NSTabView Class Reference

Cocoa > User Experience

2006-05-23

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

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.

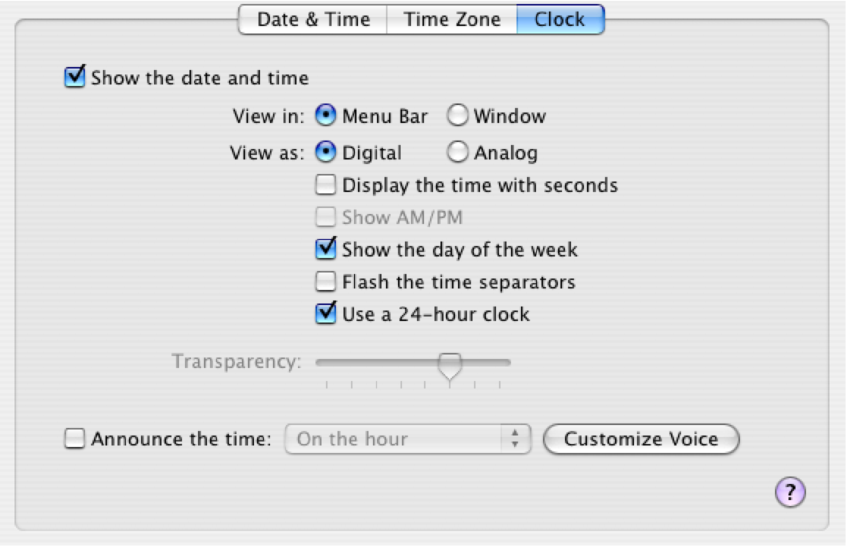
Contents

NSTabView Class Reference

Inherits from	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	Tab Views
Declared in	NSTabView.h
Related sample code	EnhancedAudioBurn JSPong MyPhoto PDF Annotation Editor Reducer

Overview

An `NSTabView` object provides a convenient way to present information in multiple pages. The view contains a row of tabs that give the appearance of folder tabs, as shown in the following figure. The user selects the desired page by clicking the appropriate tab or using the arrow keys to move between pages. Each page displays a view hierarchy provided by your application.



☑ Show the date and time

View in: ● Menu Bar ○ Window

View as: ● Digital ○ Analog

□ Display the time with seconds

□ Show AM/PM

☑ Show the day of the week

□ Flash the time separators

☑ Use a 24-hour clock

Transparency:

□ Announce the time: On the hour

Tasks

Adding and Removing Tabs

- `addTabViewItem:` (page 9)

  Adds the tab item specified by *tabViewItem*.

- `insertTabViewItem:atIndex:` (page 13)

  Inserts *tabViewItem* into the receiver’s array of tab view items at *index*.

- `removeTabViewItem:` (page 14)

  Removes the item specified by *tabViewItem* from the receiver’s array of tab view items.

Accessing Tabs

- `indexOfTabViewItem:` (page 12)

  Returns the index of the item that matches *tabViewItem*, or `NSNotFound` if the item is not found.

- `indexOfTabViewItemWithIdentifier:` (page 12)

  Returns the index of the item that matches *identifier*, or `NSNotFound` if the item is not found.

- `numberOfTabViewItems` (page 13)

  Returns the number of items in the receiver’s array of tab view items.

- `tabViewItemAtIndex:` (page 19)

  Returns the tab view item at *index* in the tab view’s array of items.

- `tabViewItems` (page 20)

  Returns the receiver’s array of tab view items.

Selecting a Tab

- `selectFirstTabViewItem:` (page 15)
  This action method selects the first tab view item.
- `selectLastTabViewItem:` (page 15)
  This action method selects the last tab view item.
- `selectNextTabViewItem:` (page 15)
  This action method selects the next tab view item in the sequence.
- `selectPreviousTabViewItem:` (page 15)
  This action method selects the previous tab view item in the sequence.
- `selectTabViewItem:` (page 16)
  Selects the tab view item specified by *`tabViewItem`*.
- `selectTabViewItemAtIndex:` (page 16)
  Selects the tab view item specified by *`index`*.
- `selectTabViewItemWithIdentifier:` (page 17)
  Selects the tab view item specified by *`identifier`*.
- `selectedTabViewItem` (page 14)
  Returns the tab view item for the currently selected tab, or `nil` if no item is selected.
- `takeSelectedTabViewItemFromSender:` (page 21)
  If *`sender`* responds to the `indexOfSelectedItem` method, this method invokes that method and selects the tab view item at the specified index. If *`sender`* does not respond to `indexOfSelectedItem` but is an instance of NSMatrix, this method uses the index of the matrix's currently selected cell.

Modifying the Font

- `font` (page 12)
  Returns the font for tab label text.
- `setFont:` (page 19)
  Sets the font for tab label text to *`font`*.

Modifying the Tab Type

- `setTabViewType:` (page 19)
  Sets the tab type to *`tabViewType`*.
- `tabViewType` (page 20)
  Returns the tab type for the receiver.

Modifying Controls Tint

- `controlTint` (page 11)
  Returns the receiver's control tint.
- `setControlTint:` (page 18)
  Sets the receiver's control tint to *`controlTint`*.

Manipulating the Background

- `drawsBackground` (page 11)
  Returns `YES` if the receiver draws a background color when the tab view type is `NSNoTabsNoBorder`. If the receiver uses bezeled edges or a line border, the appropriate background color for that border is used.
- `setDrawsBackground:` (page 18)
  Sets whether a background is drawn when the view type is `NSNoTabsNoBorder` to *`flag`*.

Determining the Size

- `minimumSize` (page 13)
  Returns the minimum size necessary for the receiver to display tabs in a useful way.
- `contentRect` (page 10)
  Returns the rectangle describing the content area of the receiver.
- `controlSize` (page 10)
  Returns the size of the receiver.
- `setControlSize:` (page 17)
  Sets the size of the receiver to *`controlSize`*.

Truncating Tab Labels

- `allowsTruncatedLabels` (page 10)
  Returns `YES` if the receiver allows truncating for labels that don't fit on a tab.
- `setAllowsTruncatedLabels:` (page 17)
  Sets whether the receiver allows truncating for names that don't fit on a tab to *`allowTruncatedLabels`*.

Assigning a Delegate

- `setDelegate:` (page 18)
  Sets the receiver's delegate to *`anObject`*.
- `delegate` (page 11)
  Returns the receiver's delegate.

Event Handling

- `tabViewItemAtPoint:` (page 20)
  Returns the tab view item identified by *`point`*.

View's Window

- `window` (page 21)
  Returns the receiver's window object, or `nil` if there is none.

Selecting an item

- `tabView:shouldSelectTabViewItem:` (page 22) *delegate method*
  Invoked just before *`tabViewItem`* in *`tabView`* is selected.
- `tabView:willSelectTabViewItem:` (page 22) *delegate method*
  Informs the delegate that *`tabView`* is about to select *`tabViewItem`*.
- `tabView:didSelectTabViewItem:` (page 21) *delegate method*
  Informs the delegate that *`tabView`* has selected *`tabViewItem`*.

Changing number of items in view

- `tabViewDidChangeNumberOfTabViewItems:` (page 22) *delegate method*
  Informs the delegate that the number of tab view items in *`tabView`* has changed.

Instance Methods

addTabViewItem:

Adds the tab item specified by *`tabViewItem`*.

```
- (void)addTabViewItem:(NSTabViewItem *)tabViewItem
```

Discussion
The item is added at the end of the array of tab items, so the new tab appears on the right side of the view. If the delegate supports it, invokes the delegate's `tabViewDidChangeNumberOfTabViewItems:` (page 22) method.

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

See Also
- `insertTabViewItem:atIndex:` (page 13)
- `numberOfTabViewItems` (page 13)
- `removeTabViewItem:` (page 14)
- `tabViewItemAtIndex:` (page 19)
- `tabViewItems` (page 20)

Declared In
`NSTabView.h`

allowsTruncatedLabels

Returns YES if the receiver allows truncating for labels that don't fit on a tab.

```
- (BOOL)allowsTruncatedLabels
```

Discussion
The default is NO. When truncating is allowed, the tab view inserts an ellipsis, if necessary, to fit a label in the tab.

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

See Also
- setAllowsTruncatedLabels: (page 17)

Declared In
NSTabView.h

contentRect

Returns the rectangle describing the content area of the receiver.

```
- (NSRect)contentRect
```

Discussion
This area does not include the space required for the receiver's tabs or borders (if any).

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

Declared In
NSTabView.h

controlSize

Returns the size of the receiver.

```
- (NSControlSize)controlSize
```

Discussion
Valid return values are described in NSCell's "Constants".

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

See Also
- setControlSize: (page 17)

Declared In
NSTabView.h

controlTint

Returns the receiver's control tint.

```
- (NSControlTint)controlTint
```

Discussion
Valid return values are described in "Constants" (page 23).

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

See Also
`- setControlTint:` (page 18)

Declared In
`NSTabView.h`

delegate

Returns the receiver's delegate.

```
- (id)delegate
```

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

See Also
`- setDelegate:` (page 18)

Declared In
`NSTabView.h`

drawsBackground

Returns `YES` if the receiver draws a background color when the tab view type is `NSNoTabsNoBorder`. If the receiver uses bezeled edges or a line border, the appropriate background color for that border is used.

```
- (BOOL)drawsBackground
```

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

See Also
`- setTabViewType:` (page 19)
`- setDrawsBackground:` (page 18)

Declared In
`NSTabView.h`

font

Returns the font for tab label text.

```
- (NSFont *)font
```

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

See Also
- `- setFont:` (page 19)

Declared In
`NSTabView.h`

indexOfTabViewItem:

Returns the index of the item that matches *`tabViewItem`*, or `NSNotFound` if the item is not found.

```
- (NSInteger)indexOfTabViewItem:(NSTabViewItem *)tabViewItem
```

Discussion
A tab view keeps an array containing one tab view item for each tab in the view—this array is the one that is searched. The returned index is base 0.

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

See Also
- `- indexOfTabViewItemWithIdentifier:` (page 12)
- `- insertTabViewItem:atIndex:` (page 13)
- `- numberOfTabViewItems` (page 13)
- `- tabViewItemAtIndex:` (page 19)

Declared In
`NSTabView.h`

indexOfTabViewItemWithIdentifier:

Returns the index of the item that matches *`identifier`*, or `NSNotFound` if the item is not found.

```
- (NSInteger)indexOfTabViewItemWithIdentifier:(id)identifier
```

Discussion
A tab view keeps an array containing one tab view item for each tab in the view—this array is the one that is searched. The returned index is base 0.

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

See Also
- `- indexOfTabViewItem:` (page 12)
- `- insertTabViewItem:atIndex:` (page 13)

- numberOfTabViewItems (page 13)
- tabViewItemAtIndex: (page 19)

Declared In
NSTabView.h

insertTabViewItem:atIndex:

Inserts *tabViewItem* into the receiver's array of tab view items at *index*.

```
- (void)insertTabViewItem:(NSTabViewItem *)tabViewItem atIndex:(NSInteger)index
```

Discussion
The *index* parameter is base 0. If there is a delegate and the delegate supports it, sends the delegate the tabViewDidChangeNumberOfTabViewItems: (page 22) message.

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

See Also
- indexOfTabViewItem: (page 12)
- indexOfTabViewItemWithIdentifier: (page 12)
- numberOfTabViewItems (page 13)
- tabViewItemAtIndex: (page 19)

Declared In
NSTabView.h

minimumSize

Returns the minimum size necessary for the receiver to display tabs in a useful way.

```
- (NSSize)minimumSize
```

Discussion
You can use the value returned by this method to limit how much a user can resize a tab view.

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

See Also
- setTabViewType: (page 19)

Declared In
NSTabView.h

numberOfTabViewItems

Returns the number of items in the receiver's array of tab view items.

```
- (NSInteger)numberOfTabViewItems
```

Discussion

Because there is one item in the array for each tab in the view, this number is equivalent to the number of tabs in the view.

Availability

Available in Mac OS X v10.0 and later.

See Also

- `indexOfTabViewItem:` (page 12)
- `tabViewItems` (page 20)

Declared In

`NSTabView.h`

removeTabViewItem:

Removes the item specified by *`tabViewItem`* from the receiver's array of tab view items.

```
- (void)removeTabViewItem:(NSTabViewItem *)tabViewItem
```

Discussion

If there is a delegate and the delegate supports it, sends the delegate the `tabViewDidChangeNumberOfTabViewItems:` (page 22) message.

Availability

Available in Mac OS X v10.0 and later.

See Also

- `addTabViewItem:` (page 9)
- `insertTabViewItem:atIndex:` (page 13)
- `tabViewItems` (page 20)

Declared In

`NSTabView.h`

selectedTabViewItem

Returns the tab view item for the currently selected tab, or `nil` if no item is selected.

```
- (NSTabViewItem *)selectedTabViewItem
```

Availability

Available in Mac OS X v10.0 and later.

See Also

- `selectTabViewItemAtIndex:` (page 16)

Related Sample Code

CarbonCocoaCoreImageTab

Declared In

`NSTabView.h`

selectFirstTabViewItem:

This action method selects the first tab view item.

```
- (void)selectFirstTabViewItem:(id)sender
```

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

See Also
`- selectTabViewItem:` (page 16)

Declared In
`NSTabView.h`

selectLastTabViewItem:

This action method selects the last tab view item.

```
- (void)selectLastTabViewItem:(id)sender
```

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

See Also
`- selectTabViewItem:` (page 16)

Declared In
`NSTabView.h`

selectNextTabViewItem:

This action method selects the next tab view item in the sequence.

```
- (void)selectNextTabViewItem:(id)sender
```

Discussion
If the currently visible item is the last item in the sequence, this method does nothing, and the last page remains displayed.

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

See Also
`- selectTabViewItem:` (page 16)

Declared In
`NSTabView.h`

selectPreviousTabViewItem:

This action method selects the previous tab view item in the sequence.

```
- (void)selectPreviousTabViewItem:(id)sender
```

Discussion

If the currently visible item is the first item in the sequence, this method does nothing, and the first page remains displayed.

Availability

Available in Mac OS X v10.0 and later.

See Also

- selectTabViewItem: (page 16)

Declared In

`NSTabView.h`

selectTabViewItem:

Selects the tab view item specified by *tabViewItem*.

```
- (void)selectTabViewItem:(NSTabViewItem *)tabViewItem
```

Discussion

If there is a delegate and the delegate supports it, sends the delegate the `tabView:shouldSelectTabViewItem:` (page 22) message.

Availability

Available in Mac OS X v10.0 and later.

See Also

- insertTabViewItem:atIndex: (page 13)
- selectedTabViewItem (page 14)

Declared In

`NSTabView.h`

selectTabViewItemAtIndex:

Selects the tab view item specified by *index*.

```
- (void)selectTabViewItemAtIndex:(NSInteger)index
```

Discussion

The *index* parameter is base 0. If there is a delegate and the delegate supports it, sends the delegate the `tabView:shouldSelectTabViewItem:` (page 22) message.

Availability

Available in Mac OS X v10.0 and later.

See Also

- insertTabViewItem:atIndex: (page 13)
- selectedTabViewItem (page 14)

Declared In

`NSTabView.h`

selectTabViewItemWithIdentifier:

Selects the tab view item specified by *identifier*.

```
- (void)selectTabViewItemWithIdentifier:(id)identifier
```

Availability

Available in Mac OS X v10.0 and later.

See Also

- `setIdentifier:` (NSTabViewItem)
- `identifier` (NSTabViewItem)
- `selectTabViewItemAtIndex:` (page 16)
- `selectedTabViewItem` (page 14)

Declared In

`NSTabView.h`

setAllowsTruncatedLabels:

Sets whether the receiver allows truncating for names that don't fit on a tab to *allowTruncatedLabels*.

```
- (void)setAllowsTruncatedLabels:(BOOL)allowTruncatedLabels
```

Availability

Available in Mac OS X v10.0 and later.

See Also

- `allowsTruncatedLabels` (page 10)

Declared In

`NSTabView.h`

setControlSize:

Sets the size of the receiver to *controlSize*.

```
- (void)setControlSize:(NSControlSize)controlSize
```

Discussion

Valid values for *controlSize* are described in "Constants".

Availability

Available in Mac OS X v10.0 and later.

See Also

- `controlSize` (page 10)

Declared In

`NSTabView.h`

setControlTint:

Sets the receiver's control tint to *`controlTint`*.

```
- (void)setControlTint:(NSControlTint)controlTint
```

Discussion

Valid values for *`controlTint`* are described in "Constants" (page 23).

Availability

Available in Mac OS X v10.0 and later.

See Also

`- controlTint` (page 11)

Declared In

`NSTabView.h`

setDelegate:

Sets the receiver's delegate to *`anObject`*.

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

Availability

Available in Mac OS X v10.0 and later.

See Also

`- delegate` (page 11)

Declared In

`NSTabView.h`

setDrawsBackground:

Sets whether a background is drawn when the view type is `NSNoTabsNoBorder` to *`flag`*.

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

Discussion

If the receiver has a bezeled border or a line border, the appropriate background for that border is used.

Availability

Available in Mac OS X v10.0 and later.

See Also

`- setTabViewType:` (page 19)

`- drawsBackground` (page 11)

Declared In

`NSTabView.h`

setFont:

Sets the font for tab label text to *font*.

```
- (void)setFont:(NSFont *)font
```

Discussion

Tab height is adjusted automatically to accommodate a new font size. If the view allows truncating, tab labels are truncated as needed.

Availability

Available in Mac OS X v10.0 and later.

See Also

`- allowsTruncatedLabels` (page 10)
`- font` (page 12)
`- setAllowsTruncatedLabels:` (page 17)

Declared In

`NSTabView.h`

setTabViewType:

Sets the tab type to *tabViewType*.

```
- (void)setTabViewType:(NSTabViewType)tabViewType
```

Discussion

The available types are described in "Constants" (page 23).

Availability

Available in Mac OS X v10.0 and later.

See Also

`- tabViewType` (page 20)

Declared In

`NSTabView.h`

tabViewItemAtIndex:

Returns the tab view item at *index* in the tab view's array of items.

```
- (NSTabViewItem *)tabViewItemAtIndex:(NSInteger)index
```

Discussion

The *index* parameter is base 0.

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

See Also
- indexOfTabViewItem: (page 12)
- insertTabViewItem:atIndex: (page 13)
- tabViewItems (page 20)

Declared In
`NSTabView.h`

tabViewItemAtPoint:

Returns the tab view item identified by *point*.

```
- (NSTabViewItem *)tabViewItemAtPoint:(NSPoint)point
```

Discussion
You can use this method to find a tab view item based on a user's mouse click.

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

Declared In
`NSTabView.h`

tabViewItems

Returns the receiver's array of tab view items.

```
- (NSArray *)tabViewItems
```

Discussion
A tab view keeps an array containing one tab view item for each tab in the view.

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

See Also
- numberOfTabViewItems (page 13)
- tabViewItemAtIndex: (page 19)

Declared In
`NSTabView.h`

tabViewType

Returns the tab type for the receiver.

```
- (NSTabViewType)tabViewType
```

Discussion
The available types are described in "Constants" (page 23).

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

Declared In
`NSTabView.h`

takeSelectedTabViewItemFromSender:

If *sender* responds to the `indexOfSelectedItem` method, this method invokes that method and selects the tab view item at the specified index. If *sender* does not respond to `indexOfSelectedItem` but is an instance of NSMatrix, this method uses the index of the matrix's currently selected cell.

```
- (void)takeSelectedTabViewItemFromSender:(id)sender
```

Discussion
The location of the selected cell is a zero-based number, obtained by counting the number of cells up to and including the selected cell. Cells are counted from left to right and from top to bottom. For example in a 5-by-5 matrix, if the selected cell is three rows down in column five (location [2,4] in the matrix), the corresponding index would be 14.

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

Declared In
`NSTabView.h`

window

Returns the receiver's window object, or `nil` if there is none.

```
- (NSWindow *)window
```

Discussion
If the receiver is not the currently visible tab view this method returns `nil`.

Delegate Methods

tabView:didSelectTabViewItem:

Informs the delegate that *tabView* has selected *tabViewItem*.

```
- (void)tabView:(NSTabView *)tabView didSelectTabViewItem:(NSTabViewItem
    *)tabViewItem
```

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

Declared In
`NSTabView.h`

tabView:shouldSelectTabViewItem:

Invoked just before *`tabViewItem`* in *`tabView`* is selected.

```
- (BOOL)tabView:(NSTabView *)tabView shouldSelectTabViewItem:(NSTabViewItem
    *)tabViewItem
```

Discussion
The delegate can return `NO` to prevent selection of specific tabs.

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

Declared In
`NSTabView.h`

tabView:willSelectTabViewItem:

Informs the delegate that *`tabView`* is about to select *`tabViewItem`*.

```
- (void)tabView:(NSTabView *)tabView willSelectTabViewItem:(NSTabViewItem
    *)tabViewItem
```

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

Declared In
`NSTabView.h`

tabViewDidChangeNumberOfTabViewItems:

Informs the delegate that the number of tab view items in *`tabView`* has changed.

```
- (void)tabViewDidChangeNumberOfTabViewItems:(NSTabView *)tabView
```

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

See Also
`- numberOfTabViewItems` (page 13)

Declared In
`NSTabView.h`

Constants

NSTabViewType

These constants specify the tab view's type:

```
typedef enum _NSTabViewType {
    NSTopTabsBezelBorder     = 0,
    NSLeftTabsBezelBorder    = 1,
    NSBottomTabsBezelBorder  = 2,
    NSRightTabsBezelBorder   = 3,
    NSNoTabsBezelBorder      = 4,
    NSNoTabsLineBorder       = 5,
    NSNoTabsNoBorder         = 6
} NSTabViewType;
```

Constants

`NSTopTabsBezelBorder`

The view includes tabs on the top of the view and has a bezeled border (the default).

Available in Mac OS X v10.0 and later.

Declared in `NSTabView.h`.

`NSNoTabsBezelBorder`

The view does not include tabs and has a bezeled border.

Available in Mac OS X v10.0 and later.

Declared in `NSTabView.h`.

`NSNoTabsLineBorder`

The view does not include tabs and has a lined border.

Available in Mac OS X v10.0 and later.

Declared in `NSTabView.h`.

`NSNoTabsNoBorder`

The view does not include tabs and has no border.

Available in Mac OS X v10.0 and later.

Declared in `NSTabView.h`.

`NSBottomTabsBezelBorder`

Tabs are on the bottom of the view with a bezeled border.

Available in Mac OS X v10.0 and later.

Declared in `NSTabView.h`.

`NSLeftTabsBezelBorder`

Tabs are on the left of the view with a bezeled border.

Available in Mac OS X v10.0 and later.

Declared in `NSTabView.h`.

`NSRightTabsBezelBorder`

Tabs are on the right of the view with a bezeled border.

Available in Mac OS X v10.0 and later.

Declared in `NSTabView.h`.

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

Declared In
`NSTabView.h`

Document Revision History

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

Date	Notes
2006-05-23	First publication of this content as a separate document.

Index