initFrame:
free

Setting up the NXTableView setDataSource:
dataSource

Setting and reporting formatting
formatterAt::
dynamicRows
dynamicColumns
isRowHeadingVisible
isColumnHeadingVisible
setIntercell:
getIntercell:
setGridVisible:
isGridVisible
acceptArrowKeys:
doesAcceptArrowKeys
allowVectorReordering:
doesAllowVectorReordering
allowVectorResizing:
doesAllowVectorResizing

Notifying the NXTableView of a change
reloadData:
layoutChanged:
rowsChangedFrom:to:
columnsChangedFrom:to:

Handling rows and columns columnCount
rowCount
columnList
rowList
rowAt:
columnAt:
addColumn:at:
addColumn:withTitle:
addColumn:withFormatter:andTitle:at:
removeColumnAt:
moveColumnFrom:to:
addRow:at:
addRow:withTitle:
addRow:withFormatter:andTitle:at:
removeRowAt:
moveRowFrom:to:
setRowHeight:

Editing support editFieldAt::
setEditable:
isEditable
endEditing

Handling the selection setMode:
mode
allowEmptySel:

setColumnSelectionOn::to:
selectRow:byExtension:
selectColumn:byExtension:
deselectRow:
deselectColumn:
selectedRowAfter:
selectedColumnAfter:
sendAction:to:forSelectedRows:
sendAction:to:forSelectedColumns:

Setting NXTableView components

setRowHeadingVisible:
setColumnHeadingVisible:

Adjusting the view drawSelf::

scrollClip:to:
isHorizScrollerVisible
setHorizScrollerRequired:
isVertScrollerVisible
setVertScrollerRequired:
tile
sizeTo::
scrollRowToVisible:
scrollColumnToVisible:
acceptsFirstResponder

Transmitting action setAction:

action
setDoubleAction:
doubleAction
setTarget:
target

Archiving read:

write:
finishUnarchiving

Appointing a delegate setDelegate:

delegate

acceptArrowKeys:(BOOL)flag

Sets according to flag whether or not the NXTableView allows the user to press arrow keys to cha
column. The default is YES. Returns self.

When at least one row is selected,

doesAcceptArrowKeys

(BOOL)acceptsFirstResponder

Returns YES if the NXTableView accepts arrow keys, NO otherwise.

doesAcceptArrowKeys

addColumn:columnVector at:(unsigned int)columnIndex

Inserts columnVector, an NXTableVector, as a static column into the NXTableView at columnInd
makes all columns static. Returns self.

columnAt:, dynamicColumns

addColumn:columnIdentifier
withFormatter:aFormatter
andTitle:(const char *)title
at:(unsigned int)columnIndex

Creates a new static column vector and inserts it into the NXTableView at columnIndex by invoki
data for the new column is provided by the data source from its property for columnIdentifier. The
title, and its formatter is set to aFormatter (an NXFormatter). If aFormatter is nil, the column vecto
NXTextFormatter or NXEditableFormatter as needed. Returns self.

See the NXTableDataSources informal protocol specification for information on identifiers.

formatterAt::

addColumn:columnIdentifier withTitle:(const char *)title

Appends a new static column vector after the last column in the NXTableView by invoking addCo
andTitle:at: with columnIdentifier and title, and with nil as the formatter. Returns self.

addRow:rowVector at:(unsigned int)rowIndex

Inserts rowVector, an NXTableVector, as a static column into the NXTableView at rowIndex. Add
rows static. Returns self.

rowAt:, dynamicRows

addRow:rowIdentifier
withFormatter:aFormatter
andTitle:(const char *)title
at:(unsigned int)rowIndex

Creates a new static row vector and inserts it into the NXTableView at rowIndex by invoking addR
the new row is provided by the data source from its property for rowIdentifier. The row's title is se
formatter is set to aFormatter (an NXFormatter). If aFormatter is nil, the row vector will use a shar
NXEditableFormatter as needed. Returns self.

See the NXTableDataSources informal protocol specification for information on identifiers.

Sets according to flag whether the NXTableView allows an empty selection (a user deselects a sing
Shift-clicking on it). The default is NO. Returns self.

doesAllowEmptySel

allowVectorReordering:(BOOL)flag

Sets according to flag whether the user can drag title tabs to reorder rows or columns (it isn't possi
column without a title). The default is YES. Returns self.

doesAllowVectorReordering

allowVectorResizing:(BOOL)flag

Sets according to flag whether the user can resize a static vector by dragging the edge of its title tab
resize a row or column without a title). The default is YES. Returns self.

doesAllowVectorResizing

(id <NXTableVectors>)columnAt:(unsigned int)columnIndex

Returns the table vector that controls the formatting of the static column at columnIndex.

(unsigned int)columnCount

For an NXTableView with static columns, returns the number of columns. For a table view whose
sends columnCount the data source and returns the result.

dynamicColumns

columnList

Returns a List of the table view's column vectors, ordered as they're displayed, or nil is columns a

dynamicColumns

columnsChangedFrom:(unsigned int)startIndex to:(unsigned int)endIndex

Informs the NXTableView that its data source has changed values for columns from startIndex to e
redisplay. Returns self.

dataSource

deselectAll:sender

If empty selection is permitted and if the delegate responds YES to tableViewWillChangeSelection and their headings (title tabs). If empty selection isn't permitted, deselects all but the first selected refuses, does nothing. Notifies the delegate and the data source if the selection does change by sen tableViewDidChangeSelection: message, and sends the action message to the NXTableView's targ

doesAllowEmptySel

deselectColumn:(unsigned int)columnIndex

Deselects the column at columnIndex, or if that's only selected column and empty selection isn't a Returns self.

doesAllowEmptySel

deselectRow:(unsigned int)rowIndex

Deselects the row at rowIndex, or if that's only selected row and empty selection isn't allowed, do self.

doesAllowEmptySel

(BOOL)doesAcceptArrowKeys

Returns YES if the NXTableView accepts arrow key presses to change the selection, NO otherwis

acceptArrowKeys:, acceptsFirstResponder

(BOOL)doesAllowEmptySel

Returns YES if the NXTableView allows zero vectors to be selected, NO otherwise. The default is

allowEmptySel:

(BOOL)doesAllowVectorReordering

Returns YES if the NXTableView allows the user to rearrange vectors by dragging their title tabs, possible to drag a row or column without a title). The default is YES. You can rearrange vectors pr regardless of this setting.

allowVectorReordering:, moveColumnFrom:to:, moveRowFrom:to:

(SEL)doubleAction

Returns the message sent to the target when the user double-clicks in the NXTableView. If editing
message is sent just before the NXTableView begins editing the field clicked.

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

Redraws the NXTableView. Your application shouldn't need to call this method directly. Returns

drawSelf:: (View class of the Application Kit)

(BOOL)dynamicColumns

Returns YES if the NXTableView's columns are dynamic that is, if they correspond to records in t
NO if the columns are static, representing attributes for all rows.

(BOOL)dynamicRows

Returns YES if the NXTableView's rows are dynamic that is, if they correspond to records in the d
NO if the rows are static, representing attributes for all columns.

editFieldAt:(unsigned int)rowIndex :(unsigned int)columnIndex

Selects the field at rowIndex and columnIndex and has the formatter for that field begin editing. Th
programmatically the effect the user can produce by double-clicking a field in the NXTableView's
self.

If the NXTableView isn't editable or if the delegate responds NO to tableViewWillChangeSelectio
editing can begin, checks to see that the formatter for the indicated field responds to editFieldAt::in
withAttributes::usePositions::. If it doesn't respond, the NXTableView aborts if it does, the NXTab
message (this doesn't mean the formatter will actually edit the field). See the description for editFi
NXEditableFormatter for more information.

endEditing

Ends any editing and redraws the field that was being edited. Returns self.

finishUnarchiving

Invoked as the last step in reading an NXTableView from an archive, to set up determinable state r
method is invoked automatically as part of the process of reading from an archive. Returns self.

free

Frees the storage used by the NXTableView and returns nil.

getIntercell:(NXSize *)theSize

Copies the vertical and horizontal spacing between fields into theSize. The default is 2.0, 2.0. Retu

initFrame:(const NXRect *)frameRect

Initializes a newly allocated NXTableView with boundaries specified by frameRect. The new NXT
or columns, and both axes are considered dynamic until a column or row is added. The new NXTa
column headings but not row headings and vertical scroll bars but not horizontal ones. Returns self

addColumn:withTitle:, addRow:withTitle:

(BOOL)isColumnHeadingVisible

Returns YES if the column-heading view (containing the title tabs for all columns) is displayed, N

(BOOL)isColumnSelected:(unsigned int)columnIndex

Returns YES if the column at columnIndex is selected, NO otherwise.

(BOOL)isEditable

Returns YES if the NXTableView is editable, NO otherwise.

editFieldAt::

(BOOL)isGridVisible

Returns YES if the NXTableView draws grid lines around fields, NO if it doesn't. The default is N

(BOOL)isHorizScrollerVisible

Returns YES if the horizontal scroller is displayed, NO if it isn't. The default is NO.

setHorizScrollerRequired:

(BOOL)isVertScrollerVisible

Returns YES if the vertical scroller is displayed, NO if it isn't. The default is YES.

setVertScrollerRequired:

layoutChanged:sender

Recalculates the position and size of the NXTableView's components. Returns self.

(int)mode

Returns the selection mode, which may be NX_RADIOMODE, NX_LISTMODE, or NX_NOSEL
description of setMode: for more information.

(BOOL)moveColumnFrom:(unsigned int)oldIndex to:(unsigned int)newIndex

Moves the static column and heading at oldIndex to newIndex (that is, in the new sequence, it will
that used to be at newIndex). Returns YES if the move succeeds, NO otherwise. Dynamic columns

This method invokes the delegate method tableView:movedColumnFrom:to:.

(BOOL)moveRowFrom:(unsigned int)oldIndex to:(unsigned int)newIndex

Moves the static row and heading at oldIndex to newIndex (that is, in the new sequence, it will pre
to be at newIndex). Returns YES if the move succeeds, NO otherwise. Dynamic rows can't be mov

This method invokes the delegate method tableView:movedRowFrom:to:.

read:(NXTypedStream *)stream

Reads the NXTableView object from the type stream stream. Returns self.

finishUnarchiving

reloadData:sender

Deselects all rows and columns, recalculates the NXTableView's layout, and redisplays its data (w
retrieved from the data source as its displayed). Returns self.

(id <NXTableVectors>)rowAt:(unsigned int)rowIndex

Returns the table vector that controls the formatting of the static row at rowIndex.

(unsigned int)rowCount

For an NXTableView with static rows, returns the number of rows. For a table view whose rows ar
rowCount the data source and returns the result.

dynamicRows

(float)rowHeight

Returns the default row height.

setRowHeight:

rowList

Returns a List of the table view's row vectors, ordered as they're displayed, or nil is rows are dyna

dynamicRows

rowsChangedFrom:(unsigned int)startIndex to:(unsigned int)endIndex

Informs the NXTableView that its data source has changed values for rows from startIndex to endI
redisplay. Returns self.

scrollClip:aClipView to:(const NXPoint *)newOrigin

Changes aClipView's bounds rectangle origin to newOrigin, resulting in the document view being
is usually sent automatically during scrolling. It's used to coordinate the scrolling of the content vi
views. Returns self.

scrollColumnToVisible:(unsigned int)columnIndex

Scrolls the document view and column headings horizontally so that column at columnIndex is vis

scrollRowToVisible:(unsigned int)rowIndex

selectColumn:(unsigned int)columnIndex byExtension:(BOOL)flag

Selects the column and heading identified by columnIndex. If flag is YES and the NXTableView's
NX_LISTMODE, includes columnIndex in the set of selected columns. Otherwise, this method de
Returns self.

setMode:

(int)selectedColumn

Returns the index of the selected column, or 1 of no column is selected.

(unsigned int)selectedColumnAfter:(unsigned int)columnIndex

Returns the index of the first selected column further to the right than columnIndex. If columnInde
at least one selected column is selected, returns the first selected column. If no column is selected
column to the right of columnIndex, returns NX_NoIndex.

(unsigned int)selectedColumnCount

Returns the number of selected columns.

(int)selectedRow

Returns the index of the selected row, or 1 of no row is selected.

selectRow:(unsigned int)rowIndex byExtension:(BOOL)flag

Selects the row and heading identified by rowIndex. If flag is YES and the NXTableView's selecti
NX_LISTMODE, includes rowIndex in the set of selected rows. Otherwise, this method deselects
self.

setMode:

(unsigned int)selectedRowAfter:(unsigned int)rowIndex

Returns the index of the first selected row further to the right than rowIndex. If rowIndex is NX_N
selected row is selected, returns the first selected row. If no row is selected or if there's no selected
rowIndex, returns NX_NoIndex.

Sends anAction to anObject once for each column (if flag is NO) or once for each selected column
Returns self.

sendAction:(SEL)anAction
to:anObject
forSelectedRows:(BOOL)flag

Sends anAction to anObject once for each row (if flag is NO) or once for each selected row (if flag

setAction:(SEL)aSelector

Sets to aSelector the action method sent to the NXTableView's target when a mouse-up or key-do
NXTableView.

setTarget:

setColumnHeadingVisible:(BOOL)flag

Sets according to flag whether the NXTableView displays column headings (title tabs). If flag is Y
NXTableView doesn't have a column heading view, creates one. If the setting changes, the NXTab
redisplays itself. Returns self.

setColumnSelectionOn:(unsigned int)startIndex
:(unsigned int)endIndex
to:(BOOL)flag

Selects or deselects according to flag the columns from startIndex to endIndex, inclusive. Does not
NXTableView's selection mode is NX_RADIOMODE and the given indexes specify more than on

selectColumn:byExtension:, setMode:

setDataSource:anObject

Makes anObject the data source from which the NXTableView gets its values, and redisplays the t

setDelegate:anObject

Sets the NXTableView's delegate to anObject. Returns self.

setDoubleAction:(SEL)aSelector

setGridVisible:(BOOL)flag

Sets according to flag whether the NXTableView draws grid lines around fields. The space used fo
addition to the intercell spacing. The default is NO. Returns self.

setIntercell:

setHorizScrollerRequired:(BOOL)flag

Adds or removes a horizontal scroller for the NXTableView according to flag, as described for the
Application Kit. Also recalculates the layout of the NXTableView and redisplays its data. Returns

isHorizScrollerVisible

setIntercell:(const NXSize *)aSize

Sets the width and height between fields to those in aSize and redisplays the NXTableView. When
space they use is in addition to the intercell spacing. Returns self.

setGridVisible:

setMode:(int)aMode

Sets the NXTableView's selection mode to aMode and returns self. Possible values for aMode and

setRowHeadingVisible:(BOOL)flag

Sets according to flag whether the NXTableView displays row headings (title tabs). If flag is YES
doesn't have a row heading view, creates one. If the setting changes, the NXTableView retiles and
Returns self.

setRowHeight:(float)rowHeight

Sets the row height.

rowHeight

setTarget:anObject

Sets the NXTableView's target to anObject and returns self. The NXTableView sends its action m
when a mouse-up or key-down event occurs in the NXTableView.

setAction:

setVertScrollerRequired:(BOOL)flag

Adds or removes a vertical scroller for the NXTableView according to flag, as described for the Sc
Application Kit. Also recalculates the layout of the NXTableView and redisplays its data. Returns

isVertScrollerVisible

sizeTo:(NXCoord)width :(NXCoord)height

Resizes the NXTableView to width and height, recomputes its layout, and redisplays it. Returns se

target

Returns the NXTableView's target. The NXTableView sends its action message to the target when
down event occurs in the NXTableView.

action

tile

Arranges the NXTableView's three component views (content, column heading, and row heading
are visible) within the NXTableView's frame. Doesn't display the NXTableView. Returns self.

write:(NXTypedStream *)stream

Writes the NXTableView object to the typed stream stream. Returns self.

tableView:tableView
movedColumnFrom:(unsigned int)oldIndex
to:(unsigned int)newIndex

Invoked when the user changes the position of a static column. The delegate can use this informati
display, for example, by sorting the rows based on the values in the first column. Returns self.

tableViewDidChangeSelection:tableView

Sent to the delegate and to the data source when the user has changed the selection. The delegate o
this information to update other information or displays. Returns self.

(BOOL)tableViewWillChangeSelection:tableView

Sent to the delegate just before the selection will be changed. If the delegate returns YES the selec
delegate returns NO the change is aborted.