Core

The Core Controls are a set of Controls, designed to provide a minimal core set of functionality. Therefore, the Core Controls also reduce their size and provide increased performance, by limiting its feature set and reducing the number of used behaviors.

The Core Controls are designed for

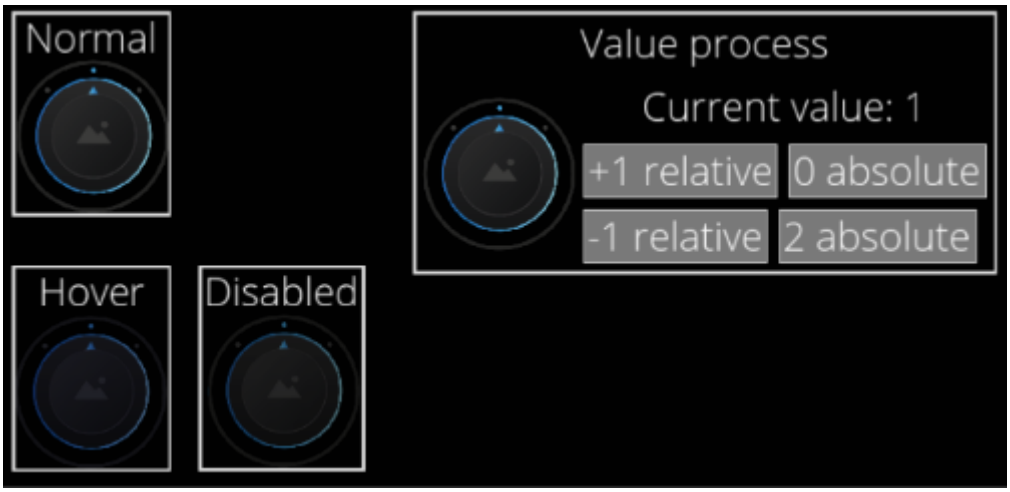
- **Low-End Platforms**
  Low memory footprint and increased performance by reduced feature set
- **Basis for custom Controls**
  As the Core Controls cover basic features, it is easy possible to extend them for custom behavior (e.g. a specific custom behavior for a custom button) instead of stripping down a full featured Control

Please note, that the not all Controls have a Core Control variant, as it is not suitable for all available Controls.

CoreDial

This control is used to set values within a specified range. Rotating the knob changes the value and updates the corresponding display. The degree of rotation is adjusted according to the configured Step Size and snaps to predefined positions. The knob's state is represented by displaying an image from the BackgroundImages list corresponding to the current value. For example, in a three-step rotation setup where the minimum rotation is -45 degrees, the maximum rotation is +45 degrees, and the step size is 45 degrees, three BackgroundImages need to be assigned. Additionally, values can be adjusted using specified keyboard keys (Increment Key Code, Decrement Key Code).

In the Core version, focus and color blending functionalities are not included, and the design is specialized for basic rotational operations.

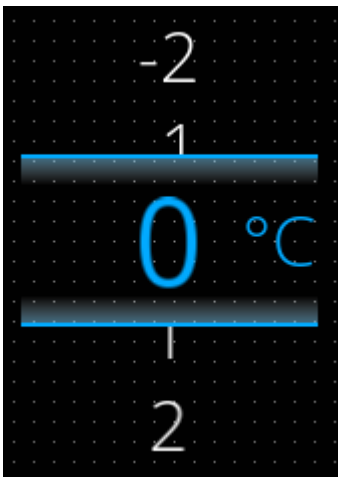


Properties

Property Name	Type	Description
Enabled	bool	If set to false it will not react on focus events and does not have to support different visualizations for the different states.
Focusable	bool	If true, the control is focusable. Otherwise the control can not be focused.
Current Value	FeatStd::Int32	Current value for the dial.
Step Size	FeatStd::Float	Step size in degrees.
Minimum Rotation	FeatStd::Float	Minimum rotation of the dial.
Maximum Rotation	FeatStd::Float	Maximum rotation of the dial.
Knob Image	Candera::Image2D	The image to set on the effect.
BackgroundImages	Candera::Image2D	A list of images which should be available for selection.
Disabled Color	Candera::Color	The color which can be used from other Behaviors.
Increment Key Code	Candera::KeyboardKeyCode::Enum	Check the KeyCode of the pressed key (can also be modifier).
Decrement Key Code	Candera::KeyboardKeyCode::Enum	Check the KeyCode of the pressed key (can also be modifier).

CoreRoll

CoreRoll is a control that provides an interface for displaying numeric values as a vertical list within a defined range, allowing users to select or change values. The current value is emphasized in the center "lens" area. An option for changing via touch, pointer is kept, but the keyboard input has been removed to optimize the control. It is optimized for lightweight and intuitive UI implementations that focus on numeric display and selection. Essential visual customizations, such as color, font style, lens display control, and unit text settings, are retained.



Properties

Property Name	Type	Description
Focusable	bool	If true, the control is focusable. Otherwise the control cannot be focused.
Value	Float	The value which will be changed using the roll
Min	Float	The minimum value which can be reached. This value is always smaller or equal to the MaxValue.
Max	Float	The maximum value which can be reached. This value is always bigger or equal to MinValue
Step	Float	The increment/decrement of the value.
LensVisible	bool	Is lens group visible? Enables highlighting of the current value, default zooming and bordering it.
Use Cyclic Roll	bool	Is roll cyclic? If true the MinValue of the Control can be reached after the MaxValue of the Control and vice versa

Allow Dragging Past Limit	bool	Allow temporary over-dragging (dragging to blank values before first and after last entry). If true there is visible animation of the Control being „out of range“ while moving below minimum and above maximum value.
Value Style	Candera::TextRendering::Style	Sets the text style for the values outside the Lens.
Value Color	Candera::Color	Specifies color of values in the vertical list outside the Lens
Lens Style	Candera::TextRendering::Style	Sets the text style for the values inside the Lens.
Unit Text	String	Text used in lens (it means: when the lens is used at all) to present a unit of the value
Unit Style	Candera::TextRendering::Styler	Sets the text style for the Unit Text.
Unit Margin	Candera::Margin	Specifies margin of the UnitText
Unit Color	Candera::Color	Specifies color of the UnitText
Decoration Image	Candera::Image2D	Specifies a border image of the Lens

CoreSpinBox

CoreSpinBox is a control that allows users to increment or decrement numeric values using up/down arrow buttons. The values can be modified via the arrow buttons only and The option for changing the values via keyboard inputs has been removed to optimize the control, with support for basic parameters such as minimum, maximum, and step size.

This control prioritizes simplified visual presentation and user interaction. Compared to SpinBox, it is more lightweight and suited for cases where UI performance and streamlined layout are important.



Properties

Property Name	Type	Description

Enabled	bool	If set to false it will not react on focus events and does not have to support different visualizations for the different states.
Focusable	bool	If true, the control is focusable. Otherwise the control can not be focused.
Value	Variant	The value to be formatted
Step Size	Variant	Increment/Decrement step value.
Minimum Value	Variant	Minimum value
Maximum Value	Variant	Maximum value
Text Format	String	The format string which includes one format specifier (beginning with %). See C printf() function.
Text Alignment	Enum (Left, Centered, Right, Automatic)	The horizontal text alignment, using the culture text direction, is used to define how text will be horizontally aligned within its layout area.
Multiline	bool	If multi line is disabled then the new line control characters will be ignored and text will always be rendered in one line.
Text Color	Candera::Color	Text color.
Text Disabled Color	Candera::Color	Disabled text color.
Text Style	Candera::TextRendering::Style	The Style that will be used.
Background Image	Candera::Color	The image to set on the effect.
Arrow Image	Candera::Color	The increment/Decrement button arrow image.
Pressed Image	Candera::Image2D	The increment/Decrement button pressed image.
Arrow Disabled Image	Candera::Image2D	The increment/Decrement button disabled arrow image.

CoreText

Displays a given text like a label. It supports properties to set the Text, Text Style, Text Color. A Bindable Property is provided to set the Text by Data Binding. And Supports language translation.

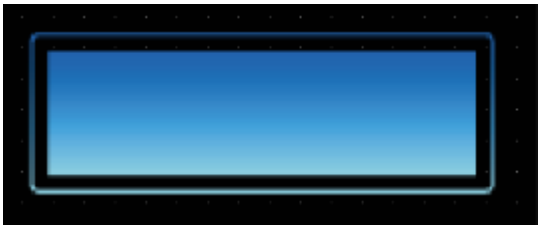


Properties

Property Name	Type	Description
Text	String	The Text that will be set to the Text Node.
Text Color	Candera::Color	The color of the text.
Text Style	Candera::TextRendering::Style	The Text that will be set to the Text Node.

CoreButton

Defines the basic states of a button and the actions executed when clicked. The button's appearance is determined by setting different images for each state. Additionally, text, icons, and other content can be placed inside the button, and spacing between the content and the button can be adjusted. However, the Core version does not include advanced visual effects or focus-related features.



Properties

Property Name	Type	Description
Enabled	bool	If this option is set to false, the button will be disabled and will no longer accept clicks.
Focusable	bool	If true, the control is focusable. Otherwise the control can not be focused.
Normal Image	9 Patch Image	The image used as the button image.
Pressed Image	9 Patch Image	The image used as the button image when the pressed state is applied.

Disabled Image	9 Patch Image	The button image displayed when the button is disabled.
Padding	Candera::Margin	The padding (margin) of the button. Used when content is attached.
Identifier	Candera::AbstractNodePointer	The identifier used when an event such as Click Event occurs.

CoreDigitalClock

Represents a digital clock, whereas the time-values (hours, minutes, seconds) are rendered as text. A data-bound value from the player can be used to represent the current time. The control offers properties to customize the clock, like showing the hours in 12h.



Properties

Property Name	Type	Description
Time	Variant	Specifies the type and the value to pass onto Process Value Behavior.
AM Text	String	The string of the AM label.
PM Text	String	The string of the PM label.
Show Seconds	bool	If true, seconds will be shown.
Show AMPM	bool	If ture, shows AM or PM depending on time.
Hours Text Style	Candera::TextRendering::Style	The style set for the “hour” text.
Minutes Colon Text Style	Candera::TextRendering::Style	The style applied to the separator between the “hour” and “minute” text.
Minutes Text Style	Candera::TextRendering::Style	The style set for the “minute” text.
Seconds Text Style	Candera::TextRendering::Style	The style set for the “second” text.
AMPM Text Style	Candera::TextRendering::Style	The style set for the AM/PM text.
Hours Color	Candera::Color	The color set for the “hour” text.

Minutes Color	Candera::Color	The color set for the “minute” text.
Seconds Color	Candera::Color	The color set for the “second” text.
AM PM Color	Candera::Color	The color set for the AM/PM text.

CoreProgressBar

Represents a bar that changes size according to the input value. It is possible to set background and bar images, with a 9-patch image recommended for the bar. In the Core version, animation for value changes is not included, focusing solely on simple progress display functionality.



Properties

Property Name	Type	Description
Value	Float	The current value of the Progress Bar.
Minimum	bool/Variant	Minimum value
Maximum	bool/Variant	Maximum value
Bar Image	Candera::Image2D	The image of the Progress Bar that will be scaled.
Background Image	Candera::Image2D	The background image of the Progress Bar.

CoreTextButton

Defines the basic behavior of a button and the actions executed when clicked. In the Core version, visual customization features are limited, and the button's appearance is determined by the specified image and text. Properties are provided to adjust the display and style of the text. The button's behavior when pressed can be easily configured using the EventHandler.



Properties

Property Name	Type	Description
Enabled	bool	If this option is set to false, the button will be disabled and will no longer accept clicks.
Focusable	bool	If true, the control is focusable. Otherwise the control can not be focused.
Normal Image	Candera::Image2D	The image displayed when the button is in its normal state.
Pressed Image	Candera::Image2D	The image displayed when the button is pressed.
Disabled Image	9 Patch Image	The image displayed when the button is disabled.
Text	String	Text to be displayed in the button.
Text Style	Candera::TextRendering::Style	Set the style of the text.
Text Enabled Color	Candera::Color	Text to be set in the text node.
Text Disabled Color	Candera::Color	Text color when disabled.

CoreCheckBox

A simple UI element that indicates the selection status. In the Core version, the normal and selected states can be configured using a combination of text and images. The behavior when the state of this control changes can be set using the EventHandler.
Note that features related to Focus and Blend effects are not included.



Properties

Property Name	Type	Description
Enabled	bool	If this option is set to false, the checkbox will not respond to clicks.
Focusable	bool	If true, the control is focusable. Otherwise the control can not be focused.

Value	bool/Variant	Specifies the initial selection state of the checkbox (true = selected, false = not selected).
Text	String	The text of the checkbox.
Text Style	Candera::TextRendering::Style	The style set for the text node.
Normal Image	Candera::Image2D	The checkbox image displayed when in the normal state.
Selected Image	Candera::Image2D	The checkbox image displayed when in the selected state.
Normal Disabled Image	Candera::Image2D	The checkbox image displayed when in the invalid state.
Selected Disabled Image	Candera::Image2D	The image of a selected checkbox in the invalid state.
Normal Text Color	Candera::Image2D	The text color when in the valid state.
Disabled Text Color	Candera::Image2D	The text color when in the invalid state.

CoreToggleButton

A simple UI element that toggles the button state. It is typically used to enable or disable specific functions. The structure is simple, allowing you to configure two images corresponding to the ON/OFF states of the toggle and their associated text. The behavior when the button is clicked can be set in the EventHandler tab. Note that features related to Focus and Blend effects are not included.



Properties

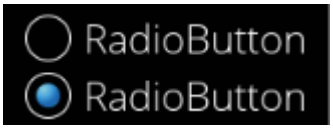
Property Name	**Type**	**Description**
Enabled	bool	If this option is set to disabled, the toggle button will be disabled and will no longer accept click operations.

Focusable	bool	If true, the control is focusable. Otherwise the control can not be focused.
Value	bool/Variant	Specify the initial state of the toggle (true = enabled, false = disabled).
First Text	String	The text that will be displayed when the toggle is set to On (the first toggle image).
Second Text	String	The text that will be displayed when the toggle is set to Off (the second toggle image).
Text Style	Candera::TextRendering::Style	The style set for the text node.
First Image	Candera::Image2D	The image that will be displayed when the toggle is set to On (the first toggle image).
Second Image	Candera::Image2D	The image displayed when the toggle is in the Off state (second toggle image).
First Disabled Image	Candera::Image2D	The image displayed when the toggle is in the On state when it is disabled.
Second Disabled Image	Candera::Image2D	The image displayed when the toggle is in the Off state when it is disabled.
Text Enabled Color	Candera::Image2D	The text color when it is enabled.
Text Disabled Color	Candera::Image2D	The text color when it is disabled.

CoreRadioButton

A simple UI element that allows the user to select one option from multiple choices. In the Core version, the normal and selected states can be configured using a combination of text and images. The behavior when the state of the CoreRadioButton control changes can be set using the EventHandler.

The functionality of the CoreRadioButton is similar to that of the CoreCheckBox, but it is designed to allow only one selection among multiple options.



Properties

Property Name	Type	Description
Enabled	bool	If you set this option to disabled, the radio buttons will not work and will not accept clicks.
Focusable	bool	If true, the control is focusable. Otherwise the control can not be focused.
Text	String	The text of the radio buttons.
TextStyle	Candera::TextRendering::Style	Set the style of the text.
Normal Image	Candera::Image2D	The image of the radio buttons displayed in the normal state.
Selected Image	Candera::Image2D	The image of the radio buttons displayed when selected.
Normal Disabled Image	Candera::Image2D	The image of the normal radio buttons in the disabled state.
Selected Disabled Image	Candera::Image2D	The image of the selected radio buttons in the disabled state.
Text Enabled Color	Candera::Image2D	The color of the text in the enabled state.
Text Disabled Color	Candera::Image2D	The color of the text in the disabled state.
Selected	bool/Variant	Specify the type and value to pass to the Process Value behavior.

CoreTextValue

Generates a formatted string by combining Value and Format Text. It supports formatting numbers and text in a manner similar to the C++ printf() function. The value can be set directly or modified through events. Events are provided through behaviors that send or transfer values.



Properties

Property Name	Type	Description
Value	Variant	The value to be formatted

Format Text	String	The format string which includes one format specifier (beginning with %). See C printf() function.
Style	Candera::TextRendering::Style	The style of the text to be used.
Text Color	Candera::Color	Specify the color of the text.

CoreSlider

This is a control where the value changes based on the position of the knob. The knob can be operated by dragging, and values can be set directly or modified through events (e.g., using the Send Value as Event Behavior).

In the Core version, focus and blend effect functionalities are not included, and the design is specialized for basic slider operations.



Properties

Property Name	Type	Description
Enabled	bool	If set to false it will not react on focus events and does not have to support different visualizations for the different states.
Focusable	bool	If true, the control is focusable. Otherwise the control can not be focused.
Value	Variant	The bindable value of the slider
Minimum	bool/Variant	Minimum Range
Maximum	bool/Variant	Maximum Range
Track Image	Candera::Image2D	The image used for the slider track.
Track Disabled Image	Candera::Image2D	The image used for the slider track when disabled.
Track Fill Image	Candera::Image2D	The image indicating the progress of the slider.
Track Disabled Fill Image	Candera::Image2D	The image indicating the progress of the slider when disabled.

Knob Normal Image	Candera::Image2D	The image used for the knob in its normal state.
Knob Pressed Image	Candera::Image2D	The image used for the knob when pressed.
Knob Disabled Image	Candera::Image2D	The image used for the knob when disabled.
Knob Center Margin	Candera::Margin	The center margin of the knob. Used to position the knob around the slider value. To avoid clipping issues in standalone sliders, ensure that the margin does not exceed the boundary.
Track Margin	Candera::Margin	The margin of the track. Ensures sufficient space for displaying the knob around the track.
Use Center	bool	If set to true, the knob's alignment is based on its predefined center value in the bitmap. If false, a calculated center value is used.

CoreScrollView

UI component that provides a scrollable area where arbitrary nodes (including controls) can be placed. If the attached node's size exceeds the layout size of the ScrollView, a scrollbar is automatically displayed.

In the Core version, node layout functionality is not included, and the component is specialized for displaying scrollable content. Therefore, automatic alignment features like those found in lists are not provided, but freely positioned controls and nodes can be made scrollable.



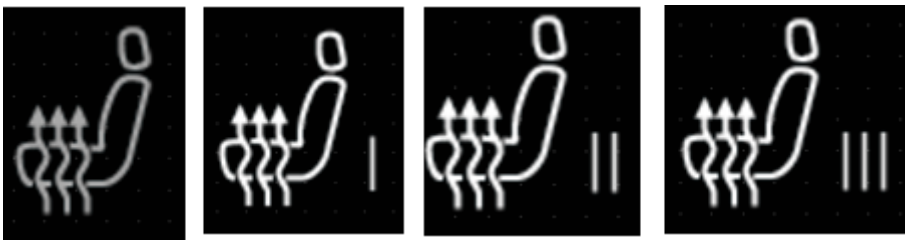
Properties

Property Name	Type	Description

Vertical Scrollbar Visibility	ScrollbarVisibility::Enum (Automatic, Collapsed, Visible)	Configure the visibility of the scrollbar. By default it is set to Automatic which will determine the visibility based on the actual content.
Horizontal Scrollbar Visibility	ScrollbarVisibility::Enum (Automatic, Collapsed, Visible)	Configure the visibility of the scrollbar. By default it is set to Automatic which will determine the visibility based on the actual content.

CoreMultiStateButton

A UI element that allows for the easy definition of buttons with multiple states. Each click changes the state, and once the maximum value is reached, it loops back to the initial state. The corresponding image is selected based on the button's state, and the displayed text can also be modified using the Text property. The behavior of the MultiStateButton can be set in the EventHandler, and actions corresponding to the current state can be executed using the Process Value Changed Event behavior.



Properties

Property Name	Type	Description
Focusable	bool	If true, the control is focusable. Otherwise the control can not be focused.
Current State	Variant	Current state of the MultiStateButton (starts with 0).
Maximum State	bool/Variant	Maximum value of MultiStateButton (starts with 0).

Revision #26
Created 2025-02-14 03:54:08 UTC by Tsuyoshi.Kato
Updated 2025-06-25 05:38:20 UTC by Tsuyoshi.Kato