BecomeKeyOnlyIfNeeded:(BOOL)flagSets whether Panel waits to become key window
OOL)doesBecomeKeyOnlyIfNeededReturns whether Panel waits to become key window
FloatingPanel:(BOOL)flagSets whether the Panel floats above other windows
OOL)isFloatingPanelReturns whether the Panel floats above other windows
WorksWhenModal:(BOOL)flagSets whether the Panel can operate on an attention panel
OOL)worksWhenModalReturns whether Panel can operate on an attention panel

eReleases the Pasteboard object's storage
eGloballyFrees the object and the domain for its name

nst char *)nameReturns the Pasteboard object's name

clareTypes:(const char *const *)newTypesSets data types and owner of the Pasteboard
    num:(int)numTypes
    owner:newOwner
t)addTypes:(const char *const *)newTypesAddsdata types to the pasteboard
    num:(int)numTypes
    owner:newOwner
teType:(const char *)dataTypeWrites theData to the pasteboard server
    data:(const char *)theData
    length:(int)numBytes
teType:(const char *)dataTypeWrites stream data to the pasteboard server
    fromStream:(NXStream *)stream
OOL)writeFileContents:Writesdata from filename to the pasteboard server
    (const char *)filename

nst NXAtom *)typesReturns an array of the Pasteboard's data types
nst char *)findAvailableTypeFrom:Returns first type in types that matches a pasteboard
    (const char *const *)typestype

toFile:(const char *)filename
llocatePasteboardData:(char *)dataDeallocates data received from the pasteboard
length:(int)numBytes

steboard:sender Implemented to write promised data to sender as type
provideData:(NXAtom)type
steboardChangedOwner:sender Notifies prior owner that ownership changed

tInitializes a new PopUpList

dItem:(const char *)titleAdds an item with title as its title to the end of the list
ertItem:(const char *)titleInserts an item with title as its title at position index
at:(unsigned int)index
noveItem:(const char *)titleRemoves the item matching title
noveItemAt:(unsigned int)indexRemoves the item at the specified index
t)indexOfItem:(const char *)titleReturns the index of the item matching title
signed int)countReturns the number of items in the list

ngeButtonTitle:(BOOL)flagSets whether the PopUpList is a pop-up or a pull-down list
ButtonFrame:(NXRect *)bFrameGets the size needed for the Button that pops up the list

pUp:triggerPops the list up over trigger

nst char *)selectedItemReturns the title of selected item

Font:fontObjectSets the Font used to draw the items
tReturns the Font used to draw the items

Action:(SEL)aSelectorSets the PopUpList's action method to aSelector
EL)actionReturns the PopUpList's action method
Target:anObjectSets the PopUpList's target object to anObject
getReturns the PopUpList's target object

tInitializes the PrintInfo instance after it's allocated
eDeallocates the PrintInfo object

MarginLeft:(NXCoord)leftMarginSets the margins
right:(NXCoord)rightMargin
top:(NXCoord)topMargin
bottom:(NXCoord)bottomMargin
MarginLeft:(NXCoord *)leftMarginReturns the margins by reference
right:(NXCoord *)rightMargin
top:(NXCoord *)topMargin
bottom:(NXCoord *)bottomMargin
Orientation:(char)modeSets the orientation as portrait or landscape
andAdjust:(BOOL)flag
ar)orientationReturns the orientation is portrait or landscape
PaperRect:(const NXRect *)aRectSets the width and height of the paper
andAdjust:(BOOL)flag
nst NXRect *)paperRectReturns the rectangle for the paper size
PaperType:(const char *)typeSets the paper type
andAdjust:(BOOL)flag
nst char *)paperTypeReturns the paper type

FirstPage:(int)anIntSets the page number of first page to be printed
t)firstPageReturns the page number of the first page to be printed
LastPage:(int)anIntSets the page number of last page to be printed
t)lastPageReturns the page number of the last page to be printed
AllPages:(BOOL)flagSets whether all the pages are to be printed
OOL)isAllPagesReturns whether all the pages are to be printed
t)currentPageReturns the page number of the page being printed

HorizPagination:(int)modeSets the horizontal pagination mode
t)horizPaginationReturns the horizontal pagination mode
VertPagination:(int)modeSets the vertical pagination mode
t)vertPaginationReturns the vertical pagination mode
ScalingFactor:(float)aFloatSets the scaling factor
oat)scalingFactorReturns the scaling factor

HorizCentered:(BOOL)flagSets whether the image is centered horizontally
OOL)isHorizCenteredReturns whether the image is centered horizontally
VertCentered:(BOOL)flagSets whether the image is centered vertically
OOL)isVertCenteredReturns whether the image is centered vertically
PagesPerSheet:(short)aShortSets the number of pages printed per sheet of paper
ort)pagesPerSheetReturns the number of pages printed per sheet of paper

nst char **)jobFeaturesReturns the keys to the job features table
PageOrder:(char)modeSets the order in which pages will be printed
ar)pageOrderReturns the order in which pages will be printed
ReversePageOrder:(BOOL)flagSets whether the page order is reversed
OOL)reversePageOrderReturns whether the page order is reversed
Copies:(int)anIntSets the number of copies to be printed
t)copiesReturns the number of copies to be printed
PaperFeed:(const char *)paperFeedSlotSets the paper feed slot used during printing
nst char *)paperFeedReturns the paper feed slot used during printing

Printer:(NXPrinter *)aPrinterSets the printer that's used in subsequent printing jobs
XPrinter *)printerReturns the NXPrinter that's used for printing

OutputFile:(const char *)aStringSets the output file for printing
nst char *)outputFileReturns the output file for printing
Context:(DPSContext)aContextSets the DPS context used for printing
PSContext)contextReturns the DPS context used for printing

d:(NXTypedStream *)streamReads the PrintInfo from the typed stream
te:(NXTypedStream *)streamWrites the PrintInfo to the typed stream

eDeallocates the PrintPanel

AccessoryView:aViewAdds a View to the panel
cessoryViewReturns the accessory View

t)runModalDisplays the Print panel and begins its event loop

dateFromPrintInfoReads PrintPanel's values from the PrintInfo object
alWritePrintInfoWrites PrintPanel's values to the PrintInfo object

NextResponder:aResponderMakes aResponder the receiver's next responder
xtResponderReturns the receiver's next responder

OOL)acceptsFirstResponderReturns NO to refuse first responder status
comeFirstResponderNotifies the receiver it's the first responder
ignFirstResponderNotifies the receiver it's not the first responder

OOL)performKeyEquivalent:(NXEvent *)theEvent
Returns NO to indicate theEvent isn't handled
OOL)tryToPerform:(SEL)anActionAids in dispatching action messages
with:anObject

useDown:(NXEvent *)theEventPasses the message to the receiver's next responder
ntMouseDown:(NXEvent *)theEventPasses the message to the receiver's next responder
useDragged:(NXEvent *)theEventPasses the message to the receiver's next responder
ntMouseDragged:(NXEvent *)theEventPasses the message to the receiver's next responder
useUp:(NXEvent *)theEventPasses the message to the receiver's next responder
ntMouseUp:(NXEvent *)theEventPasses the message to the receiver's next responder
useMoved:(NXEvent *)theEventPasses the message to the receiver's next responder
useEntered:(NXEvent *)theEventPasses the message to the receiver's next responder
useExited:(NXEvent *)theEventPasses the message to the receiver's next responder
yDown:(NXEvent *)theEventPasses the message to the receiver's next responder
yUp:(NXEvent *)theEventPasses the message to the receiver's next responder
gsChanged:(NXEvent *)theEventPasses the message to the receiver's next responder
ResponderFor:(const char *)eventTypePrints warning message to syslog if debugging

idRequestorForSendType:(NXAtom)typeSent
andReturnType:(NXAtom)typeReturnedImplemented by subclasses to determine available services

d:(NXTypedStream *)streamReads the Responder from the typed stream stream
te:(NXTypedStream *)streamWrites the Responder to the typed stream stream

eDeallocates the SavePanel

AccessoryView:aViewAdds application-customized view to the panel
essoryViewReturns the application-customized view
Title:(const char *)titleSets the title of the SavePanel to title
Prompt:(const char *)promptSets the title of the file name form field

Directory:(const char *)pathSets the current directory of the SavePanel
RequiredFileType:(const char *)typeSets the required file type (if any)
nst char *)requiredFileTypeGets the required file type (if any)

t)runModalForDirectory:(const char *)pathDisplays the SavePanel and begins its event loop
file:(const char *)name
t)runModalDisplays the SavePanel and begins its event loop

nst char *)directoryReturns directory chosen file resides in
nst char *)filenameReturns full name of file to be saved

OOL)commandKey:(NXEvent *)theEventEnables command-space to do filename completion

senderMethod invoked by the OK button
ncel:senderMethod invoked by the Cancel button

ectText:senderCalled when TAB is pressed in the form
tDidEnd:textObjectDetermines whether TAB or BACKTAB was pressed
endChar:(unsigned short)endChar
tDidGetKeys:textObj isEmpty:(BOOL)flagDetermines whether there's any text in the form

Delegate:anObject Makes anObject the SavePanel's delegate

inDirectory:(const char *)directory
OOL)panelValidateFilenames:senderYES if the filename is acceptable to the delegate

tFrame:(const NXRect *)frameRectInitializes a new Scroller

XRect *)calcRect:(NXRect *)aRectGets the rectangle that encloses partCode
forPart:(int)partCode
eckSpaceForPartsChecks for room for knob and scroll buttons
ArrowsPosition:(int)whereSets position of scroll buttons in Scroller

oat)floatValueReturns Scroller's float value
FloatValue:(float)aFloatSets value positions knob
FloatValue:(float)aFloat :(float)percentSets value positions and sizes knob

eTo:(NXCoord)width :(NXCoord)heightSizes the Scroller

wArrow:(BOOL)whichButton :(BOOL)flagDraws highlighted and unhighlighted arrows
wKnobDraws the knob
wPartsCaches Bitmaps for knob and scroll arrows
wSelf:(const NXRect *)rects :(int)rectCountDraws the Scroller
hlight:(BOOL)flagHighlights scroll button that's under mouse

Action:(SEL)aSelectorSets the Scroller's action to aSelector
EL)actionReturns the Scroller's action
Target:anObjectSets the Scroller's target to anObject
getReturns the Scroller's target

OOL)acceptsFirstMouseMakes the Scroller respond to the first mouse event
t)hitPartReturns Scroller part that received mouse-down
useDown:(NXEvent *)theEventResponds to mouse-down events
t)testPart:(const NXPoint *)thePointReturns Scroller part that's under thePoint
ckKnob:(NXEvent *)theEventResponds to mouse-down events on the knob
ckScrollButtons:(NXEvent *)theEventResponds to mouse-down events on buttons

tFrame:(const NXRect *)frameRectInitializes a new ScrollView

ContentSize:(NXSize *)contentViewSizeGets the content view's size
DocVisibleRect:(NXRect *)aRectGets the visible portion of the document view

izeSubviews:(const NXSize *)oldSizeRetiles the ScrollView after a sizeTo::
HorizScrollerRequired:(BOOL)flagMakes space for a horizontal scroller
VertScrollerRequired:(BOOL)flagMakes space for a vertical scroller
Retiles the scrollers and content view

DocView:aViewMakes aView the ScrollView's document view
cViewReturns the current document view
HorizScroller:anObjectSets the horizontal Scroller object
izScrollerReturns the horizontal Scroller
VertScroller:anObjectSets the vertical Scroller object
tScrollerReturns the vertical Scroller
lectScroll:cViewUpdates the Scrollers

BorderType:(int)aTypeDetermines the border type of the ScrollView
t)borderTypeReturns the border type
BackgroundColor:(NXColor)colorSets the ScrollView's background color
XColor) backgroundColorReturns the ScrollView's background color
BackgroundGray:(float)valueSets the ScrollView's background gray
oat)backgroundGrayReturns the ScrollView's background gray

CopyOnScroll:(BOOL)flagSets how newly exposed areas are redrawn
DisplayOnScroll:(BOOL)flagSets how the doc view is displayed during scrolling

DocCursor:anObjSets the cursor for the document view

d:(NXTypedStream *)streamReads the ScrollView from the typed stream
te:(NXTypedStream *)streamWrites the ScrollView to the typed stream

Initializes a new SelectionCell with ªListItemº as its title
TextCell:(const char *)aStringInitializes a new SelectionCell with aString as its title

cCellSize:(NXSize *)theSizeCalculates the size of the SelectionCell within aRect
inRect:(const NXRect *)aRect

Leaf:(BOOL)flagSets whether SelectionCell is a leaf or a branch
OOL)isLeafReturns whether the SelectionCell is a leaf or a branch
OOL)isOpaqueReturns YES, since SelectionCells are opaque

wSelf:(const NXRect *)cellFrameDraws the SelectionCell in cellFrame within aView
inView:aView
wInside:(const NXRect *)cellFrameDraws the inside of the SelectionCell in aView
inView:aView
hlight:(const NXRect *)cellFrameHighlights the SelectionCell within cellFrame in
inView:aView controlView
lit:(BOOL)flag

akeReinitializes the SelectionCell when it's unarchived

tFrame:(const NXRect *)frameRectInitializes a new Slider in frameRect

TitleCell:aCellSets the Cell used to draw the background title
eCellReturns the Cell used to draw the background title
TitleFont:fontObjectSets the Font used to draw the background title
eFontReturns the Font used to draw the background title
TitleColor:(NXColor)aColorSets the color of text in the background title to aColor
XColor)titleColorReturns the color of text in the background title
TitleGray:(float)aFloatSets the gray of text in the background title to aFloat
oat)titleGrayReturns the gray of text in the background title
t)isVerticalReturns 1 if vertical, 0 if horizontal, 1 if unknown

MinValue:(double)aDoubleSets the Slider's minimum value to aDouble
uble)minValueReturns the Slider's minimum value
MaxValue:(double)aDoubleSets the Slider's maximum value to aDouble
uble)maxValueReturns the Slider's maximum value

eToFitModifies the Slider's size to fit its Cell

OOL)acceptsFirstMouseReturns YES, since Sliders always accept first mouse
Enabled:(BOOL)flagSets whether the Slider reacts to events
useDown:(NXEvent *)theEventResponds to mouse-down by initiating tracking

tInitializes a new SliderCell

cCellSize:(NXSize *)theSizeReturns the size of the SliderCell
inRect:(const NXRect *)aRect
KnobRect:(NXRect*)knobRectGets the rectangle the knob will be drawn in
flipped:(BOOL)flipped

MinValue:(double)aDoubleSets the SliderCell's minimum value to aDouble
uble)minValueReturns the SliderCell's minimum value
MaxValue:(double)aDoubleSets the maximum value of the SliderCell to aDouble
uble)maxValueReturns the SliderCell's maximum value

DoubleValue:(double)aDoubleSets the SliderCell's value to aDouble

KnobThickness:(NXCoord)aFloatSets the knob's thickness to aFloat
XCoord)knobThicknessReturns the knob's thickness
Image:imageSets the background image to image
ageReturns the background image
Title:(const char *)aStringSets the background title to a copy of aString
TitleNoCopy:(const char *)aStringSets the background title to aString
nst char *)titleReturns the background title
TitleCell:aCellSets the Cell used to draw the background title
eCellReturns the Cell used to draw the background title
TitleFont:fontObjectSets the Font used to draw the background title
eFontReturns the Font used to draw the background title
TitleColor:(NXColor)aColorSets the color of text in the background title to aColor
XColor)titleColorReturns the color of text in the background title
TitleGray:(float)aFloatSets the gray of text in the background title to aFloat
oat)titleGrayReturns the gray of text in the background title
OOL)isOpaqueReturns YES (SliderCells are always opaque)
t)isVerticalReturns 1 if vertical, 0 if horizontal, 1 if unknown

wSelf:(const NXRect *)cellFrameDraws the SliderCell's bar and knob in controlView
inView:controlView
wInside:(const NXRect *)cellFrameDraws the inside of the SliderCell in controlView
inView:controlView
wBarInside:(const NXRect *)aRectDraws the SliderCell's bar
flipped:(BOOL)flipped
wKnobDraws the SliderCell's knob
wKnob:(const NXRect*)knobRectDraws the SliderCell's knob in knobRect

Enabled:(BOOL)flagSets whether the SliderCell reacts to events
Continuous:(BOOL)flagSets whether the Slider is continuous
OOL)isContinuousReturns whether the Slider is continuous
AltIncrementValue:(double)incValueSets how far the SliderCell moves when the knob is dragged one pixel w
held down
uble)altIncrementValueReturns how far the SliderCell moves when alt-dragged

OOL)trackMouse:(NXEvent *)theEventTracks the mouse
inRect:(const NXRect *)cellFrame
ofView:controlView
OOL)startTrackingAt:(const NXPoint *)startPoint
inView:controlViewBegins a tracking session
OOL)continueTracking:(const NXPoint *)lastPoint
at:(const NXPoint *)currentPointContinues tracking the mouse

te:(NXTypedStream *)streamWrites the SliderCell to stream
akeCaches knob icons when the SliderCell is unarchived

tInitializes the Speaker after it has been allocated

eDeallocates the Speaker (but not its ports)

SendTimeout:(int)msSets how long to wait for messages to be delivered
t)sendTimeoutReturns how long to wait for messages to be delivered
ReplyTimeout:(int)msSets how long Speaker will wait for a reply
t)replyTimeoutReturns how long Speaker will wait for a reply

SendPort:(port_t)aPortMakes aPort the port messages will be sent to
rt_t)sendPortReturns the port the Speaker will send messages to
ReplyPort:(port_t)aPortMakes aPort the port where replies are received
rt_t)replyPortReturns the port where Speaker receives replies

t)openFile:(const char *)fullPathSends a remote message to open fullPath file
ok:(int *)flag
t)openTempFile:(const char *)fullPathSends a remote message to open fullPath file
ok:(int *)flag

t)msgCalc:(int *)flagSends message to update the current window
t)msgCopyAsType:(const char *)aTypeSends message to copy selection as aType data
ok:(int *)flag
t)msgCutAsType:(const char *)aTypeSends message to cut selection as aType data
ok:(int *)flag
t)msgDirectory:(char *const *)fullPathSends message requesting the current directory
ok:(int *)flag
t)msgFile:(char *const *)fullPathSends message requesting the current document
ok:(int *)flag
t)msgPaste:(int *)flagSends message to paste data from pasteboard
t)msgPosition:(char *const *)aStringSends message requesting selection information
posType:(int *)anInt
ok:(int *)flag
t)msgPrint:(const char *)fullPathSends message to print fullPath file

ok:(int *)flag
t)msgVersion:(char *const *)aStringSends message requesting version information
ok:(int *)flag

t)performRemoteMethod:(const char *)methodName
Sends remote methodName message
t)performRemoteMethod:(const char *)methodName
with:(const char *)dataSends remote message with numBytes of data
length:(int)numBytes
t)selectorRPC:(const char *)methodNameSends remote message with variable arguments
paramTypes:(char *)params,
t)sendOpenFileMsg:(const char *)fullPathSends an openFile:ok: remote message
ok:(int *)flag
andDeactivateSelf:(BOOL)deactivateFirst
t)sendOpenTempFileMsg:(const char *)fullPath
ok:(int *)flagSends an openTempFile:ok: remote message
andDeactivateSelf:(BOOL)deactivateFirst

Delegate:anObjectMakes anObject the Speaker's delegate
egateReturns the Speaker's delegate

d:(NXTypedStream *)streamReads the Speaker from stream
te:(NXTypedStream *)streamWrites the Speaker to stream

tFrame:(const NXRect *)frameRectInitialize a new Text object
tFrame:(const NXRect *)frameRectInitialize a new Text object
text:(const char *)theText

MaxSize:(NXSize *)theSizeGets maximum size of the Text object
MinSize:(const NXSize *)newMinSizeSets minimum size of the Text object
MinSize:(NXSize *)theSizeGets minimum size of the Text object
VertResizable:(BOOL)flagSets whether frame height can change
OOL)isVertResizableReturns whether frame height can change
HorizResizable:(BOOL)flagSets whether frame width can change
OOL)isHorizResizableReturns whether frame width can change
eTo:(NXCoord)width :(NXCoord)heightResizes the Text object to width and height
eToFitResizes the frame to accommodate the text
izeText:(const NXRect *)oldBoundsUsed by Text object to resize and redisplay itself
:(const NXRect *)maxRect
veTo:(NXCoord)x :(NXCoord)yMoves the Text object to (x, y)

MarginLeft:(NXCoord)leftMarginAdjusts margins around the text
right:(NXCoord)rightMargin
top:(NXCoord)topMargin
bottom:(NXCoord)bottomMargin
MarginLeft:(NXCoord *)leftMarginGets dimensions of margins around the text
right:(NXCoord *)rightMargin
top:(NXCoord *)topMargin
bottom:(NXCoord *)bottomMargin
MinWidth:(NXCoord *)widthCalculates area needed to display the text
minHeight:(NXCoord *)height
maxWidth:(NXCoord)widthMax
maxHeight:(NXCoord)heightMax
Alignment:(int)modeSets how text is aligned at margins
t)alignmentReturns how text is aligned at margins
gnSelLeft:senderAligns the text to the left margin
gnSelCenter:senderAligns the text between the margins
gnSelRight:senderAligns the text to the right margin
SelProp:(NXParagraphProp)propSets the paragraph style for one or more paragraphs
to:(NXCoord)val
ngeTabStopAt:(NXCoord)oldXResets the position of the specified tab stop
to:(NXCoord)newX
t)calcLineCalculates line breaks
CharWrap:(BOOL)flagReturns whether extra long words are wrapped
OOL)charWrapSets whether extra long words are wrapped
NoWrapDisables word wrap
ParaStyle:(void *)paraStyleSets paragraph style for the entire text
id *)defaultParaStyleReturns the default paragraph style
id *)calcParagraphStyle:fontIdRecalculates paragraph style
:(int)alignment
LineHeight:(NXCoord)valueSets height of a line of text
XCoord)lineHeightReturns height of a line of text
DescentLine:(NXCoord)valueSets distance from base line to bottom of line
XCoord)descentLineReturns distance from base line to bottom of line

dText:(NXStream *)streamReplaces current text with text from stream
rtReadingRichTextSent before Text object begins reading RTF data
dRichText:(NXStream *)streamReplaces text with RTF data from stream
dRichText:(NXStream *)streamLets you add RTF data to stream
atPosition:(int)position
ishReadingRichTextSent after Text object reads RTF data
XRTFDError)openRTFDFrom:(const char *)path
Opens the RTFD file package specified by path
XRTFDError)saveRTFDTo:(const char *)path
removeBackup:(BOOL)removeBackupSaves the contents (text and images) of the Text object
errorHandler:errorHandlerto the file package specified by path
teText:(NXStream *)streamWrites all the text to stream
teRichText:(NXStream *)streamWrites all the text to stream using RTF
teRichText:(NXStream *)streamWrites text to stream using RTF
from:(int)start
to:(int)end
teRTFDSelectionTo:(NXStream *)streamWrites the selectionÐtext and imagesÐto stream
teRTFDTo:(NXStream *)streamWrites all the text and images to stream
XStream *)streamReturns stream access to Text object's text
XTextBlock *)firstTextBlockReturns pointer to first text block
Paragraph:(int)prNumberGets position, length, and size of a paragraph
start:(int *)startPos
end:(int *)endPos
rect:(NXRect *)paragraphRect
t)getSubstring:(char *)bufCopies numChars at startPos to buf
start:(int)startPos
length:(int)numChars
t)byteLengthReturns length of the Text object's contents in bytes
t)charLengthReturns number of characters in the text
t)textLengthReturns number of characters in the text

Editable:(BOOL)flagSets whether the text can be edited
OOL)isEditableReturns whether the text can be edited

MonoFont:(BOOL)flagControls whether multiple fonts and parastyles are OK
OOL)isMonoFontReturns whether only one font and parastyle is permitted

by:senderCopies selected text to the pasteboard
byFont:senderCopies selected text's font to the pasteboard
byRuler:senderCopies selected text's style to the pasteboard
ste:senderReplaces selection with pasteboard's contents
steFont:senderReplaces selection's font with pasteboard's contents

bscript:senderSubscripts the current selection
perscript:senderSuperscripts the current selection
script:senderRemoves sub/super script in the current selection
derline:senderToggles the underline attribute of text
wCaretDisplays the previously hidden caret
eCaretRemoves the caret from the text display
Selectable:(BOOL)flagSets whether the text can be selected
OOL)isSelectableReturns whether the text can be selected
ectErrorSelects all the text
ectNullDeselects the current selection
Sel:(int)start :(int)endSelects text from start through end
Sel:(NXSelPt *)start :(NXSelPt *)endGets start and end of the selection
laceSel:(const char *)aStringReplaces the selection with aString
laceSel:(const char *)aStringReplaces selection with length bytes of aString
length:(int)length
laceSel:(const char *)aStringReplaces selection with length bytes of aString
length:(int)length
runs:(NXRunArray *)insertRuns
laceSelWithRichText:(NXStream *)streamReplaces selection with RTF from stream
laceSelWithRTFD:(NXStream *)streamReplaces selection with RTFD data from stream
ollSelToVisibleBrings the selection within the frame rectangle

FontPanelEnabled:(BOOL)flagSets whether the Font panel can affect text
OOL)isFontPanelEnabledSets whether the Font panel can affect text
ngeFont:senderChanges font of selection
Font:fontObjSets Font object for the entire text
tReturns a monofont Text object's font
Font:fontObj paraStyle:(void *)paraStyleSets Font and paragraph style for all text
SelFont:fontIdSets Font object for the selection
SelFontFamily:(const char *)fontNameSets font family for the selection
SelFontSize:(float)sizeSets font size for the selection
SelFontStyle:(NXFontTraitMask)traitsSets font style for the selection
SelFont:fontId paraStyle:(void *)paraStyleSets font and paragraph style for the selection

eckSpelling:senderSearches for a misspelled word in the text
wGuessPanel:senderDisplays panel suggesting spelling corrections

gleRuler:senderControls the display of the ruler
XColor)isRulerVisibleReturns whether the ruler is visible in the superview

oat)backgroundGrayReturns the gray value of the text background
BackgroundColor:(NXColor)colorSets background color of the text
XColor)backgroundColorReturns the background color of the text
SelGray:(float)valueSets the gray value of the selected text
oat)selGrayReturns the gray value of the selected text
oat)runGray:(NXRun *)runReturns the gray value for the specified text run
SelColor:(NXColor)colorSets the color of the selected text
XColor)selColorReturns the color of the selected text
XColor)runColor:(NXRun *)runReturns the color of the specified text run
TextGray:(float)valueSets the gray value of the entire text
oat)textGrayReturns the gray value of the entire text
TextColor:(NXColor)colorSets the text color of the entire text
XColor)textColorReturns the text color of the draw entire text

ewFont:newFontIdResets Text object to draw different text
    text:(const char *)newText
    frame:(const NXRect *)newFrame
    tag:(int)newTag
ewFont:(const char *)newFontNameResets Text object to draw different text
    size:(float)newFontSize
    style:(int)newFontStyle
    text:(const char *)newText
    frame:(const NXRect *)newFrame
    tag:(int)newTag
ewRuns:(NXRunArray *)newRunsResets Text object to draw different text
    text:(const char *)newText
    frame:(const NXRect *)newFrame
    tag:(int)newTag
ndowChanged:newWindowHides caret whenever the Text's window changes

wSelf:(const NXRect *)rects :(int)rectCountDraws the Text object
RetainedWhileDrawing:(BOOL)flagAllows use of retained window when drawing
OOL)isRetainedWhileDrawingReturns whether retained window is used for drawing

Tag:(int)anIntMakes anInt the Text object's tag
t)tagReturns the Text object's tag

OOL)acceptsFirstResponderReturns whether receiver can be the first responder
comeFirstResponderInforms Text object that it's becoming first responder
ignFirstResponderStops being the first responder, if delegate agrees
comeKeyWindowActivates caret if selection has width of 0
ignKeyWindowDeactivates the caret

laceSelWithView:viewUnimplemented
Location:(NXPoint *)originSets origin of cell
ofCell:cell
Location:(NXPoint *)originPlaces coordinates of graphic object into origin
ofCell:cell
Location:(NXPoint *)originUnimplemented
ofView:view
GraphicsImportEnabled:(BOOL)flagSets whether a Text object imports TIFF and EPS images
OOL)isGraphicsImportEnabledReturns YES if the object imports TIFF and EPS images

idRequestorForSendType:(NXAtom)sendType
andReturnType:(NXAtom)returnTypeDetermines which Service menu items are enabled
dSelectionFromPasteboard:pboardReplaces selection with data from pboard
OOL)writeSelectionToPasteboard:pboardCopies selection to pboard
types:(NXAtom *)types

CharFilter:(NXCharFilterFunc)aFuncMakes aFunc the character filter function
XCharFilterFunc)charFilterReturns the current character filter function
TextFilter:(NXTextFilterFunc)aFuncMakes aFunc the text filter function
XTextFilterFunc)textFilterReturns the current text filter function
BreakTable:(const NXFSM *)aTableSets table defining word boundaries
nst NXFSM *)breakTableGets table defining word boundaries
PreSelSmartTable:(const unsigned char *)aTable
Sets cut and paste table for left word boundary
nst unsigned char *)preSelSmartTableGets cut and paste table for left word boundary
PostSelSmartTable:(const unsigned char *)aTable
Sets cut and paste table for right word boundary
nst unsigned char *)postSelSmartTableGets cut and paste table for right word boundary
CharCategoryTable:(const unsigned char *)aTable
Sets table defining character categories
nst unsigned char *)charCategoryTableReturns table defining character categories
ClickTable:(const NXFSM *)aTableSets table defining double-click selection
nst NXFSM *)clickTableGets table defining double-click selection
ScanFunc:(NXTextFunc)aFuncMakes aFunc the scan function
XTextFunc)scanFuncReturns the current scan function
DrawFunc:(NXTextFunc)aFuncMakes aFunc the function that draws the text
XTextFunc)drawFuncReturns the current draw function

ustPageHeightNew:(float *)newBottomAssists automatic pagination of text
top:(float)oldTop
bottom:(float)oldBottom
limit:(float)bottomLimit

egateReturns the Text object's delegate

tWillResize:senderInforms delegate of impending size change
tDidResize:senderReports size change to delegate
    oldBounds:(const NXRect *)oldBounds
    invalid:(NXRect *)invalidRect
OOL)textWillChange:senderInforms delegate of impending text change
tDidChange:senderAlerts delegate to change in text
OOL)textWillEnd:senderWarns of impending loss of first responder status
tDidEnd:senderReports to delegate loss of first responder status
    endChar:(unsigned short)whyEnd
tDidGetKeys:sender isEmpty:(BOOL)flagInforms delegate of each text change