

---

# Slider Programming Topics

User Experience: Controls



2006-10-03



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, 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

---

## **Introduction to Sliders 7**

Organization of This Document 7

---

## **About Sliders 9**

Setting a Slider's Values 9

Displaying a Slider's Name and Values 9

---

## **Using Slider Tick Marks 11**

---

## **Setting the Shape Of a Slider 13**

---

## **Document Revision History 15**

---



# Figures and Tables

## Using Slider Tick Marks 11

---

Figure 1	Horizontal bar slider with tick marks	11
Figure 2	Horizontal bar slider with no tick marks	11
Table 1	Arguments to add tick marks to horizontal sliders	11
Table 2	Arguments to add tick marks to vertical sliders	11

## Setting the Shape Of a Slider 13

---

Figure 1	Circular Sliders	13
----------	------------------	----



# Introduction to Sliders

---

A slider is a user interface element that displays a range of values and has an indicator, or knob, which indicates the current setting.

## Organization of This Document

[“About Sliders”](#) (page 9) gives basic information on sliders.

[“Using Slider Tick Marks”](#) (page 11) describes how to set and control the appearance of the tick marks on a slider.

[“Setting the Shape Of a Slider”](#) (page 13) describes how to specify whether your slider is a horizontal bar, vertical bar, or circular dial.





# About Sliders

---

A slider is a control that displays a range of values and has an indicator, or knob, which indicates the current setting. Optionally, it can have tick marks at regularly spaced intervals. A slider can be a bar or a dial. The user moves the knob along the slider's bar, or around the dial, to change the setting.

**Note:** Don't use an `NSSlider` object when an `NSScroller` would be better. A scroller represents the relative position of the visible portion of a view and lets the user choose which portion to view. A slider represents a range of values for something in the application and lets the user choose a setting.

A slider's behavior and appearance is controlled by the class `NSSliderCell`. An `NSSliderCell` object must be contained within a subclass of `NSControl`. If you need one slider, use an `NSSlider` object that contains a single `NSSliderCell` object. If you need a group of related sliders, use an `NSMatrix` object that contains several `NSSliderCell` objects. Most `NSSliderCell` methods have covers in `NSSlider`, which simply call the `NSSliderCell` equivalent. For more information, see the `NSSliderCell` and `NSSlider` class specifications.

**Note:** For a description of the relationship between controls (`NSControl` objects) and cells (`NSCell` objects), see the section on control-cell architecture in "The Core Application Architecture" of *Cocoa Fundamentals Guide*.

## Setting a Slider's Values

To set the slider's minimum and maximum values, use `setMinValue:` and `setMaxValue:`. To read the slider's value, as represented by the knob's current position, use an `NSControl` "get" method, such as `floatValue`; To set the slider's value, use an `NSControl` "set" method, such as `setFloatValue:`.

## Displaying a Slider's Name and Values

You should use text fields to display the slider's name, minimum value, maximum value, and tick-mark value. Although you can set a title for a slider that's displayed automatically, it's displayed right in the slider's bar and is easily obscured by the knob. There is no way for a slider to display automatically its minimum, maximum, and tick-mark values.



# Using Slider Tick Marks

To set the number of tick marks, use `setNumberOfTickMarks :`. The tick marks are evenly spaced between the minimum and maximum values. For example, if the minimum value is 0, the maximum value is 100, and the tick mark count is five, the tick marks are at 0, 25, 50, 75, and 100. The following figure shows a horizontal bar slider with five tick marks.

**Figure 1** Horizontal bar slider with tick marks




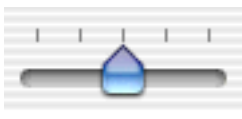
The following figure shows a slider with no tick marks.

**Figure 2** Horizontal bar slider with no tick marks



To set where the tick marks appear, use `setTickMarkPosition :`. For horizontal sliders, the possible arguments are shown below.



**Table 1** Arguments to add tick marks to horizontal sliders

	Tick marks below	Tick marks above
<b>Argument</b>	<code>NSTickMarkBelow</code>	<code>NSTickMarkAbove</code>
<b>Illustration</b>		

For vertical sliders, the arguments are shown below.

**Table 2** Arguments to add tick marks to vertical sliders

	Tick marks below	Tick marks above
<b>Argument</b>	<code>NSTickMarkLeft</code>	<code>NSTickMarkRight</code>

	Tick marks below	Tick marks above
<b>Illustration</b>		

The default values are `NSTickMarkBelow` and `NSTickMarkLeft`. These arguments are used only with bar sliders; for circular sliders, the tickmarks are always outside the circle.

To restrict a slider's value to only the values at tick marks, use `setAllowsTickMarkValuesOnly:`. After a user moves the slider's knob, the knob jumps to the tick mark nearest the cursor. For example, if a slider is restricted to tick mark values only and has a minimum value of 0, a maximum value of 100, and a marker count of five, the allowable values are 0, 25, 50, 75, and 100. By default, a slider can have any value between its minimum and maximum.

To get the value of the tick mark that's closest to another value, use `closestTickMarkValueToValue:`. To get the value that corresponds to a specific tick mark, use `tickMarkValueAtIndex:`. To find the tick mark closest to a specific point, use `indexOfTickMarkAtPoint:`. Note that the lowest tick mark has an index of 0.

# Setting the Shape Of a Slider

---

Sliders come in three shapes: horizontal bar, vertical bar, and circular dial. To determine whether a slider is a bar or a dial, use `setSliderType:`. For a dial, set the slider type to `NSCircularSlider` (that is, call `[[slider cell] setSliderType:NSCircularSlider];`). If you are using Interface Builder, you can drag a circular slider widget into your design window, or select the circular slider checkbox in the slider's info window.

The following figure shows circular dial sliders.

**Figure 1**      Circular Sliders



A bar-shaped slider automatically determines whether it's horizontal or vertical by the shape of its containing rectangle. If the slider is wider than it is tall, it's horizontal. Otherwise, it's vertical. Use the `initWithFrame:` method to initialize a slider, passing in an `NSRect` with the size and shape you want.

For a circular slider, you must pass an `NSRect` at least large enough to contain the control. For a regular circular slider, the `NSRect` must be at least 28 by 30 pixels. For a small circular slider, it must be at least 18 by 20 pixels. Add 4 pixels in each dimension if your slider has tick marks.

If you specify a larger `NSRect`, the slider will be centered in the rectangle. If you specify a smaller rectangle, the control will be clipped.

A vertical slider has its minimum on the bottom; a horizontal slider has its minimum on the left; a circular slider has its minimum at the top.



# Document Revision History

---

This table describes the changes to *Slider Programming Topics*.

Date	Notes
2006-10-03	Provided a cross-reference to the description of control-cell architecture.
2003-07-29	Added information about circular sliders.
2002-11-12	Revision history was added to existing topic.

