

---

# NSTextField Class Reference

User Experience: Controls



2009-07-02



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, Cocoa, Mac, Mac OS, and Quartz 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

## **NSTextField Class Reference 5**

---

Overview	5
Tasks	5
Controlling Editability and Selectability	5
Controlling Rich Text Behavior	6
Setting the Text Color	6
Controlling the Background	6
Setting a Border	6
Selecting the Text	7
Working with the Responder Chain	7
Using Keyboard Interface Control	7
Setting the Delegate	7
NSText Delegate Method Implementations	7
Instance Methods	8
acceptsFirstResponder	8
allowsEditingTextAttributes	8
backgroundColor	9
bezelStyle	9
delegate	9
drawsBackground	10
importsGraphics	10
isBezeled	10
isBordered	11
isEditable	11
isSelectable	12
nextText	12
previousText	12
selectText:	13
setAllowsEditingTextAttributes:	13
setBackgroundColor:	13
setBezeled:	14
setBezelStyle:	14
setBordered:	15
setDelegate:	15
setDrawsBackground:	15
setEditable:	16
setImportsGraphics:	17
setNextText:	17
setPreviousText:	17
setSelectable:	18
setTextColor:	18

setTitleWithMnemonic: 19  
textColor 19  
textDidBeginEditing: 19  
textDidChange: 20  
textDidEndEditing: 20  
textShouldBeginEditing: 21  
textShouldEndEditing: 22

---

**Document Revision History 23**

---

**Index 25**

---

# NSTextField Class Reference

---

<b>Inherits from</b>	NSControl : NSView : NSResponder : NSObject
<b>Conforms to</b>	NSUserInterfaceValidations 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>	Text Fields
<b>Declared in</b>	NSTextField.h
<b>Related sample code</b>	EnhancedAudioBurn EnhancedDataBurn FunHouse ImageClient OpenALExample

## Overview

An `NSTextField` object is a kind of `NSControl` that displays text that the user can select or edit and that sends its action message to its target when the user presses the Return key while editing.

The `NSTextField` class uses the `NSTextFieldCell` class to implement its user interface.

## Tasks

### Controlling Editability and Selectability

- [setEditable:](#) (page 16)  
Controls whether the user can edit the receiver's text.
- [isEditable](#) (page 11)  
Returns a Boolean value indicating whether the user is allowed to select and edit the receiver's text.
- [setSelectable:](#) (page 18)  
Sets whether the receiver is selectable (but not editable).

- [isSelectable](#) (page 12)  
Returns a Boolean value indicating whether the user is allowed to select the receiver's text.

## Controlling Rich Text Behavior

- [setAllowsEditingTextAttributes:](#) (page 13)  
Controls whether the receiver allows the user to change font attributes of the receiver's text.
- [allowsEditingTextAttributes](#) (page 8)  
Returns a Boolean value indicating whether the user is allowed to change font attributes of the receiver's text.
- [setImportsGraphics:](#) (page 17)  
Controls whether the receiver allows the user to drag image files into it.
- [importsGraphics](#) (page 10)  
Returns a Boolean value indicating whether the receiver allows the user to drag image files into it.

## Setting the Text Color

- [setTextColor:](#) (page 18)  
Sets the color used to draw the receiver's text.
- [textColor](#) (page 19)  
Returns the color used to draw the receiver's text.

## Controlling the Background

- [setBackgroundcolor:](#) (page 13)  
Sets the color of the background that the receiver's cell draws behind the text.
- [backgroundcolor](#) (page 9)  
Returns the color of the background that the receiver's cell draws behind the text.
- [setDrawsBackground:](#) (page 15)  
Controls whether the receiver's cell draws its background color behind its text.
- [drawsBackground](#) (page 10)  
Returns a Boolean value indicating whether the receiver's cell draws its background color.

## Setting a Border

- [setBezeled:](#) (page 14)  
Controls whether the receiver draws a beveled border around its contents.
- [isBezeled](#) (page 10)  
Returns a Boolean value indicating whether the receiver draws a beveled frame.
- [setBezelStyle:](#) (page 14)  
Sets the receiver's bezel style.
- [bezelStyle](#) (page 9)  
Returns the receiver's bezel style.

- [setBordered:](#) (page 15)  
Controls whether the receiver draws a solid black border around its contents.
- [isBordered](#) (page 11)  
Returns a Boolean value indicating whether the receiver draws a black border around its contents.

## Selecting the Text

- [selectText:](#) (page 13)  
Ends editing and selects the entire contents of the receiver if it's selectable.

## Working with the Responder Chain

- [acceptsFirstResponder](#) (page 8)  
Returns a Boolean value indicating whether the receiver is editable.
- [nextText](#) (page 12)  
Returns the object selected when the user presses Tab. (**Deprecated.** Use the `NSView` method `nextKeyView` instead.)
- [previousText](#) (page 12)  
Returns the object selected when the user presses Shift-Tab. (**Deprecated.** Use the `NSView` method `previousKeyView` instead.)
- [setNextText:](#) (page 17)  
Sets the object selected when the user presses Tab. (**Deprecated.** Use the `NSView` method `setNextKeyView:` instead.)
- [setPreviousText:](#) (page 17)  
Sets the object selected when the user presses Shift-Tab. (**Deprecated.** Use the `NSView` method `setNextKeyView:` instead.)

## Using Keyboard Interface Control

- [setTitleWithMnemonic:](#) (page 19)  
Sets the receiver's string value, using the embedded character as the keyboard mnemonic.

## Setting the Delegate

- [setDelegate:](#) (page 15)  
Sets the receiver's delegate.
- [delegate](#) (page 9)  
Returns the receiver's delegate.

## NSText Delegate Method Implementations

- [textShouldBeginEditing:](#) (page 21)  
Requests permission to begin editing a text object.

- [textDidBeginEditing:](#) (page 19)  
Posts a notification that the text is about to begin editing to the default notification center.
- [textDidChange:](#) (page 20)  
Posts a notification that the text has changed and forwards this message to the receiver's cell if it responds.
- [textShouldEndEditing:](#) (page 22)  
Performs validation on the receiver's new value.
- [textDidEndEditing:](#) (page 20)  
Handles an end of editing.

## Instance Methods

### acceptsFirstResponder

Returns a Boolean value indicating whether the receiver is editable.

- (BOOL)acceptsFirstResponder

#### Return Value

YES if the receiver is editable, NO otherwise.

#### Availability

Available in Mac OS X v10.0 and later.

#### Declared In

NSTextField.h

### allowsEditingTextAttributes

Returns a Boolean value indicating whether the user is allowed to change font attributes of the receiver's text.

- (BOOL)allowsEditingTextAttributes

#### Return Value

YES if the receiver allows the user to change font attributes of the receiver's text, otherwise NO. You can change text attributes programmatically regardless of this setting.

#### Availability

Available in Mac OS X v10.0 and later.

#### See Also

- [importsGraphics](#) (page 10)
- [setAllowsEditingTextAttributes:](#) (page 13)

#### Declared In

NSTextField.h

## backgroundColor

Returns the color of the background that the receiver's cell draws behind the text.

- (NSColor \*)backgroundColor

### Return Value

The color used to draw the background.

### Availability

Available in Mac OS X v10.0 and later.

### See Also

- [drawsBackground](#) (page 10)
- [setBackground-color:](#) (page 13)

### Declared In

NSTextField.h

## bezelStyle

Returns the receiver's bezel style.

- (NSTextFieldBezelStyle)bezelStyle

### Return Value

A constant indicating the bezel style. Possible values are described in the [Bezel Styles](#) section of NSTextFieldCell Class Reference.

### Availability

Available in Mac OS X v10.2 and later.

### See Also

- [setBezelStyle:](#) (page 14)

### Declared In

NSTextField.h

## delegate

Returns the receiver's delegate.

- (id < NSTextFieldDelegate >)delegate

### Return Value

The object that acts as the receiver's delegate.

### Availability

Available in Mac OS X v10.0 and later.

### See Also

- [setDelegate:](#) (page 15)

**Declared In**

NSTextField.h

**drawsBackground**

Returns a Boolean value indicating whether the receiver's cell draws its background color.

- (BOOL)drawsBackground

**Return Value**

YES if the receiver's cell draws its background color behind its text, NO if it draws no background.

**Discussion**

In order to prevent inconsistent rendering, background color rendering is disabled for rounded-bezel text fields.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [backgroundColor](#) (page 9)
- [drawsBackground](#) (NSTextFieldCell)
- [setDrawsBackground:](#) (page 15)

**Declared In**

NSTextField.h

**importsGraphics**

Returns a Boolean value indicating whether the receiver allows the user to drag image files into it.

- (BOOL)importsGraphics

**Return Value**

YES if the receiver allows the user to drag image files into it, otherwise NO. You can add images programmatically regardless of this setting.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [allowsEditingTextAttributes](#) (page 8)
- [importsGraphics](#) (NSTextView)
- [setImportsGraphics:](#) (page 17)

**Declared In**

NSTextField.h

**isBezeled**

Returns a Boolean value indicating whether the receiver draws a bezeled frame.

- (BOOL)isBezeled

**Return Value**

YES if the receiver draws a bezeled frame around its contents; otherwise, NO.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [isBordered](#) (page 11)
- [setBezeled:](#) (page 14)

**Declared In**

NSTextField.h

## isBordered

Returns a Boolean value indicating whether the receiver draws a black border around its contents.

- (BOOL)isBordered

**Return Value**

YES if the receiver draws a solid black border around its contents; otherwise NO.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [isBezeled](#) (page 10)
- [setBordered:](#) (page 15)

**Declared In**

NSTextField.h

## isEditable

Returns a Boolean value indicating whether the user is allowed to select and edit the receiver's text.

- (BOOL)isEditable

**Return Value**

YES if the user is allowed to select and edit the receiver's text, NO if the user isn't allowed to edit it (though the user may be able to select it).

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [isSelectable](#) (page 12)
- [setEditable:](#) (page 16)

**Declared In**

NSTextField.h

## isSelectable

Returns a Boolean value indicating whether the user is allowed to select the receiver's text.

- (BOOL)isSelectable

### Return Value

YES if the user is allowed to select the receiver's text; otherwise NO.

### Discussion

Selectable text isn't necessarily editable; use [isEditable](#) (page 11) to check for editability.

### Availability

Available in Mac OS X v10.0 and later.

### See Also

- [setSelectable:](#) (page 18)

### Declared In

NSTextField.h

## nextText

Returns the object selected when the user presses Tab. (**Deprecated.** Use the `NSView` method `nextKeyView` instead.)

- (id)nextText

### Return Value

The object to be selected.

### Availability

Available in Mac OS X v10.0 through Mac OS X v10.1.

### Declared In

NSTextField.h

## previousText

Returns the object selected when the user presses Shift-Tab. (**Deprecated.** Use the `NSView` method `previousKeyView` instead.)

- (id)previousText

### Return Value

The object to be selected.

### Availability

Available in Mac OS X v10.0 through Mac OS X v10.1.

### Declared In

NSTextField.h

## selectText:

Ends editing and selects the entire contents of the receiver if it's selectable.

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

### Parameters

*sender*

The sender of the message.

### Discussion

If the receiver isn't in some window's view hierarchy, this method has no effect.

### Availability

Available in Mac OS X v10.0 and later.

### See Also

- [isSelected](#) (page 12)

### Related Sample Code

QTMetadataEditor

TipWrapper

### Declared In

NSTextField.h

## setAllowsEditingTextAttributes:

Controls whether the receiver allows the user to change font attributes of the receiver's text.

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

### Parameters

*flag*

If YES, the user is permitted to change font attributes of the receiver's text; if *flag* is NO, the user isn't so permitted. You can change text attributes programmatically regardless of this setting.

### Availability

Available in Mac OS X v10.0 and later.

### See Also

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

- [allowsEditingTextAttributes](#) (page 8)

### Declared In

NSTextField.h

## setBackground-color:

Sets the color of the background that the receiver's cell draws behind the text.

```
- (void)setBackgroundColor:(NSColor *)aColor
```

**Parameters***aColor*

The color used to draw the background.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [setDrawsBackground:](#) (page 15)
- [backgroundColor](#) (page 9)

**Declared In**

NSTextField.h

**setBezeled:**

Controls whether the receiver draws a bezeled border around its contents.

- (void)setBezeled:(BOOL)*flag***Parameters***flag*If YES, it draws a bezeled border and invokes [setDrawsBackground:](#) (page 15) with an argument of NO; if NO, the receiver does not draw a border.**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [isBezeled](#) (page 10)
- [setBordered:](#) (page 15)

**Related Sample Code**

FunHouse

Quartz Composer QCTV

Vertex Optimization

**Declared In**

NSTextField.h

**setBezelStyle:**

Sets the receiver's bezel style.

- (void)setBezelStyle:(NSTextFieldBezelStyle)*style***Parameters***style*A constant indicating the bezel style. Possible values for *style* are described in the [Bezel Styles](#) section of NSTextFieldCell Class Reference. You must have already sent the receiver [setBezeled:](#) (page 14) with an argument of YES

### Availability

Available in Mac OS X v10.2 and later.

### See Also

- [bezelStyle](#) (page 9)

### Declared In

NSTextField.h

## setBordered:

Controls whether the receiver draws a solid black border around its contents.

- (void)setBordered:(BOOL)flag

### Parameters

*flag*

If YES, the receiver draws a border; if NO, it draws no border.

### Availability

Available in Mac OS X v10.0 and later.

### See Also

- [isBordered](#) (page 11)

- [setBeveled:](#) (page 14)

### Declared In

NSTextField.h

## setDelegate:

Sets the receiver's delegate.

- (void)setDelegate:(id < NSTextFieldDelegate >)anObject

### Parameters

*anObject*

The object that acts as the receiver's delegate.

### Availability

Available in Mac OS X v10.0 and later.

### See Also

- [delegate](#) (page 9)

### Declared In

NSTextField.h

## setDrawsBackground:

Controls whether the receiver's cell draws its background color behind its text.

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

#### Parameters

*flag*

If YES, the receiver's cell draws its background; if NO, it draws nothing behind its text.

#### Discussion

In order to prevent inconsistent rendering, background color rendering is disabled for rounded-bezel text fields.

#### Availability

Available in Mac OS X v10.0 and later.

#### See Also

- [setBackground-color:](#) (page 13)
- [setDrawsBackground:](#) (NSTextFieldCell)
- [drawsBackground](#) (page 10)

#### Related Sample Code

FunHouse

Quartz Composer QCTV

#### Declared In

NSTextField.h

## setEditable:

Controls whether the user can edit the receiver's text.

- (void)setEditable:(BOOL)*flag*

#### Parameters

*flag*

If YES, then the user is allowed to both select and edit text. If *flag* is NO, then the user isn't permitted to edit text, and the receiver's selectability is restored to its previous value.

#### Discussion

For example, if an NSTextField object is selectable but not editable, then made editable for a time, then made not editable, it remains selectable. To guarantee that text is neither editable nor selectable, simply use [setSelectable:](#) (page 18) to turn off selectability.

#### Availability

Available in Mac OS X v10.0 and later.

#### See Also

- [isEditable](#) (page 11)

#### Related Sample Code

FinalCutPro\_AppleEvents

FunHouse

SharedMemory

UIElementInspector

With and Without Bindings

**Declared In**

NSTextField.h

**setImportsGraphics:**

Controls whether the receiver allows the user to drag image files into it.

- (void)setImportsGraphics:(BOOL)flag

**Parameters***flag*

If YES, the receiver accepts dragged images; if NO, it doesn't. You can add images programmatically regardless of this setting.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [setAllowsEditingTextAttributes:](#) (page 13)
- [setImportsGraphics:](#) (NSTextView)
- [importsGraphics](#) (page 10)

**Declared In**

NSTextField.h

**setNextText:**Sets the object selected when the user presses Tab. (**Deprecated.** Use the `NSView` method `setNextKeyView:` instead.)

- (void)setNextText:(id)anObject

**Parameters***anObject*

The object to be selected.

**Availability**

Available in Mac OS X v10.0 through Mac OS X v10.1.

**Declared In**

NSTextField.h

**setPreviousText:**Sets the object selected when the user presses Shift-Tab. (**Deprecated.** Use the `NSView` method `setNextKeyView:` instead.)

- (void)setPreviousText:(id)anObject

**Parameters***anObject*

The object to be selected.

**Availability**

Available in Mac OS X v10.0 through Mac OS X v10.1.

**Declared In**

NSTextField.h

**setSelectable:**

Sets whether the receiver is selectable (but not editable).

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

**Parameters**

*flag*

If YES, the receiver is made selectable but not editable (use [setEditable:](#) (page 16) to make text both selectable and editable). If NO, the text is neither editable nor selectable.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [setEditable:](#) (page 16)

**Related Sample Code**

Quartz Composer QCTV

With and Without Bindings

**Declared In**

NSTextField.h

**setTextColor:**

Sets the color used to draw the receiver's text.

```
- (void)setTextColor:(NSColor *)aColor
```

**Parameters**

*aColor*

The color used to draw text.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [setBackground-color:](#) (page 13)

- [setText-color:](#) (NSTextFieldCell)

- [text-color](#) (page 19)

**Related Sample Code**

IdentitySample

Vertex Optimization

**Declared In**

NSTextField.h

**setTitleWithMnemonic:**

Sets the receiver's string value, using the embedded character as the keyboard mnemonic.

```
- (void)setTitleWithMnemonic:(NSString *)aString
```

**Parameters**

*aString*

The string to set as the title. The first character preceded by an ampersand ('&') is used as the mnemonic (the first ampersand character is stripped out).

**Discussion**

Mnemonics are not supported in Mac OS X.

**Availability**

Available in Mac OS X v10.0 and later.

**Declared In**

NSTextField.h

**textColor**

Returns the color used to draw the receiver's text.

```
- (NSColor *)textColor
```

**Return Value**

The color used to draw text.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [backgroundColor](#) (page 9)
- [textColor](#) (NSTextFieldCell)
- [setTextColor:](#) (page 18)

**Declared In**

NSTextField.h

**textDidBeginEditing:**

Posts a notification that the text is about to begin editing to the default notification center.

```
- (void)textDidBeginEditing:(NSNotification *)aNotification
```

**Parameters**

*aNotification*

The `NSNotification` notification to post.

**Discussion**

This action causes the receiver's delegate to receive a `controlTextDidBeginEditing:` message. See the `NSControl` class specification for more information on the text delegate method.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [textDidBeginEditing:](#) (page 19)
- [textDidChange:](#) (page 20)
- [textShouldEndEditing:](#) (page 22)
- [textDidEndEditing:](#) (page 20)

**Declared In**

`NSTextField.h`

**textDidChange:**

Posts a notification that the text has changed and forwards this message to the receiver's cell if it responds.

```
- (void)textDidChange:(NSNotification *)aNotification
```

**Parameters**

*aNotification*

The `NSControlTextDidChangeNotification` notification that is posted to the default notification center.

**Discussion**

This method causes the receiver's delegate to receive a `controlTextDidChange:` message. See the `NSControl` class specification for more information on the text delegate method.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [textShouldBeginEditing:](#) (page 21)
- [textDidBeginEditing:](#) (page 19)
- [textShouldEndEditing:](#) (page 22)
- [textDidEndEditing:](#) (page 20)

**Declared In**

`NSTextField.h`

**textDidEndEditing:**

Handles an end of editing.

```
- (void)textDidEndEditing:(NSNotification *)aNotification
```

**Parameters**

*aNotification*

The notification that editing has ended.

**Discussion**

After validating the new value, posts an `NSControlTextDidEndEditingNotification` to the default notification center. This posting causes the receiver's delegate to receive a `controlTextDidEndEditing:` message. After this message, sends `endEditing:` to the receiver's cell and handles the key that caused editing to end:

- If the user ended editing by pressing Return, this method tries to send the receiver's action to its target; if unsuccessful, it sends `performKeyEquivalent:` to its `NSView` (for example, to handle the default button on a panel); if that also fails, the receiver simply selects its text.
- If the user ended editing by pressing Tab or Shift-Tab, the receiver tries to have its `NSWindow` object select its next or previous key view, using the `NSWindow` method `selectKeyViewFollowingView:` or `selectKeyViewPrecedingView:`. If unsuccessful in doing this, the receiver simply selects its text.

See the `NSControl` class specification for more information on the text delegate method.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [textShouldBeginEditing:](#) (page 21)
- [textDidBeginEditing:](#) (page 19)
- [textDidChange:](#) (page 20)
- [textShouldEndEditing:](#) (page 22)

**Declared In**

`NSTextField.h`

**textShouldBeginEditing:**

Requests permission to begin editing a text object.

```
- (BOOL)textShouldBeginEditing:(NSText *)textObject
```

**Parameters**

*textObject*

The object to begin editing.

**Return Value**

YES if editing should be allowed to occur, NO otherwise.

**Discussion**

If the receiver isn't editable, returns NO immediately. If it is editable and its delegate responds to `control:textShouldBeginEditing:`, this invokes that method and returns the result. Otherwise it simply returns YES to allow editing to occur. See the `NSControl` class specification for more information on the text delegate method.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [textDidBeginEditing:](#) (page 19)
- [textDidChange:](#) (page 20)

- [textShouldEndEditing:](#) (page 22)
- [textDidEndEditing:](#) (page 20)

**Declared In**

NSTextField.h

**textShouldEndEditing:**

Performs validation on the receiver's new value.

- (BOOL)textShouldEndEditing:(NSText \*)*textObject*

**Parameters**

*textObject*

The text object requesting permission to end editing.

**Return Value**

YES if the new value is valid; otherwise NO.

**Discussion**

This method validates the receiver's new value using the `NSCell` method `isEntryAcceptable:`. If the new value is valid and the delegate responds to `control:textShouldEndEditing:`, invokes that method and returns the result, in addition beeping if the delegate returns NO. See the `NSControl` class specification for more information on the text delegate method.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [textShouldBeginEditing:](#) (page 21)
- [textDidBeginEditing:](#) (page 19)
- [textDidChange:](#) (page 20)
- [textDidEndEditing:](#) (page 20)

**Declared In**

NSTextField.h

# Document Revision History

---

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

Date	Notes
2009-07-02	Added descriptions of obsolete methods removed from header files.
2009-05-24	Updated for Mac OS X v10.6. Delegate methods moved to NSTextFieldDelegate Protocol Reference.
2009-01-06	Added note that background color rendering is disabled for rounded-bezel text fields.
2007-10-31	Corrected minor errors. Clarified the NSText delegate method group heading.
	Removed mention of error action message from <code>textShouldEndEditing</code> .
2006-05-23	First publication of this content as a separate document.

## REVISION HISTORY

### Document Revision History

# Index

---

## A

---

acceptsFirstResponder [instance method 8](#)  
allowsEditingTextAttributes [instance method 8](#)

## B

---

backgroundColor [instance method 9](#)  
bezelStyle [instance method 9](#)

## D

---

delegate [instance method 9](#)  
drawsBackground [instance method 10](#)

## I

---

importsGraphics [instance method 10](#)  
isBezeled [instance method 10](#)  
isBordered [instance method 11](#)  
isEditable [instance method 11](#)  
isSelectable [instance method 12](#)

## N

---

nextText [instance method 12](#)

## P

---

previousText [instance method 12](#)

## S

---

selectText: [instance method 13](#)  
setAllowsEditingTextAttributes: [instance method 13](#)  
setBackgroundColor: [instance method 13](#)  
setBezeled: [instance method 14](#)  
setBezelStyle: [instance method 14](#)  
setBordered: [instance method 15](#)  
setDelegate: [instance method 15](#)  
setDrawsBackground: [instance method 15](#)  
setEditable: [instance method 16](#)  
setImportsGraphics: [instance method 17](#)  
setNextText: [instance method 17](#)  
setPreviousText: [instance method 17](#)  
setSelectable: [instance method 18](#)  
setTextColor: [instance method 18](#)  
setTitleWithMnemonic: [instance method 19](#)

## T

---

textColor [instance method 19](#)  
textDidBeginEditing: [instance method 19](#)  
textDidChange: [instance method 20](#)  
textDidEndEditing: [instance method 20](#)  
textShouldBeginEditing: [instance method 21](#)  
textShouldEndEditing: [instance method 22](#)