

---

# NSTabView Class Reference

User Experience: Windows & Views



2009-06-23



Apple Inc.  
© 2009 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, Carbon, Cocoa, iChat, 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

## **NSTableView Class Reference** 5

---

Overview	5
Tasks	6
Adding and Removing Tabs	6
Accessing Tabs	6
Selecting a Tab	7
Modifying the Font	7
Modifying the Tab Type	7
Modifying Controls Tint	7
Manipulating the Background	8
Determining the Size	8
Truncating Tab Labels	8
Assigning a Delegate	8
Event Handling	8
Instance Methods	9
addTabViewItem:	9
allowsTruncatedLabels	9
contentRect	10
controlSize	10
controlTint	10
delegate	11
drawsBackground	11
font	11
indexOfTabViewItem:	12
indexOfTabViewItemWithIdentifier:	12
insertTabViewItem:atIndex:	13
minimumSize	13
numberOfTabViewItems	14
removeTabViewItem:	14
selectedTabViewItem	15
selectFirstTabViewItem:	15
selectLastTabViewItem:	15
selectNextTabViewItem:	16
selectPreviousTabViewItem:	16
selectTabViewItem:	17
selectTabViewItemAtIndex:	17
selectTabViewItemWithIdentifier:	18
setAllowsTruncatedLabels:	18
setControlSize:	19
setControlTint:	19
setDelegate:	19

- setDrawsBackground: 20
- setFont: 20
- setTabViewType: 21
- tabViewItemAtIndex: 21
- tabViewItemAtPoint: 21
- tabViewItems 22
- tabViewType 22
- takeSelectedTabViewItemFromSender: 22
- Constants 23
  - NSTabViewType 23
  - NSAppKitVersionNumberWithDirectionalTabs 24

---

**Document Revision History 25**

---

**Index 27**

---

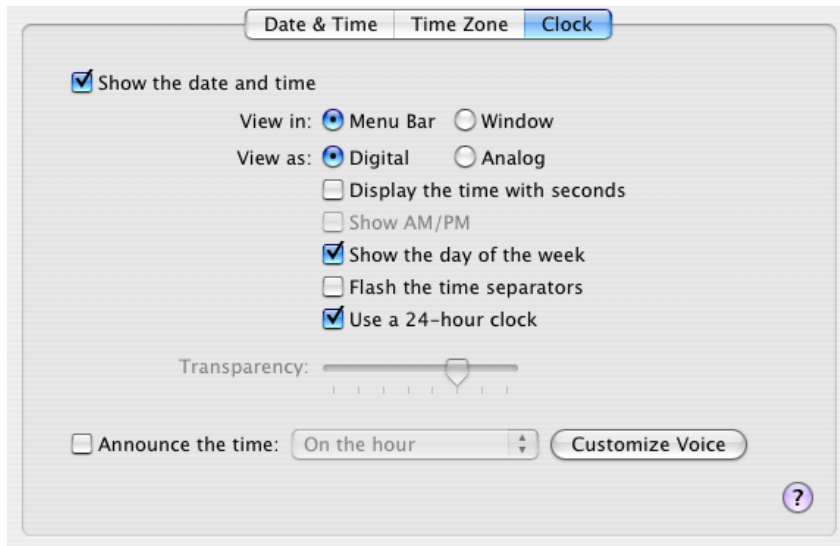
# NSTabView Class Reference

---

<b>Inherits from</b>	NSView : NSResponder : NSObject
<b>Conforms to</b>	NSAnimatablePropertyContainer (NSView) NSCoding (NSResponder) NSObject (NSObject)
<b>Framework</b>	/System/Library/Frameworks/AppKit.framework
<b>Availability</b>	Available in Mac OS X v10.0 and later.
<b>Companion guide</b>	Tab Views
<b>Declared in</b>	NSTabView.h
<b>Related sample code</b>	EnhancedAudioBurn GLUT iChatTheater MyPhoto PDF Annotation Editor

## 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.



## 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 specified item in the tab view.
- [indexOfTabViewItemWithIdentifier:](#) (page 12)  
Returns the index of the item that matches the specified identifier. *identifier*, or `NSNotFound` if the item is not found.
- [numberOfTabViewItems](#) (page 14)  
Returns the number of items in the receiver's array of tab view items.
- [tabViewItemAtIndex:](#) (page 21)  
Returns the tab view item at *index* in the tab view's array of items.
- [tabViewItems](#) (page 22)  
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 16)  
This action method selects the next tab view item in the sequence.
- [selectPreviousTabViewItem:](#) (page 16)  
This action method selects the previous tab view item in the sequence.
- [selectTabViewItem:](#) (page 17)  
Selects the specified tab view item.
- [selectTabViewItemAtIndex:](#) (page 17)  
Selects the tab view item specified by *index*.
- [selectTabViewItemWithIdentifier:](#) (page 18)  
Selects the tab view item specified by *identifier*.
- [selectedTabViewItem](#) (page 15)  
Returns the tab view item for the currently selected tab
- [takeSelectedTabViewItemFromSender:](#) (page 22)  
Sets the selected tab view item to the selected item obtained from the sender.

## Modifying the Font

- [font](#) (page 11)  
Returns the font for tab label text.
- [setFont:](#) (page 20)  
Sets the font for tab label text to *font*.

## Modifying the Tab Type

- [setTabViewType:](#) (page 21)  
Sets the tab type to *tabViewType*.
- [tabViewType](#) (page 22)  
Returns the tab type for the receiver.

## Modifying Controls Tint

- [controlTint](#) (page 10)  
Returns the receiver's control tint.
- [setControlTint:](#) (page 19)  
Sets the receiver's control tint to *controlTint*.

## Manipulating the Background

- `drawsBackground` (page 11)  
Returns whether if the receiver draws a background color when the tab view type is `NSNoTabsNoBorder`.
- `setDrawsBackground:` (page 20)  
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 19)  
Sets the size of the receiver to *controlSize*.

## Truncating Tab Labels

- `allowsTruncatedLabels` (page 9)  
Returns whether if the receiver allows truncating for labels that don't fit on a tab.
- `setAllowsTruncatedLabels:` (page 18)  
Sets whether the receiver allows truncating for names that don't fit on a tab.

## Assigning a Delegate

- `setDelegate:` (page 19)  
Sets the receiver's delegate.
- `delegate` (page 11)  
Returns the receiver's delegate.

## Event Handling

- `tabViewItemAtPoint:` (page 21)  
Returns the tab view item at the specified point.

## Instance Methods

### addTabViewItem:

Adds the tab item specified by *tabViewItem*.

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

#### Parameters

*tabViewItem*

The tab view item to be added.

#### 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:` method.

#### Availability

Available in Mac OS X v10.0 and later.

#### See Also

- [insertTabViewItemAtIndex:](#) (page 13)
- [numberOfTabViewItems](#) (page 14)
- [removeTabViewItem:](#) (page 14)
- [tabViewItemAtIndex:](#) (page 21)
- [tabViewItems](#) (page 22)

#### Declared In

NSTabView.h

### allowsTruncatedLabels

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

```
- (BOOL)allowsTruncatedLabels
```

#### Return Value

YES if the receiver allows truncating for labels that don't fit on a tab, otherwise NO.

#### 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 18)

#### Declared In

NSTabView.h

## contentRect

Returns the rectangle describing the content area of the receiver.

- (NSRect)contentRect

### Return Value

Returns the rectangle describing the content area of the receiver.

### 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

### Return Value

Valid return values are described in `controlSizes` in *NSCell Class Reference*.

### Availability

Available in Mac OS X v10.0 and later.

### See Also

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

### Declared In

NSTabView.h

## controlTint

Returns the receiver's control tint.

- (NSControlTint)controlTint

### Return Value

The tab view's control tint. See `NSControlTint` for the supported values.

### Availability

Available in Mac OS X v10.0 and later.

### See Also

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

### Declared In

NSTabView.h

## delegate

Returns the receiver's delegate.

- (id < NSTabViewDelegate >)delegate

### Return Value

The object's delegate.

### Availability

Available in Mac OS X v10.0 and later.

### See Also

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

### Declared In

NSTabView.h

## drawsBackground

Returns whether if the receiver draws a background color when the tab view type is `NSNoTabsNoBorder`.

- (BOOL)drawsBackground

### Return Value

YES if the receiver draws a background color when the tab view type is `NSNoTabsNoBorder`, otherwise NO.

### Discussion

If the receiver uses beveled edges or a line border, the appropriate background color for that border is used.

### Availability

Available in Mac OS X v10.0 and later.

### See Also

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

- [setDrawsBackground:](#) (page 20)

### Declared In

NSTabView.h

## font

Returns the font for tab label text.

- (NSFont \*)font

### Return Value

Returns the tab view label font.

### Availability

Available in Mac OS X v10.0 and later.

### See Also

- [setFont:](#) (page 20)

**Declared In**

NSTabView.h

**indexOfTabViewItem:**

Returns the index of the specified item in the tab view.

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

**Parameters***tabViewItem*

The tab view item.

**Return Value**The zero-based index of *tabViewItem*, or [NSNotFound] if the item is not found.**Discussion**

The returned index is zero-based.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [indexOfTabViewItemWithIdentifier:](#) (page 12)
- [insertTabViewItemAtIndex:](#) (page 13)
- [numberOfTabViewItems](#) (page 14)
- [tabViewItemAtIndex:](#) (page 21)

**Declared In**

NSTabView.h

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

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

**Parameters***identifier*

The identifier of a tab view item.

**Return Value**The zero-based index of the tab view item corresponding to *identifier*, or [NSNotFound] if the item is not found.**Discussion**

The returned index is zero-based.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [indexOfTabViewItem:](#) (page 12)
- [insertTabViewItem:atIndex:](#) (page 13)
- [numberOfTabViewItems](#) (page 14)
- [tabViewItemAtIndex:](#) (page 21)

**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
```

**Parameters***tabViewItem*

The tab view item to be added.

*index*

The index at which to insert the tab view item. The *index* parameter is zero-based.

**Discussion**

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

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [indexOfTabViewItem:](#) (page 12)
- [indexOfTabViewItemWithIdentifier:](#) (page 12)
- [numberOfTabViewItems](#) (page 14)
- [tabViewItemAtIndex:](#) (page 21)

**Declared In**

NSTabView.h

**minimumSize**

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

```
- (NSSize)minimumSize
```

**Return Value**

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

**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 21)

**Declared In**

NSTabView.h

**numberOfTabViewItems**

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

```
- (NSInteger)numberOfTabViewItems
```

**Return Value**

The number of items in the tab view.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [indexOfTabViewItem:](#) (page 12)

- [tabViewItems](#) (page 22)

**Declared In**

NSTabView.h

**removeTabViewItem:**

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

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

**Parameters**

*tabViewItem*

The tab view item to be removed.

**Discussion**

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

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [addTabViewItem:](#) (page 9)

- [insertTabViewItem:atIndex:](#) (page 13)

- [tabViewItems](#) (page 22)

**Declared In**

NSTabView.h

## selectedTabViewItem

Returns the tab view item for the currently selected tab

- (NSTabViewItem \*)selectedTabViewItem

### Return Value

The currently selected tab view item, or `nil` if no item is selected.

### Availability

Available in Mac OS X v10.0 and later.

### See Also

- [selectTabViewItemAtIndex:](#) (page 17)

### Related Sample Code

iChatTheater

### Declared In

NSTabView.h

## selectFirstTabViewItem:

This action method selects the first tab view item.

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

### Parameters

*sender*

Typically the object that sent invoked the message.

### Availability

Available in Mac OS X v10.0 and later.

### See Also

- [selectTabViewItem:](#) (page 17)

### Declared In

NSTabView.h

## selectLastTabViewItem:

This action method selects the last tab view item.

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

### Parameters

*sender*

Typically the object that sent invoked the message.

### Availability

Available in Mac OS X v10.0 and later.

**See Also**

- [selectTabViewItem:](#) (page 17)

**Declared In**

NSTabView.h

**selectNextTabViewItem:**

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

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

**Parameters**

*sender*

Typically the object that sent invoked the message.

**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 17)

**Declared In**

NSTabView.h

**selectPreviousTabViewItem:**

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

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

**Parameters**

*sender*

Typically the object that sent invoked the message.

**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 17)

**Declared In**

NSTabView.h

## selectTabViewItem:

Selects the specified tab view item.

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

### Parameters

*tabViewItem*

The tab item to select.

### Discussion

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

### Availability

Available in Mac OS X v10.0 and later.

### See Also

- [insertTabViewItemAtIndex:](#) (page 13)
- [selectedTabViewItem](#) (page 15)

### Related Sample Code

From A View to A Movie  
 From A View to A Picture  
 iChatTheater

### Declared In

NSTabView.h

## selectTabViewItemAtIndex:

Selects the tab view item specified by *index*.

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

### Parameters

*index*

The index of the tab item to selected.

### Discussion

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

### Availability

Available in Mac OS X v10.0 and later.

### See Also

- [insertTabViewItemAtIndex:](#) (page 13)
- [selectedTabViewItem](#) (page 15)

### Related Sample Code

EnhancedAudioBurn  
 PDFKitLinker2  
 QTAudioContextInsert

XMLBrowser

**Declared In**  
NSTabView.h

### **selectTabViewItemWithIdentifier:**

Selects the tab view item specified by *identifier*.

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

#### **Parameters**

*identifier*

The identifier of the tab item to select.

#### **Availability**

Available in Mac OS X v10.0 and later.

#### **See Also**

- [setIdentifier: \(NSTabViewItem\)](#)
- [identifier \(NSTabViewItem\)](#)
- [selectTabViewItemAtIndex: \(page 17\)](#)
- [selectedTabViewItem \(page 15\)](#)

#### **Related Sample Code**

FinalCutPro\_AppleEvents

**Declared In**  
NSTabView.h

### **setAllowsTruncatedLabels:**

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

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

#### **Parameters**

*allowTruncatedLabels*

YES if the receiver allows truncating for labels that don't fit on a tab, otherwise NO.

#### **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**

- [allowsTruncatedLabels \(page 9\)](#)

**Declared In**  
NSTabView.h

**setSize:**

Sets the size of the receiver to *controlSize*.

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

**Parameters**

*controlSize*

The size of the receiver. Valid values are described in `Control Sizes` in *NSCell Class Reference*

**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
```

**Parameters**

*controlTint*

The tab view's control tint. See `NSControlTint` for the supported values.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [controlTint](#) (page 10)

**Declared In**

NSTabView.h

**setDelegate:**

Sets the receiver's delegate.

```
- (void)setDelegate:(id < NSTabViewDelegate >)anObject
```

**Parameters**

*anObject*

The delegate object. It must conform to the `NSTabViewDelegate` protocol.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [delegate](#) (page 11)

**Related Sample Code**

JSPong

**Declared In**

NSTabView.h

**setDrawsBackground:**

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

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

**Parameters***flag*

YES if the background should be drawn, otherwise NO.

**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 21)
- [drawsBackground](#) (page 11)

**Declared In**

NSTabView.h

**setFont:**

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

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

**Parameters***font*

The font to use as the tab view label 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 9)
- [font](#) (page 11)
- [setAllowsTruncatedLabels:](#) (page 18)

**Declared In**

NSTabView.h

**setTabViewType:**

Sets the tab type to *tabViewType*.

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

**Parameters**

*tabViewType*

The tab type to display the tabs. The supported values are listed in [NSTabViewType](#) (page 23)

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [tabViewType](#) (page 22)

**Declared In**

NSTabView.h

**tabViewItemAtIndex:**

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

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

**Parameters**

*index*

The index at which to insert the tab view item. The *index* parameter is zero-based.

**Return Value**

The tab view item at the specified index.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [indexOfTabViewItem:](#) (page 12)  
 - [insertTabViewItem:atIndex:](#) (page 13)  
 - [tabViewItems](#) (page 22)

**Declared In**

NSTabView.h

**tabViewItemAtPoint:**

Returns the tab view item at the specified point.

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

**Parameters**

*point*

The hit point.

**Return Value**

The tab view item under the hit point, or `nil` if no tab view item is under that location.

**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

**Return Value**

An array of tab view items.

**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 14)
- [tabViewItemAtIndex:](#) (page 21)

**Declared In**

`NSTabView.h`

**tabViewType**

Returns the tab type for the receiver.

- (NSTabViewType)tabViewType

**Return Value**

The tab type to display the tabs. The supported values are listed in [NSTabViewType](#) (page 23)

**Availability**

Available in Mac OS X v10.0 and later.

**Declared In**

`NSTabView.h`

**takeSelectedTabViewItemFromSender:**

Sets the selected tab view item to the selected item obtained from the sender.

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

### Parameters

*sender*

Typically the object that sent invoked the message.

### Discussion

If *sender* responds to the `indexOfSelectedItem` method, this method invokes that method and selects the tab view item at the specified index.

### Availability

Available in Mac OS X v10.0 and later.

### Declared In

NSTabView.h

## Constants

### NSTabViewType

These constants specify the tab view's type as used by `tabViewType` (page 22) and `setTabViewType:` (page 21).

```
enum {
    NSTopTabsBezelBorder      = 0,
    NSLeftTabsBezelBorder    = 1,
    NSBottomTabsBezelBorder  = 2,
    NSRightTabsBezelBorder   = 3,
    NSNoTabsBezelBorder      = 4,
    NSNoTabsLineBorder       = 5,
    NSNoTabsNoBorder         = 6
};
typedef NSUInteger 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 beveled 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 beveled 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 beveled border.

Available in Mac OS X v10.0 and later.

Declared in `NSTabView.h`.

## **NSAppKitVersionNumberWithDirectionalTabs**

This constant specifies the minimum version of the Application Kit that supports directional tabs.

```
#define NSAppKitVersionNumberWithDirectionalTabs 631.0
```

### **Constants**

`NSAppKitVersionNumberWithDirectionalTabs`

The specific version of the AppKit framework that introduced support for directional tab items. Developers should not need to use this constant unless they are writing applications for Mac OS X v10.1 and earlier.

Available in Mac OS X v10.2 and later.

Declared in `NSTabView.h`.

# Document Revision History

---

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

Date	Notes
2009-06-23	Added <code>NSAppKitVersionNumberWithDirectionalTabs</code> define.
2009-05-24	Updated for Mac OS X v10.6. Delegate methods moved to <code>NSTabViewDelegate Protocol Reference</code> .
2006-05-23	First publication of this content as a separate document.
	First publication of this content as a separate document.

## REVISION HISTORY

### Document Revision History

# Index

---

## A

---

`addTabViewItem:` [instance method 9](#)  
`allowsTruncatedLabels` [instance method 9](#)

## C

---

`contentRect` [instance method 10](#)  
`controlSize` [instance method 10](#)  
`controlTint` [instance method 10](#)

## D

---

`delegate` [instance method 11](#)  
`drawsBackground` [instance method 11](#)

## F

---

`font` [instance method 11](#)

## I

---

`indexOfTabViewItem:` [instance method 12](#)  
`indexOfTabViewItemWithIdentifier:` [instance method 12](#)  
`insertTabViewItem:atIndex:` [instance method 13](#)

## M

---

`minimumSize` [instance method 13](#)

## N

---

`NSAppKitVersionNumberWithDirectionalTabs` [24](#)  
`NSAppKitVersionNumberWithDirectionalTabs` [constant 24](#)  
`NSBottomTabsBezelBorder` [constant 24](#)  
`NSLeftTabsBezelBorder` [constant 24](#)  
`NSNoTabsBezelBorder` [constant 23](#)  
`NSNoTabsLineBorder` [constant 23](#)  
`NSNoTabsNoBorder` [constant 24](#)  
`NSRightTabsBezelBorder` [constant 24](#)  
`NSTabButtonType` [23](#)  
`NSTopTabsBezelBorder` [constant 23](#)  
`numberOfTabViewItems` [instance method 14](#)

## R

---

`removeTabViewItem:` [instance method 14](#)

## S

---

`selectedTabViewItem` [instance method 15](#)  
`selectFirstTabViewItem:` [instance method 15](#)  
`selectLastTabViewItem:` [instance method 15](#)  
`selectNextTabViewItem:` [instance method 16](#)  
`selectPreviousTabViewItem:` [instance method 16](#)  
`selectTabViewItemAtIndex:` [instance method 17](#)  
`selectTabViewItem:` [instance method 17](#)  
`selectTabViewItemWithIdentifier:` [instance method 18](#)  
`setAllowsTruncatedLabels:` [instance method 18](#)  
`setControlSize:` [instance method 19](#)  
`setControlTint:` [instance method 19](#)  
`setDelegate:` [instance method 19](#)  
`setDrawsBackground:` [instance method 20](#)  
`setFont:` [instance method 20](#)  
`setTabButtonType:` [instance method 21](#)

T

---

tabViewItemAtIndex: **instance method** [21](#)  
tabViewItemAtPoint: **instance method** [21](#)  
tabViewItems **instance method** [22](#)  
tabViewType **instance method** [22](#)  
takeSelectedTabViewItemFromSender: **instance method** [22](#)