'Left', 'Top', 'Width', 'Height', 'Increment', 'Minimum', 'Maximum', 'Orient', 'Disable'

SET

Scale.Set(Value)

Return → Set scale to given value

GET

Scale.Get()

Return → Current values of scale

SPINNER

Config List: 'Background', 'Foreground', 'Border_Color', 'Border_Size', 'Resize_Font', 'Resize', 'Resize_Width', 'Resize_Height', 'Move', 'Move_Left', 'Move_Top', 'Left', 'Top', 'Width', 'Height', 'Font_Size', 'Font_Weight', 'Font_Family', 'Align', 'Increment', 'Minimum', 'Maximum', 'Disable'

SET

Spinner.Set(Value)

Return → Set spinner to given value

GET

Spinner.Get()

Return → Current values of spinner

SELECT

Config List: 'Background', 'Foreground', 'Border_Color', 'Border_Size', 'Resize_Font', 'Resize', 'Resize_Width', 'Resize_Height', 'Move', 'Move_Left', 'Move_Top', 'Left', 'Top', 'Width', 'Height', 'Font_Size', 'Font_Weight', 'Font_Family', 'Disable'

ADD

Select.Add(Value)

Return → Adds the value to select

REMOVE

Select.Remove(Value)

Return → Removes the value from select

SET

Select.Set(Value)

Return → Sets the value to select

GET

Select.Get()

Return → The current value to select

CLEAR

Select.Clear()

Return → Delete all the values in select

SORT

Select.Sort()

Return → Sorts list values A - Z

SEPARATOR

Config List: 'Background', 'Border_Color', 'Border_Size', 'Resize', 'Resize_Width', 'Resize_Height', 'Move', 'Move_Left', 'Move_Top', 'Left', 'Top', 'Width', 'Height'

ADD

Separator.Add(Frame)

Return → Adds frame to control the movement and size, Need two frames and open provided, frames can't be changed.

SWITCH

Config List: 'Background', 'Border_Color', 'Border_Size', 'Resize', 'Resize_Width', 'Resize_Height', 'Move', 'Move_Left', 'Move_Top', 'Left', 'Top', 'Width', 'Height'

SET

Check.Set(Check=True)

Return → Sets the current state of check box

GET

Check.Get()

Return → The current state of check button

TEXT

Config List: 'Background', 'Foreground', 'Border_Color', 'Border_Size', 'Resize', 'Resize_Width', 'Resize_Height', 'Move', 'Move_Left', 'Move_Top', 'Left', 'Top', 'Width', 'Height', 'Font_Size', 'Font_Weight', 'Disable'

TAG

```
Text.Tag(Name, Font_Size=12, Font_Weight='normal', Font='Times New Roman',  
Foreground="#000000", Background="#FFFFFF")
```

Return → Creates a custom tag to use in text

ADD

```
Text.Add(Value, Tag='Default')
```

Return → Sets the current state of check box

SET

```
Text.Set(Value)
```

Return → Sets values to text area

GET

```
Text.Get()
```

Return → The values of text area

TREE

Config List: 'Background', 'Foreground', 'Border_Color', 'Border_Size', 'Resize', 'Resize_Width', 'Resize_Height', 'Move', 'Move_Left', 'Move_Top', 'Left', 'Top', 'Width', 'Height', 'Font_Size', 'Font_Weight', 'Background_Selected', 'Foreground_Selected '

ADD

`Tree.Add(Name, Parent='', Index='end', Value=[], ID=None, Path=None)`

Return → Create a tree item with specific value. Path for Image file.

EDIT

`Tree.Edit(Name=False, Value=False, Tag=False, ID=False)`

Return → Edits parameters of on focus tree item or by providing the id of item

GET

`Tree.Get(ID=False)`

Return → The values of on focus tree item or by providing the id of item

GET_ALL

`Tree.Get_All(ID=False)`

Return → The everything of on focus tree item or by providing the id of item

REMOVE

`Tree.Remove(ID)`

Return → Deletes given item

REMOVE_ALL

`Tree.Remove_All()`

Return → Deletes all items in the tree

REMOVE_SELECTED

Tree.Remove_Selected()

Return → Deletes all selected items in the tree

CURRENT

Tree.Current()

Return → The item in focus

SELECTED

Tree.Selected()

Return → List of selected items

CHILD

Tree.Child(ID)

Return → List of all children of item

PARENT

Tree.Parent(ID)

Return → Parent of given item

INDEX

Tree.Index(ID)

Return → Index of an item

EXPAND

Tree.Expand(ID)

Return → Expand the current tree item

SELECT

Tree.Select(ID)

Return → Makes specific item in focus

EXPORT

Tree.Export(Path)

Return → Exports tree structure to text file '.txt'

EDITOR

Setup: Editor = Canvas.Editor()

All Add Remove Update Get functions work for all Items

Line, Polyline, Polygon Args: Points, Angle, Thickness, Color

Quadrilateral Args: P1, P2, P3, P4, Angle, Thickness, Color

Rectangle Args: X, Y, Width, Height, Angle, Thickness, Color

Circle Args: X, Y, Circle, Thickness, Color

CALLBACK

Editor.Callback = Function ()

Return → Returns any changes made to items on Canvas

LOAD

Editor.Load(Path=None)

Return → Path for Image file.

ADD_RECTANGLE

Editor.Add_Rectangle()

Return → Adds a rectangle to canvas window, return an ID

UPDATE_RECTANGLE

Editor.Update_Rectangle(ID=None)

Return → Update a specific rectangle parameter

Disabled arg will disable the item from transformation

REMOVE_RECTANGLE

Editor.Remove_Rectangle(ID=None)

Return → Remove a specific rectangle parameter

GET_RECTANGLE

Editor.Get_Rectangle(ID=None)

Return → Get a specific rectangle parameter

PLAYER

Setup: Player = Canvas.Player() or Player = Frame.Player()

Required python vlc lib and vlc player installed in the system

SET

Player.Set(Path='')

Return → Path for Video file

PLAYING

Player.Playing()

Return → Returns bool for if video is playing

STATE

Player.State()

Return → Returns state of the player

STOP

Player.Stop()

Return → Stop video player

PAUSE

Player.Pause()

Return → Pause video player

PLAY

Player.Play()

Return → Play video player

LENGTH

Player.Length()

Return → Returns length of the video

TIME_GET

Player.Time_Get()

Return → Returns current video time

TIME_SET

Player.Time_Set(Value)

Return → Set current video time

POSITION_GET

Player.Position_Get()

Return → Returns current video position

POSITION_SET

Player.Position_Set(Value)

Return → Set current video position

RATE_GET

Player.Rate_Get()

Return → Returns current video rate

RATE_SET

Player.Rate_Set(Value)

Return → Set current video rate

AUDIO_GET

Player.Audio_Get()

Return → Returns current video audio track id

AUDIO_SET

Player.Audio_Set(ID)

Return → Set current video audio track

AUDIO_DELAY_GET

Player.Audio_Delay_Get()

Return → Returns current video audio delay in ms

AUDIO_DELAY_SET

Player.Audio_Delay_Set(Microseconds)

Return → Set current video audio delay in ms

AUDIO_TRACK

Player.Audio_Track()

Return → Return all audio track descriptions of the video

MUTE

Player.Mute()

Return → Mute video player sound

VOLUME_GET

Player.Volume_Get()

Return → Returns current video volume level

VOLUME_SET

Player.Volume_Set(Value)

Return → Set current video volume level

FRAME_STEP

Player.Frame_Step()

Return → Display next video frame

LOOP

Player.Loop()

Return → Loop the video on repeat

SUBTITLE

Player.Subtitle(Path)

Return → Add subtitle to the video

SUBTITLE_ON

Player.Subtitle_On()

Return → Turn subtitles on

SUBTITLE_OFF

Player.Subtitle_Off(Path)

Return → Turn subtitles of

SUBTITLE_TRACKS

Player.Subtitle_Tracks()

Return → All subtitles track description

SUBTITLE_SET

Player.Subtitle_Set(ID)

Return → Sets subtitle track on the video

SUBTITLE_DELAY_GET

Player.Subtitle_Delay_Get()

Return → Returns current video subtitle delay in ms

SUBTITLE_DELAY_SET

Player.Subtitle_Delay_Set(Microseconds)

Return → Set current video subtitle delay in ms

QUEUE

Player.Queue(Paths)

Return → List of video paths for queue

NEXT

`Player.Next()`

Return → Play next video

VIDEO

Setup: `Widget = Canvas.Video()`

Required python cv2 lib

SET

`Player.Set(Path='')`

Return → Path for Video file

PLAYING

`Player.Playing()`

Return → Returns bool for if video is playing

STATE

`Player.State()`

Return → Returns state of the player

STOP

`Player.Stop()`

Return → Stop video player

PAUSE

`Player.Pause()`

Return → Pause video player

PLAY

Player.Play()

Return → Play video player

LENGTH

Player.Length()

Return → Returns length of the video

TIME_GET

Player.Time_Get()

Return → Returns current video time

TIME_SET

Player.Time_Set(Value)

Return → Set current video time

POSITION_GET

Player.Position_Get()

Return → Returns current video position

POSITION_SET

Player.Position_Set(Value)

Return → Set current video position

RATE_GET

Player.Rate_Get()

Return → Returns current video rate

RATE_SET

Player.Rate_Set(Value)

Return → Set current video rate

FRAME_STEP

Player.Frame_Step()

Return → Display next video frame

LOOP

Player.Loop()

Return → Loop the video on repeat

QUEUE

Player.Queue(Paths)

Return → List of video paths for queue

NEXT

Player.Next()

Return → Play next video

BIND / BIND_ITEM

BIND SPECIFIC

Cursor_Hand

Cursor_Loading

Cursor_Resize_Vertical

Cursor_Resize_Horizontal

Cursor_Arrow

Cursor

On_Mouse_Wheel

On_Show (**No Event**)

On_Hide (**No Event**)

On_Close (Root & Popup) (**No Event**)

On_Resize (Root, Popup, Frame, Canvas & Scroll) (**No Event**)

On_Change (List, Select, Check, Switch & Radio) (**No Event**)

FOR ALL

On_Configure

On_Destroy

On_Expose

On_Visibility

On_Motion

On_Click

On_Release

On_Double_Click

On_Triple_Click

On_Middle_Click

On_Middle_Release

On_Middle_Double_Click

On_Middle_Triple_Click

On_Right_Click

On_Right_Release

On_Right_Double_Click

On_Right_Triple_Click

On_Drag

On_Middle_Drag

On_Right_Drag

On_Hover_In

On_Hover_Out

On_Key

On_Key_Release

On_Focus_Out

On_Map

On_Unmap

On_Copy

On_Cut

On_Paste

On_Undo

On_Redo

On_Control_Click

On_Control_Release

On_Control_Double_Click

On_Control_Triple_Click

On_Control_Middle_Click

On_Control_Middle_Release

On_Control_Middle_Double_Click

On_Control_Middle_Triple_Click

On_Control_Right_Click

On_Control_Right_Release

On_Control_Right_Double_Click

On_Control_Right_Triple_Click

On_Control_Drag

On_Control_Middle_Drag

On_Control_Right_Drag

On_Control_Mouse_Wheel

On_Control_Hover_In

On_Control_Hover_Out

On_Alt_Click

On_Alt_Release

On_Alt_Double_Click

On_Alt_Triple_Click

On_Alt_Middle_Click

On_Alt_Middle_Release

On_Alt_Middle_Double_Click

On_Alt_Middle_Triple_Click

On_Alt_Right_Click

On_Alt_Right_Release

On_Alt_Right_Double_Click

On_Alt_Right_Triple_Click

On_Alt_Drag

On_Alt_Middle_Drag

On_Alt_Right_Drag

On_Alt_Mouse_Wheel

On_Alt_Hover_In

On_Alt_Hover_Out

On_Shift_Click

On_Shift_Release

On_Shift_Double_Click

On_Shift_Triple_Click

On_Shift_Middle_Click

On_Shift_Middle_Release

On_Shift_Middle_Double_Click

On_Shift_Middle_Triple_Click

On_Shift_Right_Click

On_Shift_Right_Release

On_Shift_Right_Double_Click

On_Shift_Right_Triple_Click

On_Shift_Drag

On_Shift_Middle_Drag

On_Shift_Right_Drag

On_Shift_Mouse_Wheel

On_Shift_Hover_In

On_Shift_Hover_Out

CONFIG

Multiple values in config can be set and get at same time

```
Widget.Config(Background='#F5F5F5', Foreground='red', .....
```

```
Widget.Config_Get('Background', 'Foreground', .....
```

Will return a dictionary of requested configurations

AUTO_DARK

Sets the auto dark option of widget. With True, widget will automatically convert light colors to dark using inversion method.

```
Widget.Config(Auto_Dark=False)
```

BACKGROUND

Sets the background of widget. Can be simple HTML values like 'red', 'green', etc. or any Hex color value '#F5F5F5'. False, it will take the background of the Main Frame of the Widget

```
Widget.Config(Background='#F5F5F5')
```

LIGHT_BACKGROUND

Sets the light mode background of widget. Can be simple HTML values like 'red', 'green', etc. or any Hex color value '#F5F5F5'. False, it will take the background of the Main Frame of the Widget

```
Widget.Config(Light_Background='#F5F5F5')
```

DARK_BACKGROUND

Sets the dark mode background of widget. Can be simple HTML values like 'red', 'green', etc. or any Hex color value '#F5F5F5'. False, it will take the background of the Main Frame of the Widget

```
Widget.Config(Dark_Background='#F5F5F5')
```

FOREGROUND

Sets the foreground of widget. Can be simple HTML values like 'red', 'green', etc. or any Hex color value '#F5F5F5'


```
Widget.Config(Foreground='#F5F5F5')
```

LIGHT_FOREGROUND

Sets the light mode foreground of widget. Can be simple HTML values like 'red', 'green', etc. or any Hex color value '#F5F5F5'. False, it will take the background of the Main Frame of the Widget

```
Widget.Config(Light_Foreground='#F5F5F5')
```

DARK_FOREGROUND

Sets the dark mode foreground of widget. Can be simple HTML values like 'red', 'green', etc. or any Hex color value '#F5F5F5'. False, it will take the background of the Main Frame of the Widget

```
Widget.Config(Dark_Foreground='#F5F5F5')
```

BORDER_COLOR

Sets the border color of widget. Can be simple HTML values like 'red', 'green', etc. or any Hex color value '#F5F5F5'. This feature does not work on LITE objects.

```
Widget.Config(Border_Color='#000000')
```

LIGHT_BORDER_COLOR

Sets the light mode border color of widget. Can be simple HTML values like 'red', 'green', etc. or any Hex color value '#F5F5F5'. False, it will take the background of the Main Frame of the Widget

```
Widget.Config(Light_Border_Color='#F5F5F5')
```

DARK_BORDER_COLOR

Sets the dark mode border color of widget. Can be simple HTML values like 'red', 'green', etc. or any Hex color value '#F5F5F5'. False, it will take the background of the Main Frame of the Widget

```
Widget.Config(Dark_Border_Color='#F5F5F5')
```

BORDER_SIZE

Sets the border size of widget. In pixel values. This feature does not work on LITE objects.

```
Widget.Config(Border_Size=2)
```

RESIZE

Defines if the widget can be resized when main root window resizes.

```
Widget.Config(Resize=True)
```

RESIZE_WIDTH

Defines if the widget can be resized only width wise main root window resizes.

```
Widget.Config(Resize_Width=True)
```

RESIZE_HEIGHT

Defines if the widget can be resized only height wise main root window resizes.

```
Widget.Config(Resize_Height=True)
```

MOVE

Defines if the widget can be moved when main root window resizes.

```
Widget.Config(Move=True)
```

MOVE_TOP

Defines if the widget can be moved only up and down main root window resizes.

```
Widget.Config(Move_Top=True)
```

MOVE_LEFT

Defines if the widget can be resized only left and right main root window resizes.

```
Widget.Config(Move_Left=True)
```

TOP

Changes the top position of widget. Only accepted in pixels.

```
Widget.Config(Top=100)
```

LEFT

Changes the left position of widget. Only accepted in pixels.

```
Widget.Config(Left=150)
```

WIDTH

Changes the width of widget. Only accepted in pixels.

```
Widget.Config(Width=100)
```

* If the inside widget is provided in percentage, changed the main widgets will affect the inside widget after the application restarts

HEIGHT

Changes the height of widget. Only accepted in pixels.

```
Widget.Config(Height=100)
```

* If the inside widget is provided in percentage, changed the main widgets will affect the inside widget after the application restarts

FONT_SIZE

Font size for the widgets which support text. Only accepted in pixels.

```
Widget.Config(Font_Size=20)
```

FONT_WEIGHT

Font weight for the widgets which support text. 'normal', 'bold'

```
Widget.Config(Font_Weight='normal')
```

FONT_FAMILY

Font family for the widgets which support text. Only tKinter specific.

'Times New Roman', 'Helvetica', etc.

`Widget.Config(Font_Family= 'Times New Roman')`

SCROLLBAR

Width of scrollbar. Only accepted in pixels.

`Widget.Config(Scrollbar=20)`

* Exclusive to scroll frame

VERTICAL

If vertical scroll bar should be displayed at initial state of frame.

`Widget.Config(Vertical=True)`

* Exclusive to scroll frame

HORIZONTAL

If horizontal scroll bar should be displayed at initial state of frame.

`Widget.Config(Horizontal=True)`

* Exclusive to scroll frame

LAST

Define the last widget of scrolls frame, the widget which is most right and bottom of frame.

`Widget.Config>Last=Widget)`

* Exclusive to scroll frame

VALUE

Sets text value for widgets that support text.

`Widget.Config(Value='Click Here')`

RIDGE

If the button should have ridged border

`Widget.Config(Ridge=True)`

* Exclusive to button

DISABLE

Disables the function of widget

```
Widget.Config(Disable=True)
```

* Exclusive to button, list, entry, select, spinner, scale, text

PATH

Gives the image path for widget

```
Widget.Config(Path='./Image.png')
```

* Exclusive to image, compound, canvas image

URL

Defines if image path given is a http url

```
Widget.Config(Url=True)
```

* Exclusive to image, compound, canvas image

ARRAY

Defines if image path is cv2 array

```
Widget.Config(Array=True)
```

* Exclusive to image, compound, canvas image

PIL

Defines if image path is pillow image

```
Widget.Config(Pil=True)
```

* Exclusive to image, compound, canvas image

PHOTO

Defines if image path is tk photo image

```
Widget.Config(Photo=True)
```

* Exclusive to canvas image

ROTATE

Rotates the image to specific degree of angle

```
Widget.Config(Rotate=90)
```

* Exclusive to image, compound, canvas image

TRANSPARENT

Defines if the provided image support to be transparent in nature.

```
Widget.Config(Transparent=True)
```

* Exclusive to image, compound, canvas image

ASPECT_RATIO

If the given image should keep aspect ration when fitting to current widget.

```
Widget.Config(Aspect_Ratio=True)
```

* Exclusive to image, compound, canvas image

COMPOUND

Location of text on the image. 'left', 'right', 'top', 'bottom', 'center'

```
Widget.Config(Compound='center')
```

* Exclusive to compound

ALIGN

Location of text in widget

```
Widget.Config(Align='center')
```

* Entry: 'left', 'right', 'center'

* Spinner: 'left', 'right', 'center'

* Label: 'n', 'e', 'w', 's', 'ne', 'nw', 'se', 'sw', 'center'

SECURE

Hides the entry text with *

```
Widget.Config(Secure=True)
```

* Exclusive to Entry

VARIABLE

Provides the variable to multiple radio buttons to store the radio value.

```
Widget.Config(Variable=Temp_Variable)
```

* Exclusive to Radio

MINIMUM

Minimum value for widget

```
Widget.Config(Minimum=0)
```

* Exclusive to Scale, Spinner

MAXIMUM

Maximum value for widget

```
Widget.Config(Maximum=100)
```

* Exclusive to Scale, Spinner

INCREMENT

Increment step for widget

```
Widget.Config(Increment=1)
```

* Exclusive to Scale, Spinner

ORIENT

Orientation of the widget

```
Widget.Config(Orient='Horizontal')
```

* Exclusive to Scale, Seperator

HEIGHT_LIST

Height of the list when on click widget

```
Widget.Config(Height_List=500)
```

* Exclusive to Select

PROGRESS

Initial value for the widget

```
Widget.Config(Progress=10)
```

* Exclusive to Bar

ZERO

Position of widget. 'Left', 'Right', 'Top', 'Bottom'

```
Widget.Config(Zero='Left')
```

* Exclusive to Bar

RADIUS

Radius for rounded corners.

```
Widget.Config(Radius=20)
```

* Exclusive to Canvas, Roubel

SHADOW_SIZE

Applies shadow size to all sides.

```
Widget.Config(Shadow_Size=10)
```

* Exclusive to Canvas, Roubel

SHADOW_COLOR

Sets the shadow color of widget. Can be simple HTML values like 'red', 'green', etc. or any Hex color value '#F5F5F5'. False, it will take the background of the Main Frame of the Widget


```
Widget.Config(Shadow_Color='#FF0000')
```

* Exclusive to Canvas, Roubel

LIGHT_SHADOW_COLOR

Sets the light mode shadow color of widget. Can be simple HTML values like 'red', 'green', etc. or any Hex color value '#F5F5F5'. False, it will take the background of the Main Frame of the Widget

```
Widget.Config(Light_Shadow_Color='#F5F5F5')
```

DARK_SHADOW_COLOR

Sets the dark mode shadow color of widget. Can be simple HTML values like 'red', 'green', etc. or any Hex color value '#F5F5F5'. False, it will take the background of the Main Frame of the Widget

```
Widget.Config(Dark_Shadow_Color='#F5F5F5')
```

BACKGROUND_SELECTED

Background color of current selected item.

```
Widget.Config(Background_Selected='#FFFFFF')
```

* Exclusive to Tree

FOREGROUND_SELECTED

Foreground color of current selected item.

```
Widget.Config(Foreground_Selected='#000000')
```

* Exclusive to Tree

TRANSLUCENT

Makes Fill Color Translucent.

```
Widget.Config(Translucent=True)
```

* Exclusive to Canvas Item Rectangle, Rectangle2, Circle, Oval, Arc, Pie, & Polygon

MULTIPLE

Makes Multiple Selection.

```
Widget.Config(Multiple=True)
```

* Exclusive to List

SELECT_BACKGROUND

Sets the select text background of widget. Can be simple HTML values like 'red', 'green', etc. or any Hex color value '#F5F5F5'. False, it will take the background of the Main Frame of the Widget

```
Widget.Config(Select_Background='#F5F5F5')
```

SELECT_FOREGROUND

Sets the select text foreground of widget. Can be simple HTML values like 'red', 'green', etc. or any Hex color value '#F5F5F5'

```
Widget.Config(Select_Foreground='#F5F5F5')
```

DISABLE_BACKGROUND

Sets the disabled background of widget. Can be simple HTML values like 'red', 'green', etc. or any Hex color value '#F5F5F5'. False, it will take the background of the Main Frame of the Widget

```
Widget.Config(Disable_Background='#F5F5F5')
```

DISABLE_FOREGROUND

Sets the disabled foreground of widget. Can be simple HTML values like 'red', 'green', etc. or any Hex color value '#F5F5F5'

```
Widget.Config(Disable_Foreground='#F5F5F5')
```

HOVER_BACKGROUND

Sets the hover background of widget. Can be simple HTML values like 'red', 'green', etc. or any Hex color value '#F5F5F5'. False, it will take the background of the Main Frame of the Widget

```
Widget.Config(Hover_Background='#F5F5F5')
```

HOVER_FOREGROUND

Sets the hover foreground of widget. Can be simple HTML values like 'red', 'green', etc. or any Hex color value '#F5F5F5'. False, it will take the background of the Main Frame of the Widget

```
Widget.Config(Hover_Foreground='#F5F5F5')
```

HOVER_BORDER_COLOR

Sets the hover border color of widget. Can be simple HTML values like 'red', 'green', etc. or any Hex color value '#F5F5F5'. False, it will take the background of the Main Frame of the Widget

```
Widget.Config(Hover_Border_Color='#F5F5F5')
```

HOVER_SHADOW_COLOR

Sets the hover shadow color of widget. Can be simple HTML values like 'red', 'green', etc. or any Hex color value '#F5F5F5'. False, it will take the background of the Main Frame of the Widget

```
Widget.Config(Hover_Shadow_Color='#F5F5F5')
```

LIGHT_HOVER_BACKGROUND

Sets the light mode hover background of widget. Can be simple HTML values like 'red', 'green', etc. or any Hex color value '#F5F5F5'. False, it will take the background of the Main Frame of the Widget

```
Widget.Config(Light_Hover_Background='#F5F5F5')
```

LIGHT_HOVER_FOREGROUND

Sets the light mode hover foreground of widget. Can be simple HTML values like 'red', 'green', etc. or any Hex color value '#F5F5F5'. False, it will take the background of the Main Frame of the Widget

```
Widget.Config(Light_Hover_Foreground='#F5F5F5')
```

LIGHT_HOVER_BORDER_COLOR

Sets the light mode hover border color of widget. Can be simple HTML values like 'red', 'green', etc. or any Hex color value '#F5F5F5'. False, it will take the background of the Main Frame of the Widget

```
Widget.Config(Light_Hover_Border_Color='#F5F5F5')
```

LIGHT_HOVER_SHADOW_COLOR

Sets the light mode hover shadow color of widget. Can be simple HTML values like 'red', 'green', etc. or any Hex color value '#F5F5F5'. False, it will take the background of the Main Frame of the Widget

```
Widget.Config(Light_Hover_Shadow_Color='#F5F5F5')
```

DARK_HOVER_BACKGROUND

Sets the dark mode hover background of widget. Can be simple HTML values like 'red', 'green', etc. or any Hex color value '#F5F5F5'. False, it will take the background of the Main Frame of the Widget

```
Widget.Config(Dark_Hover_Background='#F5F5F5')
```

DARK_HOVER_FOREGROUND

Sets the dark mode hover foreground of widget. Can be simple HTML values like 'red', 'green', etc. or any Hex color value '#F5F5F5'. False, it will take the background of the Main Frame of the Widget

```
Widget.Config(Dark_Hover_Foreground='#F5F5F5')
```

DARK_HOVER_BORDER_COLOR

Sets the dark mode hover border color of widget. Can be simple HTML values like 'red', 'green', etc. or any Hex color value '#F5F5F5'. False, it will take the background of the Main Frame of the Widget

```
Widget.Config(Dark_Hover_Border_Color='#F5F5F5')
```

DARK_HOVER_SHADOW_COLOR

Sets the dark mode hover shadow color of widget. Can be simple HTML values like 'red', 'green', etc. or any Hex color value '#F5F5F5'. False, it will take the background of the Main Frame of the Widget

```
Widget.Config(Dark_Hover_Shadow_Color='#F5F5F5')
```

