

## Control Attributes

Product:	Guiliani Streaming Editor (GSE)
Release version:	2.3
Release date:	August 12, 2019

### Table of contents

Introduction:.....	6
Common Information:.....	6
Behaviour: .....	7
1. GUIBehaviour: .....	7
2. AutoRepeatBehaviour .....	8
3. GUICompositeBehaviour:.....	9
4. ExampleBehaviour.....	9
5. GUIHotkeyBehaviour:.....	10
6. GUIKeyboardBehaviour:.....	11
7. GUIMultiCmdBehaviour: .....	11
8. GUIObjectStateBehaviour: .....	12
9. GUIObjectVisualsBehaviour .....	13
10. GUISingleCmdBehaviour: .....	14

11.	GUITabSwitchBehaviour:.....	14
	Layouter:.....	15
12.	GUILayouter: .....	15
13.	GUILayouterAlignToParent:.....	15
14.	GUILayouterAnchor:.....	16
15.	Example Layouter .....	17
16.	GUILayouterGrid:.....	17
17.	GUILayouterList: .....	18
18.	GUILayouterPercentage: .....	18
19.	GUILayouterReposition: .....	19
	Command: .....	20
20.	GUICommand: .....	20
21.	GUICallAPICmd: .....	20
22.	ExampleCommand .....	21
23.	GUIDelayCmd: .....	21
24.	GUITransitionCmd: .....	22
25.	GUILoadDialogCommand .....	23
26.	GUIPlaybackSoundCommand:.....	23
27.	GUIQuitCmd: .....	24
28.	GUISetObjectStateCmd: .....	24
29.	GUISetObjectVisualsCmd .....	25
30.	GUIStartAnimationChainCmd:.....	26
31.	GUIStopAnimationChainCmd: .....	26

32.	GUISwitchResourceSetsCmd: .....	27
Common Attributes: .....		28
33.	GUIObject: .....	28
34.	NinePatchBorders: .....	29
35.	StandardText: (applies for all texts) .....	30
Controls: .....		31
36.	GUIGeometryObject: .....	31
37.	GUIImage: .....	32
38.	GUIAnimatedImage: .....	33
39.	GUIImageStack: .....	34
40.	GUIEditableText: .....	35
41.	GUIEdit: .....	35
42.	GUIRichText: .....	36
43.	GUIScrollingText: .....	37
44.	GUIBaseTextField: .....	37
45.	GUITextField: .....	38
46.	GUIInputField: .....	38
47.	GUIComboBox: .....	39
48.	GUIButton: .....	40
49.	GUIIconButton: .....	41
50.	GUIBlendButton: .....	41
51.	GUIBaseCheckBox: .....	42
52.	GUICheckBox: .....	43

---

53.	GUIRadioButton: .....	44
54.	GUIScrollBar: .....	44
55.	GUIAbstractBar:.....	45
56.	GUIBaseSlider:.....	46
57.	UISlider:.....	46
58.	GUIProgressBar: .....	47
59.	CGUICircularSlider: .....	49
60.	GUIKnob: .....	50
61.	GUISegmentBar: .....	51
62.	GUIRangeSlider:.....	52
63.	GUIRepositionCompositeObject: .....	53
64.	GUILayerContainer .....	53
65.	GUICenterFocusContainer (O):.....	54
66.	GUIScrollView:.....	54
67.	GUITouchScrollView: .....	55
68.	GUICarousel:.....	56
69.	GUICalendar: .....	58
70.	GUIChart:.....	60
71.	GUIClock: .....	62
72.	GUIGauge: .....	63
73.	GUIWheel: .....	64
74.	GUIKeyboard .....	66
75.	GUIVideo .....	67



**Introduction:**

This document describes all attributes which can be set inside the GSE for all existing objects.

**Common Information:**

All IDs starting with "**DUMMY\_**" mark that no resource is currently set for this attribute, e.g.

**DUMMY\_IMAGE** means that no ImageID is currently and so no image will be displayed.

## Behaviour:

### 1. GUIBehaviour:

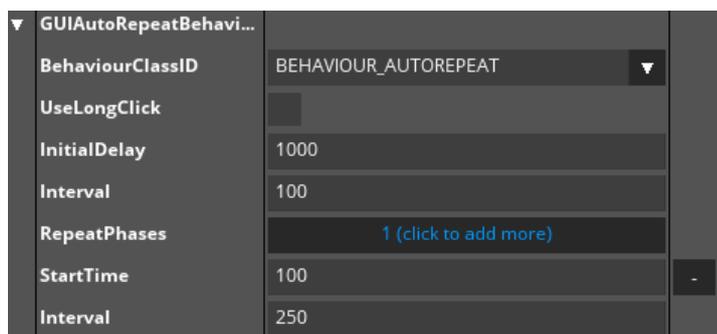


Behaviours are used for adding functionality to Guiliani's event slots. Each widget has numerous event slots that are called by the framework when specific events occur, like key presses, mouse clicks, mouse drags and so on.

Behaviours are used to control the objects behaviour in response to events sent to the object. An object has a behaviour attached to it and the event is first received by the behaviour. The object itself only receives the event if the behaviour has not handled it (i.e it returns false).

**Note:** The action of the behaviour is executed immediately and might block the running application, including internal processing and refreshing the GUI.

## 2. AutoRepeatBehaviour



This behaviour can trigger multiple Click-Events, which are sent after a certain amount of time and repeated in a specified interval. The sequence can either be triggered by a long click or a given amount of time after the ButtonDown-event was sent. The ButtonUp-event ends the sequence. The behaviour can have as many RepeatPhases. New phases are added by clicking on the Button “(click to add more)” and removed using the “-“ -button on the right side of the respective phase. Each RepeatPhase consists of a StartTime which is relative to the start of the last phase and an Interval which sets the amount of time between the trigger of Click-Events inside this phase. You can attach this behaviour to a button which deletes single characters inside an inputfield on clicking and have it executed first with an interval of 500ms and then with 200ms to erase the characters faster.

**UseLongClick:** If this is set, the Behaviour is triggered by the LongClick-event, otherwise the time in InitialDelay is waited after the ButtonDown-event.

**InitialDelay:** Amount of milliseconds, which is waited after ButtonDown-event if LongClick is not used

**Interval:** Amount of milliseconds, which specifies the time between each trigger in the first phase

**RepeatPhases:** Click on this button, if you want more RepeatPhases with varying intervals

**StartTime:** Amount of milliseconds after the start of the last phase which are waited until this phase is started

**Interval:** Amount of milliseconds, which specifies the time between each trigger in this phase

### 3. GUICompositeBehaviour:

▼ GUICompositeBehaviour	
BehaviourClassID	BEHAVIOUR_COMPOSITE ▼
NumberOfBehaviours	(click to add more) ▼
▼ GUISingleCmdBehaviour	
BehaviourClassID	BEHAVIOUR_SINGLE_CMD ▼
BehaviourType	BT_DUMMY ▼
▼ GUICommand	
CommandClassID	DUMMY_COMMAND ▼
▼ GUISingleCmdBehaviour	
BehaviourClassID	BEHAVIOUR_SINGLE_CMD ▼
BehaviourType	BT_DUMMY ▼
▼ GUICommand	
CommandClassID	DUMMY_COMMAND ▼

This Behaviour contains a number of other behaviours and forwards incoming events to them, until one of the behaviours has signalled that it has processed the event (i.e. it returns true).

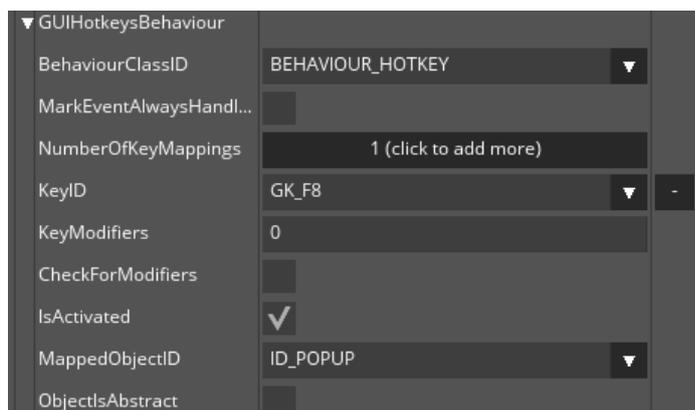
### 4. ExampleBehaviour

▼ ExampleBehaviour	
BehaviourClassID	BEHAVIOUR_CUSTOM ▼
DragAction	DA_MOVE ▼

This behaviour is used to demonstrate how to extend Guiliani with custom-functionality. See the source-code for more information.

**DragAction:** The action which is triggered by DoDrag on the attached object. DA\_MOVE moves the object to the touch-position and DA\_SIZE equally sizes the object retaining its center

## 5. GUIHotkeyBehaviour:



This behaviour can only be attached to a CompositeObject, otherwise processing will not be done.

**MarkEventAlwaysHandled:** if this is set, the behaviour always consumes the event (return true), even if the hotkey was not processed.

**NumberOfKeyMappings:** Adds a new key-mapping to the list of hot keys. Every time, you click on the value field, you get a new input in the list of hot keys. If you want to delete some keys from this list, you can use “minus” Button on the right side of “KeyID” field.

**KeyID:** Key ID which triggers the action

**KeyModifiers:** modifiers which should be checked. Value for the modifiers are: 0 for NONE, 1 for SHIFT, 2 for CTRL, 4 for ALT and 8 for META (special implementation). Can be summed up, if multiple modifiers are checked, e.g. 5 for SHIFT and ALT

**CheckForModifiers:** if set, the modifiers are checked

**IsActivated:** if set, the keymapping will be processed

**MappedObjectID:** ObjectID of the object which will receive the resulting event. Only the container this behaviour is attached to will be searched for the given ObjectID.

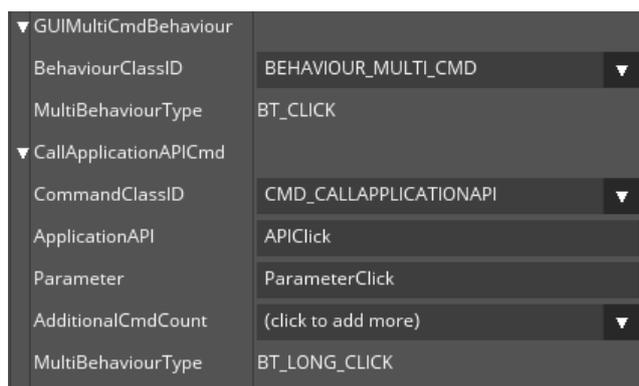
**ObjectsAbstract:** If the object is abstract object, by activating the hot key, the DoClick method will be executed. Otherwise the object will be firstly checked whether it is clickable.

## 6. GUIKeyboardBehaviour:



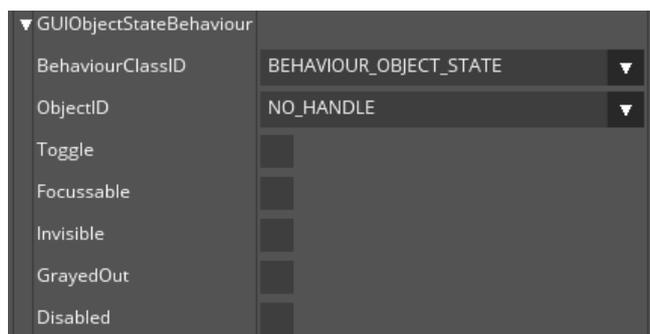
This behaviour is to be used with an OnScreen-Keyboard / Virtual Keyboard. It sends an UTF16 keycode to the Keyboard in reaction to a ButtonUp-Event. In the field KeyCode you can set the keycode.

## 7. GUIMultiCmdBehaviour:



This behaviour maps each available event to a specific action. Gives the possibility to execute a separate command for each event (e.g. click event, drag event, etc.). The field "MultiBehaviourType" shows the corresponding event-type for the mapping.

## 8. GUIObjectStateBehaviour:



This Behaviour is used for triggering changes of the states of the specified object

**ObjectID:** ID of the object to process

**Toggle:** if set the used attributes will be toggled each time this behaviour is executed. If not set, the attributes are set as they specified.

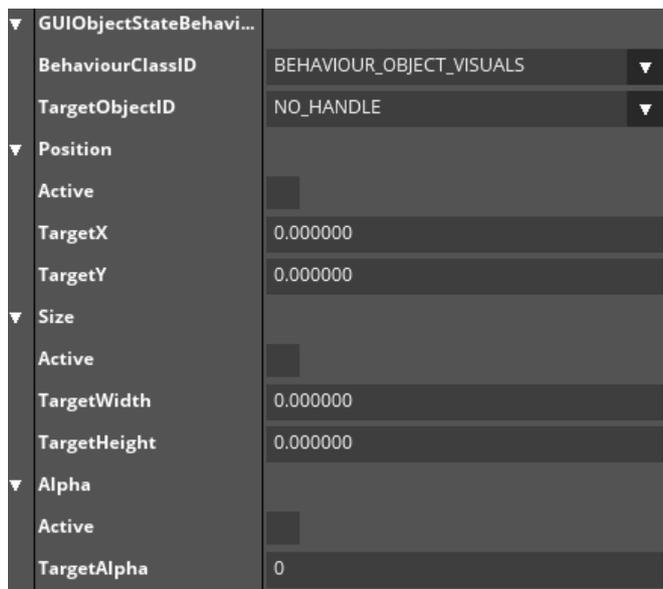
**Focussable:** processes the “Focussable”-attribute

**Invisible:** processes the “Invisible”-attribute

**GrayedOut:** processes the “GrayedOut”-attribute

**Disabled:** processes the “Disabled”-attribute

## 9. GUIObjectVisualsBehaviour



This behaviour can be used to modify the visual representation of an object. Which visual, i.e. position, size or alpha is modified can be specified separately.

**TargetObjectID:** This is the ID of the object which should be modified by this behaviour

**Position/Size/Alpha:** Modify the position and/or size of the object

**Active:** If this is set, the position, size and/or alpha-value of the object is set to the specified values

**TargetX/TargetY/TargetWidth/TargetHeight/TargetAlpha:** the according attribute of the object is set to the target-value if active

## 10. GUISingleCmdBehaviour:

▼ GUISingleCmdBehaviour	
BehaviourClassID	BEHAVIOUR_SINGLE_CMD ▼
BehaviourType	BT_KEY_UP ▼
▼ GUICommand	
CommandClassID	DUMMY_COMMAND ▼

Execute a command in response to a specific event. Gives the possibility to execute a command when a specific event (e.g. click event, drag event, etc.) occurs.

**BehaviourType:** specifies the type of event which should be linked to the command.

## 11. GUIDTabSwitchBehaviour:

▼ GUIDTabSwitchBehaviour	
BehaviourClassID	BEHAVIOUR_TABSWITCH ▼
Container	new object-ID... ▼
TargetObject	▼

This behaviour can be used to switch between different child-objects of a container and create a tab-like style

**Container:** This is the ObjectID of the container with several child-objects each representing a tab

**TargetObject:** This is the ObjectID of the child-object which will be displayed when this behaviour is triggered. All other child-objects will be set to invisible.

**Layouter:**

## 12. GUILayouter:



Layouters are used to arrange objects to a specific rule, if position or size of the attached objects changes. Layouters are used to automatically influence the position and/or size of child objects within a composite object.

**LayouterClassID:** This field gives list of different layouter classes. The default is “DUMMY\_LAYOUTER” which means no specific layouter class is applied to the object

## 13. GUILayouterAlignToParent:



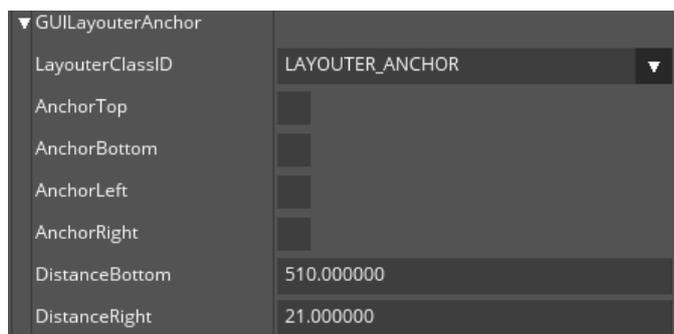
This Layouter aligns the attached object relative to its parent object.

**HorizontalObjectAlignment:** Horizontal alignment (left, center, right)

**VerticalObjectAlignment:** Vertical alignment (top, center, bottom)

Note: This will not directly refresh the layout until the attached object is resized.

## 14. GUILayouterAnchor:



This Layouter can be used to 'fix' the edges of a widget to its parent.

The following table gives you some examples which behaviour will result from which anchor-settings:

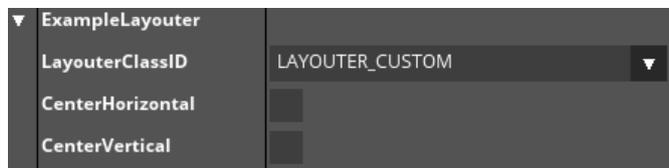
**AnchorTop/AnchorBottom/AnchorLeft/AnchorRight:** shows /defines Anchor attributes. The following table gives you some examples stating which behaviour will result from which anchor-settings:

top	bottom	left	right	result
ON	ON	ON	ON	Object gets stretched in X/Y directions
off	off	off	off	Object will not get stretched, but will scale its position along with the change in size of its parent. i.e. If the width of the parent object gets doubled, the relative X position of the child object will double as well.
ON	ON	ON	off	Object gets stretched in Y direction and is locked to the left border of its parent
ON	ON	off	ON	Object gets stretched in Y direction and is locked to the right border of its parent
ON	ON	off	off	Object gets stretched in Y direction and will scale its X position along with the change in size of its parent

**DistanceBottom/DistanceRight:** The distances of the associated object's right border to the parent's right border and of the associated object's bottom border to the parent's bottom border. This is useful

for example when the height or width of the associated object changes so that when resizing the parent later on, the current anchor length are used and not those that were saved when the anchors were chosen.

## 15. Example Layouter

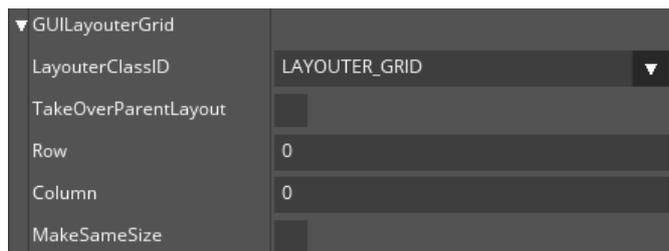


This layouter is used to demonstrate how to extend Guiliani with custom-functionality. See the source-code for more information.

**CenterHorizontal:** If this is set, the attached object will be centered horizontally inside its parent

**CenterVertical:** If this is set, the attached object will centered vertically inside its parent

## 16. GUILayouterGrid:



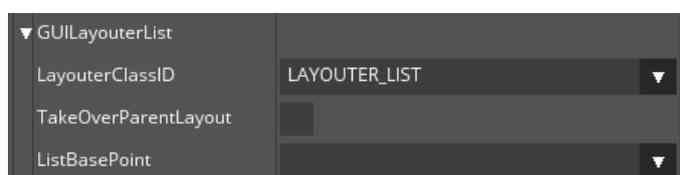
**Note:** This Layouter should only be attached to a CompositeObject, otherwise a warning-message will appear.

**TakeOverParentLayout:** If this is set, the attached object is moved to the relative position of 0,0 and gets its size set to the size of the parent-object before the actual layouting takes place.

**Row/Column:** number of rows/columns which define the size of the cells

**MakeSameSize:** If this is set, the sizes of each child-object are set to the size of one cell of the specified grid.

## 17. GUILayouterList:



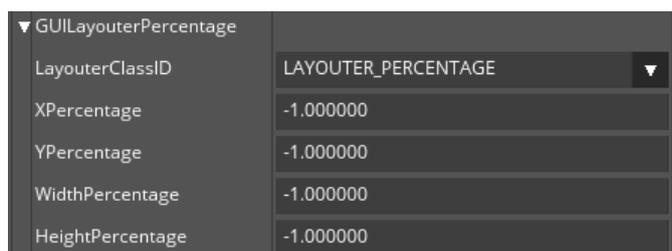
**Note:** This Layouter should only be attached to a CompositeObject, otherwise a warning-message will appear.

This Layouter aligns all child-objects in a list based on the ListBasePoint

**TakeOverParentLayout:** If this is set, the attached object is moved to the relative position 0,0 and set to the size of the parent-object, before the actual layouting takes place.

**ListBasePoint:** The base-point on which the list positioning is based (horizontal or vertical)

## 18. GUILayouterPercentage:

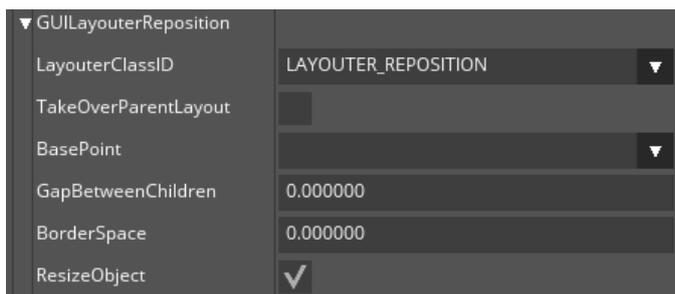


The Value -1.0 is used if the attribute should not be processed.

**XPercentage/YPercentage:** Set object to a new relative position. Values are multiplied with the parent's width/height.

**WidthPercentage/HeightPercentage:** Set object to a new size. Values are multiplied with the parent's width/height.

## 19. GUILayouterReposition:



Should only be attached to a CompositeObject, otherwise a warning-message will appear.

**TakeOverParentLayout:** If this is set, the attached object is moved to the relative position of 0,0 and get its size set to the size of the parent-object before the actual layouting takes place.

**BasePoint:** The rule on which the repositioning is based (horizontal/vertical).

**GapBetweenChildren:** The gap (in pixels) between the children used during layouting.

**BorderSpace:** Border-space used during layouting.

**ResizeObject:** If this is set, the attached object is automatically resized to be exactly large enough to contain all of its children.

**Command:**

## 20. GUICommand:

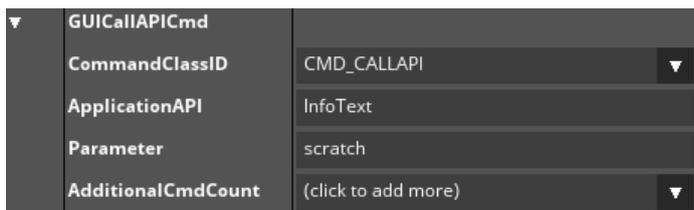


Commands are used to execute specific actions asynchronously and encapsulate specific actions covering calls to functions of the underlying application log. They can be reused and attached to the objects within the GUI. Commands when executed are added to a queue and are not executed before the next main-loop.

**CommandClassID:** This field gives list of different command classes. The default is “DUMMY\_COMMAND” which means no specific command class is applied to the object.

**AdditionalCmdCount:** Here additional commands can be selected, which will be executed after the current command is finished. This field is expanded whenever a command class is selected from the drop down list.

## 21. GUICallAPICmd:



This Command can be used to execute various actions specified by the given attributes. The handling of this command happens in the method DoCallAPI of the CGUI-class. The attributes for setting the API name and parameter is done in GSE.

**ApplicationAPI:** String which is sent to the command and can be used to execute specific actions.

**Parameter:** String which is sent to the command which specifies the used parameter

## 22. ExampleCommand

▼	ExampleCommand	
	CommandClassID	CMD_CUSTOM ▼
	TargetObjectID	NO_HANDLE ▼
	StepSize	0.000000
	AdditionalCmdCount	(click to add more) ▼

This command is used to demonstrate how to extend Guiliani with custom-functionality. See the source-code for more information.

**TargetObjectID:** This is the ID of the object which should be resized by this command

**StepSize:** This settings is used to increase/decrease the size of the target-object

## 23. GUIDelayCmd:

▼	GUIDelayCmd	
	CommandClassID	CMD_DELAY ▼
	Duration	0
	AdditionalCmdCount	(click to add more) ▼

This command is used to trigger another attached command after a specified amount of time

**Duration:** Amount of milliseconds after which the next command should be executed

## 24. GUITransitionCmd:

GUITransitionCmd	
CommandClassID	CMD_DIALOG_TRANSITION ▼
DestDialogFileName	▼
SourceObjectID	NO_HANDLE ▼
TransitionType	BLEND_ONLY ▼
EasingType	EASE_LINEAR ▼
TransitionTime	1000
AdditionalCmdCount	(click to add more) ▼

This Command executes a transition from one dialog to another. Various options can be set. This command first loads the destination-dialog and then traverses from the source-object to it using a specific type of transition.

**DestDialogFileName:** The filename of the dialog which will be the destination-object.

**SourceObjectID:** ID of the source-object. This needs to be an object within the currently active dialog or the dialog itself to have the command work.

**TransitionType:** type of transition. This can be one of the following options:

- 1) BLEND\_ONLY: Blends to the destination dialog
- 2) BLND\_FADE: Blends to the destination dialog with the fading effect.
- 3) BLEND\_SHRINK: The transition blends to the destination dialog with shrinking effect.
- 4) BLEND\_ZOOM: The transition blends to the destination dialog with shrinking effect.
- 5) PUSH\_FROM\_LEFT/PUSH\_FROM\_RIGHT/PUSH\_FROM\_TOP/PUSH\_FROM\_BOTTOM: moves the source object out of the screen while pushing the destination-dialog into the screen from either left/right/top/bottom sides.
- 6) ROTATE\_X\_AXIS\_CW/ROTATE\_X\_AXIS\_CCW/ROTATE\_Y\_AXIS\_CW/ROTATE\_Y\_AXIS\_CCW: Flips the screen as a 3D-panel, having the source-dialog on one side and the destination-dialog on the other side around the specified axis either clockwise or counter-clockwise.
- 7) DISSOLVES: The source dialog dissolves in transition and destination dialog appears.

**EasingType:** The easing type which is used for the transition

**TransitionTime:** Duration in milliseconds for the transition.

## 25. GUILoadDialogCommand

▼	GUILoadDialogComma...	
CommandClassID	CMD_LOAD_DIALOG	▼
DialogFileName	Dialog_2	▼
ParentObjectID	AID_CLOCK_3	▼
DeleteObjectID	NO_HANDLE	▼
AdditionalCmdCount	(click to add more)	▼

This command loads a new dialog and places it into the specified CompositeObject. Additionally a given object can be deleted.

**DialogFileName:** Filename of the dialog which should be loaded. This can either be an xml-file or a binary file.

**ParentObjectID:** ID of the future parent object of the loaded dialog. If this attribute is "NO\_HANDLE" the top-most object (i.e. the GUI itself) will be the new parent of the loaded dialog.

**DeleteObjectID:** ID of an object which should be deleted after the dialog was loaded.

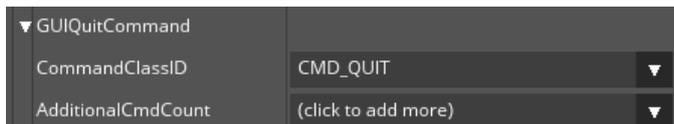
## 26. GUIPlaybackSoundCommand:

▼	GUIPlaybackSoundCommand	
CommandClassID	CMD_PLAYBACK_SOUND	▼
SoundID	DUMMY_SOUND	▼
AdditionalCmdCount	(click to add more)	▼

This command is used to playback the associated sound.

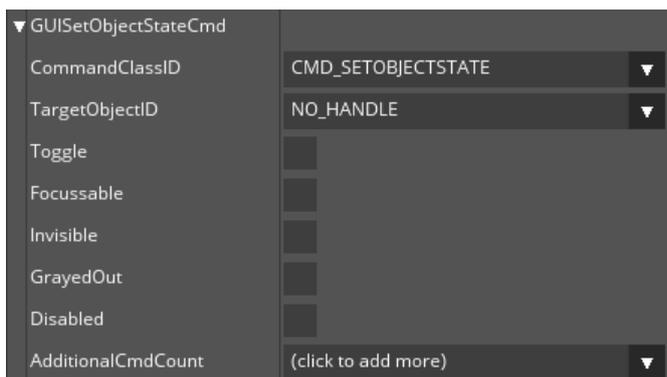
**SoundID:** ID of the sound which should be played.

## 27. GUIQuitCmd:



This command cleanly ends the current application by shutting down the main-loop of Guiliani and destroys all created Wrapper-classes (e.g. Graphics-, Sound or Font-Wrapper).

## 28. GUISetObjectStateCmd:



This command does the same as the GUISetObjectStateBehaviour, but will be executed asynchronously and not before the next main-loop. The attributes are similar to the GUISetObjectStateBehaviour.

## 29. GUISetObjectVisualsCmd

▼	GUISetObjectVisualsC...	
	CommandClassID	CMD_SETOBJECTVISUALS ▼
	TargetObjectID	AID_CLOCK_3 ▼
▼	Position	
	Active	<input checked="" type="checkbox"/>
	TargetX	50.000000
	TargetY	50.000000
	Animate	<input checked="" type="checkbox"/>
	Duration	1000
	EasingType	EASE_OUT_SINE ▼
▼	Size	
	Active	<input checked="" type="checkbox"/>
	TargetWidth	50.000000
	TargetHeight	50.000000
	Animate	<input checked="" type="checkbox"/>
	Duration	1000
	EasingType	EASE_OUT_CUBIC ▼
▼	Alpha	
	Active	<input type="checkbox"/>
	TargetAlpha	0
	Animate	<input type="checkbox"/>
	Duration	0
	EasingType	EASE_NONE ▼
	AdditionalCmdCount	(click to add more) ▼

This command does the same as the GUISetObjectVisualsBehaviour, but will be executed asynchronously and not before the next main-loop. The attributes are similar to the GUISetObjectStateBehaviour, but additionally easing and a duration can be specified for each attribute to create a smooth change of the attribute.

**Animate:** This option enables the object to have animation effect when the object state is transitioned.

**Duration:** Duration of transition.

**EasingType:** The easing type which is used for the transition.

## 30. GUIStartAnimationChainCmd:

▼	GUIStartAnimationCha...	
CommandClassID	CMD_START_ANIMATIONCHA	▼
AnimationChainID	ANI_1	▼
AdditionalCmdCount	(click to add more)	▼

This command is used to start an animation-chain which is contained in the project

**AnimationChainID:** This attribute gives list of available animation chains present in the project.

## 31. GUIStopAnimationChainCmd:

GUIStartAnimationCha...		
CommandClassID	CMD_STOP_ANIMATIONCHAIN	▼
AnimationChainID	DUMMY_ANIMATION	▼
StopAllChains	<input type="checkbox"/>	
AdditionalCmdCount	(click to add more)	▼

This command can be used to stop running animation-chains.

**AnimationChainID:** ID of the Animation-chain which should be stopped.

**StopAllChains:** If this is active all currently running animation-chains are stopped.

### 32. GUISwitchResourceSetsCmd:

GUISwitchResourceSet...	
CommandClassID	CMD_SWITCH_RESOURCESETS ▼
ResourceType	Images ▼
ResourceSetName	Dark ▼
AdditionalCmdCount	(click to add more) ▼

This command can be used to switch to different resource-sets and thus changing the design of the application during runtime.

**ResourceType:** specified the type of resource to switch. This can be Fonts/General Resources/Images/Languages/Properties or Sounds

**ResourceSetName:** name of the resource-set which will be activated

## Common Attributes:

### 33. GUIObject:

Attribute	Value
▼ GUIObject	
XPos	0.000000
YPos	0.000000
Width	100.000000
Height	100.000000
ObjectID	AID_GEOMETRYOBJECT_7 ▼
Focussable	<input type="checkbox"/>
Invisible	<input type="checkbox"/>
GrayedOut	<input type="checkbox"/>
Disabled	<input type="checkbox"/>
ClickThrough	<input type="checkbox"/>
OverriddenNeighbors	<input type="checkbox"/>
Alpha	255

**XPos/YPos:** X/Y-position of the object relative to the left side of the parent object (in pixels). If the object is moved around, this value also changes. Floating-point values are possible.

**Width/Height:** Width/Height of the object. If the object is resized via mouse, this value also changes. Floating-point values are possible.

**ObjectID:** ObjectID for the object to access it from other parts (e.g. Commands, Behaviours, etc.)

When a new object is created it receives an auto-generated ObjectID. This can be changed by selecting another ObjectID from the list or defining a new one

**Focussable:** If an object is focusable it can request or lose the focus depending on the application's workflow. Focussed objects can be controlled via keyboard.

Invisible: visible state of object

**GrayedOut:** Grayed out objects cannot receive any events and are displayed in a different way.

Disabled: Disabled objects like grayed out ones do not receive any events, but do not have a special visualization.

**ClickThrough:** If an object is click-through events occurring inside its bounding rectangle will be passed to objects “below” this object (z-order).

**OverriddenNeighbors:** If this setting is active the neighbours of the object are set for 4 way navigation

**Alpha:** Alpha-value used to display the object. 0 means fully transparent, 255 is for fully opaque.

## 34. NinePatchBorders:



A Ninepatch is a smart way to enlarge bitmaps by defining 9 different regions of the original bitmap which will be scaled up independently.

**Top:** Number of pixels which define the top-stripe of the bitmap

**Bottom:** Number of pixel which define the bottom-stripe of the bitmap

**Left:** Number of pixels which define the left-stripe of the bitmap

**Right:** Number of pixels which define the right-stripe of the bitmap

**Note:** Ninepatches will only work on rectangular objects or objects with rounded edges, not on circular objects

### 35. StandardText: (applies for all texts)

StandardText		
TextTypeID	Standard Text	▼
TextColorStandard	PROP_TEXT_COLOR_HEADLINE	V P
TextColorHighlighted	PROP_TEXT_COLOR_HEADLINE	V P
TextColorGrayedOut	PROP_TEXT_COLOR_HEADLINE	V P
TextColorPressed	PROP_TEXT_COLOR_HEADLINE	V P
TextFontID	HEADER_FNT	
TextFontSpacing	0.000000	
LineSpacing	1.000000	
SingleLine	<input checked="" type="checkbox"/>	
VerticalAlignment	V_CENTERED	▼
HorizontalAlignment	H_CENTERED	▼
TextXPos	0.000000	
TextYPos	0.000000	
TextWidth	220.000000	
TextHeight	43.000000	
TextID	SCRATCHPAD_TXT	

**TextTypeID:** This changes the type of the text. This can be “Standard Text”, “Editable Text”, “Rich Text”, “Scrolling Text” or “No Text”. Additional attributes according to the TextTypeID will be added in the list after standard-attributes.

**TextColorStandard/TextColorHighlighted/TextColor/GrayedOut/TextColorPressed:** Colours used for the text according to the states of the object. All colours can be set via static value or property.

**TextFontID:** ID of the font which will be used to render the text. If “DUMMY\_FONT” is used, no text will be displayed.

**TextFontSpacing:** Spacing (in pixels) between neighbouring letters.

**LineSpacing:** Spacing between lines. Only applies if “SingleLine”-attribute is not set.

**SingleLine:** Indicates whether this text is single- or multi-line.

**VerticalAlignment/HorizontalAlignment:** Horizontal/Vertical alignment of the text. Can be left/centered/right or top-line/centered/bottom-line

**TextXPos/TextYPos:** Position of the textfield inside the object relative to the top/right-corner

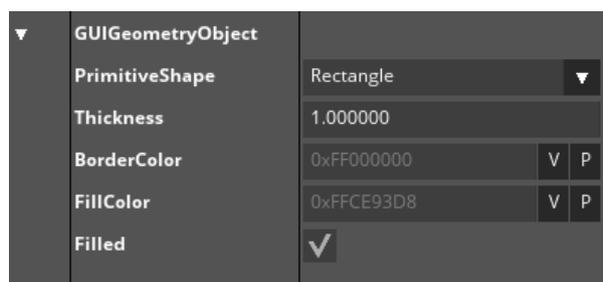
**TextWidth/TextHeight:** Width and height of the textfield inside the object

**TextID:** Choose the TextID which will be used for the text. If the currently active language-set is changed, this text will automatically be updated. If TextID is set to "text string", then a static text will be used.

**Text:** The static text which will be used instead of TextID

## Controls:

### 36. GUIGeometryObject:



Note: some combinations of PrimitiveShape and Filled may not work depending on the currently used Graphics-Wrapper in the application.

**FillColor:** color of the primitive which will be drawn. This can either be set to use a static value or a property

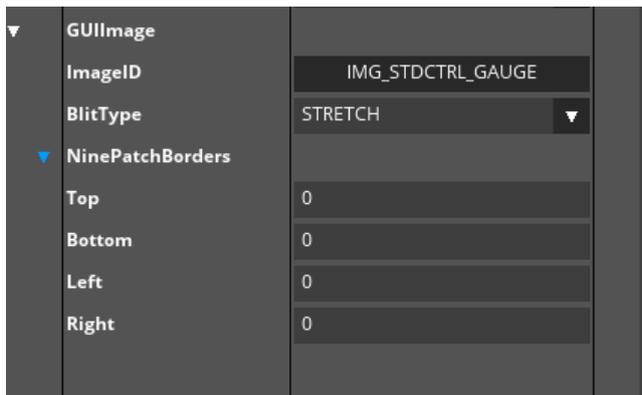
**PrimitiveShape:** One of the following options:

- Diagonal Line 1 /Diagonal Line 2: A diagonal line top/left to bottom/right or top/right to bottom/left
- Horizontal Line /Vertical Line: Vertical or horizontal line
- Ellipse/Rectangle: An ellipse or rectangle

**Thickness:** The width of the primitive

**Filled:** Used for ellipses and rectangles for filling

## 37. GUIImage:



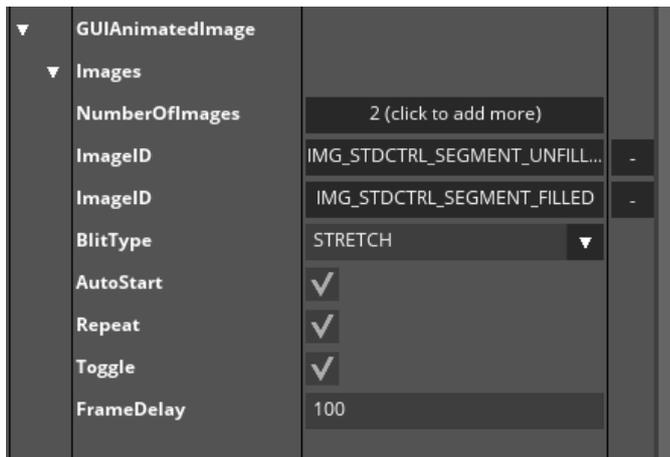
**ImageID:** Image used for blitting. It Can be set using “Image” dialog under Resources.

**Blit Type:** Blit Type: Three options are possible for this attribute.

- Stretch: Setting Blit type as “Stretch”, stretches the image to the entire size of the object.
- Centre: When this option is set the image is placed at the centre of the control without modifying its size.
- Tiled: With this option the image is tiled up to cover up the entire size of the object

**NinePatchBorders:** If this is set, stretching images smaller than the object’s size will be done using the given ninepatch-borders.

### 38. GUIAnimatedImage:



Note: at least two images should be used to see this control in action.

**NumberOfImages:** clicking on this button will add a new image

**ImageID:** image(s) which will be used for animation. Clicking on the “-“-button in the right column will remove the image next to it. The arrows pointing up and down can be used to re-order the images inside the list.

**Blit Type:** (see GUIImage)

**Repeat:** If this is set, the animation runs in an infinite loop.

**AutoStart:** If this is set, the animation automatically starts and can be viewed inside the editor.

**Toggle:** if this is set, the animation will go forward through all images and backwards to the first image

**FrameDelay:** Number of milliseconds between each transition.

### 39. GUILmageStack:

▼	GUILmageStack		
▼	Images		
	NumberOfImages	1 (click to add more)	
	ImageID	IMG_STDCTRL_SEGMENT_FILLED	-
	FrameDelay	500	
	MaxScale	1.000000	
	MinScale	0.150000	
	StaticScale	0.500000	

This control gives the possibility to swipe through a set of images which will be animated.

**NumberOfImages:** Clicking on this button will add a new image

**ImageID:** Image(s) which will be used for animation. Clicking on the “-“-button in the right column will remove the image next to it. The arrows pointing up and down can be used to re-order the images inside the list.

**FrameDelay:** Number of milliseconds between each transition.

**MaxScale/MinScale:** Maximum/minimum scaling factor for the images during the animation

**StaticScale:** Scaling factor for the image when not animated.

## 40. GUIEditableText:

▼ GUIEditableText	
CursorWidth	1.000000
SelectionColor	0XFFDFDFDF
PasswordMode	<input type="checkbox"/>
PasswordCharacter	*

**CursorWidth:** Width in pixels of the vertical cursor line

**SelectionColor:** Background-colour which is used to display selected text. Foreground-colour is the same as normal text

**PasswordMode:** If this is set the text entered will be displayed using the PasswordCharacter, not the actual entered character

**PasswordCharacter:** The character which is used to display the text when PasswordMode is active. If more than one character is entered, only the first will be used

## 41. GUIEdit:

▼ GUIEdit	
AcceptedCharSet	AC_ALL ▼
MaxLength	2147483647
ResetCursorWhenFocu...	<input checked="" type="checkbox"/>

**AcceptedCharSet:** Shows/defines accepted character set

**MaxLength:** Shows/defines the maximal number of characters

**ResetCursorWhenFocusslost:** Shows/defines cursor behaviour, when focus lost. If active: Cursor will be not seen after focus lost. In another case: you will still see the cursor even after the focus lost.

## 42. GUIRichText:

▼ GUIRichText	
BoldFontID	FNT_BEBASNEUE_BOLD
ItalicFontID	FNT_BEBASNEUE_ITALIC
Bold-ItalicFontID	FNT_BEBASNEUE_BOLD_ITALIC
TagBold	b
TagItalic	i
TagUnderlined	u
TagColor	color
TagNoBreak	nobr
TagLineBreak	br
TagOrderedList	ol
TagUnorderedList	ul
TagListItem	li
TagOpening	<
TagClosing	>
TagFinished	/

NOTE: The attributes for rich text appears when the Rich Text is selected under TextTypeID field

**BoldFontID/ItalicFontID/Bold-ItalicFontID:** The font-ids which should be used for displaying bold, italic and bold+italic text-paragraphs

**TagBold/TagItalic/TagUnderlined:** Tags which are used to identify formatting attributes within the text

**TagColor:** Tag which is used to specify the color of the following paragraph

**TagNoBreak/TagLineBreak:** Tags which are used to define no-break or line-break

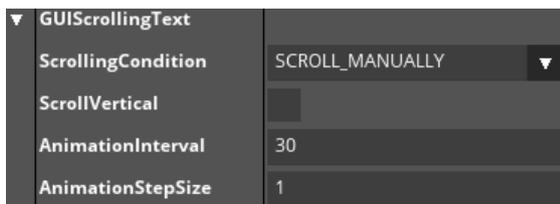
**TagOrderedList/TagUnorderedList:** Tags which are used to define ordered or unordered lists

**TagListItem:** Tag which specifies a new list-item inside a list

**TagOpening/TagClosing:** this string define the opening- and closing-part of a tag

**TagFinished:** this string defines the prefix to indentify a closing-tag

### 43. GUIScrollingText:



**ScrollingCondition:** One of the following options:

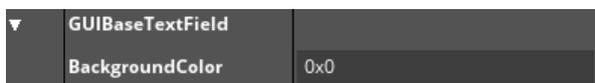
SCROLL_ALWAYS	The text will always be scrolled
SCROLL_CUT_TEXT_ALWAYS	The text will be scrolled when the text is truncated
SCROLL_CUT_TEXT_FOCUSED	The text will be scrolled when the text is truncated and the parent object is focused
SCROLL_FOCUSED	The text will be scrolled when its parent object is focused
SCROLL_MANUALLY	The Animation can be started or stopped manually using API-calls

**ScrollVertical:** If active, then text scrolled vertically, if not is will be scrolled horizontally.

**AnimationInterval:** Number of milliseconds between each animation-step

**AnimationStepSize:** Number of pixels the current position of the text will be increased/decreased for each step. If 0 is set as step size then no scrolling of text occurs

### 44. GUIBaseTextField:



**BackgroundColor:** Defines the background-colour for the textfield

## 45. UITextField:

▼	UITextField	
	BackgroundImage	DUMMY_IMAGE

**BackgroundImage:** Image which should be used for the textfield

## 46. UITextField:

▼	UITextField		
	ImageIDNormal	IMG_STDCTRL_INPUTFIELD_STAN...	
	ImageIDHighlighted	IMG_STDCTRL_INPUTFIELD_HIGH...	
	ImageIDGrayedOut	IMG_STDCTRL_INPUTFIELD_GRAY...	
	ImageIDFocussed	IMG_STDCTRL_INPUTFIELD_FOCU...	
▼	NinePatchBorders		
	Top	10	
	Bottom	10	
	Left	10	
	Right	10	

**ImageIDNormal/ImageIDHighlighted/ImageIDGrayedOut/ImageIDFocussed:** image ids which will be used as the background of the input-field according to the current state of the object.

## 47. GUIComboBox:

▼	GUIComboBox	
	InputActionType	INPUT_SEARCHED ▼
	FrameBackgroundColor	0xFF000000 V P
	FrameBorderWidth	1.000000
▼	GUICommand	
	CommandClassID	DUMMY_COMMAND ▼

**InputActionType:** Two options are present:

- 1) INPUT\_SEARCHED: Allows searching for a given entry from the drop down. Adding the entry inside the text field is not possible.
- 2) INPUT\_INSERTED: Allows inserting a given entry which also gets appended in the drop down list.

**FrameBackgroundColor:** Defines the background color of the frame which is shown when the drop down list appears:

**FrameBorderWidth:** Defines the width of the border of the frame.

## 48. GUIButton:

▼	GUIButton	
	ImageIDNormal	IMG_STDCTRL_IMGBTN_STANDA...
	ImageIDHighlighted	IMG_STDCTRL_IMGBTN_HIGHLIG...
	ImageIDPressed	IMG_STDCTRL_IMGBTN_PRESSED
	ImageIDGrayedOut	IMG_STDCTRL_IMGBTN_GRAYED...
	ImageIDFocused	IMG_STDCTRL_IMGBTN_FOCUSED
▼	NinePatchBorders	
	Top	5
	Bottom	5
	Left	5
	Right	5
	ButtonBlitType	STRETCH ▼
▼	ManualLayout	
	X	0.000000
	Y	0.000000
	Width	128.000000
	Height	32.000000

**ImageIDs:** Image-ids which will be used for the object according to its state.

**NinePatchBorders:** Defines the nine patch border which stretches images smaller than the object's size.

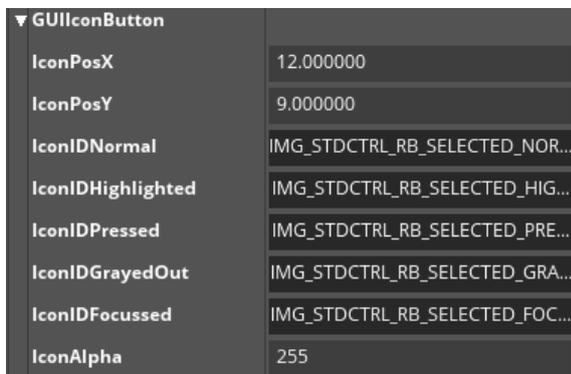
**ButtonBlitType:** Listed options are:

- STRETCH: Stretches the given image to the whole size of the control.
- CENTER: This leads to the image being drawn with its original size and being centered in the button.
- TILED: Allows the given image to tile up and cover the whole size of the button.
- MANUAL\_LAYOUT: The image is blitted based on the configuration provided inside manual layout option.

**X/Y:** The position defined for image being blitted on the button and used when the blit type is set with MANUAL\_LAYOUT option.

**Width/height:** Height and width defined for image being blitted on the button and used when the blit type is set with MANUAL\_LAYOUT option.

## 49. GUIIconButton:



▼ GUIIconButton	
IconPosX	12.000000
IconPosY	9.000000
IconIDNormal	IMG_STDCTRL_RB_SELECTED_NOR...
IconIDHighlighted	IMG_STDCTRL_RB_SELECTED_HIG...
IconIDPressed	IMG_STDCTRL_RB_SELECTED_PRE...
IconIDGrayedOut	IMG_STDCTRL_RB_SELECTED_GRA...
IconIDFocussed	IMG_STDCTRL_RB_SELECTED_FOC...
IconAlpha	255

**IconPosX/IconPosY:** Position of the icon-image relative to the object's top/left-

**IconIDNormal/IconIDHighlighted/IconIDPressed/IconIDGrayedOut/IconIDFocussed:** Image-ids which will be used for the icon according to the object's state

**IconAlpha:** Alpha-value used for the icon

## 50. GUIBlendButton:

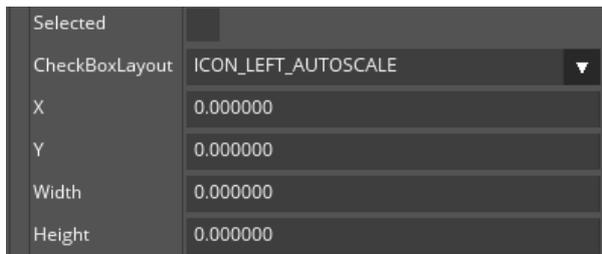


▼ GUIBlendButton	
BlendDuration	250
CrossFade	<input type="checkbox"/>

**BlendDuration:** Duration in milliseconds which will be used when the state of the object and thus the used image will change

**CrossFade:** If this is set, a cross-fade will be used to switch to the new image according to the object's state

## 51. GUIBaseCheckBox:



**Selected:** selection-state of the control

**CheckBoxLayout:** chooses how icon and text are layouted. Can be one of the following options:

MANUAL_LAYOUT	No automatic layout. Text position and size are specified manually
ICON_LEFT_AUTOSCALE	Icon left and scaled automatically, text will be drawn right to it
ICON_RIGHT_AUTOSCALE	Icon right and scaled automatically, text will be drawn left to it
ICON_LEFT_CENTERED	Icon left and not scaled, text will be drawn right to it
ICON_RIGHT_CENTERED	Icon right and not scaled, text will be drawn left to it
ICON_FILL_OBJECT	The Icon will fill the entire area of the object

**X/Y:** shows/defines position of the icon relative to the top/left-corner of the object.

**Width/Height:** size of the icon

## 52. GUICheckBox:

▼ GUICheckBox	
SelectedImageIDNormal	IMG_CHECKBOX_GREEN_SELECTED
SelectedImageIDHighli...	IMG_CHECKBOX_GREEN_SELECTED
SelectedImageIDPressed	IMG_CHECKBOX_GREEN_SELECTED
SelectedImageIDGraye...	IMG_CHECKBOX_GREEN_SELECTED
SelectedImageIDFocuss...	IMG_CHECKBOX_GREEN_SELECTED
NotSelectedImageIDNo...	IMG_CHECKBOX_GREEN_NOTSELE...
NotSelectedImageIDHig...	IMG_CHECKBOX_GREEN_NOTSELE...
NotSelectedImageIDPre...	IMG_CHECKBOX_GREEN_NOTSELE...
NotSelectedImageIDGr...	IMG_CHECKBOX_GREEN_NOTSELE...
NotSelectedImageIDFoc...	IMG_CHECKBOX_GREEN_NOTSELE...

**SelectedImageIDs:** Image-ids which are used for the selected-state according to the state of the object

**NotSelectedImageIDs:** Image-ids which are used for the not-selected-state according to the state of the object

### 53. GUIRadioButton:

▼	GUIRadioButton	
	SelectedIDNormal	IMG_STDCTRL_RB_SELECTED_NO...
	SelectedIDHighlighted	IMG_STDCTRL_RB_SELECTED_HI...
	SelectedIDPressed	IMG_STDCTRL_RB_SELECTED_PR...
	SelectedIDGrayedOut	IMG_STDCTRL_RB_SELECTED_GR...
	SelectedIDFocussed	IMG_STDCTRL_RB_SELECTED_FO...
	NotSelectedIDNormal	IMG_STDCTRL_RB_NOTSELECTED...
	NotSelectedIDHighligh...	IMG_STDCTRL_RB_NOTSELECTED...
	NotSelectedIDPressed	IMG_STDCTRL_RB_NOTSELECTED...
	NotSelectedIDGrayed...	IMG_STDCTRL_RB_NOTSELECTED...
	NotSelectedIDFocussed	IMG_STDCTRL_RB_NOTSELECTED...
▼	NinePatchBorders	
	Top	0
	Bottom	0
	Left	0
	Right	0

**SelectedImageIDs:** Image-ids which are used for the selected-state according to the state of the object

**NotSelectedImageIDs:** Image-ids which are used for the not-selected-state according to the state of the object.

**NinePatchBorders:** Defines the nine patch borders which stretch images smaller than the object's size.

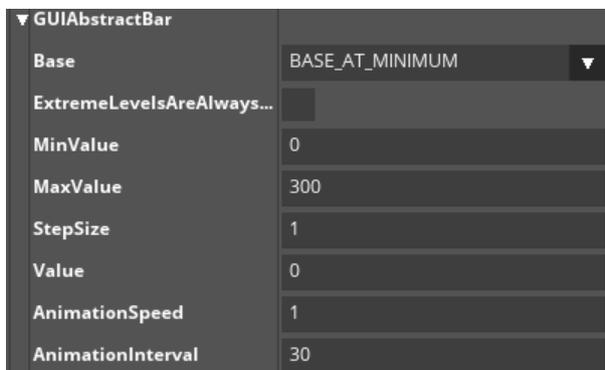
### 54. GUIScrollBar:

▼	GUIScrollBar	
	ScrollInterval	40
	ScrolledObject	ANIMATION_CONTAINER ▼

**ScrollInterval:** Number in milliseconds which defines the time between stepwise increasing/decreasing the current position, if the buttons for left/right or up/down are continuously pressed

**ScrolledObject:** ID of the target-object which should be moved using the scrollbar

## 55. GUIAbstractBar:



Property	Value
Base	BASE_AT_MINIMUM
ExtremeLevelsAreAlwaysValid	<input type="checkbox"/>
MinValue	0
MaxValue	300
StepSize	1
Value	0
AnimationSpeed	1
AnimationInterval	30

**Base:** Shows/defines base of the slider, two possible options are available

- 1) **BASE\_AT\_MAXIMUM:** With this option the slider base is set to the position of minimum values.
- 2) **BASE\_AT\_MINIMUM:** With this option the slider base is set to the position of maximum values.

**ExtremeLevelsAreAlwaysValid:** Shows/defines True, if the maximum is also a valid current value, no matter what the step size is. False, else.

**MinValue/MaxValue:** Minimum/maximum value which is used for the range

**StepSize:** Step size to display a percentage of a slider at a time.

**Value:** Currently set value

**AnimationSpeed:** Speed for animated scrolling.

**AnimationInterval:** Timer interval in milliseconds for animation.

## 56. GUIBaseSlider:

▼	GUIBaseSlider	
	Orientation	HORIZONTAL ▼
	MinPos	0.000000
	MaxPos	98.000000
	TrackDistance	0.000000

**Orientation:** Sets the orientation of the slider either to horizontal or vertical

**MinPos/MaxPos:** Position where the minimum/maximum-value of the slider are displayed

**TrackDistance:** The length of the slider track in pixel. Higher the value the lesser is the distance for the slider track to move.

## 57. GUISlider:

▼	GUISlider	
	ImageIDBackground	IMG_STDCTRL_SLD_BG
	ImageIDKnobNormal	IMG_STDCTRL_SLD_KNOB_NORM...
	ImageIDKnobHighlight...	IMG_STDCTRL_SLD_KNOB_HIGHL...
	ImageIDKnobPressed	IMG_STDCTRL_SLD_KNOB_PRESS...
	ImageIDKnobGrayedO...	DUMMY_IMAGE
	BackgroundMargin	0.000000
	Stretch	
	▼ NinePatchBorders	
	Top	0
	Bottom	0
	Left	0
	Right	0

**ImageIDBackground:** Image-id of the background (i.e. the slider area)

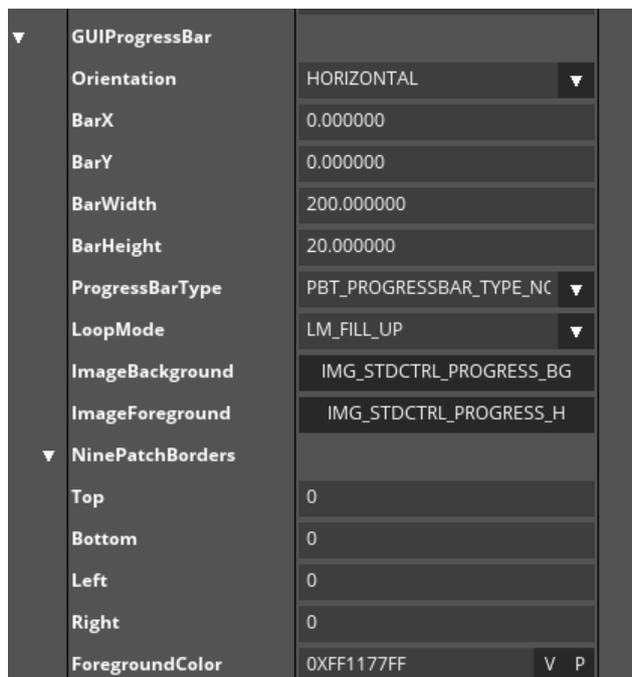
**ImageIDKnobNormal/ImageIDKnobHighlighted/ImageIDKnobPressed/ImageIDKnobGrayedOut:** Image-ids which are used for the knob according to the object's state.

**BackgroundMargin:** Number of pixels the background-image is moved relative to the top/left-corner of the object.

**Stretch:** Stretches the slider to the whole size of the object.

**NinePatchBorders:** Defines the nine patch borders which stretch images smaller than the object's size.

## 58. GUIProgressBar:



GUIProgressBar	
Orientation	HORIZONTAL
BarX	0.000000
BarY	0.000000
BarWidth	200.000000
BarHeight	20.000000
ProgressBarType	PBT_PROGRESSBAR_TYPE_NC
LoopMode	LM_FILL_UP
ImageBackground	IMG_STDCTRL_PROGRESS_BG
ImageForeground	IMG_STDCTRL_PROGRESS_H
NinePatchBorders	
Top	0
Bottom	0
Left	0
Right	0
ForegroundColor	0XFF1177FF

**Orientation:** Two options are there:

- Horizontal: If this is set the progress bar will be drawn horizontally.
- Vertical: If this option is selected, then the progress bar is drawn vertically.

**BarX/BarY:** Position of the bar relative to the top/left-corner of the object

**BarWidth/BarHeight:** Width/Height of the bar

---

**ProgressBarType:** Can be one of the following options:

PBT_PROGRESSBAR_TYPE_NORMAL	Designed for full control of the fill state via the application
PBT_PROGRESSBAR_TYPE_ACTIVITY	Designed for representation of unpredictable duration of operations

**LoopMode:** Can be one of the following options:

LM_FILL_UP	Fill up the bar until full and starts repeatedly again with an empty bar
LM_LOOP_MODE	Repeatedly move the bar from start to end
LM_REVERTLOOP_MODE	Move the bar from start to end and back to start again
LM_FILL_UP_TILE	Tile up the foreground image and fill it up on the bar until full and starts repeatedly again.

**ImageBackground/ImageForeground:** image-ids which are used for the background and foreground of the bar

**NinePatchBorders:** Defines the nine patch borders which stretch images smaller than the object's size.

**Foreground Color:** Defines the foreground color of the bar.

## 59. CGUICircularSlider:

GUICircularSlider	
SmoothControl	<input type="checkbox"/>
ImageIDBackground	IMG_STDCTRL_CIRCULAR_BG
ImageIDKnobNormal	IMG_STDCTRL_CIRCULAR_KNOB
ImageIDKnobHighlight...	IMG_STDCTRL_CIRCULAR_KNOB_H
ImageIDKnobPressed	IMG_STDCTRL_CIRCULAR_KNOB_P
ImageIDKnobGrayedO...	IMG_STDCTRL_CIRCULAR_KNOB_G
CircleRadius	55.000000
StartAngle	-135.000000
EndAngle	135.000000

**SmoothControl:** if this is active the position of the slider is not limited to valid values of the range, but can be between.

**ImageIDBackground:** Id of the image which is used for the background

**ImageIDKnobNormal/ImageIDKnobHighlighted/ImageIDKnobPressed/ImageIDKnobGrayedOut:** ids for the images used for the knob according to the object's state

**CircleRadius:** Radius of the circular area where the knob is drawn

**StartAngle/EndAngle:** Starting-/ending-angle which define the valid range where the knob is drawn, will be clamped to range -180 to 180, where 0 is at top middle

## 60. GUIKnob:

GUIKnob	
AxisControl	<input type="checkbox"/>
Orientation	HORIZONTAL ▾
SmoothControl	<input type="checkbox"/>
ImageIDBackground	IMG_STDCTRL_CIRCULAR_BG
ImageIDKnobNormal	IMG_STDCTRL_KNOB_HANDLE
ImageIDKnobHighlighted	IMG_STDCTRL_KNOB_HANDLE
ImageIDKnobPressed	IMG_STDCTRL_KNOB_HANDLE
ImageIDKnobGrayedOut	DUMMY_IMAGE
StartAngle	-135.000000
EndAngle	135.000000

**AxisControl:** If this is active the Knob can be adjusted using horizontal/vertical drags instead of circular movement on the rim.

**Orientation:** This defines the orientation for AxisControl

**SmoothControl:** if this is active the position of the slider is not limited to valid values of the range, but can be between.

**ImageIDBackground:** Id of the image which is used for the background

**ImageIDKnobNormal/ImageIDKnobHighlighted/ImageIDKnobPressed/ImageIDKnobGrayedOut:** ID's for the images used for the knob according to the object's state

**StartAngle/EndAngle:** Starting-/ending-angle which define the valid range where the knob is drawn. Will be clamped to -180 to 180 where 0 is at top middle.

## 61. GUISegmentBar:

▼	GUISegmentBar	
	InactiveImage	IMG_STDCTRL_SEGMENT_UNFILL...
	ActiveImage	IMG_STDCTRL_SEGMENT_FILLED
	▶	NinePatchBorders
	Orientation	HORIZONTAL ▼
	MarginX	0.000000
	MarginY	0.000000
	Gap	0.000000
	SegmentWidth	16.000000
	SegmentHeight	16.000000
	AutoStretch	<input checked="" type="checkbox"/>

**InactiveImage/ActiveImage:** ImageIDs which are used for inactive/active segments

**NinePatchBorders:** Ninepatch which is used for drawing the segments

**Orientation:** Orientation of the bar, allowed orientations are either horizontal or vertical.

**MarginX/MarginY:** Margin in pixels which is used to draw the segments within the object

**Gap:** number of pixels defining the gap between the segments

**SegmentWidth/SegmentHeight:** Fixed sizes in pixels for one segment, if "AutoStretch" is not used

**AutoStretch:** If this is set, the size of one segment is calculated using the object's size, the used margin and the difference between minimum and maximum value

## 62. GUIRangeSlider:

GUIRangeSlider	
BackgroundImage	IMG_STDCTRL_SLD_BG
HandleImage	IMG_KNOB
RangeImage	IMG_STDCTRL_SCROLLBAR_BACK...
Orientation	HORIZONTAL ▼
MarginX	0.000000
MarginY	0.000000
Value2	50

**BackgroundImage:** ImageID which is used to draw the background

**HandleImage:** ImageID which is used to draw the slider-handles

**RangeImage:** ImageID which is used to draw the range between the two slider-handles

**Orientation:** orientation of the slider, either horizontal or vertical orientation can be selected.

**MarginX/MarginY:** Margin in pixels which is used to draw the slider within the object.

**Value2:** second value of the range

## 63. GUIRepositionCompositeObject:



**Note:** The repositioning effect will only take place when resizing the container.

**BasePoint:** The base point on which the repositioning is based. Following options are present:

- **REPOSITION\_ALIGN\_TOP:** The base point is repositioned with top alignment and all the child objects are repositioned from top to bottom in the container.
- **REPOSITION\_ALIGN\_BOTTOM:** The base point is reposition with bottom alignment and all the child objects are repositioned from bottom to top in the container.
- **REPOSITION\_ALIGN\_LEFT:** The base point is repositioned with left alignment and all the child objects are repositioned from left to right in the container.
- **REPOSITION\_ALIGN\_RIGHT:** The base point is repositioned with right alignment and all the child objects are repositioned from right to left in the container.

**GapBetweenChildren:** The gap between the children in pixel

**BorderSpace:** The border space to the composite border in pixel.

## 64. GUILayerContainer



**LayerID:** the ID of the layer to use. Must match the ID inside layer-configuration for DC-Wrapper

**UserContent:** if true, UserContent will be displayed inside the container. Will draw red diagonal line

**ApplyClipping:** if true, the container will retain its clipping, even though layers are drawn on top

## 65. GUICenterFocusContainer (O):

▼ GUICenterFocusContainer	
CenterOnChildrenOnly	<input type="checkbox"/>
InertiaX	15.000000
InertiaY	15.000000
CenterX	50.000000
CenterY	50.000000
VerticalDragSensitivity	30
HorizontalDragSensitivity	30

**CenterOnChildrenOnly:** Automatically centres the currently focused control at the supplied position.

This centering can be animated with effects like speed-up or slow-down

**InertiaX/InertiaY:** Inertia value in horizontal/vertical direction. Inertia value is used as a divisor for movement-speed, so higher values result in slower movement

**CenterX/CenterY:** Position of the anchor-point, where the focussed object will be centered

**VerticalDragSensitivity/HorizontalDragSensitivity:** Distance in pixels which needs to be exceeded during a drag before it moves to the next child-object

## 66. GUIScrollView:

▼ GUIScrollView	
VerticalScrollbarPolicy	AUTOMATIC ▼
HorizontalScrollbarPolicy	AUTOMATIC ▼
ScrollingInertia	1.000000
ViewportXBorder	0
ViewportYBorder	0

**VerticalScrollbarPolicy/HorizontalScrollbarPolicy:** Following options are available for setting the policy of scrollbar for the control:

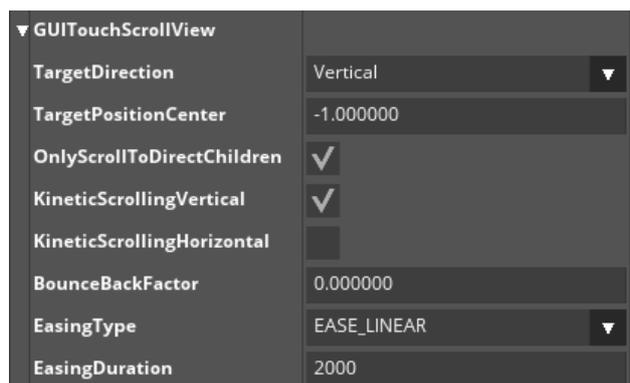
- ALWAYS\_VISIBLE: The scrollbars are always shown, even if the content might fit the available space

- **AUTOMATIC:** The scrollbars are only shown if the available width/height is not enough to display the content
- **NEVER\_VISIBLE:** The scrollbars are never shown

**ScrollingInertia:** Value for the scrolling-inertia. Valid values are greater than 1

**ViewportXBorder/ViewportYBorder:** Horizontal/Vertical (in X/Y direction) gap between visible area and the area outside of which scrolling will occur.

## 67. GUITouchScrollView:



**TargetDirection:** Defines the direction for scrolling or dragging the contents of scroll view. Following options are available for Target Direction:

- **Vertical:** The contents of scroll view move in vertical direction upon dragging using finger/mouse
- **Horizontal:** The contents of scroll view move in horizontal direction upon dragging using finger/mouse
- **Free:** The contents of scroll view are free to move in any direction and is dependent on the direction of drag
- **None:** With this option the contents doesn't move with mouse drag or touch.

**TargetPositionCenter:** Defines the target position for the XPosCenter or the YPosCenter of the active object. When clicking on the scroll view, the nearest object to the click position (in destination direction) is chosen as the active object. When dragging, the nearest object to the target position is

chosen as active object when the drag is finished. When kinetic scrolling is activated, the nearest object to the target position is chosen when the kinetic animation is about to fade out.

**OnlyScrollToDirectChildren:** Only direct children of the scroll view, or any descendant.

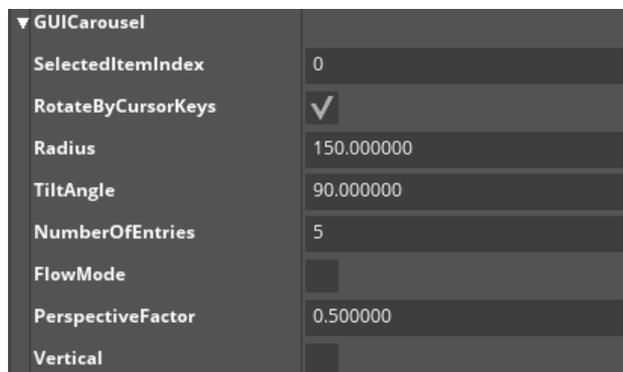
**KineticScrollingVertical/KineticScrollingHorizontal:** With this option kinetic scrolling is activated for horizontal/vertical direction which will trigger a scroll animation after the drag that slowly fades out with time.

**BounceBackFactor:** When an edge of the scroll view is reached during an animation, this factor will be used to animate a bounce-back

**EasingType:** EasingType which is used for kinetic scrolling

**EasingDuration:** Duration in milliseconds of easing animation for kinetic scrolling

## 68. GUICarousel:



Property	Value
SelectedItemIndex	0
RotateByCursorKeys	<input checked="" type="checkbox"/>
Radius	150.000000
TiltAngle	90.000000
NumberOfEntries	5
FlowMode	<input type="checkbox"/>
PerspectiveFactor	0.500000
Vertical	<input type="checkbox"/>

**SelectedItemIndex:** Index of the child object which will be selected (shown in the middle)

**RotateByCursorKeys:** Setting this option gives the user the possibility to use left/right cursor keys to rotate the carousel. If this option is set, it might interfere with standard focusing behaviour

**Radius:** Current radius (in pixel) of the virtual circle used to place the child-objects

**TiltAngle:** Tilt angle of the virtual circle used to place the child-objects

**NumberOfEntries:** Defines a virtual number of entries used in the carousel, e.g. for calculating the spacing between them. This setting is only relevant in FlowMode.

**FlowMode:** activates flow mode

**PerspectiveFactor:** shows/defines perspective Factor. The width and height of the child-objects within the carousel get recalculated according to perspective. This setting specifies the strength of perspective distortion which is applied to objects inside tilted carousels. Objects in the background will appear smaller than those in the foreground. Objects in untilted carousels, and objects that reside on the X-Axis of tilted carousels, will be displayed using their original size. Sensible values are typically in the range of 0 to 1, where 0 means no perspective effect at all, and larger values result in stronger zooming / shrinking. Note that Z-Ordering will not work if perspective-Factor is 0.

**Vertical:** If this is set, the carousel is arranged vertically, otherwise horizontally

**Note:** Using the carousel in flow-mode:

When working with huge numbers of children in a carousel, you will notice that the carousel tends to appear very crowded. This is where the "flow mode" comes in handy. When used in "flow mode", the carousel can contain very large numbers of entries of which only a few are visible at any time. You can specify the number of virtual entries in the carousel, which will affect the spacing between the visual children, and therefore also number of simultaneously visible ones. If for instance you are having a total of 1000 entries in your carousel, and set the number of virtual entries to 10, the carousel will arrange its content as if it had only 10 children, by spreading out these 10 along the carousel's 360 degrees. But in fact you will only see the currently selected (=frontmost) child object, plus the two children to its left and the two children to its right side.

## 69. GUICalendar:

▼	GUICalendar	
	Day	1
	Month	1
	Year	2019
	DayWidth	0.000000
	DayHeight	0.000000
	ShowHeadLine	<input checked="" type="checkbox"/>
	HeadlineHeight	30.000000
	HeadLineFont	FNT_DEFAULT
	HeadLineColor	0xFF000000 V P
	BackgroundImage	DUMMY_IMAGE
▶	NinePatchBorders	
	MarkerImage	IMG_STDCTRL_KEYBOARDBTN_FOCUSED
▶	NinePatchBorders	
	Font	FNT_DEFAULT
	FontColor	0xFF0088AA V P
	InactiveFont	FNT_DEFAULT
	InactiveColor	0xFF888888 V P
	ShowWeekDays	<input checked="" type="checkbox"/>
	FirstDayOfWeek	1
	FillEmptySpaces	<input checked="" type="checkbox"/>
	TouchControl	<input checked="" type="checkbox"/>

**Day/Month/Year:** Values for the currently selected day

**DayWidth/DayHeight:** Sizes in pixels for one cell representing a day. If this is set to 0, the width/height is calculated according to the object's size

**ShowHeadLine:** If this is set, a headline showing the current month and year will be displayed above the calendar

**HeadlineHeight:** Height in pixels which is used for the headline

**HeadLineFont/HeadLineColor:** FontID and colour which is used for the text inside the headline

**BackgroundImage:** ImageID which is used to draw the background

**NinePatchBorders:** Ninepatch which is used for drawing the background

**MarkerImage:** ImageID which is used to draw the marker which shows the currently selected day

**NinePatchBorders:** Ninepatch which is used for drawing the marker

**Font/FontColor:** FontID and colour which is used to draw the entries of the calendar which are part of the currently selected month.

**InactiveFont/InactiveColor:** FontID and colour which is used to draw the entries of the calendar not belonging to the currently selected month

**ShowWeekDays:** If this is active, the days of the week are shown above the calendar

**FirstDayOfWeek:** Defines the first day of the week which appears in the first column of the calendar (0 = Sunday, 6 = Saturday)

**FillEmptySpaces:** If this is set, the last days of the previous month and the first days of the next month are drawn to fill up the empty spaces

**TouchControl:** If this is set, the current month can be changed by dragging the calendar vertically, and the current year by dragging it horizontally. This does not disable clicking for selecting the currently set day.

## 70. GUIChart:

GUIChart			
ImageIDBackground	DUMMY_IMAGE		
Representation			
DataRepresentation	Points + Lines		
X-Axis-Representation	Line, Scaling and Text		
Y-Axis-Representation	Line, Scaling and Text		
TextFontID	FNT_DEFAULT		
TextColor	PROP_TEXT_COLOR_DEFAU	V P	
AxisLineWidth	1.000000		
AxisColor	PROP_TEXT_COLOR_DEFAU	V P	
PointSize	4.000000		
HelperLines			
AxisAlignment	None		
HorizontalColor	0xFFCFD8DC	V P	
VerticalColor	0xFFCFD8DC	V P	
DrawLineWidth	2.000000		
DrawLineColor			
LineColors	3 (click to add more)		
Color	PROP_CHART_COLOR1	V P	- ^ v
Color	PROP_CHART_COLOR2	V P	- ^ v
Color	PROP_CHART_COLOR3	V P	- ^ v
DrawPointColor			
PointColors	3 (click to add more)		
Color	PROP_CHART_COLOR1	V P	- ^ v
Color	PROP_CHART_COLOR2	V P	- ^ v
Color	PROP_CHART_COLOR3	V P	- ^ v

BarColor			
BarColors	3 (click to add more)		
Color	PROP_CHART_COLOR1	V P	- ^ v
Color	PROP_CHART_COLOR2	V P	- ^ v
Color	PROP_CHART_COLOR3	V P	- ^ v
MinValueX	0.000000		
MaxValueX	100.000000		
MinValueY	0.000000		
MaxValueY	100.000000		
RestrictNavigation	<input checked="" type="checkbox"/>		
ScalingXAxis	5		
ScalingYAxis	5		
LenghtOfScaling	10.000000		
ZoomFactorX	2.000000		
ZoomFactorY	2.000000		
StartValueX	1970		
Selection			
EnableSelection	<input type="checkbox"/>		
SelectionColor	0xFFFF0000	V P	
PointSelectionTolerance	0.000000		

**ImageIDBackground:** ImageID which is used to draw the background

**DataRepresentation:** Defines how data is represented on the chart, The options available in the drop down are: **Point, Lines, Points + Lines** and **Bars**

**X-Axis-Representation/Y-Axis-Representation:** Defines how the axes are drawn. Following options are possible: **hide, lines, lines and scaling, lines, scaling and text** can be used

**TextFontID:** FontId which is used for labelling the axis

**TextColor:** Colour which is used to draw the axis and the corresponding labels

**AxisLineWidth:** line-width in pixel of the axis

**AxisColor:** color which is used to draw the axis

**PointSize:** size in pixels of the points which are used to draw the data

**AxisAlignment:** alignment of the helperlines, which can be none, horizontal, vertical and both

**HorizontalColor/VerticalColor:** colors for the helperlines

**DrawLineWidth:** width of the lines which are used to draw the data

**DrawLineColor/DrawPointColor/BarColor:** Colours which are used to draw the data according to the type of data-representation. Each "Color"-entry stands for a value-line.

**MinValueX/MaxValueX:** minimum/maximum value which is shown on the X-axis

**MinValueY/MaxValueY:** minimum/maximum value which is shown on the Y-axis

**RestrictNavigation:** if this is active the view of the chart cannot be moved out of range of the displayed values, especially when zooming

**ScalingXAxis/ScalingYAxis:** number of labels which are shown on X-axis/Y-axis. The step size between each value labelled on X Axis or Y Axis is calculated as  $(\text{MaxValue} / \text{Scaling XAxis or Scaling YAxis})$

**LengthOfScaling:** length in pixels of a scale on the axis

**ZoomFactorX/ZoomFactorY:** zoom-factors which are used if ZoomIn or ZoomOut is used

**StartValueX:** first value for the label at X-axis

**EnableSelection:** if this is active, values can be selected by clicking on them. The selected value is shown in the upper part of the control

**SelectionColor:** color which is used to indicate that this value is currently selected

**PointSelectionTolerance:** tolerance in pixels around a value to simplify selection

## 71. GUIClock:

GUIClock	
StartHours	0
StartMinutes	0
StartSeconds	0
ImageIDBackground	IMG_STDCTRL_CLOCK_BG
ImageIDHourHandle	IMG_STDCTRL_CLOCK_HOUR
HourNeedleColor	0xFFFF0000 V P
HourNeedleLenght	50.000000
HourNeedleWidth	6.000000
ImageIDMinuteHandle	IMG_STDCTRL_CLOCK_MINUTE
MinuteNeedleColor	0xFFFF0000 V P
MinuteNeedleLenght	140.000000
MinuteNeedleWidth	3.000000
ImageIDSecondHandle	IMG_STDCTRL_CLOCK_SECOND
SecondNeedleColor	0xFFFF0000 V P
SecondNeedleLenght	140.000000
SecondNeedleWidth	1.000000
ShowSecondHandle	<input checked="" type="checkbox"/>
Shadows	<input type="checkbox"/>
ShadowColor	0xFF000000 V P
AutoStart	<input type="checkbox"/>

**StartHours/StartMinutes/StartSeconds:** Starting-time of the clock

**ImageIDBackground:** ImageID which is used to draw the background

**ImageIDHourHandle:** ImageID which is used to draw the hour-handle of the clock

**HourNeedleColor/HourNeedleLenght/HourNeedleWidth:** Colour, length and width which are used to draw the hour-handle with drawing-operations instead of an image. These settings are only active if DUMMY\_IMAGE is set for ImageIDHourHandle

**ImageIDMinuteHandle:** ImageID which is used to draw the minute-handle of the clock

**MinuteNeedleColor/MinuteNeedleLength/MinuteNeedleWidth:** Colour, length and width which are used to draw the minute-handle with drawing-operations instead of an image. These settings are only active if DUMMY\_IMAGE is set for ImageIDMinuteHandle

**ImageIDSecondHandle:** ImageID which is used to draw the second-handle of the clock

**SecondNeedleColor/SecondNeedleLength/SecondNeedleWidth:** Colour, length and width which are used to draw the second-handle with drawing-operations instead of an image. These settings are only active if DUMMY\_IMAGE is set for ImageIDSecondHandle

**ShowSecondHandle:** If this is set, the second-handle is displayed else the second handle is not displayed.

**Shadows:** If this is set, shadows are drawn for each handle. Shadows are only drawn if drawing-operations are used to draw the handle

**ShadowColor:** Colour which is used to draw the shadow of the handles

**AutoStart:** If this is set, the clock starts ticking on creation

## 72. GUIGauge:

GUIGauge	
MinAngle	0.000000
MaxAngle	180.000000
GaugeImageID	IMG_STDCTRL_GAUGE
NeedleImageID	IMG_STDCTRL_GAUGE_NEEDLE
NeedleColor	0xFFFF0000 V P
ShadowColor	0x88222222 V P
NeedleLength	50.000000
NeedleWidth	3.000000
Simulate	<input type="checkbox"/>

**MinAngle/MaxAngle:** Angle which is used for the needle for minimum/maximum values

**GaugeImageID/NeedleImageID:** Image used for gauge/needle

**NeedleColor/ShadowColor:** Shows/defines color of the needle. This attribute is only used if “DUMMY\_IMAGE” is set for NeedleImageID

**NeedleLength/NeedleWidth:** Length and width of the needle in pixels measured from centre of gauge to its tip. This attribute is only used if “DUMMY\_IMAGE” is set for NeedleImageID

**Simulate:** Activate simulation-mode. In Simulation-mode the gauge will move repeatedly from minimum value to maximum value and back.

## 73. GUIWheel:

GUIWheel	
BackgroundImageID	DUMMY_IMAGE
▶ NinePatchBorders	
Orientation	VERTICAL ▼
EasingType	EASE_OUT_EXPO ▼
EasingDuration	1500
EntriesFontID	FNT_DEFAULT
EntriesFontColor	0xFF000000 V P
FocussedEntryFontID	FNT_DEFAULT
FocussedEntryFontColor	0xFF000000 V P
EntryWidth	100.000000
EntryHeight	20.000000
FocussedValue	0
Cyclic	<input checked="" type="checkbox"/>
UseRange	<input checked="" type="checkbox"/>
MinValue	-10
MaxValue	10
StepSize	2
MinimalNumberOfDig...	0
FillUpString	
PrefixString	
PostfixString	
FillUpFromBeginOfText	<input checked="" type="checkbox"/>

**BackgroundImageID:** Image to be displayed on the background of the wheel. This image will always be stretched to fill the bounding rectangle of the object.

**NinePatchBorder:** Ninepatch which will be used to stretch the background-image

**Orientation:** States the orientation of the wheel, either horizontal or vertical.

**EasingType:** EasingType used for kinetic animation.

**EasingDuration:** Duration of easing animation for kinetic scrolling in milliseconds.

**EntriesFontID/FocussedEntryFontID:** Font type which is used for non-focused/focussed entries of the wheel.

**EntriesFontColor/FocussedEntryFontColor:** Color of the text which is used for non-focused/focussed entries

**EntryWidth/EntryHeight:** Width/Height for one entry

**FocussedValue:** Currently focused entry value.

**Cyclic:** If this is set, the wheel is cyclic, i.e. begins with the first element after the last element. If this is not set, the wheel has a lower and an upper end.

**UseRange:** If this is set, the entries of the wheel are auto-generated using the minimum/maximum-value and the step-size otherwise the entries are not displayed.

**MinValue/MaxValue:** Minimum/Maximum-value allowed for the entries of the wheel.

**StepSize:** Size of one step for the entries to be displayed on the wheel.

**MinimalNumberOfDigits:** Set the minimal number of digits which should be displayed for each entry.

For example, if this is set to 2 all entries from 0-9 will be filled up with a string

**FillUpString:** Defines a string which will be used to fill up missing digits, e.g. "0"

**PrefixString:** Defines a string which will be placed in front of each text-entry

**PostfixString:** Defines a string which will be placed behind each text-entry. This setting can be used for displaying a unit string after each text-entry.

**FillUpFromBeginOfText:** If this is set, entries will be filled in front of entry, if false at end.

## 74. GUIKeyboard

▼	GUIKeyboard		
	Layout	DUMMY_RESOURCE	▼
	TargetObject	NO_HANDLE	▼
	TextColorStandard	PROP_TEXT_COLOR_KEYBO	V P
	TextColorHighlighted	PROP_TEXT_COLOR_KEYBO	V P
	TextColorGrayedOut	PROP_TEXT_COLOR_KEYBO	V P
	TextColorPressed	PROP_TEXT_COLOR_KEYBO	V P
	Font	FNT_KEYBOARD	
	ImageIDNormal	IMG_KEY	
	ImageIDHighlighted	IMG_KEY_H	
	ImageIDPressed	IMG_KEY_P	
	ImageIDGrayedOut	IMG_KEY	
	ImageIDFocused	IMG_KEY	

**Layout:** Keyboard-Layout to use. If found in General Resources. DUMMY\_RESOURCE will use English layout

**TargetObject:** ID of the object which should receive keyboard-input. If NO\_HANDLE is set, the currently focussed object will receive keyboard-input

**TextColorStandard/TextColorHighlighted/TextColorGrayedOut/TextColorPressed:** colours which are used for the text displayed on the keyboard-buttons

**Font:** Font which is used for the keyboard-buttons

**ImageIDNormal/ImageIDHighlighted/ImageIDPressed/ImageIDGrayedOut/ImageIDFocused:** images used for the keyboard-buttons

## 75. GUIVideo



**VideoID:** ID of the Video which should be played. Is found in General Resources

**StartFrame:** Frame to start with

**AutoStart:** if true the video will start playing after controls has been created

**Loop:** if true the video will jump back to start when the end was reached