
WebEditingDelegate Protocol Reference

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



2009-04-08



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 Objective-C 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

WebEditingDelegate Protocol Reference 5

Overview	5
Tasks	5
Controlling Editing Behavior	5
Responding to Notifications	6
Performing Commands	6
Getting the Undo Manager	6
Instance Methods	6
undoManagerForWebView:	6
webView:doCommandBySelector:	6
webView:shouldApplyStyle:toElementsInDOMRange:	7
webView:shouldBeginEditingInDOMRange:	7
webView:shouldChangeSelectedDOMRange:toDOMRange:affinity:stillSelecting:	8
webView:shouldChangeTypingStyle:toStyle:	8
webView:shouldDeleteDOMRange:	8
webView:shouldEndEditingInDOMRange:	9
webView:shouldInsertNode:replacingDOMRange:givenAction:	9
webView:shouldInsertText:replacingDOMRange:givenAction:	10
webViewDidBeginEditing:	10
webViewDidChange:	10
webViewDidChangeSelection:	11
webViewDidChangeTypingStyle:	11
webViewDidEndEditing:	12
Constants	12
WebViewInsertAction	12

Document Revision History 15

Index 17

WebEditingDelegate Protocol Reference

(informal protocol)

Framework	/System/Library/Frameworks/WebKit.framework
Availability	Available in Mac OS X v10.3.9 and later.
Companion guide	WebKit Objective-C Programming Guide
Declared in	WebEditingDelegate.h

Overview

You use a `WebEditingDelegate` to control or augment the editing behavior of a `WebView` object. Objects conforming to the `WebEditingDelegate` informal protocol may receive *should* messages before or *did* messages after an editing action. Typically, you implement an editing delegate if you want to change the default editing behavior.

Tasks

Controlling Editing Behavior

- [webView:shouldApplyStyle:toElementsInDOMRange:](#) (page 7)
Returns whether the user should be allowed to apply a style to a range of content.
- [webView:shouldBeginEditingInDOMRange:](#) (page 7)
Returns whether the user is allowed to edit a range of content in a web view.
- [webView:shouldChangeSelectedDOMRange:toDOMRange:affinity:stillSelecting:](#) (page 8)
Returns whether the user should be allowed to change the selected range.
- [webView:shouldChangeTypingStyle:toStyle:](#) (page 8)
Returns whether the user should be allowed to change the typing style in a web view.
- [webView:shouldDeleteDOMRange:](#) (page 8)
Returns whether the user should be allowed to delete a range of content.
- [webView:shouldEndEditingInDOMRange:](#) (page 9)
Returns whether the user should be allowed to end editing.
- [webView:shouldInsertNode:replacingDOMRange:givenAction:](#) (page 9)
Returns whether the user should be allowed to insert a node in place of a range of content.
- [webView:shouldInsertText:replacingDOMRange:givenAction:](#) (page 10)
Returns whether a user should be allowed to insert text in place of a range of content.

Responding to Notifications

- [webViewDidBeginEditing:](#) (page 10)
Sent by the default notification center when the user begins editing the web view.
- [webViewDidChange:](#) (page 10)
Sent by the default notification center when the user changes content in the web view.
- [webViewDidChangeSelection:](#) (page 11)
Sent by the default notification center when the user changes the selection in the web view.
- [webViewDidChangeTypingStyle:](#) (page 11)
Sent by the default notification center when the user changes the typing style in the web view.
- [webViewDidEndEditing:](#) (page 12)
Sent by the default notification center when the user stops editing the web view.

Performing Commands

- [webView:doCommandBySelector:](#) (page 6)
Returns whether the receiver will perform a command instead of the webView.

Getting the Undo Manager

- [undoManagerForWebView:](#) (page 6)
Returns the undo manager to be used by a webView.

Instance Methods

undoManagerForWebView:

Returns the undo manager to be used by a webView.

```
- (NSUndoManager *)undoManagerForWebView:(WebView *)webView
```

Availability

Available in Mac OS X v10.3.9 and later.

Declared In

WebEditingDelegate.h

webView:doCommandBySelector:

Returns whether the receiver will perform a command instead of the webView.

```
- (BOOL)webView:(WebView *)webView doCommandBySelector:(SEL)command
```

Discussion

This method returns YES if the receiver will perform *command*, NO otherwise. Implement this method if you want to perform *command* instead of letting the *webView* perform *command*.

Availability

Available in Mac OS X v10.3.9 and later.

Declared In

WebEditingDelegate.h

webView:shouldApplyStyle:toElementsInDOMRange:

Returns whether the user should be allowed to apply a style to a range of content.

```
- (BOOL)webView:(WebView *)webView shouldApplyStyle:(DOMCSSStyleDeclaration *)style
toElementsInDOMRange:(DOMRange *)range
```

Discussion

This method returns YES if the user should be allowed to apply a style, specified by *style*, to the content, specified by *range*, in *webView*, NO otherwise.

Availability

Available in Mac OS X v10.3.9 and later.

Declared In

WebEditingDelegate.h

webView:shouldBeginEditingInDOMRange:

Returns whether the user is allowed to edit a range of content in a web view.

```
- (BOOL)webView:(WebView *)webView shouldBeginEditingInDOMRange:(DOMRange *)range
```

Discussion

This method returns YES if the user is allowed to edit *webView*, NO otherwise. This method is invoked when a web view attempts to become the first responder or when the user drops an object on *webView*. The *range* argument marks the section of the begin-editing request and is used to determine if editing is allowed. Typically, *range* is not the current selection but may become the current selection if this method returns YES.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- [webView:shouldEndEditingInDOMRange:](#) (page 9)
- [webViewDidBeginEditing:](#) (page 10)

Declared In

WebEditingDelegate.h

webView:shouldChangeSelectedDOMRange:toDOMRange:affinity:stillSelecting:

Returns whether the user should be allowed to change the selected range.

```
- (BOOL)webView:(WebView *)webView shouldChangeSelectedDOMRange:(DOMRange *)currentRange toDOMRange:(DOMRange *)proposedRange affinity:(NSSelectionAffinity)selectionAffinity stillSelecting:(BOOL)flag
```

Discussion

This method returns YES if the user is allowed to change the selected range specified by the *currentRange* and *proposedRange* arguments; otherwise, NO. The *selectionAffinity* argument specifies the direction of the selection. The *flag* argument is YES if the user is still selecting, NO otherwise.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

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

Declared In

WebEditingDelegate.h

webView:shouldChangeTypingStyle:toStyle:

Returns whether the user should be allowed to change the typing style in a web view.

```
- (BOOL)webView:(WebView *)webView shouldChangeTypingStyle:(DOMCSSStyleDeclaration *)currentStyle toStyle:(DOMCSSStyleDeclaration *)proposedStyle
```

Discussion

This method returns YES if the user should be allowed to change the typing style in *webView* to *proposedStyle*, NO otherwise. The *currentStyle* argument is the old style. You can implement this method to take some other action—for example, set the typing style to a different style—and return NO.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

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

Declared In

WebEditingDelegate.h

webView:shouldDeleteDOMRange:

Returns whether the user should be allowed to delete a range of content.

```
- (BOOL)webView:(WebView *)webView shouldDeleteDOMRange:(DOMRange *)range
```

Discussion

This method should returns YES if the user should be allowed to delete the content specified by *range*, NO otherwise. This method may perform an alternate action—for example, delete a different range—and return NO.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

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

Declared In

WebEditingDelegate.h

webView:shouldEndEditingInDOMRange:

Returns whether the user should be allowed to end editing.

```
- (BOOL)webView:(WebView *)webView shouldEndEditingInDOMRange:(DOMRange *)range
```

Discussion

This method returns YES if the user should be allowed to end editing *webView*, NO otherwise. This method is invoked when a web view attempts to resign as the first responder. Typically, the *range* argument designates the current selection, although it might not. Use the *range* argument to help determine if the user can end editing. If this method returns YES, *webView* will end editing and resign as the first responder.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- [webView:shouldBeginEditingInDOMRange:](#) (page 7)
- [webViewDidEndEditing:](#) (page 12)

Declared In

WebEditingDelegate.h

webView:shouldInsertNode:replacingDOMRange:givenAction:

Returns whether the user should be allowed to insert a node in place of a range of content.

```
- (BOOL)webView:(WebView *)webView shouldInsertNode:(DOMNode *)node
    replacingDOMRange:(DOMRange *)range givenAction:(WebViewInsertAction)action
```

Discussion

This method returns YES if the user should be allowed to insert *node* in *webView*, NO otherwise. The *range* argument is the portion of the content that will be replaced with *node*. The *action* argument indicates the type of user action that initiated the insertion. This method may perform an alternate action—for example, insert a different node—and return NO.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- [webView:shouldInsertText:replacingDOMRange:givenAction:](#) (page 10)
- [webViewDidChange:](#) (page 10)

Declared In

WebEditingDelegate.h

webView:shouldInsertText:replacingDOMRange:givenAction:

Returns whether a user should be allowed to insert text in place of a range of content.

```
- (BOOL)webView:(WebView *)webView shouldInsertText:(NSString *)text
    replacingDOMRange:(DOMRange *)range givenAction:(WebViewInsertAction)action
```

Discussion

This method returns YES if the user should be allowed to insert *text* in *webView*, NO otherwise. This method is invoked when one of the `replaceSelectionWith...` messages is sent to a *webView*. The *range* argument is the portion of the document that will be replaced with *text*. The *action* argument indicates the type of user action that initiated the insertion. This method may perform an alternate action—for example, insert different text—and return NO.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- [webView:shouldInsertNode:replacingDOMRange:givenAction:](#) (page 9)
- [webViewDidChange:](#) (page 10)

Declared In

WebEditingDelegate.h

webViewDidBeginEditing:

Sent by the default notification center when the user begins editing the web view.

```
- (void)webViewDidBeginEditing:(NSNotification *)notification
```

Discussion

The *notification* argument is always `WebViewDidBeginEditingNotification`. You can retrieve the `WebView` object by sending `object` to *notification*.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- [webViewDidEndEditing:](#) (page 12)
- [webView:shouldBeginEditingInDOMRange:](#) (page 7)

Declared In

WebEditingDelegate.h

webViewDidChange:

Sent by the default notification center when the user changes content in the web view.

```
- (void)webViewDidChange:(NSNotification *)notification
```

Discussion

The *notification* argument is always `WebViewDidChangeNotification`. You can retrieve the `WebView` object by sending `object` to *notification*.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- [webView:shouldDeleteDOMRange:](#) (page 8)
- [webView:shouldInsertNode:replacingDOMRange:givenAction:](#) (page 9)
- [webView:shouldInsertText:replacingDOMRange:givenAction:](#) (page 10)
- [webView:shouldDeleteDOMRange:](#) (page 8)

Declared In

WebEditingDelegate.h

webViewDidChangeSelection:

Sent by the default notification center when the user changes the selection in the web view.

- (void)webViewDidChangeSelection:(NSNotification *)*notification*

Discussion

The *notification* argument is always `WebViewDidChangeSelectionNotification`. You can retrieve the `WebView` object by sending `object` to *notification*.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- [webView:shouldChangeSelectedDOMRange:toDOMRange:affinity:stillSelecting:](#) (page 8)

Declared In

WebEditingDelegate.h

webViewDidChangeTypingStyle:

Sent by the default notification center when the user changes the typing style in the web view.

- (void)webViewDidChangeTypingStyle:(NSNotification *)*notification*

Discussion

The *notification* argument is always `WebViewDidChangeTypingStyleNotification`. You can retrieve the `WebView` object by sending `object` to *notification*.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- [webView:shouldChangeTypingStyle:toStyle:](#) (page 8)

Declared In

WebEditingDelegate.h

webViewDidEndEditing:

Sent by the default notification center when the user stops editing the web view.

```
- (void)webViewDidEndEditing:(NSNotification *)notification
```

Discussion

The *notification* argument is always `WebViewDidEndEditingNotification`. You can retrieve the `WebView` object by sending `object` to *notification*.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- [webViewDidBeginEditing:](#) (page 10)
- [webView:shouldEndEditingInDOMRange:](#) (page 9)

Declared In

WebEditingDelegate.h

Constants

WebViewInsertAction

Indicate the type of user action that initiated a delegate message.

```
typedef enum {
    WebViewInsertActionTyped,
    WebViewInsertActionPasted,
    WebViewInsertActionDropped,
} WebViewInsertAction;
```

Constants

`WebViewInsertActionTyped`

Indicates the user inserted content by typing.

Available in Mac OS X v10.3 and later.

Declared in `WebEditingDelegate.h`.

`WebViewInsertActionPasted`

Indicates the user inserted content by pasting.

Available in Mac OS X v10.3 and later.

Declared in `WebEditingDelegate.h`.

`WebViewInsertActionDropped`

Indicates the user inserted content by dropping.

Available in Mac OS X v10.3 and later.

Declared in `WebEditingDelegate.h`.

Discussion

These constants are described in `WebEditingDelegate`.

Availability

Available in Mac OS X v10.3.9 and later.

Document Revision History

This table describes the changes to *WebEditingDelegate Protocol Reference*.

Date	Notes
2009-04-08	Minor edits throughout.
2008-10-15	Minor edits throughout.
2006-05-23	First publication of this content as a separate document.

REVISION HISTORY

Document Revision History

Index

U

undoManagerForWebView: <NSObject> instance
method [6](#)

W

webView:doCommandBySelector: <NSObject> instance
method [6](#)

webView:shouldApplyStyle:toElementsInDOMRange:
<NSObject> instance method [7](#)

webView:shouldBeginEditingInDOMRange:
<NSObject> instance method [7](#)

webView:shouldChangeSelectedDOMRange:toDOMRange:
affinity:stillSelecting: <NSObject> instance
method [8](#)

webView:shouldChangeTypingStyle:toStyle:
<NSObject> instance method [8](#)

webView:shouldDeleteDOMRange: <NSObject>
instance method [8](#)

webView:shouldEndEditingInDOMRange: <NSObject>
instance method [9](#)

webView:shouldInsertNode:replacingDOMRange:
givenAction: <NSObject> instance method [9](#)

webView:shouldInsertText:replacingDOMRange:
givenAction: <NSObject> instance method [10](#)

webViewDidBeginEditing: <NSObject> instance
method [10](#)

webViewDidChange: <NSObject> instance method [10](#)

webViewDidChangeSelection: <NSObject> instance
method [11](#)

webViewDidChangeTypingStyle: <NSObject> instance
method [11](#)

webViewDidEndEditing: <NSObject> instance method
[12](#)

WebViewInsertAction [12](#)

WebViewInsertActionDropped constant [12](#)

WebViewInsertActionPasted constant [12](#)

WebViewInsertActionTyped constant [12](#)