
Status Bars

User Experience: Controls



2002-11-12



Apple Inc.
© 2002 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, Aqua, Mac, Mac OS, and Objective-C are trademarks of Apple Inc., registered in the United States and other countries.

Java and all Java-based trademarks are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. 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 Status Bars 7

Organization of This Document 7

About Status Bars 9

Creating Status Items 11

Document Revision History 13

Figures

About Status Bars 9

Figure 1 System status bar 9

Introduction to Status Bars

Status bars display a collection of status items that provide interaction with or feedback to the user, such as a menu or an image reflecting an application's state. A system-wide status bar resides at the right side of the menu bar and is the only status bar currently available.

Organization of This Document

Use status items sparingly and only if the alternatives (such as a Dock menu, preference pane, or status window) are not suitable. Because there is limited space in the menu bar in which to display status items, status items are not guaranteed to be available at all times. For this reason, do not rely on them being available and always provide a user preference for hiding your application's status items to free up space in the menu bar.

For more information on when to use the status bar, see *Aqua Human Interface Guidelines*.

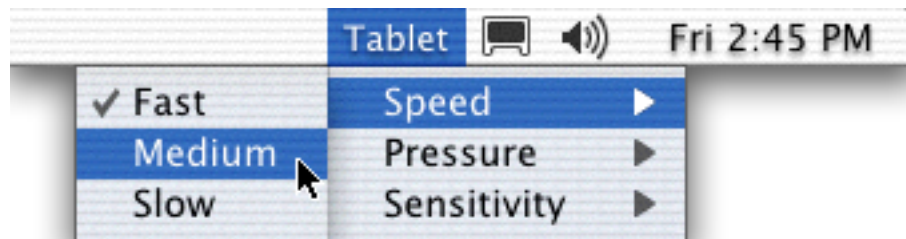
The capabilities of status bars are discussed in [“About Status Bars”](#) (page 9). Sample code and a discussion of how to create status items are in [“Creating Status Items”](#) (page 11).

About Status Bars

Status bars display a collection of status items that provide interaction with or feedback to the user. A status item can be displayed with text or an icon, can provide a menu or send a target-action message when clicked, or can be a fully customized view that you create.

Only one status bar, the system status bar, is currently available. It resides in the system-wide menu bar as shown in [Figure 1](#) (page 9). Status items appear on the right side of the menu bar, just to the left of the menu bar clock and Menu Extras, such as the Displays and Sound menus. The items remain in the menu bar even when your application is not in the foreground.

Figure 1 System status bar



Each new item is added to the left of pre-existing items. When an item is removed, items to the left of it shift to the right to reclaim its space. If the status bar extends into the current application's menu bar, the leftmost status items are hidden to make room for the menus.

Creating Status Items

You obtain the system status bar with the `systemStatusBar` class method; do not allocate an instance yourself. Invoke `statusItemWithLength:` to create a new status item and allocate space for it in the menu bar. Pass the amount of space in pixels you need to display your status item. You can use the constants `NSSquareStatusItemLength` or `NSVariableStatusItemLength` to make the width the same as the status bar's thickness (as returned by the method `thickness`) or variable based on the contents of the item, respectively. Use the former if you are displaying an icon and the latter if you are displaying static text.

Because the system status bar is shared by all applications, it cannot retain references to each application's status item objects. Instead, each application is responsible for retaining its own status items. (This is automatically done in Java.) Each status item then communicates with the status bar as its configuration changes. When deallocated, the status item removes itself from the status bar. Because the object returned by `statusItemWithLength:` is autoreleased, you must explicitly retain it if using Objective-C; otherwise it is automatically removed from the status bar when it is deallocated by the run loop's autorelease pool.

Once you have the new status item object, you can assign it a title, a menu, a target-action, a tool tip, and so on.

In the following example, a status item is added to the menu bar and assigned a menu.

```
- (void)activateStatusMenu
{
    NSStatusBar *bar = [NSStatusBar systemStatusBar];

    theItem = [bar statusItemWithLength:NSVariableStatusItemLength];
    [theItem retain];

    [theItem setTitle: NSLocalizedString(@"Tablet",@"")];
    [theItem setHighlightMode:YES];
    [theItem setMenu:theMenu];
}
```

In Java, the equivalent code is this:

```
public void activateStatusMenu() {
    NSStatusBar bar = NSStatusBar.systemStatusBar();
    theItem = bar.statusItem(NSStatusBar.VariableStatusItemLength);
    theItem.setTitle( NSBundle.localizedString("Tablet") );
    theItem.setHighlightMode(true);
    theItem.setMenu(theMenu);
}
```

In this code example, the object has an instance variable, `theMenu`, that holds an `NSMenu` object, perhaps unarchived from a nib file. Another instance variable, `theItem`, holds the status item, which is retained in the Objective-C case. When you execute this code, a menu titled "Tablet" (or a localized version if available) is added to the right side of the menu bar. The menu is available from within every application as long as your application is running and the status item exists.

Document Revision History

This table describes the changes to *Status Bars*.

Date	Notes
2002-11-12	Revision history was added to existing topic. It will be used to record changes to the content of the topic.

