
Tab Views

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



2003-11-06



Apple Inc.
© 2003 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, and Cocoa 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

Introduction to Tab Views 5

Organization of This Document 5

How Tab Views Work 7

Managing Tab View Items 9

Setting Tab View Appearance 11

Using a Tab View Delegate 13

Document Revision History 15

Introduction to Tab Views

A tab view is a convenient way to provide information in multiple pages. A tab view usually contains a row of tabs that give the visual appearance of folder tabs. When the user clicks on a tab, the tab view displays a view page provided by your application.

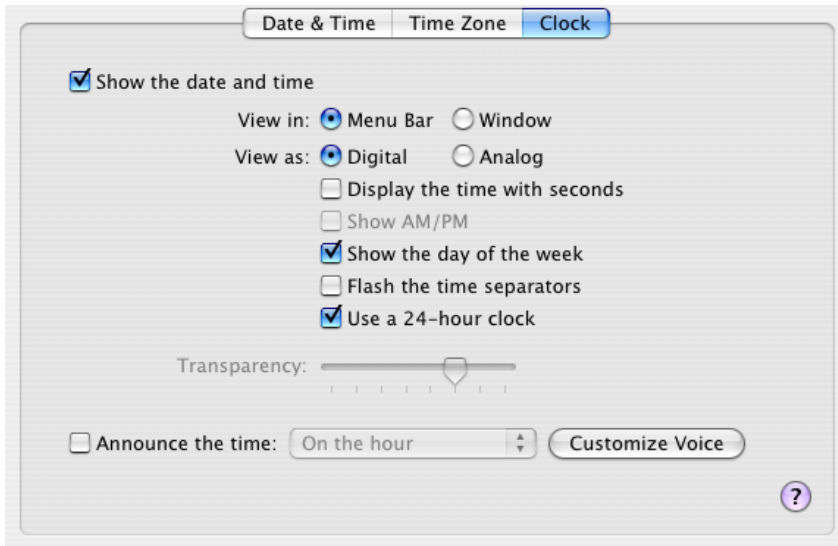
Organization of This Document

This programming topic contains the following articles:

- [“How Tab Views Work”](#) (page 7) describes how tab views work.
- [“Managing Tab View Items”](#) (page 9) describes how to manage a tab view’s list of tab view items.
- [“Setting Tab View Appearance”](#) (page 11) describes how to change a tab view’s appearance.
- [“Using a Tab View Delegate”](#) (page 13) describes how to use a tab view delegate.

How Tab Views Work

An NSTableView provides a convenient mechanism for presenting information in a multi-page format. The view usually contains a row of tabs that give the visual appearance of folder tabs, as shown in the figure below. To select the desired page, the user clicks a tab or uses the arrow keys. Each page displays a view that your application provides.



An NSTableView also supports a multi-page format without visible tabs. For example, instead of tabs, you might use a pop-up menu or radio buttons, similar to those shown in the illustration, to let the user select from several view pages. When a tab view is drawn with tabs (the default), the border must be beveled. When a tab view is drawn without tabs, the view can have a beveled border, a lined border, or no border.

An NSTableView keeps a zero-based array of NSTableViewItems, one per tab in the view. A tab view item provides access to a tab's color, state, label text, initial first responder, and associated view. Your application can supply each tab view item with an optional identifier object to customize tab handling.

Tab label text defaults to the default font and font size used for standard interface items, such as button labels and menu items. When you invoke `setFont:` to change the tab view's font, tab height and width is adjusted automatically to accommodate a new font size. If the view allows truncating, tab labels are truncated as needed.

Managing Tab View Items

These methods let you add and remove tab view items:

- To add a tab view item at the end of the tab view item array, use `addTabViewItem:`
- To insert a tab view item to a specific position in the tab view item array, use `insertTabViewItem:atIndex:`
- To remove a tab view item, use `removeTabViewItem:`

These methods let you access tab view items:

- To return the index of a tab view item, use `indexOfTabViewItem:` or `indexOfTabViewItemWithIdentifier:`
- To return a specific tab view item at a specific index, use `tabViewItemAtIndex:` or `tabViewItemAtPoint:`
- To return an array of tab view items, use `tabViewItems.`
- To return the number of tab view items, use `numberOfTabViewItems.`

These methods select tab view items:

- To select the first or last tab view item in the array, use `selectFirstTabViewItem:` or `selectLastTabViewItem:.`
- To select the tab view item immediately before or after the currently selected item, use `selectPreviousTabViewItem:` or `selectNextTabViewItem:.`
- To select a tab view item at a specific index, use `selectTabViewItemAtIndex:.`
- To select the tab view item at the same index as the selected item in another control, use `takeSelectedTabViewItemFromSender:`
- To return the currently selected tab view item, use `selectedTabViewItem:.`

Setting Tab View Appearance

These methods let you change the appearance of a tab view:

- To change the font used for tab labels, use `setFont:`
- To change the tab view's size, use `setControlSize:` with an argument of either `NSRegularControlSize` or `NSSmallControlSize`.
- To change the tab view's tint, use `setControlTint:` with an argument of either `NSDefaultControlTint` or `NSClearControlTint`.
- To choose whether to allow a tab view to truncate the tab labels, use `setAllowsTruncatedLabels:`.
- To choose the tab view's border style and whether it has visible tabs, use `setTabViewType:` with one of these as arguments:
 - `NSTopTabsBezelBorder`. The view includes tabs and has a beveled border. This is the default.
 - `NSNoTabsBezelBorder`. The view does not include tabs and has a beveled border.
 - `NSNoTabsLineBorder`. The view does not include tabs and has a lined border.
 - `NSNoTabsNoBorder`. The view does not include tabs and has no border.

These methods let you change the appearance of a tab view item:

- To set the tab view item's label, use `setLabel:`.
- To set the tab view item's color, use `setColor:`.

Using a Tab View Delegate

`NSTabView` defines delegate messages to allow the delegate to control or react to changes in selection and changes in the number of tabs:

- `tabViewDidChangeNumberOfTabViewItems`: informs the delegate that the number of tab view items in the tab view has changed.
- `tabView:didSelectTabViewItem`: informs the delegate that the specified tab view item has been selected.
- `tabView:shouldSelectTabViewItem`: informs the delegate that the specified tab view item is about to be selected. The delegate can return `NO` to prevent the selection.
- `tabView:willSelectTabViewItem`: informs the delegate that the specified tab view item will be selected. The delegate can perform tasks related to the selection, but cannot prevent it.

Document Revision History

This table describes the changes to *Tab Views*.

Date	Notes
2003-11-06	Updated art in "How Tab Views Work" (page 7). Reorganized introduction.
2002-11-12	Revision history was added to existing topic. It will be used to record changes to the content of the topic.

