

setEnabled:

setBezeled:  
setBordered:  
setAlignment:  
setFloatingPointFormat:left:right:  
setFont:  
setIcon:

Manipulating ActionCell values

doubleValue  
floatValue  
intValue  
setStringValue:  
setStringValueNoCopy:shouldFree:  
stringValue

Displaying drawSelf:inView:

controlView

Target and action setAction:

action  
setTarget:  
target

Assigning a tag setTag:

tag

Archiving write:

read:

(SEL)action

returned by this method only to (indirectly) redisplay the ActionCell. For example, the subclasses by the Application Kit invoke this method in order to send the Control a message such as updateC

The Control in which an ActionCell is drawn is set automatically by the drawSelf:inView: method set the Control.

drawSelf:inView:

(double)doubleValue

Returns the ActionCell's contents as a double-precision floating point number. If the ActionCell is editing when this message is received, editing is validated first.

setDoubleValue: (Cell), floatValue, intValue, stringValue, validateEditing (Control)

drawSelf:(const NXRect \*)cellFrame inView:controlView

Displays the ActionCell. Sets the ActionCell's Control to controlView and performs drawing if any is a Control object (an instance of Control or a subclass thereof). You must lock focus on the Control before calling this method (Control's display method automatically performs this).

drawSelf:inView: (Cell)

(float)floatValue

Returns the ActionCell's contents as a single-precision floating point number. If the ActionCell is editing when this message is received, editing is validated first.

setFloatValue: (Cell), doubleValue, intValue, stringValue, validateEditing (Control)

(int)intValue

Returns the ActionCell's contents as an integer. If the ActionCell is being edited when this message is received, editing is validated first.

setIntValue: (Cell), doubleValue, floatValue, stringValue, validateEditing (Control)

read:(NXTypedStream \*)stream

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

write:

setAction:(SEL)aSelector

NX\_LEFTALIGNED, NX\_CENTERED, or NX\_RIGHTALIGNED. If it's currently in a Control view, the ActionCell is redisplayed or marked as needing redisplay. Returns self.

alignment (Cell)

setBezeled:(BOOL)flag

Adds or removes the ActionCell's bezel, according to the value of flag. Adding a bezel will remove the border, if any. If it's currently in a Control view, the ActionCell is redisplayed or marked as needing redisplay. Returns self.

isBezeled (Cell), setBordered:

setBordered:(BOOL)flag

Adds or removes the ActionCell's border, according to the value of flag. The border is black and 1 pixel wide. Adding a border will remove the ActionCell's bezel, if any. If it's currently in a Control view, the ActionCell is redisplayed or marked as needing redisplay. Returns self.

isBordered (Cell), setBezeled:

setEnabled:(BOOL)flag

Enables or disables the ActionCell's ability to receive mouse events, according to the value of flag. If it's currently in a Control view, the ActionCell is redisplayed or marked as needing redisplay. Returns self.

isEnabled (Cell)

setFloatingPointFormat:(BOOL)autoRange

left:(unsigned int)leftDigits

right:(unsigned int)rightDigits

Sets the ActionCell's floating point format as described in the Cell class specification for the setFloatingPointFormat: method. If it's currently in a Control view, the ActionCell is redisplayed or marked as needing redisplay. Returns self.

setFloatingPointFormat:left:right: (Cell)

setFont:fontObject

If the ActionCell is a text Cell (type NX\_TEXTCELL), this sets its Font to fontObject. In addition, if it's currently in a Control view, the ActionCell is redisplayed or marked as needing redisplay. Returns self.

font (Cell)

setIcon:(const char \*)iconName

marked as needing redisplay. Returns self.

setStringValueNoCopy:shouldFree:, setStringValue: (Cell), stringValue, doubleValue, floatValue

setStringValueNoCopy:(char \*)aString shouldFree:(BOOL)flag

Sets the ActionCell's contents to a aString. If flag is YES, aString will be freed when the ActionCell is currently in a Control view, the ActionCell is redisplayed or marked as needing redisplay. Returns self.

setStringValue:, setStringValueNoCopy:shouldFree: (Cell), stringValue, doubleValue, floatValue

setTag:(int)anInt

Sets the ActionCell's tag to anInt. The tag can be used to identify the ActionCell in a Control that contains it (Matrix, for example). Returns self.

tag, setTag: (Control)

setTarget:anObject

Sets the ActionCell's target to anObject. This is the object that is sent the ActionCell's action method when the target is clicked.

target, setAction:

(const char \*)stringValue

Returns the ActionCell's contents as a string. If the ActionCell is being edited when this message is sent, the string is validated first.

setStringValue:, stringValue (Cell), validateEditing (Control), doubleValue, floatValue, intValue

(int)tag

Returns the ActionCell's tag. The tag can be used to identify the ActionCell in a Control that contains it (Matrix, for example).

setTag:, tag (Control)

target

Returns the ActionCell's target, the object that is sent the ActionCell's action method.

setTarget:, action

