

Setting the delegate setDelegate:  
Target and action setAction:

initWithFrame:  
free  
delegate

Setting NXBrowser behavior  
setMultipleSelectionEnabled:  
isMultipleSelectionEnabled  
setBranchSelectionEnabled:  
isBranchSelectionEnabled  
setEmptySelectionEnabled:  
isEmptySelectionEnabled  
reuseColumns:  
setEnabled:  
acceptsFirstResponder  
acceptArrowKeys:andSendActionMessages:  
getTitleFromPreviousColumn:

Configuring controls useScrollBars:  
useScrollButtons:  
setHorizontalScrollButtonsEnabled:  
areHorizontalScrollButtonsEnabled  
setHorizontalScrollerEnabled:  
isHorizontalScrollerEnabled

Setting the NXBrowser's appearance  
setMinColumnWidth:  
minColumnWidth  
setMaxVisibleColumns:  
maxVisibleColumns  
numVisibleColumns  
firstVisibleColumn  
lastVisibleColumn  
lastColumn  
separateColumns:  
columnsAreSeparated

Manipulating columns loadColumnZero  
isLoading  
addColumn  
reloadColumn:  
displayColumn:  
displayAllColumns  
setLastColumn:  
selectAll:  
selectedColumn  
columnOf:  
validateVisibleColumns

Manipulating column titles setTitle:  
isTitled  
setTitle:ofColumn:  
titleOfColumn:  
getTitleFrame:ofColumn:  
titleHeight  
drawTitle:inRect:ofColumn:  
clearTitleInRect:ofColumn:

Scrolling an NXBrowser scrollColumnsLeftBy:  
scrollColumnsRightBy:  
scrollColumnToVisible:  
scrollUpOrDown:  
scrollViaScroller:

getSelectedCells:  
 Getting column frames setFrame:ofColumn:  
 setFrame:ofInsideOfColumn:  
 Manipulating paths setPathSeparator:  
 setPath:  
 getPath:toColumn:  
 Drawing drawSelf::  
 Resizing the NXBrowser sizeTo::  
 sizeToFit  
 Arranging an NXBrowser's components  
 tile

acceptArrowKeys:(BOOL)acceptFlag andSendActionMessages:(BOOL)sendFlag

Sets NXBrowser handling of arrow key input. If acceptFlag is YES, then the keyboard arrow keys whenever the NXBrowser or one of its subviews is the first responder if acceptFlag is NO, arrow keys move the selection. Further, if sendFlag is YES, then when an arrow key is pressed, the NXBrowser's action message is sent to the user had clicked on the new selection if sendFlag is NO, then arrow keys only move the selection (do not send an action message). Returns self.

This method replaces the acceptArrowKeys: method from NEXTSTEP Release 2.

acceptsFirstResponder

(BOOL)acceptsFirstResponder

Returns YES if the NXBrowser accepts arrow key input, NO otherwise. The default setting is NO.

acceptArrowKeys:andSendActionMessages:

(SEL)action

Returns the action sent to the target by the NXBrowser when the user makes a selection in one of its subviews.

setAction:, doubleAction, target

addColumn

Adds a column to the right of the last column in the NXBrowser and, if necessary, scrolls the NXBrowser so that the new column is visible. Your code should never invoke this method it's invoked as needed by doClick: when the user selects a single branch node entry in the NXBrowser, and by setPath: when it matches a path to a branch node entry. Override this method if you need the NXBrowser to do any additional updating when a column is added, but be sure to send this message to super. Returns self.

loadColumnZero, setPath:, reloadColumn:

Returns the NXBrowser's prototype Cell. This Cell is copied to create new Cells in the columns of  
setCellPrototype:

clearTitleInRect:(const NXRect \*)aRect ofColumn:(int)column

Clears the title displayed in aRect above column. Your code shouldn't invoke this method directly  
column title needs to be cleared. You can override this method if you draw your own column titles.  
NXBrowser's coordinate system. Returns self.

(int)columnOf:matrix

Returns the index of the column containing matrix the leftmost (root) column is 0. Returns 1 if no  
matrix.

matrixInColumn:

(BOOL)columnsAreSeparated

Returns YES if columns are separated by a beveled bar NO if they're separated by a black line. If  
to display column titles, its columns are automatically separated by bezels.

separateColumns:, setTitle:

delegate

Returns the delegate of the NXBrowser, the object that provides data for the NXBrowser and responds  
notification messages.

setDelegate:

displayAllColumns

Causes all columns currently visible in the NXBrowser to be redisplayed. This method is useful for  
NXBrowser after manipulating it with display disabled in the window (for instance, if Cells in some  
deleted). Returns self.

displayColumn:

displayColumn:(int)column

Validates and displays column number column. column must already be loaded. This method is useful  
NXBrowser after manipulating column with display disabled in the window. Returns self.

displayAllColumns

You may want to override this method to add specific behavior for mouse clicks.

action, target, doDoubleClick:

doDoubleClick:sender

Your code should never invoke this method. This is the action message sent to the NXBrowser by the sender when a double-click occurs in a column. This method simply sends the double-click action message to the target. If no double-click action message is set, it sends the regular (single-click) action. Returns self.

You may want to override this method to add specific behavior for double-click events.

doubleAction, target, doClick:

(SEL)doubleAction

Returns the action sent by the NXBrowser to its target when the user double-clicks an entry. If no double-click action message has been set, this method returns the regular (single-click) action.

setDoubleAction:, action, target, doDoubleClick:

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

Draws the NXBrowser, loading column 0 if it has not been loaded. Override this method if you want the NXBrowser to draw itself. Your code should never invoke this method it's invoked by the display method.

display (View)

drawTitle:(const char \*)title  
inRect:(const NXRect \*)aRect  
ofColumn:(int)column

Your code should never invoke this method. It's invoked whenever the NXBrowser needs to draw column titles. You may override it if you want your own column titles drawn. Returns self.

(int)firstVisibleColumn

Returns the index of the leftmost visible column.

lastVisibleColumn

free

Frees the NXBrowser and all the objects it manages: ScrollViews, Matrices, Cells, scroll Buttons, etc. Returns nil.

getFrame.ofInsideOfColumn:

(NXRect \*)getFrame:(NXRect \*)theRect ofInsideOfColumn:(int)column

Returns a pointer to the rectangle (in NXBrowser coordinates) containing the "inside" of column n. The pointer is returned both explicitly by the method and by reference in theRect. The "inside" is defined as the area of the column that contains the Cells and only that area (that is, no bezels). If column isn't currently loaded, the method returns NULL, without changing the coordinates of the rectangle represented in theRect. If theRect is NULL.

getFrame:ofColumn:

getLoadedCellAtRow:(int)row inColumn:(int)column

Returns the Cell at row in column, if that column is currently in the NXBrowser. This method creates the Cell if necessary. It's the safest way to get a particular Cell in a column, since lazy delegates don't load cells until they're displayed, and very lazy delegates don't even create all Cells until they're displayed. This method is preferred over cellAt: because it doesn't require a selection. If the specified column isn't in the NXBrowser, or if row doesn't exist in column, returns NULL.

browser:loadCellAtRow:inColumn: (delegate method)

(char \*)getPath:(char \*)thePath toColumn:(int)column

Returns a pointer to the string representing the path to column, both explicitly and by reference in thePath. When invoking this method, you must allocate sufficient memory to accept the entire path string, and set thePath to that memory. If column isn't loaded or thePath is a null pointer, this method returns NULL.

The path is constructed by concatenating the string values in the selected Cells in each column, prepended by the path separator. For example, consider a path separator "@" and an NXBrowser with two columns. If the left column has the string value "fowl" and the selected Cell in the right column has the string value "duck", the resulting path is "@fowl@duck". If there is no selection in the NXBrowser, the path will be "" (an empty string). The default pathSeparator is the slash character ("/").

If multiple selection is enabled and there are multiple Cells selected, this method returns the path of the (lowest) Cell in the last column. Once the path is retrieved, the selected Cells can all be retrieved with the path and their titles used with the path to manipulate the data they represent.

getSelectedCells:, pathSeparator, setPath:

getSelectedCells:(List \*)aList

Fills the provided List with the Cells in the NXBrowser that are selected. The previous contents of aList should be sure aList is empty before invoking this method. If aList is nil, this method creates a new List. Returns aList, or the created List.

selectedCell

(NXRect \*)getTitleFrame:(NXRect \*)theRect ofColumn:(int)column

column to its left, leaving column 0 untitled use setTitleOfColumn: to give column 0 a title. This receiver only when it is titled (that is, when isTitled returns YES).

By default, the NXBrowser is set to get column titles from the previous column. Send this message argument if your delegate implements the browser:titleOfColumn: method or if you use the setTitle: to set all column titles. Returns self.

setTitle:;, setTitle:ofColumn:;, browser:titleOfColumn: (delegate method)

initWithFrame:(const NXRect \*)frameRect

Initializes a new instance of NXBrowser with a bounds of frameRect. The initialized NXBrowser titles, to get titles from previous columns, and to use scrollbars. The minimum column width is set separator is set to the slash (^/°) character. The NXBrowser is set not to clip. This method invoke arrange the components of the NXBrowser (titles, scroll bars, Matrices, and so on).

(BOOL)isBranchSelectionEnabled

Returns YES if a branch node can be selected, NO otherwise.

setBranchSelectionEnabled:

(BOOL)isEmptySelectionEnabled

Returns YES if it is possible to have no Cells selected, NO otherwise.

setEmptySelectionEnabled:

(BOOL)isHorizontalScrollerEnabled

Returns YES if the NXBrowser uses a horizontal Scroller (instead of Buttons), and if that Scroller otherwise.

setHorizontalScrollerEnabled:;, setHorizontalScrollButtonsEnabled:

(BOOL)isMultipleSelectionEnabled

Returns YES if more than one leaf node can be selected, NO otherwise.

setMultipleSelectionEnabled:

(BOOL)isLoading

Returns YES if any of the NXBrowser's columns are loaded.

loadColumnZero, setPath:

keyDown:(NXEvent \*)theEvent

Handles arrow key events. This method is invoked when the NXBrowser or one of its subviews is the NXBrowser has been set to accept arrow keys, and the key represented in theEvent is an arrow scrolls through the NXBrowser in the direction indicated.

acceptArrowKeys:andSendActionMessages:, acceptsFirstResponder, mouseDown:

(int)lastColumn

Returns the index of the last loaded column in the NXBrowser.

lastVisibleColumn

(int)lastVisibleColumn

Returns the index of the rightmost visible column. This may be less than the value returned by lastVisibleColumn if the NXBrowser has been scrolled left.

firstVisibleColumn, lastColumn

loadColumnZero

Loads and displays data in column 0 of the NXBrowser, unloading any columns to the right that were loaded. Invoke this method to force the NXBrowser to be loaded for example, after initializing the NXBrowser delegates, or when changing the data set managed by the delegate. You may want to override this method in a subclass of NXBrowser.

setPath:, reloadColumn:

matrixInColumn:(int)column

Returns the Matrix found in column number column. Returns nil if column number column isn't loaded in the NXBrowser.

(int)maxVisibleColumns

Returns the maximum number of visible columns allowed. No matter how many loaded columns the NXBrowser contains, or how large the NXBrowser is made (for example, by resizing its window), it will never display more than the number of columns. If the number of loaded columns can exceed the value returned by this method, the NXBrowser will display left and right scroll buttons.

setMaxVisibleColumns:, numVisibleColumns, setHorizontalScrollerEnabled:, setHorizontalScrollerEnabled:

mouseDown:(NXEvent \*)theEvent

Handles a mouse down in the NXBrowser's left or right scroll buttons. Override this method if you need custom event processing. Returns self.

keyDown:

(int)numVisibleColumns

Returns the number of columns which can be visible at the same time in the NXBrowser (that is, the number of columns, of the NXBrowser). This may be less than the value returned by maxVisibleColumns if the NXBrowser has been resized.

setMaxVisibleColumns:, maxVisibleColumns:

reflectScroll:clipView

This method updates scroll bars in the column containing clipView. Scroll bars are enabled if a column contains more data than can be displayed at once and disabled if the column can display all data. You should never override this method, but you may want to override it. Returns self.

useScrollBars:

reloadColumn:(int)column

Reloads the previously loaded column number column by sending a message to the delegate to update the Matrix, then reselecting the previously selected Cell if it's still in the Matrix. You should never select a column that hasn't been loaded (you can check for this with the lastColumn method). Redraws the column. Returns self.

lastColumn

reuseColumns:(BOOL)flag

Sets whether the NXBrowser saves a column's Matrix and ClipView or ScrollView when the column is unloaded, and whether it then reuses these subviews when the column is reloaded. If flag is YES, the NXBrowser reuses these subviews, resulting in a somewhat faster display of columns as they are reloaded. If flag is NO, the NXBrowser frees columns when unloaded, reducing average memory use. Returns self.

scrollColumnsLeftBy:(int)shiftAmount

Scrolls the columns in the NXBrowser left by shiftAmount columns, making higher numbered columns visible. If shiftAmount exceeds the number of loaded columns to the right of the first visible column, then the first visible column becomes the last visible column. Redraws and returns self.

scrollColumnsRightBy:

`scrollColumnToVisible:(int)column`

Scrolls the NXBrowser to make column number column visible. If there's no column in the NXBrowser scrolls to the right as far as possible. Redraws and returns self.

`scrollUpOrDown:sender`

Scrolls a column up or down. Your code shouldn't send this message NXBrowser receives it from buttons. You may want to override it to perform additional updating. Returns self.

`scrollViaScroller:sender`

Scrolls the NXBrowser's columns left or right based on the position of the Scroller sending the message. Your code shouldn't send this message, but you may want to override it. Returns self.

`selectAll:sender`

Selects all entries in the last column loaded in the NXBrowser if multiple selection is allowed. Returns self. See `setMultipleSelectionEnabled:`

`selectedCell`

If there is a selection, returns the last selected Cell. "Last," in this context, means furthest to the right column.

`getSelectedCells:`

`(int)selectedColumn`

Returns the column number of the rightmost column containing a selected Cell. This won't be the column number of the selected Cell isn't a leaf. Returns 1 if no column in the NXBrowser contains a selected Cell.

`lastColumn`

`sendAction`

Sends the NXBrowser's action to its target and returns self.

`sendAction:to: (Control)`

`separateColumns:(BOOL)flag`

Sets the action of the NXBrowser. aSelector is the selector for the message sent to the NXBrowser when a mouse-down event occurs in a column of the NXBrowser. Returns self.

action, setDoubleAction:, setTarget:, doClick

setBranchSelectionEnabled:(BOOL)flag

Sets whether the user can select multiple branch and leaf node entries. If flag is YES and multiple (by setMultipleSelectionEnabled:), then multiple branch and leaf node entries can be selected. By choose only multiple leaf node entries when multiple entry selection is enabled. Returns self.

This method replaces the allowBranchSel: method from NEXTSTEP Release 2.

isBranchSelectionEnabled, setMultipleSelectionEnabled:

setCellClass:classId

Sets the class of Cell used when adding Cells to a Matrix in a column of the NXBrowser. classId is returned when sending the class message to NXBrowserCell or a subclass of NXBrowserCell. Since NXBrowser always has its Matrices copy prototype Cells, this method simply makes a prototype, sends it an in that prototype. Returns self.

You shouldn't use Control's class method setCellClass: with an NXBrowser.

setCellPrototype:

setCellPrototype:aCell

Sets aCell as the Cell prototype copied when adding Cells to the Matrices in the columns of NXBrowser. Returns an instance of NXBrowserCell or a subclass of NXBrowserCell. Each Matrix gets its own copy of prototype, and will free that copy when the Matrix is freed. Returns the previous Cell prototype if either NXBrowserCell or a subclass of NXBrowserCell otherwise, returns nil.

You shouldn't use Control's class method setCellClass: with an NXBrowser.

cellPrototype, setCellClass:

setDelegate:anObject

Sets the delegate of the NXBrowser to anObject and returns self. If anObject is of a class that implements the delegate protocol indicated in the class description for a normal, lazy, or very lazy delegate, it's set as the NXBrowser's delegate. If the delegate is set to nil. Returns self.

delegate

setDoubleAction:(SEL)aSelector

Cell selected. By default, the setting is NO, and NXBrowser selects the first item in the first column.  
isEmptySelectionEnabled, setMultipleSelectionEnabled:

setEnabled:(BOOL)flag

Enables the NXBrowser when flag is YES disables it when flag is NO. Returns self.

isEnabled (Control)

setHorizontalScrollButtonsEnabled:(BOOL)flag

If flag is YES, sets the NXBrowser to use left and right scroll buttons. Generally, you should allow scroll horizontally unless your data is nonhierarchical, thus limited to a single column, or restricted so that NXBrowser will always display enough columns for all data. Returns self.

This method, along with setHorizontalScrollerEnabled:, replaces the hideLeftAndRightScrollButtons method in NEXTSTEP Release 2. In order to completely remove horizontal scrolling Controls, invoke both methods with arguments of NO.

setHorizontalScrollerEnabled:, areHorizontalScrollButtonsEnabled

setHorizontalScrollerEnabled:(BOOL)flag

If flag is YES, sets the NXBrowser to use a horizontal Scroller. Generally, you should allow your data to scroll horizontally unless your data is nonhierarchical, thus limited to a single column, or restricted so that NXBrowser will always display enough columns for all data. Returns self.

This method, along with setHorizontalScrollButtonsEnabled:, replaces the hideLeftAndRightScrollButtons method in NEXTSTEP Release 2. In order to completely remove horizontal scrolling Controls, invoke both methods with arguments of NO.

setHorizontalScrollButtonsEnabled:, areHorizontalScrollButtonsEnabled

setLastColumn:(int)column

Makes column number column the last column loaded and displayed by the NXBrowser. Removes columns to the right of column from the NXBrowser, and scrolls columns in the NXBrowser to make the new last column the one that wasn't previously. If column number column isn't already loaded, this method does nothing. Returns self.

lastColumn

setMatrixClass:className

Sets the class of Matrix used when adding new columns to the NXBrowser. className must be the name of the class message to Matrix or a subclass of Matrix otherwise this method retains the previous setting. Returns NXBrowser's Matrix class. NXBrowser initializes the Matrix of a new column with the initWithFrame:numRows:numCols: method. Returns self.

be reestablished to its desired value.

maxVisibleColumns, setMinColumnWidth:

setMinColumnWidth:(int)columnWidth

Sets the minimum width for each column to columnWidth. If the new minimum width is different this method also redisplay the NXBrowser with columns set to the new width. columnWidth is in points rounded to the nearest integer. The default setting is 100. Returns self.

minColumnWidth

setMultipleSelectionEnabled:(BOOL)flag

Sets whether the user can select multiple entries in a column. If flag is YES, the user can choose multiple entries in a column (or leaf and branch entries in a column if enabled by setBranchSelectionEnabled). If flag is NO, the user can choose just one entry in a column at a time. Returns self.

This method replaces the allowMultiSel: method from NEXTSTEP Release 2.

isMultipleSelectionEnabled, setBranchSelectionEnabled:

setPath:(const char \*)aPath

Parses aPath a string consisting of one or more substrings separated by the path separator and searches the NXBrowser that match the substrings. If the first character in aPath is the path separator, this method begins searching for matches in column 0 otherwise, it begins searching in the last column loaded. If no column is loaded, this method loads column 0 and begins the search there. While parsing the current substring, it tries to find an exact match in the search column. If it finds an exact match, this method selects that entry and moves to the next column (if necessary) to search for the next substring.

If this method finds a valid path (one in which each substring is matched by an entry in the corresponding column) it returns self. If it doesn't find an exact match on a substring, it stops parsing aPath and returns nil. Entries that it has already selected remain selected.

Your code should never try to set a path or select items by sending Cell selection messages to the NXBrowser's columns. This bypasses every mechanism that allows the NXBrowser to update its columns and Cells properly.

getPath:toColumn, pathSeparator, setPathSeparator, browser:selectCell:inColumn: (delegate method)

setPathSeparator:(unsigned short)charCode

Sets the character used as the path separator the default is the slash character (^/°). Returns self.

getPath:toColumn, setPath:

setTarget:anObject

nothing. Returns self.

getTitleFromPreviousColumn:, setTitle:, browser:TitleOfColumn: (delegate method)

setTitle:(BOOL)flag

If flag is YES, columns display titles and are separated by beveled borders. Returns self.

getTitleFromPreviousColumn:, setTitle:ofColumn:, browser:TitleOfColumn: (delegate method)

sizeTo:(NXCoord)width :(NXCoord)height

Resizes the NXBrowser to the new width and height. Usually sent by the NXBrowser's Window. If your subclass of NXBrowser need to do any special updating when resized. Returns self.

sizeToFit

Resizes the NXBrowser to contain all the columns and Controls displayed in it. Returns self.

target

Returns the target for the NXBrowser's action message. This object receives action messages when double-clicks on items in the NXBrowser.

setTarget:, action, setAction:, doubleAction, setDoubleAction:

tile

Arranges the various subviews of NXBrowserDScrollers, columns, titles, and so on without redrawing. You shouldn't send this message. It's invoked any time the appearance of the NXBrowser changes for example Buttons or scroll bars are set, a column is added, and so on. Override this method if your code changes the NXBrowser (for example, if you draw your own titles above columns). Returns self.

(NXCoord)titleHeight

Returns the height of titles drawn above the columns of the NXBrowser. Override this method if you draw titles above the NXBrowser's columns.

(const char \*)titleOfColumn:(int)column

Returns a pointer to the title string displayed above column number column. If that column isn't in the NXBrowser, returns NULL.

useScrollBars:(BOOL)flag

Sets NXBrowser to use scroll bars for its columns based on flag. If flag is YES and the NXBrowser uses only scroll Buttons, this method causes the scroll Buttons to display at the base of the scroll bars. If the NXBrowser uses scroll bars without Buttons send useScrollButtons: to also have Buttons at the base of the scroll bars. This method redraws the NXBrowser. Returns self.

useScrollButtons:

useScrollButtons:(BOOL)flag

Sets the NXBrowser to use scroll Buttons for its columns based on flag. If flag is YES and the NXBrowser previously set to use scroll bars, this method causes scroll buttons to display at the base of the scroll bars. If scroll Buttons are displayed underneath each column. To get only Buttons under the columns, send useScrollButtons: YES as the argument, then send this message with YES as the argument. This method redraws the NXBrowser.

useScrollBars:

validateVisibleColumns

Validates the columns visible in the NXBrowser by invoking the delegate method browser:columnIsVisible:inColumn: for each column. Use this method to confirm that the entries displayed in each visible column are valid before displaying them.

browser:columnIsValid: (delegate method)

(BOOL)browser:sender columnIsValid:(int)column

This method is invoked by NXBrowser's validateVisibleColumns method to determine whether the data loaded in column number column need to be updated. This is useful for data sets that may change in a file system, or lists from a shared set of data that others can change. Returns YES if the content is valid, otherwise.

browser:selectCell:inColumn: (delegate method)

browserDidScroll:sender

Notifies the delegate when the browser has finished scrolling horizontally. This can be useful for a delegate to update interface items with the columns of the browser (for example, an icon path or a series of pop-up lists).

browserDidScroll: (delegate method)

(int)browser:sender

fillMatrix:matrix

inColumn:(int)column

browser:loadCell:atRow:inColumn: (delegate method)

(int)browser:sender getNumRowsInColumn:(int)column

Implemented by very lazy delegates, this method is invoked by the NXBrowser to ask the delegate in column number column. This method allows the NXBrowser to resize its scroll bar for a column the Cells in that column. Returns the number of rows in column.

If you implement this method, don't implement the delegate method browser:fillMatrix:inColumn:

browser:loadCell:atRow:inColumn: (delegate method)

browser:sender

loadCell:cell

atRow:(int)row

inColumn:(int)column

Implemented by lazy and very lazy delegates. This method loads the entry in the provided NXBrowser specified row and column in the NXBrowser. The NXBrowser will resize the Cell to fit in the Matrix the size of an NXBrowserCell. Returns self.

A lazy delegate should send a setLoaded: message to cell at load time it can send setLeaf:, setString:, setEnabled: messages at load time or later. A very lazy delegate should send setLoaded:, setLeaf:, messages to cell at load time, and setEnabled: when needed.

browser:fillMatrix:inColumn: (delegate method), browser:getNumRowsInColumn: (delegate method)

(BOOL)browser:sender

selectCell:(const char \*)entry

inColumn:(int)column

Asks NXBrowser's delegate to validate and select an entry in column number column. This method corresponding to entry if necessary, send it setLoaded: and setLeaf: messages as needed to indicate column's Matrix a selectCell: or selectCellAt:: message to select that Cell. If there is no Cell corresponding selection should be cleared by sending selectCellAt:: to the Matrix with 1 and 1 as the arguments. This method successfully selects the Cell corresponding to entry in column NO otherwise.

If the delegate doesn't implement this method, the NXBrowser searches for entries by scanning through Cells in the column. This will always work properly for NXBrowsers that browse static data. However, change while the NXBrowser is in use (for example, if a new file is created or deleted), this method will find that new data and add it to the column, or to find out that it no longer exists (or that its status has changed) it as disabled or remove the Cell altogether with Matrix's removeRowAt:andFree: method (be sure to call removeRowAt:andFree: on the Matrix).

browser:columnIsValid: (delegate method), matrixInColumn:, selectCell: (Matrix), selectCellAt: (Matrix), removeRowAt:andFree: (Matrix)

(const char \*)browser:sender titleOfColumn:(int)column

Invoked by NXBrowser to get the title for column from the delegate. This method is invoked if the browser is titled, but only when the NXBrowser is titled and has received a getTitleFromPreviousColumn: message.

user interface items to prepare for aligning them with the columns of the browser (for example, and pop-up lists). Returns self.

browserDidScroll: (delegate method)