
NSBrowser Class Reference

[Cocoa > Events & Other Input](#)



2007-04-05



Apple Inc.
© 2007 Apple Inc.
All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, electronic, photocopying, recording, or otherwise, without prior written permission of Apple Inc., with the following exceptions: Any person is hereby authorized to store documentation on a single computer for personal use only and to print copies of documentation for personal use provided that the documentation contains Apple's copyright notice.

The Apple logo is a trademark of Apple Inc.

Use of the "keyboard" Apple logo (Option-Shift-K) for commercial purposes without the prior written consent of Apple may constitute trademark infringement and unfair competition in violation of federal and state laws.

No licenses, express or implied, are granted with respect to any of the technology described in this document. Apple retains all intellectual property rights associated with the technology described in this document. This document is intended to assist application developers to develop applications only for Apple-labeled computers.

Every effort has been made to ensure that the information in this document is accurate. Apple is not responsible for typographical errors.

Apple Inc.
1 Infinite Loop
Cupertino, CA 95014
408-996-1010

Apple, the Apple logo, Cocoa, Mac, and Mac OS are trademarks of Apple Inc., registered in the United States and other countries.

Simultaneously published in the United States and Canada.

Even though Apple has reviewed this document, APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS DOCUMENT IS PROVIDED "AS IS," AND YOU, THE READER, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.

IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY

DEFECT OR INACCURACY IN THIS DOCUMENT, even if advised of the possibility of such damages.

THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESS OR IMPLIED. No Apple dealer, agent, or employee is authorized to make any modification, extension, or addition to this warranty.

Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.

Contents

NSBrowser Class Reference 7

Overview	7
Tasks	8
Configuring Browsers	8
Getting Browser Information	9
Managing Component Types	9
Managing Selection Behavior	9
Managing Selection	10
Accessing Components	10
Managing the Path	11
Managing Columns	11
Accessing Column Titles	12
Scrolling	12
Dragging	13
Getting Column Frames	13
Managing Actions	13
Handling Mouse-Click Events	14
Sizing	14
Displaying Expansion Tooltips	15
Class Methods	15
cellClass	15
removeSavedColumnsWithAutosaveName:	15
Instance Methods	16
acceptsArrowKeys	16
addColumn	16
allowsBranchSelection	16
allowsEmptySelection	17
allowsMultipleSelection	17
allowsTypeSelect	17
backgroundColor	18
canDragRowsWithIndexes:inColumn:withEvent:	18
cellPrototype	19
columnContentWidthForColumnWidth:	19
columnOfMatrix:	20
columnResizingType	20
columnsAutosaveName	20
columnWidthForColumnContentWidth:	21
delegate	21
displayAllColumns	22
displayColumn:	22
doClick:	22

doDoubleClick: 23
doubleAction 23
draggingImageForRowsWithIndexes:inColumn:withEvent:offset: 23
draggingSourceOperationMaskForLocal: 24
drawTitleOfColumn:inRect: 24
firstVisibleColumn 25
frameOfColumn: 25
frameOfInsideOfColumn: 25
hasHorizontalScroller 26
isLoading 26
isOpaque 26
isTitled 27
lastColumn 27
lastVisibleColumn 27
loadColumnZero 28
loadedCellAtRow:column: 28
matrixClass 28
matrixInColumn: 29
maxVisibleColumns 29
minColumnWidth 29
namesOfPromisedFilesDroppedAtDestination: 30
numberOfVisibleColumns 30
path 30
pathSeparator 31
pathToColumn: 31
prefersAllColumnUserResizing 32
reloadColumn: 32
reusesColumns 33
scrollColumnsLeftBy: 33
scrollColumnsRightBy: 33
scrollColumnToVisible: 34
scrollViaScroller: 34
selectAll: 34
selectedCell 35
selectedCellInColumn: 35
selectedCells 36
selectedColumn 36
selectedRowInColumn: 36
selectedRowIndexesInColumn: 37
selectRow:inColumn: 37
selectRowIndexes:inColumn: 38
sendAction 38
sendsActionOnArrowKeys 38
separatesColumns 39
setAcceptsArrowKeys: 39
setAllowsBranchSelection: 39

setAllowsEmptySelection:	40
setAllowsMultipleSelection:	40
setAllowsTypeSelect:	40
setBackgroundColor:	41
setCellClass:	41
setCellPrototype:	42
setColumnResizingType:	42
setColumnsAutosaveName:	42
setDelegate:	43
setDoubleAction:	43
setDraggingSourceOperationMask:forLocal:	44
setHasHorizontalScroller:	44
setLastColumn:	45
setMatrixClass:	45
setMaxVisibleColumns:	45
setMinColumnWidth:	46
setPath:	46
setPathSeparator:	47
setPrefersAllColumnUserResizing:	47
setReusesColumns:	48
setSendsActionOnArrowKeys:	48
setSeparatesColumns:	48
setTakesTitleFromPreviousColumn:	49
setTitle:ofColumn:	49
setTitled:	50
setWidth:ofColumn:	50
takesTitleFromPreviousColumn	51
tile	51
titleFrameOfColumn:	51
titleHeight	52
titleOfColumn:	52
updateScroller	52
validateVisibleColumns	53
widthOfColumn:	53
Delegate Methods	53
browser:acceptDrop:atRow:column:dropOperation:	53
browser:canDragRowsWithIndexes:inColumn:withEvent:	54
browser:createRowsForColumn:inMatrix:	55
browser:draggingImageForRowsWithIndexes:inColumn:withEvent:offset:	55
browser:isColumnValid:	56
browser:namesOfPromisedFilesDroppedAtDestination:forDraggedRowsWithIndexes: inColumn:	56
browser:nextTypeSelectMatchFromRow:toRow:inColumn:forString:	57
browser:numberOfRowsInColumn:	58
browser:selectCellWithString:inColumn:	58
browser:selectRow:inColumn:	59

- browser:shouldShowCellExpansionForRow:column: 60
- browser:shouldSizeColumn:forUserResize:toWidth: 60
- browser:shouldTypeSelectForEvent:withCurrentSearchString: 61
- browser:sizeToFitWidthOfColumn: 61
- browser:titleOfColumn: 62
- browser:typeSelectStringForRow:inColumn: 62
- browser:validateDrop:proposedRow:column:dropOperation: 63
- browser:willDisplayCell:atRow:column: 64
- browser:writeRowsWithIndexes:inColumn:toPasteboard: 65
- browserColumnConfigurationDidChange: 66
- browserDidScroll: 66
- browserWillScroll: 66
- Constants 67
 - NSBrowserColumnResizingType 67
 - NSBrowserDropOperation 68
- Notifications 68
 - NSBrowserColumnConfigurationDidChangeNotification 68

Document Revision History 69

Index 71

NSBrowser Class Reference

Inherits from	NSControl : NSView : NSResponder : NSObject
Conforms to	NSAnimatablePropertyContainer (NSView) NSCoding (NSResponder) NSObject (NSObject)
Framework	/System/Library/Frameworks/AppKit.framework
Availability	Available in Mac OS X v10.0 and later.
Companion guide	Browsers
Declared in	NSBrowser.h
Related sample code	NewsReader

Overview

This class provides a user interface for displaying and selecting items from a list of data or from hierarchically organized lists of data such as directory paths. Instances of this class are known as **browsers**. When working with a hierarchy of data, the levels are displayed in columns, which are indexed from left to right.

This class uses the NSBrowserCell class to implement its user interface.

Browsers have the following components:

- Columns
- Scroll views
- Matrices
- Browser cells

To the user, browsers display data in columns and rows within each column. These components are arranged in the following component hierarchy:

```
Browser
|---Columns [1..*]
    |---Scroll view
        |---Matrix
            |---Rows [0..*]
```

Tasks

Configuring Browsers

- [reusesColumns](#) (page 33)
Returns a Boolean value indicating whether the browser reuses `NSMatrix` objects after their columns are unloaded.
- [setReusesColumns:](#) (page 48)
Specifies whether `NSMatrix` objects are reused.
- [maxVisibleColumns](#) (page 29)
Returns the maximum number of visible columns.
- [setMaxVisibleColumns:](#) (page 45)
Sets the maximum number of columns displayed.
- [backgroundColor](#) (page 18)
Provides the receiver's background color. Default: `[NSColor whiteColor]`.
- [setBackgroundColor:](#) (page 41)
Specifies the receiver's background color.
- [minColumnWidth](#) (page 29)
Returns the minimum column width.
- [setMinColumnWidth:](#) (page 46)
Sets the minimum column width.
- [separatesColumns](#) (page 39)
Returns a Boolean value indicating whether columns are separated by beveled borders.
- [setSeparatesColumns:](#) (page 48)
Sets whether to separate columns with beveled borders.
- [takesTitleFromPreviousColumn](#) (page 51)
Returns a Boolean value indicating whether a column takes its title from the selected cell in the previous column.
- [setTakesTitleFromPreviousColumn:](#) (page 49)
Sets whether the title of a column is set to the string value of the selected cell in the previous column.
- [tile](#) (page 51)
Adjusts the various subviews of the receiver—scrollers, columns, titles, and so on—without redrawing.
- [acceptsArrowKeys](#) (page 16)
Returns a Boolean value indicating whether the browser allows navigation using the arrow keys.
- [setAcceptsArrowKeys:](#) (page 39)
Specifies whether the browser allows navigation using the arrow keys.
- [delegate](#) (page 21)
Returns the receiver's delegate.
- [setDelegate:](#) (page 43)
Sets the receiver's delegate.

Getting Browser Information

- [isOpaque](#) (page 26)
Indicates whether the receiver is opaque.
- [browser:isColumnValid:](#) (page 56) *delegate method*
Returns whether the contents of the specified column are valid.
- [browser:numberOfRowsInColumn:](#) (page 58) *delegate method*
Returns the number of rows of data in the specified column.
- [browser:titleOfColumn:](#) (page 62) *delegate method*
Asks the delegate for the title to display above the specified column.

Managing Component Types

- + [cellClass](#) (page 15)
Returns the `NSBrowserCell` class.
- [setCellClass:](#) (page 41)
Sets the class of `NSCell` used by the matrices in the columns of the receiver.
- [cellPrototype](#) (page 19)
Returns the receiver's prototype `NSCell`.
- [setCellPrototype:](#) (page 42)
Sets the `NSCell` instance copied to display items in the matrices in the columns of the receiver.
- [matrixClass](#) (page 28)
Returns the matrix class used in the receiver's columns.
- [setMatrixClass:](#) (page 45)
Sets the matrix class used in the receiver's columns.

Managing Selection Behavior

- [allowsBranchSelection](#) (page 16)
Returns a Boolean value indicating whether the user can select branch items.
- [setAllowsBranchSelection:](#) (page 39)
Sets whether the user can select branch items.
- [allowsEmptySelection](#) (page 17)
Returns a Boolean value indicating whether there can be nothing selected.
- [setAllowsEmptySelection:](#) (page 40)
Sets whether there can be nothing selected.
- [allowsMultipleSelection](#) (page 17)
Returns a Boolean value indicating whether the user can select multiple items.
- [setAllowsMultipleSelection:](#) (page 40)
Sets whether the user can select multiple items.
- [selectedRowIndexesInColumn:](#) (page 37)
Provides the indexes of the selected rows in a given column of the receiver.

- [selectRowIndexes:inColumn:](#) (page 38)
Specifies the selected rows in a given column of the receiver.
- [allowsTypeSelect](#) (page 17)
Indicates whether the receiver allows keystroke-based selection (type select).
- [setAllowsTypeSelect:](#) (page 40)
Specifies whether the receiver allows keystroke-based selection.
- [browser:shouldTypeSelectForEvent:withCurrentSearchString:](#) (page 61) *delegate method*
Sent to the delegate to determine whether keyboard-based selection (type select) for a given event and search string should proceed.
- [browser:typeSelectStringForRow:inColumn:](#) (page 62) *delegate method*
Sent to the delegate to get the keyboard-based selection (type select) string for a given row and column.
- [browser:nextTypeSelectMatchFromRow:toRow:inColumn:forString:](#) (page 57) *delegate method*
Sent to the delegate to customize a browser's keyboard-based selection (type select) behavior.

Managing Selection

- [selectedCell](#) (page 35)
Returns the last (rightmost and lowest) selected NSCell.
- [selectedCellInColumn:](#) (page 35)
Returns the last (lowest) NSCell selected in the given column.
- [selectedCells](#) (page 36)
Returns all cells selected in the rightmost column.
- [selectAll:](#) (page 34)
Selects all NSCell objects in the last column of the receiver.
- [selectedRowInColumn:](#) (page 36)
Returns the row index of the selected cell in the specified column.
- [selectRow:inColumn:](#) (page 37)
Selects the cell at the specified row and column index.
- [browser:selectCellWithString:inColumn:](#) (page 58) *delegate method*
Asks the delegate to select the NSCell with the given title in the specified column.
- [browser:selectRow:inColumn:](#) (page 59) *delegate method*
Asks the delegate to select the NSCell at the specified row and column location.

Accessing Components

- [loadedCellAtRow:column:](#) (page 28)
Loads, if necessary, and returns the NSCell at the specified row and column location.
- [matrixInColumn:](#) (page 29)
Returns the matrix located in the specified column.

Managing the Path

- [path](#) (page 30)
Returns a string representing the receiver's current path.
- [setPath:](#) (page 46)
Sets the path displayed by the receiver.
- [pathToColumn:](#) (page 31)
Returns a string representing the path from the first column up to, but not including, the column at the given index.
- [pathSeparator](#) (page 31)
Returns the path separator.
- [setPathSeparator:](#) (page 47)
Sets the path separator.

Managing Columns

- [addColumn](#) (page 16)
Adds a column to the right of the last column.
- [displayAllColumns](#) (page 22)
Updates the receiver to display all loaded columns.
- [displayColumn:](#) (page 22)
Updates the receiver to display the given column.
- [columnOfMatrix:](#) (page 20)
Returns the column number in which the given matrix is located.
- [selectedColumn](#) (page 36)
Returns the index of the last column with a selected item.
- [lastColumn](#) (page 27)
Returns the index of the last column loaded.
- [setLastColumn:](#) (page 45)
Sets the last column.
- [firstVisibleColumn](#) (page 25)
Returns the index of the first visible column.
- [numberOfVisibleColumns](#) (page 30)
Returns the number of columns visible.
- [lastVisibleColumn](#) (page 27)
Returns the index of the last visible column.
- [validateVisibleColumns](#) (page 53)
Invokes the delegate method [browser:isColumnValid:](#) (page 56) for visible columns.
- [isLoading](#) (page 26)
Returns whether column 0 is loaded.
- [loadColumnZero](#) (page 28)
Loads column 0; unloads previously loaded columns.
- [reloadColumn:](#) (page 32)
Reloads the given column if it exists and sets it to be the last column.

- `browser:createRowsForColumn:inMatrix:` (page 55) *delegate method*
Creates a row in the given matrix for each row of data to be displayed in the specified column of the browser.
- `browser:willDisplayCell:atRow:column:` (page 64) *delegate method*
Gives the delegate the opportunity to modify the specified cell at the given row and column position before it's displayed by the `NSBrowser`.

Accessing Column Titles

- `titleOfColumn:` (page 52)
Returns the title displayed for the given column.
- `setTitle:ofColumn:` (page 49)
Sets the title of the given column.
- `isTitled` (page 27)
Returns a Boolean value indicating whether columns display titles.
- `setTitled:` (page 50)
Sets whether columns display titles.
- `drawTitleOfColumn:inRect:` (page 24)
Draws the title for the specified column within the given rectangle.
- `titleHeight` (page 52)
Returns the height of column titles.
- `titleFrameOfColumn:` (page 51)
Returns the bounds of the title frame for the specified column.

Scrolling

- `updateScroller` (page 52)
Updates the horizontal scroller to reflect column positions.
- `hasHorizontalScroller` (page 26)
Returns a Boolean value indicating whether the browser has a horizontal scroller.
- `setHasHorizontalScroller:` (page 44)
Sets whether an `NSScroller` is used to scroll horizontally.
- `scrollColumnToVisible:` (page 34)
Scrolls to make the specified column visible.
- `scrollColumnsLeftBy:` (page 33)
Scrolls columns left by the specified number of columns.
- `scrollColumnsRightBy:` (page 33)
Scrolls columns right by the specified number of columns.
- `scrollViaScroller:` (page 34)
Scrolls columns left or right based on an `NSScroller`.
- `browserWillScroll:` (page 66) *delegate method*
Notifies the delegate when the `NSBrowser` will scroll.
- `browserDidScroll:` (page 66) *delegate method*
Notifies the delegate when the `NSBrowser` has scrolled.

Dragging

- [draggingSourceOperationMaskForLocal:](#) (page 24)
Indicates the types of dragging operations the source object allows on the dragged image's data.
- [setDraggingSourceOperationMask:forLocal:](#) (page 44)
Specifies the drag-operation mask for dragging operations with local or external destinations.
- [canDragRowsWithIndexes:inColumn:withEvent:](#) (page 18)
Indicates whether the receiver can attempt to initiate a drag of the given rows for the given event.
- [browser:canDragRowsWithIndexes:inColumn:withEvent:](#) (page 54) *delegate method*
Sent to the delegate to determine whether the browser can attempt to initiate a drag of the given rows for the given event.
- [draggingImageForRowsWithIndexes:inColumn:withEvent:offset:](#) (page 23)
Provides an image to represent dragged rows during a drag operation on the receiver.
- [browser:draggingImageForRowsWithIndexes:inColumn:withEvent:offset:](#) (page 55) *delegate method*
Sent to the delegate to obtain an image to represent dragged rows during a drag operation on a browser.
- [browser:validateDrop:proposedRow:column:dropOperation:](#) (page 63) *delegate method*
Sent to the delegate during a dragging session to determine whether to accept a drop, and to obtain the drop location. Required for a browser to be a drag destination.
- [browser:acceptDrop:atRow:column:dropOperation:](#) (page 53) *delegate method*
Sent to the delegate during a dragging session to determine whether to accept the drop. Required for a browser to be a drag destination.
- [namesOfPromisedFilesDroppedAtDestination:](#) (page 30)
Provides the names of the files that the receiver promises to create at a specified location.
- [browser:writeRowsWithIndexes:inColumn:toPasteboard:](#) (page 65) *delegate method*
Determines whether a drag operation can proceed. Required for a browser to be a drag source.
- [browser:namesOfPromisedFilesDroppedAtDestination:forDraggedRowsWithIndexes:inColumn:](#) (page 56) *delegate method*
Implements file-promise drag operations.

Getting Column Frames

- [frameOfColumn:](#) (page 25)
Returns the rectangle containing the given column.
- [frameOfInsideOfColumn:](#) (page 25)
Returns the rectangle containing the specified column, not including borders.

Managing Actions

- [doubleAction](#) (page 23)
Returns the receiver's double-click action method.
- [setDoubleAction:](#) (page 43)
Sets the receiver's double-click action.

- [sendsActionOnArrowKeys](#) (page 38)
Returns a Boolean value indicating whether pressing an arrow key causes an action message to be sent.
- [setSendsActionOnArrowKeys:](#) (page 48)
Sets whether pressing an arrow key causes the action message to be sent.
- [sendAction](#) (page 38)
Sends the action message to the target.

Handling Mouse-Click Events

- [doClick:](#) (page 22)
Responds to (single) mouse clicks in a column of the receiver.
- [doDoubleClick:](#) (page 23)
Responds to double clicks in a column of the receiver.

Sizing

- + [removeSavedColumnsWithAutosaveName:](#) (page 15)
Removes the column configuration data stored under the given name from the application's user defaults.
- [columnsAutosaveName](#) (page 20)
Returns the name used to automatically save the receiver's column configuration.
- [setColumnsAutosaveName:](#) (page 42)
Sets the name used to automatically save the receiver's column configuration.
- [columnContentWidthForColumnWidth:](#) (page 19)
Given the column width, returns the content width.
- [columnWidthForColumnContentWidth:](#) (page 21)
Given the content width, returns the column width.
- [columnResizingType](#) (page 20)
Returns the receiver's column resizing type.
- [setColumnResizingType:](#) (page 42)
Sets the receiver's column resizing type.
- [prefersAllColumnUserResizing](#) (page 32)
Returns a Boolean value indicating if the browser is set to resize all columns simultaneously rather than resizing a single column at a time.
- [setPrefersAllColumnUserResizing:](#) (page 47)
Specifies whether the browser resizes all columns simultaneously rather than resizing a single column at a time.
- [widthOfColumn:](#) (page 53)
Returns the width of the specified column.
- [setWidth:ofColumn:](#) (page 50)
Sets the width of the specified column.
- [browser:shouldSizeColumn:forUserResize:toWidth:](#) (page 60) *delegate method*
Used for determining a column's initial size.

- [browser:sizeToFitWidthOfColumn:](#) (page 61) *delegate method*
Returns the ideal width for a column.
- [browserColumnConfigurationDidChange:](#) (page 66) *delegate method*
Used by clients to implement their own column width persistence.

Displaying Expansion Tooltips

- [browser:shouldShowCellExpansionForRow:column:](#) (page 60) *delegate method*
Determines whether an expansion tooltip appears for a cell at the given row in the given column.

Class Methods

cellClass

Returns the `NSBrowserCell` class.

```
+ (Class)cellClass
```

Return Value

Always returns the `NSBrowserCell` class (even if the developer has sent a [setCellClass:](#) (page 41) message to a particular instance).

Discussion

This method is used by `NSControl` during initialization and is not meant to be used by applications.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [cellPrototype](#) (page 19)
- [setCellPrototype:](#) (page 42)

Declared In

`NSBrowser.h`

removeSavedColumnsWithAutosaveName:

Removes the column configuration data stored under the given name from the application's user defaults.

```
+ (void)removeSavedColumnsWithAutosaveName:(NSString *)name
```

Parameters

name

The name of the column configuration data to remove.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [columnsAutosaveName](#) (page 20)
- [setColumnsAutosaveName:](#) (page 42)

Declared In

NSBrowser.h

Instance Methods

acceptsArrowKeys

Returns a Boolean value indicating whether the browser allows navigation using the arrow keys.

- (BOOL)acceptsArrowKeys

Return Value

YES if the arrow keys are enabled; otherwise NO.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [setAcceptsArrowKeys:](#) (page 39)

Declared In

NSBrowser.h

addColumn

Adds a column to the right of the last column.

- (void)addColumn

Availability

Available in Mac OS X v10.0 and later.

See Also

- [columnOfMatrix:](#) (page 20)
- [displayColumn:](#) (page 22)
- [selectedColumn](#) (page 36)

Declared In

NSBrowser.h

allowsBranchSelection

Returns a Boolean value indicating whether the user can select branch items.

- (BOOL)allowsBranchSelection

Return Value

YES if the user can select branch items when multiple selection is enabled; otherwise NO.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [setAllowsBranchSelection:](#) (page 39)

Declared In

NSBrowser.h

allowsEmptySelection

Returns a Boolean value indicating whether there can be nothing selected.

- (BOOL)allowsEmptySelection

Return Value

YES if the browser allows the selection to be empty; otherwise NO.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [setAllowsEmptySelection:](#) (page 40)

Declared In

NSBrowser.h

allowsMultipleSelection

Returns a Boolean value indicating whether the user can select multiple items.

- (BOOL)allowsMultipleSelection

Return Value

YES if the browser allows the user to select multiple items at once; otherwise NO.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [setAllowsMultipleSelection:](#) (page 40)

Declared In

NSBrowser.h

allowsTypeSelect

Indicates whether the receiver allows keystroke-based selection (type select).

- (BOOL)allowsTypeSelect

Return Value

YES (default) when the receiver allows keystroke-based selection, NO otherwise.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [setAllowsTypeSelect:](#) (page 40)

Declared In

NSBrowser.h

backgroundColor

Provides the receiver's background color. Default: [NSColor whiteColor].

- (NSColor *)backgroundColor

Return Value

The receiver's background color.

Default: [NSColor whiteColor]

Availability

Available in Mac OS X v10.5 and later.

See Also

- [setBackgroundcolor:](#) (page 41)

- [isOpaque](#) (page 26)

Declared In

NSBrowser.h

canDragRowsWithIndexes:inColumn:withEvent:

Indicates whether the receiver can attempt to initiate a drag of the given rows for the given event.

- (BOOL)canDragRowsWithIndexes:(NSIndexSet *)rowIndexes
inColumn:(NSInteger)columnIndex withEvent:(NSEvent *)dragEvent

Parameters

rowIndexes

Rows the user is dragging

columnIndex

Column containing the rows the user is dragging.

dragEvent

Mouse-drag event.

Return Value

YES when *rowIndexes* identifies at least one row and all the identified rows are enabled, NO otherwise.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [browser:canDragRowsWithIndexes:inColumn:withEvent:](#) (page 54)

Declared In

NSBrowser.h

cellPrototype

Returns the receiver's prototype `NSCell`.

- (id)cellPrototype

Return Value

The prototype `NSCell`. The prototype `NSCell` instance is copied to display items in the matrices of the browser.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [setCellPrototype:](#) (page 42)

- [setCellClass:](#) (page 41)

Declared In

NSBrowser.h

columnContentWidthForColumnWidth:

Given the column width, returns the content width.

- (CGFloat)columnContentWidthForColumnWidth:(CGFloat)columnWidth

Parameters

columnWidth

The width of the column. This width is the entire scrolling text view.

Return Value

The width of the content for the column. This is the width of the matrix in the column.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [columnWidthForColumnContentWidth:](#) (page 21)

Declared In

NSBrowser.h

columnOfMatrix:

Returns the column number in which the given matrix is located.

- (NSInteger)columnOfMatrix:(NSMatrix *)*matrix*

Parameters

matrix

The matrix for which to return the column number.

Return Value

The index of the column in which the specified matrix appears.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [matrixInColumn:](#) (page 29)

Declared In

NSBrowser.h

columnResizingType

Returns the receiver's column resizing type.

- (NSBrowserColumnType)columnResizingType

Return Value

A constant indicating the column resizing type. Possible return values are described in [NSBrowserColumnType](#) (page 67). The default is `NSBrowserAutoColumnResizing`

Availability

Available in Mac OS X v10.3 and later.

See Also

- [setColumnResizingType:](#) (page 42)

Declared In

NSBrowser.h

columnsAutosaveName

Returns the name used to automatically save the receiver's column configuration.

- (NSString *)columnsAutosaveName

Return Value

The name used to save the column configuration.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [setColumnsAutosaveName:](#) (page 42)

Declared In

NSBrowser.h

columnWidthForColumnContentWidth:

Given the content width, returns the column width.

```
- (CGFloat)columnWidthForColumnContentWidth:(CGFloat)columnContentWidth
```

Parameters

columnContentWidth

The width of the column's content (the width of the the matrix in the column).

Return Value

The width of the column (the width of the entire scrolling text view).

Discussion

For example, to guarantee that 16 pixels of your browser cell are always visible, call:

```
[browser setMinColumnWidth: [browser columnWidthForColumnContentWidth:16]]
```

Availability

Available in Mac OS X v10.3 and later.

See Also

- [columnContentWidthForColumnWidth:](#) (page 19)

Declared In

NSBrowser.h

delegate

Returns the receiver's delegate.

```
- (id)delegate
```

Return Value

The receiver's delegate.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [setDelegate:](#) (page 43)

Declared In

NSBrowser.h

displayAllColumns

Updates the receiver to display all loaded columns.

- (void)displayAllColumns

Availability

Available in Mac OS X v10.0 and later.

See Also

- [addColumn](#) (page 16)
- [validateVisibleColumns](#) (page 53)

Declared In

NSBrowser.h

displayColumn:

Updates the receiver to display the given column.

- (void)displayColumn:(NSInteger)column

Parameters

column

The index of the column to display.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [addColumn](#) (page 16)
- [validateVisibleColumns](#) (page 53)

Declared In

NSBrowser.h

doClick:

Responds to (single) mouse clicks in a column of the receiver.

- (void)doClick:(id)sender

Availability

Available in Mac OS X v10.0 and later.

See Also

- [sendAction](#) (page 38)

Declared In

NSBrowser.h

doDoubleClick:

Responds to double clicks in a column of the receiver.

- (void)doDoubleClick:(id) *sender*

Availability

Available in Mac OS X v10.0 and later.

See Also

- [setDoubleClickAction:](#) (page 43)

Declared In

NSBrowser.h

doubleAction

Returns the receiver's double-click action method.

- (SEL)doubleAction

Return Value

The action method invoked when the user double-clicks on the browser.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [setDoubleClickAction:](#) (page 43)

Declared In

NSBrowser.h

draggingImageForRowsWithIndexes:inColumn:withEvent:offset:

Provides an image to represent dragged rows during a drag operation on the receiver.

- (NSImage *)draggingImageForRowsWithIndexes:(NSIndexSet *) *rowIndexes*
inColumn:(NSInteger) *columnIndex* withEvent:(NSEvent *) *dragEvent*
offset:(NSPointPointer) *dragImageOffset*

Parameters

rowIndexes

Rows the user is dragging.

columnIndex

Column with the rows the user is dragging.

dragEvent

Mouse drag event.

inout_dragImageOffset

Offset for the returned image:

- NSZeroPoint: The image is centered under the pointer.

Return Value

Image representing the visible cells identified by rowIndexes.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [browser:draggingImageForRowsWithIndexes:inColumn:withEvent:offset:](#) (page 55)

Declared In

NSBrowser.h

draggingSourceOperationMaskForLocal:

Indicates the types of dragging operations the source object allows on the dragged image's data.

- (NSDragOperation)draggingSourceOperationMaskForLocal:(BOOL)localDestination

Parameters

localDestination

Indicates the location of the dragging operation's destination object: YES for this application, NO for another application.

Return Value

NSDragOperationEvery when localDestination is YES.

NSDragOperationNone when localDestination is NO.

Discussion

This method overrides NSDraggingSource draggingSourceOperationMaskForLocal:.

drawTitleOfColumn:inRect:

Draws the title for the specified column within the given rectangle.

- (void)drawTitleOfColumn:(NSInteger)column inRect:(NSRect)aRect

Parameters

column

The index of the column for which to draw the title.

aRect

The rectangle within which to draw the title.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [setTitle:ofColumn:](#) (page 49)

- [titleLabelOfColumn:](#) (page 51)

- [titleLabelHeight:](#) (page 52)

Declared In

NSBrowser.h

firstVisibleColumn

Returns the index of the first visible column.

- (NSInteger)firstVisibleColumn

Return Value

The index of the first visible column.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [lastVisibleColumn](#) (page 27)
- [numberOfVisibleColumns](#) (page 30)

Declared In

NSBrowser.h

frameOfColumn:

Returns the rectangle containing the given column.

- (NSRect)frameOfColumn:(NSInteger)column

Parameters

column

The index of the column for which to retrieve the frame.

Return Value

The rectangle containing the specified column.

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSBrowser.h

frameOfInsideOfColumn:

Returns the rectangle containing the specified column, not including borders.

- (NSRect)frameOfInsideOfColumn:(NSInteger)column

Parameters

column

The index of the column for which to retrieve the inside frame.

Return Value

The rectangle containing the column, not including the column borders.

Availability

Available in Mac OS X v10.0 and later.

Declared In
NSBrowser.h

hasHorizontalScroller

Returns a Boolean value indicating whether the browser has a horizontal scroller.

- (BOOL)hasHorizontalScroller

Return Value
YES if the browser uses an NSScroller object to scroll horizontally; otherwise NO.

Availability
Available in Mac OS X v10.0 and later.

See Also
- [setHasHorizontalScroller:](#) (page 44)

Declared In
NSBrowser.h

isLoading

Returns whether column 0 is loaded.

- (BOOL)isLoading

Return Value
YES if column 0 is loaded; otherwise NO.

Availability
Available in Mac OS X v10.0 and later.

See Also
- [loadColumnZero](#) (page 28)
- [reloadColumn:](#) (page 32)

Declared In
NSBrowser.h

isOpaque

Indicates whether the receiver is opaque.

- (BOOL)isOpaque

Return Value
YES when the receiver doesn't have a title and its background color's alpha component is 1.0, NO otherwise.

Discussion
This method overrides `NSView isOpaque`.

isTitled

Returns a Boolean value indicating whether columns display titles.

- (BOOL)isTitled

Return Value

YES if the columns in a browser display titles; otherwise NO.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [setTitle:](#) (page 50)

Declared In

NSBrowser.h

lastColumn

Returns the index of the last column loaded.

- (NSInteger)lastColumn

Return Value

The index of the last loaded column.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [selectedColumn](#) (page 36)

- [setLastColumn:](#) (page 45)

Declared In

NSBrowser.h

lastVisibleColumn

Returns the index of the last visible column.

- (NSInteger)lastVisibleColumn

Return Value

The index of the last visible column.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [firstVisibleColumn](#) (page 25)

- [numberOfVisibleColumns](#) (page 30)

Declared In

NSBrowser.h

loadColumnZero

Loads column 0; unloads previously loaded columns.

- (void)loadColumnZero

Availability

Available in Mac OS X v10.0 and later.

See Also

- [isLoading](#) (page 26)
- [reloadColumn:](#) (page 32)

Declared In

NSBrowser.h

loadedCellAtRow:column:Loads, if necessary, and returns the `NSCell` at the specified row and column location.- (id)loadedCellAtRow:(`NSInteger`)row column:(`NSInteger`)column**Parameters***row*

The row index of the cell to return.

column

The column index of the cell to return.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [selectedCellInColumn:](#) (page 35)

Declared In

NSBrowser.h

matrixClass

Returns the matrix class used in the receiver's columns.

- (Class)matrixClass

Return ValueThe class of `NSMatrix` used in the browser's columns.**Availability**

Available in Mac OS X v10.0 and later.

See Also

- [setMatrixClass:](#) (page 45)

Declared In

NSBrowser.h

matrixInColumn:

Returns the matrix located in the specified column.

- (NSMatrix *)matrixInColumn:(NSInteger)column

Parameters

column

The column index of the matrix to obtain.

Return Value

The matrix located in the column.

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSBrowser.h

maxVisibleColumns

Returns the maximum number of visible columns.

- (NSInteger)maxVisibleColumns

Return Value

The maximum number of visible columns.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [setMaxVisibleColumns:](#) (page 45)

Declared In

NSBrowser.h

minColumnWidth

Returns the minimum column width.

- (CGFloat)minColumnWidth

Return Value

The minimum column width, in pixels.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [setMinColumnWidth:](#) (page 46)

Declared In

NSBrowser.h

namesOfPromisedFilesDroppedAtDestination:

Provides the names of the files that the receiver promises to create at a specified location.

- (NSArray *)namesOfPromisedFilesDroppedAtDestination:(NSURL *)*dropDestination*

Return Value

Result of sending

`browser:namesOfPromisedFilesDroppedAtDestination:forDraggedRowsWithIndexes:inColumn:` to the delegate.

Discussion

Implementation of `NSDraggingSource` `namesOfPromisedFilesDroppedAtDestination:`.

See Also

- [browser:namesOfPromisedFilesDroppedAtDestination:forDraggedRowsWithIndexes:inColumn:](#) (page 56)

numberOfVisibleColumns

Returns the number of columns visible.

- (NSInteger)numberOfVisibleColumns

Return Value

The number of visible columns.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [validateVisibleColumns](#) (page 53)

Declared In

NSBrowser.h

path

Returns a string representing the receiver's current path.

- (NSString *)path

Return Value

The path representing the current selection. The components of this path are separated with the string returned by [pathSeparator](#) (page 31).

Discussion

Invoking this method is equivalent to invoking [pathToColumn:](#) (page 31) for all columns.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [setPath:](#) (page 46)

Declared In

NSBrowser.h

pathSeparator

Returns the path separator.

- (NSString *)pathSeparator

Return Value

The path separator. The default is “/”.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [setPathSeparator:](#) (page 47)

Declared In

NSBrowser.h

pathToColumn:

Returns a string representing the path from the first column up to, but not including, the column at the given index.

- (NSString *)pathToColumn:(NSInteger)column

Parameters

column

The index of the column at which the path stops.

Return Value

The path of the current selection up to, but not including, the specified column. The components of this path are separated with the string returned by [pathSeparator](#) (page 31).

Availability

Available in Mac OS X v10.0 and later.

See Also

- [path](#) (page 30)

- [setPath:](#) (page 46)

Declared In

NSBrowser.h

prefersAllColumnUserResizing

Returns a Boolean value indicating if the browser is set to resize all columns simultaneously rather than resizing a single column at a time.

- (BOOL)prefersAllColumnUserResizing

Return Value

YES if the browser is set to resize all columns simultaneously; otherwise NO. The default is NO.

Discussion

This setting applies only to browsers that allow the user to resize columns (see [NSBrowserUserColumnResizing](#) (page 67)). Holding down the Option key while resizing switches the type of resizing used.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [setPrefersAllColumnUserResizing:](#) (page 47)
- [setColumnResizingType:](#) (page 42)

Declared In

NSBrowser.h

reloadColumn:

Reloads the given column if it exists and sets it to be the last column.

- (void)reloadColumn:(NSInteger)column

Parameters

column

The index of the column to reload.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [isLoading:](#) (page 26)
- [loadColumnZero:](#) (page 28)

Declared In

NSBrowser.h

reusesColumns

Returns a Boolean value indicating whether the browser reuses `NSMatrix` objects after their columns are unloaded.

- (BOOL)reusesColumns

Return Value

YES if `NSMatrix` objects aren't freed when their columns are unloaded; otherwise NO.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [setReusesColumns:](#) (page 48)

Declared In

NSBrowser.h

scrollColumnsLeftBy:

Scrolls columns left by the specified number of columns.

- (void)scrollColumnsLeftBy:(NSInteger)shiftAmount

Parameters

shiftAmount

The number of columns by which to scroll the browser.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [scrollViaScroller:](#) (page 34)

- [updateScroller](#) (page 52)

Declared In

NSBrowser.h

scrollColumnsRightBy:

Scrolls columns right by the specified number of columns.

- (void)scrollColumnsRightBy:(NSInteger)shiftAmount

Parameters

shiftAmount

The number of columns by which to scroll the browser.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [scrollViaScroller:](#) (page 34)

- [updateScroller](#) (page 52)

Declared In

NSBrowser.h

scrollColumnToVisible:

Scrolls to make the specified column visible.

- (void)scrollColumnToVisible:(NSInteger)*column*

Parameters

column

The index of the column to scroll to.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [scrollViaScroller:](#) (page 34)

- [updateScroller](#) (page 52)

Declared In

NSBrowser.h

scrollViaScroller:

Scrolls columns left or right based on an NSScroller.

- (void)scrollViaScroller:(NSScroller *)*sender*

Parameters

sender

The NSScroller object that determines the scrolling of the browser columns.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [updateScroller](#) (page 52)

Declared In

NSBrowser.h

selectAll:

Selects all NSCell objects in the last column of the receiver.

- (void)selectAll:(id)*sender*

Availability

Available in Mac OS X v10.0 and later.

See Also

- [selectedCell](#) (page 35)
- [selectedCells](#) (page 36)
- [selectedColumn](#) (page 36)

Declared In

NSBrowser.h

selectedCell

Returns the last (rightmost and lowest) selected `NSCell`.

- (id)selectedCell

Return Value

The selected cell.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [loadedCellAtRow:column:](#) (page 28)
- [selectedCell](#) (page 35)
- [selectRow:inColumn:](#) (page 37)

Declared In

NSBrowser.h

selectedCellInColumn:

Returns the last (lowest) `NSCell` selected in the given column.

- (id)selectedCellInColumn:(NSInteger)column

Parameters

column

The column for which to return the last selected cell.

Return Value

The last (or lowest) selected cell.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [loadedCellAtRow:column:](#) (page 28)
- [selectedCell](#) (page 35)
- [selectedRowInColumn:](#) (page 36)

Declared In

NSBrowser.h

selectedCells

Returns all cells selected in the rightmost column.

- (NSArray *)selectedCells

Return Value

An array of `NSCell` objects representing the selected cells in the rightmost browser column.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [selectAll:](#) (page 34)
- [selectedCell](#) (page 35)

Declared In

NSBrowser.h

selectedColumn

Returns the index of the last column with a selected item.

- (NSInteger)selectedColumn

Return Value

The index of the last column with a selected item.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [columnOfMatrix:](#) (page 20)
- [selectAll:](#) (page 34)

Declared In

NSBrowser.h

selectedRowInColumn:

Returns the row index of the selected cell in the specified column.

- (NSInteger)selectedRowInColumn:(NSInteger)column

Parameters

column

The column index specifying the column for which to return the selected row.

Return Value

The row index of the selected cell in the specified column.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [loadedCellAtRow:column:](#) (page 28)
- [selectedCell](#) (page 35)
- [selectedCellInColumn:](#) (page 35)

Related Sample Code

NewsReader

Declared In

NSBrowser.h

selectedRowIndexesInColumn:

Provides the indexes of the selected rows in a given column of the receiver.

```
- (NSIndexSet *)selectedRowIndexesInColumn:(NSInteger)columnIndex
```

Parameters*columnIndex*

Column whose selected rows are provided.

Return ValueRows selected in column *columnIndex*.**Availability**

Available in Mac OS X v10.5 and later.

See Also

- [selectRowIndexes:inColumn:](#) (page 38)

Declared In

NSBrowser.h

selectRow:inColumn:

Selects the cell at the specified row and column index.

```
- (void)selectRow:(NSInteger)row inColumn:(NSInteger)column
```

Parameters*row*

The row index of the cell to select.

column

The column index of the cell to select.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [loadedCellAtRow:column:](#) (page 28)

Declared In

NSBrowser.h

selectRowIndexes:inColumn:

Specifies the selected rows in a given column of the receiver.

```
- (void)selectRowIndexes:(NSIndexSet *)rowIndexes inColumn:(NSInteger)columnIndex
```

Parameters

rowIndexes

Rows to be selected in column *columnIndex*.

columnIndex

Column in which to select rows *rowIndexes*.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [selectedRowIndexesInColumn:](#) (page 37)

Declared In

NSBrowser.h

sendAction

Sends the action message to the target.

```
- (BOOL)sendAction
```

Return Value

YES if successful, NO if no target for the message could be found.

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSBrowser.h

sendsActionOnArrowKeys

Returns a Boolean value indicating whether pressing an arrow key causes an action message to be sent.

```
- (BOOL)sendsActionOnArrowKeys
```

Return Value

NO if pressing an arrow key only scrolls the receiver, YES if it also sends the action message specified by `setAction:`.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [acceptsArrowKeys](#) (page 16)

- [setSendsActionOnArrowKeys:](#) (page 48)

Declared In

NSBrowser.h

separatesColumns

Returns a Boolean value indicating whether columns are separated by bezeled borders.

- (BOOL)separatesColumns

Return Value

YES if the browser's columns are separated by bezeled borders; otherwise NO.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [setSeparatesColumns:](#) (page 48)

Declared In

NSBrowser.h

setAcceptsArrowKeys:

Specifies whether the browser allows navigation using the arrow keys.

- (void)setAcceptsArrowKeys:(BOOL)flag

Parameters

flag

YES to enable the use of the arrow keys for navigating within and between browsers; otherwise NO.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [acceptsArrowKeys](#) (page 16)

- [sendsActionOnArrowKeys](#) (page 38)

Declared In

NSBrowser.h

setAllowsBranchSelection:

Sets whether the user can select branch items.

- (void)setAllowsBranchSelection:(BOOL)flag

Parameters

flag

YES if the user can select branch items when multiple selection is enabled; otherwise NO.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [allowsBranchSelection](#) (page 16)

Declared In

NSBrowser.h

setAllowsEmptySelection:

Sets whether there can be nothing selected.

- (void)setAllowsEmptySelection:(BOOL)flag

Parameters

flag

YES if the browser allows an empty selection; otherwise NO.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [allowsEmptySelection](#) (page 17)

Declared In

NSBrowser.h

setAllowsMultipleSelection:

Sets whether the user can select multiple items.

- (void)setAllowsMultipleSelection:(BOOL)flag

Parameters

flag

YES if the user can select multiple items at once; otherwise NO.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [allowsMultipleSelection](#) (page 17)

Declared In

NSBrowser.h

setAllowsTypeSelect:

Specifies whether the receiver allows keystroke-based selection.

- (void)setAllowsTypeSelect:(BOOL)allowsTypeSelection

Parameters*allowsTypeSelection*

YES to allow type selection, NO to disallow it.

Availability

Available in Mac OS X v10.5 and later.

See Also- [allowsTypeSelect](#) (page 17)**Declared In**

NSBrowser.h

setBackground-color:

Specifies the receiver's background color.

- (void)setBackgroundColor:(NSColor *)*backgroundColor***Parameters***backgroundColor*

[NSColor clearColor] specifies a transparent background.

Availability

Available in Mac OS X v10.5 and later.

See Also- [backgroundColor](#) (page 18)**Declared In**

NSBrowser.h

setCellClass:

Sets the class of NSCell used by the matrices in the columns of the receiver.

- (void)setCellClass:(Class)*factoryId***Parameters***factoryId*The class of NSCell used by the matrices in the columns of the browser. This method creates an instance of the class and calls [setCellPrototype:](#) (page 42).**Availability**

Available in Mac OS X v10.0 and later.

See Also+ [cellClass](#) (page 15)- [cellPrototype](#) (page 19)**Declared In**

NSBrowser.h

setCellPrototype:

Sets the `NSCell` instance copied to display items in the matrices in the columns of the receiver.

```
- (void)setCellPrototype:(NSCell *)aCell
```

Parameters

aCell

The prototype `NSCell` instance.

Availability

Available in Mac OS X v10.0 and later.

See Also

+ [cellClass](#) (page 15)

- [cellPrototype](#) (page 19)

- [setCellClass:](#) (page 41)

Declared In

`NSBrowser.h`

setColumnResizingType:

Sets the receiver's column resizing type.

```
- (void)setColumnResizingType:(NSBrowserColumnResizingType)columnResizingType
```

Parameters

columnResizingType

A constant specifying the column resizing type. Possible values are described in [NSBrowserColumnResizingType](#) (page 67). The default is `NSBrowserAutoColumnResizing`. This setting is persistent.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [columnResizingType](#) (page 20)

Declared In

`NSBrowser.h`

setColumnsAutosaveName:

Sets the name used to automatically save the receiver's column configuration.

```
- (void)setColumnsAutosaveName:(NSString *)name
```

Parameters

name

The name used to save the column configuration. If *name* is different from the current name, this method also reads in any column configuration data previously saved under *name* and applies the values to the browser.

Discussion

Column configuration is defined as an array of column content widths. One width is saved for each level the user has reached. That is, the browser saves column width based on depth, not on unique paths. To do more complex column persistence, you should register for [NSBrowserColumnConfigurationDidChangeNotification](#) (page 68) and handle persistence yourself. This setting is persistent.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [columnsAutosaveName](#) (page 20)

Declared In

NSBrowser.h

setDelegate:

Sets the receiver's delegate.

- (void)setDelegate:(id)*anObject*

Parameters

anObject

The object to set at the receiver's delegate.

Discussion

Raises `NSBrowserIllegalDelegateException` if the delegate specified by *anObject* doesn't respond to `browser:willDisplayCell:atRow:column:` (page 64) and either of the methods `browser:numberOfRowsInColumn:` (page 58) or `browser:createRowsForColumn:inMatrix:` (page 55).

Availability

Available in Mac OS X v10.0 and later.

See Also

- [delegate](#) (page 21)

Declared In

NSBrowser.h

setDoubleClickAction:

Sets the receiver's double-click action.

- (void)setDoubleClickAction:(SEL)*aSelector*

Parameters

aSelector

The action method to invoke when the receiver is double-clicked.

Discussion

For the method to have any effect, the receiver's action and target must be set to the class in which the selector is declared. See *Action Messages* for additional information on action messages.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [doubleAction](#) (page 23)
- [sendAction](#) (page 38)

Declared In

NSBrowser.h

setDraggingSourceOperationMask:forLocal:

Specifies the drag-operation mask for dragging operations with local or external destinations.

```
- (void)setDraggingSourceOperationMask:(NSDragOperation)dragOperationMask
    forLocal:(BOOL)localDestination
```

Parameters

dragOperationMask

Dragging operation mask to use for either local or external drag operations, as specified by *localDestination*.

localDestination

Indicates the location of the dragging operation's destination object:

YES for this application, NO for another application.

Discussion

Important: Do not override this method.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [draggingSourceOperationMaskForLocal:](#) (page 24)

Declared In

NSBrowser.h

setHasHorizontalScroller:

Sets whether an `NSScroller` is used to scroll horizontally.

```
- (void)setHasHorizontalScroller:(BOOL)flag
```

Parameters

flag

YES if the browser uses an `NSScroller` object to scroll horizontally; otherwise NO.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [hasHorizontalScroller](#) (page 26)

Declared In

NSBrowser.h

setLastColumn:

Sets the last column.

- (void)setLastColumn:(NSInteger)*column*

Parameters

column

The index of the last column.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [lastColumn](#) (page 27)

- [lastVisibleColumn](#) (page 27)

Declared In

NSBrowser.h

setMatrixClass:

Sets the matrix class used in the receiver's columns.

- (void)setMatrixClass:(Class)*factoryId*

Parameters

factoryId

The matrix class (NSMatrix or an NSMatrix subclass) used in the browser's columns.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [matrixClass](#) (page 28)

Declared In

NSBrowser.h

setMaxVisibleColumns:

Sets the maximum number of columns displayed.

- (void)setMaxVisibleColumns:(NSInteger)*columnCount*

Parameters*columnCount*

The maximum number of visible columns.

Availability

Available in Mac OS X v10.0 and later.

See Also- [maxVisibleColumns](#) (page 29)**Declared In**

NSBrowser.h

setMinColumnWidth:

Sets the minimum column width.

- (void)setMinColumnWidth:(CGFloat)*columnWidth***Parameters***columnWidth*

The minimum column width, specified in pixels.

Availability

Available in Mac OS X v10.0 and later.

See Also- [minColumnWidth](#) (page 29)**Declared In**

NSBrowser.h

setPath:

Sets the path displayed by the receiver.

- (BOOL)setPath:(NSString *)*path***Parameters***path*The path to display. If *path* is prefixed by the path separator, the path is absolute, containing the full path from the receiver's first column. Otherwise, the path is relative, extending the receiver's current path starting at the last column.**Return Value**

YES if the given path is valid; otherwise, NO.

DiscussionWhile parsing *path*, the receiver compares each component with the entries in the current column. If an exact match is found, the matching entry is selected, and the next component is compared to the next column's entries. If no match is found for a component, the method exits and returns NO; the final path is set to the valid portion of *path*. If each component of *path* specifies a valid branch or leaf in the receiver's hierarchy, the method returns YES.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [path](#) (page 30)
- [pathToColumn:](#) (page 31)
- [pathSeparator](#) (page 31)
- [setPathSeparator:](#) (page 47)

Declared In

NSBrowser.h

setPathSeparator:

Sets the path separator.

```
- (void)setPathSeparator:(NSString *)newString
```

Parameters

newString

The new path separator.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [pathSeparator](#) (page 31)

Declared In

NSBrowser.h

setPrefersAllColumnUserResizing:

Specifies whether the browser resizes all columns simultaneously rather than resizing a single column at a time.

```
- (void)setPrefersAllColumnUserResizing:(BOOL)prefersAllColumnResizing
```

Parameters

prefersAllColumnResizing

YES to cause the browser to resize all columns simultaneously; the default is single column resizing (NO). This setting applies only to browsers that allow the user to resize columns (see [NSBrowserUserColumnResizing](#) (page 67)). Holding down the Option key while resizing switches the type of resizing used. This setting is persistent.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [prefersAllColumnUserResizing](#) (page 32)
- [setColumnResizingType:](#) (page 42)

Declared In

NSBrowser.h

setReusesColumns:Specifies whether `NSMatrix` objects are reused.

- (void)setReusesColumns:(BOOL)flag

Parameters*flag*YES to prevent `NSMatrix` objects from being freed when their columns are unloaded, so they can be reused; otherwise NO.**Availability**

Available in Mac OS X v10.0 and later.

See Also- [reusesColumns](#) (page 33)**Declared In**

NSBrowser.h

setSendsActionOnArrowKeys:

Sets whether pressing an arrow key causes the action message to be sent.

- (void)setSendsActionOnArrowKeys:(BOOL)flag

Parameters*flag*YES if pressing an arrow key should send the action message specified by `setAction:` in addition to scrolling the browser; NO if it should only scroll the browser.**Availability**

Available in Mac OS X v10.0 and later.

See Also- [sendsActionOnArrowKeys](#) (page 38)**Declared In**

NSBrowser.h

setSeparatesColumns:

Sets whether to separate columns with beveled borders.

- (void)setSeparatesColumns:(BOOL)flag

Parameters*flag*

YES if the browser's columns should be separated by bezeled borders; otherwise NO. This value is ignored if [isTitled](#) (page 27) does not return NO

Availability

Available in Mac OS X v10.0 and later.

See Also

- [separatesColumns](#) (page 39)

Declared In

NSBrowser.h

setTakesTitleFromPreviousColumn:

Sets whether the title of a column is set to the string value of the selected cell in the previous column.

- (void)setTakesTitleFromPreviousColumn:(BOOL)*flag*

Parameters*flag*

YES if the title of a column should be set to the string value of the selected `NSCell` in the previous column; otherwise NO.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [takesTitleFromPreviousColumn](#) (page 51)

Declared In

NSBrowser.h

setTitle:ofColumn:

Sets the title of the given column.

- (void)setTitle:(NSString *)*aString* ofColumn:(NSInteger)*column*

Parameters*aString*

The title of the column.

column

The index of the column whose title should be set.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [drawTitleOfColumn:inRect:](#) (page 24)

- [titleOfColumn:](#) (page 52)

Declared In

NSBrowser.h

setTitled:

Sets whether columns display titles.

- (void)setTitled:(BOOL)flag

Parameters*flag*

YES if the columns in a browser display titles; otherwise NO.

Availability

Available in Mac OS X v10.0 and later.

See Also- [isTitled](#) (page 27)**Declared In**

NSBrowser.h

setWidth:ofColumn:

Sets the width of the specified column.

- (void)setWidth:(CGFloat)columnWidth ofColumn:(NSInteger)columnIndex

Parameters*columnWidth*

The new width of the specified column.

columnIndex

The index of the column for which to set the width.

Discussion

This method can be used to set the initial width of browser columns unless the column sizing is automatic; `setWidth:ofColumn:` does nothing if `columnResizingType` (page 20) is `NSBrowserAutoColumnResizing`. To set the default width for new columns (that don't otherwise have initial widths from defaults or via the delegate), use a `columnIndex` of `-1`. A value set for `columnIndex` of `-1` is persistent. An `NSBrowserColumnConfigurationDidChangeNotification` (page 68) notification is posted (not immediately), if necessary, so that the receiver can autosave the new column configuration.

Availability

Available in Mac OS X v10.3 and later.

See Also- [widthOfColumn:](#) (page 53)- [browser:shouldSizeColumn:forUserResize:toWidth:](#) (page 60)**Declared In**

NSBrowser.h

takesTitleFromPreviousColumn

Returns a Boolean value indicating whether a column takes its title from the selected cell in the previous column.

- (BOOL)takesTitleFromPreviousColumn

Return Value

YES if the title of a column is set to the string value of the selected `NSCell` in the previous column; otherwise NO.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [setTakesTitleFromPreviousColumn:](#) (page 49)

Declared In

`NSBrowser.h`

tile

Adjusts the various subviews of the receiver—scrollers, columns, titles, and so on—without redrawing.

- (void)tile

Discussion

Your code shouldn't send this message. It's invoked any time the appearance of the receiver changes.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`NSBrowser.h`

titleFrameOfColumn:

Returns the bounds of the title frame for the specified column.

- (NSRect)titleFrameOfColumn:(NSInteger)column

Parameters

column

The index of the column for which to return the title frame.

Return Value

The rectangle specifying the bounds of the column's title frame.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [drawTitleOfColumn:inRect:](#) (page 24)

Declared In

NSBrowser.h

titleHeight

Returns the height of column titles.

- (CGFloat)titleHeight

Return Value

The height of the column titles for the browser.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [drawTitleOfColumn:inRect:](#) (page 24)

Declared In

NSBrowser.h

titleOfColumn:

Returns the title displayed for the given column.

- (NSString *)titleOfColumn:(NSInteger)column

Parameters

column

The index of the column for which to get the title.

Return Value

The title of the specified column.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [setTitle:ofColumn:](#) (page 49)

Declared In

NSBrowser.h

updateScroller

Updates the horizontal scroller to reflect column positions.

- (void)updateScroller

Availability

Available in Mac OS X v10.0 and later.

See Also

- [scrollViaScroller:](#) (page 34)

Declared In

NSBrowser.h

validateVisibleColumns

Invokes the delegate method [browser:isColumnValid:](#) (page 56) for visible columns.

- (void)validateVisibleColumns

Availability

Available in Mac OS X v10.0 and later.

See Also

- [numberOfVisibleColumns](#) (page 30)

Declared In

NSBrowser.h

widthOfColumn:

Returns the width of the specified column.

- (CGFloat)widthOfColumn:(NSInteger)column

Parameters

column

The index of the column for which to retrieve the width.

Return Value

The width of the column.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [setWidth:ofColumn:](#) (page 50)

Declared In

NSBrowser.h

Delegate Methods

browser:acceptDrop:atRow:column:dropOperation:

Sent to the delegate during a dragging session to determine whether to accept the drop. Required for a browser to be a drag destination.

```
- (BOOL)browser:(NSBrowser *)sender acceptDrop:(id <NSDraggingInfo>)dragInfo
    atRow:(NSInteger)dropReferenceRowIndex column:(NSInteger)dropColumnIndex
    dropOperation:(NSBrowserDropOperation)dropRowRelativeLocation
```

Parameters*sender*

Browser querying its delegate.

dragInfo

Drag session information.

dropReferenceRowIndex

The drop row.

dropColumnIndex

The drop column.

*dropRowRelativeLocation*Drop location relative to *dropRowIndex*.**Return Value**

YES to accept the drop, NO to decline it.

DiscussionSent after [browser:validateDrop:proposedRow:column:dropOperation:](#) (page 63) allows the drop.

The delegate should incorporate the pasteboard data from the dragging session (*dragInfo.draggingPasteboard*)

Availability

Available in Mac OS X v10.5 and later.

Declared In

NSBrowser.h

browser:canDragRowsWithIndexes:inColumn:withEvent:

Sent to the delegate to determine whether the browser can attempt to initiate a drag of the given rows for the given event.

```
- (BOOL)browser:(NSBrowser *)sender canDragRowsWithIndexes:(NSIndexSet *)rowIndexes
    inColumn:(NSInteger)columnIndex withEvent:(NSEvent *)dragEvent
```

Parameters*sender*

Browser querying its delegate.

rowIndexes

Rows the user is dragging

columnIndex

Column containing the rows the user is dragging.

dragEvent

Mouse-drag event.

Return Value

YES to allow the drag operation, NO to disallow it.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [canDragRowsWithIndexes:inColumn:withEvent:](#) (page 18)

Declared In

NSBrowser.h

browser:createRowsForColumn:inMatrix:

Creates a row in the given matrix for each row of data to be displayed in the specified column of the browser.

```
- (void)browser:(NSBrowser *)sender createRowsForColumn:(NSInteger)column
    inMatrix:(NSMatrix *)matrix
```

Parameters

sender

The browser.

column

The index of the column for which to create the rows.

matrix

The `NSMatrix` object in which to create the rows.

Discussion

Either this method or [browser:numberOfRowsInColumn:](#) (page 58) must be implemented, but not both (or an `NSBrowserIllegalDelegateException` will be raised).

Availability

Available in Mac OS X v10.0 and later.

See Also

- [browser:willDisplayCell:atRow:column:](#) (page 64)

Declared In

NSBrowser.h

browser:draggingImageForRowsWithIndexes:inColumn:withEvent:offset:

Sent to the delegate to obtain an image to represent dragged rows during a drag operation on a browser.

```
- (NSImage *)browser:(NSBrowser *)sender draggingImageForRowsWithIndexes:(NSIndexSet
    *)rowIndexes inColumn:(NSInteger)columnIndex withEvent:(NSEvent *)dragEvent
    offset:(NSPointPointer)inout_dragImageOffset
```

Parameters

sender

Browser querying its delegate.

rowIndexes

Rows the user is dragging.

columnIndex

Column with the rows the user is dragging.

dragEvent

Mouse drag event.

inout_dragImageOffset

Offset for the returned image:

- `NSZeroPoint`: Centers the image under the pointer.

Return ValueImage representing the visible rows identified by *rowIndexes*.**Availability**

Available in Mac OS X v10.5 and later.

See Also- [draggingImageForRowsWithIndexes:inColumn:withEvent:offset:](#) (page 23)**Declared In**

NSBrowser.h

browser:isColumnValid:

Returns whether the contents of the specified column are valid.

- (BOOL)browser:(NSBrowser *)sender isColumnValid:(NSInteger)column

Parameters*sender*

The browser for which to validate the contents.

column

The index of the column to validate.

Return ValueYES if the column's contents are valid; otherwise NO. If NO is returned, *sender* reloads the column.**Discussion**This method is invoked in response to [validateVisibleColumns](#) (page 53) being sent to *sender*.**Availability**

Available in Mac OS X v10.0 and later.

Declared In

NSBrowser.h

browser:namesOfPromisedFilesDroppedAtDestination:forDraggedRowsWithIndexes:inColumn:

Implements file-promise drag operations.


```
- (NSArray *)browser:(NSBrowser *)sender
  namesOfPromisedFilesDroppedAtDestination:(NSURL *)dropURL
  forDraggedRowsWithIndexes:(NSIndexSet *)rowIndexes
  inColumn:(NSInteger)columnIndex
```

Parameters*sender*

Browser querying its delegate.

dropURL

Drop filesystem location.

rowIndexes

Rows the user is dropping.

columnIndex

Column with the rows the user is dropping.

Return Value

Filenames (not pathnames) for the actual files represented by the rows the user is dropping.

Discussion

Note: File-promise drag operation support requires adding `NSFilesPromisePboardType` to the pasteboard in `browser:writeRowsWithIndexes:inColumn:toPasteboard:` (page 65)-`browser:writeRowWithIndexes:inColumn:toPasteboard:`

Availability

Available in Mac OS X v10.5 and later.

See Also- `namesOfPromisedFilesDroppedAtDestination:` (page 30)**Declared In**

NSBrowser.h

browser:nextTypeSelectMatchFromRow:toRow:inColumn:forString:

Sent to the delegate to customize a browser's keyboard-based selection (type select) behavior.

```
- (NSInteger)browser:(NSBrowser *)sender
  nextTypeSelectMatchFromRow:(NSInteger)startRowIndex toRow:(NSInteger)endRowIndex
  inColumn:(NSInteger)columnIndex forString:(NSString *)searchString
```

Parameters*sender*

Browser querying its delegate.

startRowIndex

Beginning of row-set to search.

*endRowIndex*End of row-set to search. Can be less than *startRowIndex* when the search wraps to the beginning.*columnIndex*

Column with the rows being searched.

searchString

Keyboard-based selection string. It's nil when no keyboard-based selection has begun.

Return Value

Index—between `startRowIndex` and `endRowIndex - 1`, inclusive—of the first row that matches *searchString*.

-1 indicates that there's no match.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [browser:shouldTypeSelectForEvent:withCurrentSearchString:](#) (page 61)
- [browser:typeSelectStringForRow:inColumn:](#) (page 62)

Declared In

NSBrowser.h

browser:numberOfRowsInColumn:

Returns the number of rows of data in the specified column.

```
- (NSInteger)browser:(NSBrowser *)sender numberOfRowsInColumn:(NSInteger)column
```

Parameters

sender

The browser.

column

The index of the column for which to return the number of rows.

Return Value

The number of rows of data in the specified column.

Discussion

Either this method or [browser:createRowsForColumn:inMatrix:](#) (page 55) must be implemented, but not both.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [browser:willDisplayCell:atRow:column:](#) (page 64)

Declared In

NSBrowser.h

browser:selectCellWithString:inColumn:

Asks the delegate to select the `NSCell` with the given title in the specified column.

```
- (BOOL)browser:(NSBrowser *)sender selectCellWithString:(NSString *)title
inColumn:(NSInteger)column
```

Parameters*sender*

The browser.

title

The title of the cell to select.

column

The index of the column containing the cell to select.

Return Value

YES if the NSCell was successfully selected; otherwise, NO.

Discussion

It is the delegate's responsibility to select the cell, rather than the browser. Invoked in response to [setPath:](#) (page 46) being received by *sender*.

Availability

Available in Mac OS X v10.0 and later.

See Also- [selectedCellInColumn:](#) (page 35)**Declared In**

NSBrowser.h

browser:selectRow:inColumn:

Asks the delegate to select the NSCell at the specified row and column location.

```
- (BOOL)browser:(NSBrowser *)sender selectRow:(NSInteger)row
      inColumn:(NSInteger)column
```

Parameters*sender*

The browser.

row

The index of the row containing the cell to select.

column

The index of the column containing the cell to select.

Return Value

NO if the NSCell was not selected; otherwise YES.

Discussion

It is the delegate's responsibility to select the cell, rather than the browser. Invoked in response to [selectRow:inColumn:](#) (page 37) being received by *sender*.

Availability

Available in Mac OS X v10.0 and later.

See Also- [selectedRowInColumn:](#) (page 36)- [selectRow:inColumn:](#) (page 37)

Declared In

NSBrowser.h

browser:shouldShowCellExpansionForRow:column:

Determines whether an expansion tooltip appears for a cell at the given row in the given column.

```
- (BOOL)browser:(NSBrowser *)sender shouldShowCellExpansionForRow:(NSInteger)rowIndex
column:(NSInteger)columnIndex
```

Parameters*sender*

Browser querying its delegate.

rowIndex

Row requesting an expansion tooltip.

columnIndex

Column with the requesting row.

Return Value

YES to allow the cell expansion tooltip, NO to disallow it.

Availability

Available in Mac OS X v10.5 and later.

Declared In

NSBrowser.h

browser:shouldSizeColumn:forUserResize:toWidth:

Used for determining a column's initial size.

```
- (CGFloat)browser:(NSBrowser *)browser shouldSizeColumn:(NSInteger)columnIndex
forUserResize:(BOOL)forUserResize toWidth:(CGFloat)suggestedWidth
```

Parameters*browser*

The browser.

columnIndex

The index of the column to size.

forUserResize

As currently implemented, this method is always called with *forUserResize* set to NO.

suggestedWidth

The suggested width for the column.

Return Value

The delegate's desired initial width for a newly added column. If you want to accept the suggested width, return *suggestedWidth*. If you return 0 or a size too small to display the resize handle and a portion of the column, the actual size used will be larger than you requested.

Discussion

Implementation is optional and applies only to browsers with `resize type` `NSBrowserNoColumnResizing` or `NSBrowserUserColumnResizing` (see [NSBrowserColumnResizingType](#) (page 67)).

Availability

Available in Mac OS X v10.3 and later.

See Also

- [setWidth:ofColumn:](#) (page 50)

Declared In

`NSBrowser.h`

browser:shouldTypeSelectForEvent:withCurrentSearchString:

Sent to the delegate to determine whether keyboard-based selection (type select) for a given event and search string should proceed.

```
- (BOOL)browser:(NSBrowser *)sender shouldTypeSelectForEvent:(NSEvent *)keyEvent
withCurrentSearchString:(NSString *)searchString
```

Parameters

sender

Browser querying its delegate.

keyEvent

Keyboard event being processed.

searchString

Keyboard-based selection string. It's `nil` when no keyboard-based selection has begun.

Return Value

YES to allow the selection, NO to disallow it.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [allowsTypeSelect](#) (page 17)

Declared In

`NSBrowser.h`

browser:sizeToFitWidthOfColumn:

Returns the ideal width for a column.

```
- (CGFloat)browser:(NSBrowser *)browser sizeToFitWidthOfColumn:(NSInteger)columnIndex
```

Parameters

browser

The browser.

columnIndex

The index of the column to size. If *columnIndex* is `-1`, the result is used for a “right-size-all” operation.

Return Value

The ideal width of the column. This method is used when performing a “right-size” operation; that is, when sizing a column to the smallest width that contains all the content without clipping or truncating.

If *columnIndex* is `-1`, you should return a size that can be uniformly applied to all columns (that is, every column will be set to this size). It is assumed that the implementation may be expensive, so it will be called only when necessary.

Discussion

Implementation is optional and is for browsers with `resizeType` `NSBrowserUserColumnResizing` only.

Availability

Available in Mac OS X v10.3 and later.

Declared In

`NSBrowser.h`

browser:titleOfColumn:

Asks the delegate for the title to display above the specified column.

```
-(NSString *)browser:(NSBrowser *)sender titleOfColumn:(NSInteger)column
```

Parameters

sender

The browser.

column

The index of the column for which to return the title.

Return Value

The title of the specified column.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [setTitle:ofColumn:](#) (page 49)
- [titleOfColumn:](#) (page 52)

Declared In

`NSBrowser.h`

browser:typeSelectStringForRow:inColumn:

Sent to the delegate to get the keyboard-based selection (type select) string for a given row and column.

```
-(NSString *)browser:(NSBrowser *)sender typeSelectStringForRow:(NSInteger)rowIndex
inColumn:(NSInteger)columnIndex
```

Parameters

sender

Browser querying its delegate.

rowIndex

Row being inquired about.

columnIndex

Column with the row being inquired about.

Return Value

Keyboard-based selection string for the identified row and column.

Returning the empty string or `nil` (for example, when the cell does not contain text) specifies that the [*columnIndex*, *rowIndex*] cell has no text to search.

Discussion

If the delegate does not implement this method, all cells with text are searched, and the browser determines the keyboard-based selection text by sending `stringValue` to the cell specified by *columnIndex* and *rowIndex*.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [browser:shouldTypeSelectForEvent:withCurrentSearchString:](#) (page 61)

Declared In

`NSBrowser.h`

browser:validateDrop:proposedRow:column:dropOperation:

Sent to the delegate during a dragging session to determine whether to accept a drop, and to obtain the drop location. Required for a browser to be a drag destination.

```
- (NSDragOperation)browser:(NSBrowser *)sender validateDrop:(id
    <NSDraggingInfo>)dragInfo proposedRow:(NSInteger *)inout_dropReferenceRowIndex
    column:(NSInteger *)inout_dropColumnIndex dropOperation:(NSBrowserDropOperation
    *) inout_dropRowRelativeLocation
```

Parameters

sender

Browser querying its delegate.

dragInfo

Drag session information.

inout_dropReferenceRowIndex

Together with *inout_dropRowRelativeLocation* specifies the drop location.

On input, the proposed drop reference row.

On output, the drop reference row.

inout_dropColumnIndex

Column on which the dragged rows are being dropped.

On input, the proposed drop column.

On output, the drop column.

inout_dropRowRelativeLocation

Drop location, within the rows identified in *rowIndexes*, of the drag operation.

On input, the proposed drop location.

On output, the drop location.

Return Value

Drag operation the data source is to perform. Must not be `NSDragOperationNone` for sender to accept the drop.

Discussion

The browser proposes a drop column, row, and row-relative location for the drop based on the pointer position, as shown in this table:

Drop relative location	Description
<code>NSBrowserDropOn</code>	Dragging location (<i>dragInfo.draggingLocation</i>) is closer to the middle of <i>inout_dropReferenceRowIndex</i> than to either of its vertical sides.
<code>NSBroserDropAbove</code>	Dragging location is between two rows. Indicates a drop location above <i>inout_dropReferenceRowIndex</i> and below <i>inout_dropReferenceRowIndex - 1</i> .

These are a few examples of how to specify a drop location:

	Reference row index	Row-relative location
On row 2	2	<code>NSBrowserDropOn</code>
Between rows 2 and 3	3	<code>NSBrowserDropAbove</code>
Below the last row	[sender numberOfRows]	<code>NSBrowserDropAbove</code>
All rows	-1	<code>NSBrowserDropOn</code>

Availability

Available in Mac OS X v10.5 and later.

See Also

- `registerForDraggedTypes:`

Declared In

`NSBrowser.h`

browser:willDisplayCell:atRow:column:

Gives the delegate the opportunity to modify the specified cell at the given row and column position before it's displayed by the `NSBrowser`.

```
- (void)browser:(NSBrowser *)sender willDisplayCell:(id)cell atRow:(NSInteger)row
column:(NSInteger)column
```


Parameters*sender*

The browser.

cell

The cell to be displayed.

row

The row index of the cell to be displayed.

column

The column index of the cell to be displayed.

Discussion

The delegate should set any state necessary for the correct display of the cell.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [browser:createRowsForColumn:inMatrix:](#) (page 55)
- [browser:numberOfRowsInColumn:](#) (page 58)

Declared In

NSBrowser.h

browser:writeRowsWithIndexes:inColumn:toPasteboard:

Determines whether a drag operation can proceed. Required for a browser to be a drag source.

```
- (BOOL)browser:(NSBrowser *)sender writeRowsWithIndexes:(NSIndexSet *)rowIndexes
inColumn:(NSInteger)columnIndex toPasteboard:(NSPasteboard *)pasteboard
```

Parameters*sender*

Browser querying its delegate.

rowIndexes

Rows the user is dragging.

columnIndex

Column with the dragged rows.

pasteboard

Content from the dragged rows.

Return Value

YES to allow the dragging operation to proceed (see discussion for further details), NO to disallow it.

Discussion

Called after a drag operation has been allowed to start

(browser:canDragRowsWithIndexes:inColumn:withEvent: (page 54) returns YES) but before it actually begins.

Availability

Available in Mac OS X v10.5 and later.

Declared In

NSBrowser.h

browserColumnConfigurationDidChange:

Used by clients to implement their own column width persistence.

- (void)browserColumnConfigurationDidChange:(NSNotification *)*notification***Parameters***notification*

The notification of the configuration change.

Discussion

Implementation is optional, and used for browsers with `resize type` `NSBrowserUserColumnResizing` only. It is called when the method `setWidth:ofColumn:` (page 50) is used to change the width of any browser columns or when the user resizes any columns. If the user resizes more than one column, a single notification is posted when the user is finished resizing.

Availability

Available in Mac OS X v10.3 and later.

See Also[NSBrowserColumnConfigurationDidChangeNotification](#) (page 68)- `setWidth:ofColumn:` (page 50)**Declared In**

NSBrowser.h

browserDidScroll:Notifies the delegate when the `NSBrowser` has scrolled.- (void)browserDidScroll:(NSBrowser *)*sender***Parameters***sender*

The browser sending the message.

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSBrowser.h

browserWillScroll:Notifies the delegate when the `NSBrowser` will scroll.- (void)browserWillScroll:(NSBrowser *)*sender*

Parameters*sender*

The browser sending the message.

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSBrowser.h

Constants

NSBrowserColumnResizingType

Describe types of browser column resizing.

```
typedef enum _NSBrowserColumnResizingType {
    NSBrowserNoColumnResizing = 0,
    NSBrowserAutoColumnResizing = 1,
    NSBrowserUserColumnResizing = 2
} NSBrowserColumnResizingType;
```

Constants

NSBrowserNoColumnResizing

Neither NSBrowser nor the user can change the column width. The developer must explicitly set all column widths.

Available in Mac OS X v10.3 and later.

Declared in NSBrowser.h.

NSBrowserAutoColumnResizing

All columns have the same width, calculated using a combination of the minimum column width and maximum number of visible columns settings. The column width changes as the window size changes. The user cannot resize columns.

Available in Mac OS X v10.3 and later.

Declared in NSBrowser.h.

NSBrowserUserColumnResizing

The developer chooses the initial column widths, but users can resize all columns simultaneously or each column individually.

Available in Mac OS X v10.3 and later.

Declared in NSBrowser.h.

DiscussionThese constants are used by the [setColumnResizingType:](#) (page 42) and [columnResizingType](#) (page 20) methods.**Availability**

Available in Mac OS X v10.3 and later.

Declared In

NSBrowser.h

NSBrowserDropOperation

This type is used to specifying the drop type of a drag-and-drop operation. See `browser:validateDrop:proposedRow:column:dropOperation` for more information.

```
enum {
    NSBrowserDropOn,
    NSBrowserDropAbove
};
typedef NSUInteger NSBrowserDropOperation;
```

Availability

Available in Mac OS X v10.5 and later.

Declared In

`NSBrowser.h`

Notifications

NSBrowserColumnConfigurationDidChangeNotification

Notifies the delegate when the width of a browser column has changed. The notification object is the browser whose column sizes need to be made persistent. This notification does not contain a `userInfo` dictionary. If the user resizes more than one column, a single notification is posted when the user is finished resizing.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [browserColumnConfigurationDidChange:](#) (page 66)

Declared In

`NSBrowser.h`

Document Revision History

This table describes the changes to *NSBrowser Class Reference*.

Date	Notes
2007-04-05	Updated for Mac OS X v10.5.
2006-05-23	First publication of this content as a separate document.

REVISION HISTORY

Document Revision History

Index

A

acceptsArrowKeys **instance method** 16
addColumn **instance method** 16
allowsBranchSelection **instance method** 16
allowsEmptySelection **instance method** 17
allowsMultipleSelection **instance method** 17
allowsTypeSelect **instance method** 17

B

backgroundColor **instance method** 18
browser:acceptDrop:atRow:column:dropOperation:
<NSObject> **delegate method** 53
browser:canDragRowsWithIndexes:inColumn:withEvent:
<NSObject> **delegate method** 54
browser:createRowsForColumn:inMatrix:
<NSObject> **delegate method** 55
browser:draggingImageForRowsWithIndexes:inColumn:
withEvent:offset: <NSObject> **delegate method**
55
browser:isColumnValid: <NSObject> **delegate**
method 56
browser:namesOfPromisedFilesDroppedAtDestination:
forDraggedRowsWithIndexes:inColumn:
<NSObject> **delegate method** 56
browser:nextTypeSelectMatchFromRow:toRow:inColumn:
forString: <NSObject> **delegate method** 57
browser:numberOfRowsInColumn: <NSObject>
delegate method 58
browser:selectCellWithString:inColumn:
<NSObject> **delegate method** 58
browser:selectRow:inColumn: <NSObject> **delegate**
method 59
browser:shouldShowCellExpansionForRow:column:
<NSObject> **delegate method** 60
browser:shouldSizeColumn:forUserResize:toWidth:
<NSObject> **delegate method** 60

browser:shouldTypeSelectForEvent:
withCurrentSearchString: <NSObject> **delegate**
method 61
browser:sizeToFitWidthOfColumn: <NSObject>
delegate method 61
browser:titleOfColumn: <NSObject> **delegate**
method 62
browser:typeSelectStringForRow:inColumn:
<NSObject> **delegate method** 62
browser:validateDrop:proposedRow:column:
dropOperation: <NSObject> **delegate method** 63
browser:willDisplayCell:atRow:column:
<NSObject> **delegate method** 64
browser:writeRowsWithIndexes:inColumn:
toPasteboard: <NSObject> **delegate method** 65
browserColumnConfigurationDidChange:
<NSObject> **delegate method** 66
browserDidScroll: <NSObject> **delegate method** 66
browserWillScroll: <NSObject> **delegate method**
66

C

canDragRowsWithIndexes:inColumn:withEvent:
instance method 18
cellClass **class method** 15
cellPrototype **instance method** 19
columnContentWidthForColumnWidth: **instance**
method 19
columnOfMatrix: **instance method** 20
columnResizingType **instance method** 20
columnsAutosaveName **instance method** 20
columnWidthForColumnContentWidth: **instance**
method 21

D

delegate **instance method** 21
displayAllColumns **instance method** 22

displayColumn: **instance method** 22
doClick: **instance method** 22
doDoubleClick: **instance method** 23
doubleAction **instance method** 23
draggingImageForRowsWithIndexes:inColumn:
withEvent:offset: **instance method** 23
draggingSourceOperationMaskForLocal: **instance method** 24
drawTitleOfColumn:inRect: **instance method** 24

F

firstVisibleColumn **instance method** 25
frameOfColumn: **instance method** 25
frameOfInsideOfColumn: **instance method** 25

H

hasHorizontalScroller **instance method** 26

I

isLoading **instance method** 26
isOpaque **instance method** 26
isTitled **instance method** 27

L

lastColumn **instance method** 27
lastVisibleColumn **instance method** 27
loadColumnZero **instance method** 28
loadedCellAtRow:column: **instance method** 28

M

matrixClass **instance method** 28
matrixInColumn: **instance method** 29
maxVisibleColumns **instance method** 29
minColumnWidth **instance method** 29

N

namesOfPromisedFilesDroppedAtDestination:
instance method 30

NSBrowserAutoColumnResizing **constant** 67
NSBrowserColumnConfigurationDidChangeNotification
notification 68
NSBrowserColumnResizingType **data type** 67
NSBrowserDropOperation **data type** 68
NSBrowserNoColumnResizing **constant** 67
NSBrowserUserColumnResizing **constant** 67
numberOfVisibleColumns **instance method** 30

P

path **instance method** 30
pathSeparator **instance method** 31
pathToColumn: **instance method** 31
prefersAllColumnUserResizing **instance method** 32

R

reloadColumn: **instance method** 32
removeSavedColumnsWithAutosaveName: **class method** 15
reusesColumns **instance method** 33

S

scrollColumnsLeftBy: **instance method** 33
scrollColumnsRightBy: **instance method** 33
scrollColumnToVisible: **instance method** 34
scrollViaScroller: **instance method** 34
selectAll: **instance method** 34
selectedCell **instance method** 35
selectedCellInColumn: **instance method** 35
selectedCells **instance method** 36
selectedColumn **instance method** 36
selectedRowInColumn: **instance method** 36
selectedRowIndexInColumn: **instance method** 37
selectRow:inColumn: **instance method** 37
selectRowIndex:inColumn: **instance method** 38
sendAction **instance method** 38
sendsActionOnArrowKeys **instance method** 38
separatesColumns **instance method** 39
setAcceptsArrowKeys: **instance method** 39
setAllowsBranchSelection: **instance method** 39
setAllowsEmptySelection: **instance method** 40
setAllowsMultipleSelection: **instance method** 40
setAllowsTypeSelect: **instance method** 40
setBackground-color: **instance method** 41
setCellClass: **instance method** 41
setCellPrototype: **instance method** 42

setColumnResizingType: [instance method 42](#)
 setColumnsAutosaveName: [instance method 42](#)
 setDelegate: [instance method 43](#)
 setDoubleClickAction: [instance method 43](#)
 setDraggingSourceOperationMask:forLocal:
 [instance method 44](#)
 setHasHorizontalScroller: [instance method 44](#)
 setLastColumn: [instance method 45](#)
 setMatrixClass: [instance method 45](#)
 setMaxVisibleColumns: [instance method 45](#)
 setMinColumnWidth: [instance method 46](#)
 setPath: [instance method 46](#)
 setPathSeparator: [instance method 47](#)
 setPrefersAllColumnUserResizing: [instance
method 47](#)
 setReusesColumns: [instance method 48](#)
 setSendsActionOnArrowKeys: [instance method 48](#)
 setSeparatesColumns: [instance method 48](#)
 setTakesTitleFromPreviousColumn: [instance
method 49](#)
 setTitle:ofColumn: [instance method 49](#)
 setTitled: [instance method 50](#)
 setWidth:ofColumn: [instance method 50](#)

T

takesTitleFromPreviousColumn [instance method 51](#)
 tile [instance method 51](#)
 titleFrameOfColumn: [instance method 51](#)
 titleHeight [instance method 52](#)
 titleOfColumn: [instance method 52](#)

U

updateScroller [instance method 52](#)

V

validateVisibleColumns [instance method 53](#)

W

widthOfColumn: [instance method 53](#)