
WebView Class 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, Carbon, Cocoa, iChat, Mac, Mac OS, Objective-C, and Safari are trademarks of Apple Inc., registered in the United States and other countries.

WebScript is a trademark of Apple Inc.

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

WebView Class Reference 9

Overview 9

Tasks 10

Registering Document Views and Representations 10

Initializing Views 10

Closing the View 11

Getting the Main Frame 11

Loading Content 11

Background Drawing 11

Moving Back and Forward 11

Changing the Text Size 12

Getting and Setting Delegates 12

Getting and Setting the Window 13

Getting and Setting Preferences 13

Getting and Setting Frame Contents 13

Getting and Setting Content Information 13

Searching the Document 14

Getting and Setting the Group Name 14

Getting and Setting User-agent Strings 14

Processing JavaScript 15

Using the Pasteboard 15

Dragging 15

Cut, Copy and Paste Action Methods 15

Content Alignment Action Methods 16

Changing the Font, Color and Other Attributes When Editing 16

Spell-checking Action Methods 16

Find Panel Action Method 16

Controlling Speakable Text 17

Getting and Setting Document Editing Attributes 17

Editing Documents 17

Selecting Content in the Document 18

Getting and Setting CSS Properties 18

Using WebScript 19

Class Methods 19

canShowMIMETYPE: 19

canShowMIMETYPEAsHTML: 19

MIMETypesShownAsHTML 20

registerURLSchemeAsLocal: 20

registerViewClass:representationClass:forMIMETYPE: 20

setMIMETypesShownAsHTML: 21

URLFromPasteboard: 21

- URLTitleFromPasteboard: 22
- Instance Methods 22
 - alignCenter: 22
 - alignJustified: 23
 - alignLeft: 23
 - alignRight: 24
 - applicationNameForUserAgent 24
 - applyStyle: 24
 - backForwardList 25
 - canGoBack 25
 - canGoForward 26
 - canMakeTextLarger 26
 - canMakeTextSmaller 27
 - canMakeTextStandardSize 27
 - changeAttributes: 27
 - changeColor: 28
 - changeDocumentBackgroundColor: 28
 - changeFont: 28
 - checkSpelling: 29
 - close 29
 - computedStyleForElement:pseudoElement: 30
 - copy: 30
 - copyFont: 31
 - customTextEncodingName 31
 - customUserAgent 31
 - cut: 32
 - delete: 32
 - deleteSelection 33
 - downloadDelegate 33
 - drawsBackground 34
 - editableDOMRangeForPoint: 34
 - editingDelegate 34
 - elementAtPoint: 35
 - estimatedProgress 35
 - frameLoadDelegate 36
 - goBack 36
 - goBack: 36
 - goForward 37
 - goForward: 37
 - goToBackForwardItem: 38
 - groupName 38
 - hostWindow 38
 - initWithFrame:frameName:groupName: 39
 - isContinuousSpellCheckingEnabled 39
 - isEditable 40
 - isLoading 40

mainFrame 40
mainFrameDocument 41
mainFrameIcon 41
mainFrameTitle 41
mainFrameURL 42
maintainsInactiveSelection 42
makeTextLarger: 42
makeTextSmaller: 43
makeTextStandardSize: 43
mediaStyle 43
moveDragCaretToPoint: 44
moveToBeginningOfSentence: 44
moveToBeginningOfSentenceAndModifySelection: 44
moveToEndOfSentence: 45
moveToEndOfSentenceAndModifySelection: 45
paste: 45
pasteAsPlainText: 46
pasteAsRichText: 46
pasteboardTypesForElement: 47
pasteboardTypesForSelection 47
pasteFont: 47
performFindPanelAction: 48
policyDelegate 48
preferences 48
preferencesIdentifier 49
reload: 49
removeDragCaret 50
replaceSelectionWithArchive: 50
replaceSelectionWithMarkupString: 50
replaceSelectionWithNode: 51
replaceSelectionWithText: 52
resourceLoadDelegate 52
searchFor:direction:caseSensitive:wrap: 52
selectedDOMRange 53
selectedFrame 54
selectionAffinity 54
selectSentence: 54
setApplicationNameForUserAgent: 55
setContinuousSpellCheckingEnabled: 55
setCustomTextEncodingName: 55
setCustomUserAgent: 56
setDownloadDelegate: 56
setDrawsBackground: 57
setEditable: 57
setEditingDelegate: 58
 setFrameLoadDelegate: 58

setGroupName:	58
setHostWindow:	59
setMainFrameURL:	59
setMaintainsBackForwardList:	60
setMediaStyle:	60
setPolicyDelegate:	61
setPreferences:	61
setPreferencesIdentifier:	61
setResourceLoadDelegate:	62
setSelectedDOMRange:affinity:	62
setShouldCloseWithWindow:	63
setSmartInsertDeleteEnabled:	63
setTextSizeMultiplier:	64
setTypingStyle:	64
setUIDelegate:	64
shouldCloseWithWindow	65
showGuessPanel:	65
smartInsertDeleteEnabled	65
spellCheckerDocumentTag	66
startSpeaking:	66
stopLoading:	67
stopSpeaking:	67
stringByEvaluatingJavaScriptFromString:	67
styleDeclarationWithText:	68
supportsTextEncoding	68
takeStringURLFrom:	69
textSizeMultiplier	69
toggleContinuousSpellChecking:	69
toggleSmartInsertDelete:	70
typingStyle	70
UIDelegate	70
undoManager	71
userAgentForURL:	71
windowScriptObject	71
writeElement:withPasteboardTypes:toPasteboard:	72
writeSelectionWithPasteboardTypes:toPasteboard:	72
Constants	73
Notifications	74
WebViewDidBeginEditingNotification	74
WebViewDidChangeNotification	74
WebViewDidChangeSelectionNotification	74
WebViewDidChangeTypingStyleNotification	74
WebViewDidEndEditingNotification	75
WebViewProgressEstimateChangedNotification	75
WebViewProgressFinishedNotification	75
WebViewProgressStartedNotification	75

Document Revision History 77

Index 79

WebView Class Reference

Inherits from	NSView : NSResponder : NSObject
Conforms to	NSAnimatablePropertyContainer (NSView) NSCoding (NSResponder) NSObject (NSObject)
Framework	/System/Library/Frameworks/WebKit.framework
Availability	Available in Mac OS X v10.2 with Safari 1.0 and later. Available in Mac OS X v10.2.7 and later.
Companion guide	WebKit Objective-C Programming Guide
Declared in	WebView.h
Related sample code	CallJS CarbonCocoaCoreImageTab HIView-NSView NewsReader SpecialPictureProtocol

Overview

`WebView` is the core view class in the WebKit framework that manages interactions between the `WebFrame` and `WebFrameView` classes. To embed web content in your application, you just create a `WebView` object, attach it to a window, and send a `loadRequest:` message to its main frame.

Behind the scenes, `WebFrame` objects encapsulate the content contained in a single frame element. A hierarchy of `WebFrame` objects is used to model an entire webpage where the root is called the **main frame**. There is a `WebFrameView` object per `WebFrame` object used to display the frame content. Therefore, there is a parallel hierarchy of `WebFrameView` objects used to render an entire page. The `WebView` object is also the parent view of this hierarchy. You do not need to create `WebFrame` and `WebFrameView` objects directly. These objects are automatically created when the page loads, either programmatically or by the user clicking a link.

You customize your embedded web content by implementing `WebView` delegates to handle certain aspects of the process. `WebView` objects have multiple delegates because the process of loading a webpage is asynchronous and complicated if errors occur. All the `WebView` delegates use informal protocols so you only need to implement only the delegates and methods that define the behavior you wish to change—default implementations are already provided.

For example, you might want to implement the frame load and resource load delegates to monitor the load progress and display status messages. Applications that use multiple windows may want to implement a user interface delegate. See the individual informal delegate protocols for more details: *WebFrameLoadDelegate Protocol Reference*, *WebPolicyDelegate Protocol Reference*, *WebResourceLoadDelegate Protocol Reference*, and *WebUIDelegate Protocol Reference*.

Another way to monitor load progress with less control is to observe the [WebViewProgressEstimateChangedNotification](#) (page 75), [WebViewProgressFinishedNotification](#) (page 75), and [WebViewProgressStartedNotification](#) (page 75) notifications. For example, you could observe these notifications to implement a simple progress indicator in your application. You update the progress indicator by invoking the `estimatedProgress` method to get an estimate of the amount of content that is currently loaded.

A `WebView` object is intended to support most features you would expect in a web browser except that it doesn't implement the specific user interface for those features. You are responsible for implementing the user interface objects such as status bars, toolbars, buttons, and text fields. For example, a `WebView` object manages a back-forward list by default, and has [goBack:](#) (page 36) and [goForward:](#) (page 37) action methods. It is your responsibility to create the buttons that would send these action messages. Note, there is some overhead in maintaining a back-forward list and page cache, so you should disable it if your application doesn't use it.

You use a `WebPreferences` object to encapsulate the preferences of a `WebView` object, such as the font, text encoding, and image settings. You can modify the preferences for individual `WebView` objects or specify a shared `WebPreferences` object using the [setPreferencesIdentifier:](#) (page 61) method. Use the `setAutosaves:WebPreferences` method to specify whether the preferences should be automatically saved to the user defaults database.

You can also extend `WebKit` by implementing your own document view and representation classes for specific MIME types. Use the [registerViewClass:representationClass:forMIMEType:](#) (page 20) class method to register your custom classes with a `WebView` object.

Tasks

Registering Document Views and Representations

- + [registerURLSchemeAsLocal:](#) (page 20)
Adds the specified URL scheme to the list of local schemes.
- + [registerViewClass:representationClass:forMIMEType:](#) (page 20)
Specifies the view and representation objects to be used for specific MIME types.

Initializing Views

- [initWithFrame:frameName:groupName:](#) (page 39)
Initializes the receiver with a frame rectangle, frame name, and group name.

Closing the View

- [close](#) (page 29)
Closes the web view when it's no longer needed.
- [shouldCloseWithWindow](#) (page 65)
Returns whether the web view should close when its window or host window closes.
- [setShouldCloseWithWindow:](#) (page 63)
Sets whether the web view should close when its window or host window closes.

Getting the Main Frame

- [mainFrame](#) (page 40)
Returns the main frame, the root of the web frame hierarchy for this page.

Loading Content

- [stopLoading:](#) (page 67)
An action method that stops the loading of any web frame content managed by the receiver.
- [takeStringURLFrom:](#) (page 69)
Sets the receiver's current location by obtaining a URL string from the sender.
- [reload:](#) (page 49)
Reloads the current page.
- [estimatedProgress](#) (page 35)
Returns an estimate, as a percentage, of the amount of content that is currently loaded.

Background Drawing

- [drawsBackground](#) (page 34)
Returns whether the web view draws a background.
- [setDrawsBackground:](#) (page 57)
Sets whether a default background is drawn when the webpage has no background set.

Moving Back and Forward

- [setMaintainsBackForwardList:](#) (page 60)
Sets whether to use a back-forward list.
- [backForwardList](#) (page 25)
Returns the receiver's back-forward list.
- [canGoBack](#) (page 25)
Returns whether the previous location can be loaded.
- [goBack](#) (page 36)
Loads the previous location in the back-forward list.

- [goBack:](#) (page 36)
An action method that loads the previous location in the back-forward list.
- [canGoForward](#) (page 26)
Returns whether the next location can be loaded.
- [goForward](#) (page 37)
Loads the next location in the back-forward list.
- [goForward:](#) (page 37)
An action method that loads the next location in the back-forward list.
- [goToBackForwardItem:](#) (page 38)
Loads a specific location from the back-forward list and sets it as the current item.

Changing the Text Size

- [canMakeTextLarger](#) (page 26)
Returns whether the text can be made larger.
- [makeTextLarger:](#) (page 42)
Increases the text size by one unit.
- [canMakeTextSmaller](#) (page 27)
Returns whether the text can be made smaller.
- [makeTextSmaller:](#) (page 43)
Reduces the text size by one unit.

Getting and Setting Delegates

- [downloadDelegate](#) (page 33)
Return the receiver's download delegate.
- [setDownloadDelegate:](#) (page 56)
Sets the receiver's shared download delegate.
- [frameLoadDelegate](#) (page 36)
Return the receiver's frame load delegate.
- [setFrameLoadDelegate:](#) (page 58)
Sets the receiver's frame load delegate.
- [policyDelegate](#) (page 48)
Returns the receiver's policy delegate.
- [setPolicyDelegate:](#) (page 61)
Sets the receiver's policy delegate.
- [resourceLoadDelegate](#) (page 52)
Returns the receiver's resource load delegate.
- [setResourceLoadDelegate:](#) (page 62)
Sets the receiver's resource load delegate.
- [UIDelegate](#) (page 70)
Returns the receiver's user interface delegate.

- [setUIDelegate:](#) (page 64)
Sets the receiver's user interface delegate.

Getting and Setting the Window

- [hostWindow](#) (page 38)
Returns the receiver's host window.
- [setHostWindow:](#) (page 59)
Sets the receiver's host window.

Getting and Setting Preferences

- [preferences](#) (page 48)
Returns the receiver's preferences.
- [setPreferences:](#) (page 61)
Sets the receiver's preferences.
- [preferencesIdentifier](#) (page 49)
Returns the identifier of the receiver's preferences.
- [setPreferencesIdentifier:](#) (page 61)
Sets the receiver's preferences identifier, creating a preferences object if needed.

Getting and Setting Frame Contents

- [isLoading](#) (page 40)
Returns whether the web view is loading content.
- [selectedFrame](#) (page 54)
Returns the frame with the active selection.
- [setMainFrameURL:](#) (page 59)
Sets the URL that the main frame loads.
- [mainFrameURL](#) (page 42)
Returns the URL that the main frame loads.
- [mainFrameTitle](#) (page 41)
Returns the HTML title of the loaded page.
- [mainFrameIcon](#) (page 41)
Returns the site's favicon.
- [mainFrameDocument](#) (page 41)
Returns the DOM document for the main frame.

Getting and Setting Content Information

- + [canShowMIMETYPE:](#) (page 19)
Returns whether the receiver can display content of a given MIME type.

- + [MIMETypesShownAsHTML](#) (page 20)
Returns a list of MIME types that WebKit renders as HTML.
- + [setMIMETypesShownAsHTML:](#) (page 21)
Sets the MIME types that WebKit attempts to render as HTML.
- + [canShowMIMETYPEAsHTML:](#) (page 19)
Returns whether the receiver interprets a MIME type as HTML.
- [supportsTextEncoding](#) (page 68)
Returns whether the document view supports different text encodings.
- [customTextEncodingName](#) (page 31)
Returns the custom text encoding name.
- [setCustomTextEncodingName:](#) (page 55)
Sets the custom text encoding name.
- [textSizeMultiplier](#) (page 69)
Returns the font size multiplier for text displayed in web frame view objects managed by the receiver.
- [setTextSizeMultiplier:](#) (page 64)
Change the font size multiplier for text displayed in web frame view objects managed by the receiver.

Searching the Document

- [searchFor:direction:caseSensitive:wrap:](#) (page 52)
Searches a document view for a string and highlights it if it is found.

Getting and Setting the Group Name

- [groupName](#) (page 38)
Returns the receiver's group name.
- [setGroupName:](#) (page 58)
Sets the receiver's group name.

Getting and Setting User-agent Strings

- [userAgentForURL:](#) (page 71)
Returns the appropriate user-agent string for a given URL.
- [applicationNameForUserAgent](#) (page 24)
Returns the receiver's application name that is used in the user-agent string.
- [setApplicationNameForUserAgent:](#) (page 55)
Sets the application name used in the user-agent string.
- [customUserAgent](#) (page 31)
Returns the receiver's custom user-agent string.
- [setCustomUserAgent:](#) (page 56)
Sets the receiver's custom user-agent string.

Processing JavaScript

- [stringByEvaluatingJavaScriptFromString:](#) (page 67)
Returns the result of running a script.

Using the Pasteboard

- + [URLFromPasteboard:](#) (page 21)
Returns a URL from the specified pasteboard.
- + [URLTitleFromPasteboard:](#) (page 22)
Returns the title of a URL from the specified pasteboard.
- [pasteboardTypesForElement:](#) (page 47)
Returns an array of pasteboard types for an element.
- [pasteboardTypesForSelection](#) (page 47)
Returns an array of pasteboard types that can be used for the current selection of the receiver.
- [writeElement:withPasteboardTypes:toPasteboard:](#) (page 72)
Writes an element to the pasteboard using a list of types.
- [writeSelectionWithPasteboardTypes:toPasteboard:](#) (page 72)
Writes the receiver's current selection to a pasteboard using a list of types.

Dragging

- [elementAtPoint:](#) (page 35)
Returns a dictionary description of the element at a given point in the receiver's coordinates.
- [moveDragCaretToPoint:](#) (page 44)
Moves the drag caret that indicates the destination of a drag operation to a given point.
- [removeDragCaret](#) (page 50)
Removes the drag caret that indicates the destination of a drag operation.

Cut, Copy and Paste Action Methods

- [copy:](#) (page 30)
Action method that copies the selected content to the general pasteboard.
- [copyFont:](#) (page 31)
An action method that copies font information onto the font pasteboard.
- [cut:](#) (page 32)
An action method that deletes selected content and puts it on the general pasteboard.
- [delete:](#) (page 32)
An action method that deletes the selected content.
- [paste:](#) (page 45)
An action method that pastes content from the pasteboard at the insertion point or over the selection.
- [pasteFont:](#) (page 47)
An action method that pastes font information from the font pasteboard.

- [pasteAsPlainText:](#) (page 46)
An action method that pastes pasteboard content as plain text.
- [pasteAsRichText:](#) (page 46)
An action method that pastes pasteboard content into the receiver as rich text, maintaining its attributes.

Content Alignment Action Methods

- [alignCenter:](#) (page 22)
An action method that applies center alignment to selected content or all content if there's no selection.
- [alignJustified:](#) (page 23)
An action method that applies full justification to selected content or all content if there's no selection.
- [alignLeft:](#) (page 23)
An action method that applies left justification to selected content or all content if there's no selection.
- [alignRight:](#) (page 24)
An action method that applies right justification to selected content or all content if there is no selection.

Changing the Font, Color and Other Attributes When Editing

- [changeFont:](#) (page 28)
An action method that changes the font of the selection, or all content if there is no selection.
- [changeAttributes:](#) (page 27)
An action method that changes the attributes of the current selection.
- [changeDocumentBackgroundColor:](#) (page 28)
Sets the background color of the selected content.
- [changeColor:](#) (page 28)
Sets the color of the selected content.

Spell-checking Action Methods

- [checkSpelling:](#) (page 29)
An action method that searches for a misspelled word in the receiver.
- [showGuessPanel:](#) (page 65)
An action method that shows a spelling correction panel.

Find Panel Action Method

- [performFindPanelAction:](#) (page 48)
An action method that opens the Find menu and Find panel.

Controlling Speakable Text

- [startSpeaking:](#) (page 66)
An action method that starts speaking the selected text or all text if there's no selection.
- [stopSpeaking:](#) (page 67)
An action method that stops speaking that is in progress.

Getting and Setting Document Editing Attributes

- [isEditable](#) (page 40)
Returns whether the user is allowed to edit the document.
- [setEditable:](#) (page 57)
Sets whether the receiver allows the user to edit its HTML document.
- [smartInsertDeleteEnabled](#) (page 65)
Returns whether smart-space insertion and deletion is enabled.
- [setSmartInsertDeleteEnabled:](#) (page 63)
Sets whether the receiver should insert or delete spaces around selected words to preserve proper spacing and punctuation.
- [isContinuousSpellCheckingEnabled](#) (page 39)
Returns whether the web view has continuous spell-checking enabled.
- [setContinuousSpellCheckingEnabled:](#) (page 55)
Sets whether the web view has continuous spell-checking enabled.
- [spellCheckerDocumentTag](#) (page 66)
Returns the spell-checker document tag for this document.
- [undoManager](#) (page 71)
Returns the receiver's undo manager.
- [editingDelegate](#) (page 34)
Returns the receiver's editing delegate.
- [setEditingDelegate:](#) (page 58)
Sets the receiver's editing delegate.
- [editableDOMRangeForPoint:](#) (page 34)
Returns the editable DOM object located at a given point.

Editing Documents

- [replaceSelectionWithNode:](#) (page 51)
Replaces the receiver's current selection with the specified DOM node.
- [replaceSelectionWithText:](#) (page 52)
Replaces the current selection with a string of text.
- [replaceSelectionWithMarkupString:](#) (page 50)
Replaces the current selection with mixed text and markup.
- [replaceSelectionWithArchive:](#) (page 50)
Replaces the current selection with an archive's contents.

- [deleteSelection](#) (page 33)
Deletes the receiver's current selection unless it's collapsed.
- [moveToBeginningOfSentence:](#) (page 44)
Moves the insertion point to the beginning of the current sentence.
- [moveToBeginningOfSentenceAndModifySelection:](#) (page 44)
Moves the insertion point and extends the selection to the beginning of the current sentence.
- [moveToEndOfSentence:](#) (page 45)
Moves the insertion point to the end of the current sentence.
- [moveToEndOfSentenceAndModifySelection:](#) (page 45)
Moves the insertion point and extends the selection to the end of the current sentence.
- [selectSentence:](#) (page 54)
Selects the entire sentence around the insertion point.
- [toggleContinuousSpellChecking:](#) (page 69)
Toggles whether continuous spell checking is available.
- [toggleSmartInsertDelete:](#) (page 70)
Toggles whether spaces around selected words are inserted or deleted to preserve proper spacing and punctuation.
- [canMakeTextStandardSize](#) (page 27)
Returns whether the current text size is a multiple of 1.
- [makeTextStandardSize:](#) (page 43)
Resets the text size to a multiple of 1.
- [maintainsInactiveSelection](#) (page 42)
Returns whether the selection is maintained when focus is lost.

Selecting Content in the Document

- [selectedDOMRange](#) (page 53)
Returns the range of the current selection.
- [setSelectedDOMRange:affinity:](#) (page 62)
Selects a range of nodes.
- [selectionAffinity](#) (page 54)
Returns the current selection affinity.

Getting and Setting CSS Properties

- [computedStyleForElement:pseudoElement:](#) (page 30)
Returns the computed style of an element and its pseudo element.
- [mediaStyle](#) (page 43)
Returns the receiver's CSS media property.
- [setMediaStyle:](#) (page 60)
Sets the receiver's CSS media property.
- [typingStyle](#) (page 70)
Returns the receiver's CSS typing style.

- [setTypingStyle:](#) (page 64)
Sets the receiver's CSS typing style.
- [styleDeclarationWithText:](#) (page 68)
Returns the CSS style declaration for the specified text.
- [applyStyle:](#) (page 24)
Applies the CSS typing style to the current selection.

Using WebScript

- [windowScriptObject](#) (page 71)
Returns the receiver's window object from the scripting environment.

Class Methods

canShowMIMEType:

Returns whether the receiver can display content of a given MIME type.

```
+ (BOOL)canShowMIMEType:(NSString *)MIMEType
```

Parameters

MIMEType

The MIME type of the content.

Return Value

YES if the receiver can display content of the specified MIME type where *MIMEType* is one of the standard types like "image/gif"; otherwise, NO.

Availability

Available in Mac OS X v10.2 with Safari 1.0 and later.

Available in Mac OS X v10.2.7 and later.

Declared In

WebView.h

canShowMIMETypeAsHTML:

Returns whether the receiver interprets a MIME type as HTML.

```
+ (BOOL)canShowMIMETypeAsHTML:(NSString *)MIMEType
```

Parameters

MIMEType

The MIME type of the content.

Return Value

YES if the receiver interprets *MIMEType* as HTML; otherwise, NO.

Availability

Available in Mac OS X v10.2 with Safari 1.0 and later.

Available in Mac OS X v10.2.7 and later.

Declared In

WebView.h

MIMETypesShownAsHTML

Returns a list of MIME types that WebKit renders as HTML.

```
+ (NSArray *)MIMETypesShownAsHTML
```

Return Value

An array containing `NSString` objects that represent the MIME types WebKit attempts to render as HTML.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

+ [setMIMETypesShownAsHTML:](#) (page 21)

Declared In

WebView.h

registerURLSchemeAsLocal:

Adds the specified URL scheme to the list of local schemes.

```
+ (void)registerURLSchemeAsLocal:(NSString *)scheme
```

Parameters

scheme

The scheme to add to the list.

Discussion

You need to register a scheme as local to access resources with file URLs and to have the same security checks as a local file.

Availability

Available in Mac OS X v10.4.11 and later.

Declared In

WebView.h

registerViewClass:representationClass:forMIMEType:

Specifies the view and representation objects to be used for specific MIME types.

```
+ (void)registerViewClass:(Class)viewClassrepresentationClass:(Class)representationClassforMIMEType:(NSString *)MIMEType
```

Parameters*viewController*

A class conforming to the `WebDocumentView` protocol that displays the specified MIME types.

representationClass

The class conforming to `WebDocumentRepresentation` protocol that represents the specified MIME types.

MIMETYPE

The MIME type of the content.

This may be a primary MIME type or subtype. For example, if *MIMETYPE* is “video/” the specified view and representation objects are used for all video types. More specific subtype mappings, such as “image/gif”, takes precedence over primary type matching, such as “image/”.

Discussion

After invoking this method, when *MIMETYPE* content is encountered, instances of *representationClass* and *viewController* are created to handle and display it.

Availability

Available in Mac OS X v10.2 with Safari 1.0 and later.

Available in Mac OS X v10.2.7 and later.

Declared In

WebView.h

setMIMETypesShownAsHTML:

Sets the MIME types that WebKit attempts to render as HTML.

```
+ (void)setMIMETypesShownAsHTML:(NSArray *)MIMETypes
```

Parameters*MIMETypes*

An array of `NSString` objects representing the MIME types. Typically, you create the *MIMETypes* array by adding additional types to the array returned by the [MIMETypesShownAsHTML](#) (page 20) class method.

Availability

Available in Mac OS X v10.3.9 and later.

Declared In

WebView.h

URLFromPasteboard:

Returns a URL from the specified pasteboard.

```
+ (NSURL *)URLFromPasteboard:(NSPasteboard *)pasteboard
```

Parameters*pasteboard*

The pasteboard containing a URL.

Return Value

The URL from the specified pasteboard or `nil` if there's no URL on *pasteboard*.

Discussion

This method supports multiple pasteboard types including `NSRULPboardType`.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

+ [URLTitleFromPasteboard:](#) (page 22)

Declared In

WebView.h

URLTitleFromPasteboard:

Returns the title of a URL from the specified pasteboard.

```
+ (NSString *)URLTitleFromPasteboard:(NSPasteboard *)pasteboard
```

Parameters

pasteboard

The pasteboard containing the URL.

Return Value

The title of the URL on *pasteboard*. Returns `nil` if there's no URL on *pasteboard* or the URL has no title.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

+ [URLFromPasteboard:](#) (page 21)

Declared In

WebView.h

Instance Methods

alignCenter:

An action method that applies center alignment to selected content or all content if there's no selection.

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

Parameters

sender

The object that sent this message.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- [alignJustified](#): (page 23)
- [alignLeft](#): (page 23)
- [alignRight](#): (page 24)

Declared In

WebView.h

alignJustified:

An action method that applies full justification to selected content or all content if there's no selection.

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

Parameters

sender

The object that sent this message.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- [alignJustified](#): (page 23)
- [alignLeft](#): (page 23)
- [alignRight](#): (page 24)

Declared In

WebView.h

alignLeft:

An action method that applies left justification to selected content or all content if there's no selection.

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

Parameters

sender

The object that sent this message.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- [alignJustified](#): (page 23)
- [alignCenter](#): (page 22)
- [alignRight](#): (page 24)

Declared In

WebView.h

alignRight:

An action method that applies right justification to selected content or all content if there is no selection.

- (void)alignRight:(id)sender

Parameters

sender

The object that sent this message.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- [alignJustified:](#) (page 23)
- [alignLeft:](#) (page 23)
- [alignCenter:](#) (page 22)

Declared In

WebView.h

applicationNameForUserAgent

Returns the receiver's application name that is used in the user-agent string.

- (NSString *)applicationNameForUserAgent

Return Value

The application name to use in the user-agent string. The user-agent is used by websites to identify the client browser.

Availability

Available in Mac OS X v10.2 with Safari 1.0 and later.

Available in Mac OS X v10.2.7 and later.

See Also

- [setApplicationNameForUserAgent:](#) (page 55)
- [setCustomUserAgent:](#) (page 56)
- [customUserAgent](#) (page 31)

Declared In

WebView.h

applyStyle:

Applies the CSS typing style to the current selection.

- (void)applyStyle:(DOMCSSStyleDeclaration *)style

Parameters

style

The style to apply to the current selection.

Discussion

This method does nothing if there is no current selection or if the current selection is collapsed.

This method hides the complexities of applying styles to elements. If necessary, this method will make multiple passes over the range of the current selection to ensure that the requested style is applied to the elements in that range, and takes into account the complexities of CSS style application rules. This method also simplifies styling attributes so that the minimum number of styling directives are used to yield a given computed style.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- [setTypingStyle:](#) (page 64)
- [typingStyle](#) (page 70)
- [computedStyleForElement:pseudoElement:](#) (page 30)

Declared In

WebView.h

backForwardList

Returns the receiver's back-forward list.

```
- (WebBackForwardList *)backForwardList
```

Return Value

The receiver's back-forward list.

Availability

Available in Mac OS X v10.2 with Safari 1.0 and later.

Available in Mac OS X v10.2.7 and later.

See Also

- [goBack](#) (page 36)
- [goForward](#) (page 37)
- [goToBackForwardItem:](#) (page 38)

Declared In

WebView.h

canGoBack

Returns whether the previous location can be loaded.

```
- (BOOL)canGoBack
```

Return Value

YES if able to move backward; otherwise, NO.

Availability

Available in Mac OS X v10.2 with Safari 1.0 and later.

Available in Mac OS X v10.2.7 and later.

See Also

- [goBack:](#) (page 36)
- [canGoForward](#) (page 26)
- [goForward:](#) (page 37)

Declared In

WebView.h

canGoForward

Returns whether the next location can be loaded.

- (BOOL)canGoForward

Return Value

YES if able to move forward; otherwise, NO.

Availability

Available in Mac OS X v10.2 with Safari 1.0 and later.

Available in Mac OS X v10.2.7 and later.

See Also

- [goForward:](#) (page 37)
- [canGoBack](#) (page 25)
- [goBack:](#) (page 36)

Declared In

WebView.h

canMakeTextLarger

Returns whether the text can be made larger.

- (BOOL)canMakeTextLarger

Return Value

YES if able to make the text larger; otherwise, NO.

Availability

Available in Mac OS X v10.2 with Safari 1.0 and later.

Available in Mac OS X v10.2.7 and later.

See Also

- [makeTextLarger:](#) (page 42)
- [makeTextSmaller:](#) (page 43)
- [canMakeTextSmaller](#) (page 27)

Declared In

WebView.h

canMakeTextSmaller

Returns whether the text can be made smaller.

- (BOOL)canMakeTextSmaller

Return Value

YES if able to make the text smaller; otherwise, NO.

Availability

Available in Mac OS X v10.2 with Safari 1.0 and later.

Available in Mac OS X v10.2.7 and later.

See Also

- [makeTextSmaller:](#) (page 43)
- [canMakeTextLarger](#) (page 26)
- [makeTextLarger:](#) (page 42)

Declared In

WebView.h

canMakeTextStandardSize

Returns whether the current text size is a multiple of 1.

- (BOOL)canMakeTextStandardSize

Return Value

YES if the current text size is a multiple of 1; otherwise, NO.

Availability

Available in Mac OS X v10.4.11 and later.

Declared In

WebView.h

changeAttributes:

An action method that changes the attributes of the current selection.

- (void)changeAttributes:(id)sender

Parameters

sender

The object that sent this message.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- [changeFont:](#) (page 28)

Declared In

WebView.h

changeColor:

Sets the color of the selected content.

- (void)changeColor:(id) *sender*

Parameters

sender

The object that sent this message.

Discussion

This method is invoked by the `NSColorPanel` sender.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- [changeDocumentBackgroundColor:](#) (page 28)

Declared In

WebView.h

changeDocumentBackgroundColor:

Sets the background color of the selected content.

- (void)changeDocumentBackgroundColor:(id) *sender*

Parameters

sender

The object that sent this message.

Discussion

This method is invoked by the `NSColorPanel` sender.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- [changeColor:](#) (page 28)

Declared In

WebView.h

changeFont:

An action method that changes the font of the selection, or all content if there is no selection.

- (void)changeFont:(id) *sender*

Parameters*sender*

The object that sent this message.

Discussion

If the receiver doesn't use the Fonts panel, this method does nothing.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- [changeAttributes](#): (page 27)

Declared In

WebView.h

checkSpelling:

An action method that searches for a misspelled word in the receiver.

- (void)checkSpelling:(id)sender

Parameters*sender*

The object that sent this message.

Discussion

This action method starts a search at the end of the selection and continues until it reaches a word suspected of being misspelled or the end of the content. If a word isn't recognized by the spelling server, a [showGuessPanel](#): (page 65) message is sent to the receiver which opens the Guess panel and allows the user to make a correction or add the word to the local dictionary.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- [showGuessPanel](#): (page 65)

Declared In

WebView.h

close

Closes the web view when it's no longer needed.

- (void)close

Discussion

Closes the web view by unloading its webpage and canceling any pending load requests. A closed web view no longer responds to new requests nor sends delegate messages. If the application uses garbage collection, this method needs to be invoked before an instance is collected. It is invoked automatically if the receiver's enclosing window or host window is closed and sending [shouldCloseWithWindow](#) (page 65) to the receiver returns YES. Applications that do not use garbage collection can still use this method to stop the receiver from loading and sending delegate messages.

Availability

Available in Mac OS X v10.4.11 and later.

Declared In

WebView.h

computedStyleForElement:pseudoElement:

Returns the computed style of an element and its pseudo element.

```
- (DOMCSSStyleDeclaration *)computedStyleForElement:(DOMElement
 *)elementpseudoElement:(NSString *)pseudoElement
```

Parameters

element

The element whose computed style is returned.

pseudoElement

The pseudo element for *element*.

Return Value

An immutable object describing the computed style of *element* and *pseudoElement* according to the Cascading Style Sheets Specification at <http://www.w3.org/TR/CSS21>. Returns `nil` if the receiver doesn't display *element*.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- [typingStyle](#) (page 70)
- [setTypingStyle:](#) (page 64)
- [applyStyle:](#) (page 24)

Declared In

WebView.h

copy:

Action method that copies the selected content to the general pasteboard.

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

Parameters

sender

The object that sent this message.

Discussion

This action method copies the selected content onto the general pasteboard, in as many formats as the receiver supports. For example, a plain text object uses `NSStringPboardType` for plain text, and a rich text object also uses `NSRTEPboardType`.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- [cut:](#) (page 32)
- [paste:](#) (page 45)

Declared In

WebView.h

copyFont:

An action method that copies font information onto the font pasteboard.

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

Parameters

sender

The object that sent this message.

Discussion

This action method copies the font information for the first character of the selection (or for the insertion point) onto the font pasteboard as `NSFontPboardType`.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- [pasteFont:](#) (page 47)

Declared In

WebView.h

customTextEncodingName

Returns the custom text encoding name.

```
- (NSString *)customTextEncodingName
```

Return Value

The receiver's custom text encoding name or `nil` if no custom text encoding name was set.

Availability

Available in Mac OS X v10.2 with Safari 1.0 and later.

Available in Mac OS X v10.2.7 and later.

See Also

- [setCustomTextEncodingName:](#) (page 55)

Declared In

WebView.h

customUserAgent

Returns the receiver's custom user-agent string.

- (NSString *)customUserAgent

Return Value

The user-agent string to identify the client browser. The custom user-agent string is used for all URLs.

Availability

Available in Mac OS X v10.2 with Safari 1.0 and later.

Available in Mac OS X v10.2.7 and later.

See Also

- [setApplicationNameForUserAgent:](#) (page 55)
- [applicationNameForUserAgent](#) (page 24)
- [customUserAgent](#) (page 31)

Declared In

WebView.h

cut:

An action method that deletes selected content and puts it on the general pasteboard.

- (void)cut:(id)sender

Parameters

sender

The object that sent this message.

Discussion

This action method deletes the selected content and places it onto the general pasteboard, in as many formats as the receiver supports. For example, a plain text object uses `NSStringPboardType` for plain text, and a rich text object also uses `NSRTFPboardType`.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- [copy:](#) (page 30)
- [paste:](#) (page 45)

Declared In

WebView.h

delete:

An action method that deletes the selected content.

- (void)delete:(id)sender

Parameters

sender

The object that sent this message.

Discussion

The pasteboard is unaffected by invoking this method.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- [cut](#): (page 32)

Declared In

WebView.h

deleteSelection

Deletes the receiver's current selection unless it's collapsed.

- (void)deleteSelection

Discussion

No content is removed if the current selection is collapsed (a range is selected with the same nodes and offsets for the start and end) or if there is no current selection.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- [replaceSelectionWithText](#): (page 52)
- [replaceSelectionWithMarkupString](#): (page 50)
- [replaceSelectionWithArchive](#): (page 50)
- [replaceSelectionWithNode](#): (page 51)

Declared In

WebView.h

downloadDelegate

Return the receiver's download delegate.

- (id)downloadDelegate

Return Value

The receiver's download delegate that implements the `WebDownload` protocol.

Availability

Available in Mac OS X v10.2 with Safari 1.0 and later.

Available in Mac OS X v10.2.7 and later.

See Also

- [setDownloadDelegate](#): (page 56)

Declared In

WebView.h

drawsBackground

Returns whether the web view draws a background.

- (BOOL)drawsBackground

Return Value

YES if a background is drawn; otherwise, NO.

Availability

Available in Mac OS X v10.4.11 and later.

Declared In

WebView.h

editableDOMRangeForPoint:

Returns the editable DOM object located at a given point.

- (DOMRange *)editableDOMRangeForPoint:(NSPoint)point

Parameters

point

The location of the editable DOM object.

Return Value

A single range object of the editable DOM object located at *point* in the receiver's coordinates.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- [selectedDOMRange](#) (page 53)
- [setSelectedDOMRange:affinity:](#) (page 62)

Declared In

WebView.h

editingDelegate

Returns the receiver's editing delegate.

- (id)editingDelegate

Return Value

The receiver's editing delegate.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- [setEditingDelegate:](#) (page 58)

Declared In
WebView.h

elementAtPoint:

Returns a dictionary description of the element at a given point in the receiver's coordinates.

- (NSDictionary *)elementAtPoint:(NSPoint)point

Parameters

point

The point to represent as a dictionary.

Return Value

A dictionary description of the element at *point* in the receiver's coordinates.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- webView:dragDestinationActionMaskForDraggingInfo: (WebUIDelegate)
- webView:dragSourceActionMaskForPoint:

Declared In
WebView.h

estimatedProgress

Returns an estimate, as a percentage, of the amount of content that is currently loaded.

- (double)estimatedProgress

Return Value

A number ranging from 0 to 1.0 and, once a load completes, 1.0 until a new load starts, at which point it resets to 0.

The value is an estimate based on the total number of bytes expected to be received for a document, including all its possible subresources. For more accurate load progress information, implement delegates conforming to the `WebFrameLoadDelegate` and `WebResourceLoadDelegate` informal protocols.

Availability

Available in Mac OS X v10.2 with Safari 1.0 and later.

Available in Mac OS X v10.2.7 and later.

See Also

- [WebViewProgressEstimateChangedNotification](#) (page 75)
- [WebViewProgressFinishedNotification](#) (page 75)
- [WebViewProgressStartedNotification](#) (page 75)

Declared In
WebView.h

frameLoadDelegate

Return the receiver's frame load delegate.

- (id)frameLoadDelegate

Return Value

A frame load delegate that conforms to the `WebFrameLoadDelegate` protocol.

Availability

Available in Mac OS X v10.2 with Safari 1.0 and later.

Available in Mac OS X v10.2.7 and later.

See Also

- [setFrameLoadDelegate:](#) (page 58)

Declared In

WebView.h

goBack

Loads the previous location in the back-forward list.

- (BOOL)goBack

Return Value

YES if able to move backward; otherwise, NO.

Availability

Available in Mac OS X v10.2 with Safari 1.0 and later.

Available in Mac OS X v10.2.7 and later.

See Also

- [backForwardList](#) (page 25)
- [goForward](#) (page 37)
- [goToBackForwardItem:](#) (page 38)

Declared In

WebView.h

goBack:

An action method that loads the previous location in the back-forward list.

- (void)goBack:(id)sender

Parameters

sender

The object that sent this message.

Discussion

This method does nothing if it is unable to move backward.

Availability

Available in Mac OS X v10.2 with Safari 1.0 and later.

Available in Mac OS X v10.2.7 and later.

See Also

- [goForward:](#) (page 37)

Declared In

WebView.h

goForward

Loads the next location in the back-forward list.

- (BOOL)goForward

Return Value

YES if able to move forward; otherwise, NO.

Availability

Available in Mac OS X v10.2 with Safari 1.0 and later.

Available in Mac OS X v10.2.7 and later.

See Also

- [goToBackForwardItem:](#) (page 38)

- [goBack](#) (page 36)

Declared In

WebView.h

goForward:

An action method that loads the next location in the back-forward list.

- (void)goForward:(id) *sender*

Parameters

sender

The object that sent this message.

Discussion

This method does nothing if it is unable to move forward.

Availability

Available in Mac OS X v10.2 with Safari 1.0 and later.

Available in Mac OS X v10.2.7 and later.

See Also

- [goBack:](#) (page 36)

Declared In

WebView.h

goToBackForwardItem:

Loads a specific location from the back-forward list and sets it as the current item.

- (BOOL)goToBackForwardItem:(WebHistoryItem *)*item*

Parameters

item

The index of the location to load. This method sets the current item in the back-forward list to *item*.

Return Value

YES if *item* is in the back-forward list; otherwise, NO.

Availability

Available in Mac OS X v10.2 with Safari 1.0 and later.

Available in Mac OS X v10.2.7 and later.

See Also

- [goForward](#) (page 37)

- [goBack](#) (page 36)

- [backForwardList](#) (page 25)

Declared In

WebView.h

groupName

Returns the receiver's group name.

- (NSString *)groupName

Return Value

The receiver's group name.

Availability

Available in Mac OS X v10.2 with Safari 1.0 and later.

Available in Mac OS X v10.2.7 and later.

See Also

- [setGroupName:](#) (page 58)

Declared In

WebView.h

hostWindow

Returns the receiver's host window.

- (NSWindow *)hostWindow

Return Value

The receiver's host window.

Availability

Available in Mac OS X v10.2 with Safari 1.0 and later.

Available in Mac OS X v10.2.7 and later.

See Also

- [setHostWindow:](#) (page 59)

Declared In

WebView.h

initWithFrame:frameName:groupName:

Initializes the receiver with a frame rectangle, frame name, and group name.

```
- (id)initWithFrame:(NSRect) frameRect frameName:(NSString *)frameName groupName:(NSString *)groupName
```

Parameters

frameRect

The frame rectangle for the created view object.

frameName

The web frame's name. This should not be one of the predefined frame names (see the `WebFramefindFrameNamed:` method for a description of their meaning), but a custom name or a name used in HTML source. This parameter can be `nil`.

groupName

An arbitrary identifier used to group related frames. For example, JavaScript running in a frame can access any other frame in the same group. It's up to the application how it chooses to scope related frames. This parameter can be `nil`.

Return Value

An initialized view object or `nil` if the object couldn't be created.

Discussion

This method is the designated initializer for the `WebView` class.

Availability

Available in Mac OS X v10.2 with Safari 1.0 and later.

Available in Mac OS X v10.2.7 and later.

Declared In

WebView.h

isContinuousSpellCheckingEnabled

Returns whether the web view has continuous spell-checking enabled.

```
- (BOOL)isContinuousSpellCheckingEnabled
```

Return Value

YES if the object has continuous spell-checking enabled; otherwise, NO.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- [setContinuousSpellCheckingEnabled:](#) (page 55)

Declared In

WebView.h

isEditable

Returns whether the user is allowed to edit the document.

- (BOOL)isEditable

Return Value

YES if the receiver allows the user to edit the HTML document, NO if it doesn't.

Discussion

You can change the receiver's document programmatically regardless of this setting.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- [setEditable:](#) (page 57)

Declared In

WebView.h

isLoading

Returns whether the web view is loading content.

- (BOOL)isLoading

Return Value

YES if the web view is currently loading any resources; otherwise, NO.

Availability

Available in Mac OS X v10.4.11 and later.

Declared In

WebView.h

mainFrame

Returns the main frame, the root of the web frame hierarchy for this page.

- (WebFrame *)mainFrame

Return Value

The main frame.

Availability

Available in Mac OS X v10.2 with Safari 1.0 and later.

Available in Mac OS X v10.2.7 and later.

Related Sample Code

CarbonCocoaCoreImageTab

Declared In

WebView.h

mainFrameDocument

Returns the DOM document for the main frame.

```
- (DOMDocument *)mainFrameDocument
```

Return Value

The DOM document for the main frame.

Discussion

Invoking this method is equivalent to `[[webView mainFrame] DOMDocument]`.

Availability

Available in Mac OS X v10.4.11 and later.

Declared In

WebView.h

mainFrameIcon

Returns the site's favicon.

```
- (NSImage *)mainFrameIcon
```

Return Value

The site's icon. Returns `nil` if no favicon is provided.

Availability

Available in Mac OS X v10.4.11 and later.

Declared In

WebView.h

mainFrameTitle

Returns the HTML title of the loaded page.

```
- (NSString *)mainFrameTitle
```

Return Value

The HTML title of the loaded page. Returns @"" if the loaded document is not HTML.

Availability

Available in Mac OS X v10.4.11 and later.

Declared In

WebView.h

mainFrameURL

Returns the URL that the main frame loads.

- (NSString *)mainFrameURL

Return Value

The main frame URL string.

Availability

Available in Mac OS X v10.4.11 and later.

Declared In

WebView.h

maintainsInactiveSelection

Returns whether the selection is maintained when focus is lost.

- (BOOL)maintainsInactiveSelection

Return Value

YES if the selection is maintained when focus is lost; otherwise, NO.

Availability

Available in Mac OS X v10.4.11 and later.

Declared In

WebView.h

makeTextLarger:

Increases the text size by one unit.

- (void)makeTextLarger:(id)sender

Parameters

sender

The object that sent this message.

Availability

Available in Mac OS X v10.2 with Safari 1.0 and later.

Available in Mac OS X v10.2.7 and later.

See Also

- [canMakeTextLarger](#) (page 26)
- [canMakeTextSmaller](#) (page 27)
- [makeTextSmaller:](#) (page 43)

Declared In

WebView.h

makeTextSmaller:

Reduces the text size by one unit.

- (void)makeTextSmaller:(id)sender

Parameters

sender

The object that sent this message.

Availability

Available in Mac OS X v10.2 with Safari 1.0 and later.

Available in Mac OS X v10.2.7 and later.

See Also

- [canMakeTextSmaller](#) (page 27)
- [canMakeTextLarger](#) (page 26)
- [makeTextLarger:](#) (page 42)

Declared In

WebView.h

makeTextStandardSize:

Resets the text size to a multiple of 1.

- (void)makeTextStandardSize:(id)sender

Parameters

sender

The object that sent this message.

Availability

Available in Mac OS X v10.4.11 and later.

Declared In

WebView.h

mediaStyle

Returns the receiver's CSS media property.

- (NSString *)mediaStyle

Return Value

The receiver's CSS media property. `nil` if no media style was set.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- [setMediaStyle:](#) (page 60)

Declared In

WebView.h

moveDragCaretToPoint:

Moves the drag caret that indicates the destination of a drag operation to a given point.

```
- (void)moveDragCaretToPoint:(NSPoint)point
```

Parameters

point

The point to move the drag caret to.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- [removeDragCaret](#) (page 50)

Declared In

WebView.h

moveToBeginningOfSentence:

Moves the insertion point to the beginning of the current sentence.

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

Parameters

sender

The object that sent this message.

Availability

Available in Mac OS X v10.4.11 and later.

Declared In

WebView.h

moveToBeginningOfSentenceAndModifySelection:

Moves the insertion point and extends the selection to the beginning of the current sentence.

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

Parameters

sender

The object that sent this message.

Availability

Available in Mac OS X v10.4.11 and later.

Declared In

WebView.h

moveToEndOfSentence:

Moves the insertion point to the end of the current sentence.

- (void)moveToEndOfSentence:(id) *sender*

Parameters

sender

The object that sent this message.

Availability

Available in Mac OS X v10.4.11 and later.

Declared In

WebView.h

moveToEndOfSentenceAndModifySelection:

Moves the insertion point and extends the selection to the end of the current sentence.

- (void)moveToEndOfSentenceAndModifySelection:(id) *sender*

Parameters

sender

The object that sent this message.

Availability

Available in Mac OS X v10.4.11 and later.

Declared In

WebView.h

paste:

An action method that pastes content from the pasteboard at the insertion point or over the selection.

- (void)paste:(id) *sender*

Parameters

sender

The object that sent this message.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- [cut](#): (page 32)
- [copy](#): (page 30)

Declared In

WebView.h

pasteAsPlainText:

An action method that pastes pasteboard content as plain text.

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

Parameters

sender

The object that sent this message.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- [pasteAsRichText](#): (page 46)

Declared In

WebView.h

pasteAsRichText:

An action method that pastes pasteboard content into the receiver as rich text, maintaining its attributes.

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

Parameters

sender

The object that sent this message.

Discussion

The text is inserted at the insertion point if there is one; otherwise, it replaces the selection.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- [pasteAsPlainText](#): (page 46)

Declared In

WebView.h

pasteboardTypesForElement:

Returns an array of pasteboard types for an element.

```
- (NSArray *)pasteboardTypesForElement:(NSDictionary *)element
```

Parameters

element

The element whose pasteboard types you want.

Return Value

An array of pasteboard types for an element.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- [pasteboardTypesForSelection](#) (page 47)
- + [URLFromPasteboard:](#) (page 21)
- + [URLTitleFromPasteboard:](#) (page 22)

Declared In

WebView.h

pasteboardTypesForSelection

Returns an array of pasteboard types that can be used for the current selection of the receiver.

```
- (NSArray *)pasteboardTypesForSelection
```

Return Value

An array of pasteboard types that can be used for the current selection of the receiver.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- [pasteboardTypesForElement:](#) (page 47)
- + [URLFromPasteboard:](#) (page 21)
- + [URLTitleFromPasteboard:](#) (page 22)

Declared In

WebView.h

pasteFont:

An action method that pastes font information from the font pasteboard.

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

Parameters

sender

The object that sent this message.

Discussion

This action method pastes font information from the font pasteboard onto the selected content or insertion point of a rich text object, or over all text of the receiver.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- [copyFont](#): (page 31)

Declared In

WebView.h

performFindPanelAction:

An action method that opens the Find menu and Find panel.

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

Parameters

sender

The object that sent this message.

Availability

Available in Mac OS X v10.3.9 and later.

Declared In

WebView.h

policyDelegate

Returns the receiver's policy delegate.

```
- (id)policyDelegate
```

Return Value

A policy delegate that conforms to the `WebPolicyDelegate` protocol.

Availability

Available in Mac OS X v10.2 with Safari 1.0 and later.

Available in Mac OS X v10.2.7 and later.

See Also

- [setPolicyDelegate](#): (page 61)

Declared In

WebView.h

preferences

Returns the receiver's preferences.

- (WebPreferences *)preferences

Return Value

The receiver's preferences or the standard preferences, if the preferences were not set using the [setPreferences:](#) (page 61) method.

Availability

Available in Mac OS X v10.2 with Safari 1.0 and later.

Available in Mac OS X v10.2.7 and later.

See Also

- [setPreferences:](#) (page 61)

+ standardPreferences

Declared In

WebView.h

preferencesIdentifier

Returns the identifier of the receiver's preferences.

- (NSString *)preferencesIdentifier

Return Value

The preferences identifier.

Availability

Available in Mac OS X v10.2 with Safari 1.0 and later.

Available in Mac OS X v10.2.7 and later.

See Also

- [setPreferencesIdentifier:](#) (page 61)

Declared In

WebView.h

reload:

Reloads the current page.

- (void)reload:(id)sender

Parameters

sender

The object that sent this message.

Availability

Available in Mac OS X v10.2 with Safari 1.0 and later.

Available in Mac OS X v10.2.7 and later.

See Also

- [setResourceLoadDelegate:](#) (page 62)

Declared In

WebView.h

removeDragCaret

Removes the drag caret that indicates the destination of a drag operation.

```
- (void)removeDragCaret
```

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- [moveDragCaretToPoint:](#) (page 44)

Declared In

WebView.h

replaceSelectionWithArchive:

Replaces the current selection with an archive's contents.

```
- (void)replaceSelectionWithArchive:(WebArchive *)archive
```

Parameters

archive

The archive that replaces the current selection.

Discussion

If the current selection is collapsed (a range is selected with the same nodes and offsets for the start and end) then no content is removed when inserting the archive, and the selection is collapsed and moved to the end of the inserted content. If no content is selected, the archive is not inserted.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- [replaceSelectionWithText:](#) (page 52)
- [replaceSelectionWithMarkupString:](#) (page 50)
- [replaceSelectionWithNode:](#) (page 51)
- [deleteSelection](#) (page 33)

Declared In

WebView.h

replaceSelectionWithMarkupString:

Replaces the current selection with mixed text and markup.

```
- (void)replaceSelectionWithMarkupString:(NSString *)markupString
```

Parameters*markupString*

The markup string that replaces the current selection.

Discussion

If the current selection is collapsed (a range is selected with the same nodes and offsets for the start and end) then no content is removed when inserting the markup, and the selection is collapsed and moved to the end of the inserted content. If no content is selected, the markup is not inserted.

See <http://msdn.microsoft.com/workshop/networking/clipboard/htmlclipboard.asp> for a specification of the supported HTML markup.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- [replaceSelectionWithText:](#) (page 52)
- [replaceSelectionWithNode:](#) (page 51)
- [replaceSelectionWithArchive:](#) (page 50)
- [deleteSelection](#) (page 33)

Declared In

WebView.h

replaceSelectionWithNode:

Replaces the receiver's current selection with the specified DOM node.

```
- (void)replaceSelectionWithNode:(DOMNode *)node
```

Parameters*node*

The node that replaces the current selection.

Discussion

If the current selection is collapsed (a range is selected with the same nodes and offsets for the start and end) then no content is removed when inserting the node, and the selection is collapsed and moved to the end of the inserted content. If no content is selected, the node is not inserted.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- [replaceSelectionWithText:](#) (page 52)
- [replaceSelectionWithMarkupString:](#) (page 50)
- [replaceSelectionWithArchive:](#) (page 50)
- [deleteSelection](#) (page 33)

Declared In

WebView.h

replaceSelectionWithText:

Replaces the current selection with a string of text.

```
- (void)replaceSelectionWithText:(NSString *)text
```

Parameters

text

The text that replaces the current selection.

Discussion

If the current selection is collapsed (a range is selected with the same nodes and offsets for the start and end) then no content is removed when inserting the text, and the selection is collapsed and moved to the end of the inserted content. If no content is selected, the text is not inserted.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- [replaceSelectionWithNode:](#) (page 51)
- [replaceSelectionWithMarkupString:](#) (page 50)
- [replaceSelectionWithArchive:](#) (page 50)
- [deleteSelection](#) (page 33)

Declared In

WebView.h

resourceLoadDelegate

Returns the receiver's resource load delegate.

```
- (id)resourceLoadDelegate
```

Return Value

A resource load delegate that conforms to the `WebResourceLoadDelegate` protocol.

Availability

Available in Mac OS X v10.2 with Safari 1.0 and later.

Available in Mac OS X v10.2.7 and later.

See Also

- [setResourceLoadDelegate:](#) (page 62)

Declared In

WebView.h

searchFor:direction:caseSensitive:wrap:

Searches a document view for a string and highlights it if it is found.

```
- (BOOL)searchFor:(NSString *)stringdirection:(BOOL)forwardcaseSensitive:(BOOL)caseFlagwrap:(BOOL)wrapFlag
```

Parameters

string

The search string.

forward

If YES the direction of the search is forward; if NO, the direction is backward.

caseFlag

If YES if the search is case sensitive; otherwise, it is not.

wrapFlag

If YES if the search wraps; otherwise, it does not.

Return Value

YES if the search is successful; otherwise, NO.

Discussion

The search for *string* begins from the current selection and continues in the direction specified by *forward*. The search continues across all frames.

Availability

Available in Mac OS X v10.2 with Safari 1.0 and later.

Available in Mac OS X v10.2.7 and later.

See Also

- [applicationNameForUserAgent](#) (page 24)
- [setCustomUserAgent:](#) (page 56)
- [customUserAgent](#) (page 31)

Declared In

WebView.h

selectedDOMRange

Returns the range of the current selection.

- (DOMRange *)selectedDOMRange

Return Value

The range of the current selection. *nil* if nothing is selected.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- [selectionAffinity](#) (page 54)
- [setSelectedDOMRange:affinity:](#) (page 62)
- [editableDOMRangeForPoint:](#) (page 34)

Declared In

WebView.h

selectedFrame

Returns the frame with the active selection.

- (WebFrame *)selectedFrame

Return Value

The frame that contains the first responder. If it doesn't exist, the frame that contains a non-zero-length selection; otherwise, `nil`.

Availability

Available in Mac OS X v10.4.11 and later.

Declared In

WebView.h

selectionAffinity

Returns the current selection affinity.

- (NSSelectionAffinity)selectionAffinity

Return Value

The preferred direction of selection—upward or downward—of the receiver's current selection. For example, if text wraps across line boundaries, the value returned by this method indicates whether or not the insertion point appears after the last character of the first line or before the first character of the following line.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- [selectedDOMRange](#) (page 53)
- [setSelectedDOMRange:affinity:](#) (page 62)

Declared In

WebView.h

selectSentence:

Selects the entire sentence around the insertion point.

- (void)selectSentence:(id)sender

Parameters

sender

The object that sent this message.

Availability

Available in Mac OS X v10.4.11 and later.

Declared In

WebView.h

setApplicationNameForUserAgent:

Sets the application name used in the user-agent string.

```
- (void)setApplicationNameForUserAgent:(NSString *)applicationName
```

Parameters

applicationName

The application name to use in the user-agent string. The user-agent is used by websites to identify the client browser.

Availability

Available in Mac OS X v10.2 with Safari 1.0 and later.

Available in Mac OS X v10.2.7 and later.

See Also

- [applicationNameForUserAgent](#) (page 24)

- [setCustomUserAgent:](#) (page 56)

- [customUserAgent](#) (page 31)

Declared In

WebView.h

setContinuousSpellCheckingEnabled:

Sets whether the web view has continuous spell-checking enabled.

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

Parameters

flag

YES if the object should have continuous spell-checking enabled; otherwise, NO.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- [isContinuousSpellCheckingEnabled](#) (page 39)

Declared In

WebView.h

setCustomTextEncodingName:

Sets the custom text encoding name.

```
- (void)setCustomTextEncodingName:(NSString *)encodingName
```

Parameters

encodingName

A text encoding name. If `nil`, the default encoding is restored.

Discussion

This method overrides the default text encoding, including any encoding that is specified in the webpage header or HTTP response. Invoking this method stops any load in progress. The default encoding is restored when the main frame changes to a new location, or if *encodingName* is *nil*.

Availability

Available in Mac OS X v10.2 with Safari 1.0 and later.

Available in Mac OS X v10.2.7 and later.

See Also

- [customTextEncodingName](#) (page 31)

Declared In

WebView.h

setCustomUserAgent:

Sets the receiver's custom user-agent string.

```
- (void)setCustomUserAgent:(NSString *)userAgentString
```

Parameters

userAgentString

The custom user-agent string. The user-agent string is used by websites to identify the client browser.

The custom user-agent string is used for all URLs. If *nil*, then the receiver constructs a user-agent string that produces the best rendering results for each URL.

Availability

Available in Mac OS X v10.2 with Safari 1.0 and later.

Available in Mac OS X v10.2.7 and later.

See Also

- [setApplicationNameForUserAgent:](#) (page 55)

- [applicationNameForUserAgent](#) (page 24)

- [customUserAgent](#) (page 31)

Declared In

WebView.h

setDownloadDelegate:

Sets the receiver's shared download delegate.

```
- (void)setDownloadDelegate:(id)delegate
```

Parameters

delegate

The download delegate that implements the `WebDownload` protocol.

Discussion

WebKit may create `WebDownload` objects automatically to handle downloads that start with a webpage or link.

Availability

Available in Mac OS X v10.2 with Safari 1.0 and later.

Available in Mac OS X v10.2.7 and later.

See Also

- [downloadDelegate](#) (page 33)

Declared In

WebView.h

setDrawsBackground:

Sets whether a default background is drawn when the webpage has no background set.

- (void)setDrawsBackground:(BOOL)drawsBackground

Parameters

drawsBackground

If YES, a default background is drawn; if NO, it is not.

Availability

Available in Mac OS X v10.4.11 and later.

Declared In

WebView.h

setEditable:

Sets whether the receiver allows the user to edit its HTML document.

- (void)setEditable:(BOOL)flag

Parameters

flag

YES if the receiver allows the user to edit the document. NO if an element in the receiver's document can be edited only if the CONTENTEDITABLE attribute has been set on the element or one of its parent elements.

Discussion

You can change the receiver's document programmatically regardless of this setting. By default a `WebView` object is not editable.

Normally, an HTML document is not editable unless the elements within the document are editable. This method provides a low-level way to make the contents of a `WebView` object editable without altering the document or DOM structure.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- [isEditable](#) (page 40)

Declared In

WebView.h

setEditingDelegate:

Sets the receiver's editing delegate.

- (void)setEditingDelegate:(id)delegate

Parameters

delegate

The editing delegate for the web view that conforms to the `WebEditingDelegate` protocol.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- [editingDelegate](#) (page 34)

Declared In

WebView.h

setFrameLoadDelegate:

Sets the receiver's frame load delegate.

- (void)setFrameLoadDelegate:(id)delegate

Parameters

delegate

A frame load delegate that conforms to the `WebFrameLoadDelegate` protocol.

Availability

Available in Mac OS X v10.2 with Safari 1.0 and later.

Available in Mac OS X v10.2.7 and later.

See Also

- [frameLoadDelegate](#) (page 36)

Declared In

WebView.h

setGroupName:

Sets the receiver's group name.

- (void)setGroupName:(NSString *)groupName

Parameters

groupName

An arbitrary identifier used to group related frames.

Discussion

You might use this method to set the group name of a `WebView` object after it is loaded from a nib file.

Availability

Available in Mac OS X v10.2 with Safari 1.0 and later.

Available in Mac OS X v10.2.7 and later.

See Also

- [initWithFrame:frameName:groupName:](#) (page 39)
- [groupName](#) (page 38)

Declared In

WebView.h

setHostWindow:

Sets the receiver's host window.

```
- (void)setHostWindow:(NSWindow *)hostWindow
```

Parameters

hostWindow

A host window.

Discussion

This method sets the receiver's host window to *hostWindow*. Your application should only use this method if a web view is going to be removed from its window temporarily, and you want the web view to continue operating (for example, you don't want to interrupt a load in progress). Since the receiver retains *hostWindow*, it is your responsibility to set the host window to `nil` before closing the window to avoid a retain loop.

For example, you might invoke this method if you attach a web view to an `NSTabView` object (as in a tabbed browser implementation). The `NSTabView` object takes views out of the window when they are not in the active tab, so you need to invoke this method before the web view is removed from its window. If you don't invoke this method, plug-ins will stop operating when the web view is removed from its window.

Note: Plug-ins and JavaScript depend on a window to function properly even if the web view is not in an actual window.

Availability

Available in Mac OS X v10.2 with Safari 1.0 and later.

Available in Mac OS X v10.2.7 and later.

See Also

- [hostWindow](#) (page 38)

Declared In

WebView.h

setMainFrameURL:

Sets the URL that the main frame loads.

```
- (void)setMainFrameURL:(NSString *)URLString
```

Parameters

URLString

The main frame URL string.

Discussion

This method is functionally equivalent to `[[webView mainFrame] loadRequest:]`.

Availability

Available in Mac OS X v10.4.11 and later.

Declared In

WebView.h

setMaintainsBackForwardList:

Sets whether to use a back-forward list.

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

Parameters

flag

If `NO`, clears the back-forward list and release the page cache; otherwise, it does not.

Discussion

The back-forward list maintains a page cache, so applications that do not use the [goForward](#) (page 37) or [goBack](#) (page 36) methods should disable it.

Availability

Available in Mac OS X v10.2 with Safari 1.0 and later.

Available in Mac OS X v10.2.7 and later.

See Also

- [backForwardList](#) (page 25)

Declared In

WebView.h

setMediaStyle:

Sets the receiver's CSS media property.

```
- (void)setMediaStyle:(NSString *)mediaStyle
```

Parameters

mediaStyle

The CSS media property for the receiver.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- [mediaStyle](#) (page 43)

Declared In

WebView.h

setPolicyDelegate:

Sets the receiver's policy delegate.

- (void)setPolicyDelegate:(id)delegate

Parameters*delegate*A policy delegate that conforms to the `WebPolicyDelegate` protocol.**Availability**

Available in Mac OS X v10.2 with Safari 1.0 and later.

Available in Mac OS X v10.2.7 and later.

See Also- [policyDelegate](#) (page 48)**Declared In**

WebView.h

setPreferences:

Sets the receiver's preferences.

- (void)setPreferences:(WebPreferences *)preferences

Parameters*preferences*

The web view's preferences.

DiscussionTypically, you do not invoke this method directly. Use the [setPreferencesIdentifier:](#) (page 61) method to change the receiver's preferences.**Availability**

Available in Mac OS X v10.2 with Safari 1.0 and later.

Available in Mac OS X v10.2.7 and later.

See Also- [preferences](#) (page 48)**Declared In**

WebView.h

setPreferencesIdentifier:

Sets the receiver's preferences identifier, creating a preferences object if needed.

- (void)setPreferencesIdentifier:(NSString *)anIdentifier

Parameters*anIdentifier*

The unique identifier for the preferences—it is fixed to the keys used to store the receiver’s preferences in the user defaults database. `WebView` objects can share instances of the `WebPreferences` class by using the same preferences identifier.

Discussion

This method sets the receiver’s preferences to the specified preferences object if it exists. Otherwise, this method creates a new `WebPreferences` object for the receiver initialized with *anIdentifier*. Use this method to change the preferences used by the receiver’s `WebFrameView` objects. If you do not directly set the preferences, `WebFrameView` objects use the preferences returned by the `standardPreferences` class method of `WebPreferences`.

Availability

Available in Mac OS X v10.2 with Safari 1.0 and later.

Available in Mac OS X v10.2.7 and later.

See Also

- [preferencesIdentifier](#) (page 49)

- `setAutosaves:`

Declared In

`WebView.h`

setResourceLoadDelegate:

Sets the receiver's resource load delegate.

```
- (void)setResourceLoadDelegate:(id)delegate
```

Parameters*delegate*

A resource load delegate that conforms to the `WebResourceLoadDelegate` protocol.

Availability

Available in Mac OS X v10.2 with Safari 1.0 and later.

Available in Mac OS X v10.2.7 and later.

See Also

- [resourceLoadDelegate](#) (page 52)

Declared In

`WebView.h`

setSelectedDOMRange:affinity:

Selects a range of nodes.

```
- (void)setSelectedDOMRange:(DOMRange
 *)rangeaffinity:(NSSelectionAffinity)selectionAffinity
```

Parameters*range*

The range of nodes to select. If *range* is `nil`, the current selection is cleared. This method raises a `DOMRangeException` if the range has been detached or refers to nodes not displayed by the receiver.

selectionAffinity

See the [selectionAffinity](#) (page 54) method for information on selection affinity.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- [selectedDOMRange](#) (page 53)
- [selectionAffinity](#) (page 54)
- [editableDOMRangeForPoint:](#) (page 34)

Declared In

WebView.h

setShouldCloseWithWindow:

Sets whether the web view should close when its window or host window closes.

```
- (void)setShouldCloseWithWindow:(BOOL)close
```

Parameters*close*

If YES, the web view should close; otherwise, it should not.

Availability

Available in Mac OS X v10.4.11 and later.

Declared In

WebView.h

setSmartInsertDeleteEnabled:

Sets whether the receiver should insert or delete spaces around selected words to preserve proper spacing and punctuation.

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

Parameters*flag*

If YES, the receiver performs smart insert and delete; if NO, it inserts and deletes exactly what's selected.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- [smartInsertDeleteEnabled](#) (page 65)

Declared In

WebView.h

setTextSizeMultiplier:

Change the font size multiplier for text displayed in web frame view objects managed by the receiver.

- (void)setTextSizeMultiplier:(float)*multiplier*

Parameters

multiplier

A fractional percentage value where 1.0 denotes 100%.

Availability

Available in Mac OS X v10.2 with Safari 1.0 and later.

Available in Mac OS X v10.2.7 and later.

See Also

- [textSizeMultiplier](#) (page 69)

Declared In

WebView.h

setTypingStyle:

Sets the receiver's CSS typing style.

- (void)setTypingStyle:(DOMCSSStyleDeclaration *)*style*

Parameters

style

The receiver's CSS typing style.

Discussion

The typing style is reset automatically when the receiver's selection changes.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- [typingStyle](#) (page 70)

- [computedStyleForElement:pseudoElement:](#) (page 30)

- [applyStyle:](#) (page 24)

Declared In

WebView.h

setUIDelegate:

Sets the receiver's user interface delegate.

- (void)setUIDelegate:(id)*delegate*

Parameters

delegate

A user interface delegate that conforms to the `WebUIDelegate` protocol.

Availability

Available in Mac OS X v10.2 with Safari 1.0 and later.

Available in Mac OS X v10.2.7 and later.

See Also

- [UIDelegate](#) (page 70)

Declared In

WebView.h

shouldCloseWithWindow

Returns whether the web view should close when its window or host window closes.

- (BOOL)shouldCloseWithWindow

Return Value

If YES, the web view should close; otherwise, it should not.

Availability

Available in Mac OS X v10.4.11 and later.

Declared In

WebView.h

showGuessPanel:

An action method that shows a spelling correction panel.

- (void)showGuessPanel:(id)sender

Parameters

sender

The object that sent this message.

Discussion

This action method opens the Spelling panel, allowing the user to make a correction during spell checking.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- [checkSpelling:](#) (page 29)

Declared In

WebView.h

smartInsertDeleteEnabled

Returns whether smart-space insertion and deletion is enabled.

- (BOOL)smartInsertDeleteEnabled

Return Value

YES if the receiver inserts or deletes space around selected words so as to preserve proper spacing and punctuation. NO if it inserts and deletes exactly what's selected.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- [setSmartInsertDeleteEnabled:](#) (page 63)

Declared In

WebView.h

spellCheckerDocumentTag

Returns the spell-checker document tag for this document.

- (NSInteger)spellCheckerDocumentTag

Return Value

The document tag for this web view. A tag identifying the receiver's text as a document for the spell-checker server. See the [NSSpellChecker](#) and [NSSpellServer](#) class specifications for more information on how this tag is used.

The return value changed from `unsigned int` to a `NSInteger` in Mac OS X v10.5.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

[spellCheckerDocumentTag \(NSTextView\)](#)

Declared In

WebView.h

startSpeaking:

An action method that starts speaking the selected text or all text if there's no selection.

- (void)startSpeaking:(id)sender

Parameters

sender

The object that sent this message.

Discussion

Speech continues asynchronously until the end of the text or until terminated by invoking the [stopSpeaking:](#) (page 67) method.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- [stopSpeaking:](#) (page 67)

Declared In
WebView.h

stopLoading:

An action method that stops the loading of any web frame content managed by the receiver.

- (void)stopLoading:(id) *sender*

Parameters

sender

The object that sent this message.

Discussion

Stops any content in the process of being loaded by the main frame or any of its children frames. Does nothing if no content is being loaded.

Availability

Available in Mac OS X v10.2 with Safari 1.0 and later.

Available in Mac OS X v10.2.7 and later.

Declared In
WebView.h

stopSpeaking:

An action method that stops speaking that is in progress.

- (void)stopSpeaking:(id) *sender*

Parameters

sender

The object that sent this message.

Discussion

This action method stops speech that was previously started with [startSpeaking:](#) (page 66).

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- [startSpeaking:](#) (page 66)

Declared In
WebView.h

stringByEvaluatingJavaScriptFromString:

Returns the result of running a script.

- (NSString *)stringByEvaluatingJavaScriptFromString:(NSString *) *script*

Parameters*script*

The script to run.

Return ValueThe result of running a JavaScript specified by *script*, or an empty string if the script failed.**Availability**

Available in Mac OS X v10.2 with Safari 1.0 and later.

Available in Mac OS X v10.2.7 and later.

Related Sample Code

WebKitPluginWithSimpleGUI

Declared In

WebView.h

styleDeclarationWithText:

Returns the CSS style declaration for the specified text.

- (DOMCSSStyleDeclaration *)styleDeclarationWithText:(NSString *)text

Parameters*text*

The text whose style declaration is returned.

Return ValueThe style declaration for *text*.**Availability**

Available in Mac OS X v10.3.9 and later.

Declared In

WebView.h

supportsTextEncoding

Returns whether the document view supports different text encodings.

- (BOOL)supportsTextEncoding

Return Value

YES if the receiver's document view can support different text encodings; otherwise, NO.

Availability

Available in Mac OS X v10.2 with Safari 1.0 and later.

Available in Mac OS X v10.2.7 and later.

Declared In

WebView.h

takeStringURLFrom:

Sets the receiver's current location by obtaining a URL string from the sender.

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

Parameters

sender

The object that sent this message.

Discussion

This method sets the receiver's current location to the value obtained by sending a `stringValue` message to *sender*, then starts loading the URL returned by *sender*.

Availability

Available in Mac OS X v10.2 with Safari 1.0 and later.

Available in Mac OS X v10.2.7 and later.

See Also

- `loadRequest:` (`WebFrame`)

Declared In

WebView.h

textSizeMultiplier

Returns the font size multiplier for text displayed in web frame view objects managed by the receiver.

```
- (float)textSizeMultiplier
```

Return Value

The font size multiplier, a fractional percentage value where 1.0 denotes 100%.

Availability

Available in Mac OS X v10.2 with Safari 1.0 and later.

Available in Mac OS X v10.2.7 and later.

See Also

- [setTextSizeMultiplier:](#) (page 64)

Declared In

WebView.h

toggleContinuousSpellChecking:

Toggles whether continuous spell checking is available.

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

Parameters

sender

The object that sent this message.

Availability

Available in Mac OS X v10.4.11 and later.

Declared In

WebView.h

toggleSmartInsertDelete:

Toggles whether spaces around selected words are inserted or deleted to preserve proper spacing and punctuation.

- (void)toggleSmartInsertDelete:(id)sender

Parameters

sender

The object that sent this message.

Availability

Available in Mac OS X v10.4.11 and later.

Declared In

WebView.h

typingStyle

Returns the receiver's CSS typing style.

- (DOMCSSStyleDeclaration *)typingStyle

Return Value

The receiver's CSS typing style.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- [setTypingStyle:](#) (page 64)
- [computedStyleForElement:pseudoElement:](#) (page 30)
- [applyStyle:](#) (page 24)

Declared In

WebView.h

UIDelegate

Returns the receiver's user interface delegate.

- (id)UIDelegate

Return Value

A user interface delegate that conforms to the `WebUIDelegate` protocol.

Availability

Available in Mac OS X v10.2 with Safari 1.0 and later.

Available in Mac OS X v10.2.7 and later.

See Also

- [setUIDelegate:](#) (page 64)

Declared In

WebView.h

undoManager

Returns the receiver's undo manager.

- (NSUndoManager *)undoManager

Return Value

The receiver's undo manager.

Availability

Available in Mac OS X v10.3.9 and later.

Declared In

WebView.h

userAgentForURL:

Returns the appropriate user-agent string for a given URL.

- (NSString *)userAgentForURL:(NSURL *)URL

Parameters

URL

The URL that you need the user-agent string for.

Return Value

The user-agent string for a given URL. The user-agent string is used by websites to identify the client browser.

Availability

Available in Mac OS X v10.2 with Safari 1.0 and later.

Available in Mac OS X v10.2.7 and later.

See Also

- [setCustomUserAgent:](#) (page 56)

- [customUserAgent](#) (page 31)

Declared In

WebView.h

windowScriptObject

Returns the receiver's window object from the scripting environment.

- (WebScriptObject *)windowScriptObject

Return Value

The receiver's window object.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- objectForWebScript (WebPlugin) (WebFrameLoadDelegate)
- webView:windowScriptObjectAvailable:

Related Sample Code

QT Capture Widget

WebKitPluginWithJavaScript

Declared In

WebView.h

writeElement:withPasteboardTypes:toPasteboard:

Writes an element to the pasteboard using a list of types.

```
- (void)writeElement:(NSDictionary *)elementwithPasteboardTypes:(NSArray *)types
toPasteboard:(NSPasteboard *)pasteboard
```

Parameters

element

The element to write to the pasteboard.

types

The pasteboard types to use for the element.

pasteboard

The pasteboard to use for writing.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- [writeSelectionWithPasteboardTypes:toPasteboard:](#) (page 72)

Declared In

WebView.h

writeSelectionWithPasteboardTypes:toPasteboard:

Writes the receiver's current selection to a pasteboard using a list of types.

```
- (void)writeSelectionWithPasteboardTypes:(NSArray *)types
toPasteboard:(NSPasteboard *)pasteboard
```


Parameters*types*

The pasteboard types to use for the selection.

pasteboard

The pasteboard to use for writing.

Availability

Available in Mac OS X v10.3.9 and later.

See Also- [writeElement:withPasteboardTypes:toPasteboard:](#) (page 72)**Declared In**

WebView.h

Constants

These constants represent predefined keys used to access an element dictionary. An element dictionary is an `NSDictionary` representation of an HTML element, as in a clicked or selected element. Some methods in the `WebPolicyDelegate` informal protocol have an element dictionary argument. The descriptions below describe the dictionary value for the key.

Constant	Description
<code>WebElementDOMNodeKey</code>	The <code>DOMNode</code> for this element.
<code>WebElementFrameKey</code>	The <code>WebFrame</code> object associated with this element.
<code>WebElementImageAltStringKey</code>	An <code>NSString</code> of the ALT attribute of an image element.
<code>WebElementImageKey</code>	An <code>NSImage</code> representing an image element.
<code>WebElementImageRectKey</code>	An <code>NSValue</code> containing an <code>NSRect</code> , the size of an image element.
<code>WebElementImageURLKey</code>	An <code>NSURL</code> containing the location of an image element.
<code>WebElementIsSelectedKey</code>	An <code>NSNumber</code> used as a <code>BOOL</code> value to indicate whether a text element is selected or not. Zero value indicates false, true otherwise.
<code>WebElementLinkURLKey</code>	An <code>NSURL</code> containing the location of a link if the element is within an anchor.
<code>WebElementLinkTargetFrameKey</code>	The <code>WebFrame</code> object associated with the target of the anchor.
<code>WebElementLinkTitleKey</code>	An <code>NSString</code> containing the title of an anchor.
<code>WebElementLinkLabelKey</code>	An <code>NSString</code> containing the text within an anchor.

Notifications

WebViewDidBeginEditingNotification

Posted when a web view begins any operation that changes its contents in response to user editing. The notification object is the `WebView` object that the user is editing. This notification does not contain a `userInfo` dictionary.

Availability

Available in Mac OS X v10.3.9 and later.

Declared In

`WebView.h`

WebViewDidChangeNotification

Posted when a web view performs any operation that changes its contents in response to user editing. The notification object is the `WebView` object that the user is editing. This notification does not contain a `userInfo` dictionary.

Availability

Available in Mac OS X v10.3.9 and later.

Declared In

`WebView.h`

WebViewDidChangeSelectionNotification

Posted when a web view changes its typing selection. The notification object is the `WebView` that changed its typing selection. This notification does not contain a `userInfo` dictionary.

Availability

Available in Mac OS X v10.3.9 and later.

Declared In

`WebView.h`

WebViewDidChangeTypingStyleNotification

Posted when a web view changes its typing style. The notification object is the `WebView` that changed its typing style. This notification does not contain a `userInfo` dictionary.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- [setTypingStyle:](#) (page 64)

Declared In

`WebView.h`

WebViewDidEndEditingNotification

Posted when a web view ends any operation that changes its contents in response to user editing. The notification object is the `WebView` that the user is editing. This notification does not contain a `userInfo` dictionary.

Availability

Available in Mac OS X v10.3.9 and later.

Declared In

`WebView.h`

WebViewProgressEstimateChangedNotification

Posted by a `WebView` object when the estimated progress value of a load changes. This notification may be posted zero or more times after a [WebViewProgressStartedNotification](#) (page 75) notification is posted. The notification object is the `WebView` for which the progress value has changed. This notification does not contain a `userInfo` dictionary.

Availability

Available in Mac OS X v10.2 with Safari 1.0 and later.

Available in Mac OS X v10.2.7 and later.

See Also

- [estimatedProgress](#) (page 35)

Declared In

`WebView.h`

WebViewProgressFinishedNotification

Posted by a `WebView` object when the load has finished. The notification object is the `WebView` that finished loading. This notification does not contain a `userInfo` dictionary.

Availability

Available in Mac OS X v10.2 with Safari 1.0 and later.

Available in Mac OS X v10.2.7 and later.

See Also

- [estimatedProgress](#) (page 35)

Declared In

`WebView.h`

WebViewProgressStartedNotification

Posted by a `WebView` object when a load begins, including a load that is initiated in a subframe. The notification object is the `WebView` that began loading. This notification does not contain a `userInfo` dictionary.

Availability

Available in Mac OS X v10.2 with Safari 1.0 and later.

Available in Mac OS X v10.2.7 and later.

See Also

- [estimatedProgress](#) (page 35)

Declared In

WebView.h

Document Revision History

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

Date	Notes
2009-04-08	Made minor edits throughout.
2008-10-15	Minor edits throughout.
2008-06-19	Updated for Mac OS X v10.5.
2006-07-31	Includes new API for Mac OS X v.10.5
2006-05-23	First publication of this content as a separate document.

REVISION HISTORY

Document Revision History

Index

A

`alignCenter`: instance method [22](#)
`alignJustified`: instance method [23](#)
`alignLeft`: instance method [23](#)
`alignRight`: instance method [24](#)
`applicationNameForUserAgent`: instance method [24](#)
`applyStyle`: instance method [24](#)

B

`backForwardList`: instance method [25](#)

C

`canGoBack`: instance method [25](#)
`canGoForward`: instance method [26](#)
`canMakeTextLarger`: instance method [26](#)
`canMakeTextSmaller`: instance method [27](#)
`canMakeTextStandardSize`: instance method [27](#)
`canShowMIMETypeAsHTML`: class method [19](#)
`canShowMIMEType`: class method [19](#)
`changeAttributes`: instance method [27](#)
`changeColor`: instance method [28](#)
`changeDocumentBackgroundColor`: instance method [28](#)
`changeFont`: instance method [28](#)
`checkSpelling`: instance method [29](#)
`close`: instance method [29](#)
`computedStyleForElement:pseudoElement`: instance method [30](#)
`copy`: instance method [30](#)
`copyFont`: instance method [31](#)
`customTextEncodingName`: instance method [31](#)
`customUserAgent`: instance method [31](#)
`cut`: instance method [32](#)

D

`delete`: instance method [32](#)
`deleteSelection`: instance method [33](#)
`downloadDelegate`: instance method [33](#)
`drawsBackground`: instance method [34](#)

E

`editableDOMRangeForPoint`: instance method [34](#)
`editingDelegate`: instance method [34](#)
`elementAtPoint`: instance method [35](#)
`estimatedProgress`: instance method [35](#)

F

`frameLoadDelegate`: instance method [36](#)

G

`goBack`: instance method [36](#)
`goBack`: instance method [36](#)
`goForward`: instance method [37](#)
`goForward`: instance method [37](#)
`goToBackForwardItem`: instance method [38](#)
`groupName`: instance method [38](#)

H

`hostWindow`: instance method [38](#)

I

`initWithFrame:frameName:groupName:` **instance method 39**
`isContinuousSpellCheckingEnabled` **instance method 39**
`isEditable` **instance method 40**
`isLoading` **instance method 40**

M

`mainFrame` **instance method 40**
`mainFrameDocument` **instance method 41**
`mainFrameIcon` **instance method 41**
`mainFrameTitle` **instance method 41**
`mainFrameURL` **instance method 42**
`maintainsInactiveSelection` **instance method 42**
`makeTextLarger:` **instance method 42**
`makeTextSmaller:` **instance method 43**
`makeTextStandardSize:` **instance method 43**
`mediaStyle` **instance method 43**
`MIMETypesShownAsHTML` **class method 20**
`moveDragCaretToPoint:` **instance method 44**
`moveToBeginningOfSentenceAndModifySelection:` **instance method 44**
`moveToBeginningOfSentence:` **instance method 44**
`moveToEndOfSentenceAndModifySelection:` **instance method 45**
`moveToEndOfSentence:` **instance method 45**

P

`pasteAsPlainText:` **instance method 46**
`pasteAsRichText:` **instance method 46**
`pasteboardTypesForElement:` **instance method 47**
`pasteboardTypesForSelection` **instance method 47**
`paste:` **instance method 45**
`pasteFont:` **instance method 47**
`performFindPanelAction:` **instance method 48**
`policyDelegate` **instance method 48**
`preferences` **instance method 48**
`preferencesIdentifier` **instance method 49**

R

`registerURLSchemeAsLocal:` **class method 20**
`registerViewClass:representationClass:forMIMETYPE:` **class method 20**

`reload:` **instance method 49**
`removeDragCaret` **instance method 50**
`replaceSelectionWithArchive:` **instance method 50**
`replaceSelectionWithMarkupString:` **instance method 50**
`replaceSelectionWithNode:` **instance method 51**
`replaceSelectionWithText:` **instance method 52**
`resourceLoadDelegate` **instance method 52**

S

`searchFor:direction:caseSensitive:wrap:` **instance method 52**
`selectedDOMRange` **instance method 53**
`selectedFrame` **instance method 54**
`selectionAffinity` **instance method 54**
`selectSentence:` **instance method 54**
`setApplicationNameForUserAgent:` **instance method 55**
`setContinuousSpellCheckingEnabled:` **instance method 55**
`setCustomTextEncodingName:` **instance method 55**
`setCustomUserAgent:` **instance method 56**
`setDownloadDelegate:` **instance method 56**
`setDrawsBackground:` **instance method 57**
`setEditable:` **instance method 57**
`setEditingDelegate:` **instance method 58**
`setFrameLoadDelegate:` **instance method 58**
`setGroupName:` **instance method 58**
`setHostWindow:` **instance method 59**
`setMainFrameURL:` **instance method 59**
`setMaintainsBackForwardList:` **instance method 60**
`setMediaStyle:` **instance method 60**
`setMIMETypesShownAsHTML:` **class method 21**
`setPolicyDelegate:` **instance method 61**
`setPreferences:` **instance method 61**
`setPreferencesIdentifier:` **instance method 61**
`setResourceLoadDelegate:` **instance method 62**
`setSelectedDOMRange:affinity:` **instance method 62**
`setShouldCloseWithWindow:` **instance method 63**
`setSmartInsertDeleteEnabled:` **instance method 63**
`setTextSizeMultiplier:` **instance method 64**
`setTypingStyle:` **instance method 64**
`setUIDelegate:` **instance method 64**
`shouldCloseWithWindow` **instance method 65**
`showGuessPanel:` **instance method 65**
`smartInsertDeleteEnabled` **instance method 65**
`spellCheckerDocumentTag` **instance method 66**
`startSpeaking:` **instance method 66**
`stopLoading:` **instance method 67**
`stopSpeaking:` **instance method 67**

stringByEvaluatingJavaScriptFromString:
 instance method 67
 styleDeclarationWithText: **instance method 68**
 supportsTextEncoding **instance method 68**

T

takeStringURLFrom: **instance method 69**
 textSizeMultiplier **instance method 69**
 toggleContinuousSpellChecking: **instance method 69**
 toggleSmartInsertDelete: **instance method 70**
 typingStyle **instance method 70**

U

UIDelegate **instance method 70**
 undoManager **instance method 71**
 URLFromPasteboard: **class method 21**
 URLTitleFromPasteboard: **class method 22**
 userAgentForURL: **instance method 71**

W

WebElementDOMNodeKey **constant 73**
 WebElementFrameKey **constant 73**
 WebElementImageAltStringKey **constant 73**
 WebElementImageKey **constant 73**
 WebElementImageRectKey **constant 73**
 WebElementImageURLKey **constant 73**
 WebElementIsSelectedKey **constant 73**
 WebElementLinkLabelKey **constant 73**
 WebElementLinkTargetFrameKey **constant 73**
 WebElementLinkTitleKey **constant 73**
 WebElementLinkURLKey **constant 73**
 WebViewDidBeginEditingNotification **notification 74**
 WebViewDidChangeNotification **notification 74**
 WebViewDidChangeSelectionNotification **notification 74**
 WebViewDidChangeTypingStyleNotification **notification 74**
 WebViewDidEndEditingNotification **notification 75**
 WebViewProgressEstimateChangedNotification **notification 75**
 WebViewProgressFinishedNotification **notification 75**
 WebViewProgressStartedNotification **notification 75**

windowScriptObject **instance method 71**
 writeElement:withPasteboardTypes:toPasteboard:
 instance method 72
 writeSelectionWithPasteboardTypes:toPasteboard:
 instance method 72