

---

# NSTabViewItem Class Reference

[Cocoa](#) > [User Experience](#)



2006-05-23



Apple Inc.  
© 2006 Apple Computer, 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, 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

## **NSTabViewItem Class Reference 5**

---

Overview	5
Adopted Protocols	5
Tasks	6
Creating a Tab View Item	6
Working with Labels	6
Checking the Tab Display State	6
Assigning an Identifier Object	6
Setting the Color	6
Assigning a View	6
Setting the Initial First Responder	7
Accessing the Parent Tab View	7
Instance Methods	7
color	7
drawLabel:inRect:	7
identifier	8
initialFirstResponder	8
initWithIdentifier:	8
label	9
setColor:	9
setIdentifier:	9
setInitialFirstResponder:	10
setLabel:	10
setView:	10
sizeOfLabel:	11
tabState	11
tabView	12
view	12
Constants	13
NSTabState	13

## **Document Revision History 15**

---

## **Index 17**

---



# NSTabViewItem Class Reference

---

<b>Inherits from</b>	NSObject
<b>Conforms to</b>	NSCoding 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>	NSTabViewItem.h
<b>Related sample code</b>	CarbonCocoaCoreImageTab CoreRecipes JSPong Reducer

## Overview

An NSTabViewItem is a convenient way for presenting information in multiple pages. A tab view is usually distinguished by a row of tabs that give the visual appearance of folder tabs. When the user clicks a tab, the tab view displays a view page provided by your application. A tab view keeps a zero-based array of NSTabViewItems, one for each tab in the view.

## Adopted Protocols

### NSCoding

- encodeWithCoder:
- initWithCoder:

## Tasks

### Creating a Tab View Item

- [initWithIdentifier:](#) (page 8)  
Performs default initialization for the receiver.

### Working with Labels

- [drawLabel:inRect:](#) (page 7)  
Draws the receiver's label in *tabRect*, which is the area between the curved end caps.
- [label](#) (page 9)  
Returns the label text for the receiver.
- [setLabel:](#) (page 10)  
Sets the label text for the receiver to *label*.
- [sizeOfLabel:](#) (page 11)  
Calculates the size of the receiver's label.

### Checking the Tab Display State

- [tabState](#) (page 11)  
Returns the current display state of the tab associated with the receiver.

### Assigning an Identifier Object

- [identifier](#) (page 8)  
Returns the receiver's optional identifier object.
- [setIdentifier:](#) (page 9)  
Sets the receiver's optional identifier object to *identifier*.

### Setting the Color

- [color](#) (page 7)  
Returns the color for the receiver.
- [setColor:](#) (page 9)  
Deprecated. NSTabViewItems use a color supplied by the current theme.

### Assigning a View

- [view](#) (page 12)  
Returns the view associated with the receiver.

- [setView:](#) (page 10)  
Sets the view associated with the receiver to *view*.

## Setting the Initial First Responder

- [initialFirstResponder](#) (page 8)  
Returns the initial first responder for the view associated with the receiver.
- [setInitialFirstResponder:](#) (page 10)  
Sets the initial first responder for the view associated with the receiver (the view that is displayed when a user clicks on the tab) to *view*.

## Accessing the Parent Tab View

- [tabView](#) (page 12)  
Returns the parent tab view for the receiver.

## Instance Methods

### color

Returns the color for the receiver.

- (NSColor \*)color

### Discussion

The color is specified by the current theme.

### Availability

Available in Mac OS X v10.0 and later.

### See Also

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

### Declared In

NSTabViewItem.h

### drawLabel:inRect:

Draws the receiver's label in *tabRect*, which is the area between the curved end caps.

- (void)drawLabel:(BOOL)shouldTruncateLabel inRect:(NSRect)tabRect

### Discussion

If *shouldTruncateLabel* is NO, draws the full label in the rectangle specified by *tabRect*. If *shouldTruncateLabel* is YES, draws the truncated label. You can override this method to perform customized label drawing. For example, you might want to add an icon to each tab in the view.

### Availability

Available in Mac OS X v10.0 and later.

### See Also

- [sizeOfLabel:](#) (page 11)

### Declared In

NSTabViewItem.h

## identifier

Returns the receiver's optional identifier object.

- (id)identifier

### Discussion

To customize how your application works with tabs, you can initialize each tab view item with an identifier object.

### Availability

Available in Mac OS X v10.0 and later.

### See Also

- [initWithIdentifier:](#) (page 8)

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

### Declared In

NSTabViewItem.h

## initialFirstResponder

Returns the initial first responder for the view associated with the receiver.

- (id)initialFirstResponder

### Availability

Available in Mac OS X v10.0 and later.

### See Also

- [setInitialFirstResponder:](#) (page 10)

### Declared In

NSTabViewItem.h

## initWithIdentifier:

Performs default initialization for the receiver.

- (id)initWithIdentifier:(id)identifier



**Discussion**

Sets the receiver's identifier object to *identifier*, if it is not `nil`. Use this method when creating tab view items programmatically.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [identifier](#) (page 8)
- [setIdentifier:](#) (page 9)

**Declared In**

NSTabViewItem.h

**label**

Returns the label text for the receiver.

```
- (NSString *)label
```

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [setLabel:](#) (page 10)

**Declared In**

NSTabViewItem.h

**setColor:**

Deprecated. NSTabViewItems use a color supplied by the current theme.

```
- (void)setColor:(NSColor *)color
```

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [color](#) (page 7)

**Declared In**

NSTabViewItem.h

**setIdentifier:**

Sets the receiver's optional identifier object to *identifier*.

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

**Discussion**

To customize how your application works with tabs, you can specify an identifier object for each tab view item.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [identifier](#) (page 8)
- [initWithIdentifier:](#) (page 8)

**Declared In**

NSTabViewItem.h

**setInitialFirstResponder:**

Sets the initial first responder for the view associated with the receiver (the view that is displayed when a user clicks on the tab) to *view*.

```
- (void)setInitialFirstResponder:(NSView *)view
```

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [initialFirstResponder](#) (page 8)

**Declared In**

NSTabViewItem.h

**setLabel:**

Sets the label text for the receiver to *label*.

```
- (void)setLabel:(NSString *)label
```

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [label](#) (page 9)

**Related Sample Code**

CoreRecipes

**Declared In**

NSTabViewItem.h

**setView:**

Sets the view associated with the receiver to *view*.

```
- (void)setView:(NSView *)view
```

**Discussion**

This is the view displayed when a user clicks the tab. When you set a new view, the old view is released.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [view](#) (page 12)

**Related Sample Code**

CoreRecipes

**Declared In**

NSTabViewItem.h

**sizeOfLabel:**

Calculates the size of the receiver's label.

```
- (NSSize)sizeOfLabel:(BOOL)shouldTruncateLabel
```

**Discussion**

If *shouldTruncateLabel* is NO, returns the size of the receiver's full label. If *shouldTruncateLabel* is YES, returns the truncated size. If your application does anything to change the size of tab labels, such as overriding the [drawLabel:inRect:](#) (page 7) method to add an icon to each tab, you should override [sizeOfLabel:](#) (page 11) too so the NSTabView knows the correct size for the tab label.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [drawLabel:inRect:](#) (page 7)

- [setFont:](#) (NSTabView)

**Declared In**

NSTabViewItem.h

**tabState**

Returns the current display state of the tab associated with the receiver.

```
- (NSTabState)tabState
```

**Discussion**

The possible values are `NSSelectedTab`, `NSBackgroundTab`, or `NSPressedTab`. Your application does not directly set the tab state.

**Availability**

Available in Mac OS X v10.0 and later.

**Declared In**

NSTabViewItem.h

**tabView**

Returns the parent tab view for the receiver.

- (NSTabView \*)tabView

**Discussion**

Note that this is the tab view itself, not the view displayed when a user clicks the tab.

A tab view item normally learns about its parent tab view when it is inserted into the view's array of items. The NSTabView methods `addTabViewItem:` and `insertTabViewItem:atIndex:` set the tab view for the added or inserted item.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [setView:](#) (page 10)

- [view](#) (page 12)

**Declared In**

NSTabViewItem.h

**view**

Returns the view associated with the receiver.

- (id)view

**Discussion**

This is the view displayed when a user clicks the tab.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [setView:](#) (page 10)

**Related Sample Code**

CarbonCocoaCoreImageTab

Reducer

**Declared In**

NSTabViewItem.h

## Constants

### NSTabState

These constants describe the current display state of a tab:

```
typedef enum _NSTabState {
    NSSelectedTab = 0,
    NSBackgroundTab = 1,
    NSPressedTab = 2
} NSTabState;
```

#### Constants

NSBackgroundTab

A tab that's not being displayed.

Available in Mac OS X v10.0 and later.

Declared in NSTabViewItem.h.

NSPressedTab

A tab that the user is in the process of clicking. That is, the user has pressed the mouse button while the cursor is over the tab but has not released the mouse button.

Available in Mac OS X v10.0 and later.

Declared in NSTabViewItem.h.

NSSelectedTab

The tab that's being displayed.

Available in Mac OS X v10.0 and later.

Declared in NSTabViewItem.h.

#### Availability

Available in Mac OS X v10.0 and later.

#### Declared In

NSTabViewItem.h



# Document Revision History

---

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

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

**REVISION HISTORY**

Document Revision History



# Index

---

## C

---

color [instance method 7](#)

## D

---

drawLabel:inRect: [instance method 7](#)

## I

---

identifier [instance method 8](#)  
initialFirstResponder [instance method 8](#)  
initWithIdentifier: [instance method 8](#)

## L

---

label [instance method 9](#)

## N

---

NSBackgroundTab [constant 13](#)  
NSPressedTab [constant 13](#)  
NSSelectedTab [constant 13](#)  
NSTabState [data type 13](#)

## S

---

setColor: [instance method 9](#)  
setIdentifier: [instance method 9](#)  
setInitialFirstResponder: [instance method 10](#)  
setLabel: [instance method 10](#)  
setView: [instance method 10](#)  
sizeofLabel: [instance method 11](#)

## T

---

tabState [instance method 11](#)  
tabView [instance method 12](#)

## V

---

view [instance method 12](#)