
AHK Object Oriented GUIs Documentation

Release alpha

RUNIE

Jun 06, 2018

Table of Contents

1	Getting Started	3
1.1	Installation	3
1.2	Testing installation	3
2	Additional information	5
2.1	Namespace reservations	5
3	Documentation	7
3.1	GuiBase	7
3.2	Base types	10
3.3	Control types	11
3.4	Position types	14
3.5	Gui events	14
3.6	Control events	15
4	Examples	17
4.1	Class example	17
5	About	19
5.1	Need help?	19
6	Indices and tables	21

AHK-OOG is a library makes it easy to work with GUIs in AutoHotkey in an object-oriented fashion.

CHAPTER 1

Getting Started

It is recommended you have some prior experience working with Guis in AutoHotkey, as this library builds upon existing functionality

1.1 Installation

1. Download or clone the repository at <https://github.com/Run1e/AHK-Object-Oriented-GUIs>
2. Copy the `gui` folder into your project
3. Include the library in your project by doing `#Include gui\GuiBase.ahk`

1.2 Testing installation

Here's a small snippet you can try to check if your installation was successful:

```
MyGui := new GuiBase("Title", "-MinimizeBox")
MyGui.AddButton("w200", "Button Text")
MyGui.Show()
```

It will create and show a small, simplistic GUI consisting of a single button.

The main components of the library lives under one namespace, namely `GuiBase`. However the library introduces a few other namespace intrusions.

2.1 Namespace reservations

Type (*cls*)

This function is just a shorthand for `cls.__Class`

Parameters `cls` – Any class object/instance.

Returns `cls.__Class`

IsInstance (*obj, cls**)

Similar to Python's function of the same name. It returns `true` if the type (or technically class name) of `obj` matches any of those specified in `cls`.

Parameters

- `obj` – A class object/instance.
- `cls` – Any variadic amount of class objects/instances.

Returns Boolean value.

IndirectReferenceDelete (*this*)

This function should not be touched by the user, it's used by *IndirectReferenceHolder*.

A real world example of `IsInstance()`

```
; assume MyGui is a gui with one text control

MyControl := MyGui.Controls[1] ; gets the control

msgbox % IsInstance(MyControl, GuiBase.TextControl)
; the line above will result to true, since the control
```

(continues on next page)

(continued from previous page)

```
; is of type GuiBase.TextControl

msgbox % IsInstance(MyControl, GuiBase.ControlType)
; the line above will also result to true, since the
; control type (GuiBase.TextControl) inherit from
; GuiBase.ContentControlType which in turn inherit
; from GuiBase.ControlType

msgbox % IsInstance(MyControl, GuiBase, GuiBase.ContentControlType)
; also true. It'll check if MyControl is an instance
; of GuiBase first and since it isn't, it'll continue and
; check the next one, which it of course is an instance of
```

Only one class is bundled with the library at the moment. Currently the only usage of this class is in the constructor of *GuiBase.ControlType*.

class IndirectReferenceHolder

This class creates an indirect reference to an object and holds the original reference itself.

It uses all five meta-functions to create a layer between the actual object and what the user has access to.

3.1 GuiBase

The *GuiBase* class is the main class that represents a GUI.

It can either be instantiated directly, or you can extend upon it for more complicated GUIs. See the examples.

GuiBase.__Version

Contains the installed version of the library.

GuiBase.GetGui (*hwnd*)

Parameters *hwnd* – GUI hwnd.

Returns A *GuiBase* instance, if found.

class GuiBase

Represents a GUI object.

Title

Get or set the window title.

hwnd

Contains the window handle.

ahk_id

Shorthand for "ahk_id" . hwnd

Visible

Bool indicating whether the window is visible or hidden.

BackgroundColor

Get or set the background color.

ControlColor

Get or set the control (foreground) color.

Controls

An array of all control instances the GUI instance has created.

Position

Instance of *GuiBase.WindowPosition* representing the position and size of the GUI window.

New (*Title* := "AutoHotkey Window", *Options* := "")

Creates a new instance of the class.

Note: You shouldn't call this meta-function directly, but use the `new` keyword. See the AutoHotkey documentation on [constructing and deconstructing objects](#).

Parameters

- **Title** – Title of the window.
- **Options** – Options string.

Returns A *GuiBase* instance.

Show (*Options* := "")

Shows the GUI window.

Parameters **Options** – Options string.

Hide (*Options* := "")

Hides the GUI window.

Parameters **Options** – Options string.

Destroy (*Options* := "")

Destroys the GUI, and frees all *GuiBase.ControlType* instances related to it. It's a good idea to clear all references you have kept yourself so the *GuiBase* instance can be freed properly.

Options (*Options*)

Change the options of the GUI.

Parameters **Options** – Options string.

SetDefault ()

Sets this GUI as the default GUI.

SetDefaultListView (*ListView*)

Sets the default *ListView* control.

Parameters **ListView** – *GuiBase.ListViewControl* instance.

Control (*Command* := "", *Control* := "", *ControlParams* := "")

Calls *GuiControl*.

Parameters

- **Command** – The *GuiControl* command to perform.
- **Control** – The control instance to apply the command on.
- **ControlParams** – The parameters for the command.

GetControl (*hwnd*)

Gets a control instance.

Parameters **hwnd** – *hwnd* of the control.

Returns A control instance, if found.

Margins (*x := "", y := ""*)

Sets the control spacing margins for newly created controls.

Parameters

- **x** – Horizontal spacing.
- **y** – Vertical spacing.

Font (*Options := "", Font := ""*)

Changes the font for newly created controls.

Parameters

- **Options** – Options string.
- **Font** – Font name.

Focus ()

Focuses the GUI window.

Enable ()

Enables the GUI window if previously disabled.

Disable ()

Disables the GUI window if previously enabled.

SetIcon (*Icon*)

Changes the GUI window icon.

Parameters **Icon** – Path to an icon file.

AddText (*Options := "", Text := ""*)

Adds a text control.

Parameters

- **Options** – Options string.
- **Text** – Text contents of the control.

Returns *GuiBase.TextControl* instance.

AddButton (*Options := "", Text := ""*)

Adds a text control.

Parameters

- **Options** – Options string.
- **Text** – Text contents of the control.

Returns *GuiBase.ButtonControl* instance.

AddEdit (*Options := "", Text := ""*)

Adds an edit control.

Parameters

- **Options** – Options string.
- **Text** – Text contents of the control.

Returns *GuiBase.EditControl* instance.

AddListView (*Options := "", Headers := ""*)

Adds a ListView control.

Parameters

- **Options** – Options string.
- **Headers** – Either an array of header names, or a string of header names separated by | (pipe).

Returns *GuiBase.ListViewControl* instance.

AddStatusBar (*Options* := "", *Text* := "")

Adds a statusbar.

Parameters

- **Options** – Options string.
- **Text** – Text contents of the control.

Returns *GuiBase.StatusBarControl* instance.

3.2 Base types

3.2.1 ControlType

class *GuiBase.ControlType*

Represents a GUI control object.

Gui

Reference to the GUI instance that created this instance.

hwnd

Handle of the control.

Position

GuiBase.ControlPosition instance.

__New (*Gui*, *Options* := "", *Text* := "")

Creates a new control instance.

Note: Don't construct control instances directly, use the methods in *GuiBase*.

Parameters

- **Gui** – The GUI instance that created this control.
- **Options** – Options string.
- **Text** – Initial text contents of the control, if applicable.

Returns An indirect reference to the control instance.

Options (*Options*)

Change the options/settings of the control.

Parameters **Options** – Options string.

Control (*Command* := "", *Options* := "")

Calls the *GuiControl* command.

Parameters

- **Command** – The action to do. See documentation link above.
- **Options** – *Param3* in the documentation link above.

OnEvent (*Func* := "")

Makes the control call *Func* when an event happens.

Parameters **Func** – Function reference or boundfunc to call when events happen. :return: The control instance.

3.2.2 ContentControlType

class `GuiBase.ContentControlType`

This class extends `GuiBase.ControlType`

Represents a control with a singular content field, such as `GuiBase.TextControl`, `GuiBase.ButtonControl` and `GuiBase.EditControl`

Text

Get or set the contents of the control.

GetText ()

Returns The text contents of the control.

SetText (*Text*)

Parameters **Text** – New contents of the control.

3.2.3 PositionType

class `GuiBase.PositionType`

This class handles setting and getting positions of controls and windows. It has four properties (X, Y, W, H) which can be get and set.

Note: This class has a custom enumerator that will loop through the properties.

3.3 Control types

3.3.1 TextControl

class `GuiBase.TextControl`

This class extends `GuiBase.ContentControlType`

Type

The type of control, contains "Text "

3.3.2 ButtonControl

class `GuiBase.ButtonControl`

This class extends `GuiBase.ContentControlType`

Type

The type of control, contains "Button"

3.3.3 EditControl

class `GuiBase.EditControl`

This class extends *GuiBase.ContentControlType*

Type

The type of control, contains "Edit"

3.3.4 ListViewControl

class `GuiBase.ListViewControl`

This class extends *GuiBase.ControlType*

Type

The type of control, contains "ListView"

RowCount

Contains the amount of rows.

ColumnCount

Contains the amount of columns.

SelectedCount

Contains the amount of selected rows.

ImageList

Contains an `GuiBase.ImageList` instance if one is assigned to the listview via `SetImageList`.

Add (*Options* := "", *Fields**)

Adds a row to the listview.

Parameters

- **Options** – Options string.
- **Fields*** – Variadic parameter array of field contents.

Insert (*Row*, *Options* := "", *Fields**)

Identical to `Add()` but with an additional parameter `Row`

Parameters **Row** – Which row to insert the new row at.

Delete (*Row* := "")

Deletes one or all rows.

Parameters **Row** – If blank all rows are deleted, otherwise the row number specified.

Returns Selected row count of the listview.

GetSelected ()

Gets all the rows that are selected.

Returns An array of all the row numbers that are selected.

GetChecked ()

Gets all the rows that are checked.

Returns An array of all the row numbers that are checked.

SetImageList (*ImageList*)

Set an imagelist for the listview.

Parameters **ImageList** – An `GuiBase.ImageList` instance.

GetNextSelected (*Start := 0*)

Gets the next selected row after *Start*.

Returns Row number of the next selected row.

GetNextChecked (*Start := 0*)

Gets the next checked row after *Start*.

Returns Row number of the next checked row.

GetNextFocused (*Start := 0*)

Gets the next focused row after *Start*.

Returns Row number of the next focused row.

GetCount (*Option := ""*)

Calls `LV_GetCount()`. It is however recommended you use the methods above.

Parameters **Option** – What kind of rows to count.

Returns Amount of rows.

GetNext (*Start := 0, Option := ""*)

Calls `LV_GetNext()`. It is however recommended you use the methods above.

Parameters

- **Start** – Which row to start at when finding the next.
- **Option** – What kind of row to find.

Returns Row number of the next checked or focused row.

3.3.5 StatusBarControl

class `GuiBase.StatusBarControl`

This class extends `GuiBase.ControlType`

Type

The type of control, contains "StatusBar"

SetText (*NewText, PartNumber := 1, Style := ""*)

Calls `SB_SetText()`.

Parameters

- **NewText** – New text contents.
- **PartNumber** – Which part to change to *NewText*.
- **Style** – See the documentation link above.

SetParts (*Width**)

Calls `SB_SetParts()`.

Parameters **Width*** – New widths of the parts.

SetParts (*File, IconNumber := "", PartNumber := 1*)

Calls `SB_SetIcon()`.

Parameters

- **File** – Icon file.
- **IconNumber** – Icon number of the file.
- **PartNumber** – Which part to set the icon on.

3.4 Position types

3.4.1 WindowPosition

class `GuiBase.WindowPosition`

This class extends `GuiBase.PositionType`

You can use the properties to set and get position/size of a window.

3.4.2 ControlPosition

class `GuiBase.ControlPosition`

This class extends `GuiBase.PositionType`

You can use the properties to set and get position/size of a control.

3.5 Gui events

This section contains a list of all Gui events.

These methods are not supposed to be called. For a practical example of how these are used, see [Examples](#).

`GuiBase.Close()`

Called when the close button is clicked on the Gui.

`GuiBase.Escape()`

Called when the escape button is pressed while the Gui is focused.

`GuiBase.Size(EventInfo, Width, Height)`

Called when the Gui is resized.

Parameters

- **EventInfo** – 0 if resized, 1 if minimized and 2 if maximized.
- **Width** – The new width of the Gui window.
- **Height** – The new height of the Gui window.

`GuiBase.DropFiles(FileArray, Control, X, Y)`

Called when files are dropped onto the Gui.

Parameters

- **FileArray** – Array of files.
- **Control** – Which control the files were “dropped” onto.
- **X** – X coordinate of where the files were dropped.
- **Y** – Y coordinate.

`GuiBase.ContextMenu` (*Control, EventInfo, IsRightClick, X, Y*)

Called when the user right-clicks anywhere in the window except the title or menu bar.

Parameters

- **Control** – Control which was right-clicked, if any.
- **EventInfo** – See the [documentation](#).
- **Height** – The new height of the Gui window.

3.6 Control events

4.1 Class example

In this example we extend the `GuiBase` class and build upon it to make our own GUI.

CHAPTER 5

About

AHK Object Oriented GUIs is an attempt at making working with GUIs in AHK v1.1 more intuitive and clean. It aims at providing an object oriented way of working with GUIs.

5.1 Need help?

Talk to me on the official AutoHotkey Discord server!

Invite: <https://discord.gg/g5MagEr>

CHAPTER 6

Indices and tables

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

B

ButtonControl (class in GuiBase), 11
 ButtonControl.Type (in module GuiBase), 11

C

ContentControlType (class in GuiBase), 11
 ContentControlType.GetText() (in module GuiBase), 11
 ContentControlType.SetText() (in module GuiBase), 11
 ContentControlType.Text (in module GuiBase), 11
 ControlPosition (class in GuiBase), 14
 ControlType (class in GuiBase), 10
 ControlType.__New() (in module GuiBase), 10
 ControlType.Control() (in module GuiBase), 10
 ControlType.Gui (in module GuiBase), 10
 ControlType.hwnd (in module GuiBase), 10
 ControlType.OnEvent() (in module GuiBase), 11
 ControlType.Options() (in module GuiBase), 10
 ControlType.Position (in module GuiBase), 10

E

EditControl (class in GuiBase), 12
 EditControl.Type (in module GuiBase), 12

G

GuiBase (built-in class), 7
 GuiBase.__New() (built-in function), 8
 GuiBase.__Version (built-in variable), 7
 GuiBase.AddButton() (built-in function), 9
 GuiBase.AddEdit() (built-in function), 9
 GuiBase.AddListView() (built-in function), 9
 GuiBase.AddStatusBar() (built-in function), 10
 GuiBase.AddText() (built-in function), 9
 GuiBase.ahk_id (built-in variable), 7
 GuiBase.BackgroundColor (built-in variable), 7
 GuiBase.Close() (in module GuiBase), 14
 GuiBase.ContextMenu() (in module GuiBase), 14
 GuiBase.Control() (built-in function), 8
 GuiBase.ControlColor (built-in variable), 7
 GuiBase.Controls (built-in variable), 7

GuiBase.Destroy() (built-in function), 8
 GuiBase.Disable() (built-in function), 9
 GuiBase.DropFiles() (in module GuiBase), 14
 GuiBase.Enable() (built-in function), 9
 GuiBase.Escape() (in module GuiBase), 14
 GuiBase.Focus() (built-in function), 9
 GuiBase.Font() (built-in function), 9
 GuiBase.GetControl() (built-in function), 8
 GuiBase.GetGui() (built-in function), 7
 GuiBase.Hide() (built-in function), 8
 GuiBase.hwnd (built-in variable), 7
 GuiBase.Margins() (built-in function), 8
 GuiBase.Options() (built-in function), 8
 GuiBase.Position (built-in variable), 7
 GuiBase.SetDefault() (built-in function), 8
 GuiBase.SetDefaultListView() (built-in function), 8
 GuiBase.SetIcon() (built-in function), 9
 GuiBase.Show() (built-in function), 8
 GuiBase.Size() (in module GuiBase), 14
 GuiBase.Title (built-in variable), 7
 GuiBase.Visible (built-in variable), 7

I

IndirectReferenceDelete() (built-in function), 5
 IndirectReferenceHolder (built-in class), 6
 IsInstance() (built-in function), 5

L

ListViewControl (class in GuiBase), 12
 ListViewControl.Add() (in module GuiBase), 12
 ListViewControl.ColumnCount (in module GuiBase), 12
 ListViewControl.Delete() (in module GuiBase), 12
 ListViewControl.GetChecked() (in module GuiBase), 12
 ListViewControl.GetCount() (in module GuiBase), 13
 ListViewControl.GetNext() (in module GuiBase), 13
 ListViewControl.GetNextChecked() (in module
 GuiBase), 13
 ListViewControl.GetNextFocused() (in module
 GuiBase), 13

ListViewControl.GetNextSelected() (in module
GuiBase), [13](#)
ListViewControl.GetSelected() (in module GuiBase), [12](#)
ListViewControl.ImageList (in module GuiBase), [12](#)
ListViewControl.Insert() (in module GuiBase), [12](#)
ListViewControl.RowCount (in module GuiBase), [12](#)
ListViewControl.SelectedCount (in module GuiBase), [12](#)
ListViewControl.SetImageList() (in module GuiBase), [12](#)
ListViewControl.Type (in module GuiBase), [12](#)

P

PositionType (class in GuiBase), [11](#)

S

StatusBarControl (class in GuiBase), [13](#)
StatusBarControl.SetParts() (in module GuiBase), [13](#)
StatusBarControl.SetText() (in module GuiBase), [13](#)
StatusBarControl.Type (in module GuiBase), [13](#)

T

TextControl (class in GuiBase), [11](#)
TextControl.Type (in module GuiBase), [11](#)
Type() (built-in function), [5](#)

W

WindowPosition (class in GuiBase), [14](#)