

QSwipeCheckbox - Index

QSwipeCheckbox - Index

Buildversion: 2020-05-30-git6f418f30e4264323520de3ea56e1ce1dc5aed751

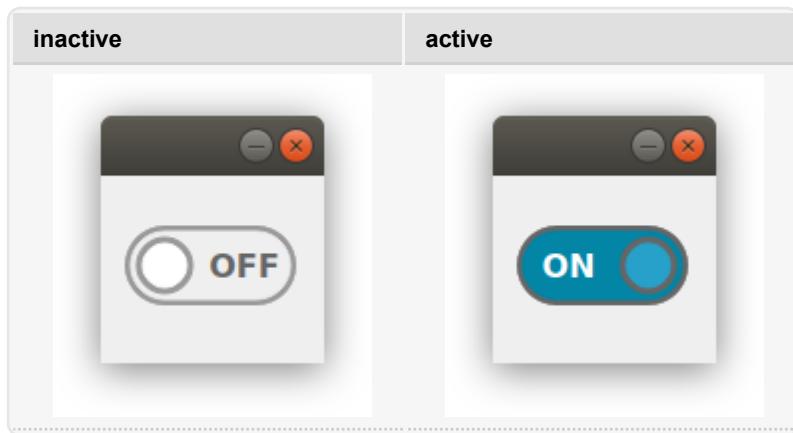
QSwipeCheckbox

Overview

Flippable-Style Checkbox for the Qt Framework supporting different styles and lots of customization options.

The following screenshots are just examples of the different styles. Each style can be individually customized!

"Classic" Style:



"iOS" Style:



"Material" Style:



"Windows" Style:



QSwipeCheckbox Class

QSwipeCheckbox

Buildversion: 2020-05-30-git6f418f30e4264323520de3ea56e1ce1dc5aed751

Flippable-Style Checkbox for the Qt Framework. [More...](#)

Header: #include <QSwipeCheckbox>

- [List of all members, including inherited members](#)

Public Types

```
enum ClickableArea { Full, Handle, Custom }  
enum DisplayStyle { Classic, iOS, Material, Windows }
```

Properties

- **activeBackgroundColor** : QColor
- **activeBorderColor** : QColor
- **activePaddingColor** : QColor
- **activeSwitchBorderColor** : QColor
- **activeSwitchKnobColor** : QColor
- **activeText** : QString
- **activeTextColor** : QColor
- **activeTextFont** : QFont
- **animationSpeed** : int
- **borderWidth** : int
- **checked** : bool
- **clickableArea** : ClickableArea
- **displayStyle** : DisplayStyle
- **enabled** : bool
- **inactiveBackgroundColor** : QColor
- **inactiveBorderColor** : QColor
- **inactivePaddingColor** : QColor
- **inactiveSwitchBorderColor** : QColor
- **inactiveSwitchKnobColor** : QColor
- **inactiveText** : QString
- **inactiveTextColor** : QColor
- **inactiveTextFont** : QFont
- **paddingWidth** : int
- **shadowAngle** : int
- **shadowColor** : QColor
- **shadowEnabled** : bool
- **shadowOpacity** : int
- **shadowSize** : int
- **size** : QSize
- **switchKnobSize** : int

Public Functions - 1/3

```
QString activeText() const  
QFont activeTextFont() const  
int animationSpeed() const  
int borderWidth() const  
bool checkState() const  
bool checked() const  
QPainterPath customClickableArea() const  
bool enabled() const  
QColor getActiveBackgroundColor() const  
QColor getActiveBorderColor() const  
QColor getActivePaddingColor() const  
QColor getActiveSwitchBorderColor() const  
QColor getActiveSwitchKnobColor() const  
QColor getActiveTextColor() const  
QSwipeCheckbox::ClickableArea getClickableArea() const  
QSwipeCheckbox::DisplayStyle getDisplayStyle() const  
QString getDisplayStyleName() const  
QColor getInactiveBackgroundColor() const  
QColor getInactiveBorderColor() const  
QColor getInactivePaddingColor() const  
QColor getInactiveSwitchBorderColor() const  
QColor getInactiveSwitchKnobColor() const  
QColor getInactiveTextColor() const
```

Public Functions - 2/3

```
int getShadowAngle() const
QColor getShadowColor() const
int getShadowSize() const
QSize getSize() const
QString inactiveText() const
QFont inactiveTextFont() const
bool isAnimationRunning() const
bool isChecked() const
bool isEnabled() const
int paddingWidth() const
void setActiveBackgroundColor(QColor color)
void setActiveBorderColor(QColor color)
void setActivePaddingColor(QColor color)
void setActiveSwitchBorderColor(QColor color)
void setActiveSwitchKnobColor(QColor color)
void setActiveText(QString str)
void setActiveTextColor(QColor color)
void setActiveTextFont(const QFont &font)
void setAnimationSpeed(int milliseconds = ...)
void setBackgroundColors(QColor active, QColor inactive)
void setBorderColors(QColor active, QColor inactive)
void setBorderWidth(int width = ...)
void setCheckState(bool state)
void setCheckState(Qt::CheckState state)
void setCheckState(int state)
void setChecked(bool checked)
void setClickableArea(QSwipeCheckbox::ClickableArea area)
void setClickableArea(QString area)
void setCustomClickableArea(QPainterPath &pPath)
void setDisplayStyle(QSwipeCheckbox::DisplayStyle style)
void setDisplayStyle(QString style)
void setEnabled(bool enabled)
void setHeight(int height)
```

Public Functions - 3/3

```
void setInactiveBackgroundColor(QColor color)
void setInactiveBorderColor(QColor color)
void setInactivePaddingColor(QColor color)
void setInactiveSwitchBorderColor(QColor color)
void setInactiveSwitchKnobColor(QColor color)
void setInactiveText(QString str)
void setInactiveTextColor(QColor color)
void setInactiveTextFont(const QFont &font)
void setPaddingColors(QColor active, QColor inactive)
void setPaddingWidth(int width = ...)
void setShadowAngle(int angle)
void setShadowColor(QColor color)
void setShadowEnabled(bool enabled = true)
void setShadowOpacity(int opacity = 60)
void setShadowSize(int size)
void setSize(const QSize &size)
void setSize(int width, int height)
void setSwitchBorderColors(QColor active, QColor inactive)
void setSwitchKnobColors(QColor active, QColor inactive)
void setSwitchKnobSize(int diameter = ...)
void setTextColors(QColor active, QColor inactive)
void setWidth(int width)
bool shadowEnabled() const
int shadowOpacity() const
int switchKnobSize() const
void visualizeCustomMouseRegion(bool enable = false, QColor color = QColor(255,0,0,127))
```

Reimplemented Public Functions

```
virtual QSize minimumSizeHint() const override  
virtual QSize sizeHint() const override
```

Signals

```
void sizeChanged(QSize)  
void stateChanged(bool state)  
void animationStarted(void)  
void animationFinished(void)
```

Detailed Description

Flippable-Style Checkbox for the Qt Framework supporting different styles and lots of customization options.

enum `QSwipeCheckbox::ClickableArea`

This enum specifies which part of the widget is supposed to react to mouse events.

Constant	Value	Description
<code>QSwipeCheckbox::Full</code>	0	The whole widget area is receiving mouse events.
<code>QSwipeCheckbox::Handle</code>	1	The widget's knob/handle is receiving mouse events.
<code>QSwipeCheckbox::Custom</code>	2	The area within the QPainterPath specified using the <code>setCustomClickableArea()</code> is receiving mouse events.

enum `QSwipeCheckbox::DisplayStyle`

This enum sets the widgets base style.

Constant	Value	Description
<code>QSwipeCheckbox::Classic</code>	0	The "Classic" <code>QSwipeCheckbox</code> style.
<code>QSwipeCheckbox::iOS</code>	1	The "iOS" <code>QSwipeCheckbox</code> style.
<code>QSwipeCheckbox::Material</code>	2	The "Material" <code>QSwipeCheckbox</code> style.
<code>QSwipeCheckbox::Windows</code>	3	The "Windows" <code>QSwipeCheckbox</code> style.

Property Documentation

activeBackgroundColor : QColor

This property holds the Background Color for the active state.

Access functions:

```
QColor getActiveBackgroundColor() const
```

```
void setActiveBackgroundColor(QColor color)
```

See also [inactiveBackgroundColor](#).

activeBorderColor : QColor

This property holds the Border Color for the active state.

Access functions:

```
QColor getActiveBorderColor() const
```

```
void setActiveBorderColor(QColor color)
```

activePaddingColor : QColor

This property holds the color of the padded area for the active state.

Access functions:

```
QColor getActivePaddingColor() const
```

```
void setActivePaddingColor(QColor color)
```

See also [QSwipeCheckbox::inactivePaddingColor](#).

activeSwitchBorderColor : QColor

This property holds the border color of the knob/handle for the active state.

Access functions:

```
QColor getActiveSwitchBorderColor() const  
void setActiveSwitchBorderColor(QColor color)
```

See also [QSwipeCheckbox::inactiveSwitchBorderColor](#).

activeSwitchKnobColor : QColor

This property holds the inner Knob/Handle Color for the active state.

Access functions:

```
QColor getActiveSwitchKnobColor() const  
void setActiveSwitchKnobColor(QColor color)
```

See also [inactiveSwitchKnobColor](#).

activeText : QString

This property holds the text for the active state.

Access functions:

```
QString activeText() const  
void  setActiveText(QString str)
```

See also [inactiveText](#).

activeTextColor : QColor

This property holds the Text Color for the active state.

Access functions:

```
QColor getActiveTextColor() const
```

```
void setActiveTextColor(QColor color)
```

See also [inactiveTextColor](#).

activeTextFont : QFont

This property holds the font style for the active state.

Access functions:

```
QFont activeTextFont() const
```

```
void  setActiveTextFont(const QFont &font)
```

See also [inactiveTextFont](#).

animationSpeed : int

This property holds the speed (duration) of the animation.

Setting this value to 0 disables the animation (instant switch flip).

Access functions:

```
int animationSpeed() const
```

```
void setAnimationSpeed(int milliseconds = ...)
```

borderWidth : int

This property holds the width of the [QSwipeCheckbox](#) border.

This property's default is 2.

Access functions:

```
int borderWidth() const
```

```
void setBorderWidth(int width = ...)
```

`checked : bool`

This property holds whether the `QSwipeCheckbox`' state is checked.

Note: Any `QSwipeCheckox` can only be either checked (`true`)or unchecked (`false`). A tri-state option is **not supported**.

Access functions:

```
bool isChecked() const  
void setChecked(bool checked)
```

Notifier signal:

```
void stateChanged(bool state)
```

`clickableArea : ClickableArea`

This property holds the `ClickableArea` setting of the widget.

Access functions:

```
QSwipeCheckbox::ClickableArea getClickableArea() const  
void setClickableArea(QSwipeCheckbox::ClickableArea area)  
void setClickableArea(QString area)
```

`displayStyle : DisplayStyle`

This property holds the `DisplayStyle` of the widget.

Access functions:

```
QSwipeCheckbox::DisplayStyle getDisplayStyle() const  
void setDisplayStyle(QSwipeCheckbox::DisplayStyle style)  
void setDisplayStyle(QString style)
```

`enabled : bool`

This property holds whether the `QSwipeCheckbox` is enabled.

This property's default is true.

Access functions:

```
bool isEnabled() const  
void setEnabled(bool enabled)
```

`inactiveBackgroundColor : QColor`

This property holds the Background Color for the inactive state.

Access functions:

```
QColor getInactiveBackgroundColor() const  
void setInactiveBackgroundColor(QColor color)
```

See also `activeBackgroundColor`.

`inactiveBorderColor : QColor`

This property holds the Border Color for the inactive state.

Access functions:

```
QColor getInactiveBorderColor() const  
void setInactiveBorderColor(QColor color)
```

`inactivePaddingColor : QColor`

This property holds the color of the padded area for the inactive state.

Access functions:

```
QColor getInactivePaddingColor() const  
void setInactivePaddingColor(QColor color)
```

See also `QSwipeCheckbox::activePaddingColor`.

inactiveSwitchBorderColor : QColor

This property holds the border color of the knob/handle for the inactive state.

Access functions:

```
QColor getInactiveSwitchBorderColor() const  
void setInactiveSwitchBorderColor(QColor color)
```

See also [QSwipeCheckbox::activeSwitchBorderColor](#).

inactiveSwitchKnobColor : QColor

This property holds the inner knob/handle color for the inactive state.

Access functions:

```
QColor getInactiveSwitchKnobColor() const  
void setInactiveSwitchKnobColor(QColor color)
```

See also [activeSwitchKnobColor](#).

inactiveText : QString

This property holds the text for the inactive state.

Access functions:

```
QString inactiveText() const  
void setInactiveText(QString str)
```

See also [activeText](#).

inactiveTextColor : QColor

This property holds the Text Color for the inactive state.

Access functions:

```
QColor getInactiveTextColor() const  
void setInactiveTextColor(QColor color)
```

See also [activeTextColor](#).

inactiveTextFont : QFont

This property holds the font style for the inactive state.

Access functions:

```
QFont inactiveTextFont() const  
void setInactiveTextFont(const QFont &font)
```

See also [activeTextFont](#).

paddingWidth : int

This property holds the width in pixels of the padded area.

Access functions:

```
int paddingWidth() const  
void setPaddingWidth(int width = ...)
```

See also [setPaddingWidth\(\)](#).

shadowAngle : int

This property holds the angle of the shadow-effect in degrees.

Access functions:

```
int getShadowAngle() const  
void setShadowAngle(int angle)
```

See also [QSwipeCheckbox::setShadowAngle\(\)](#).

shadowColor : QColor

This property holds the shadow color of the knob/handle for styles that support a shadow-effect.

Note: The alpha component of this color is ignored, as the opacity of the shadow-effect is set separately using the [setShadowOpacity\(\)](#)-function.

Access functions:

```
QColor getShadowColor() const  
void setShadowColor(QColor color)
```

See also [setShadowOpacity\(\)](#) and [shadowOpacity\(\)](#).

shadowEnabled : bool

This property holds whether the shadow-effect is enabled.

Access functions:

```
bool shadowEnabled() const  
void setShadowEnabled(bool enabled = true)
```

shadowOpacity : int

This property holds the opacity of the shadow-effect.

Access functions:

```
int shadowOpacity() const  
void setShadowOpacity(int opacity = 60)
```

See also [QSwipeCheckbox::setShadowOpacity\(\)](#).

shadowSize : int

This property holds the size of the shadow-effect.

Access functions:

```
int getShadowSize() const  
void setShadowSize(int size)
```

See also [setShadowSize\(\)](#).

size : QSize

This property holds the widget's size.

Access functions:

```
QSize getSize() const  
void setSize(int width, int height)  
void setSize(const QSize &size)
```

Notifier signal:

```
void sizeChanged(QSize)
```

```
switchKnobSize : int
```

This property holds the size (diameter) of the `QSwipeCheckbox` knob/handle.

Access functions:

```
int switchKnobSize() const  
void setSwitchKnobSize(int diameter = ...)
```

QString QSwipeCheckbox::activeText() const

This function returns the text string for the active state.

See also [setActiveText\(\)](#).

QFont QSwipeCheckbox::activeTextFont() const

This function returns the font of the text for the active state.

See also [setActiveTextFont\(\)](#).

int QSwipeCheckbox::animationSpeed() const

Returns the speed (duration) of the animation.

See also [setAnimationSpeed\(\)](#).

int QSwipeCheckbox::borderWidth() const

This function returns the border width in pixels.

See also [setBorderWidth\(\)](#).

bool QSwipeCheckbox::checkState() const

This function returns the widget's check state.

See also [setCheckState\(\)](#) and [isChecked\(\)](#).

bool QSwipeCheckbox::checked() const

This function returns the widget's check state.

See also [setChecked\(\)](#).

`QPainterPath QSwipeCheckbox::customClickableArea() const`

This function returns the QPainterPath which contains the area within the widget that reacts to mouse events.

See also `setCustomClickableArea()`.

`bool QSwipeCheckbox::enabled() const`

This function returns whether the widget is enabled.

See also `setEnabled()`.

`QColor QSwipeCheckbox::getActiveBackgroundColor() const`

This function returns the background color for the active state.

`QColor QSwipeCheckbox::getActiveBorderColor() const`

This function returns the border color for the active state.

`QColor QSwipeCheckbox::getActivePaddingColor() const`

This function returns the color of the padded area for the active state.

`QColor QSwipeCheckbox::getActiveSwitchBorderColor() const`

This function returns the border color of the knob/handle for the active state.

`QColor QSwipeCheckbox::getActiveSwitchKnobColor() const`

This function returns the color of the knob/handle for the active state.

`QColor QSwipeCheckbox::getActiveTextColor() const`

This function returns the text color for the active state.

QSwipeCheckbox::ClickableArea **QSwipeCheckbox::getClickableArea()**
const

This function returns a QString containing the current setting of which part of the widget reacts to mouse events.

See also [getClickableArea\(\)](#) and [setClickableArea\(\)](#).

QSwipeCheckbox::DisplayStyle **QSwipeCheckbox::getDisplayStyle()**
const

This function returns current the [DisplayStyle](#).

QString **QSwipeCheckbox::getDisplayStyleName()** **const**

This function returns a QString-value containing the [DisplayStyle](#) name.

QColor **QSwipeCheckbox::getInactiveBackgroundColor()** **const**

This function returns the background color for the inactive state.

QColor **QSwipeCheckbox::getInactiveBorderColor()** **const**

This function returns the border color for the inactive state.

QColor **QSwipeCheckbox::getInactivePaddingColor()** **const**

This function returns the color of the padded area for the inactive state.

QColor **QSwipeCheckbox::getInactiveSwitchBorderColor()** **const**

This function returns the border color of the knob/handle for the inactive state.

QColor **QSwipeCheckbox::getInactiveSwitchKnobColor()** **const**

This function returns the color of the knob/handle for the inactive state.

QColor QSwipeCheckbox::getInactiveTextColor() const

This function returns the text color for the inactive state.

int QSwipeCheckbox::getShadowAngle() const

This function returns the angle of the shadow effect.

See also [setShadowAngle\(int\)](#).

QColor QSwipeCheckbox::getShadowColor() const

This function returns the color of the shadow-effect.

int QSwipeCheckbox::getShadowSize() const

This function returns the shadow size.

See also [setShadowSize\(int\)](#).

QSize QSwipeCheckbox::getSize() const

This function returns the size of the Widget.

QString QSwipeCheckbox::inactiveText() const

This function returns the text string for the inactive state.

See also [setInactiveText\(\)](#).

QFont QSwipeCheckbox::inactiveTextFont() const

This function returns the font of the text for the inactive state.

See also [setInactiveTextFont\(\)](#).

```
bool QSwipeCheckbox::isAnimationRunning() const
```

This function returns whether there are currently animations running.

```
QSize QSwipeCheckbox::minimumSizeHint() [override virtual]  
const
```

This function returns the widget's minimum size.

See also [sizeHint\(\)](#).

```
int QSwipeCheckbox::paddingWidth() const
```

This function returns the padding width in pixels.

See also [setPaddingWidth\(\)](#).

```
void QSwipeCheckbox:: setActiveBackgroundColor(QColor color)
```

This function sets the background *color* for the active state.

```
void QSwipeCheckbox:: setActiveBorderColor(QColor color)
```

This function sets the border *color* for the active state.

```
void QSwipeCheckbox:: setActivePaddingColor(QColor color)
```

This function sets the *color* of the padding area for the active state.

```
void QSwipeCheckbox:: setActiveSwitchBorderColor(QColor color)
```

This function sets the border *color* of the knob/handle for the active state.

```
void QSwipeCheckbox:: setActiveSwitchKnobColor(QColor color)
```

This function sets the *color* of the knob/handle for the active state.

```
void QSwipeCheckbox:: setActiveText(QString str)
```

This function sets the text string *str* for the active state.

See also [activeText\(\)](#).

```
void QSwipeCheckbox:: setActiveTextColor(QColor color)
```

This function sets the text *color* for the active state.

```
void QSwipeCheckbox:: setActiveTextFont(const QFont &font)
```

This function sets the *font* of the text for the active state.

See also [activeTextFont\(\)](#).

```
void QSwipeCheckbox:: setAnimationSpeed(int milliseconds = ...)
```

This function sets the animation speed (duration) in *milliseconds*.

See also [animationSpeed\(\)](#).

```
void QSwipeCheckbox:: setBackgroundColors(QColor active, QColor inactive)
```

This function is provided for convenience to combine [setActiveBackgroundColor\(\)](#) and [setInactiveBackgroundColor\(\)](#).

See also [setActiveBackgroundColor\(\)](#) and [setInactiveBackgroundColor\(\)](#).

```
void QSwipeCheckbox:: setBorderColors(QColor active, QColor inactive)
```

This function is provided for convenience to combine [setActiveBorderColor\(\)](#) and [setInactiveBorderColor\(\)](#).

See also [setActiveBorderColor\(\)](#) and [setInactiveBorderColor\(\)](#).

```
void QSwipeCheckbox::setBorderWidth(int width = ...)
```

This function sets the border *width* in pixels.

See also [borderWidth\(\)](#).

```
void QSwipeCheckbox::setCheckState(bool state)
```

This function sets the *state* of the widget.

See also [checkState\(\)](#).

```
void QSwipeCheckbox::setCheckState(Qt::CheckState state)
```

This function overloads [setCheckState\(bool\)](#);

```
void QSwipeCheckbox::setCheckState(int state)
```

This function overloads [setCheckState\(bool\)](#);

```
void QSwipeCheckbox::setChecked(bool checked)
```

This function sets whether the widget's state is *checked*.

See also [checked\(\)](#).

```
void  
QSwipeCheckbox::setClickableArea(QSwipeCheckbox::ClickableArea  
area)
```

This function sets the *area* that reacts to mouse events for the widget.

See also [ClickableArea](#) and [setClickableArea\(QString\)](#).

```
void QSwipeCheckbox::setClickableArea(QString area)
```

This function overloads [QSwipeCheckbox::setClickableArea\(QSwipeCheckbox::ClickableArea\)](#).

This function sets the *area* by a string value that reacts to mouse events for the widget.

See also [setClickableArea\(QSwipeCheckbox::ClickableArea\)](#).

```
void QSwipeCheckbox::setCustomClickableArea(QPainterPath &ppath)
```

This function sets the QPainterPath (*ppath*) containing the area within the widget supposed to react to mouse events.

See also [customClickableArea\(\)](#).

```
void QSwipeCheckbox::setDisplayStyle(QSwipeCheckbox::DisplayStyle  
style)
```

This function sets the [DisplayStyle](#) *style* for the widget.

See also [DisplayStyle](#).

```
void QSwipeCheckbox::setDisplayStyle(QString style)
```

This function overloads [QSwipeCheckbox::setDisplayStyle\(DisplayStyle style\)](#).

This function sets the [DisplayStyle](#) *style* for the widget.

See also [DisplayStyle](#) and [QSwipeCheckbox::setDisplayStyle\(QSwipeCheckbox::DisplayStyle style\)](#).

```
void QSwipeCheckbox::setEnabled(bool enabled)
```

This function sets whether the widget is *enabled*.

See also [enabled\(\)](#).

```
void QSwipeCheckbox::setHeight(int height)
```

This function sets the *height* of the widget.

```
void QSwipeCheckbox::setInactiveBackgroundColor(QColor color)
```

This function sets the background *color* for the inactive state.

```
void QSwipeCheckbox::setInactiveBorderColor(QColor color)
```

This function sets the border *color* for the inactive state.

```
void QSwipeCheckbox::setInactivePaddingColor(QColor color)
```

This function sets the *color* of the padding area for the inactive state.

```
void QSwipeCheckbox::setInactiveSwitchBorderColor(QColor color)
```

This function sets the border *color* of the knob/handle for the inactive state.

```
void QSwipeCheckbox::setInactiveSwitchKnobColor(QColor color)
```

This function sets the *color* of the knob/handle for the inactive state.

```
void QSwipeCheckbox::setInactiveText(QString str)
```

This function sets the text string *str* for the inactive state.

See also [inactiveText\(\)](#).

```
void QSwipeCheckbox::setInactiveTextColor(QColor color)
```

This function sets the text *color* for the inactive state.

```
void QSwipeCheckbox::setInactiveTextFont(const QFont &font)
```

This function sets the *font* of the text for the inactive state.

See also [inactiveTextFont\(\)](#).

```
void QSwipeCheckbox::setPaddingColors(QColor active, QColor inactive)
```

This function is provided for convenience to combine [setActivePaddingColor\(\)](#) and [setInactivePaddingColor\(\)](#).

See also [setActivePaddingColor\(\)](#) and [setInactivePaddingColor\(\)](#).

```
void QSwipeCheckbox::setPaddingWidth(int width = ...)
```

This function sets the padding *width* in pixels.

See also [paddingWidth\(\)](#).

```
void QSwipeCheckbox::setShadowAngle(int angle)
```

This function sets the shadow *angle* in degrees (range from 0 to 360) where 0 equals the 12-o'clock position.

See also [getShadowAngle\(\)](#).

```
void QSwipeCheckbox::setShadowColor(QColor color)
```

This function sets the *color* of the shadow-effect.

Note: The alpha component of this color is ignored, as the opacity of the shadow-effect is set separately using the [setShadowOpacity\(\)](#)-function.

Note: Setter function for property [shadowColor](#).

```
void QSwipeCheckbox::setShadowEnabled(bool enabled = true)
```

This function sets whether the shadow effect is *enabled*.

See also [shadowEnabled\(\)](#).

```
void QSwipeCheckbox::setShadowOpacity(int opacity = 60)
```

This function sets the shadow *opacity* (range from 0 *transparent* to 100 *opaque*).

See also [shadowOpacity\(\)](#).

```
void QSwipeCheckbox::setShadowSize(int size)
```

This function sets the shadow *size* in pixels.

See also [getShadowSize\(\)](#).

```
void QSwipeCheckbox::setSize(const QSize &size)
```

Sets the desired widget *size*.

See also [getSize\(\)](#).

```
void QSwipeCheckbox::setSize(int width, int height)
```

This function overloads `setSize(QSize)`.

This function sets the *width* and *height* of the widget.

```
void QSwipeCheckbox::setSwitchBorderColors(QColor active, QColor inactive)
```

This function is provided for convenience to combine [setActiveSwitchBorderColor\(\)](#) and [setInactiveSwitchBorderColor\(\)](#).

See also [setActiveSwitchBorderColor\(\)](#) and [setInactiveSwitchBorderColor\(\)](#).

```
void QSwipeCheckbox::setSwitchKnobColors(QColor active, QColor inactive)
```

This function is provided for convenience to combine `setActiveSwitchKnobColor()` and `setInactiveSwitchKnobColor()`.

See also `setActiveSwitchKnobColor()` and `setInactiveSwitchKnobColor()`.

```
void QSwipeCheckbox::setSwitchKnobSize(int diameter = ...)
```

This function returns the knob/handle's size (*diameter*) in pixels.

See also `switchKnobSize()`.

```
void QSwipeCheckbox::setTextColors(QColor active, QColor inactive)
```

This function is provided for convenience to combine `setActiveTextColor()` and `setInactiveTextColor()`.

See also `setActiveTextColor()` and `setInactiveTextColor()`.

```
void QSwipeCheckbox::setWidth(int width)
```

This function sets the *width* of the widget.

```
bool QSwipeCheckbox::shadowEnabled() const
```

This function returns whether the shadow effect is enabled.

See also `setShadowEnabled(bool)`.

```
int QSwipeCheckbox::shadowOpacity() const
```

This function returns the shadow opacity.

See also `setShadowOpacity(int)`.

```
QSize QSwipeCheckbox::sizeHint() const [override virtual]
```

Returns the recommended size for the widget.

See also [size](#), [minimumSize](#), and [minimumSizeHint](#).

```
int QSwipeCheckbox::switchKnobSize() const
```

This function returns the knob/handle's size in pixels.

See also [setSwitchKnobSize](#)().

```
void QSwipeCheckbox::visualizeCustomMouseRegion(bool enable = false, QColor color = QColor(255,0,0,127))
```

This function sets whether the rendering of a custom region (set via [setClickableArea](#)()) is enabled (*enable* = true) for debugging purposes and sets the *color* in which the area is being highlighted.

Note: This function is purposely included in **debug-releases** only.

See also [setClickableArea](#)().

Properties / DisplayStyle

Properties / DisplayStyle

Buildversion: 2020-05-30-git6f418f30e4264323520de3ea56e1ce1dc5aed751

Property	QSwipeCheckbox::DisplayStyle			
	Classic	iOS	Material	Windows
QSwipeCheckbox::activeBackgroundColor	X	X	X	X
QSwipeCheckbox::inactiveBackgroundColor	X	X	X	X
QSwipeCheckbox::activeTextColor	X	--	--	X
QSwipeCheckbox::inactiveTextColor	X	--	--	X
QSwipeCheckbox::activeBorderColor	X	X	--	X
QSwipeCheckbox::inactiveBorderColor	X	X	--	X
QSwipeCheckbox::activeSwitchBorderColor	X	--	--	--
QSwipeCheckbox::inactiveSwitchBorderColor	X	--	--	--
QSwipeCheckbox::activeSwitchKnobColor	X	X	X	X
QSwipeCheckbox::inactiveSwitchKnobColor	X	X	X	X
QSwipeCheckbox::activePaddingColor	--	--	--	X
QSwipeCheckbox::inactivePaddingColor	--	--	--	X
QSwipeCheckbox::shadowColor	--	X	X	--
QSwipeCheckbox::activeText	X	--	--	X
QSwipeCheckbox::inactiveText	X	--	--	X
QSwipeCheckbox::borderWidth	X	X	X	X
QSwipeCheckbox::paddingWidth	--	--	X	X
QSwipeCheckbox::switchKnobSize	X	-- 1	-- 2	X ³
QSwipeCheckbox::shadowOpacity	--	X	X	--
QSwipeCheckbox::shadowAngle	--	X	X	--
QSwipeCheckbox::shadowSize	--	X	X	--
QSwipeCheckbox::shadowEnabled	--	X	X	--
QSwipeCheckbox::activeTextFont	X	--	--	X
QSwipeCheckbox::inactiveTextFont	X	--	--	X

Remarks

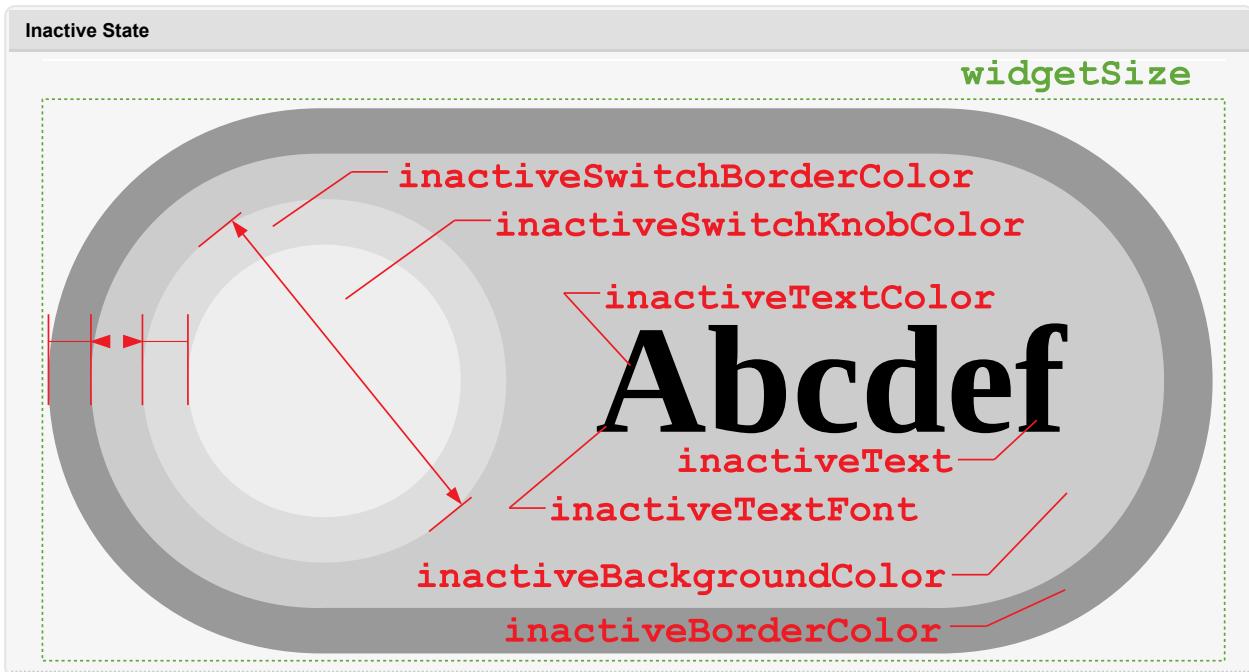
- 1 The size (diameter) of the knob/handle is the same as the height of the widget minus the *border width* set using the `QSwipeCheckbox::setBorderWidth()` function.
 - 2 The size (diameter) of the knob/handle is the same as the height of the widget minus the *border width* set using the `QSwipeCheckbox::setBorderWidth()` function.
 - 3 The `QSwipeCheckbox::switchKnobSize()` function sets not the diameter, but the width of the knob/handle when using `QSwipeCheckbox::Windows`.
-

Classic-style Properties

Classic-style Properties

Buildversion: 2020-05-30-git6f418f30e4264323520de3ea56e1ce1dc5aed751

- ↳ activeBackgroundColor
- ↳ inactiveBackgroundColor
- ↳ activeTextColor
- ↳ inactiveTextColor
- ↳ activeBorderColor
- ↳ inactiveBorderColor
- ↳ activeSwitchBorderColor
- ↳ inactiveSwitchBorderColor
- ↳ activeSwitchKnobColor
- ↳ inactiveSwitchKnobColor
- ↳ activeText
- ↳ inactiveText
- ↳ borderWidth
- ↳ switchKnobSize
- ↳ activeTextFont
- ↳ inactiveTextFont



iOS-style Properties

iOS-style Properties

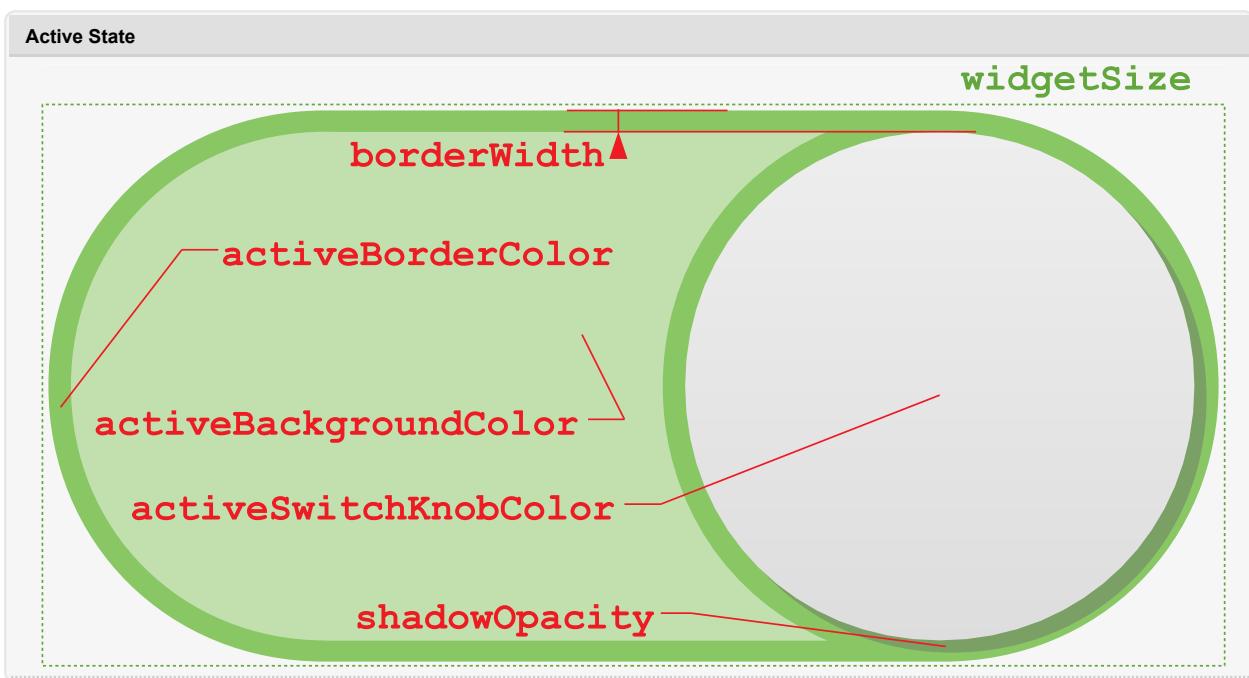
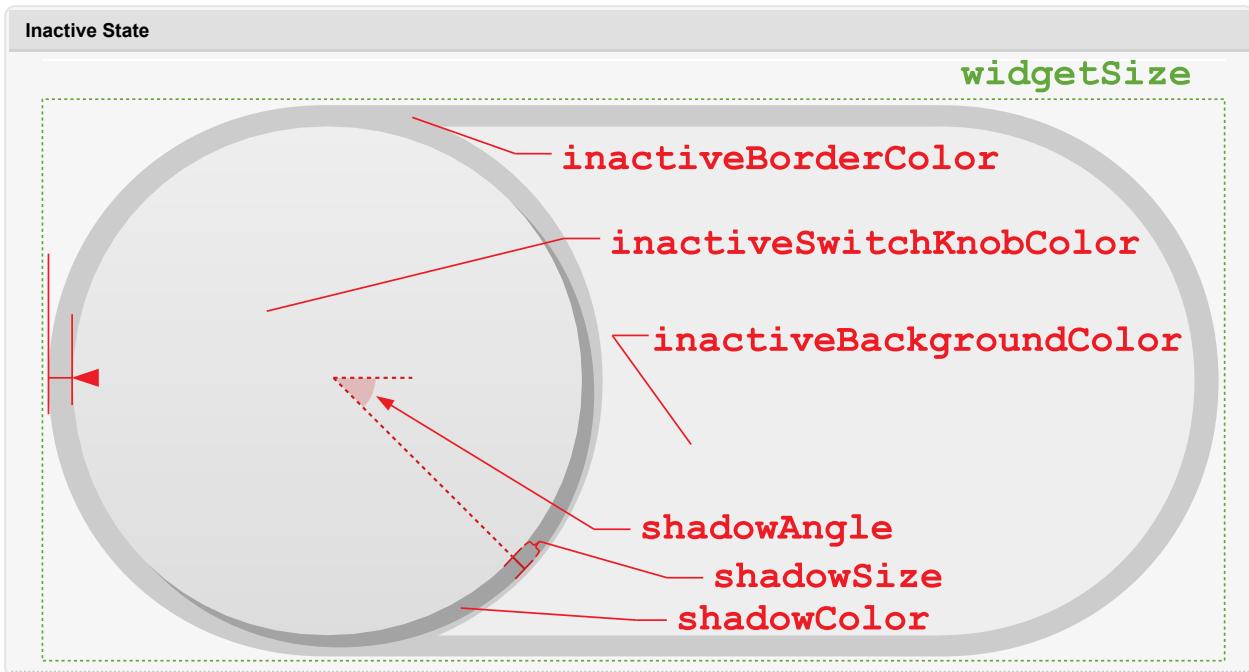
Buildversion: 2020-05-30-git6f418f30e4264323520de3ea56e1ce1dc5aed751

- ↳ activeBackgroundColor
- ↳ inactiveBackgroundColor
- ↳ activeBorderColor
- ↳ inactiveBorderColor
- ↳ activeSwitchKnobColor
- ↳ inactiveSwitchKnobColor
- ↳ shadowColor
- ↳ borderWidth
- ↳ shadowOpacity
- ↳ shadowAngle
- ↳ shadowSize
- ↳ shadowEnabled

Note: The gradient on the knob/handle is defined by setting a single color only (`activeSwitchKnobColor` and `inactiveSwitchKnobColor` respectively). The gradient colors are then internally calculated using a *lighter tone by 4%* for the *top* gradient color and a *darker tone by 4%* for the *bottom* gradient color.

Note: The gradient is enabled or disabled alongside the shadow-effect. Setting `shadowEnabled` therefore simultaneously sets whether the gradient is applied.

Warning: For the full iOS-like experience, there is a subtle color-gradient and a shadow-effect applied to the `QSwipeCheckbox` widget. Please be aware that the rendering of the widget may become slow on some devices using these effects enabled by the `setShadowEnabled(true)` function. However, even with `setShadowEnabled(false)` you will still get a very convincing `QSwipeCheckbox`!



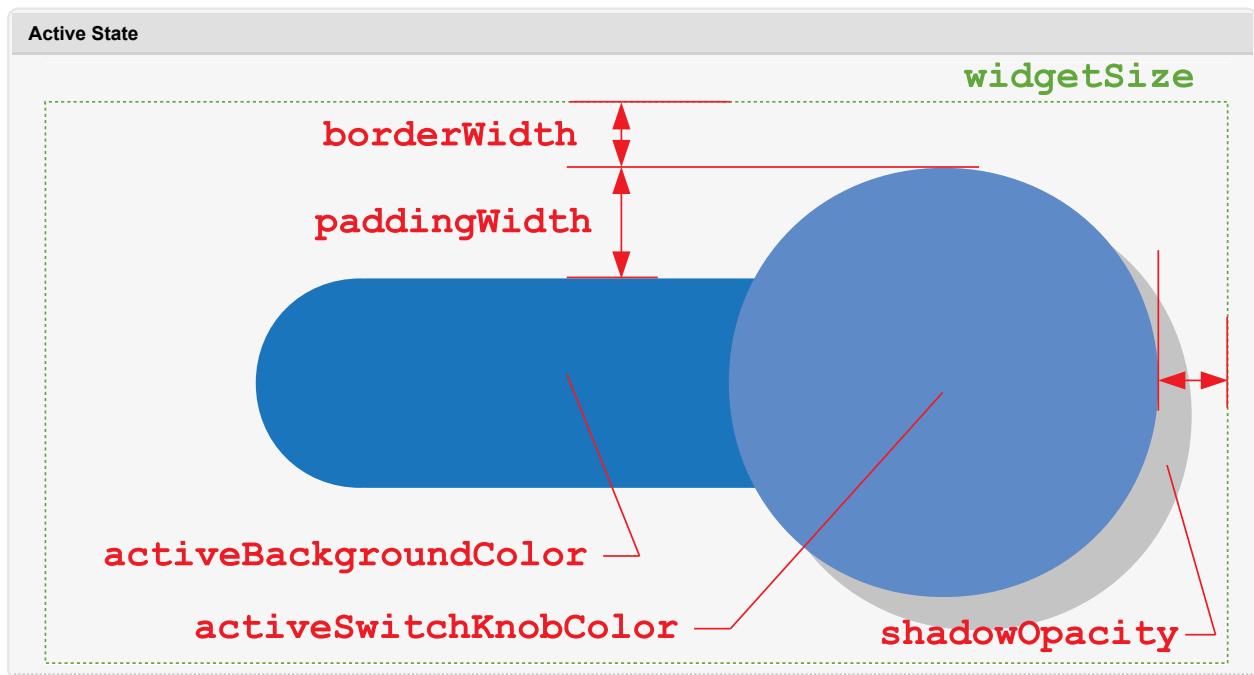
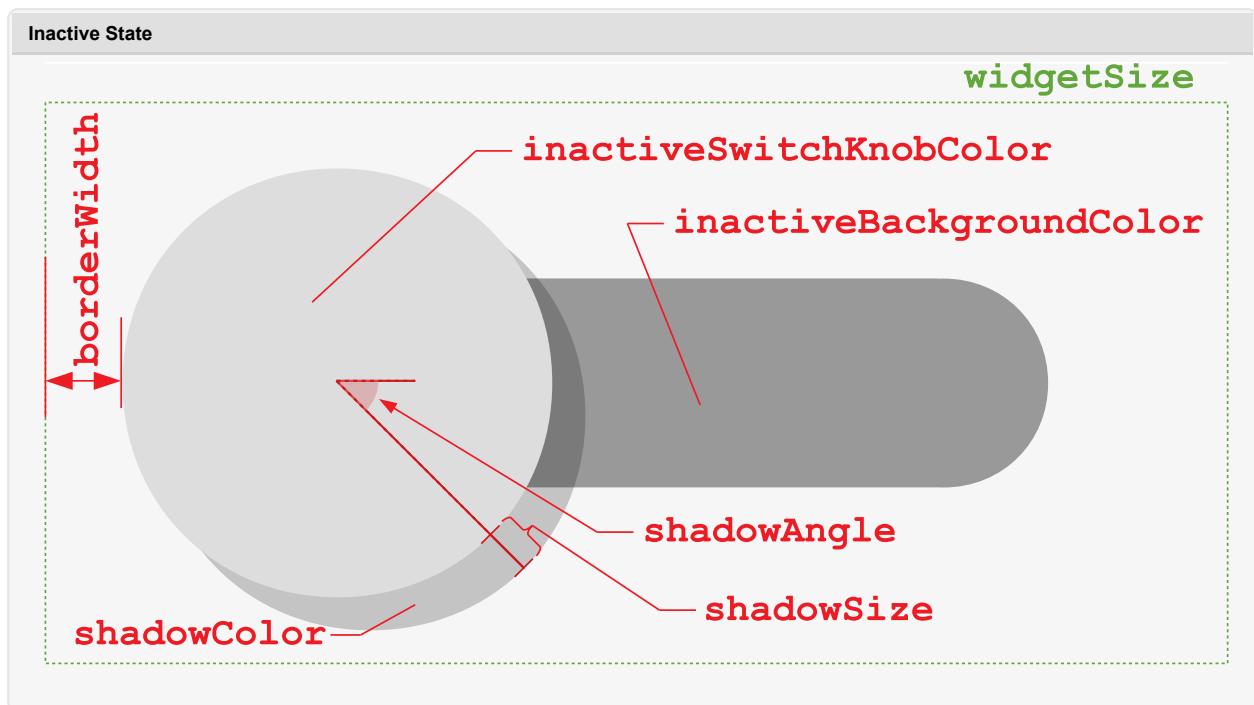
Material-style Properties

Material-style Properties

Buildversion: 2020-05-30-git6f418f30e4264323520de3ea56e1ce1dc5aed751

- ↳ [activeBackgroundColor](#)
- ↳ [inactiveBackgroundColor](#)
- ↳ [activeSwitchKnobColor](#)
- ↳ [inactiveSwitchKnobColor](#)
- ↳ [shadowColor](#)
- ↳ [borderWidth](#)
- ↳ [paddingWidth](#)
- ↳ [shadowOpacity](#)
- ↳ [shadowAngle](#)
- ↳ [shadowSize](#)
- ↳ [shadowEnabled](#)

Warning: For the full Material-style experience, there is a shadow-effect applied to the [QSwipeCheckbox](#) widget. Please be aware that the rendering of the widget may become slow on some devices using these effects enabled by the `setShadowEnabled(true)` function. However, even with `setShadowEnabled(false)` you will still get a very convincing [QSwipeCheckbox](#)!



Windows-style Properties

Windows-style Properties

Buildversion: 2020-05-30-git6f418f30e4264323520de3ea56e1ce1dc5aed751

- ↳ activeBackgroundColor
- ↳ inactiveBackgroundColor
- ↳ activeTextColor
- ↳ inactiveTextColor
- ↳ activeBorderColor
- ↳ inactiveBorderColor
- ↳ activeSwitchKnobColor
- ↳ inactiveSwitchKnobColor
- ↳ activePaddingColor
- ↳ inactivePaddingColor
- ↳ activeText
- ↳ inactiveText
- ↳ borderWidth
- ↳ paddingWidth
- ↳ switchKnobSize
- ↳ activeTextFont
- ↳ inactiveTextFont

