formatterAt::

columnCount

columnList
rowList

addRow:at:

editFieldAt::

endEditing

Handling the selection setMode:

setColumnHeadingVisible:

Adjusting the view drawSelf::

scrollClip:to:
isHorizScrollerVisible
setHorizScrollerRequired:
isVertScrollerVisible
setVertScrollerRequired:
tile
sizeTo::

read:

write:
finishUnarchiving

Appointing a delegate setDelegate:

acceptArrowKeys:(BOOL)flag

Enables or disables the arrow keys for keystrokes the user makes within the DBTableView, as flag
default when a DBTableView is initialized is YES. Returns self.

When at least one row is selected,

doesAcceptArrowKeys

(BOOL)acceptsFirstResponder

Returns YES if the DBTableView accepts the role of first responder for its Window.

Inserts a new static column into the DBTableView. The data for the new column will come from t
attribute identified by identifier. The new column will be inserted so that it precedes the column w
(before the insertion) was aPosition. No title is assigned to the new column its formatting will be
formatter. Return self.

addColumn:identifier
withFormatter:formatter
andTitle:(const char *)title
at:(unsigned int)aPosition

Inserts a new static column into the DBTableView. The data for the new column will come from t
attribute identified by identifier. Text for the new column's title will be taken from title. The colu
handled by formatter. The new column will be inserted so that it precedes the column whose colu
insertion) was aPosition.

addColumn:identifier withTitle:(const char *)title

Appends a new static column following the last existing column in the DBTableView. The data fo
come from the DBRecordList's attribute identified by identifier. Text for the new column's title w
The new column has its own default DBTextFormatter. Returns self.

addRow:identifier at:(unsigned int)aPosition

Inserts a new static row into the DBTableView. The data for the new row will come from the DBF
identified by identifier. The new row will be inserted so that it precedes the row whose row-numb
was aPosition. No title is assigned to the new row its formatting will be handled by a default form

addRow:identifier
withFormatter:formatter
andTitle:(const char *)title
at:(unsigned int)aPosition

Inserts a new static row into the DBTableView. The data for the new row will come from the DBF
identified by identifier. Text for the new row's title will be taken from title. The row's formatting
formatter. The new row will be inserted so that it precedes the row whose row-number (before the
aPosition. Returns self.

addRow:identifier withTitle:(const char *)title

Appends a new static row following the last existing row in the DBTableView. The data for the ne
the DBRecordList's attribute identified by identifier. Text for the new row's title will be taken fro
gets it its own DBTextFormatter. If the DBTableView previously had no rows, adding a row mak
self.

allowVectorReordering:(BOOL)flag

Permits the user to drag a static vector to a new position within the DBTableView (or prohibits it, v
default is YES. To drag a vector, the user must click in the vector's title area (to select it) and then
drag an untitled vector. The new ordering of vectors depends on the ordering of their midpoints. T
to the right of column A, to reverse their positions the user must drag B until its midpoint is to the
Returns self.

doesAllowVectorReordering

allowVectorResizing:(BOOL)flag

Permits the user to drag the edges of a static vector so as to change its height or width (or prohibits
To resize a vector, the user must start to drag from a position over the title's edge. In that position,
double arrow (like this

doesAllowVectorResizing

(id <DBTableVectors>)columnAt:(unsigned int)aPosition

Returns the object that controls the formatting of the (static) column identified by aPosition.

(unsigned int)columnCount

For a DBTableView with static columns, returns the number of columns. For a table view whose c
returns the number of columns in the data source.

columnHeading

Returns the view that contains the DBTable's column headings.

columnList

Returns a list of the identifiers of successive columns in the order that they currently appear in the
columns aren't static, returns nil.)

columnsChangedFrom:(unsigned int)startColumn to:(unsigned int)endColumn

Notification that the data source has changed the values in a block of consecutive columns, so their
redrawn. The first of the changed columns is identified by startColumn, the last by endColumn. R

dataSource

any of the actions that cause a change in the row or column selected.

deselectAll:sender

If empty selection is permitted, deselects all selected vectors and their row or column headings. If permitted, deselects all but the first. Notifies the deleegate by sending it a tableViewDidChangeSe sends an action message to the DBTableViews's target. Returns self.

allowEmptySel

deselectColumn:(unsigned int)column

Deselects the indicated column. However, if this is the only selected column and an empty selecti nothing. Returns self.

deselectRow:(unsigned int)row

Deselects the indicated row. However, if this is the only selected row and an empty selection is n nothing. Returns self.

(BOOL)doesAcceptArrowKeys

Returns YES if arrow keys are enabled while the DBTableView is first responder.

acceptArrowKeys

allowEmptySel

(BOOL)doesAllowVectorReordering

Returns YES if the DBTableView permits the user to drag a static vector (row or column) to a new is YES.

allowVectorReordering

(BOOL)doesAllowVectorResizing

Returns YES if the DBTableView permits the user to resize a static vector (row or column). The d

allowVectorResizing

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

Invoked by various methods during scrolling or dragging to redraw the DBTableView. Your appli
to call this method directly. The argument rects is a list of pointers to the coordinates of rectangles
DBTableView is visible, while count is the number of such rectangles. Returns self.

(BOOL)dynamicColumns

Returns YES if the DBTableView's columns are dynamic: that is, if the number of available colum
number of records available (in contrast to the static number of attributes).

(BOOL)dynamicRows

Returns YES if the DBTableView's rows are dynamic: that is, if the number of available rows is d
number of records available (in contrast to the static number of attributes).

editFieldAt:(unsigned int)row :(unsigned int)column

Selects the entry at the indicated row and column, and invokes an editor. This achieves programm
user would produce by double-clicking a field within the DBTableView's content view.

Editing a field permits the user to change the text displayed there. When the user signals completi
or by clicking outside the field being edited), the editor may invoke methods to validate the revised
acceptable, copy its value to the table view's data source. Returns self.

endEditing

Invoked automatically to redraw the field that has been edited at the conclusion of editing. Return

finishUnarchiving

Invoked as the last step in reading a DBTableView from an archive, to position the table view with
rows and columns and their headings (if appropriate), and initialize the selection of rows and colu
need to invoke this explicitly, since it is done automatically as part of the process of reading from a
self.

formatterAt:(unsigned int)row :(unsigned int)column

Returns the formatter responsible for the field at the intersection of the indicatated row and column
typical display, one axis (usually columns) is static and the other (usually rows) is dynamic. In th
formatter applies throughout a given static position, and the dynamic index is immaterial. If there
explicitly assigned to the specified field, the method returns a default formatter for the type of data

You may want to override this method in order to apply different formatting rules.

getIntercell:(NXSize *)theSize

Reports the number of pixels of spacing between adjacent cells, by setting theSize with the two val
vertical separation. The default is 2, 2. Returns self.

initFrame:(const NXRect *)newFrame

Initializes a DBTableView instance within the frame boundaries specified by newFrame. The new
columns, and both axes are considered dynamic. Initially, there is no title there are column headin
vertical scrollbars but not horizontal ones. Reordering and resizing are enabled (but this has no eff
columns become static). The arrow keys are enabled. Returns self.

(BOOL)isColumnHeadingVisible

Returns YES if the column-heading view (containing the headings for all columns) is visible.

(BOOL)isColumnSelected:(unsigned int)column

Returns YES if the indicated column is selected.

(BOOL)isEditable

Returns YES if the DBTableView is editable.

setEditable

(BOOL)isGridVisible

Returns YES if the DBTableView's grid lines are visible.

setGridVisible

(BOOL)isHorizScrollerVisible

Returns YES if the horizontal scroller is visible. The default is NO.

setHorizScrollerRequired

(BOOL)isRowHeadingVisible

Returns YES if the row-heading view (containing the headings for all rows) is visible.

setVertScrollerRequired

layoutChanged:sender

Invoked when there is any change in the number, position, width, height, titling, or format of the D
to update all of these. Returns self.

(int)mode

Returns the selection mode.

setMode

(BOOL)moveColumnFrom:(unsigned int)oldPos to:(unsigned int)newPos

Changes the position of one of the static columns. The column to move is identified by oldPos, its
move. Its new position will be newPos. That is, in the new sequence, it will precede the column th
newPos. The method also makes the corresponding change in the column headings. Returns YES
permitted, NO otherwise. It is never permissible to move a dynamic column.

allowVectorReordering:, doesAllowVectorReordering

(BOOL)moveRowFrom:(unsigned int)oldPos to:(unsigned int)newPos

Changes the position of one of the static rows. The row to move is identified by oldPos, its positio
new position will be newPos. That is, in the new sequence, it will precede the row that used to be
method also makes the corresponding change in the row headings. Returns YES if the move is per
It is never permissible to move a dynamic row.

allowVectorReordering:, doesAllowVectorReordering

read:(NXTypedStream *)stream

Unarchives a DBTableView object from the archive identified by stream.

reloadData:sender

Rechecks the layout and redraws the display. Returns self.

removeColumnAt:(unsigned int)columnPosition

Deletes a static column (and its heading) from the display. Returns self.

(id <DBTableVectors>)rowAt:(unsigned int)aPosition

Returns the object that controls the formatting of the static row whose row number is aPosition.

(unsigned int)rowCount

For a DBTableView with static rows, returns the number of rows. For a table view whose rows ar number of rows in the data source.

rowHeading

Returns the view that contains the DBTableView's row headings.

rowList

Returns a list of the identifiers of successive static rows in the order that they currently appear in th rows aren't static, returns nil.)

rowsChangedFrom:(unsigned int)startRow to:(unsigned int)endRow

Notification that the data source has change the values in a block of rows, so their display should b the changed rows is identified by startRow, and the last by endRow. Returns self.

scrollClip:aClip to:(const NXPoint *)newOrigin

Changes the portion of the content of the clip view aClip that is visible. The change makes the pos content view's coordinates) appear at the clip view's origin (that is, its lower left corner). This me automatically, in response to scrolling in the view aClip. It is used to coordinate the scrolling of th two heading views with a table view, or when the arrow keys make the selected portion of the view Returns self.

scrollColumnToVisible:(unsigned int)column

Scrolls the content view and column headings horizontally so that the requested column is visible.

scrollRowToVisible:(unsigned int)row

Scrolls the content view and row headings vertically so that the requested row is visible. Returns s

Selects the column (and its heading) identified by column. When flag is YES and the DBTableVie
multiple selection, includes column in the set of selected columns. Otherwise, this method deselec
Returns self.

(int)selectedColumn

Returns the column number of the selected column. Column numbers are successive integers start
columns actually displayed, in their current left-to-right order in the display. Returns 1 of no colu

(unsigned int)selectedColumnAfter:(unsigned int)aColumn

Returns the column number of the first selected column that is further to the right than aColumn. I
DB_NoIndex and there is at least one selected column, returns the first selected column. If no colu
is no selected column to the right of aColumn, returns DB_NoIndex.

(unsigned int)selectedColumnCount

Returns the number of selected columns.

(int)selectedRow

Returns the row number of the selected row. Row numbers are successive integers starting at 0, fo
displayed, in their current top-to-bottom order in the display. Returns 1 if no row is selected.

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

Selects the row (and its heading) identified by row. When flag is YES and the DBTableView's mo
selection, includes row in the set of selected rows. Otherwise, this method deselects other rows. R

(unsigned int)selectedRowAfter:(unsigned int)aRow

Returns the row number of the first selected row that is further down than aRow. If aRow is DB_N
least one selected row, returns the first selected row. If no row is selected, or there is no selected r
aColumn, returns DB_NoIndex.

(unsigned int)selectedRowCount

Returns the number of selected rows.

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

Sends the message anAction to the object anObject once for each row (when flag is NO) or once fo
(when flag is YES). Returns self.

setAction:(SEL)aSelector

Sets the action method that will be sent to the DBTableView's target when a target/action event oc
DBTableView.

action

setColumnHeading:newColumnHeading

Sets the view that contains the DBTable's column headings.

columnHeading

setColumnHeadingVisible:(BOOL)flag

Causes the DBTableView to include a heading view across the top of the columns (when flag is Y
flag is NO). This in turn causes the DBTableView to recompute its layout and redisplay in respons

setColumnSelectionOn:(unsigned int)start
:(unsigned int)end
to:(BOOL)flag

Selects (when flag is YES) or deselects (when flag is NO) the block of columns from start to end, i

setDataSource:aSource

Makes aSource the data source from which the DBTableView gets its values, and redisplays the ta

setDelegate:delegate

Makes delegate the DBTableView's delegate. Returns self.

delegate

setDoubleAction:(SEL)aSelector

setGridVisible:(BOOL)flag

Makes grid lines between adjacent rows and columns of the content view visible or not (as flag is Y
the gridlines use is in addition to the intercell spacing. (Row and column headings always have a s
regardless of whether there's a grid in the content view.) Returns self.

isGridVisible, setIntercell:

setHorizScrollerRequired:(BOOL)flag

Includes or omits a horizontal scroller along the lower edge of the DBTableView, as flag is YES o
scroller takes space away from the area otherwise available for the content display. When a scrolle
contains a slider and scroll buttons when the total width of the columns exceeds the width of the di
blank. Returns self.

isHorizScrollerVisible

setIntercell:(const NXSize *)aSize

Sets the number of pixels that separate adjacent rows and columns. The argument aSize specifies t
horizontal and vertical separation. When gridlines are used, the space they use is in addition to the
Returns self.

setMode:(int)newMode

Sets the DBTableView's selection mode. The possible values are member of the enumeration set l
wit:

DB_LISTMODEShift-clicking a vector adds it to the current selection if it is not already select
(Deselecting a vector may not be permitted if it is the only sele
selection is not permitted.)

DB_RADIOMODESelecting a vector automatically deselects the previous selection.

DB_NOSELECTSelecting a vector is not permitted.

setRowHeading:newRowHeading

Sets the view that contains the DBTable's row headings.

rowHeading

setRowHeadingVisible:(BOOL)flag

Selects (when flag is YES) or deselects (when flag is NO) the block of rows from start to end, incl

setTarget:anObject

Makes anObject the target of a target/action message sent in response to an event within the DBTa

setVertScrollerRequired:(BOOL)flag

Includes or omits a vertical scroller along the left edge of the DBTableView, as flag is YES or NO takes space away from the area otherwise available for the content display. When a scroller is incl slider and scroll buttons while when the total width of the columns exceeds the width of the displa blank. Returns self.

isVertScrollerVisible

sizeTo:(NXCoord)width :(NXCoord)height

Resets the overall size of the DBTableView, and then recomputes its layout and redisplays it.

target

Returns the object that is the target for a target/action event in the DBTableView.

tile

Places the DBTableView's three component views (content, column heading, and row headingÐdo have been made visible) within the DBTableView's frame. Returns self.

write:(NXTypedStream *)stream

Archives the DBTableView object by writing it to the NXTypedStream identified by stream. Retu

tableView:sender movedColumnFrom:(unsigned int)old to:(unsigned int)new

Invoked when the user changes the position of a static column. By implementing this method, the corresponding action of its own for example, it might recompute a sort of the displayed record refl sequence of columns. Returns self.

Invoked when the user has changed the selection. The delegate may wish to respond by making co
another display that is synchronized with the TableView that sent the message. Returns self.

(BOOL)tableViewWillChangeSelection:aTableView

Invoked when the user has taken action to change the selection. By implementing this method, the
to interpose some test of its own. Returning YES permits the change in selection to proceed.