init

Determining component sizes calcCellSize:inRect:
getKnobRect:flipped:

Setting value limits setMinValue:
minValue
setMaxValue:
maxValue

Setting values setDoubleValue:
doubleValue
setFloatValue:
floatValue
setIntValue:
intValue
setStringValue:
stringValue

Modifying a SliderCell's appearance
setKnobThickness:
knobThickness
setImage:
image
setTitle:
setTitleNoCopy:
title
setTitleCell:
titleCell
setTitleFont:
titleFont
setTitleColor:
titleColor
setTitleGray:
titleGray
isOpaque
isVertical

altIncrementValue
Tracking the mouse+ prefersTrackingUntilMouseUp
trackMouse:inRect:ofView:
startTrackingAt:inView:
continueTracking:at:inView:
stopTracking:at:inView:mouseIsUp:
Archiving read:
write:
awake

(double)altIncrementValue

Returns the amount that the SliderCell will alter its value when the user drags the knob one pixel w
held down. If the Alternate-dragging feature isn't enabled, this method returns 1.0.

setAltIncrementValue:

awake

Retrieves the system images used to draw SliderCell knobs, and returns self. This message is sent
you never send it yourself.

read:

calcCellSize:(NXSize *)theSize inRect:(const NXRect *)aRect

Returns self, and by reference in theSize the minimum width and height needed to draw the Slider
too small to fit the knob and bezel, the width and height of theSize are set to 0.0.

If the SliderCell hasn't had its tracking rectangle set, this method will determine from aRect wheth
be vertical or horizontal, and will set a vertical SliderCell's height to aRect->size.height, a horizon
aRect->size.width, and the other dimension of either type to the minimum SliderCell breadth.

If you draw your own knob on the SliderCell and that knob is not the same size as a standard Slide
draw the SliderCell itself differently, you should override this method to take your knob's dimensi
must also override getKnobRect:flipped: and drawKnob:.

getKnobRect:flipped:, drawKnob:

(double)doubleValue

Returns the value of the SliderCell as a double-precision floating point number.

setDoubleValue:, floatValue, intValue, stringValue

drawBarInside:(const NXRect *)cellFrame flipped:(BOOL)flipped

Draws the SliderCell's background bar, but not the bezel around it or the knob. flipped indicates w coordinate system is flipped or not. Returns self.

Override this method if you want to draw your own slider bar. Note, however, that the setImage: m conveniently customize the appearance of the SliderCell's background.

drawInside:inView:, drawSelf:inView:, isFlipped (View), setImage:, lockFocus (View)

drawInside:(const NXRect *)cellFrame inView:controlView

Draws the SliderCell's background bar and knob, along with the background title, but not the beze must be locked on controlView when this message is sent. Returns self.

drawBarInside:flipped:, drawKnob, drawSelf:inView:, lockFocus (View)

drawKnob

Calculates the rectangle in which the knob should be drawn and invokes drawKnob: to actually dra PostScript focus must be locked on the SliderCell's View when this message is sent. You never ov override drawKnob: instead.

drawKnob:, lockFocus (View)

drawKnob:(const NXRect*)knobRect

Draws the knob in knobRect. The PostScript focus must be locked on the SliderCell's View when

Override this method and getKnobRect:flipped: if you want to draw your own knob. You should a calcCellSize:inRect: if your knob is of a different size from the standard SliderCell knob.

drawKnob, getKnobRect:flipped:, calcCellSize:inRect:, isVertical, lockFocus (View)

drawSelf:(const NXRect *)cellFrame inView:controlView

Draws the SliderCell background bar (including the bezel) and knob. The knob is drawn at a posit current value of the SliderCell. This method doesn't invoke drawInside:inView:. The PostScript f controlView when this message is sent. Returns self.

(float)floatValue

Returns the value of the SliderCell as a single-precision floating point number.

setFloatValue:, doubleValue, intValue, stringValue

getKnobRect:(NXRect*)knobRect flipped:(BOOL)flipped

Returns self, and by reference in knobRect the rectangle into which the knob will be drawn. This r
from the SliderCell's value in relation to its tracking rectangle and its minimum and maximum val
whether the SliderCell's View has a flipped coordinate system.

Override this method and drawKnob: if you want to draw your own knob. You should also overri
if your knob is of a different size from the standard SliderCell knob (and be careful of setting the k
Remember to take into account the flipping of the View in vertical SliderCells otherwise, your kno
correct distance from the wrong end.

drawKnob:, calcCellSize:inRect:, isVertical, isFlipped (View)

image

Returns the NXImage that the SliderCell displays as its background.

setImage:

init

Initializes and returns the receiver, a new instance of SliderCell. Its value is set to 0.0, minimum v
value to 1.0. New SliderCells are continuous by default.

This method is the designated initializer for SliderCell override it if you create a subclass of Slider
own initialization. You shouldn't use Cell's designated initializers, initIconCell: or initTextCell:, t
SliderCell.

setMinValue:, setMaxValue:, setFloatValue:, setContinuous:

(int)intValue

Returns the value of the SliderCell as an integer.

setIntValue:, doubleValue, floatValue, stringValue

(BOOL)isContinuous

Returns YES if the action is sent to the target continuously as mouse-dragged events occur while t
up event NO if the action is sent only on a mouse-up event.

setContinuous:

(int)isVertical

Returns 1 if the SliderCell is vertical, 0 if it's horizontal, and 1 if the orientation can't be determine SliderCell hasn't been drawn in a View, for example). A SliderCell is vertical if its height is great

(NXCoord)knobThickness

Returns the thickness of the SliderCell's knob (that is, its extent along the bar's length) in the Slide system.

setKnobThickness:

(double)maxValue

Returns the maximum value of the SliderCell.

setMaxValue:, minValue

(double)minValue

Returns the minimum value of the SliderCell.

setMinValue:, maxValue

read:(NXTypedStream *)stream

Reads the SliderCell from the typed stream stream. Returns self.

write:, awake

setAltIncrementValue:(double)incValue

Sets the amount that the SliderCell will alter its value when the user drags the knob one pixel with down. incValue should be greater than 0.0, and less than the SliderCell's maximum value it can al this feature is disabled. Normally, you'll want to use this method with incValue less than 1.0, so t slowly than the mouse.

altIncrementValue, maxValue

setContinuous:(BOOL)flag

If flag is YES, the SliderCell will send its action to its target continuously as mouse-dragged event or on a mouse-up event. If NO, the SliderCell will send its action only on a mouse-up event. The self.

setEnabled:(BOOL)flag

If flag is YES, the SliderCell will become enabled if NO, the SliderCell will become disabled. A d
draws its non-image background in light gray. An enabled SliderCell draws its non-image backgro

isEnabled (ActionCell)

setFloatValue:(float)aFloat

Sets the value of the SliderCell to aFloat. Updates the SliderCell knob position to reflect the new v

floatValue, setDoubleValue:, setIntValue:, setStringValue:

setImage:image

Sets the NXImage used as the SliderCell's background. This method doesn't scale the NXImage.

image

setIntValue:(int)anInt

Sets the value of the SliderCell to anInt. Updates the SliderCell knob position to reflect the new va

intValue, setDoubleValue:, setFloatValue:, setStringValue:

setKnobThickness:(NXCoord)aFloat

Sets the thickness of the SliderCell's knob (that is, its extent along the bar's length) in its own coor
must be greater than 0.0, and shouldn't be greater than the Slider's length. If the knob thickness ch
inside is redrawn. Returns self.

knobThickness

setMaxValue:(double)aDouble

Sets the maximum value of the SliderCell to aDouble and returns self. If the maximum value chan
inside is redrawn to reposition the knob relative to the new maximum.

maxValue, setMinValue:

setMinValue:(double)aDouble

knob position is updated to reflect the new value otherwise, does nothing. Returns self.

stringValue, setDoubleValue:, setFloatValue:, setIntValue:

setTitle:(const char *)aString

Sets the title drawn over the SliderCell's background to aString. Returns self.

setTitleNoCopy:, title

setTitleCell:aCell

Sets the Cell used to draw the SliderCell's background title. aCell should be an instance of TextFi
subclass). Doesn't redraw the SliderCell further, a setTitle: message is required to display a title, e
has a string value. Returns the old Cell.

titleCell, setTitle:

setTitleColor:(NXColor)color

Sets the color used to draw the SliderCell's background title, redraws the SliderCell's inside, and r
is to draw in a gray level of 0.0 (NX_BLACK).

titleColor, setTitleGray:

setTitleFont:fontObject

Sets the Font used to draw the SliderCell's background title and redraws the SliderCell's inside. T
default system font as set by the user (with the Preferences application), and its size is 12.0 point.

titleFont

setTitleGray:(float)aFloat

Sets the gray value used to draw the SliderCell's background title, redraws the SliderCell's inside,
default gray level is 0.0 (NX_BLACK).

titleGray, setTitleColor:

setTitleNoCopy:(const char *)aString

Sets the title drawn over the SliderCell's background to aString, but doesn't copy the string. Retur

setTitle:, title

at:(const NXPoint *)stopPoint
inView:controlView
mouseIsUp:(BOOL)flag

Ends tracking by moving the knob to stopPoint. Returns self.

trackMouse:inRect:ofView:, startTrackingAt:inView:, continueTracking:at:inView:

(const char *)stringValue

Returns a string representing the value of the SliderCell. The floating point format is applied when representation.

setStringValue:, doubleValue, floatValue, intValue, setFloatingPointFormat:left:right: (Cell)

(const char *)title

Returns the string used as the SliderCell's background title. The title is drawn over the SliderCell's self.

setTitle:

titleCell

Returns the TextFieldCell used to draw the SliderCell. If the SliderCell doesn't have a title, a new created and returned. This doesn't result in a title getting set.

setTitleCell:

(NXColor)titleColor

Returns the color used to draw the SliderCell's background title. The default is to draw in a gray l (NX_BLACK). Returns self.

setTitleColor:, titleGray

titleFont

Returns the Font used to draw the SliderCell's title. The default font is the default system font as s Preferences application), and its size is 12.0 point.

setTitleFont:

(float)titleGray

Tracks the mouse until it goes up or until it goes outside the cellFrame. If cellFrame is NULL, the mouse goes up. Since SliderCell responds YES to prefersTrackingUntilMouseUp, this method wi NULL cellFrame. Returns YES if the mouse goes up, NO otherwise.

If the SliderCell is continuous, then the action will be continuously sent to the target as the mouse cellFrame isn't the same cellFrame that was passed to the last drawSelf:inView:, then this method

startTrackingAt:inView:, continueTracking:at:inView:, stopTracking:at:inView:mouseIsUp:, se

write:(NXTypedStream *)stream

Writes the receiving SliderCell to the typed stream stream. Returns self.

read: