	initFrame:
	free
Setting the Control's Cell	+ setCellClass:
	setCell:
	cell
Enabling and disabling the Control	
	isEnabled
	setEnabled:
Identifying the selected Cell	selectedCell
	selectedTag
Setting the Control's value	setFloatValue:
	floatValue
	setDoubleValue:
	doubleValue
	setIntValue:
	intValue
	setStringValue:
	setStringValueNoCopy:
	setStringValueNoCopy:shouldFree:
	stringValue
Interacting with other Controls	takeDoubleValueFrom:
	takeFloatValueFrom:
	takeIntValueFrom:
	takeStringValueFrom:
Formatting text	setAlignment:
	alignment
	setFont:
	font
	setFloatingPointFormat:left:right:
Managing the field editor	abortEditing
	currentEditor
	validateEditing
Managing the cursor	resetCursorRects
Resizing the Control	calcSize
	sizeTo::
	sizeToFit
Displaying the Control and Cell	drawCell:
	drawCellInside:
	drawSelf::
	selectCell:
	update
	updateCell:
	updateCellInside:
Target and action	setAction:

Tracking the mouse ignoreMultiClick:
mouseDown:
mouseDownFlags

Archiving read:
write:

abortEditing

Terminates and discards any editing of text displayed by the receiving Control. Returns self, or nil
going on in the receiving Control. This method doesn't redisplay the old value of the Control.

endEditingFor: (Window), validateEditing

(SEL)action

Returns the action message sent by the Control's Cell, or the default action message for a Control v
(such as a Matrix or Form). To retrieve the action message, this method sends an action message t
Controls with multiple Cells, it's better to get the action message for a particular Cell using:

setAction:, target, sendAction:to:

(int)alignment

Returns the alignment mode of the text in the Control's Cell. The return value can be one of three
NX_LEFTALIGNED, NX_CENTERED or NX_RIGHTALIGNED.

setAlignment:

calcSize

Recomputes any internal sizing information for the Control, if necessary, by invoking its Cell's cal
This method doesn't actually draw. It can be used for more sophisticated sizing operations as well
calcSize is automatically invoked whenever the Control is displayed and something has changed y
Returns self.

calcSize (Matrix, Form), sizeToFit

currentEditor

If the receiving Control is being edited (that is, has a Text object acting as its editor, and is the first
Window), this method returns the Text object being used to perform that editing. If the Control isn
method returns nil.

abortEditing, validateEditing

(double)doubleValue

Returns the value of the Control's selected Cell as a double-precision floating point number. If the
cells (for example, Matrix), then the value of the currently selectedCell is returned. If the Control
editing the affected Cell, then validateEditing is invoked before the value is extracted and returned

setDoubleValue:, floatValue, intValue, stringValue

drawCell:aCell

If aCell is the cell used to implement this Control, then the Control is displayed. This method is pr
support a consistent set of methods between Controls with single and multiple Cells, since a Contr
needs to be able to draw a single Cell at a time. Returns self.

updateCell:, drawCellInside:, updateCellInside:,
drawCell: (Matrix)

drawCellInside:aCell

Draws the inside of a Control (the area within a bezel or border). This method invokes Cell's draw
drawCellInside: is used by setStringValue: and similar content-setting methods to provide a minim
Control when its value is changed. Returns self.

drawCell:, drawInside:inView: (Cell), drawCellInside: (Matrix), updateCellInside:

drawSelf:(const NXRect *)rects :(int)rectCount

Draws the Control. This method invokes the drawSelf:inView: method of the Control's Cell. You
method if you have a Control with multiple Cells. Returns self.

drawSelf:inView: (Cell)

(float)floatValue

Returns the value of the Control's selected Cell as a single-precision floating point number. See d
details.

setFloatValue:, doubleValue, intValue, stringValue

Frees the memory used by the Control and its Cells. Aborts editing if the text of the Control was c
Returns nil.

free (View)

ignoreMultiClick:(BOOL)flag

Sets the Control to ignore multiple clicks if flag is YES. By default, double-clicks (and higher ord
the same as single clicks. You can use this method to ªdebounceº a Control, so that it won't inadv
message twice when double-clicked. Returns self.

initFrame:(const NXRect *)frameRect

Initializes and returns the receiver, a new instance of Control, by setting the value pointed to by fra
rectangle. Makes the new instance an opaque View. Since Control is an abstract class, messages t
should appear only in subclass methods that is, there should always be a more specific designated
subclass. initFrame: is the designated initializer for the Control class.

(int)intValue

Returns the value of the Control's selected Cell as an integer (see doubleValue for more details).

setIntValue:, doubleValue, floatValue, stringValue

(BOOL)isContinuous

Returns YES if the Control's Cell continuously sends its action message to its target during mouse

setContinuous:

(BOOL)isEnabled

Returns YES if the Control is enabled, NO otherwise.

setEnabled:

mouseDown:(NXEvent *)theEvent

Highlights the Control, and sends trackMouse:inRect:ofView: to the Control's Cell (or whichever
occured in if the Control has multiple Cells). This method is invoked when the mouse button goes
is within the bounds of the Control. The Control's Cell tracks the cursor until it goes outside the b
the Control is unhighlighted. If the cursor goes back into the bounds, then the Control highlights a
tracking again. This behavior continues until the mouse button goes up. If it goes up with the curs

flags are valid only in the action method invoked upon the Control's target.

mouseDownFlags (Cell), sendAction:to:

read:(NXTypedStream *)stream

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

resetCursorRects

Reestablishes the cursor rectangles for the Control's Cell (or Cells). If the Cell displays text, and t
selectable, then resetCursorRect:inView: is sent to the Cell. resetCursorRect:inView: in turn, sends
cursor: back to the Control, so that the cursor will change to an I-beam when it enters the Cell's re

resetCursorRect:inView: (Cell), addCursorRect:cursor: (View)

selectCell:aCell

If aCell is a Cell of the receiving Control and is unselected, this method selects aCell and redraws t
self.

selectedCell

Returns the Control's selected Cell. The target of the Control should use this method when it want
sending Control. Note that even though the cell method will return the same value for Controls wi
it's strongly suggested that this method be used since it will work for Controls with either a single

sendAction:to:, selectedCell (Matrix)

(int)selectedTag

Returns the tag of the Control's selected Cell. This is equivalent to:

1 if there is no selected Cell. The Cell's tag can be set with ActionCell's setTag: method. You sho
setTag: and tag methods in conjunction with findViewWithTag:. When you set the tag of a Contro
Interface Builder, it sets both the tags of both Control and Cell as a convenience.

sendAction:to:

sendAction:(SEL)theAction to:theTarget

Sends a sendAction:to:from: message to NXApp, which in turn sends a message to theTarget to pe
sendAction:to:from: adds the Control as theAction's only argument. If theAction is NULL, no me
sendAction:to: is invoked primarily by Cell's trackMouse:inRect:ofView:

Uses mask to record the events that cause sendAction:to: to be invoked during tracking of the mou
in Cell's trackMouse:inRect:ofView:. Returns the old event mask.

sendAction:to:, sendActionOn: (Cell), trackMouse:inRect:ofView: (Cell)

setAction:(SEL)aSelector

Makes aSelector the Control's action method. If aSelector is NULL, then no action messages will
Control. Returns self.

action, setTarget:, sendAction:to:

setAlignment:(int)mode

Sets the alignment mode of the text in the Control's Cell, or of all the Control's Cells if it has more
the Control. mode should be one of: NX_LEFTALIGNED, NX_CENTERED or NX_RIGHTALI

alignment

setCell:aCell

Sets the Cell of the Control to be cell. Use this method with great care as it can irrevocably damag
specifically, you should only use this method in initializers for subclasses of Control. Returns the

setContinuous:(BOOL)flag

Sets whether the Control will continuously send its action message to its target as the mouse is trac

setContinuous: (ButtonCell, SliderCell), sendActionOn:

setDoubleValue:(double)aDouble

Sets the value of the Control's selected Cell to be aDouble (a double-precision floating point numb
is being edited, that editing is aborted and the value being typed is discarded in favor of aDouble.
then the Cell's inside (the area within a bezel or border) is redrawn. Returns self.

doubleValue, setFloatValue:, setIntValue:, setStringValue:, abortEditing, drawInside:inView: (
(View), setAutodisplay: (View)

setEnabled:(BOOL)flag

Sets whether the Control is active or not (that is, whether it tracks the mouse and sends its action to
NO, any editing is aborted. Redraws the entire Control if autodisplay is on. Subclasses may want
redraw only a portion of the Control when the enabled state changes (Button and Slider do this). R

setFloatingPointFormat:(BOOL)autoRange
left:(unsigned)leftDigits
right:(unsigned)rightDigits

Sets the autoranging and floating point number format of the Control's Cell, so that at most leftDig
left of the decimal point, and rightDigits to the right. If the Control has more than one Cell, they're
description of this method in the Cell class specification for more detail. This method doesn't redr
setFloatingPointFormat:left:right: affects only subsequent invocations of setFloatValue:. Returns s

setFloatingPointFormat:left:right: (Cell)

setFont:fontObject

Sets the Font object used to draw the text (if any) in the Control's Cell, or in all the Cells if the Co
You only need to use this method if you don't want to use the user's default system font (as set by
Preferences application). If autodisplay is on, then the inside of the Cell is redrawn. Returns self.

font, isAutodisplay (View), setAutodisplay: (View)

setIntValue:(int)anInt

Same as setDoubleValue:, but sets the value as an integer. Returns self.

intValue, setDoubleValue:, setFloatValue:, setStringValue:

setStringValue:(const char *)aString

Same as setDoubleValue:, but sets the value as a string by copying it from aString. Returns self.

stringValue, setStringValueNoCopy:, setStringValueNoCopy:shouldFree:, setDoubleValue:, se
setIntValue:

setStringValueNoCopy:(const char *)aString

Like setStringValue:, but doesn't copy the string. Returns self.

stringValue, setStringValue:, setStringValueNoCopy:, setStringValueNoCopy:shouldFree:, set
setFloatValue:, setIntValue:

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

Like setStringValueNoCopy:, but lets you specify whether the string should be freed when the Co
self.

stringValue, setStringValue: setStringValueNoCopy:, setDoubleValue:, setFloatValue:, setInt setTarget:anObject

Sets the target for the action message of the Control's Cell. Returns self.

If anObject is nil, then when an action message is sent, NXApp looks for an object that can respon
following the responder chain, as detailed in the class description.

target, setAction:, sendAction:to:

sizeTo:(NXCoord)width :(NXCoord)height

Changes the width and the height of the Control's frame. Redisplays the Control if autodisplay is

isAutodisplay (View), setAutodisplay: (View)

sizeToFit

Changes the width and the height of the Control's frame so that they are the minimum needed to c
Control has more than one Cell, then you must override this method. Returns self.

sizeToFit (Matrix), sizeToCells (Matrix)

(const char *)stringValue

Returns the value of the Control's selected Cell as a string. If the Control is in the process of editi
then validateEditing is invoked before the value is extracted and returned.

setStringValue:, doubleValue, floatValue, intValue

(int)tag

Returns the receiving Control's tag (not the tag of the Control's Cell).

setTag:, selectedTag, tag (Cell)

takeDoubleValueFrom:sender

Sets the double-precision floating-point value of the receiving Control's selected Cell to the value
doubleValue message to sender. Returns self.

This method can be used in action messages between Controls. It permits one Control (the sender
another Control (the receiver) by invoking this method in an action message to the receiver. For e
be made the target of a Slider. Whenever the Slider is moved, it will send a takeDoubleValueFro
TextField. The TextField will then get the Slider's floating-point value, turn it into a text string, a
tracking the value of the Slider.

setDoubleValue:, doubleValue

takeIntValueFrom:sender

Sets the integer value of the receiving Control's selected Cell to the value returned by sending an i
sender. Returns self.

See takeDoubleValueFrom: for an example.

setIntValue:, intValue

takeStringValueFrom:sender

Sets the character string of the receiving Control's selected Cell to a string obtained by sending a s
sender. Since this is an action method, there is no alternate like takeStringValueFrom:noCopy:. R

See takeDoubleValueFrom: for an example.

stringValue, setStringValue:

target

Returns the target for the action message of the Control's cell, or the Control's target for a Control
nil, then any action messages sent by the Control will be sent up the responder chain, as detailed in

setTarget:, action, sendAction:to:

update

If autodisplay is enabled, sends a display message to itself. Otherwise it simply sets a flag indicati
needs to be displayed. This method also makes sure that calcSize is performed. Returns self.

updateCell:, updateCellInside:

updateCell:aCell

If aCell is a Cell used to implement this Control, and if autodisplay is on, then draws the Control's
needsDisplay and calcSize flags to YES. Returns self.

update, updateCellInside:, isAutodisplay (View), setAutodisplay: (View)

updateCellInside:aCell

If aCell is a Cell used to implement this Control, and if autodisplay is on, draws the inside portion
sets the needsDisplay flag to YES. Returns self.

update, updateCell:, isAutodisplay (View), setAutodisplay: (View) endEditingFor: (Window), abortEditing

write:(NXTypedStream *)stream

Writes the Control to the typed stream stream.

read: