changeSpelling:

NXIgnoreMisspelledWords spellDocumentTag

NXReadOnlyTextStream openTextStream
seekToCharacterAt:relativeTo:
readCharacters:count:
currentCharacterOffset
isAtEOTS
closeTextStream

NXSelectText selectCharactersFrom:to:
selectionCharacterCount
readCharactersFromSelection:count:
makeSelectionVisible

initFrame:

isHorizResizable
sizeTo::
sizeToFit
resizeText::
moveTo::

Laying out the text setMarginLeft:right:top:bottom:
getMarginLeft:right:top:bottom:
getMinWidth:minHeight:maxWidth:maxHeight:
setAlignment:
alignment
alignSelLeft:
alignSelCenter:
alignSelRight:
setSelProp:to:
changeTabStopAt:to:
calcLine
setCharWrap:
charWrap
setNoWrap
setParaStyle:
defaultParaStyle
calcParagraphStyle::
setLineHeight:
lineHeight
setDescentLine:
descentLine

Reporting line and position
positionFromLine:
offsetFromPosition:
positionFromOffset:

Setting, reading, and writing the text
setText:
readText:
startReadingRichText
readRichText:
readRichText:atPosition:
readRTFDFrom:
finishReadingRichText
openRTFDFrom:
saveRTFDTo:removeBackup:errorHandler:
writeText:
writeRichText:
writeRichText:from:to:
writeRTFDSelectionTo:
writeRTFDTo:
stream
firstTextBlock
getParagraph:start:end:rect:
getSubstring:start:length:
byteLength
charLength
textLength

Setting editability setEditable:
isEditable

cut:
delete:
clear:
selectAll:
selectText:

Managing the selection subscript:
superscript:
unscript:
underline:
showCaret
hideCaret
setSelectable:
isSelectable
selectError
selectNull
setSel::
getSel::
replaceSel:
replaceSel:length:
replaceSel:length:runs:
replaceSelWithRichText:
replaceSelWithRTFD:
scrollSelToVisible

Setting the font setFontPanelEnabled:
isFontPanelEnabled
changeFont:
setFont:
font
setFont:paraStyle:
setSelFont:
setSelFontFamily:
setSelFontSize:
setSelFontStyle:
setSelFont:paraStyle:

Checking spelling checkSpelling:
showGuessPanel:

Managing the ruler toggleRuler:
isRulerVisible

Finding text findText:ignoreCase:backwards:wrap:

Modifying graphic attributes setBackgroundGray:
backgroundGray
setBackgroundColor:
backgroundColor
setSelGray:
selGray
runGray:
setSelColor:
selColor
runColor:
setTextGray:
textGray
setTextColor:
textColor

Handling event messages acceptsFirstResponder
becomeFirstResponder
resignFirstResponder
becomeKeyWindow
resignKeyWindow
mouseDown:
keyDown:
moveCaret:

Handling graphics within the text
+ registerDirective:forClass:
replaceSelWithCell:
setLocation:ofCell:
getLocation:ofCell:
setGraphicsImportEnabled:
isGraphicsImportEnabled

Using the Services menu+ excludeFromServicesMenu:
validRequestorForSendType:andReturnType:
readSelectionFromPasteboard:
writeSelectionToPasteboard:types:

Setting tables and functions setCharFilter:
charFilter
setTextFilter:
textFilter
setBreakTable:
breakTable
setPreSelSmartTable:
preSelSmartTable
setPostSelSmartTable:
postSelSmartTable
setCharCategoryTable:
charCategoryTable
setClickTable:
clickTable
setScanFunc:
scanFunc
setDrawFunc:
drawFunc

Printing adjustPageHeightNew:top:bottom:limit:

Archiving read:
write:

Assigning a delegate setDelegate:
delegate

validRequestorForSendType:andReturnType:, registerServicesMenuSendTypes:andReturnTypes

replaceSelWithCell:

setLineHeight:, + newFont:size: (Font)

(BOOL)acceptsFirstResponder

Assuming the text is selectable, returns YES to let the Text object become the first responder other
acceptsFirstResponder messages are sent for you you never send them yourself.

setSelectable:, setDelegate:, resignFirstResponder

adjustPageHeightNew:(float *)newBottom
top:(float)oldTop
bottom:(float)oldBottom
limit:(float)bottomLimit

During automatic pagination, this method is performed to help lay a grid of pages over the top-leve
newBottom is passed in undefined and must be set by this method. oldTop and oldBottom are the
horizontal strip being created. bottomLimit is the topmost value newBottom can be set to. If this l
value is ignored. By default, this method tries to prevent the view from being cut in two. All para
own coordinate system. Returns self.

setAlignment:

alignSelCenter:sender

Sets the paragraph style of one or more paragraphs so that text is centered between the left and righ
Text object, all paragraphs are affected. For a rich Text object, only those paragraphs marked by t
affected. The sending object passes its id as part of the alignSelCenter: message. The text is rewra
Returns self.

alignSelLeft:, alignSelRight:, setSelProp:to:, setMonoFont:

alignSelLeft:sender

Sets the paragraph style of one or more paragraphs so that text is aligned to the left margin. For a
paragraphs are affected. For a rich Text object, only those paragraphs marked by the selection are
object passes its id as part of the alignSelLeft: message. The text is rewrapped and redrawn. Retu

alignSelCenter:, alignSelRight:, setSelProp:to:, setMonoFont:

alignSelRight:sender

Sets the paragraph style of one or more paragraphs so that text is aligned to the right margin. For a
paragraphs are affected. For a rich Text object, only those paragraphs marked by the selection are
object passes its id as part of the alignSelRight: message. The text is rewrapped and redrawn. Ret

alignSelCenter:, alignSelLeft:, setSelProp:to:, setMonoFont:

(NXColor)backgroundColor

Returns the color used to draw the text's background on color displays.

setBackgroundGray:, backgroundGray:, setBackgroundColor:, setTextGray:, textGray, setText
setSelGray:, selGray, setSelColor:

(float)backgroundGray

Returns the gray value used to draw the text's background on monochrome displays.

setBackgroundGray:, setBackgroundColor:, backgroundColor, setTextGray:, textGray, setText
setSelGray:, selGray, setSelColor:

becomeKeyWindow

Activates the caret if it exists. becomeKeyWindow messages are sent by an application's Window receiving a mouse-down event, sends a becomeKeyWindow message to the first responder. You s send this message to a Text object. Returns self.

showCaret, hideCaret, becomeKeyWindow (Window)

(const NXFSM *)breakTable

Returns a pointer to the break table, the finite-state machine table that the Text object uses to deter

setBreakTable:

(int)byteLength

Returns the number of bytes used by the characters in the receiving Text object. The number does terminator (`\0') that getSubstring:start:length: returns if you ask for all the text in a Text object.

In a standard Text object, the number of bytes is equal to the number of characters (thus, this meth same value as charLength or textLength). Subclasses of Text that use more than one byte per char this method to return the number of bytes used to store the text.

charLength, textLength, getSubstring:start:length:

(int)calcLine

Calculates the array of line breaks for the text. The text will then be redrawn if autodisplay is set.

This message should be sent after the Text object's frame is changed. These methods send a calcL their implementation:

initFrame:text:alignment: readText:
read: renewFont:size:style:text:frame:tag:
renewFont:text:frame:tag: setFont:
renewRuns:text:frame:tag: setParaStyle:
setFont:paraStyle: setText:

In addition, if a vertically resizable Text object is the document view of a ScrollView, and the Scro Text object receives a calcLine message. Has no significant return value.

readText:, renewRuns:text:frame:tag:

(void *)calcParagraphStyle:fontId :(int)alignment

Recalculates the default paragraph style given the Font's fontId and alignment. The Text object se you after its font has been changed you will rarely need to send a calcParagraphStyle:: message di pointer to an NXTextStyle structure that describes the default style.

defaultParaStyle

changeTabStopAt:(NXCoord)oldX to:(NXCoord)newX

Moves the tab stop from the receiving Text object's x coordinate oldX to the coordinate newX. Fo
all paragraphs are affected. For a rich Text object, only those paragraphs marked by the selection a
rewrapped and redrawn. Returns self.

setMonoFont:, setSelProp:to:

(const unsigned char *)charCategoryTable

Returns a pointer to the character category table, the table that maps ASCII characters to character

setCharCategoryTable:

(NXCharFilterFunc)charFilter

Returns the character filter function, the function that analyzes each character the user enters. By d
NXEditorFilter().

setCharFilter:

(int)charLength

Returns the number of characters in a Text object. The length doesn't include the null terminator (
start:length: returns if you ask for all the text in a Text object. The charLength and textLength met
related method byteLength returns the length of the text in bytes, which, depending on the number
character, may return a larger value.

byteLength, textLength, getSubstring:start:length:

(BOOL)charWrap

Returns a flag indicating how words whose length exceeds the line length should be treated. If YE
wrapped on a character basis. If NO, long words are truncated at the boundary of the bodyRect.

setCharWrap:

checkSpelling:sender

Searches for a misspelled word in the text of the receiving Text object. The search starts at the cur
continues until it reaches a word suspected of being misspelled or the end of the text. If a word isn
spelling server or listed in the user's local dictionary in ~/.NeXT/LocalDictionary, it's highlighted.
message will then display the Guess panel and allow the user to make a correction or add the word
Returns self.

showGuessPanel:

(const NXFSM *)clickTable

Returns a pointer to the click table, the finite-state machine table that defines word boundaries for

setClickTable:

copy:sender

Copies the selected text from the Text object to the selection pasteboard. The selection remains un
pasteboard receives the text and its corresponding run information. The pasteboard types used are
and NXRTFPboardType.

The sender passes its id as part of the copy: message. Returns self.

cut:, paste:, delete:, copyFont:, pasteFont:, copyRuler:, pasteRuler:

copyFont:sender

Copies font information for the selected text to the font pasteboard. If the selection spans more tha
information copied is that of the first font in the selection. The selection remains unchanged. The
NXFontPboardType.

The sender passes its id as the argument of the copyFont: message. Returns self.

pasteFont:, copyRuler:, pasteRuler:, copy:, cut:, paste:, delete:

copyRuler:sender

Copies ruler information for the paragraph containing the selection to the ruler pasteboard. The se
paragraph boundaries.

The ruler controls a paragraph's text alignment, tab settings, and indentation. If the selection spans
paragraph, the information copied is that of the first paragraph in the selection. The pasteboard typ
NXRulerPboardType.

Once copied to the pasteboard, ruler information can be pasted into another object or application th
data into its document.

The sender passes its id as the argument of the copyRuler: message. Returns self.

pasteRuler:, copyFont:, pasteFont:, copy:, cut:, paste:, delete:

cut:sender

Copies the selected text to the pasteboard and then deletes it from the Text object. The pasteboard
its corresponding font information.

If the Text object's delegate implements the method, it receives a textDidGetKeys:isEmpty: messa
the cut operation. If this is the first change since the Text object became the first responder (and th
the method), a textDidChange: message is also sent to the delegate.

The sender passes its id as part of the cut: message. Returns self.

setAlignment:, setLineHeight:, and setDescentLine:.

setParaStyle:, setAlignment:, setLineHeight:, setDescentLine:

delegate

Returns the Text object's delegate.

setDelegate:

delete:sender

Deletes the selection without adding it to the pasteboard. The sender passes its id as part of the del

If the Text object's delegate implements the method, it receives a textDidGetKeys:isEmpty: messa the delete operation. If this is the first change since the Text object became the first responder (and implements the method), a textDidChange: message is also sent to the delegate.

The delete: method replaces clear:. Returns self.

cut:, copy:, paste:, textDidGetKeys:isEmpty:, textDidChange:

(NXCoord)descentLine

Returns the default descent line for the Text object. The descent line is the distance from the botto the base line of the text.

setDescentLine:

(NXTextFunc)drawFunc

Returns the draw function, the function that's called to draw each line of text. NXDrawALine() is function.

setDrawFunc:, setScanFunc:

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

Draws the Text object. You never send a drawSelf:: message directly, although you may want to change the way a Text object draws itself. Returns self.

drawSelf:: (View)

(BOOL)findText:(const char *)string
ignoreCase:(BOOL)ignoreCaseflag
backwards:(BOOL)backwardsflag

finishReadingRichText

Notifies the Text object that it has finished reading RTF data. The Text object responds by sendin
textWillFinishReadingRichText: message, assuming there is a delegate and it responds to this mes
then perform any required cleanup. Alternatively, a subclass of Text could put these cleanup routi
implementation of this method. Returns self.

(NXTextBlock *)firstTextBlock

Returns a pointer to the first text block. You can traverse this head of the linked list of text blocks
the Text object. In most cases, however, it's better to use the getSubstring:start:length: method to
text or the stream method to get read-only access to the entire contents of the Text object.

getSubstring:start:length:, stream

font

Returns the Font object for a plain Text object. For rich Text objects, the Font object for the first t

setFont:

free

Releases the storage for a Text object.

free (View)

getLocation:(NXPoint *)origin ofCell:cell

Places the x and y coordinates of cell in the NXPoint structure specified by origin. The coordinate
object's coordinate system. cell is a Cell object that's displayed as part of the text.

Returns nil if the Cell object isn't part of the text otherwise, returns self.

replaceSelWithCell:, setLocation:ofCell:, calcCellSize: (Cell)

getMarginLeft:(NXCoord *)leftMargin
right:(NXCoord *)rightMargin
top:(NXCoord *)topMargin
bottom:(NXCoord *)bottomMargin

setMaxSize:, getMinSize:

getMinSize:(NXSize *)theSize

Copies the minimum size of the Text object into the structure referred to by theSize. Returns self.

setMinSize:, getMaxSize:

getMinWidth:(NXCoord *)width
minHeight:(NXCoord *)height
maxWidth:(NXCoord)widthMax
maxHeight:(NXCoord)heightMax

Calculates the minimum width and height needed to contain the text. Given a maximum width and
heightMax), this method copies the minimum width and height to the addresses pointed to by the v
arguments. This method doesn't rewrap the text. To get the absolute minimum dimensions of the
getMinWidth:minHeight:maxWidth:maxHeight: message only after sending a calcLine message.

The values derived by this method are accurate only if the Text object hasn't been scaled. Returns

sizeToFit

getParagraph:(int)prNumber
start:(int *)startPos
end:(int *)endPos
rect:(NXRect *)paragraphRect

Copies the positions of the first and last characters of the specified paragraph to the addresses start
copies the paragraph's bounding rectangle into the structure referred to by paragraphRect. A parag
character the first paragraph is paragraph 0, the second is paragraph 1, and so on. Returns self.

getSubstring:start:length:, firstTextBlock

getSel:(NXSelPt *)start :(NXSelPt *)end

Copies the starting and ending character positions of the selection into the addresses referred to by
points to the beginning of the selection end points to the end of the selection. Returns self.

setSel::

(int)getSubstring:(char *)buf
start:(int)startPos
length:(int)numChars

Copies a substring of the text to a specified memory location. The substring is specified by startPo
is the position of the first character of the substring numChars is the number of characters to be co
address of the memory location for the substring. getSubstring:start:length: returns the number of

hideCaret

Removes the caret from the text. The Text object sends itself hideCaret messages whenever the di
would be inappropriate you rarely need to send a hideCaret message directly. Occasions when the
sent include whenever the Text object receives a resignKeyWindow, mouseDown:, or keyDown: n

showCaret

initFrame:(const NXRect *)frameRect

Initializes a new Text object. This method invokes the initFrame:text:alignment: method with the
specified by frameRect. Text alignment is set to NX_LEFTALIGNED. Returns self.

initFrame:text:alignment:

initFrame:(const NXRect *)frameRect
text:(const char *)theText
alignment:(int)mode

Initializes a new Text object. This is the designated initializer for Text objects: If you subclass Te
designated initializer must maintain the initializer chain by sending a message to super to invoke th
introduction to the class specifications for more information.

The three arguments specify the Text object's frame rectangle, its text, and the alignment of the tex
argument specifies the Text object's location and size in its superview's coordinates. A Text objec
a flipped view that's neither scaled nor rotated. The second argument, theText, is a null-terminate
The mode argument determines how the text is drawn with respect to the bodyRect:

initFrame:

(BOOL)isEditable

Returns YES if the text can be edited, NO if not. The default value is YES.

isSelectable, setDelegate:

(BOOL)isFontPanelEnabled

Returns YES if the Text object will respond to the Font panel, NO if not. The default value is YES

(BOOL)isHorizResizable

Returns YES if the text can automatically change size horizontally, NO if not. The default value is

setVertResizable:, isVertResizable, setHorizResizable:

(BOOL)isMonoFont

Returns YES if the Text object permits only one font and paragraph style for its text, NO if not. T
YES.

setMonoFont:

(BOOL)isRetainedWhileDrawing

Returns YES if the Text object automatically changes its window's buffering type from buffered to
redraws itself, NO if not.

setRetainedWhileDrawing:, drawSelf::

(BOOL)isRulerVisible

Returns YES if the ruler is visible in the Text object's superview, a ScrollView otherwise, returns

toggleRuler:

(BOOL)isSelectable

Returns YES if the text can be selected, NO if not. The default value is YES.

isEditable, setDelegate:

(BOOL)isVertResizable

Returns YES if the text can automatically change size vertically, NO if not. The default value is N

setVertResizable:, setHorizResizable:, isHorizResizable

keyDown:(NXEvent *)theEvent

Analyzes key-down events received by the Text object. keyDown: first uses the Text object's char
determine whether the event should be interpreted as a command to move the cursor or as a comma
object's status as the first responder. If the latter, the Text object's delegate is given an opportunity

(int)lineFromPosition:(int)position

Returns the line number that contains the character at position. To get more information about the object, use the stream returned by the stream method to read the contents of the Text object.

positionFromLine:, stream

(NXCoord)lineHeight

Returns the default line height for the Text object.

setLineHeight:

mouseDown:(NXEvent *)theEvent

Responds to mouse-down events. When a Text object that allows selection receives a mouseDown mouse-dragged events and responds by adjusting the selection and autoscrolling, if necessary. You message, though you may want to override it.

setEditable:, setDelegate:, getNextEvent:waitFor: (Application)

moveCaret:(unsigned short)theKey

Moves the caret either left, right, up, or down if theKey is NX_LEFT, NX_RIGHT, NX_UP, or NX isn't one of these four values, the caret doesn't move. Returns self.

keyDown:

moveTo:(NXCoord)x :(NXCoord)y

Moves the origin of the Text object's frame rectangle to (x, y) in its superview's coordinates. Retu

moveTo:: (View)

(int)offsetFromPosition:(int)charPosition

Returns the byte offset corresponding to the character position charPosition in the Text object's tex software release, where each character is represented by a byte, a character's position and its byte o

positionFromOffset:, positionFromLine:, lineFromPosition:

(NXRTFDError)openRTFDFrom:(const char *)path

Opens the RTFD file package specified by path. The last element in the path must be the name of directoryÐfor example, ª/tmp/MyFile.rtfdºÐnot the name of the RTF document within the direct Text object's contents are replaced with the text and images found in the file package, and the new

Before the paste operation, a textDidChange: message is sent to the delegate, assuming that this is the Text object became the first responder and that the delegate implements the method. After the delegate receives a textDidGetKeys:isEmpty: message, if it implements the method.

sender is the id of the sending object. paste: returns nil if the pasteboard can provide neither NXA NXRTFPboardType format types otherwise, returns self.

copy:, cut:, delete:, copyFont:, copyRuler:, pasteFont:, pasteRuler:, textDidGetKeys:isEmpty:

pasteFont:sender

Takes font information from the font pasteboard and applies it to the current selection. If the selec those characters subsequently entered at the insertion point are affected.

pasteFont: works only with rich Text objects (see setMonoFont:). Attempting to paste a font into a generates a system beep without altering any fonts.

Before the paste operation, a textDidChange: message is sent to the delegate, assuming that this is the Text object became the first responder and that the delegate implements the method. After the delegate receives a textDidGetKeys:isEmpty: message, if it implements the method.

sender is the id of the sending object. After the font is pasted, the text is rewrapped and redrawn. the pasteboard has no data of the type NXFontPboardType otherwise, returns self.

copyFont:, copyRuler:, pasteRuler:, copy:, cut:, delete:, paste:, setMonoFont:, textDidGetKey textDidChange:

pasteRuler:sender

Takes ruler information from the ruler pasteboard and applies it to the paragraph or paragraphs ma selection. The ruler controls a paragraph's text alignment, tab settings, and indentation.

pasteRuler: works only with rich Text objects (see setMonoFont:). Attempting to paste a ruler into generates a system beep without altering any ruler settings.

Before the paste operation, a textDidChange: message is sent to the delegate, assuming that this is the Text object became the first responder and that the delegate implements the method. After the delegate receives a textDidGetKeys:isEmpty: message, if it implements the method.

sender is the id of the sending object. After the ruler is pasted, the text is rewrapped and redrawn. it's also updated. pasteRuler: returns nil if the pasteboard has no data of the type NXRulerPboardT self.

copyRuler:, copyFont:, pasteFont:, copy:, cut:, delete:, paste:, setMonoFont:- textDidGetKeys textDidChange:

(int)positionFromLine:(int)line

Returns the character position of the line numbered line. Each line is terminated by a Return chara in a Text object is line 1. To find the length of a line, you can send the positionFromLine: message lines, and use the difference of the two to get the line length. To get more information about the co object, use the stream returned by the stream method to read the contents of the Text object.

lineFromPosition:, stream

offsetFromPosition:, positionFromLine:, lineFromPosition:

(const unsigned char *)postSelSmartTable

Returns a pointer to the table that specifies which characters on the right end of a selection are trea
space character.

setPostSelSmartTable:, setPreSelSmartTable:, preSelSmartTable

(const unsigned char *)preSelSmartTable

Returns a pointer to the table that specifies which characters on the left end of a selection are treate
space character.

setPreSelSmartTable:, setPostSelSmartTable:, postSelSmartTable

read:(NXTypedStream *)stream

Reads the Text object in from the typed stream stream. A read: message is sent in response to arch
this message directly. Returns self.

readRichText:(NXStream *)stream

Reads RTF text from stream into the Text object and formats the text accordingly. The Text objec
enough for all the text to be visible. Returns self.

writeRichText:

readRichText:(NXStream *)stream atPosition:(int)position

Reads RTF text from stream into the Text object's text at position and formats the text accordingly
message, but may want to override it to read special RTF directives while the Text object is readin
self.

readRTFDFrom:(NXStream *)stream

Reads the RTFD data contained in stream. The Text object's contents are replaced with the text ar
stream, and the new contents are displayed. Returns self if the data is successfully read from the st
nil.

openRTFDFrom:, replaceSelWithRTFD:, writeRTFDSelectionTo:, writeRTFDTo:

readText:(NXStream *)stream

Reads new text into the Text object from stream. All previous text is deleted. The Text object wra
text if autodisplay is enabled. This method doesn't affect the object's frame or bounds rectangle.
rectangle to make the text entirely visible, use the sizeToFit method. Returns self. This method ra
NX_textBadRead exception if an error occurs while reading from stream.

setSel::, setText:, readRichText:, sizeToFit

renewFont:(const char *)newFontName
size:(float)newFontSize
style:(int)newFontStyle
text:(const char *)newText
frame:(const NXRect *)newFrame
tag:(int)newTag

Resets a Text object so that it can be reused to draw or edit another piece of text. If newText is NU
same as the previous text. newTag sets the Text object's tag. A font object is created with newFo
and newFontStyle. This method is a convenient cover for the renewRuns:text:frame:tag: method.

renewRuns:text:frame:tag:, setText:

renewFont:newFontId
text:(const char *)newText
frame:(const NXRect *)newFrame
tag:(int)newTag

Resets a Text object so that it can be reused to draw or edit another piece of text. If newText is NU
same as the previous text. newTag sets a Text object's tag. This method is a convenient cover for
frame:tag: method. Returns self.

setText:

renewRuns:(NXRunArray *)newRuns
text:(const char *)newText
frame:(const NXRect *)newFrame
tag:(int)newTag

Resets a Text object so that it can be reused to draw or edit another piece of text. If newRuns is N
the same runs as the previous text. If newText is NULL, the new text is the same as the previous t
Text object's tag. Returns self.

setText:

replaceSel:(const char *)aString

Replaces the current selection with text from aString, a null-terminated character string, and then r
the text. Returns self.

replaceSel:(const char *)aString
length:(int)length
runs:(NXRunArray *)insertRuns

Replaces the current selection with length characters of text from aString, using insertRuns to desc
Another way to replace the selection with multiple-run text is with replaceSelWithRichText:.

After replacing the selection, this method rewraps and redisplays the text. Returns self.

replaceSel:, replaceSelWithRichText:

replaceSelWithCell:cell

Replaces the current selection with the image provided by cell. This method works only with rich
setMonoFont:.)

The image is treated like a single character. Its height and width are determined by sending the Ce
message. The height determines the line height of the line containing the image, and the width sets
placement in the line. The image is drawn by sending the Cell a drawSelf:inView: message.

After receiving a replaceSelWithCell: message, a Text object rewraps and redisplays its contents.

setMonoFont:,