
WebPreferences Class Reference

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



2008-10-15



Apple Inc.
© 2008 Apple Inc.
All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, electronic, photocopying, recording, or otherwise, without prior written permission of Apple Inc., with the following exceptions: Any person is hereby authorized to store documentation on a single computer for personal use only and to print copies of documentation for personal use provided that the documentation contains Apple's copyright notice.

The Apple logo is a trademark of Apple Inc.

Use of the "keyboard" Apple logo (Option-Shift-K) for commercial purposes without the prior written consent of Apple may constitute trademark infringement and unfair competition in violation of federal and state laws.

No licenses, express or implied, are granted with respect to any of the technology described in this document. Apple retains all intellectual property rights associated with the technology described in this document. This document is intended to assist application developers to develop applications only for Apple-labeled computers.

Every effort has been made to ensure that the information in this document is accurate. Apple is not responsible for typographical errors.

Apple Inc.
1 Infinite Loop
Cupertino, CA 95014
408-996-1010

Apple, the Apple logo, Cocoa, Logic, Mac, Mac OS, Objective-C, Pages, and Safari are trademarks of Apple Inc., registered in the United States and other countries.

Java and all Java-based trademarks are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

Simultaneously published in the United States and Canada.

Even though Apple has reviewed this document, APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS DOCUMENT IS

PROVIDED "AS IS," AND YOU, THE READER, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.

IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT OR INACCURACY IN THIS DOCUMENT, even if advised of the possibility of such damages.

THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESS OR IMPLIED. No Apple dealer, agent, or employee is authorized to make any modification, extension, or addition to this warranty.

Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.

Contents

WebPreferences Class Reference 5

Overview	5
Adopted Protocols	6
Tasks	6
Getting the Standard Preferences	6
Initializing Preferences	6
Getting the Identifier	6
Saving Preferences to the User Defaults Database	6
Enabling Java	6
Enabling JavaScript	7
Enabling Plug-ins	7
Enabling Style Sheets	7
Getting and Setting Fonts	7
Getting and Setting Font Sizes	8
Getting and Setting Text Encoding	8
Handling Images	8
Printing Backgrounds	9
Enabling Private Browsing	9
Controlling User Focus	9
Caching	9
Class Methods	10
standardPreferences	10
Instance Methods	10
allowsAnimatedImageLooping	10
allowsAnimatedImages	10
arePlugInsEnabled	11
autosaves	11
cacheModel	11
cursiveFontFamily	12
defaultFixedFontSize	12
defaultFontSize	12
defaultTextEncodingName	13
fantasyFontFamily	13
fixedFontFamily	13
identifier	14
initWithIdentifier:	14
isJavaEnabled	15
isJavaScriptEnabled	15
javaScriptCanOpenWindowsAutomatically	15
loadImagesAutomatically	16
minimumFontSize	16

- minimumLogicalFontSize 16
- privateBrowsingEnabled 17
- sansSerifFontFamily 17
- serifFontFamily 17
- setAllowsAnimatedImageLooping: 18
- setAllowsAnimatedImages: 18
- setAutosaves: 19
- setCacheModel: 19
- setCursiveFontFamily: 19
- setDefaultFixedFontSize: 20
- setDefaultFontSize: 20
- setDefaultTextEncodingName: 21
- setFantasyFontFamily: 21
- setFixedFontFamily: 21
- setJavaEnabled: 21
- setJavaScriptCanOpenWindowsAutomatically: 22
- setJavaScriptEnabled: 22
- setLoadsImagesAutomatically: 23
- setMinimumFontSize: 23
- setMinimumLogicalFontSize: 23
- setPluginsEnabled: 24
- setPrivateBrowsingEnabled: 24
- setSansSerifFontFamily: 25
- setSerifFontFamily: 25
- setShouldPrintBackgrounds: 25
- setStandardFontFamily: 26
- setTabsToLinks: 26
- setUserStyleSheetEnabled: 26
- setUserStyleSheetLocation: 27
- setUsesPageCache: 27
- shouldPrintBackgrounds 27
- standardFontFamily 28
- tabsToLinks 28
- userStyleSheetEnabled 28
- userStyleSheetLocation 29
- usesPageCache 29
- Constants 29
 - WebCacheModel 29
- Notifications 30
 - WebPreferencesChangedNotification 30

Document Revision History 31

Index 33

WebPreferences Class Reference

Inherits from	NSObject
Conforms to	NSCoding 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	WebPreferences.h

Overview

WebPreferences encapsulates the preferences you can change per WebView object. These preferences include font, text encoding, and image settings. Normally a WebView object uses the standard preferences returned by the [standardPreferences](#) (page 10) class method. However, you can modify the preferences for individual WebView instances too. Use the `setPreferencesIdentifier:WebView` method to change a WebView object's preferences, or to share preferences between WebView objects. Use the `setAutosaves:` (page 19) method to specify if the preferences object should be automatically saved to the user defaults database.

WebPreferences also manages the font preferences for a web view. You can set custom font families for each of the primary web font styles (standard, serif, sans-serif, cursive, and fantasy) as well as their font sizes. The font size preferences alter the display font sizes in a certain way. If the HTML or CSS in the web view's content specifies font sizes in a relative fashion (such as `font-size=-1` in HTML or `font-size: medium` in CSS), the default font size settings (set by the font size methods prefaced with "default") have an effect. They do not have an effect for font sizes specified absolutely. The values specified by the minimum font size settings (set by the font size methods prefaced with "minimum") override all the HTML and CSS font size definitions, and so have an effect on the entirety of the content. The values specified by the minimum logical font size settings (set by the font size methods prefaced with "minimumLogical") affect all relative font size declarations for HTML and CSS, but also override any CSS font size declarations in the content, whether they are relative or absolute.

The font size for a web view is different than its logical font size. The minimum logical font size, for example, is the absolute minimum size at which the font will display onscreen. This is meant to be a functional boundary and not a style boundary. For example, the default value for a web view's minimum logical font size is 9 points, because typical web content looks good on Mac OS X at font sizes of 9 point and above. The constraint assures that web content will always look good in a web view. If you know that your content will look good only at 12 points or above, you should change the minimum font size to 12 points and leave the minimum

logical font size alone. This will assure that your content will never display at sizes less than 12 points, but the functional font size boundary of the web view will remain at 9 points to prevent any chance of displaying unnecessarily small text.

Adopted Protocols

NSCoding

```
encodeWithCoder:
initWithCoder:
```

Tasks

Getting the Standard Preferences

+ [standardPreferences](#) (page 10)

Returns the standard set of preferences that may be used by all WebView objects.

Initializing Preferences

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

Returns an initialized WebPreferences object, creating one if it does not exist.

Getting the Identifier

- [identifier](#) (page 14)

Returns the receiver's identifier.

Saving Preferences to the User Defaults Database

- [autosaves](#) (page 11)

Returns whether or not the receiver's attributes are automatically stored in the user defaults database.

- [setAutosaves:](#) (page 19)

Sets whether or not the receiver's attributes are stored in the user defaults database.

Enabling Java

- [setJavaEnabled:](#) (page 21)

Sets whether or not the web view allows Java.

- [isJavaEnabled](#) (page 15)

Returns whether or not Java is enabled for the web view.

Enabling JavaScript

- [setJavaScriptEnabled:](#) (page 22)
Sets whether or not the web view allows JavaScript.
- [isJavaScriptEnabled](#) (page 15)
Returns whether or not JavaScript is enabled for the web view.
- [setJavaScriptCanOpenWindowsAutomatically:](#) (page 22)
Sets whether or not the web view allows JavaScript to open windows automatically.
- [javaScriptCanOpenWindowsAutomatically](#) (page 15)
Returns whether or not JavaScript can open windows automatically for the web view.

Enabling Plug-ins

- [setPlugInsEnabled:](#) (page 24)
Sets whether or not the web view allows plug-ins.
- [arePlugInsEnabled](#) (page 11)
Returns whether or not the web view allows plug-ins.

Enabling Style Sheets

- [setUserStyleSheetEnabled:](#) (page 26)
Sets whether or not the web view enables user style sheets.
- [userStyleSheetEnabled](#) (page 28)
Returns whether the web view enables user style sheets.
- [setUserStyleSheetLocation:](#) (page 27)
Sets the location of the user style sheet.
- [userStyleSheetLocation](#) (page 29)
Returns the location of the user style sheet.

Getting and Setting Fonts

- [setCursiveFontFamily:](#) (page 19)
Sets the cursive font family of the web view.
- [cursiveFontFamily](#) (page 12)
Returns the cursive font family for the web view.
- [setFantasyFontFamily:](#) (page 21)
Sets the fantasy font family of the web view.
- [fantasyFontFamily](#) (page 13)
Returns the fantasy font family for the web view.
- [setFixedFontFamily:](#) (page 21)
Sets the fixed font family of the web view.
- [fixedFontFamily](#) (page 13)
Returns the fixed font family for the web view.

- [setSansSerifFontFamily:](#) (page 25)
Sets the sans serif font family of the web view.
- [sansSerifFontFamily](#) (page 17)
Returns the sans serif font family for the web view.
- [setSerifFontFamily:](#) (page 25)
Sets the serif font family of the web view.
- [serifFontFamily](#) (page 17)
Returns the serif font family for the web view.
- [setStandardFontFamily:](#) (page 26)
Sets the standard font family of the web view.
- [standardFontFamily](#) (page 28)
Returns the standard font family used by the web view.

Getting and Setting Font Sizes

- [setDefaultFixedFontSize:](#) (page 20)
Sets the default fixed font size of the web view.
- [defaultFixedFontSize](#) (page 12)
Returns the default fixed font size for the web view.
- [setDefaultFontSize:](#) (page 20)
Sets the default font size of the web view.
- [defaultFontSize](#) (page 12)
Returns the default font size for the web view.
- [setMinimumFontSize:](#) (page 23)
Sets the minimum font size of the web view.
- [minimumFontSize](#) (page 16)
Returns the minimum font size for the web view.
- [setMinimumLogicalFontSize:](#) (page 23)
Sets the minimum logical font size of the web view.
- [minimumLogicalFontSize](#) (page 16)
Returns the minimum logical font size for the web view.

Getting and Setting Text Encoding

- [setDefaultTextEncodingName:](#) (page 21)
Sets the default text encoding of the web view.
- [defaultTextEncodingName](#) (page 13)
Returns the default text encoding for the web view.

Handling Images

- [setAllowsAnimatedImageLooping:](#) (page 18)
Sets whether or not the receiver allows animated images to loop.

- [allowsAnimatedImageLooping](#) (page 10)
Returns whether or not the web view allows animated images to loop.
- [setAllowsAnimatedImages:](#) (page 18)
Sets whether or not the receiver allows animated images.
- [allowsAnimatedImages](#) (page 10)
Returns whether or not the web view allows animated images.
- [setLoadsImagesAutomatically:](#) (page 23)
Sets whether or not the web view allows images to be loaded automatically.
- [loadsImagesAutomatically](#) (page 16)
Returns whether images are loaded automatically for the web view.

Printing Backgrounds

- [setShouldPrintBackgrounds:](#) (page 25)
Sets whether or not the web view should include backgrounds when printing.
- [shouldPrintBackgrounds](#) (page 27)
Returns whether the web view should include backgrounds when printing.

Enabling Private Browsing

- [privateBrowsingEnabled](#) (page 17)
Returns whether or not private browsing is enabled.
- [setPrivateBrowsingEnabled:](#) (page 24)
Sets whether or not private browsing is enabled.

Controlling User Focus

- [tabsToLinks](#) (page 28)
Returns whether or not the tab key will focus links.
- [setTabsToLinks:](#) (page 26)
Sets whether or not the web view will focus control on links when tabbing.

Caching

- [setUsesPageCache:](#) (page 27)
Sets whether the web views associated with the receiver should use the shared page cache.
- [usesPageCache](#) (page 29)
Returns whether the web views associated with the receiver should use the shared page cache.
- [setCacheModel:](#) (page 19)
Sets the cache model for the web views associated with the receiver.
- [cacheModel](#) (page 11)
Returns the cache model for a web view.

Class Methods

standardPreferences

Returns the standard set of preferences that may be used by all WebView objects.

```
+ (WebPreferences *)standardPreferences
```

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

WebPreferences.h

Instance Methods

allowsAnimatedImageLooping

Returns whether or not the web view allows animated images to loop.

```
- (BOOL)allowsAnimatedImageLooping
```

Discussion

The number of times that an image loops is determined by parameters of the image file itself and cannot be set in the web view.

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

- [allowsAnimatedImageLooping](#) (page 10)
- [setAllowsAnimatedImages:](#) (page 18)
- [setAllowsAnimatedImages:](#) (page 18)
- [loadsImagesAutomatically](#) (page 16)
- [setLoadsImagesAutomatically:](#) (page 23)

Declared In

WebPreferences.h

allowsAnimatedImages

Returns whether or not the web view allows animated images.

```
- (BOOL)allowsAnimatedImages
```

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

- [allowsAnimatedImageLooping](#) (page 10)
- [setAllowsAnimatedImages:](#) (page 18)
- [loadsImagesAutomatically](#) (page 16)
- [setLoadsImagesAutomatically:](#) (page 23)

Declared In

WebPreferences.h

arePluginsEnabled

Returns whether or not the web view allows plug-ins.

- (BOOL)arePluginsEnabled

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

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

Declared In

WebPreferences.h

autosaves

Returns whether or not the receiver's attributes are automatically stored in the user defaults database.

- (BOOL)autosaves

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

- [setAutosaves:](#) (page 19)

Declared In

WebPreferences.h

cacheModel

Returns the cache model for a web view.

- (WebCacheModel)cacheModel

Return Value

The cache model for the web views associated with the receiver. Possible values are described in [WebCacheModel](#) (page 29).

Availability

Available in Mac OS X v10.4.11 and later.

See Also

- [setCacheModel](#): (page 19)

Declared In

WebPreferences.h

cursiveFontFamily

Returns the cursive font family for the web view.

- (NSString *)cursiveFontFamily

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

- [setCursiveFontFamily](#): (page 19)

Declared In

WebPreferences.h

defaultFixedFontSize

Returns the default fixed font size for the web view.

- (int)defaultFixedFontSize

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

- [setDefaultFixedFontSize](#): (page 20)

Declared In

WebPreferences.h

defaultFontSize

Returns the default font size for the web view.

- (int)defaultFontSize

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

- [setDefaultFontSize:](#) (page 20)

Declared In

WebPreferences.h

defaultTextEncodingName

Returns the default text encoding for the web view.

- (NSString *)defaultTextEncodingName

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

- [setDefaultTextEncodingName:](#) (page 21)

Declared In

WebPreferences.h

fantasyFontFamily

Returns the fantasy font family for the web view.

- (NSString *)fantasyFontFamily

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

- [setFantasyFontFamily:](#) (page 21)

Declared In

WebPreferences.h

fixedFontFamily

Returns the fixed font family for the web view.

- (NSString *)fixedFontFamily

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

- [setFixedFontFamily:](#) (page 21)

Declared In

WebPreferences.h

identifier

Returns the receiver's identifier.

```
- (NSString *)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

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

Declared In

WebPreferences.h

initWithIdentifier:

Returns an initialized WebPreferences object, creating one if it does not exist.

```
- (id)initWithIdentifier:(NSString *)anIdentifier
```

Discussion

This method returns either the receiver initialized with *anIdentifier*, or another WebPreferences object matching *anIdentifier* if it exists.

The *anIdentifier* argument should be a unique identifier—it will be prepended to the keys used to store the receiver's attributes in the user defaults database. WebView objects can share instances of WebPreferences by using the same preferences identifier.

Typically, you do not invoke this method directly. Instead, you set the preferences identifier by sending a `setPreferencesIdentifier:` message to your WebView object. This method is the designated initializer for the WebPreferences 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.

See Also

- [identifier](#) (page 14)

Declared In

WebPreferences.h

isJavaEnabled

Returns whether or not Java is enabled for the web view.

- (BOOL)isJavaEnabled

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

- [setJavaEnabled:](#) (page 21)

Declared In

WebPreferences.h

isJavaScriptEnabled

Returns whether or not JavaScript is enabled for the web view.

- (BOOL)isJavaScriptEnabled

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

- [setJavaScriptEnabled:](#) (page 22)

Declared In

WebPreferences.h

javaScriptCanOpenWindowsAutomatically

Returns whether or not JavaScript can open windows automatically for the web view.

- (BOOL)javaScriptCanOpenWindowsAutomatically

Discussion

Explicit calls to a JavaScript window opener that are activated by user action (such as a button click) are not affected by this setting.

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

- [setJavaScriptCanOpenWindowsAutomatically:](#) (page 22)

Declared In

WebPreferences.h

loadsImagesAutomatically

Returns whether images are loaded automatically for the web view.

- (BOOL)loadsImagesAutomatically

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

- [setLoadsImagesAutomatically:](#) (page 23)
- [allowsAnimatedImages](#) (page 10)
- [setAllowsAnimatedImages:](#) (page 18)
- [allowsAnimatedImageLooping](#) (page 10)
- [setAutosaves:](#) (page 19)

Declared In

WebPreferences.h

minimumFontSize

Returns the minimum font size for the web view.

- (int)minimumFontSize

Discussion

The default value is 1, meaning the minimum font size will only be constrained by the minimum logical font size.

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

- [setMinimumFontSize:](#) (page 23)

Declared In

WebPreferences.h

minimumLogicalFontSize

Returns the minimum logical font size for the web view.

- (int)minimumLogicalFontSize

Discussion

The minimum logical font size is the smallest font size that will display in a WebKit view when the content's font size is not explicitly defined. Its default value is a 9-point font size.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- [setMinimumLogicalFontSize:](#) (page 23)

Declared In

WebPreferences.h

privateBrowsingEnabled

Returns whether or not private browsing is enabled.

- (BOOL)privateBrowsingEnabled

Discussion

Private browsing prevents the web view from maintaining any history, cache, or AutoFill information for the pages being visited.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

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

Declared In

WebPreferences.h

sansSerifFontFamily

Returns the sans serif font family for the web view.

- (NSString *)sansSerifFontFamily

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

- [setSansSerifFontFamily:](#) (page 25)

Declared In

WebPreferences.h

serifFontFamily

Returns the serif font family for the web view.

- (NSString *)serifFontFamily

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

- [setSerifFontFamily:](#) (page 25)

Declared In

WebPreferences.h

setAllowsAnimatedImageLooping:

Sets whether or not the receiver allows animated images to loop.

- (void)setAllowsAnimatedImageLooping:(BOOL)flag

Discussion

If *flag* is YES the web view will loop animated images, otherwise it will display them as static images. The number of times that an image loops is determined by parameters of the image file itself and cannot be set in the web view.

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

- [allowsAnimatedImageLooping](#) (page 10)
- [allowsAnimatedImages](#) (page 10)
- [setAllowsAnimatedImages:](#) (page 18)
- [loadsImagesAutomatically](#) (page 16)
- [setLoadsImagesAutomatically:](#) (page 23)

Declared In

WebPreferences.h

setAllowsAnimatedImages:

Sets whether or not the receiver allows animated images.

- (void)setAllowsAnimatedImages:(BOOL)flag

Discussion

If *flag* is YES the web view allows animated images, otherwise it does not.

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

- [allowsAnimatedImages](#) (page 10)
- [setAutosaves:](#) (page 19)
- [allowsAnimatedImageLooping](#) (page 10)
- [loadsImagesAutomatically](#) (page 16)
- [setLoadsImagesAutomatically:](#) (page 23)

Declared In

WebPreferences.h

setAutosaves:

Sets whether or not the receiver's attributes are stored in the user defaults database.

- (void)setAutosaves:(BOOL)flag

Discussion

If *flag* is YES the receiver's attributes are automatically stored in the user defaults database, otherwise they are not. The default value is 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

- [autosaves](#) (page 11)

Declared In

WebPreferences.h

setCacheModel:

Sets the cache model for the web views associated with the receiver.

- (void)setCacheModel:(WebCacheModel)cacheModel

Parameters

cacheModel

The cache model for the web views associated with the receiver. Possible values are described in [WebCacheModel](#) (page 29).

Discussion

Set this property to optimize WebKit's cache footprint (on disk and in memory) to best fit the use of the web view. If a web view is used only for a single webpage, use the [WebCacheModelDocumentViewer](#) (page 30) constant instead.

Availability

Available in Mac OS X v10.4.11 and later.

See Also

- [cacheModel](#) (page 11)

Declared In

WebPreferences.h

setCursiveFontFamily:

Sets the cursive font family of the web view.

- (void)setCursiveFontFamily:(NSString *)*family*

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

- [cursiveFontFamily](#) (page 12)

Declared In

WebPreferences.h

setDefaultFixedFontSize:

Sets the default fixed font size of the web view.

- (void)setDefaultFixedFontSize:(int)*size*

Discussion

The font size specified by *size* should always be greater than zero.

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

- [defaultFixedFontSize](#) (page 12)

Declared In

WebPreferences.h

setDefaultFontSize:

Sets the default font size of the web view.

- (void)setDefaultFontSize:(int)*size*

Discussion

The font size specified by *size* should always be greater than zero.

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

- [defaultFontSize](#) (page 12)

Declared In

WebPreferences.h

setDefaultTextEncodingName:

Sets the default text encoding of the web view.

```
- (void)setDefaultTextEncodingName:(NSString *)encoding
```

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

- [defaultTextEncodingName](#) (page 13)

Declared In

WebPreferences.h

setFontFamily:

Sets the fantasy font family of the web view.

```
- (void)setFontFamily:(NSString *)family
```

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

- [fontFamily](#) (page 13)

Declared In

WebPreferences.h

setFontFixedFamily:

Sets the fixed font family of the web view.

```
- (void)setFontFixedFamily:(NSString *)family
```

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

- [fontFixedFamily](#) (page 13)

Declared In

WebPreferences.h

setJavaEnabled:

Sets whether or not the web view allows Java.

- (void)setJavaEnabled:(BOOL)flag

Discussion

If *flag* is YES the web view allows Java, otherwise it does not.

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

- [isJavaEnabled](#) (page 15)

Declared In

WebPreferences.h

setJavaScriptCanOpenWindowsAutomatically:

Sets whether or not the web view allows JavaScript to open windows automatically.

- (void)setJavaScriptCanOpenWindowsAutomatically:(BOOL)flag

Discussion

If *flag* is YES the web view allows JavaScript to open windows automatically, otherwise it does not. Explicit calls to a JavaScript window opener that are activated by user action (such as a button click) are not affected by this setting.

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

- [JavaScriptCanOpenWindowsAutomatically](#) (page 15)

Declared In

WebPreferences.h

setJavaScriptEnabled:

Sets whether or not the web view allows JavaScript.

- (void)setJavaScriptEnabled:(BOOL)flag

Discussion

If *flag* is YES the web view allows JavaScript, otherwise it does not.

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

- [isJavaScriptEnabled](#) (page 15)

Declared In

WebPreferences.h

setLoadsImagesAutomatically:

Sets whether or not the web view allows images to be loaded automatically.

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

Discussion

If *flag* is YES the web view allows images to be loaded automatically, otherwise it does not.

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

- [loadsImagesAutomatically](#) (page 16)
- [allowsAnimatedImages](#) (page 10)
- [setAllowsAnimatedImages:](#) (page 18)
- [allowsAnimatedImageLooping](#) (page 10)
- [setAutosaves:](#) (page 19)

Declared In

WebPreferences.h

setMinimumFontSize:

Sets the minimum font size of the web view.

```
- (void)setMinimumFontSize:(int)size
```

Discussion

You should use this method to explicitly set the minimum display font size for the web view. The font size specified by *size* should always be greater than zero.

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

- [minimumFontSize](#) (page 16)

Declared In

WebPreferences.h

setMinimumLogicalFontSize:

Sets the minimum logical font size of the web view.

```
- (void)setMinimumLogicalFontSize:(int)size
```

Discussion

The minimum logical font size is the smallest font size that will display in a web view when the content's font size is not explicitly defined. Most clients will not want to use this method; rather, explicitly set the minimum display font size using the [setMinimumFontSize:](#) (page 23) method.

The font size specified by *size* should always be greater than zero.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- [minimumLogicalFontSize](#) (page 16)

Declared In

WebPreferences.h

setPluginsEnabled:

Sets whether or not the web view allows plug-ins.

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

Discussion

If *flag* is YES the web view allows plug-ins, otherwise it does not.

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

- [arePluginsEnabled](#) (page 11)

Declared In

WebPreferences.h

setPrivateBrowsingEnabled:

Sets whether or not private browsing is enabled.

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

Discussion

If *flag* is YES, the web view will not store information about the websites the user visits. Private browsing prevents the web view from maintaining any history, cache, or AutoFill information for the pages being visited.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- [privateBrowsingEnabled](#) (page 17)

Declared In

WebPreferences.h

setSansSerifFontFamily:

Sets the sans serif font family of the web view.

- (void)setSansSerifFontFamily:(NSString *)*family*

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

- [sansSerifFontFamily](#) (page 17)

Declared In

WebPreferences.h

setSerifFontFamily:

Sets the serif font family of the web view.

- (void)setSerifFontFamily:(NSString *)*family*

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

- [serifFontFamily](#) (page 17)

Declared In

WebPreferences.h

setShouldPrintBackgrounds:

Sets whether or not the web view should include backgrounds when printing.

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

Discussion

If *flag* is YES the web view prints the backgrounds, otherwise it does not.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- [shouldPrintBackgrounds](#) (page 27)

Declared In

WebPreferences.h

setStandardFontFamily:

Sets the standard font family of the web view.

```
- (void)setStandardFontFamily:(NSString *)family
```

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

- [standardFontFamily](#) (page 28)

Declared In

WebPreferences.h

setTabsToLinks:

Sets whether or not the web view will focus control on links when tabbing.

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

Discussion

If *flag* is YES the web view tabs to links, otherwise it does not.

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- [tabsToLinks](#) (page 28)

Declared In

WebPreferences.h

setUserStyleSheetEnabled:

Sets whether or not the web view enables user style sheets.

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

Discussion

If *flag* is YES the web view enables user style sheets, otherwise it does not.

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

- [userStyleSheetEnabled](#) (page 28)

Declared In

WebPreferences.h

setUserStyleSheetLocation:

Sets the location of the user style sheet.

- (void)setUserStyleSheetLocation:(NSURL *)URL

Discussion

The user style sheet will override all existing CSS definitions on the page.

[setUserStyleSheetEnabled:](#) (page 26) must have already been set to YES for this method to have an effect.

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

- [userStyleSheetLocation](#) (page 29)

Declared In

WebPreferences.h

setUsesPageCache:

Sets whether the web views associated with the receiver should use the shared page cache.

- (void)setUsesPageCache:(BOOL)usesPageCache

Parameters

usesPageCache

YES if the web views should use a page cache; otherwise, NO.

Discussion

Pages are cached when they are added to a back-forward list, and removed from the cache when they are removed from a back-forward list. Because the page cache is global, caching a page in one back-forward list may cause a page in another back-forward list to be removed from the cache.

Availability

Available in Mac OS X v10.4.11 and later.

See Also

- [usesPageCache](#) (page 29)

Declared In

WebPreferences.h

shouldPrintBackgrounds

Returns whether the web view should include backgrounds when printing.

- (BOOL)shouldPrintBackgrounds

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- [setShouldPrintBackgrounds:](#) (page 25)

Declared In

WebPreferences.h

standardFontFamily

Returns the standard font family used by the web view.

- (NSString *)standardFontFamily

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

- [setStandardFontFamily:](#) (page 26)

Declared In

WebPreferences.h

tabsToLinks

Returns whether or not the tab key will focus links.

- (BOOL)tabsToLinks

Availability

Available in Mac OS X v10.3.9 and later.

See Also

- [setTabsToLinks:](#) (page 26)

Declared In

WebPreferences.h

userStyleSheetEnabled

Returns whether the web view enables user style sheets.

- (BOOL)userStyleSheetEnabled

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

- [setUserStyleSheetEnabled:](#) (page 26)

Declared In

WebPreferences.h

userStyleSheetLocation

Returns the location of the user style sheet.

- (NSURL *)userStyleSheetLocation

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

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

Declared In

WebPreferences.h

usesPageCache

Returns whether the web views associated with the receiver should use the shared page cache.

- (BOOL)usesPageCache

Return Value

YES if the web views should use a page cache; otherwise, NO.

Availability

Available in Mac OS X v10.4.11 and later.

See Also

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

Declared In

WebPreferences.h

Constants

WebCacheModel

Specifies the caching model for a web view.

```
enum {
    WebCacheModelDocumentViewer = 0,
    WebCacheModelDocumentBrowser = 1,
    WebCacheModelPrimaryWebBrowser = 2
};
typedef NSUInteger WebCacheModel;
```

Constants**WebCacheModelDocumentViewer**

Releases resources when they are no longer referenced and caches remote resources on disk. This model is appropriate for displaying a static document with no navigation user interface. This is the most memory-efficient model.

Available in Mac OS X v10.5 and later.

Declared in `WebPreferences.h`.

WebCacheModelDocumentBrowser

Caches a reasonable number of resources and previously viewed documents in memory and on disk. This model is appropriate for displaying and navigating between multiple documents.

Available in Mac OS X v10.5 and later.

Declared in `WebPreferences.h`.

WebCacheModelPrimaryWebBrowser

Caches a large number of resources and previously viewed documents in memory and on disk. This model is appropriate for a web view that behaves like a web browser.

Available in Mac OS X v10.5 and later.

Declared in `WebPreferences.h`.

Availability

Available in Mac OS X v10.4.11 and later.

Declared In

`WebPreferences.h`

Notifications

WebPreferencesChangedNotification

Posted when the web preference settings are changed.

The notification object is the `WebPreferences` object that 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.

Declared In

`WebPreferences.h`

Document Revision History

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

Date	Notes
2008-10-15	Minor edits throughout.
2008-06-19	Updated for Mac OS X v10.5.
2006-05-23	First publication of this content as a separate document.

REVISION HISTORY

Document Revision History

Index

A

allowsAnimatedImageLooping **instance method** [10](#)
allowsAnimatedImages **instance method** [10](#)
arePlugInsEnabled **instance method** [11](#)
autosaves **instance method** [11](#)

C

cacheModel **instance method** [11](#)
cursiveFontFamily **instance method** [12](#)

D

defaultFixedFontSize **instance method** [12](#)
defaultFontSize **instance method** [12](#)
defaultTextEncodingName **instance method** [13](#)

F

fantasyFontFamily **instance method** [13](#)
fixedFontFamily **instance method** [13](#)

I

identifier **instance method** [14](#)
initWithIdentifier: **instance method** [14](#)
isJavaEnabled **instance method** [15](#)
isJavaScriptEnabled **instance method** [15](#)

J

javaScriptCanOpenWindowsAutomatically **instance method** [15](#)

L

loadsImagesAutomatically **instance method** [16](#)

M

minimumFontSize **instance method** [16](#)
minimumLogicalFontSize **instance method** [16](#)

P

privateBrowsingEnabled **instance method** [17](#)

S

sansSerifFontFamily **instance method** [17](#)
serifFontFamily **instance method** [17](#)
setAllowsAnimatedImageLooping: **instance method** [18](#)
setAllowsAnimatedImages: **instance method** [18](#)
setAutosaves: **instance method** [19](#)
setCacheModel: **instance method** [19](#)
setCursiveFontFamily: **instance method** [19](#)
setDefaultFixedFontSize: **instance method** [20](#)
setDefaultFontSize: **instance method** [20](#)
setDefaultTextEncodingName: **instance method** [21](#)
setFantasyFontFamily: **instance method** [21](#)
setFixedFontFamily: **instance method** [21](#)
setJavaEnabled: **instance method** [21](#)
setJavaScriptCanOpenWindowsAutomatically: **instance method** [22](#)

setJavaScriptEnabled: **instance method** [22](#)
setLoadsImagesAutomatically: **instance method** [23](#)
setMinimumFontSize: **instance method** [23](#)
setMinimumLogicalFontSize: **instance method** [23](#)
setPlugInsEnabled: **instance method** [24](#)
setPrivateBrowsingEnabled: **instance method** [24](#)
setSansSerifFontFamily: **instance method** [25](#)
setSerifFontFamily: **instance method** [25](#)
setShouldPrintBackgrounds: **instance method** [25](#)
setStandardFontFamily: **instance method** [26](#)
setTabsToLinks: **instance method** [26](#)
setUserStyleSheetEnabled: **instance method** [26](#)
setUserStyleSheetLocation: **instance method** [27](#)
setUsesPageCache: **instance method** [27](#)
shouldPrintBackgrounds **instance method** [27](#)
standardFontFamily **instance method** [28](#)
standardPreferences **class method** [10](#)

T

tabsToLinks **instance method** [28](#)

U

userStyleSheetEnabled **instance method** [28](#)
userStyleSheetLocation **instance method** [29](#)
usesPageCache **instance method** [29](#)

W

WebCacheModel **data type** [29](#)
WebCacheModelDocumentBrowser **constant** [30](#)
WebCacheModelDocumentViewer **constant** [30](#)
WebCacheModelPrimaryWebBrowser **constant** [30](#)
WebPreferencesChangedNotification **notification**
[30](#)