OOL)isBackgroundTransparentReturns whether the background between Cells is transparent
CellBackgroundTransparent:(BOOL)flagSets whether the background within Cells is transparent
OOL)isCellBackgroundTransparentReturns whether the background within Cells is transparent
Font:fontObjectSets the Font used to display text in the Cells
tReturns the Font used to display text in the Cells

ectText:senderSelects the text in the first or last editable Cell
ectTextAt:(int)row :(int)colSelects the text of the Cell at row, col in the Matrix

NextText:anObjectSets the object selected when the user presses Tab while editing the last text Cell
PreviousText:anObjectSets the object selected when user presses Shift-Tab while editing the first text Cell

TextDelegate:anObjectSets the delegate for messages from the field editor
tDelegateReturns the delegate for messages from the field editor

OOL)textWillChange:textObjectResponds to a message from the field editor (see Text)
tDidChange:textObjectResponds to a message from the field editor (see Text)
tDidGetKeys:textObjectResponds to a message from the field editor (see Text)
isEmpty:(BOOL)flag
OOL)textWillEnd:textObjectResponds to a message from the field editor (see Text)
tDidEnd:textObjectResponds to a message from the field editor (see Text)
endChar:(unsigned short)whyEnd

AutosizeCells:(BOOL)flagSets whether the Matrix resizes its Cells automatically
OOL)doesAutosizeCellsReturns whether the Matrix resizes its Cells automatically
cSizeCalculates Cell sizes
eTo:(float)width :(float)heightResizes the Matrix to width and height
eToCellsResizes the Matrix to fit its Cells exactly
eToFitResizes the Cells and Matrix to fit the Cell contents
idateSize:(BOOL)flagSets whether the Cell size needs to be recalculated

AutoScroll:(BOOL)flagSets whether the Matrix automatically scrolls when dragged in
Scrollable:(BOOL)flagMakes all the Cells scrollable
ollCellToVisible:(int)row :(int)colScrolls Matrix so the Cell at row and col is visible

playDraws the Matrix and its Cells
wSelf:(const NXRect *)rects :(int)rectCountDraws the Matrix and its Cells
wCell:aCellDraws aCell if it's in the Matrix
wCellAt:(int)row :(int)colDisplays the Cell at row and col
wCellInside:aCellDraws the inside of aCell if it's in the Matrix
hlightCellAt:(int)row :(int)col lit:(BOOL)flagHighlights (or unhighlights) the Cell at row, col

EL)errorActionReturns the action method for editing errors
Target:anObject at:(int)row :(int)colAssigns anObject as the target of the Cell at row, col
Action:(SEL)aSelector at:(int)row :(int)colAssigns aSelector as the action method of the Cell at row, col
dActionSends the selected Cell's action, or the Matrix's action if the Cell doesn't have one
dAction:(SEL)theAction to:theTargetHas the Application object send theAction to anObject
dAction:(SEL)aSelectorSends aSelector to anObject, for all Cells if flag is YES
    to:anObject
    forAllCells:(BOOL)flag
dDoubleActionSends the action corresponding to a double-click
Reaction:(BOOL)flagSets whether sending an action clears the selection

OOL)acceptsFirstMouseReturns NO only if mode is NX_LISTMODE
useDown:(NXEvent *)theEventResponds to a mouse-down event
t)mouseDownFlagsReturns the event flags in effect at start of tracking
OOL)performKeyEquivalent:(NXEvent *)theEvent
    Simulates mouse click in the appropriate Cell

etCursorRectsResets cursor rectangles so that the cursor becomes an I-beam over text Cells

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

tInitializes a new Menu with the title ªMenuº
tTitle:(const char *)aTitleInitializes a new Menu with aTitle as its title

dItem:(const char *)aStringAdds a new item to the end of the Menu
    action:(SEL)aSelector
    keyEquivalent:(unsigned short)charCode
ItemList:aMatrixReplaces the current Matrix of items with aMatrix
mListReturns the Menu's Matrix of MenuCell items

veTopLeftTo:(NXCoord)x :(NXCoord)yMoves the Menu's top left corner to x, y
ndowMoved:(NXEvent *)theEventHandles a submenu being torn off its supermenu
Location:(NXPoint *)theLocationDetermines where to display an attached submenu
forSubmenu:aSubmenuwhen it's brought up
eToFitResizes the Menu to exactly fit the command items
seRemoves the Menu (and any submenus) from the screen

playDisplays the Menu, resizing if needed
Autoupdate:(BOOL)flagSets whether Menu reacts to update messages
dateUpdates each MenuCell item

useDown:(NXEvent *)theEventTracks the cursor in the Menu and submenus
ntMouseDown:(NXEvent *)theEventPops the main menu up under the cursor

d:(NXTypedStream *)streamReads the Menu from stream
te:(NXTypedStream *)streamWrites the Menu to stream
akeReinitializes a Menu as it's unarchived

tInitializes a new MenuCell with ªMenu Itemº as its title
tTextCell:(const char *)aStringInitializes a new MenuCell with aString as its title

UpdateAction:(SEL)aSelectorSets the update action for the MenuCell to aSelector,
forMenu:aMenuand sets aMenu to auto-update
EL)updateActionReturns the update action for the MenuCell

OOL)hasSubmenuReturns whether the MenuCell has a submenu

OOL)trackMouse:(NXEvent *)theEventRefers mouse tracking to the MenuCell's Menu
inRect:(const NXRect *)cellFrame
ofView:controlView

tFromSection:(const char *)nameInitializes the new object from TIFF data in the section
tFromFile:(const char *)filenameInitializes the new object from TIFF data in filename
tFromStream:(NXStream *)streamInitializes the new object from TIFF data in stream
tData:(unsigned char *)dataInitializes the new object using data read from an image
fromRect:(const NXRect *)rect
tData:(unsigned char *)dataInitializes the new object from raw bitmap data
pixelsWide:(int)width
pixelsHigh:(int)height
bitsPerSample:(int)bps
samplesPerPixel:(int)spp
hasAlpha:(BOOL)alpha
isPlanar:(BOOL)config
colorSpace:(NXColorSpace)space
bytesPerRow:(int)rowBytes
bitsPerPixel:(int)pixelBits
tDataPlanes:(unsigned char **)planesInitializes the new object from raw bitmap data in the
pixelsWide:(int)widthplanes data buffers
pixelsHigh:(int)height
bitsPerSample:(int)bps
samplesPerPixel:(int)spp
hasAlpha:(BOOL)alpha
isPlanar:(BOOL)config
colorSpace:(NXColorSpace)space
bytesPerRow:(int)rowBytes
bitsPerPixel:(int)pixelBits

t)bitsPerPixelReturns how many bits are needed to specify one pixel
t)samplesPerPixelReturns the number of samples (components) in the data
OOL)isPlanarReturns YES if in planar configuration, NO if meshed
t)numPlanesReturns the number of data planes
t)bytesPerPlaneReturns the number of bytes in each data plane
t)bytesPerRowReturns the number of bytes in a scan line
XColorSpace)colorSpaceReturns how bitmap data is to be interpreted

signed char *)dataReturns a pointer to the bitmap data
DataPlanes:(unsigned char *)planesProvides pointers to each plane of bitmap data

OOL)drawDraws the image at (0.0, 0.0) in current coordinates
OOL)drawIn:(const NXRect *)rectModifies coordinates so image is drawn in rect rectangle

teTIFF:(NXStream *)streamWrites a TIFF representation of the image to stream
teTIFF:(NXStream *)streamWrites a TIFF representation of the image to stream
usingCompression:(int)compression
teTIFF:(NXStream *)streamWrites a TIFF representation of the image to stream
usingCompression:(int)compression
andFactor:(float)factor

OOL)canBeCompressedUsing:(int)compression
YES if the image can be compressed using compression
id)getCompression:(int *)compression Returns the compression type and compression factor
andFactor:(float *)factor
id)setCompression:(int)compression Sets the compression type and compression factor
andFactor:(float)factor

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

Action:(SEL)aSelectorSets the NXBrowser's action method to aSelector
EL)actionReturns the NXBrowser's action method
Target:anObjectSets the NXBrowser's target object to anObject
getReturns the NXBrowser's target object
DoubleAction:(SEL)aSelectorSets the NXBrowser's double-click action to aSelector
EL)doubleActionReturns the NXBrowser's double-click action method

MatrixClass:classIdSets the class of Matrix used in the NXBrowser's columns
CellClass:classIdSets the class of Cell used in the columns of NXBrowser
CellPrototype:aCellSets the Cell instance copied to display items in the columns of NXBrowser
lPrototypeReturns the NXBrowser's prototype Cell

MultipleSelectionEnabled:(BOOL)flagSets whether the user can select multiple items
OOL)isMultipleSelectionEnabledReturns whether the user can select multiple items
BranchSelectionEnabled:(BOOL)flagSets whether the user can select branch items when multiple selection is
OOL)isBranchSelectionEnabledReturns whether the user can select branch items when multiple selection is e
EmptySelectionEnabled:(BOOL)flagSets whether there can be nothing selected
OOL)isEmptySelectionEnabledReturns whether there can be nothing selected
seColumns:(BOOL)flagPrevents Matrices from being freed when their columns are unloaded, so they can be
Enabled:(BOOL)flagSets whether the NXBrowser reacts to events
OOL)acceptsFirstResponder
eptArrowKeys:(BOOL)acceptFlagEnables arrow keys for scrolling and sending action
andSendActionMessages:(BOOL)sendFlagmessages
TitleFromPreviousColumn:(BOOL)flagSets whether the title of a column is set to the title of the selected Cel
column

ScrollBars:(BOOL)flagSets whether Scrollers are used to scroll columns
ScrollButtons:(BOOL)flagSets whether buttons are used to scroll columns
HorizontalScollButtonsEnabled:(BOOL)flag
Sets whether buttons are used to scroll horizontally
OOL)areHorizontalScollButtonsEnabledReturns whether buttons are used to scroll horizontally
HorizontalScollerEnabled:(BOOL)flagSets whether Scrollers are used to scroll horizontally
OOL)areHorizontalScollerEnabledReturns whether Scrollers are used to scroll horizontally

MinColumnWidth:(int)columnWidthSets the minimum column width
t)minColumnWidthReturns the minimum column width
MaxVisibleColumns:(int)columnCountSets the maximum number of columns displayed
t)maxVisibleColumnsReturns the maximum number of visible columns

dColumnZeroLoads column zero unloads previously loaded columns
OOL)isLoadedReturns whether column zero is loaded
lColumnAdds a column to the right of the last column
oadColumn:(int)columnReloads column if it is loaded sets it as the last column
playColumn:(int)columnUpdates to display columns through index column
playAllColumnsUpdates the NXBrowser to display all loaded columns
LastColumn:(int)columnSets the last column to column
ectAll:senderSelects all Cells in the last column of the NXBrowser
t)selectedColumnReturns the index of the last column with a selected item
t)columnOf:matrixReturns the column number in which matrix is located
idateVisibleColumnsInvokes delegate method browser:columnIsValid: for visible columns

Titled:(BOOL)flagSets whether columns display titles
OOL)isTitledReturns whether columns display titles
Title:(const char *)aStringSets the title of the column at index column to aString
ofColumn:(int)column
nst char *)titleOfColumn:(int)columnReturns the title displayed for the column at index column
XRect *)getTitleFrame:(NXRect *)theRectReturns the bounds of the title frame for the column at
ofColumn:(int)columnindex column
XCoord)titleHeightReturns the height of column titles
wTitle:(const char *)titleDraws the title for the column at index column
inRect:(const NXRect *)aRect
ofColumn:(int)column
arTitleInRect:(const NXRect *)aRectClears the title for the column at index column
ofColumn:(int)column

ollColumnsLeftBy:(int)shiftAmountScrolls columns left by shiftAmount columns
ollColumnsRightBy:(int)shiftAmountScrolls columns right by shiftAmount columns
ollColumnToVisible:(int)columnScrolls to make the column at index column visible
ollUpOrDown:senderScrolls a column up or down
ollViaScroller:senderScrolls columns left or right based on a Scroller
lectScroll:clipViewUpdates scroll buttons to reflect column contents
dateScrollerUpdates the horizontal Scroller to reflect column positions

useDown:(NXEvent *)theEventHandles mouse-down events in the NXBrowser
vDown:(NXEvent *)theEventHandles key-down events
Click:senderResponds to mouse clicks in a column of NXBrowser
DoubleClick:senderResponds to double-clicks in a column of NXBrowser

LoadedCellAtRow:(int)rowLoads if necessary and returns the Cell at row in column

XRect *)getFrame:(NXRect *)theRectReturns the rectangle containing the column at index
ofInsideOfColumn:(int)columncolumn, not includingborders

PathSeparator:(unsigned short)charCodeSets the path separator to charCode
Path:(const char *)pathParses path and selects corresponding items in columns
ar *)getPath:(char *)thePathReturns string representing path from the first
toColumn:(int)columncolumn to the column at index column

wSelf:(const NXRect *)rects :(int)rectCountDraws the NXBrowser

eTo:(NXCoord)width :(NXCoord)heightResizes the NXBrowser to width and height
eToFitResizes the NXBrowser to fit all its contents

Adjusts the NXBrowser's components

OOL)browser:senderReturns whether the contents of the column are valid
columnIsValid:(int)column
owserDidScroll:senderNotifies the delegate when the NXBrowser has scrolled
t)browser:senderReturns the number of rows in a column and loads
fillMatrix:matrixNXBrowserCells in matrix
inColumn:(int)column
t)browser:senderReturns the number of rows of data in the column at index
getNumRowsInColumn:(int)columncolumn
owser:senderRequests the delegate to load Cell at row in the column at
loadCell:cellindex column
atRow:(int)row
inColumn:(int)column
OOL)browser:senderRequests the delegate to select the Cell with title title in the
selectCell:(const char *)titlecolumn at index column
inColumn:(int)column
nst char *)browser:senderQueries the delegate for the title to display above the
titleOfColumn:(int)columncolumn at index column
owserWillScroll:senderNotifies the delegate when the NXBrowser will scroll

tInitializes a new NXBrowserCell with ªBrowserItemº as its title
tTextCell:(const char *)aStringInitializes a newNXBrowserCell with aString as its title

anchIconReturns the NXImage for branch NXBrowserCells
ranchIconHReturns the NXImage for highlighted branches

wInside:(const NXRect *)cellFrameDraws the inside of the NXBrowserCell in aView
inView:aView
wSelf:(const NXRect *)cellFrameDraws the entire NXBrowserCell in aView
inView:aView
hlight:(const NXRect *)cellFrameIf lit is YES, highlights the NXBrowserCell in aView
inView:aView lit:(BOOL)lit

Leaf:(BOOL)flagSets whether the NXBrowserCell is a leaf or a branch
OOL)isLeafReturns whether the NXBrowserCell is a leaf or a branch

Loaded:(BOOL)flagSets whether the NXBrowserCell is loaded and displayble
OOL)isLoadedReturns whether the NXBrowserCell is loaded

Highlights the NXBrowserCell and sets its state to 1
etUnhighlights the NXBrowserCell and sets its state to 0

tFromWindow:(Window *)aWindowInitializes the new NXCachedImageRep for an image to
rect:(const NXRect *)aRectbe drawn in aWindow
pyFromZone:(NXZone *)theZoneCreates and returns a copy of the receiver

eDeallocates the NXCachedImageRep

Window:(Window **)theWindowProvides the Window and rectangle where the image is
andRect:(NXRect *)theRectcached

OOL)drawReads the cached image and renders it

d:(NXTypedStream *)streamReads the NXCachedImageRep from stream

t)colorMaskReturns the color mask of the NXColorPanel
ColorMask:(int)colormaskSets the color mask of the NXColorPanel
Continuous:(BOOL)flagSets the NXColorPanel to continuously send the action message to the target
Mode:(int)modeSets the mode and returns the NXColorPanel
AccessoryView:aViewSets the accessory view to aView
Action:(SEL)aSelectorSets the action message sent to the target
ShowAlpha:(BOOL)flagSets the NXColorPanel to show alpha values
Target:anObjectSets the target of the NXColorPanel

or:(NXColor *)colorReturns the currently selected color
Color:(NXColor)colorSets the color of the NXColorPanel
ragColor:(NXColor *)colorDrags color into a destination view from sourceView.
withEvent:(NXEvent *)eventevent is usually an NX_MOUSEUP
fromView:sourceView

tFromPickerMask:(int)theMaskInitializes the receiver for the specified mask and
withColorPanel:thePanelcolor panel

videNewButtonImageReturns a new button image for the color picker
ertNewButtonImage:newImage Override to customize newImage before insertion
in:newButtonCellin newButtonCell

wSizeChanged:senderDoes nothing. Override to respond to size change.

achColorList:colorListOverride to attach a color list to a color picker
achColorList:colorListOverride to detach a color list from a color picker

Mode:(int)modeOverride to set the color picker's mode

tFrame:(const NXRect *)theFrameInitializes and returns a new instance of NXColorWell

OOL)acceptsFirstMouseReturns YES
useDown:(NXEvent *)theEventResponds to mouse down in the NXColorWell

wSelf:(const NXRect *)rectsDraws the entire NXColorWell, including borders
:(int)rectCount
wWellInside:(const NXRect *)insideRectDraws the colored area inside the NXColorWell,without drawing bo

ctivateDeactivates and returns the NXColorWell
eactivateAllWellsDeactivates all currently active NXColorWells
ivate:(int)exclusiveActivates and returns the NXColorWell
OOL)isActiveReturns YES if the NXColorWell is active
Enabled:(BOOL)enabledEnables the NXColorWell

iveWellsTakeColorFrom:senderChanges color of all active wells to that of sender
iveWellsTakeColorFrom:senderContinuously changes color of all active, continuous
continuous:(BOOL)continuouslywells to that of sender
XColor)colorReturns the color of the NXColorPanel
eColorFrom:senderChanges color of the well to that of sender
eptColor:(NXColor)colorChanges color of the well to color when aPoint is a
atPoint:(NXPoint *)aPointpoint in the bounds of the NXColorWell
Color:(NXColor)colorSets the color of the well to color
dateCustomColorListSaves the current color list in NX_COLORLISTMODE

EL)actionReturns the NXColorWell's action message
Action:(SEL) aSelectorSets the NXColorWell's action message
Target:anObjectSets the NXColorWell's target

tInitializes a new NXCursor, but doesn't set the image
tFromImage:imageInitializes a new NXCursor object with image

Image:newImageSets the NXImage object that supplies the cursor image
ageReturns the NXImage object that has the cursor image
HotSpot:(const NXPoint *)spotSets the point on the cursor that's aligned with the mouse

shMakes the NXCursor the current cursor
pRestores the previous cursor
opRestores the previous cursor
Sets the NXCursor to be the current cursor
OnMouseEntered:(BOOL)flagDetermines whether mouseEntered: sets cursor
OnMouseExited:(BOOL)flagDetermines whether mouseExited: sets cursor
useEntered:(NXEvent *)theEventResponds to a mouse-entered event
useExited:(NXEvent *)theEventResponds to a mouse-exited event
urrentCursorReturns the current cursor

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

tDrawMethod:(SEL)aSelectorInitializes the new object so that anObject's aSelector
inObject:anObjectmethod will draw the image

OOL)drawSends a message to draw the image

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

tFromFile:(const char *)filenameInitializes a new instance from filename

teToPasteboard:(Pasteboard *)pasteboardWrites the link onto the pasteboard pasteboard
eLinkIn:(const char *)directoryNameSaves the link with a file name provided by the user
teToFile:(const char *)filenameWrites the link into the file filename

XDataLinkManager *)managerReturns the link's manager
XDataLinkDisposition)dispositionIdentifies the link's type
XDataLinkNumber)linkNumberReturns the link's number

nst char *)sourceAppNameReturns the name of the application containing the source
nst char *)sourceFilenameReturns the file name of the source document
XSelection *)sourceSelectionReturns the source selection
enSourceOpens the document corresponding to source selection
me_t)lastUpdateTimeReturns the last time the link was updated
nst NXAtom *)typesReturns the types that the source document can provide

nst char *)destinationAppNameReturns the name of the application containing the destination link
nst char *)destinationFilenameReturns the file name of the destination document
XSelection *)destinationSelectionReturns the destination selection

rceEditedSent to a source link to inform it that the data referred to by its source selection has changed
dateDestinationUpdates the data referred to by the link's destination selection
UpdateMode:(NXDataLinkUpdateMode)modeSets the link's update mode to mode
XDataLinkUpdateMode)updateModeReturns the link's update mode
akBreaks the link

tWithDelegate:anObjectInitializes and returns a newly allocated instance
tWithDelegate:anObjectInitializes and returns a newly allocated instance
fromFile:(const char *)path
eFrees the objects and storage held by the link manager

dLink:(NXDataLink *)linkAdds the link link to the document
at:(NXSelection *)selection
dLinkAsMarker:(NXDataLink *)linkIncorporates link into the document as a marker
at:(NXSelection *)selection

cumentEditedInforms link manager that document has been edited
cumentRevertedInforms link manager that changes have been reverted
cumentSavedInforms link manager that document has been saved
cumentSavedAs:(const char *)pathInforms link manager that document has been saved
cumentSavedTo:(const char *)pathInforms link manager that document has been saved

nst char *)filenameReturns the filename for the link manager's document
OOL)isEditedReturns YES if the document was edited since the last save
LinksVerifiedByDelegate:(BOOL)flagSets whether the delegate is asked to verify updates
OOL)areLinksVerifiedByDelegateReturns YES is delegate is asked to verify updates
egateReturns the data link manager's delegate
InteractsWithUser:(BOOL)flagSets whether manager displays panels if link errors occur
OOL)interactsWithUserTells whether manager displays panels if link errors occur

LinkOutlinesVisible:Sets whether outlines are visible
OOL)areOutlinesVisibleReturns YES if outlines are visible
XDataLink *)findDestinationLinkWithSelection:(NXSelection *)destSel
Returns the destination link for the selection destSel
pareEnumerationState:(NXLinkEnumerationState *)state
forLinksOfType:(NXDataLinkDisposition *)srcOrDest
Prepares manager to enumerate links
XDataLink *)nextLinkUsing:(NXLinkEnumerationState *)state
Returns the link manager's next link based on state

Link:(NXDataLink *)linkInforms the receiver of the current document and selection
andManager:(NXDataLinkManager *)linkManager
isMultiple:(BOOL)flag
etLink:(NXDataLink **)linkGets information about the currently selected link
andManager:(NXDataLinkManager **)linkManager

kedBreakAllLinks:senderInvoked when the user clicks the Break All Links button
kedBreakLink:senderInvoked when the user clicks the Break Link button
kedOpenSource:senderInvoked when the user clicks the Open Source button
kedUpdateDestination:senderInvoked when the user clicks Update from Source button
kedUpdateMode:senderInvoked when the user selects the update mode

tFromSection:(const char *)nameInitializes the new object from EPS code in the section
tFromFile:(const char *)filenameInitializes the new object from EPS code in filename
tFromStream:(NXStream *)streamInitializes the new object from EPS code in stream

pyFromZone:(NXZone *)zoneReturns a copy of the NXEPSImageRep
eDeallocates the NXEPSImageRep

BoundingBox:(NXRect *)rectCopies the EPS bounding box into the rect rectangle

EPS:(char **)theEPS length:(int *)numBytesProvides a pointer to the EPS code

pareGStateImplemented by subclasses to prepare the graphics state
OOL)drawDraws the image at (0.0, 0.0) in current coordinates
OOL)drawIn:(const NXRect *)rectDraws the image so it fits within the rect rectangle

dSupplement:(const char *)helpDirectoryAdds supplemental helpto the text displayed in the panel
inPath:(const char *)supplementPath
eFrees the NXHelpPanel and its storage

nt:senderPrints the currently displayed help text
ntPanel:senderPrints the currently displayed help text

XAtom)helpDirectoryReturns the absolute path of the help directory
XAtom) helpFileReturns the path of the currently loaded help file

owFile:(const char *)filenameCauses the Help panel to display the help contained in
atMarker:(const char *)markerNamefilename at markerName
OOL)showHelpAttachedTo:anObjectCauses the Help panel to display help attached to anObject

tInitializes the new NXImage without setting its size
tSize:(const NXSize *)aSizeInitializes the new NXImage to the specified size
tFromSection:(const char *)nameInitializes the new object from the data in name section
tFromFile:(const char *)filenameInitializes the new NXImage from the data in filename
tFromPasteboard:(Pasteboard *)pasteboardInitializes the new NXImage from the data in pasteboard
tFromStream:(NXStream *)streamInitializes the new NXImage from the data in stream
tFromImage:(NXImage *)imageInitializes the new NXImage to be a subimage of image
rect:(const NXRect *)rect
pyFromZone:(NXZone *)zoneCreates and returns a copy of the NXImage in zone

OOL)setName:(const char *)stringAssigns string as the name of the NXImage object
nst char *)nameReturns the name of the NXImage object
ndImageNamed:(const char *)nameReturns the NXImage object with name

OOL)useDrawMethod:(SEL)aSelectorCreates a representation that will use a delegated method
inObject:anObjectto draw the image
OOL)useFromSection:(const char *)nameCreates representations for the data in the name section
OOL)useFromFile:(const char *)filenameCreates representations for the data in filename file
OOL)useRepresentation:(NXImageRep *)imageRep
Adds imageRep to the List of representations
OOL)useCacheWithDepth:(NXWindowDepth)depth
Creates an empty representation to draw in
OOL)loadFromStream:(NXStream *)streamCreates representation for the data read from stream
OOL)loadFromFile:(const char *)fileNameCreates representation for the data read from filename
OOL)lockFocusPrepares for drawing in the best representation
OOL)lockFocusOn:(NXImageRep *)imageRepPrepares for drawing in imageRep
ockFocusBalances a previous lockFocus or lockFocusOn:

mposite:(int)opComposites the image to aPoint
toPoint:(const NXPoint *)aPoint
mposite:(int)opComposites the aRect portion of the image to aPoint
fromRect:(const NXRect *)aRect
toPoint:(const NXPoint *)aPoint
solve:(float)deltaComposites the image using the dissolve operator
toPoint:(const NXPoint *)aPoint
solve:(float)deltaComposites the image using the dissolve operator
fromRect:(const NXRect *)aRect
toPoint:(const NXPoint *)aPoint

ColorMatchPreferred:(BOOL)flagDetermines whether color matches are preferred
OOL)isColorMatchPreferredReturns whether color matches are preferred
EPSUsedOnResolutionMismatch:(BOOL)flag
Sets whether to use EPS representations on mismatch
OOL)isEPSUsedOnResolutionMismatchReturns whether to use EPS representations on mismatch
MatchedOnMultipleResolution:(BOOL)flagSets whether resolution multiples match
OOL)isMatchedOnMultipleResolutionReturns whether resolution multiples match

XImageRep *)lastRepresentationReturns the last representation added to the NXImage
XImageRep *)bestRepresentationReturns the best representation for the deepest screen
st *)representationListReturns the List of all the representations

CacheDepthBounded:(BOOL)flagSets whether the default depth limit applies to caches
OOL)isCacheDepthBoundedReturns whether the default depth limit applies to caches
Image:(NXImage **)imageGets the image that the receiver is a subimage of
rect:(NXRect *)rect

Flipped:(BOOL)flagInverts the polarity of the y-axis for drawing the image
OOL)isFlippedReturns whether the polarity of the y-axis is inverted
Scalable:(BOOL)flagDetermines whether representations are scaled to fit
OOL)isScalableReturns whether representations are scaled to fit
BackgroundColor:(NXColor)aColorSets the background color of the image
XColor)backgroundColorReturns the background color of the image
OOL)drawRepresentation:(NXImageRep *)imageRep
inRect:(const NXRect *)rectHas imageRep draw the representation
acheInvalidates caches of all representations, so they will be redrawn

Delegate:anObjectMakes anObject the delegate of the NXImage
egateReturns the delegate of the NXImage

teTIFF:(NXTypedStream *)streamWrites TIFF for the best representation to stream
teTIFF:(NXTypedStream *)streamWrites TIFF for all the representations to stream
allRepresentations:(BOOL)flag

XImage *)imageDidNotDraw:senderResponds to message that image couldn't be composited
inRect:(NXRect *)aRect

tFromPasteboard:(Pasteboard *)pasteboardInitializes the receiver from pasteboard

Size:(const NXSize *)aSizeSets the size of the image
Size:(NXSize *)theSizeCopies the size of the image into the theSize structure

NumColors:(int)anIntInforms the object that there are anInt color components
t)numColorsReturns the number of color components
Alpha:(BOOL)flagInforms object whether there is a coverage component
OOL)hasAlphaReturns whether there is a coverage component
BitsPerSample:(int)anIntInforms object there are anInt bits/pixel in a component
t)bitsPerSampleReturns the number of bits per pixel in each component
PixelsHigh:(int)anIntInforms object that data is for an image anInt pixels high
t)pixelsHighReturns the height specified in the image data
PixelsWide:(int)anIntInforms object that data is for an image anInt pixels wide
t)pixelsWideReturns the width specified in the image data

OOL)drawImplemented by subclasses to draw the image
OOL)drawAt:(const NXPoint *)pointModifies current coordinates so image is drawn at point
OOL)drawIn:(const NXRect *)rectModifies current coordinates so image is drawn in rect

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

EventStatus:(int)eventStatusControls recording and playback
soundStatus:(int)soundStatus
eventStream:(NXStream *)stream
soundfile:(const char *)soundfile
EventStatus:(int *)eventStatusPtrProvides status information about the NXJournaler
soundStatus:(int *)soundStatusPtr
eventStream:(NXStream **)streamPtr
soundfile:(char **)soundfilePtr
RecordDevice:(int)deviceSets whether CODEC or DSP is used for sound input
t)recordDeviceReturns NX_CODEC or NX_DSP

eakerReturns the NXJournaler's Speaker object
enerReturns the NXJournaler's Listener object
Delegate:anObjectSets the NXJournaler's delegate
egateReturns the NXJournaler's delegate

rnalerDidEnd:journalerInforms the delegate that the session terminated
rnalerDidUserAbort:journalerInforms the delegate that the user aborted the session

nst char *)domainReturns the name of the printer's domain
nst char *)hostReturns the name of the printer's host computer
nst char *)nameReturns the printer's name
nst char *)noteReturns the note associated with the printer
nst char *)typeReturns the name of the printer's type
OOL)isReallyAPrinterReturns whether the object corresponds to an actual printer

OOL)isValidReturns whether the NXPrinter is valid
t)languageLevelReturns the PostScript Language Level recognized by the printer
OOL)isOutputStackInReverseOrderReturns whether the printer outputs pages in reverse page order

OOL)booleanForKey:(const char *)key Returns a boolean value for the given key in the given table
inTable:(const char *)table
id *)dataForKey:(const char *)keyReturns untyped data for the key in the table
inTable:(const char *)table
length:(int *)bytes
oat)floatForKey:(const char *)keyReturns a float value for the key in the table
inTable:(const char *)table
t)intForKey:(const char *)keyReturns an integer value for the key in the table
inTable:(const char *)table
XRect)rectForKey:(const char *)keyReturns an NXRect for the key in the table
inTable:(const char *)table
XSize)sizeForKey:(const char *)keyReturns an NXSize for the key in the table
inTable:(const char *)table
nst char *)stringForKey:(const char *)keyReturns a string for the key in the table
inTable:(const char *)table
nst char **)stringListForKey:(const char *)key
inTable:(const char *)tableReturns an array of strings for the key in the table
t)statusForTable:(const char *)tableReturns the status of the given table
OOL)isKey:(const char *)keyReturns whether key is a key to table
inTable:(const char *)tableg

llingPanelReturns the NXSpellChecker's panel
essoryViewReturns the spell panel's accessory view
AccessoryView:aViewMakes a view an accessory of the spell panel

OOL) checkSpelling:(NXSpellCheckMode)how
of:(id <NXReadOnlyTextStream, Starts the search for a misspelled word
NXSelectRange>)anObject
OOL) checkSpelling:(NXSpellCheckMode)how
of:(id <NXReadOnlyTextStream,Starts the search for a misspelled word and the count of
NXSelectRange>)anObjectwords
wordCount:(int *)theCount

forSpellClient:(int)tagInitializes the list of ignored words for a document

OOL)registerLanguage:(const char *)language
byVendor:(const char *)vendor

egateReturns the NXSpellServer's delegate
Delegate:anObjectMakes the spelling service program the delegate of the NXSpellServer object

Starts the event loop in the NXSpellServer's delegate

OOL)isInUserDictionary:(const char *)wordReturns YES if the word is in any open user dictionary
caseSensitive:(BOOL)flag

dGuess:(const char *)guessCalled by the delegate to append the guesses it has found

OOL)spellServer:(NXSpellServer *)sender Searches for a misspelled word return YES if one is found
findMisspelledWord:(int *)start
length:(int *)length
inLanguage:(const char *)language
inTextStream:(id <NXReadOnlyTextStream>)textStream
startingAt:(int)startPosition
wordCount:(int *)number
countOnly:(BOOL)flag

id)spellServer:(NXSpellServer *)sender Searches for alternatives to the misspelled word returns
suggestGuessesForWord:(const char *)word guesses as a side effect, using addGuess:
inLanguage:(const char *)language

id)spellServer:(NXSpellServer *)senderNotifies the delagte of a word added to the user's hidden
didLearnWord:(const char *)wordwordlist
inLanguage:(const char *)language

id)spellServer:(NXSpellServer *)senderNotifies the delagte of a word removed from the user's
didForgetWord:(const char *)wordhidden wordlist
inLanguage:(const char *)language

ustSubviewsAdjusts the heights of the subviews