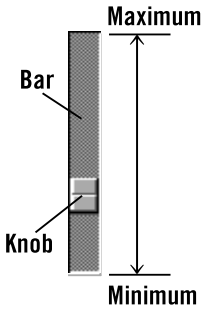
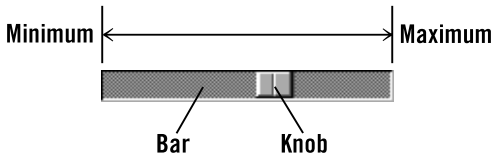
NSSlider

Inherits From:	NSControl : NSView : NSResponder : NSObject
Conforms To:	NSCoding (from NSResponder) NSObject (from NSObject)
Declared In:	AppKit/NSSlider.h

Class Description

An NSSlider has a handle, or "knob," that can be dragged in a groove, or "bar." The knob's position in the bar represents a number between a minimum and a maximum. If the slider is vertical, its minimum is at its bottom; if it is horizontal, its minimum is at its left.





The minimum and maximum can be obtained with the **minValue** and **maxValue** methods, and set with the **setMinValue:** and **setMaxValue:** methods. To read the value represented by the current position of the knob, you use an NSControl method like **floatValue**; conversely, to send a value to the slider, you use a NSControl method like **setFloatValue:**.

By default, an NSSlider is a continuous NSControl: while the user drags the slider's knob, the slider sends its action message continuously. If, instead, you want the slider to reserve its action message until the mouse is released, invoke **setContinuous:** (an NSControl method) with an argument of NO.

In its bar, an NSSlider can display an image, a title, or both. The title can be drawn in any color and any font. However, since a title in the bar may be obscured by the slider knob, you will more often label a slider by placing an NSTextField near it.

Like most NSControls, NSSlider relies heavily on a related cell class, NSSliderCell. For more information, see the NSSliderCell class specification.

Method Types

Asking about the slider's appearance	– image – knobThickness – isVertical
Changing the slider's appearance	– setImage: – setKnobThickness:
Asking about the slider's title	– title – titleCell – titleColor – titleFont
Changing the slider's title	– setTitle: – setTitleCell: – setTitleColor: – setTitleFont:
Asking about the value limits	– maxValue – minValue
Changing the value limits	– setMaxValue – setMinValue
Handling mouse-down events	– acceptsFirstMouse:

Instance Methods

acceptsFirstMouse:

– (BOOL)**acceptsFirstMouse:**(NSEvent *)*mouseDownEvent*

Returns YES by default, so that a single mouse-down event can simultaneously activate the window and take hold of the slider's knob.

If you want the slider to wait for its own mouse-down event, you must override this method.

image

– (NSImage *)**image**

Returns the image that the slider displays in its bar, or **nil** if no image has been set.

See also: – **setImage:**

isVertical

– (int)**isVertical**

Returns 1 if the slider is vertical, 0 if it's horizontal, and –1 if the orientation can't be determined (for example, if the slider hasn't been displayed yet). A slider is defined as vertical if its height is greater than its width.

knobThickness

– (float)**knobThickness**

Returns the knob's thickness, in pixels. The thickness is defined to be the extent of the knob along the long dimension of the bar. In a vertical slider, then, a knob's thickness is its height; in a horizontal slider, a knob's thickness is its width.

See also: – **setKnobThickness:**

maxValue

– (double)**maxValue**

Returns the maximum value that the slider can send to its target. A horizontal slider sends its maximum value when the knob is at the right end of the bar; a vertical slider sends it when the knob is at the top.

See also: – **setMaxValue:**

minValue

– (double)**minValue**

Returns the minimum value that the slider can send to its target. A vertical slider sends its minimum value when its knob is at the bottom; a horizontal slider, when its knob is all the way to the left.

See also: – **setMinValue:**

setImage:

– (void)**setImage:**(NSImage *)*barImage*

Sets the image that the slider displays in the bar behind its knob. The slider may scale and distort *barImage* to fit inside the bar.

The knob may cover part of the image. If you want the image to be visible all the time, you're better off placing it near the slider.

See also: – **setImage:**

setKnobThickness:

– (void)**setKnobThickness:**(float)*thickness*

Lets you set the knob's thickness, measured in pixels. The thickness is defined to be the extent of the knob along the long dimension of the bar. In a vertical slider, then, a knob's thickness is its height; in a horizontal slider, a knob's thickness is its width.

See also: – **knobThickness**

setMaxValue:

– (void)**setMaxValue:**(double)*maxValue*

Sets the maximum value that the slider can send to its target—the value that a horizontal slider sends when its knob is all the way to the right, or that a vertical slider sends when its knob is at the top.

See also: – **maxValue**

setMinValue:

– (void)**setMinValue:**(double)*minValue*

Sets the minimum value that the slider can send to its target. A horizontal slider sends its minimum value when its knob is all the way to the left; a vertical slider sends its minimum value when its knob is at the bottom.

See also: – **minValue**

setTitle:

– (void)**setTitle:**(NSString *)*barTitle*

Sets the title that the slider displays in the bar behind its knob.

The knob may cover part or all of the title. If you want the title to be visible all of the time, you're better off placing a label near the slider.

See also: – **title**

setTitleCell:

– (void)**setTitleCell:**(NSCell *)*titleCell*

Sets the cell used to draw the slider's title. You only need to invoke this method if the default title cell, NSTextFieldCell, doesn't suit your needs—that is, if you want to display the title in a manner that NSTextFieldCell doesn't permit. When you do choose to override the default, *titleCell* should be an instance of a subclass of TextFieldCell.

See also: – **titleCell**

setTitleColor:

– (void)**setTitleColor:**(NSColor *)*color*

Sets the color used to draw the slider's title.

See also: – **titleColor**

setTitleFont:

– (void)**setTitleFont:**(NSFont *)*font*

Sets the font used to draw the slider's title.

See also: – **titleFont**

title

– (NSString *)**title**

Returns the slider's title. The default title is the empty string ("").

See also: – **setTitle:**

titleCell

– (id)**titleCell**

Returns the cell used to draw the title. The default is an NSTextFieldCell.

See also: – **setTitleCell:**

titleColor

– (NSColor *)**titleColor**

Returns the color used to draw the slider's title. The default color is NSColor's **controlTextColor**.

See also: – **setTitleColor:**

titleFont

– (NSFont *)**titleFont**

Returns the font used to draw the slider's title.

See also: – **setTitleColor:**