1, width is calculated as needed.

titleCell The Cell used to draw the title.

titleEndPoint The coordinate that separates the title from the text area.

	init
	initTextCell:
	copyFromZone:
	free
Determining a FormCell's size	calcCellSize:inRect:
Enabling the FormCell	setEnabled:
Modifying the title	setTitle:
	title
	setTitleFont:
	titleFont
	setTitleAlignment:
	titleAlignment
	setTitleWidth:
	titleWidth:
	titleWidth
Modifying graphic attributes	isOpaque
Displaying	drawInside:inView:
	drawSelf:inView:
Managing cursor rectangles	resetCursorRect:inView:
Tracking the mouse	trackMouse:inRect:ofView:
Archiving	read:
	write:

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

Calculates the size of the FormCell assuming it's constrained to fit within aRect. Returns the size in theSize.

and override drawSelf:inView:, you must implement this method as well. Returns self.

drawSelf:inView:

drawSelf:(const NXRect *)cellFrame inView:controlView

Has the FormCell's title Cell drawn, then draws the editable text portion of the FormCell. returns

drawInside:inView:

free

Frees the storage used by the FormCell and returns nil.

init

Initializes and returns the receiver, a new instance of FormCell, with its contents set to an empty st
to ªFieldº, right-aligned.

initTextCell:

initTextCell:(const char *)aString

Initializes and returns the receiver, a new instance of FormCell, with its contents set to the empty s
set to aString. The font for both title and text is the user's chosen system font in 12.0 point, and th
with a bezel. This method is the designated initializer for FormCell.

init

(BOOL)isOpaque

Returns YES if the FormCell is opaque, NO otherwise. If the FormCell has a title, then it's not op
field is not opaque).

isOpaque (Cell)

read:(NXTypedStream *)stream

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

write:

resetCursorRect:(const NXRect *)cellFrame inView:controlView isEnabled (Cell)

setTitle:(const char *)aString

Sets the title of the FormCell to aString.

title

setTitleAlignment:(int)mode

Sets the alignment of the title. mode can be one of three constants: NX_LEFTALIGNED, NX_CE
NX_RIGHTALIGNED.

titleAlignment

setTitleFont:fontObject

Sets the Font used to draw the title of the FormCell.

setFont:

setTitleWidth:(NXCoord)width

Sets the width of the title field to width. If width is 1, the title field's width is always calculated w
method only if the FormCell's title isn't going to change, or if your code always resets the title wid
title.

titleWidth, titleWidth:

(const char *)title

Returns the title of the FormCell.

setTitle:

(int)titleAlignment

Returns the alignment of the title, which will be one of the following: NX_LEFTALIGNED, NX_C
NX_RIGHTALIGNED.

setTitleAlignment:

If the width of the title has already been set, then that value is returned. Otherwise, it's calculated

setTitleWidth:, titleWidth:

(NXCoord)titleWidth:(const NXSize *)aSize

If the title width has been set, then it's returned. Otherwise, the width is calculated constrained to
NULL, in which case the width is calculated without constraint.

setTitleWidth:, titleWidth:

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

Causes editing to occur. Returns YES if the mouse goes up in the FormCell, NO otherwise.

trackMouse:inRect:ofView: (TextFieldCell)

write:(NXTypedStream *)stream

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

read: