
WebDocumentView Protocol 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, Mac, Mac OS, Objective-C, and Safari are trademarks of Apple Inc., registered in the United States and other countries.

Simultaneously published in the United States and Canada.

Even though Apple has reviewed this document, APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS DOCUMENT IS PROVIDED "AS IS," AND YOU, THE READER, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.

IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY

DEFECT OR INACCURACY IN THIS DOCUMENT, even if advised of the possibility of such damages.

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

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

Contents

WebDocumentView Protocol Reference 5

- Overview 5
- Tasks 5
 - Setting the Data Source 5
 - Controlling the Layout 5
 - Attaching to a Window 6
- Instance Methods 6
 - dataSourceUpdated: 6
 - layout 6
 - setDataSource: 7
 - setNeedsLayout: 7
 - viewDidMoveToHostWindow 7
 - viewWillMoveToHostWindow: 8

Document Revision History 9

Index 11

WebDocumentView Protocol Reference

Conforms to	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	WebDocument.h

Overview

This protocol is adopted by the document view of a `WebFrameView`. You can extend WebKit to support additional MIME types by implementing your own document view and document representation classes to render data for specific MIME types. You register those classes using the `WebFrame.registerViewClass:representationClass:forMIMEType:` method. Classes that adopt this protocol are expected to be subclasses of `NSView`.

Tasks

Setting the Data Source

- [dataSource:](#) (page 7)
Invoked when the data source for this document has been changed.
- [dataSourceUpdated:](#) (page 6)
Invoked when additional data has been received.

Controlling the Layout

- [setNeedsLayout:](#) (page 7)
Sets whether or not the receiver should change its layout.
- [layout](#) (page 6)
Invoked when the receiver should change its layout.

Attaching to a Window

- [viewDidMoveToHostWindow](#) (page 7)
Invoked when a web view's host window is set.
- [viewWillMoveToHostWindow:](#) (page 8)
Invoked when a web view's host window is about to change.

Instance Methods

dataSourceUpdated:

Invoked when additional data has been received.

```
- (void)dataSourceUpdated:(WebDataSource *)dataSource
```

Discussion

The parameter *dataSource* indicates the source of the new data.

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

- [setDataSource:](#) (page 7)

Declared In

WebDocument.h

layout

Invoked when the receiver should change its layout.

```
- (void)layout
```

Discussion

This message is sent to the view as a hint to perform any calculations and update rendering information. For example, at a minimum, the receiver might set the frame rectangle. This method should not perform any drawing operations.

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

- [setNeedsLayout:](#) (page 7)

Declared In

WebDocument.h

setDataSource:

Invoked when the data source for this document has been changed.

- (void)setDataSource:(WebDataSource *)dataSource

Discussion

The parameter *dataSource* contains the new data source for this document.

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

- [dataSourceUpdated:](#) (page 6)

Declared In

WebDocument.h

setNeedsLayout:

Sets whether or not the receiver should change its layout.

- (void)setNeedsLayout:(BOOL)flag

Discussion

If *flag* is YES then the receiver will update its layout. Views conforming to this protocol should implement the `drawRect` method to invoke [layout](#) (page 6) if this flag is YES.

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

WebDocument.h

viewDidMoveToHostWindow

Invoked when a web view's host window is set.

- (void)viewDidMoveToHostWindow

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

- [viewWillMoveToHostWindow:](#) (page 8)

- `setHostWindow:`

Declared In

WebDocument.h

viewWillMoveToHostWindow:

Invoked when a web view's host window is about to change.

- (void)viewWillMoveToHostWindow:(NSWindow *)*hostWindow*

Discussion

The parameter *hostWindow* contains the new host window for the WebView.

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

- [viewDidMoveToHostWindow](#) (page 7)

- `setHostWindow:`

Declared In

WebDocument.h

Document Revision History

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

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

REVISION HISTORY

Document Revision History

Index

D

dataSourceUpdated: [protocol instance method 6](#)

L

layout [protocol instance method 6](#)

S

setDataSource: [protocol instance method 7](#)

setNeedsLayout: [protocol instance method 7](#)

V

viewDidMoveToHostWindow [protocol instance method 7](#)

viewWillMoveToHostWindow: [protocol instance method 8](#)