	init
	initTextCell:
Copying a TextFieldCell	copyFromZone:
Setting the TextFieldCell's value	setFloatValue: (Cell)
	floatValue (Cell)
	setDoubleValue: (Cell)
	doubleValue (Cell)
	setIntValue: (Cell)
	intValue (Cell)
	setStringValue: (Cell)
	setStringValueNoCopy: (Cell)
	setStringValueNoCopy:shouldFree: (Cell)
	stringValue (Cell)
Modifying Graphic Attributes	setTextColor:
	textColor
	setTextGray:
	textGray
	setBackgroundColor:
	backgroundColor
	setBackgroundGray:
	backgroundGray
	setBackgroundTransparent:
	isBackgroundTransparent
	setTextAttributes:
	setBezeled:
	isOpaque
Displaying	drawSelf:inView:
	drawInside:inView:
Tracking the Mouse	trackMouse:inRect:ofView:
Archiving	read:
	write:

(float)backgroundGray

Returns the gray level used to draw the background on monochrome displays. If the gray level is l
background is transparent.

setBackgroundGray:, backgroundColor

copyFromZone:(NXZone *)zone

Creates and returns a new TextFieldCell as a copy of the receiver, allocated from zone.

drawInside:(const NXRect *)cellFrame inView:controlView

Draws the inside of the TextFieldCell (the background and text, but not the bezel or border). This
from drawSelf:inView: and also from Control and its subclasses' drawCellInside: method. If you s
and you override drawSelf:inView:, then you should override this method as well. Returns self.

drawSelf:inView:

drawSelf:(const NXRect *)cellFrame inView:controlView

Draws the TextFieldCell's background, text, and border or bezel. Returns self.

drawInside:inView:

init

Initializes and returns the receiver, a new instance of TextFieldCell, with the default title, ªFieldº.
as described in initTextCell: below.

initTextCell:

initTextCell:(const char *)aString

Initializes and returns the receiver, a new instance of TextFieldCell, with aString as its text. The de
NX_BLACK, and the default background gray is transparent (1.0). Its font is set to the user's syst
size is 12.0 point.

This method is the designated initializer for TextFieldCell. Override this method if you create a su
that performs its own initialization. Note that TextFieldCell doesn't override Cell's initIconCell: d
your code shouldn't use that method to initialize an instance of TextFieldCell.

init

(BOOL)isBackgroundTransparent

setBezeled:, setBackgroundGray:

read:(NXTypedStream *)stream

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

write:

setBackgroundColor:(NXColor)aColor

Sets the background color for the TextFieldCell to aColor. The background color is used only on c
monochrome displays, the background gray is used. Returns self.

backgroundColor, setBackgroundGray:

setBackgroundGray:(float)value

Sets the gray level that will be used to draw the background on monochrome displays. If value is N
background will be drawn. If the cell is editable, it should have a background gray greater than or
self.

backgroundGray, setBackgroundColor:

setBackgroundTransparent:(BOOL)flag

If flag is YES, sets the background gray of the TextFieldCell to transparent (a negative value) if NO
gray to NX_WHITE. Returns self.

setBackgroundGray:

setBezeled:(BOOL)flag

If flag is YES, the TextFieldCell is drawn with a bezel around the edge if NO, nothing is drawn aro
current background gray is transparent, it's changed to NX_WHITE. Bezeled transparent TextFiel
strange, but if you want to have one, invoke setBackgroundGray: with 1.0 after invoking setBezele

isBezeled (Cell), setBackgroundGray:

setTextAttributes:textObject

Used to set the attributes of the field editor when editing the TextFieldCell (see the TextFieldCell c
the background and text colors or gray levels of textObject to those of the TextFieldCell, and retur
textObject should respond to the messages setBackgroundGray:, setBackgroundColor:, setTextGra
You rarely need to override this method you never need to invoke it.

setTextGray:, setBackgroundGray:, setTextAttributes: (Cell)

setTextColor:(NXColor)aColor

Sets the color used to draw the text. On monochrome displays this can result in unsightly dithering should only be used if the drawing view responds YES to a shouldDrawColor message. setTextCo text to be redrawn. Returns self.

textColor, setTextGray:, shouldDrawColor (View)

setTextGray:(float)value

Sets the gray level used to draw the text, regardless of the color capability of the display. value sho 0 (indicating black) to 1.0 (indicating white). To specify one of the four pure shades of gray, use c

textGray, setTextColor:

(NXColor)textColor

Returns the color used to draw the text. Returns self.

setTextColor:, textGray

(float)textGray

Returns the gray level used to draw the text. Returns self.

setTextGray:, textColor

(BOOL)trackMouse:(NXEvent*)theEvent
inRect:(const NXRect*)aRect
ofView:controlView

Causes editing to occur, and increments the state of the TextFieldCell if its enabled and the mouse Returns YES if the mouse goes up in the TextFieldCell, NO otherwise.

trackMouse:inRect:ofView: (Cell)