
PDF Kit Reference Collection

[Graphics & Imaging](#) > Cocoa



2007-12-11



Apple Inc.
© 2004, 2007 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, Mac, Mac OS, Objective-C, Pages, Quartz, and Tiger are trademarks of Apple Inc., registered in the United States and other countries.

Adobe, Acrobat, and PostScript are trademarks or registered trademarks of Adobe Systems Incorporated in the U.S. and/or 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

Introduction **PDF Kit 9**

Introduction 9

Part I **Classes 11**

Chapter 1 **PDFAction Class Reference 13**

Overview 13

Tasks 13

Instance Methods 14

Chapter 2 **PDFActionGoTo Class Reference 15**

Overview 15

Tasks 15

Instance Methods 16

Chapter 3 **PDFActionNamed Class Reference 17**

Overview 17

Tasks 17

Instance Methods 18

Constants 19

Chapter 4 **PDFActionRemoteGoTo Class Reference 21**

Overview 21

Tasks 21

Instance Methods 22

Chapter 5 **PDFActionResetForm Class Reference 25**

Overview 25

Tasks 25

Instance Methods 26

Chapter 6 **PDFActionURL Class Reference 29**

Overview 29

Tasks 29

Instance Methods 30

Chapter 7 **PDFAnnotation Class Reference 33**

Overview 33
Tasks 34
Instance Methods 35

Chapter 8 **PDFAnnotationButtonWidget Class Reference 47**

Overview 47
Tasks 47
Instance Methods 49
Constants 56

Chapter 9 **PDFAnnotationChoiceWidget Class Reference 59**

Overview 59
Tasks 59
Instance Methods 60

Chapter 10 **PDFAnnotationCircle Class Reference 67**

Overview 67
Tasks 67
Instance Methods 67

Chapter 11 **PDFAnnotationFreeText Class Reference 69**

Overview 69
Tasks 69
Instance Methods 70

Chapter 12 **PDFAnnotationInk Class Reference 73**

Overview 73
Tasks 73
Instance Methods 74

Chapter 13 **PDFAnnotationLine Class Reference 75**

Overview 75
Tasks 75
Instance Methods 76
Constants 80

Chapter 14 **PDFAnnotationLink Class Reference 81**

Overview 81
Tasks 81
Instance Methods 82

Chapter 15 **PDFAnnotationMarkup Class Reference 85**

Overview 85
Tasks 85
Instance Methods 86
Constants 87

Chapter 16 **PDFAnnotationPopup Class Reference 89**

Overview 89
Tasks 89
Instance Methods 89

Chapter 17 **PDFAnnotationSquare Class Reference 91**

Overview 91
Tasks 91
Instance Methods 91

Chapter 18 **PDFAnnotationStamp Class Reference 93**

Overview 93
Tasks 93
Instance Methods 93

Chapter 19 **PDFAnnotationText Class Reference 95**

Overview 95
Tasks 95
Instance Methods 96
Constants 97

Chapter 20 **PDFAnnotationTextWidget Class Reference 99**

Overview 99
Tasks 99
Instance Methods 100

Chapter 21 **PDFBorder Class Reference 107**

Overview 107
Tasks 107
Instance Methods 108
Constants 111

Chapter 22 **PDFDestination Class Reference 113**

Overview 113
Tasks 113
Instance Methods 114
Constants 116

Chapter 23 **PDFDocument Class Reference 117**

Overview 117
Tasks 117
Instance Methods 120
Delegate Methods 136
Constants 138
Notifications 140

Chapter 24 **PDFOutline Class Reference 145**

Overview 145
Tasks 145
Instance Methods 147

Chapter 25 **PDFPage Class Reference 153**

Overview 153
Tasks 153
Instance Methods 155
Constants 166

Chapter 26 **PDFSelection Class Reference 167**

Overview 167
Tasks 167
Instance Methods 168

Chapter 27 **PDFThumbnailView Class Reference 175**

Overview 175
Tasks 175

Instance Methods 176

Chapter 28 PDFView Class Reference 183

Overview 183

Tasks 183

Instance Methods 189

Delegate Methods 214

Constants 217

Notifications 218

Document Revision History 223

Index 225

PDF Kit

Framework	Library/Frameworks/Quartz.framework/Frameworks/PDFKit.framework
Header file directories	Library/Frameworks/Quartz.framework/Frameworks/PDFKit.framework/Headers
Declared in	PDFAction.h PDFActionGoTo.h PDFActionNamed.h PDFActionRemoteGoTo.h PDFActionResetForm.h PDFActionURL.h PDFAnnotation.h PDFAnnotationButtonWidget.h PDFAnnotationChoiceWidget.h PDFAnnotationCircle.h PDFAnnotationFreeText.h PDFAnnotationInk.h PDFAnnotationLine.h PDFAnnotationLink.h PDFAnnotationMarkup.h PDFAnnotationPopup.h PDFAnnotationSquare.h PDFAnnotationStamp.h PDFAnnotationText.h PDFAnnotationTextWidget.h PDFBorder.h PDFDestination.h PDFDocument.h PDFOutline.h PDFPage.h PDFSelection.h PDFThumbnailView.h PDFView.h

Introduction

Apple's PDF Kit, introduced with Mac OS X version 10.4, lets you add PDF viewing and navigation to your Mac OS X application with just a few clicks in Interface Builder.

With PDF Kit, your application can access much of the functionality described in the Adobe PDF specification with just a small number of Cocoa classes. The Preview application in Mac OS X 10.4 and later uses PDF Kit for PDF support, so you can use that application as an example of some of PDF Kit's features.

INTRODUCTION

PDF Kit

If you want to go beyond simple PDF viewing, PDF Kit includes a suite of Objective-C utility classes. These utility classes provide lower-level support of PDF features, allowing more control over annotations, selections, and so on.

For more details about PDF Kit and how to implement it in your application, see *PDF Kit Programming Guide*.

Classes

PDFAction Class Reference

Inherits from	NSObject
Conforms to	NSObject (NSObject)
Framework	Library/Frameworks/Quartz.framework/Frameworks/PDFKit.framework
Declared in	PDFKit/PDFAction.h
Availability	Available in Mac OS X v10.5 and later.

Overview

`PDFAction`, a subclass of `NSObject`, represents an action that is performed when, for example, a PDF annotation is activated or an outline item is clicked.

A `PDFAction` object represents an action associated with a PDF element, such as an annotation or a link, that the viewer application can perform. See the Adobe PDF Specification for more about actions and action types.

`PDFAction` is an abstract superclass of the following concrete classes:

- `PDFActionGoTo`
- `PDFActionNamed`
- `PDFActionRemoteGoTo`
- `PDFActionResetForm`
- `PDFActionURL`

Tasks

Getting the Action Type

- [type](#) (page 14)
Returns the type of the action.

Instance Methods

type

Returns the type of the action.

- (NSString *)type

Return Value

The type of the PDF action.

Discussion

The PDF action type returned by this method may not correspond precisely to the name of a `PDFAction` subclass. For example, a `PDFActionURL` object might return “URI” or “Launch,” depending on the original action as defined by the Adobe PDF Specification. In the PDF Kit, these two actions are handled in the single `PDFActionURL` subclass, and the more familiar term “URL” is used instead.

Availability

Available in Mac OS X v10.5 and later.

Declared In

`PDFAction.h`

PDFActionGoTo Class Reference

Inherits from	PDFAction : NSObject
Conforms to	NSCopying NSObject (NSObject)
Framework	Library/Frameworks/Quartz.framework/Frameworks/PDFKit.framework
Declared in	PDFKit/PDFActionGoTo.h
Availability	Available in Mac OS X v10.5 and later.

Overview

PDFActionGoTo, a subclass of PDFAction, defines methods for getting and setting the destination of a go-to action.

A PDFActionGoTo object represents the action of going to a specific location within the PDF document.

Tasks

Accessing the Destination

- [destination](#) (page 16)
Returns the destination associated with the action.
- [setDestination](#) (page 16)
Sets the destination of the go-to action.

Initializing the Action

- [initWithDestination](#) (page 16)
Initializes the go-to action.

Instance Methods

destination

Returns the destination associated with the action.

```
- (PDFDestination *)destination
```

Return Value

The destination specified by the go-to action.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [setDestination](#) (page 16)

Declared In

PDFActionGoTo.h

initWithDestination

Initializes the go-to action.

```
- (id)initWithDestination:(PDFDestination *) destination;
```

Parameters

destination

The destination with which to initialize the go-to action.

Return Value

An initialized PDFActionGoTo instance, or NULL if the object could not be initialized.

Availability

Available in Mac OS X v10.5 and later.

setDestination

Sets the destination of the go-to action.

```
- (void)setDestination:(PDFDestination *)destination
```

Parameters

destination

The destination of the go-to action.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [destination](#) (page 16)

PDFActionNamed Class Reference

Inherits from	PDFAction : NSObject
Conforms to	NSCopying NSObject (NSObject)
Framework	Library/Frameworks/Quartz.framework/Frameworks/PDFKit.framework
Declared in	PDFKit/PDFActionNamed.h
Availability	Available in Mac OS X v10.5 and later.

Overview

`PDFActionNamed` defines methods used to work with actions in PDF documents, some of which are named in the Adobe PDF Specification.

A `PDFActionNamed` object represents an action with a defined name, such as “Go back” or “Zoom in.”

Tasks

Accessing the Name of the Action

- `name` (page 18)
Returns the name of the named action.
- `setName:` (page 18)
Sets the name of the named action.

Initializing the Action

- `initWithName:` (page 18)
Initializes the `PDFActionName` object with the specified named action.

Instance Methods

initWithName:

Initializes the `PDFActionNamed` object with the specified named action.

```
- (id)initWithName:(PDFActionNamedName)name
```

Parameters

name

The action name used to initialize the named action.

Return Value

An initialized `PDFActionNamed` instance, or `NULL` if the object could not be initialized.

Discussion

See “Named Action Names” for the names of named actions you can specify.

Availability

Available in Mac OS X v10.5 and later.

Declared In

`PDFActionNamed.h`

name

Returns the name of the named action.

```
- (PDFActionNamedName)name
```

Return Value

The name of the named action.

Availability

Available in Mac OS X v10.5 and later.

See Also

- `setName`

Declared In

`PDFActionNamed.h`

setName:

Sets the name of the named action.

```
- (void)setName:(PDFActionNamedName)name
```

Parameters

name

The action name to which to set the named action.

Discussion

See “Named Action Names” for the names of named actions you can specify.

Availability

Available in Mac OS X v10.5 and later.

See Also

- name

Declared In

PDFActionNamed.h

Constants

Named Action Names

Names of supported actions.

```
enum
{
    kPDFActionNamedNone = 0,
    kPDFActionNamedNextPage = 1,
    kPDFActionNamedPreviousPage = 2,
    kPDFActionNamedFirstPage = 3,
    kPDFActionNamedLastPage = 4,
    kPDFActionNamedGoBack = 5,
    kPDFActionNamedGoForward = 6,
    kPDFActionNamedGoToPage = 7,
    kPDFActionNamedFind = 8,
    kPDFActionNamedPrint = 9,
    kPDFActionNamedZoomIn = 10,
    kPDFActionNamedZoomOut = 11
};
```

Constants

kPDFActionNamedNone

The action has no name.

Available in Mac OS X v10.5 and later.

Declared in PDFActionNamed.h.

kPDFActionNamedNextPage

The Next Page action.

Available in Mac OS X v10.5 and later.

Declared in PDFActionNamed.h.

kPDFActionNamedPreviousPage

The Previous Page action.

Available in Mac OS X v10.5 and later.

Declared in PDFActionNamed.h.

kPDFActionNamedFirstPage

The First Page action.

Available in Mac OS X v10.5 and later.

Declared in PDFActionNamed.h.

kPDFActionNamedLastPage

The Last Page action.

Available in Mac OS X v10.5 and later.

Declared in PDFActionNamed.h.

kPDFActionNamedGoBack

The Go Back action.

Available in Mac OS X v10.5 and later.

Declared in PDFActionNamed.h.

kPDFActionNamedGoForward

The Go Forward action.

Available in Mac OS X v10.5 and later.

Declared in PDFActionNamed.h.

kPDFActionNamedGoToPage

The Go to Page action.

Available in Mac OS X v10.5 and later.

Declared in PDFActionNamed.h.

kPDFActionNamedFind

The Find action.

Available in Mac OS X v10.5 and later.

Declared in PDFActionNamed.h.

kPDFActionNamedPrint

The Print action.

Available in Mac OS X v10.5 and later.

Declared in PDFActionNamed.h.

kPDFActionNamedZoomIn

The Zoom In action.

Available in Mac OS X v10.5 and later.

Declared in PDFActionNamed.h.

kPDFActionNamedZoomOut

The Zoom Out action.

Available in Mac OS X v10.5 and later.

Declared in PDFActionNamed.h.

Availability

Available in Mac OS X v10.5 and later.

Declared In

PDFActionNamed.h

PDFActionRemoteGoTo Class Reference

Inherits from	PDFAction : NSObject
Conforms to	NSCopying NSObject (NSObject)
Framework	Library/Frameworks/Quartz.framework/Frameworks/PDFKit.framework
Declared in	PDFKit/PDFActionRemoteGoTo.h
Availability	Available in Mac OS X v10.5 and later.

Overview

`PDFActionRemoteGoTo`, a subclass of `PDFAction`, defines methods for getting and setting the destination of a go-to action that targets another document.

Tasks

Initializing the Remote Go-to Action

- [initWithPageIndex:atPoint:fileURL:](#) (page 22)
Initializes the remote go-to action with the specified page index, point, and document URL.

Accessing the Page Index of the Referenced Document

- [pageIndex](#) (page 22)
Returns the zero-based page index referenced by the remote go-to action.
- [setPageIndex:](#) (page 23)
Sets the zero-based page index referenced by the remote go-to action.

Accessing a Point on the Referenced Page

- [point](#) (page 23)
Returns the point, in page space, on the page referenced by the remote go-to action.

- `setPoint:` (page 23)
Sets the point, in page space, on the page referenced by the remote go-to action.

Accessing the URL of the Referenced Document

- `URL` (page 24)
Returns the URL of the document referenced by the remote go-to action.
- `setURL:` (page 24)
Sets the URL of the document referenced by the remote go-to action.

Instance Methods

`initWithPageIndex:atPoint:fileURL:`

Initializes the remote go-to action with the specified page index, point, and document URL.

```
- (id)initWithPageIndex:(NSUInteger)pageIndex atPoint:(NSPoint)point fileURL:(NSURL*)url
```

Parameters

pageIndex

The page index of the remote document.

point

The point on the page in the remote document.

url

The URL of the remote PDF document.

Return Value

An initialized `PDFActionRemoteGoTo` instance, or `NULL` if the object could not be initialized..

Discussion

The `PDFActionRemoteGoTo` object uses a zero-based page index, not a `PDFPage` object. This simplifies the handling of remote destinations for documents that may not be instantiated yet.

Availability

Available in Mac OS X v10.5 and later.

Declared In

`PDFActionRemoteGoTo.h`

`pageIndex`

Returns the zero-based page index referenced by the remote go-to action.

```
- (NSUInteger)pageIndex
```

Return Value

The page index referenced by the remote go-to action.

Availability

Available in Mac OS X v10.5 and later.

See Also

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

Declared In

PDFActionRemoteGoTo.h

point

Returns the point, in page space, on the page referenced by the remote go-to action.

- (NSPoint)point

Return Value

The point on the page of the remote document referenced by the action. If either the x value or the y value of the point is `kPDFDestinationUnspecifiedValue`, no position on the page is specified.

Discussion

Page space is a 72-dpi coordinate system with the origin at the lower-left corner of the current page.

Availability

Available in Mac OS X v10.5 and later.

Declared In

PDFActionRemoteGoTo.h

setPageIndex:

Sets the zero-based page index referenced by the remote go-to action.

- (void)setPageIndex:(NSUInteger)pageIndex

Parameters

pageIndex

The page index in the remote document to go to.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [pageIndex](#) (page 22)

Declared In

PDFActionRemoteGoTo.h

setPoint:

Sets the point, in page space, on the page referenced by the remote go-to action.

- (void)setPoint:(NSPoint)point

Parameters*point*

The point on the remote page to go to. If either the x value or the y value of the point is `kPDFDestinationUnspecifiedValue`, no position on the page is specified.

Discussion

Page space is a 72-dpi coordinate system with the origin at the lower-left corner of the current page.

Availability

Available in Mac OS X v10.5 and later.

Declared In

PDFActionRemoteGoTo.h

setURL:

Sets the URL of the document referenced by the remote go-to action.

```
- (void)setURL:(NSURL *)url
```

Parameters*url*

The URL of the remote document to go to.

Availability

Available in Mac OS X v10.5 and later.

Declared In

PDFActionRemoteGoTo.h

URL

Returns the URL of the document referenced by the remote go-to action.

```
- (NSURL *)URL
```

Return Value

The URL of the remote document referenced by the action.

Availability

Available in Mac OS X v10.5 and later.

Declared In

PDFActionRemoteGoTo.h

PDFActionResetForm Class Reference

Inherits from	PDFAction : NSObject
Conforms to	NSCopying NSObject (NSObject)
Framework	Library/Frameworks/Quartz.framework/Frameworks/PDFKit.framework
Declared in	PDFKit/PDFActionResetForm.h
Availability	Available in Mac OS X v10.5 and later.

Overview

PDFActionResetForm, a subclass of PDFAction, defines methods for getting and clearing fields in a PDF form.

A PDFActionResetForm object represents an action associated with a PDF form.

Tasks

Initializing a Reset Form Action

- [init](#) (page 26)
Initializes a reset form action.

Accessing and Changing Fields

- [fields](#) (page 26)
Returns an array of fields associated with the reset action.
- [setFields:](#) (page 27)
Sets the array of fields associated with the reset action.

Determining Whether Fields Are Cleared When the Action Is Performed

- [fieldsIncludedAreCleared](#) (page 26)
Returns whether the fields associated with the reset action are cleared when the action is performed.

- [setFieldsIncludedAreCleared:](#) (page 27)
Sets whether the fields associated with the reset action are cleared when the action is performed.

Instance Methods

fields

Returns an array of fields associated with the reset action.

```
- (NSArray *)fields
```

Return Value

An array of `NSString` objects that corresponds to the `fieldNames` property of widget annotations (such as `PDFAnnotationButtonWidget`) on the PDF page. This method can return `NULL`.

Availability

Available in Mac OS X v10.5 and later.

See Also

- `setFields`

Declared In

`PDFActionResetForm.h`

fieldsIncludedAreCleared

Returns whether the fields associated with the reset action are cleared when the action is performed.

```
- (BOOL)fieldsIncludedAreCleared
```

Discussion

If `YES`, the reset action's fields are cleared when the action is performed. If `NO`, the fields are excluded from the reset action; that is, they are not cleared, but all other fields in the document are cleared.

Availability

Available in Mac OS X v10.5 and later.

See Also

- `setFieldsIncludedAreCleared:`

Declared In

`PDFActionResetForm.h`

init

Initializes a reset form action.

```
- (id)init
```

Return Value

An initialized `PDFActionResetForm` instance, or `NULL` if the object could not be initialized.

Discussion

Initially, there are no fields and `fieldsIncludedAreCleared` (page 26) returns `YES`.

Availability

Available in Mac OS X v10.5 and later.

Declared In

`PDFActionResetForm.h`

setFields:

Sets the array of fields associated with the reset action.

```
- (void)setFields:(NSArray *)fields
```

Parameters

fields

An array of strings that represent field names.

Availability

Available in Mac OS X v10.5 and later.

See Also

- `fields`

Declared In

`PDFActionResetForm.h`

setFieldsIncludedAreCleared:

Sets whether the fields associated with the reset action are cleared when the action is performed.

```
- (void)setFieldsIncludedAreCleared:(BOOL)include
```

Parameters

include

Pass `YES` to clear the fields associated with the action when the reset action is performed. Pass `NO` to exclude from the reset action only the fields associated with the action.

Availability

Available in Mac OS X v10.5 and later.

See Also

- `fieldsIncludedAreCleared`

Declared In

`PDFActionResetForm.h`

PDFActionURL Class Reference

Inherits from	PDFAction : NSObject
Conforms to	NSCopying NSObject (NSObject)
Framework	Library/Frameworks/Quartz.framework/Frameworks/PDFKit.framework
Declared in	PDFKit/PDFActionURL.h
Availability	Available in Mac OS X v10.5 and later.

Overview

`PDFActionURL`, a subclass of `PDFAction`, defines methods for getting and setting the URL associated with a URL action.

Tasks

Initializing a URL Action

- [initWithURL:](#) (page 30)
Initializes a URL action with the specified URL.

Accessing and Changing the URL

- [URL](#) (page 30)
Returns the URL associated with the URL action.
- [setURL:](#) (page 30)
Sets the URL associated with the URL action.

Instance Methods

initWithURL:

Initializes a URL action with the specified URL.

```
- (id)initWithURL:(NSURL *)url
```

Parameters

url

The URL to set the action to.

Return Value

An initialized `PDFActionURL` instance, or `NULL` if the object could not be initialized.

Availability

Available in Mac OS X v10.5 and later.

Declared In

`PDFActionURL.h`

setURL:

Sets the URL associated with the URL action.

```
- (void)setURL:(NSURL *)url
```

Parameters

url

The URL to set the action to.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [URL](#)

Declared In

`PDFActionURL.h`

URL

Returns the URL associated with the URL action.

```
- (NSURL *)URL
```

Return Value

The URL associated with the action, or `NULL` if no URL is specified.

Availability

Available in Mac OS X v10.5 and later.

See Also

- `setURL:`

Declared In

PDFActionURL.h

PDFAnnotation Class Reference

Inherits from	NSObject
Conforms to	NSObject (NSObject)
Framework	Library/Frameworks/Quartz.framework/Frameworks/PDFKit.framework
Declared in	PDFKit/PDFAnnotation.h
Availability	Available in Mac OS X v10.4 and later.

Overview

`PDFAnnotation`, a subclass of `NSObject`, represents an annotation in a PDF document, which associates an object (such as a note or a sound) with a location in a PDF document.

In addition to its primary textual content, a PDF file can contain annotations that represent links, form elements, highlighting circles, textual notes, and so on. Each annotation is associated with a specific location on a page and may offer interactivity with the user. See the Adobe PDF Specification for more on annotations.

You are not likely to work with a `PDFAnnotation` object by itself, because the specific subclasses, such as `PDFAnnotationCircle`, are much more useful. When a PDF file is being parsed, however, any unknown or unsupported annotation is represented as a `PDFAnnotation` object.

`PDFAnnotation` is an abstract superclass of the following concrete classes:

- `PDFAnnotationButtonWidget`
- `PDFAnnotationCircle`
- `PDFAnnotationFreeText`
- `PDFAnnotationInk`
- `PDFAnnotationLine`
- `PDFAnnotationLink`
- `PDFAnnotationMarkup`
- `PDFAnnotationPopup`
- `PDFAnnotationSquare`
- `PDFAnnotationStamp`
- `PDFAnnotationText`
- `PDFAnnotationTextWidget`

Tasks

Initializing an Annotation

- `initWithBounds:` (page 38)
Initializes a PDF annotation object.

Accessing Information About an Annotation

- `page` (page 39)
Returns the page that the annotation is associated with.
- `modificationDate` (page 38)
Returns the modification date of the annotation.
- `setModificationDate:` (page 41)
Sets the modification date of the annotation.
- `userName` (page 45)
Returns the name of the user who created the annotation.
- `setUserName:` (page 43)
Sets the name of the user who created the annotation.
- `popup` (page 39)
Returns the pop-up annotation associated with an annotation.
- `setPopup:` (page 42)
Sets the pop-up annotation associated with an annotation.
- `mouseUpAction` (page 38)
Returns the optional action performed when a user releases the mouse button within an annotation.
- `setMouseUpAction:` (page 42)
Sets the action performed when a user releases the mouse button within an annotation.
- `type` (page 45)
Returns the type of the annotation.
- `contents` (page 36)
Returns the textual content (if any) associated with the annotation.
- `setContents:` (page 41)
Specifies the textual content associated with the annotation.
- `toolTip` (page 44)
Returns text for display as a help tag.

Managing Annotation Display Characteristics

- `bounds` (page 36)
Returns the bounding box for the annotation in page space.
- `setBounds:` (page 40)
Sets the bounding box for the annotation.

- [border](#) (page 35)
Returns the border style for the annotation.
- [setBorder:](#) (page 40)
Sets the border style for the annotation.
- [color](#) (page 36)
Returns the stroke color for the annotation.
- [setColor:](#) (page 40)
Sets the stroke color for the annotation.
- [hasAppearanceStream](#) (page 37)
Returns a Boolean value that indicates whether the annotation has an appearance stream associated with it.

Managing Annotation Drawing and Output

- [drawWithBox:](#) (page 37)
Draws the annotation on its associated page.
- [shouldDisplay](#) (page 44)
Returns a Boolean value indicating whether the annotation should be displayed.
- [setShouldDisplay:](#) (page 42)
Specifies whether the annotation should be displayed.
- [shouldPrint](#) (page 44)
Returns a Boolean value indicating whether the annotation should appear when the document is printed.
- [setShouldPrint:](#) (page 43)
Specifies whether the annotation should appear when the document is printed.

Instance Methods

border

Returns the border style for the annotation.

```
- (PDFBorder *)border
```

Return Value

The border style for the annotation. See “Constants” (page 111) in the PDFBorder class for possible values.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setBorder:](#) (page 40)

Declared In

PDFAnnotation.h

bounds

Returns the bounding box for the annotation in page space.

- (NSRect)bounds

Return Value

The bounding box for the annotation in page space.

Discussion

Page space is a 72-dpi coordinate system with the origin at the lower-left corner of the current page.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setBounds:](#) (page 40)

Declared In

PDFAnnotation.h

color

Returns the stroke color for the annotation.

- (NSColor *)color

Return Value

The stroke color for the annotation.

Discussion

Where this color is used depends on the type of annotation.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setColor:](#) (page 40)

Declared In

PDFAnnotation.h

contents

Returns the textual content (if any) associated with the annotation.

- (NSString *)contents

Return Value

A string representing the textual content associated with the annotation.

Discussion

Textual content is typically associated with `PDFAnnotationText` and `PDFAnnotationFreeText` annotations.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setContentts:](#) (page 41)
- [toolTip](#) (page 44)

Declared In

PDFAnnotation.h

drawWithBox:

Draws the annotation on its associated page.

- (void)drawWithBox:(PDFDisplayBox) *box*

Parameters

box

The bounding box used to draw the annotation in.

Discussion

The annotation is drawn relative to the origin of *box* in page space.

Page space is a 72 dpi coordinate system with the origin at the lower-left corner of the current page.

For additional information see the “Constants” section in the PDFPage class.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [boundsForBox:](#) (page 157)

Declared In

PDFAnnotation.h

hasAppearanceStream

Returns a Boolean value that indicates whether the annotation has an appearance stream associated with it.

- (BOOL) hasAppearanceStream

Return Value

YES if the annotation has an appearance stream; otherwise NO.

Discussion

An appearance stream is a sequence of draw instructions used to render a PDF item. If an appearance stream exists, PDF Kit draws the annotation using the stream, which may override existing set parameters (such as the stroke color set with `setColor`).

Availability

Available in Mac OS X v10.4 and later.

Declared In

PDFAnnotation.h

initWithBounds:

Initializes a PDF annotation object.

```
- (id)initWithBounds:(NSRect)bounds
```

Parameters*bounds*

The bounding box of the annotation, in page space.

Return Value

An initialized `PDFAnnotation` instance, or `NULL` if the object could not be initialized.

Discussion

Subclasses of `PDFAnnotation` should use this method to initialize annotation instances. Provide *bounds* in page space. Invoking `initWithBounds:` directly on a `PDFAnnotation` object creates an illegal `NULL` type.

Page space is a 72 dpi coordinate system with the origin at the lower-left corner of the current page.

Availability

Available in Mac OS X v10.4 and later.

Declared In

PDFAnnotation.h

modificationDate

Returns the modification date of the annotation.

```
- (NSDate *)modificationDate
```

Return Value

The modification date of the annotation, or `NULL` if there is no modification date.

Availability

Available in Mac OS X v10.5 and later.

See Also

```
- setModificationDate
```

Declared In

PDFAnnotation.h

mouseUpAction

Returns the optional action performed when a user releases the mouse button within an annotation.

```
- (PDFAction *)mouseUpAction
```

Return Value

The PDF action performed when a user releases the mouse button within an annotation.

Availability

Available in Mac OS X v10.5 and later.

See Also

- `setMouseUpAction`

Declared In

`PDFAnnotation.h`

page

Returns the page that the annotation is associated with.

- `(PDFPage *)page`

Return Value

The PDF page associated with the annotation.

Discussion

The [addAnnotation:](#) (page 155) method in the `PDFPage` class lets you associate an annotation with a page.

Availability

Available in Mac OS X v10.4 and later.

Declared In

`PDFAnnotation.h`

popup

Returns the pop-up annotation associated with an annotation.

- `(PDFAnnotationPopup *)popup`

Return Value

The pop-up annotation associated with the annotation, or `NULL` if no pop-up exists.

Discussion

Pop-up annotations are not used with links or widgets. The bounds and open state of the pop-up annotation indicate the placement and open state of the pop-up window.

Availability

Available in Mac OS X v10.5 and later.

See Also

- `setPopup`

Declared In

`PDFAnnotation.h`

setBorder:

Sets the border style for the annotation.

```
- (void)setBorder:(PDFBorder *)border
```

Parameters

border

The border style for the annotation. See “Constants” (page 111) in the PDFBorder class for the available styles. The border style attribute is optional.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [border](#) (page 35)

Declared In

PDFAnnotation.h

setBounds:

Sets the bounding box for the annotation.

```
- (void)setBounds:(NSRect)bounds
```

Parameters

bounds

The bounding box for the annotation. Use page space for *bounds*. The bounds attribute is required for all annotations.

Discussion

Page space is a 72-dpi coordinate system with the origin at the lower-left corner of the current page.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [bounds](#) (page 36)

Declared In

PDFAnnotation.h

setColor:

Sets the stroke color for the annotation.

```
- (void)setColor:(NSColor *)color
```

Parameters

color

The stroke color for the annotation.

Discussion

Where this color is used depends on the annotation type.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [color](#) (page 36)

Declared In

PDFAnnotation.h

setContentts:

Specifies the textual content associated with the annotation.

```
- (void)setContentts:(NSString *)contentts
```

Parameters

contentts

A string representing the textual contents associated with the annotation.

Discussion

Textual content is typically associated with `PDFAnnotationText` and `PDFAnnotationFreeText` annotations. For most annotation types, `PDFView` displays the associated textual content as a help tag.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [contentts](#) (page 36)

- [toolTip](#) (page 44)

Declared In

PDFAnnotation.h

setModificationDate:

Sets the modification date of the annotation.

```
- (void)setModificationDate:(NSDate *)date
```

Parameters

date

The modification date to associate with the annotation.

Discussion

The modification date is optional.

Availability

Available in Mac OS X v10.5 and later.

See Also

- `modificationDate`

Declared In

PDFAnnotation.h

setMouseUpAction:

Sets the action performed when a user releases the mouse button within an annotation.

```
- (void)setMouseUpAction:(PDFAction *)action
```

Parameters

action

The PDF action to be performed when a user releases the mouse button within an annotation.

Discussion

The mouse-up action is optional.

Availability

Available in Mac OS X v10.5 and later.

See Also

- `mouseUpAction`

Declared In

PDFAnnotation.h

setPopup:

Sets the pop-up annotation associated with an annotation.

```
- (void)setPopup:(PDFAnnotationPopup *)popup
```

Parameters

popup

The pop-up annotation to associate with the annotation.

Discussion

A pop-up annotation is not associated with links or widgets. The bounds and open state of the pop-up annotation indicate the placement and open state of the pop-up window.

Availability

Available in Mac OS X v10.5 and later.

See Also

- `popup`

Declared In

PDFAnnotation.h

setShouldDisplay:

Specifies whether the annotation should be displayed.

```
- (void)setShouldDisplay:(BOOL)display
```

Parameters

display

Set this value to YES to display the annotation or NO otherwise.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [shouldDisplay](#) (page 44)

Declared In

PDFAnnotation.h

setShouldPrint:

Specifies whether the annotation should appear when the document is printed.

```
- (void)setShouldPrint:(BOOL)print
```

Parameters

print

Set this value to YES to ensure the annotation appears when the document is printed or NO otherwise.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [shouldPrint](#) (page 44)

Declared In

PDFAnnotation.h

setUserName:

Sets the name of the user who created the annotation.

```
- (void)userName:(NSString *)name
```

Parameters

name

The name of the user who created the annotation.

Discussion

The user name is optional.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [userName](#)

Declared In

PDFAnnotation.h

shouldDisplay

Returns a Boolean value indicating whether the annotation should be displayed.

- (BOOL)shouldDisplay

Return Value

YES if the annotation should be displayed; otherwise NO.

Discussion

PDFPage respects this flag when drawing.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setShouldDisplay](#): (page 42)

Declared In

PDFAnnotation.h

shouldPrint

Returns a Boolean value indicating whether the annotation should appear when the document is printed.

- (BOOL)shouldPrint

Return Value

YES if the annotation should appear when the PDF document is printed; otherwise NO.

Discussion

PDFPage respects this flag when printing.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setShouldPrint](#): (page 43)

Declared In

PDFAnnotation.h

toolTip

Returns text for display as a help tag.

- (NSString *)toolTip

Return Value

A string that contains help tag content, or NULL if there is no text associated with the annotation.

Discussion

This method is equivalent to sending the message `[self contents]`. PDF Kit's annotation subclasses override this behavior as appropriate. For example, a `PDFAnnotationLink` object displays a URL or page destination for its help tag.

Availability

Available in Mac OS X v10.4 and later.

Declared In

`PDFAnnotation.h`

type

Returns the type of the annotation.

- (NSString *)type

Return Value

The type of the annotation. Types include `Line`, `Link`, `Text`, and so on, referring to the `PDFAnnotation` subclasses. In the Adobe PDF Specification, this attribute is called `Subtype`, and the common "type" for all annotations in the PDF Specification is `Annot`.

Availability

Available in Mac OS X v10.4 and later.

Declared In

`PDFAnnotation.h`

userName

Returns the name of the user who created the annotation.

- (NSString *)userName

Return Value

The name of the user who created the annotation, or `NULL` if no user name is set.

Availability

Available in Mac OS X v10.5 and later.

See Also

- `setUserName`

Declared In

`PDFAnnotation.h`

PDFAnnotationButtonWidget Class Reference

Inherits from	PDFAnnotation : NSObject
Conforms to	NSObject (NSObject)
Framework	Library/Frameworks/Quartz.framework/Frameworks/PDFKit.framework
Availability	Available in Mac OS X v10.4 and later.
Declared in	PDFKit/PDFAnnotationButtonWidget.h

Overview

A `PDFAnnotationButtonWidget` object provides user interactivity on a page of a PDF document. There are three types of buttons available: push button, radio button, and checkbox.

`PDFAnnotationButtonWidget` inherits general annotation behavior from the `PDFAnnotation` class. If you use a `PDFAnnotationButtonWidget` object, your application must handle hit testing, unless you are simply using `PDFView` to display content. This is because `PDFView` automatically handles hit testing for you.

Tasks

Getting and Setting the Control Type

- `controlType` (page 50)
Returns the type of the control.
- `setControlType:` (page 53)
Sets the type of the control.
- `parentID` (page 52)
Gets the ID of the parent object. **(Deprecated.** If you need to find other buttons in the same group, such as a group of radio buttons, you do not need information about the parent object. Instead, look for button widget objects that return the same value in `fieldName`.)

Getting and Setting the Control's State

- `state` (page 56)
Returns the state of the control.

- `setState:` (page 55)
Sets the state of the control.

Getting and Setting the Control's Appearance

- `isHighlighted` (page 51)
Returns a Boolean value that indicates whether the control is highlighted when it is drawn.
- `setHighlighted:` (page 54)
Sets the control's highlighting when it is drawn.
- `backgroundColor` (page 49)
Returns the background color of the control.
- `setBackgroundColor:` (page 52)
Sets the control's background color.

Getting and Setting the Control Label Font Attributes

- `font` (page 51)
Returns the font used in the control's label.
- `setFont:` (page 54)
Sets the font of the control's label.
- `fontColor` (page 51)
Returns the font color used in the control's label.
- `setFontColor:` (page 54)
Sets the font color used in the control's label.

Getting and Setting the Control Label Text

- `caption` (page 49)
Returns the text of the label on a push button control.
- `setCaption:` (page 53)
Sets the text of the label on a push button control.

Managing Radio Button Behavior

- `allowsToggleToOff` (page 49)
Returns a Boolean value indicating whether a radio button behaves in a toggling manner.

Managing Control State Values and Form Fields

- `onStateValue` (page 52)
Returns the string associated with the on state of a radio button or checkbox control.

- [setStateValue:](#) (page 55)
Sets the string that is associated with the on state of a radio button or checkbox control.
- [fieldName](#) (page 50)
Returns the internal name of a field (used for reset-form actions).
- [setFieldName:](#) (page 53)
Sets the internal name of a field (used for reset-form actions).

Instance Methods

allowsToggleToOff

Returns a Boolean value indicating whether a radio button behaves in a toggling manner.

- (BOOL)allowsToggleToOff

Return Value

YES if clicking a radio button control that is already in the on state toggles it to the off state; otherwise NO.

Availability

Available in Mac OS X v10.5 and later.

Declared In

PDFAnnotationButtonWidget.h

backgroundColor

Returns the background color of the control.

- (NSColor *)backgroundColor

Return Value

The color drawn in the background of the control.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setBackgroundColor:](#) (page 52)

Declared In

PDFAnnotationButtonWidget.h

caption

Returns the text of the label on a push button control.

- (NSString *)caption

Return Value

The label drawn on a push button control.

Discussion

This method applies only to the label drawn on a control of type `kPDFWidgetPushButtonControl`.

Availability

Available in Mac OS X v10.5 and later.

See Also

- `setCaption:`

Declared In

`PDFAnnotationButtonWidget.h`

controlType

Returns the type of the control.

- (`PDFWidgetControlType`)`controlType`

Return Value

The type of control the button represents. See “[Constants](#)” (page 56) for the various control types.

Availability

Available in Mac OS X v10.4 and later.

See Also

- `setControlType`

Declared In

`PDFAnnotationButtonWidget.h`

fieldName

Returns the internal name of a field (used for reset-form actions).

- (`NSString *`)`fieldName`

Return Value

The internal name of a field.

Discussion

The internal name of a field is an optional value.

Availability

Available in Mac OS X v10.5 and later.

See Also

- `setFieldName`

Declared In

`PDFAnnotationButtonWidget.h`

font

Returns the font used in the control's label.

- (NSFont *)font

Return Value

The font used in the control's label.

Availability

Available in Mac OS X v10.5 and later.

See Also

- setFont:

Declared In

PDFAnnotationButtonWidget.h

fontColor

Returns the font color used in the control's label.

- (NSColor *)fontColor

Return Value

The font color used in the control's label.

Availability

Available in Mac OS X v10.5 and later.

See Also

- setFontColor:

Declared In

PDFAnnotationButtonWidget.h

isHighlighted

Returns a Boolean value that indicates whether the control is highlighted when it is drawn.

- (BOOL)isHighlighted

Return Value

YES if the control is highlighted when it is drawn; otherwise NO.

Availability

Available in Mac OS X v10.5 and later.

See Also

- setHighlighted:

Declared In

PDFAnnotationButtonWidget.h

onStateValue

Returns the string associated with the on state of a radio button or checkbox control.

- (NSString *)onStateValue

Return Value

The string associated with the on state of a radio button or checkbox control.

Discussion

This is a required string for controls of types `kPDFWidgetRadioButtonControl` and `kPDFWidgetCheckBoxControl`. The off state is always labeled “Off”.

Availability

Available in Mac OS X v10.5 and later.

See Also

- `setOnStateValue`

Declared In

`PDFAnnotationButtonWidget.h`

parentID

Gets the ID of the parent object. (**Deprecated.** If you need to find other buttons in the same group, such as a group of radio buttons, you do not need information about the parent object. Instead, look for button widget objects that return the same value in `fieldName`.)

- (unsigned)parentID

Discussion

For more information about the field names of annotations, see the Adobe PDF specification.

Availability

Available in Mac OS X v10.4 through Mac OS X v10.4.

Declared In

`PDFAnnotationButtonWidget.h`

setBackground-color:

Sets the control’s background color.

- (void)setBackgroundColor:(NSColor *)color

Parameters

color

The color to be drawn in the control’s background.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [backgroundColor](#) (page 49)

Declared In

PDFAnnotationButtonWidget.h

setCaption:

Sets the text of the label on a push button control.

```
- (void)setCaption:(NSString *)name
```

Parameters*name*

The text to be used as the label on a push button control.

Availability

Available in Mac OS X v10.5 and later.

See Also`- caption`**Declared In**

PDFAnnotationButtonWidget.h

setControlType:

Sets the type of the control.

```
- (void)setControlType:(PDFWidgetControlType)type
```

Parameters*type*The type of control for the button. “[Constants](#)” (page 56) lists the various control types you can send for this value.**Availability**

Available in Mac OS X v10.5 and later.

See Also`- controlType`**Declared In**

PDFAnnotationButtonWidget.h

setFieldName:

Sets the internal name of a field (used for reset-form actions).

```
- (void)setFieldName:(NSString *)name
```

Parameters*name*

The internal name of a field. This is an optional value.

Availability

Available in Mac OS X v10.5 and later.

See Also

- `fieldName`

Declared In

`PDFAnnotationButtonWidget.h`

setFont:

Sets the font of the control's label.

- (void)setFont:(NSFont *)*font*

Parameters

font

The desired font for the control's label.

Availability

Available in Mac OS X v10.5 and later.

See Also

- `font`

Declared In

`PDFAnnotationButtonWidget.h`

setFontColor:

Sets the font color used in the control's label.

- (void)setFontColor:(NSColor *)*color*

Parameters

color

The desired font color of the control's label.

Availability

Available in Mac OS X v10.5 and later.

See Also

- `fontColor`

Declared In

`PDFAnnotationButtonWidget.h`

setHighlighted:

Sets the control's highlighting when it is drawn.

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

Parameters*flag*

Set this value to YES to cause the control to be highlighted when it is drawn or NO otherwise.

Availability

Available in Mac OS X v10.4 and later.

See Also

- `isHighlighted`

Declared In

`PDFAnnotationButtonWidget.h`

setStateValue:

Sets the string that is associated with the on state of a radio button or checkbox control.

- (void)setStateValue:(NSString *)*name*

Discussion

Required for controls of types `kPDFWidgetRadioButtonControl` and `kPDFWidgetCheckBoxControl`, the value of *name* describes the on state of the control (the off state is always labeled “Off”). Although “On” is an acceptable string for the on state of a single checkbox, a group of two or more radio buttons should have a unique string associated with each control.

For example, a form might display a group of 3 radio buttons that allow users to indicate an account type, such as savings, checking, or investment. The strings associated with the on states of these buttons could be “Savings,” “Checking,” and “Investment.” In this example, these 3 radio buttons also would share a field name string, such as “AccountType.”

Availability

Available in Mac OS X v10.5 and later.

See Also

- `onStateValue`
- `fieldName`

Declared In

`PDFAnnotationButtonWidget.h`

setState:

Sets the state of the control.

- (void)setState:(int)*value*

Parameters*value*

The state the control should be in.

Discussion

A control’s state (for example, checked or unchecked) affects how it is drawn. Note that push buttons are always in the on state.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [state](#) (page 56)

Declared In

PDFAnnotationButtonWidget.h

state

Returns the state of the control.

- (int)state

Return Value

NSOnState if the control is on; NSOffState otherwise.

Availability

Available in Mac OS X v10.4 and later.

See Also

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

Declared In

PDFAnnotationButtonWidget.h

Constants

Types of PDF Annotation Buttons

The types of annotation buttons.

```
typedef NSInteger PDFWidgetControlType;
enum {
    kPDFWidgetUnknownControl = -1,
    kPDFWidgetPushButtonControl = 0,
    kPDFWidgetRadioButtonControl = 1,
    kPDFWidgetCheckBoxControl = 2
};
```

Constants

kPDFWidgetUnknownControl

Unknown control type.

Available in Mac OS X v10.4 and later.

Declared in PDFAnnotationButtonWidget.h.

kPDFWidgetPushButtonControl

Push button control.

Available in Mac OS X v10.4 and later.

Declared in PDFAnnotationButtonWidget.h.

kPDFWidgetRadioButtonControl

Radio button control.

Available in Mac OS X v10.4 and later.

Declared in PDFAnnotationButtonWidget.h.

kPDFWidgetCheckBoxControl

Check box control.

Available in Mac OS X v10.4 and later.

Declared in PDFAnnotationButtonWidget.h.

PDFAnnotationChoiceWidget Class Reference

Inherits from	PDFAnnotation : NSObject
Conforms to	NSObject (NSObject)
Framework	Library/Frameworks/Quartz.framework/Frameworks/PDFKit.framework
Declared in	PDFKit/PDFAnnotationChoiceWidget.h
Availability	Available in Mac OS X v10.5 and later.

Overview

A `PDFAnnotationChoiceWidget` object provides user interactivity on a page of a PDF document, in the form of pop-up menus and lists.

`PDFAnnotationChoiceWidget` inherits general annotation behavior from the `PDFAnnotation` class. If you use a `PDFAnnotationChoiceWidget` object, your application must handle hit testing, unless you are simply using `PDFView` to display content. This is because `PDFView` automatically handles hit testing for you.

Tasks

Getting and Setting the String Value

- [stringValue](#) (page 65)
Returns the selection in the widget annotation.
- [setStringValue:](#) (page 64)
Sets the selection in the widget annotation.

Managing Font and Background Color Characteristics

- [backgroundColor](#) (page 60)
Returns the color of the widget annotation background.
- [setBackground-color:](#) (page 62)
Sets the background color of the widget annotation.
- [font](#) (page 61)
Returns the font used to display the text in the widget annotation.

- `setFont:` (page 63)
Sets the font used to display the text in the widget annotation.
- `fontColor` (page 62)
Returns the font color used to display the text in the widget annotation.
- `setFontColor:` (page 63)
Sets the font color used to display the text in the widget annotation.

Managing the Associated Field Name

- `fieldName` (page 61)
Returns the internal field name associated with the widget annotation.
- `setFieldName:` (page 63)
Sets the internal field name associated with the widget annotation's value.

Determining the Type of Choice Widget Annotation

- `isListChoice` (page 62)
Returns a Boolean value indicating whether the widget annotation is a list.
- `setIsListChoice:` (page 64)
Sets whether the widget annotation is a list.

Accessing the Items in the Choice Widget Annotation

- `choices` (page 61)
Returns an array of strings that represent the items available in the list or pop-up menu of the choice widget annotation.
- `setChoices:` (page 62)
Sets the items available in the list or pop-up menu of the choice widget annotation.

Instance Methods

backgroundColor

Returns the color of the widget annotation background.

- (NSColor *)backgroundColor

Return Value

The color of the widget annotation background.

Availability

Available in Mac OS X v10.5 and later.

Declared In

PDFAnnotationChoiceWidget.h

choices

Returns an array of strings that represent the items available in the list or pop-up menu of the choice widget annotation.

```
- (NSArray *)choices
```

Return Value

An array of strings that represent the items in the list or pop-up menu choice widget annotation.

Availability

Available in Mac OS X v10.5 and later.

Declared In

PDFAnnotationChoiceWidget.h

fieldName

Returns the internal field name associated with the widget annotation.

```
- (NSString *)fieldName
```

Return Value

The internal field name associated with the widget annotation.

Discussion

If the widget annotation is backed by PDF form data, it can associate an optional field name with a value or other data.

Availability

Available in Mac OS X v10.5 and later.

Declared In

PDFAnnotationChoiceWidget.h

font

Returns the font used to display the text in the widget annotation.

```
- (NSFont *)font
```

Return Value

The font used to display the text in the widget annotation.

Availability

Available in Mac OS X v10.5 and later.

Declared In

PDFAnnotationChoiceWidget.h

fontColor

Returns the font color used to display the text in the widget annotation.

```
- (NSColor *)fontColor
```

Return Value

The color of the font used for the text in the widget annotation.

Availability

Available in Mac OS X v10.5 and later.

Declared In

PDFAnnotationChoiceWidget.h

isListChoice

Returns a Boolean value indicating whether the widget annotation is a list.

```
- (BOOL)isListChoice
```

Return Value

YES if the widget annotation is a list, NO otherwise.

Discussion

A choice widget annotation can be either a list or a pop-up menu.

Availability

Available in Mac OS X v10.5 and later.

Declared In

PDFAnnotationChoiceWidget.h

setBackground-color:

Sets the background color of the widget annotation.

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

Parameters

color

The color to use in the background of the widget annotation.

Availability

Available in Mac OS X v10.5 and later.

Declared In

PDFAnnotationChoiceWidget.h

setChoices:

Sets the items available in the list or pop-up menu of the choice widget annotation.

```
- (void)setChoices:(NSArray *)options
```

Parameters

options

Send an array of strings, each of which represents an item in the list or pop-up menu of the choice annotation widget.

Availability

Available in Mac OS X v10.5 and later.

Declared In

PDFAnnotationChoiceWidget.h

setFieldName:

Sets the internal field name associated with the widget annotation's value.

```
- (void)setFieldName:(NSString *)name
```

Parameters

name

The name to be used as the internal field name associated with the widget annotation.

Discussion

If the widget annotation is backed by PDF form data, it can associate an optional field name with a value or other data.

Availability

Available in Mac OS X v10.5 and later.

Declared In

PDFAnnotationChoiceWidget.h

setFont:

Sets the font used to display the text in the widget annotation.

```
- (void)setFont:(NSFont *)font
```

Parameters

font

The font to be used for the text in the widget annotation.

Availability

Available in Mac OS X v10.5 and later.

Declared In

PDFAnnotationChoiceWidget.h

setFontColor:

Sets the font color used to display the text in the widget annotation.

```
- (void)setFontColor:(NSColor *)color
```

Parameters

color

The color of the font to be used for the text in the widget annotation.

Availability

Available in Mac OS X v10.5 and later.

Declared In

PDFAnnotationChoiceWidget.h

setIsListChoice:

Sets whether the widget annotation is a list.

```
- (void)setIsListChoice:(BOOL)isList
```

Parameters

isList

Send YES to set the choice widget annotation is a list, NO otherwise.

Discussion

A choice widget annotation can be either a list or a pop-up menu.

Availability

Available in Mac OS X v10.5 and later.

Declared In

PDFAnnotationChoiceWidget.h

setStringValue:

Sets the selection in the widget annotation.

```
- (void)setStringValue:(NSString *)value
```

Parameters

value

The string that represents the selection in the widget annotation.

Discussion

If the widget annotation object is backed by PDF form data, this method updates the value associated with the appropriate field in the form object.

Availability

Available in Mac OS X v10.5 and later.

Declared In

PDFAnnotationChoiceWidget.h

stringValue

Returns the selection in the widget annotation.

```
- (NSString *)stringValue
```

Return Value

The string that represents the selection in the widget annotation.

Discussion

If the widget annotation object is backed by PDF form data, this method returns the value associated with the appropriate field in the form object, if possible.

Availability

Available in Mac OS X v10.5 and later.

Declared In

PDFAnnotationChoiceWidget.h

PDFAnnotationCircle Class Reference

Inherits from	PDFAnnotation : NSObject
Conforms to	NSObject (NSObject)
Framework	Library/Frameworks/Quartz.framework/Frameworks/PDFKit.framework
Declared in	PDFKit/PDFAnnotationCircle.h
Availability	Available in Mac OS X v10.4 and later.

Overview

A PDFAnnotationCircle object displays an ellipse on a page. Circle annotations are like square annotations (instances of the PDFAnnotationSquare class) apart from the shape.

The [setLineWidth:](#) (page 110) and [setStyle:](#) (page 110) methods of the annotation's associated PDFBorder object determines the stroke thickness and style. The [setColor:](#) (page 40) method of the PDFAnnotation class determines the stroke color.

Tasks

Accessor Methods

- [interiorColor](#) (page 67)
Returns the fill color used for drawing the annotation.
- [setInteriorColor:](#) (page 68)
Sets the fill color used for drawing the annotation.

Instance Methods

interiorColor

Returns the fill color used for drawing the annotation.

```
- (NSColor *)interiorColor
```

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setInteriorColor:](#) (page 68)

Declared In

PDFAnnotationCircle.h

setInteriorColor:

Sets the fill color used for drawing the annotation.

- (void)setInteriorColor:(NSColor *)*color*

Availability

Available in Mac OS X v10.4 and later.

See Also

- [interiorColor](#) (page 67)

Declared In

PDFAnnotationCircle.h

PDFAnnotationFreeText Class Reference

Inherits from	PDFAnnotation : NSObject
Conforms to	NSObject (NSObject)
Framework	Library/Frameworks/Quartz.framework/Frameworks/PDFKit.framework
Declared in	PDFKit/PDFAnnotationFreeText.h
Availability	Available in Mac OS X v10.4 and later.

Overview

A `PDFAnnotationFreeText` object displays text on a page.

Unlike a `PDFAnnotationText` object, a `PDFAnnotationFreeText` object has no open or closed state; its text is always visible. The text annotation performed in Preview uses `PDFAnnotationFreeText`.

The `PDFAnnotation` class's [contents](#) (page 36) and [setContentts:](#) (page 41) methods let you get and set the textual content for a `PDFAnnotationFreeText` object.

Tasks

Managing Text Alignment

- [alignment](#) (page 70)
Returns the horizontal alignment of text within the bounds of the annotation.
- [setAlignment:](#) (page 71)
Sets the horizontal alignment of text within the bounds of the annotation.

Managing Font and Font Color

- [font](#) (page 70)
Returns the font used for the annotation's text field.
- [setFont:](#) (page 71)
Sets the font used in the text field of the annotation.
- [fontColor](#) (page 70)
Returns the font color used in the text field of the annotation.

- [setFontColor:](#) (page 71)
Sets the font color used in the text field of the annotation.

Instance Methods

alignment

Returns the horizontal alignment of text within the bounds of the annotation.

- (NSTextAlignment)alignment

Return Value

The horizontal alignment of text within the bounds of the annotation. Supported values are `NSLeftTextAlignment`, `NSRightTextAlignment`, and `NSCenterTextAlignment`.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setAlignment:](#) (page 71)

Declared In

`PDFAnnotationFreeText.h`

font

Returns the font used for the annotation's text field.

- (NSFont *)font

Return Value

The font used for the annotation's text field.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setFont:](#) (page 71)

Declared In

`PDFAnnotationFreeText.h`

fontColor

Returns the font color used in the text field of the annotation.

- (NSColor *)fontColor

Return Value

The font color used in the text field of the annotation.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [setFontColor:](#)

Declared In

PDFAnnotationFreeText.h

setAlignment:

Sets the horizontal alignment of text within the bounds of the annotation.

- (void)setAlignment:(NSTextAlignment)*alignment*

Parameters

alignment

Send `NSLeftTextAlignment`, `NSRightTextAlignment`, or `NSCenterTextAlignment` to set the horizontal alignment of text within the bounds of the annotation.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [alignment](#) (page 70)

Declared In

PDFAnnotationFreeText.h

setFont:

Sets the font used in the text field of the annotation.

- (void)setFont:(NSFont *)*font*

Parameters

font

The font to be used in the text field of the annotation.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [font](#) (page 70)

Declared In

PDFAnnotationFreeText.h

setFontColor:

Sets the font color used in the text field of the annotation.

- (void)setFontColor:(NSColor *)*color*

Parameters

color

The font color to be used in the text field of the annotation.

Availability

Available in Mac OS X v10.5 and later.

See Also

- `fontColor`

Declared In

PDFAnnotationFreeText.h

PDFAnnotationInk Class Reference

Inherits from	PDFAnnotation : NSObject
Conforms to	NSObject (NSObject)
Framework	Library/Frameworks/Quartz.framework/Frameworks/PDFKit.framework
Declared in	PDFKit/PDFAnnotationInk.h
Availability	Available in Mac OS X v10.4 and later.

Overview

A PDFAnnotationInk object displays one or more disjoint Bezier paths on a page. This is typically used to represent a freehand jotting or “scribble” of handwritten text.

The [setLineWidth:](#) (page 110) and [setStyle:](#) (page 110) methods of the annotation’s associated PDFBorder object determines the stroke thickness and style. The [setColor:](#) (page 40) method of the PDFAnnotation class determines the stroke color.

Tasks

Accessor Methods

- [paths](#) (page 74)
Returns an array containing the Bezier paths that make up an annotation.

Working with Bezier Paths

- [addBezierPath:](#) (page 74)
Adds a Bezier path to an annotation.
- [removeBezierPath:](#) (page 74)
Removes a Bezier path from an annotation.

Instance Methods

addBezierPath:

Adds a Bezier path to an annotation.

- (void)addBezierPath:(NSBezierPath *)*path*

Availability

Available in Mac OS X v10.4 and later.

See Also

- [removeBezierPath:](#) (page 74)
- [paths](#) (page 74)

Declared In

PDFAnnotationInk.h

paths

Returns an array containing the Bezier paths that make up an annotation.

- (NSArray *)*paths*

Availability

Available in Mac OS X v10.4 and later.

See Also

- [addBezierPath:](#) (page 74)
- [removeBezierPath:](#) (page 74)

Declared In

PDFAnnotationInk.h

removeBezierPath:

Removes a Bezier path from an annotation.

- (void)removeBezierPath:(NSBezierPath *)*path*

Availability

Available in Mac OS X v10.4 and later.

See Also

- [addBezierPath:](#) (page 74)
- [paths](#) (page 74)

Declared In

PDFAnnotationInk.h

PDFAnnotationLine Class Reference

Inherits from	PDFAnnotation : NSObject
Conforms to	NSObject (NSObject)
Framework	Library/Frameworks/Quartz.framework/Frameworks/PDFKit.framework
Declared in	PDFKit/PDFAnnotationLine.h
Availability	Available in Mac OS X v10.4 and later.

Overview

A `PDFAnnotationLine` object displays a single line on a page.

The `setLineWidth:` (page 110) and `setStyle:` (page 110) methods of the annotation's associated `PDFBorder` object determines the stroke thickness and style. The `setColor:` (page 40) method of the `PDFAnnotation` class determines the stroke color.

Tasks

Specifying the Starting and Ending Points

- `startPoint` (page 79)
Returns the starting point for the line.
- `setStartPoint:` (page 79)
Sets the starting point for the line.
- `endPoint` (page 76)
Returns the ending point for the line in page space.
- `setEndPoint:` (page 77)
Sets the ending point for the line.

Specifying the Line Ending Styles

- `startLineStyle` (page 79)
Returns the line ending style for the starting point of the line.

- [setStartLineStyle:](#) (page 78)
Sets the line ending style for the starting point of the line.
- [endLineStyle](#) (page 76)
Returns the line ending style for the ending point of the line.
- [setEndLineStyle:](#) (page 77)
Sets the line ending style for the ending point of the line.

Specifying the Color of Line-end Ornaments

- [interiorColor](#) (page 77)
Returns the color used to fill the ornament at the ends of the line.
- [setInteriorColor:](#) (page 78)
Sets the color used to fill the ornament at the ends of the line.

Instance Methods

endLineStyle

Returns the line ending style for the ending point of the line.

- (PDFLineStyle)endLineStyle

Return Value

The line ending style for the ending point of the line.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setEndLineStyle:](#) (page 77)
- [startLineStyle](#) (page 79)

Declared In

PDFAnnotationLine.h

endPoint

Returns the ending point for the line in page space.

- (NSPoint)endPoint

Return Value

The ending point for the line, in page space.

Discussion

Page space is a 72-dpi coordinate system with the origin at the lower-left corner of the current page.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setEndPoint:](#) (page 77)
- [startPoint](#) (page 79)

Declared In

PDFAnnotationLine.h

interiorColor

Returns the color used to fill the ornament at the ends of the line.

- (NSColor *) interiorColor

Return Value

The color used in the line-end ornament at the ends of the line.

Availability

Available in Mac OS X v10.5 and later.

Declared In

PDFAnnotationLine.h

setEndLineStyle:

Sets the line ending style for the ending point of the line.

- (void)setEndLineStyle:(PDFLineStyle)*style*

Parameters

style

The line ending style for the ending point of the line.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [endLineStyle](#) (page 76)
- [setStartLineStyle:](#) (page 78)

Declared In

PDFAnnotationLine.h

setEndPoint:

Sets the ending point for the line.

- (void)setEndPoint:(NSPoint)*point*

Parameters*point*

The ending point for the line, in page space.

Discussion

Page space is a 72-dpi coordinate system with the origin at the lower-left corner of the current page.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [endPoint](#) (page 76)
- [setStartPoint:](#) (page 79)

Declared In

PDFAnnotationLine.h

setInteriorColor:

Sets the color used to fill the ornament at the ends of the line.

```
- (void)setInteriorColor:(NSColor *)color
```

Parameters*color*

The color to be used to fill in the ornament at the ends of the line.

Discussion

The ornament at the end of a line is optional (for more information, see the Adobe PDF Specification 1.4).

Availability

Available in Mac OS X v10.5 and later.

Declared In

PDFAnnotationLine.h

setStartLineStyle:

Sets the line ending style for the starting point of the line.

```
- (void)setStartLineStyle:(PDFLineStyle)style
```

Parameters*style***Availability**

Available in Mac OS X v10.4 and later.

See Also

- [startLineStyle](#) (page 79)
- [setEndLineStyle:](#) (page 77)

Declared In

PDFAnnotationLine.h

setStartPoint:

Sets the starting point for the line.

- (void)setStartPoint:(NSPoint)*point*

Parameters

point

The starting point for the line, in page space.

Discussion

Page space is a 72-dpi coordinate system with the origin at the lower-left corner of the current page.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [startPoint](#) (page 79)
- [setEndPoint:](#) (page 77)

Declared In

PDFAnnotationLine.h

startLineStyle

Returns the line ending style for the starting point of the line.

- (PDFLineStyle)startLineStyle

Return Value

The line ending style for the starting point of the line.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setStartLineStyle:](#) (page 78)
- [endLineStyle](#) (page 76)

Declared In

PDFAnnotationLine.h

startPoint

Returns the starting point for the line.

- (NSPoint)startPoint

Return Value

The starting point for the line, in page space.

Discussion

Page space is a 72-dpi coordinate system with the origin at the lower-left corner of the current page.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setStartPoint:](#) (page 79)
- [endPoint](#) (page 76)

Declared In

PDFAnnotationLine.h

Constants

The following constants specify the available line ending styles:

Constant	Description
kPDFLineStyleNone	No line ending. Available in Mac OS X v10.4 and later. Declared in PDFAnnotationLine.h.
kPDFLineStyleSquare	A square line ending filled with the annotation's interior color, if any. Available in Mac OS X v10.4 and later. Declared in PDFAnnotationLine.h.
kPDFLineStyleCircle	A circular line ending filled with the annotation's interior color, if any. Available in Mac OS X v10.4 and later. Declared in PDFAnnotationLine.h.
kPDFLineStyleDiamond	A diamond-shaped line ending filled with the annotation's interior color, if any. Available in Mac OS X v10.4 and later. Declared in PDFAnnotationLine.h.
kPDFLineStyleOpenArrow	An open arrowhead line ending, composed from two short lines meeting in an acute angle at the line end. Available in Mac OS X v10.4 and later. Declared in PDFAnnotationLine.h.
kPDFLineStyle-ClosedArrow	A closed arrowhead line ending, consisting of a triangle with the acute vertex at the line end and filled with the annotation's interior color, if any. Available in Mac OS X v10.4 and later. Declared in PDFAnnotationLine.h.

PDFAnnotationLink Class Reference

Inherits from	PDFAnnotation : NSObject
Conforms to	NSObject (NSObject)
Framework	Library/Frameworks/Quartz.framework/Frameworks/PDFKit.framework
Declared in	PDFKit/PDFAnnotationLink.h
Availability	Available in Mac OS X v10.4 and later.

Overview

A PDFAnnotationLink object represents either a hypertext link to another location in the document (specified as a PDFDestination object) or a URL.

Tasks

Working with Link Destinations

- [destination](#) (page 82)
Gets the destination for the link when the destination was specified as a PDFDestination object.
- [setDestination:](#) (page 82)
Sets the destination for the link as a PDFDestination object.
- [URL](#) (page 83)
Gets the destination for the link when the destination was specified as a URL.
- [setURL:](#) (page 83)
Sets the destination for the link as a URL.

Highlighting the Link

- [setHighlighted:](#) (page 82)
Sets the highlighting state for the link.

Instance Methods

destination

Gets the destination for the link when the destination was specified as a PDFDestination object.

- (PDFDestination *)destination

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setDestination:](#) (page 82)

Declared In

PDFAnnotationLink.h

setDestination:

Sets the destination for the link as a PDFDestination object.

- (void)setDestination:(PDFDestination *)destination

Availability

Available in Mac OS X v10.4 and later.

See Also

- [destination](#) (page 82)

Declared In

PDFAnnotationLink.h

setHighlighted:

Sets the highlighting state for the link.

- (void)setHighlighted:(BOOL)flag

Discussion

For typical PDF interaction, when a user clicks (mouse-down) on a link, set highlighting to YES and redraw the link. On the subsequent mouse-up event, set highlighting to NO and redraw again.

Availability

Available in Mac OS X v10.4 and later.

Declared In

PDFAnnotationLink.h

setURL:

Sets the destination for the link as a URL.

- (void)setURL:(NSURL *)url

Availability

Available in Mac OS X v10.4 and later.

See Also

- [URL](#) (page 83)

Declared In

PDFAnnotationLink.h

URL

Gets the destination for the link when the destination was specified as a URL.

- (NSURL *)URL

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setURL:](#) (page 83)

Declared In

PDFAnnotationLink.h

PDFAnnotationMarkup Class Reference

Inherits from	PDFAnnotation : NSObject
Conforms to	NSObject (NSObject)
Framework	Library/Frameworks/Quartz.framework/Frameworks/PDFKit.framework
Declared in	PDFKit/PDFAnnotationMarkup.h
Availability	Available in Mac OS X v10.4 and later.

Overview

A PDFAnnotationMarkup object appears as highlighting, underlining, or a strikethrough style applied to the text of a document.

The [setLineWidth:](#) (page 110) and [setStyle:](#) (page 110) methods of the annotation's associated PDFBorder object determines the stroke thickness and style. The [setColor:](#) (page 40) method of the PDFAnnotation class determines the stroke color.

Tasks

Working with Markup Boundaries

- [quadrilateralPoints](#) (page 86)
Gets the array of quadrilateral points defining the bounds of the markup.
- [setQuadrilateralPoints:](#) (page 87)
Sets the array of quadrilateral points defining the bounds of the markup.

Working with Markup Style

- [markupType](#) (page 86)
Gets the markup style.
- [setMarkupType:](#) (page 86)
Sets the markup style.

Instance Methods

markupType

Gets the markup style.

- (PDFMarkupType)markupType

Discussion

Refer to “[Constants](#)” (page 87) for the available markup styles.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setMarkupType:](#) (page 86)

Declared In

PDFAnnotationMarkup.h

quadrilateralPoints

Gets the array of quadrilateral points defining the bounds of the markup.

- (NSArray *)quadrilateralPoints

Discussion

Each quadrilateral encompasses a word or a contiguous group of words. The quadrilateral points are ordered counterclockwise, with the first point closest to the origin in page space.

Page space is a 72 dpi coordinate system with the origin at the lower-left corner of the current page.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setQuadrilateralPoints:](#) (page 87)

Declared In

PDFAnnotationMarkup.h

setMarkupType:

Sets the markup style.

- (void)setMarkupType:(PDFMarkupType)type

Discussion

Refer to “[Constants](#)” (page 87) for the available markup styles.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [markupType](#) (page 86)

Declared In

PDFAnnotationMarkup.h

setQuadrilateralPoints:

Sets the array of quadrilateral points defining the bounds of the markup.

```
- (void)setQuadrilateralPoints:(NSArray *)points
```

Discussion

The points defined by each quadrilateral array should encompass a word or a contiguous group of words. The quadrilateral points are ordered counterclockwise, with the first point closest to the origin in page space.

Page space is a 72 dpi coordinate system with the origin at the lower-left corner of the current page.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [quadrilateralPoints](#) (page 86)

Declared In

PDFAnnotationMarkup.h

Constants

The styles available for markup annotations in PDF Kit:

Constant	Description
kPDFMarkupTypeHighlight	Highlight style for the markup. Available in Mac OS X v10.5 and later. Declared in PDFAnnotationMarkup.h.
kPDFMarkupTypeStrikeOut	Strikethrough style for the markup. Available in Mac OS X v10.5 and later. Declared in PDFAnnotationMarkup.h.
kPDFMarkupTypeUnderline	Underline style for the markup. Available in Mac OS X v10.5 and later. Declared in PDFAnnotationMarkup.h.

PDFAnnotationPopup Class Reference

Inherits from	PDFAnnotation : NSObject
Conforms to	NSObject (NSObject)
Framework	Library/Frameworks/Quartz.framework/Frameworks/PDFKit.framework
Declared in	PDFKit/PDFAnnotationPopup.h
Availability	Available in Mac OS X v10.5 and later.

Overview

A `PDFAnnotationPopup` object provides user interactivity on a PDF page in the form of a pop-up menu.

Tasks

Accessing and Setting the Open State

- [isOpen](#) (page 89)
Returns a Boolean value indicating whether the pop-up is open.
- [setIsOpen:](#) (page 90)
Sets the open state of the pop-up menu.

Instance Methods

isOpen

Returns a Boolean value indicating whether the pop-up is open.

- (BOOL)isOpen

Return Value

YES if the pop-up is open; NO otherwise.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [setIsOpen](#): (page 90)

Declared In

PDFAnnotationPopup.h

setIsOpen:

Sets the open state of the pop-up menu.

- (void)setIsOpen:(BOOL)*isOpen*

Parameters

isOpen

Pass YES to set the pop-up menu to open; NO otherwise.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [isOpen](#) (page 89)

Declared In

PDFAnnotationPopup.h

PDFAnnotationSquare Class Reference

Inherits from	PDFAnnotation : NSObject
Conforms to	NSObject (NSObject)
Framework	Library/Frameworks/Quartz.framework/Frameworks/PDFKit.framework
Declared in	PDFKit/PDFAnnotationSquare.h
Availability	Available in Mac OS X v10.4 and later.

Overview

A PDFAnnotationSquare object displays a rectangle on a page. Square annotations are like circle annotations (instances of the PDFAnnotationCircle class) apart from the shape.

The [setLineWidth:](#) (page 110) and [setStyle:](#) (page 110) methods of the annotation's associated PDFBorder object determines the stroke thickness and style. The [setColor:](#) (page 40) method of the PDFAnnotation class determines the stroke color.

Tasks

Accessor Methods

- [interiorColor](#) (page 91)
Gets the fill color used for drawing the annotation.
- [setInteriorColor:](#) (page 92)
Sets the fill color used for drawing the annotation.

Instance Methods

interiorColor

Gets the fill color used for drawing the annotation.

```
- (NSColor *)interiorColor
```

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setInteriorColor:](#) (page 92)

Declared In

PDFAnnotationSquare.h

setInteriorColor:

Sets the fill color used for drawing the annotation.

- (void)setInteriorColor:(NSColor *)*color*

Availability

Available in Mac OS X v10.4 and later.

See Also

- [interiorColor](#) (page 91)

Declared In

PDFAnnotationSquare.h

PDFAnnotationStamp Class Reference

Inherits from	PDFAnnotation : NSObject
Conforms to	NSObject (NSObject)
Framework	Library/Frameworks/Quartz.framework/Frameworks/PDFKit.framework
Declared in	PDFKit/PDFAnnotationStamp.h
Availability	Available in Mac OS X v10.5 and later.

Overview

A `PDFAnnotationStamp` object allows you to display a word or phrase, such as “Confidential,” in a PDF page.

A `PDFAnnotationStamp` object should have an appearance stream associated with it; otherwise, nothing useful is rendered.

Tasks

Accessing and Setting the Stamp Annotation

- `name` (page 93)
Returns name associated with the stamp annotation.
- `setName:` (page 94)
Sets the name associated with the stamp annotation.

Instance Methods

name

Returns name associated with the stamp annotation.

- (NSString *)name

Discussion

Note that the name value of the stamp annotation is not necessarily identical to the user-visible appearance of the stamp annotation. For example, a stamp annotation that displays “Confidential” on a PDF page may not have a name value of “Confidential”.

Availability

Available in Mac OS X v10.5 and later.

See Also

- setName:

Declared In

PDFAnnotationStamp.h

setName:

Sets the name associated with the stamp annotation.

```
- (NSString *)setName:(NSString *)name
```

Discussion

The name must be representable in ASCII. You can set a stamp annotation’s name to help you identify it, but that name is not displayed on the PDF page. You must provide the string you want displayed on the page, such as “Draft” or “Top Secret”, in the appearance stream for the annotation.

Availability

Available in Mac OS X v10.5 and later.

See Also

- name

Declared In

PDFAnnotationStamp.h

PDFAnnotationText Class Reference

Inherits from	PDFAnnotation : NSObject
Conforms to	NSObject (NSObject)
Framework	Library/Frameworks/Quartz.framework/Frameworks/PDFKit.framework
Declared in	PDFKit/PDFAnnotationText.h
Availability	Available in Mac OS X v10.4 and later.

Overview

A `PDFAnnotationText` object displays as an icon (such as a “sticky note”) attached to a specified point in the PDF document.

Each `PDFAnnotationText` object has a `PDFAnnotationPopup` object associated with it. In its closed state, the annotation appears as an icon. In its open state, it displays as a pop-up window containing the text of the note. Note that your application must do the work to put up a window containing the text in response to a `PDFViewAnnotationHitNotification` (page 219). Currently, text annotations do not scale and rotate with the page.

Tasks

Managing the Annotation’s State

- `windowIsOpen` (page 97)
Returns a Boolean value indicating whether the annotation’s note window is open. (**Deprecated.** Call `isOpen` (page 89) on the annotation’s pop-up instead.)
- `setWindowIsOpen:` (page 96)
Sets the open/closed state of the annotation to the specified value. (**Deprecated.** Call `setIsOpen:` (page 90) on the annotation’s pop-up instead.)

Managing the Annotation Icon’s Type

- `iconType` (page 96)
Returns the icon type for the annotation.

- [setIconType:](#) (page 96)
Sets the icon type for the annotation.

Instance Methods

iconType

Returns the icon type for the annotation.

- (PDFTextAnnotationIconType)iconType

Return Value

The icon type of the annotation. See “[Constants](#)” (page 97) for a list of possible return values.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setIconType:](#) (page 96)

Declared In

PDFAnnotationText.h

setIconType:

Sets the icon type for the annotation.

- (void)setIconType:(PDFTextAnnotationIconType)type

Parameters

type

The icon type for the annotation. See “[Constants](#)” (page 97) for a list of the available icon types.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [iconType](#) (page 96)

Declared In

PDFAnnotationText.h

setWindowIsOpen:

Sets the open/closed state of the annotation to the specified value. (**Deprecated.** Call [setIsOpen:](#) (page 90) on the annotation’s pop-up instead.)

- (void)setWindowIsOpen:(BOOL)isOpen

Discussion

This method does not actually open or close the annotation. Use it to record annotation state.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [windowIsOpen](#) (page 97)

Declared In

PDFAnnotationText.h

windowIsOpen

Returns a Boolean value indicating whether the annotation's note window is open. (**Deprecated.** Call [isOpen](#) (page 89) on the annotation's pop-up instead.)

- (BOOL)windowIsOpen

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setWindowIsOpen:](#) (page 96)

Declared In

PDFAnnotationText.h

Constants

Text annotations can use the following icon types:

Constant	Description
kPDFTextAnnotationIconComment	Comment annotation icon. Available in Mac OS X v10.4 and later. Declared in PDFAnnotationText.h.
kPDFTextAnnotationIconKey	Key annotation icon. Available in Mac OS X v10.4 and later. Declared in PDFAnnotationText.h.
kPDFTextAnnotationIconNote	Note annotation icon. Available in Mac OS X v10.4 and later. Declared in PDFAnnotationText.h.

Constant	Description
<code>kPDFTextAnnotationIconHelp</code>	Help annotation icon. Available in Mac OS X v10.4 and later. Declared in <code>PDFAnnotationText.h</code> .
<code>kPDFTextAnnotationIconNewParagraph</code>	New Paragraph annotation icon. Available in Mac OS X v10.4 and later. Declared in <code>PDFAnnotationText.h</code> .
<code>kPDFTextAnnotationIconParagraph</code>	Paragraph annotation icon. Available in Mac OS X v10.4 and later. Declared in <code>PDFAnnotationText.h</code> .
<code>kPDFTextAnnotationIconInsert</code>	Insert annotation icon. Available in Mac OS X v10.4 and later. Declared in <code>PDFAnnotationText.h</code> .

PDFAnnotationTextWidget Class Reference

Inherits from	PDFAnnotation : NSObject
Conforms to	NSObject (NSObject)
Framework	Library/Frameworks/Quartz.framework/Frameworks/PDFKit.framework
Declared in	PDFKit/PDFAnnotationTextWidget.h
Availability	Available in Mac OS X v10.4 and later.

Overview

A `PDFAnnotationTextWidget` object allows you to manage the appearance and content of text fields.

`PDFAnnotationTextWidget` objects support interactive forms in a PDF document. This object is comparable to an editable `NSTextField` in Cocoa or an edit text view in Carbon.

Tasks

Working with Annotation Strings

- [stringValue](#) (page 106)
Returns the string assigned to the annotation.
- [setStringValue:](#) (page 106)
Sets the string for the annotation.
- [maxLength](#) (page 102)
Returns the maximum number of characters allowed in the annotation string.
- [setMaxLength:](#) (page 105)
Sets the maximum number of characters allowed in the annotation string.

Managing the Font and Font Color

- [font](#) (page 101)
Returns the font used for the annotation's text field.
- [setFont:](#) (page 104)
Sets the font used in the text field of the annotation.

- [fontColor](#) (page 102)
Returns the font color used for the annotation's text field.
- [setFontColor:](#) (page 105)
Sets the font color used for the annotation's text field.

Managing Background Color, Alignment, and Rotation

- [backgroundColor](#) (page 101)
Returns the background color of the annotation text field.
- [setBackgroundcolor:](#) (page 103)
Sets the background color of the annotation text field.
- [alignment](#) (page 100)
Returns the text alignment setting for the annotation.
- [setAlignment:](#) (page 103)
Sets the text alignment for the annotation.
- [rotation](#) (page 103)
Returns the rotation angle of the annotation text field in degrees.
- [setRotation:](#) (page 105)
Sets the rotation angle of the annotation text field in degrees.

Working with Field Names

- [fieldName](#) (page 101)
Returns the internal name for the annotation text field.
- [setFieldName:](#) (page 104)
Sets the internal field name for the annotation text field.

Instance Methods

alignment

Returns the text alignment setting for the annotation.

- (NSTextAlignment)alignment

Return Value

The text alignment value for the annotation. Supported alignment values are `NSLeftTextAlignment`, `NSRightTextAlignment`, and `NSCenterTextAlignment`.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setAlignment:](#) (page 103)

Declared In

PDFAnnotationTextWidget.h

backgroundColor

Returns the background color of the annotation text field.

- (NSColor *)backgroundColor

Return Value

The background color of the annotation's text field.

Availability

Available in Mac OS X v10.5 and later.

See Also

- setBackgroundColor:

Declared In

PDFAnnotationTextWidget.h

fieldName

Returns the internal name for the annotation text field.

- (NSString *)fieldName

Return Value

The internal name for the annotation text field.

Discussion

Field names are optional, internal names that identify text fields in a PDF form. You use field names with the `PDFActionResetForm` action.

Note that multiple `PDFAnnotationTextWidget` objects with the same field name always have the same text associated with that field name. When text is entered into one of the objects, the text associated with that field name is changed in all objects. If you need to ensure unique text for a `PDFAnnotationTextWidget` object, you must give it a unique field name (you can use [setFieldName:](#) (page 104) to do this).

Availability

Available in Mac OS X v10.5 and later.

See Also

- setFieldName:

Declared In

PDFAnnotationTextWidget.h

font

Returns the font used for the annotation's text field.

- (NSFont *)font

Return Value

The font used for text in the annotation's text field.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setFont:](#) (page 104)

Declared In

PDFAnnotationTextWidget.h

fontColor

Returns the font color used for the annotation's text field.

- (NSColor *)fontColor

Return Value

The font color used for text in the annotation's text field.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [setFontColor:](#)

Declared In

PDFAnnotationTextWidget.h

maxLength

Returns the maximum number of characters allowed in the annotation string.

- (NSInteger)maxLength

Return Value

The maximum number of characters allowed in the annotations string. A return value of 0 means that there is no specified maximum.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [setMaxLength:](#)

Declared In

PDFAnnotationTextWidget.h

rotation

Returns the rotation angle of the annotation text field in degrees.

- (int)rotation

Return Value

The rotation angle of the annotation text field in degrees.

Discussion

Note that the rotation value is a positive multiple of 90, such as 0, 90, 180, or 270. The rotation of annotation text fields with negative rotation is converted to a corresponding positive rotation. For example, -90 is changed to 270.

Availability

Available in Mac OS X v10.5 and later.

See Also

- `setRotation:`

Declared In

PDFAnnotationTextWidget.h

setAlignment:

Sets the text alignment for the annotation.

- (void)setAlignment:(NSTextAlignment)*alignment*

Parameters

alignment

The text-alignment value to be used for the annotation. Possible values are `NSLeftTextAlignment`, `NSRightTextAlignment`, and `NSCenterTextAlignment`.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [alignment](#) (page 100)

Declared In

PDFAnnotationTextWidget.h

setBackground-color:

Sets the background color of the annotation text field.

- (void)setBackgroundColor:(NSColor *)*color*

Parameters

color

The color to be used in the background of the annotation's text field.

Availability

Available in Mac OS X v10.5 and later.

See Also

- `backgroundColor`

Declared In

`PDFAnnotationTextWidget.h`

setFieldName:

Sets the internal field name for the annotation text field.

- (void)setFieldName:(NSString *)*name*

Parameters

name

The internal field name to be used for the annotation text field.

Discussion

Field names are optional, internal names that identify text fields in a PDF form. You use field names with the `PDFActionResetForm` action.

Note that multiple `PDFAnnotationTextWidget` objects with the same field name always have the same text associated with that field name. When text is entered into one of the objects, the text associated with that field name is changed in all objects. If you need to ensure unique text for a `PDFAnnotationTextWidget` object, you must give it a unique field name.

Availability

Available in Mac OS X v10.5 and later.

See Also

- `fieldName`

Declared In

`PDFAnnotationTextWidget.h`

setFont:

Sets the font used in the text field of the annotation.

- (void)setFont:(NSFont *)*font*

Parameters

font

The font to be used in the annotation's text field.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [font](#) (page 101)

Declared In

PDFAnnotationTextWidget.h

setFontColor:

Sets the font color used for the annotation's text field.

```
- (void)setFontColor:(NSColor *)color
```

Parameters

color

The font color to be used in the annotation's text field.

Availability

Available in Mac OS X v10.5 and later.

See Also

- `fontColor`

Declared In

PDFAnnotationTextWidget.h

setMaximumLength:

Sets the maximum number of characters allowed in the annotation string.

```
- (void)setMaximumLength:(NSUInteger)maxLen
```

Parameters

maxLen

The maximum number of characters allowed in the annotation string. Pass 0 to indicate that there is no specified maximum.

Availability

Available in Mac OS X v10.5 and later.

See Also

- `maxLength`

Declared In

PDFAnnotationTextWidget.h

setRotation:

Sets the rotation angle of the annotation text field in degrees.

```
- (void)setRotation:(int)rotation
```

Parameters*rotation*

The rotation angle to be applied to the annotation text field, in degrees. The rotation angle must be a positive or negative multiple of 90 (negative angles are converted to their positive equivalents; for example -90 is changed to 270).

Availability

Available in Mac OS X v10.5 and later.

See Also

- [rotation](#)

Declared In

PDFAnnotationTextWidget.h

setStringValue:

Sets the string for the annotation.

- (void)setStringValue:(NSString *)*value*

Parameters*value*

The string to be assigned to the annotation.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [stringValue](#) (page 106)

Declared In

PDFAnnotationTextWidget.h

stringValue

Returns the string assigned to the annotation.

- (NSString *)stringValue

Return Value

The string assigned to the annotation.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setStringValue:](#) (page 106)

Declared In

PDFAnnotationTextWidget.h

PDFBorder Class Reference

Inherits from	NSObject
Conforms to	NSObject (NSObject)
Framework	Library/Frameworks/Quartz.framework/Frameworks/PDFKit.framework
Declared in	PDFKit/PDFBorder.h
Availability	Available in Mac OS X v10.4 and later.

Overview

A PDFBorder object, when used, adds an optional border to an annotation. Borders are drawn completely within the annotation rectangle.

Tasks

Working with Border Styles and Characteristics

- [style](#) (page 111)
Gets the border style.
- [setStyle:](#) (page 110)
Sets the border style.
- [lineWidth](#) (page 109)
Gets the line width for the border, in points.
- [setLineWidth:](#) (page 110)
Sets the line width (in points) for the border.
- [horizontalCornerRadius](#) (page 108)
Gets the horizontal corner radius (in points) used for a rounded-rectangle border.
- [setHorizontalCornerRadius:](#) (page 109)
Sets the horizontal corner radius (in points) used for a rounded-rectangle border.
- [verticalCornerRadius](#) (page 111)
Gets the vertical corner radius used for a rounded-rectangle border, in points.
- [setVerticalCornerRadius:](#) (page 110)
Sets the vertical corner radius (in points) used for a rounded-rectangle border.

- [dashPattern](#) (page 108)
Gets the dash pattern for the border as an array of NSNumber objects.
- [setDashPattern:](#) (page 109)
Sets the dash pattern for the border.

Drawing Borders

- [drawInRect:](#) (page 108)
Draws the border.

Instance Methods

dashPattern

Gets the dash pattern for the border as an array of NSNumber objects.

- (NSArray *)dashPattern

Discussion

Refer to the description for `NSBezierPath` for more information.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setDashPattern:](#) (page 109)

Declared In

PDFBorder.h

drawInRect:

Draws the border.

- (void)drawInRect:(NSRect)rect

Availability

Available in Mac OS X v10.4 and later.

Declared In

PDFBorder.h

horizontalCornerRadius

Gets the horizontal corner radius (in points) used for a rounded-rectangle border.

- (float)horizontalCornerRadius

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setHorizontalCornerRadius:](#) (page 109)

Declared In

PDFBorder.h

lineWidth

Gets the line width for the border, in points.

- (float)lineWidth

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setLineWidth:](#) (page 110)

Declared In

PDFBorder.h

setDashPattern:

Sets the dash pattern for the border.

- (void)setDashPattern:(NSArray *)*pattern*

Discussion

Provide *pattern* as an array of NSNumber objects. Refer to the description for NSBezierPath for more information.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [dashPattern](#) (page 108)

Declared In

PDFBorder.h

setHorizontalCornerRadius:

Sets the horizontal corner radius (in points) used for a rounded-rectangle border.

- (void)setHorizontalCornerRadius:(float)*radius*

Availability

Available in Mac OS X v10.4 and later.

See Also

- [horizontalCornerRadius](#) (page 108)

Declared In

PDFBorder.h

setLineWidth:

Sets the line width (in points) for the border.

```
- (void)setLineWidth:(float)width
```

Availability

Available in Mac OS X v10.4 and later.

See Also

- [lineWidth](#) (page 109)

Declared In

PDFBorder.h

setStyle:

Sets the border style.

```
- (void)setStyle:(PDFBorderStyle)style
```

Discussion

Refer to “[Constants](#)” (page 111) for the available border styles.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [style](#) (page 111)

Declared In

PDFBorder.h

setVerticalCornerRadius:

Sets the vertical corner radius (in points) used for a rounded-rectangle border.

```
- (void)setVerticalCornerRadius:(float)radius
```

Availability

Available in Mac OS X v10.4 and later.

See Also

- [verticalCornerRadius](#) (page 111)

Declared In

PDFBorder.h

style

Gets the border style.

- (PDFBorderStyle)style

DiscussionSee “[Constants](#)” (page 111) for possible return values.**Availability**

Available in Mac OS X v10.4 and later.

See Also- [setStyle:](#) (page 110)**Declared In**

PDFBorder.h

verticalCornerRadius

Gets the vertical corner radius used for a rounded-rectangle border, in points.

- (float)verticalCornerRadius

Availability

Available in Mac OS X v10.4 and later.

See Also- [setVerticalCornerRadius:](#) (page 110)**Declared In**

PDFBorder.h

Constants

PDF Kit annotation borders may have the following styles:

Constant	Description
kPDFBorderStyleSolid	Solid border. Available in Mac OS X v10.4 and later. Declared in PDFBorder.h.
kPDFBorderStyleDashed	Dashed border. Available in Mac OS X v10.4 and later. Declared in PDFBorder.h.

Constant	Description
<code>kPDFBorderStyleBeveled</code>	Beveled border. Available in Mac OS X v10.4 and later. Declared in <code>PDFBorder.h</code> .
<code>kPDFBorderStyleInset</code>	Inset border. Available in Mac OS X v10.4 and later. Declared in <code>PDFBorder.h</code> .
<code>kPDFBorderStyleUnderline</code>	Underline border. Available in Mac OS X v10.4 and later. Declared in <code>PDFBorder.h</code> .

PDFDestination Class Reference

Inherits from	NSObject
Conforms to	NSCopying NSObject (NSObject)
Framework	Library/Frameworks/Quartz.framework/Frameworks/PDFKit.framework
Availability	Available in Mac OS X v10.4 and later.
Declared in	PDFKit/PDFDestination.h

Overview

A `PDFDestination` object describes a point on a PDF page.

In typical usage, you do not initialize `PDFDestination` objects but rather get them as either attributes of `PDFAnnotationLink` or `PDFOutline` objects, or in response to the `PDFView` method `currentDestination` (page 195).

Tasks

Initializing a Destination

- `initWithPage:atPoint:` (page 114)
Initializes the destination.

Getting Pages and Points

- `page` (page 115)
Returns the page that the destination refers to.
- `point` (page 115)
Returns the point, in page space, that the destination refers to.

Getting a Relative Location

- `compare:` (page 114)

Returns a comparison result that indicates the location of the destination in the document, relative to the current position.

Instance Methods

compare:

Returns a comparison result that indicates the location of the destination in the document, relative to the current position.

- (NSComparisonResult)compare:(PDFDestination *)*destination*

Parameters

destination

The destination in the document to be located.

Return Value

A comparison result, indicating the position of the passed-in destination relative to the current position.

Discussion

If *destination* is between the receiver's position and the end of the document, `compare` returns `NSOrderedAscending`; if it is between the receiver's position and the beginning of the document, `compare` returns `NSOrderedDescending`. Otherwise, if *destination* matches the receiver's position, `compare` returns `NSOrderedSame`.

This method ignores the horizontal component of the destination point (the x value). If the destination's vertical component (or y value) is `kPDFDestinationUnspecifiedValue`, `compare` treats the destination as if its y value is the top point on the destination page.

An exception is raised if *destination* does not have a page associated with it or if its page is associated with a document other than the receiver's document.

Availability

Available in Mac OS X v10.5 and later.

Declared In

PDFDestination.h

initWithPage:atPoint:

Initializes the destination.

- (id)initWithPage:(PDFPage *)*page* atPoint:(NSPoint)*point*

Parameters

page

The page of the destination.

point

The point of the destination, in page space.

Return Value

An initialized `PDFDestination` instance, or `NULL` if the object could not be initialized.

Discussion

Specify *point* in page space. Typically, there's no need to initialize destinations. Instead, you get them from `PDFAnnotationLink`, `PDFOutline`, or `PDFView` objects.

Page space is a 72-dpi coordinate system with the origin at the lower-left corner of the current page.

Availability

Available in Mac OS X v10.4 and later.

Declared In

`PDFDestination.h`

page

Returns the page that the destination refers to.

- (`PDFPage *`)page

Return Value

The page referred to by the destination.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [point](#) (page 115)

Declared In

`PDFDestination.h`

point

Returns the point, in page space, that the destination refers to.

- (`NSPoint`)point

Return Value

The point, in page space, referred to by the destination.

Discussion

Page space is a 72 dpi coordinate system with the origin at the lower-left corner of the current page.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [page](#) (page 115)

Declared In

PDFDestination.h

Constants

Destination Undefined

Value used for unspecified destination.

```
#define kPDFDestinationUnspecifiedValue          FLT_MAX
```

Constants

`kPDFDestinationUnspecifiedValue`

Unspecified value used when a destination's actual x or y value is unimportant.

Available in Mac OS X v10.5 and later.

Declared in `PDFDestination.h`.

PDFDocument Class Reference

Inherits from	NSObject
Conforms to	NSObject (NSObject)
Framework	Library/Frameworks/Quartz.framework/Frameworks/PDFKit.framework
Availability	Available in Mac OS X v10.4 and later.
Declared in	PDFKit/PDFDocument.h

Overview

A `PDFDocument` object represents PDF data or a PDF file and defines methods for writing, searching, and selecting PDF data.

The other utility classes are either instantiated from methods in `PDFDocument`, as are `PDFPage` and `PDFOutline`; or support it, as do `PDFSelection` and `PDFDestination`.

You initialize a `PDFDocument` object with PDF data or with a URL to a PDF file. You can then ask for the page count, add or delete pages, perform a find, or parse selected content into an `NSString` object.

Tasks

Initializing Documents

- `initWithData:` (page 126)
Initializes a `PDFDocument` object with the passed-in data.
- `initWithURL:` (page 126)
Initializes a `PDFDocument` object with the contents at the specified URL (if the URL is invalid, this method returns `NULL`).

Accessing Document Information

- `documentURL` (page 123)
Returns the URL for the document.
- `majorVersion` (page 128)
Returns the major version of the document.

- `minorVersion` (page 128)
Returns the minor version of the document.
- `string` (page 133)
Returns a string representing the textual content for the entire document.
- `outlineItemForSelection:` (page 129)
Returns the most likely parent PDF outline object for the selection.
- `outlineRoot` (page 129)
Returns the root PDF outline object for the document.
- `documentAttributes` (page 123)
Returns a dictionary of document metadata.
- `setDocumentAttributes:` (page 133)
Sets the document attributes.
- `setOutlineRoot:` (page 133)
Sets the document's root outline to a PDF outline object.

Managing Document Security

- `isEncrypted` (page 127)
Returns a Boolean value specifying whether the document is encrypted.
- `isLocked` (page 128)
Returns a Boolean value indicating whether the document is locked.
- `unlockWithPassword:` (page 134)
Attempts to unlock an encrypted document.
- `allowsCopying` (page 120)
Returns a Boolean value indicating whether the document allows copying of content to the Pasteboard.
- `allowsPrinting` (page 121)
Returns a Boolean value indicating whether the document allows printing.

Writing Out the PDF Data

- `dataRepresentation` (page 122)
Returns a representation of the document as an `NSData` object.
- `writeToFile:` (page 134)
Writes the document to a file at the specified path.
- `writeToFile:withOptions:` (page 135)
Writes the document to a file at the specified path with the specified options.
- `writeToURL:` (page 135)
Writes the document to a location specified by the passed-in URL.
- `writeToURL:withOptions:` (page 136)
Writes the document to the specified URL with the specified options.

Working with Pages

- `pageCount` (page 130)
Returns the number of pages in the document.
- `pageAtIndex:` (page 130)
Returns the page at the specified index number.
- `indexOfPage:` (page 125)
Gets the index number for the specified page.
- `insertPage:atIndex:` (page 126)
Inserts a page at the specified index point.
- `removePageAtIndex:` (page 131)
Removes the page at the specified index point.
- `exchangePageAtIndex:withPageAtIndex:` (page 124)
Swaps one page with another.

Managing Find Operations

- `findString:withOptions:` (page 125)
Synchronously finds all instances of the specified string in the document.
- `beginFindString:withOptions:` (page 121)
Asynchronously finds all instances of the specified string in the document.
- `beginFindStrings:withOptions:` (page 122)
Asynchronously finds all instances of the specified array of strings in the document.
- `findString:fromSelection:withOptions:` (page 124)
Synchronously finds the next occurrence of a string after the specified selection (or before the selection if you specified `NSBackwardsSearch` as a search option).
- `isFinding` (page 127)
Returns a Boolean value indicating whether an asynchronous find operation is in progress.
- `cancelFindString` (page 122)
Cancels a search initiated with `beginFindString:withOptions:` (page 121).

Working with Selections

- `selectionFromPage:atCharacterIndex:toPage:atCharacterIndex:` (page 131)
Returns the specified selection based on starting and ending character indexes.
- `selectionFromPage:atPoint:toPage:atPoint:` (page 132)
Returns the specified selection based on starting and ending points.
- `selectionForEntireDocument` (page 131)
Returns a selection representing the textual content of the entire document.

Setting the Delegate

- `setDelegate:` (page 132)
Establishes the specified object as the delegate for the `PDFDocument` object.
- `delegate` (page 123)
Returns the object acting as the delegate for the `PDFDocument` object.

Searching Documents

- `didMatchString:` (page 136) *delegate method*
Called for every match found during a find operation.
- `documentDidBeginDocumentFind:` (page 136) *delegate method*
Called when the `PDFDocumentDidBeginFindNotification` notification is posted.
- `documentDidBeginPageFind:` (page 137) *delegate method*
Called when the `PDFDocumentDidBeginPageFindNotification` notification is posted.
- `documentDidEndDocumentFind:` (page 137) *delegate method*
Called when the `PDFDocumentDidEndFindNotification` notification is posted.
- `documentDidEndPageFind:` (page 137) *delegate method*
Called when the `PDFDocumentDidEndPageFindNotification` notification is posted.
- `documentDidFindMatch:` (page 138) *delegate method*
Called when the `PDFDocumentDidFindMatchNotification` notification is posted.

Unlocking Documents

- `documentDidUnlock:` (page 138) *delegate method*
Called when the `PDFDocumentDidUnlockNotification` notification is posted.

Determining the Page Class

- `pageClass` (page 130)
Returns the class that is allocated and initialized when page objects are created for the document.

Instance Methods

allowsCopying

Returns a Boolean value indicating whether the document allows copying of content to the Pasteboard.

- (BOOL)allowsCopying

Discussion

The ability to copy content from a PDF document is an attribute unrelated to whether the document is locked or unlocked. It depends on the PDF permissions set by the document's author.

This method only determines the desired permissions setting in the PDF document; it is up to the application to enforce (or ignore) the permissions.

This method always returns `YES` if the document is not encrypted. Note that in many cases an encrypted document may still be readable by all users due to the standard empty string password. For more details about user and owner passwords, see the Adobe PDF specification.

Availability

Available in Mac OS X v10.4 and later.

Declared In

PDFDocument.h

allowsPrinting

Returns a Boolean value indicating whether the document allows printing.

- (BOOL)allowsPrinting

Discussion

The ability to print a PDF document is an attribute unrelated to whether the document is locked or unlocked. It depends on the PDF permissions set by the document's author.

This method only determines the desired permissions setting in the PDF document; it is up to the application to enforce (or ignore) the permissions.

This method always returns `YES` if the document is not encrypted. Note that in many cases an encrypted document may still be readable by all users due to the standard empty string password. For more details about user and owner passwords, see the Adobe PDF specification.

Availability

Available in Mac OS X v10.4 and later.

Declared In

PDFDocument.h

beginFindString:withOptions:

Asynchronously finds all instances of the specified string in the document.

- (void)beginFindString:(NSString *)string withOptions:(int)options

Discussion

This method returns immediately. It causes notifications to be issued when searching begins and ends, on each search hit, and when the search proceeds to a new page. For options, refer to [Searching and Comparing Strings](#).

Availability

Available in Mac OS X v10.4 and later.

See Also

- [findString:withOptions:](#) (page 125)
- [isFinding](#) (page 127)

- [cancelFindString](#) (page 122)

Declared In

PDFDocument.h

beginFindStrings:withOptions:

Asynchronously finds all instances of the specified array of strings in the document.

```
- (void)beginFindStrings:(NSArray *)strings withOptions:(int)options;
```

Discussion

This method returns immediately. It causes notifications to be issued when searching begins and ends, on each search hit, and when the search proceeds to a new page. For options, refer to [Searching and Comparing Strings](#).

Availability

Available in Mac OS X v10.5 and later.

See Also

- [beginFindString:withOptions:](#)
- [findString:withOptions:](#) (page 125)
- [isFinding](#) (page 127)
- [cancelFindString](#) (page 122)

Declared In

PDFDocument.h

cancelFindString

Cancels a search initiated with [beginFindString:withOptions:](#) (page 121).

```
- (void)cancelFindString
```

Availability

Available in Mac OS X v10.4 and later.

See Also

- [findString:withOptions:](#) (page 125)
- [beginFindString:withOptions:](#) (page 121)
- [isFinding](#) (page 127)

Declared In

PDFDocument.h

dataRepresentation

Returns a representation of the document as an `NSData` object.

```
- (NSData *)dataRepresentation
```

Availability

Available in Mac OS X v10.4 and later.

See Also

- [writeToFile:](#) (page 134)
- [writeToURL:](#) (page 135)

Declared In

PDFDocument.h

delegate

Returns the object acting as the delegate for the PDFDocument object.

- (id)delegate

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setDelegate:](#) (page 132)

Declared In

PDFDocument.h

documentAttributes

Returns a dictionary of document metadata.

- (NSDictionary *)documentAttributes

Return Value

The dictionary of document metadata. The dictionary may be empty, or only some of the keys may have associated values. See “[Constants](#)” (page 138) for a list of possible key words.

Discussion

Metadata is optional for PDF documents.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setDocumentAttributes:](#) (page 133)

Declared In

PDFDocument.h

documentURL

Returns the URL for the document.

- (NSURL *)documentURL

Return Value

The URL for the document; may return `NULL` if the document was created from an `NSData` object.

Availability

Available in Mac OS X v10.4 and later.

Declared In

`PDFDocument.h`

exchangePageAtIndex:withPageAtIndex:

Swaps one page with another.

```
- (void)exchangePageAtIndex:(NSUInteger) indexA withPageAtIndex:(NSUInteger) indexB
```

Discussion

This method raises an exception if either *index* value is out of bounds.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [pageCount](#) (page 130)
- [pageAtIndex:](#) (page 130)
- [indexForPage:](#) (page 125)
- [insertPage:atIndex:](#) (page 126)
- [removePageAtIndex:](#) (page 131)

Declared In

`PDFDocument.h`

findString:fromSelection:withOptions:

Synchronously finds the next occurrence of a string after the specified selection (or before the selection if you specified `NSBackwardsSearch` as a search option).

```
- (PDFSelection *)findString:(NSString *) string fromSelection:(PDFSelection *) selection withOptions:(int) options
```

Discussion

Matches are returned as a `PDFSelection` object. If the search reaches the end (or beginning) of the document without any hits, this method returns `NULL`.

If you pass `NULL` for the selection, this method begins searching from the beginning of the document (or the end, if you specified `NSBackwardsSearch`).

You can use this method to implement “Find Again” behavior. For options, refer to [Searching and Comparing Strings](#).

Availability

Available in Mac OS X v10.4 and later.

See Also

- [isFinding](#) (page 127)
- [findString:withOptions:](#) (page 125)

Declared In

PDFDocument.h

findString:withOptions:

Synchronously finds all instances of the specified string in the document.

```
- (NSArray *)findString:(NSString *)string withOptions:(int)options
```

Discussion

Each hit gets added to an `NSArray` object as a `PDFSelection` object. If there are no hits, this method returns an empty array.

Use this method when the complete search process will be brief and when you don't need the flexibility or control offered by [beginFindString:withOptions:](#) (page 121). For options, refer to [Searching and Comparing Strings](#).

Availability

Available in Mac OS X v10.4 and later.

See Also

- [isFinding](#) (page 127)
- [findString:fromSelection:withOptions:](#) (page 124)

Declared In

PDFDocument.h

indexForPage:

Gets the index number for the specified page.

```
- (NSUInteger)indexForPage:(PDFPage *)page
```

Discussion

Indexes are zero-based. This method raises an exception and returns `NSNotFound` if *page* is not found.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [pageCount](#) (page 130)
- [pageAtIndex:](#) (page 130)
- [insertPage:atIndex:](#) (page 126)
- [removePageAtIndex:](#) (page 131)
- [exchangePageAtIndex:withPageAtIndex:](#) (page 124)

Declared In

PDFDocument.h

initWithData:

Initializes a `PDFDocument` object with the passed-in data.

```
- (id)initWithData:(NSData *)data
```

Return Value

A `PDFDocument` instance initialized with the passed-in PDF data, or `NULL` if the object could not be initialized.

Discussion

The data must be PDF data encapsulated in an `NSData` object; otherwise this method returns `NULL`.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [initWithURL:](#) (page 126)

Declared In

`PDFDocument.h`

initWithURL:

Initializes a `PDFDocument` object with the contents at the specified URL (if the URL is invalid, this method returns `NULL`).

```
- (id)initWithURL:(NSURL *)url
```

Return Value

A `PDFDocument` instance initialized with the data at the passed-in URL or `NULL` if the object could not be initialized or if the URL is invalid.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [initWithData:](#) (page 126)

Declared In

`PDFDocument.h`

insertPage:atIndex:

Inserts a page at the specified index point.

```
- (void)insertPage:(PDFPage *)page atIndex:(NSUInteger)index
```

Discussion

This method raises an exception if `index` is out of bounds.

Be aware that a PDF viewing application might use the size of the first page in the document as representative of all page sizes when reporting the size of a document. If you need to get the actual size of an individual page, you can use [boundsForBox:](#) (page 157) (note that the size is returned in points, which are typically converted to inches or centimeters by PDF viewing applications).

Availability

Available in Mac OS X v10.4 and later.

See Also

- [pageCount](#) (page 130)
- [pageAtIndex:](#) (page 130)
- [indexForPage:](#) (page 125)
- [removePageAtIndex:](#) (page 131)
- [exchangePageAtIndex:withPageAtIndex:](#) (page 124)

Declared In

PDFDocument.h

isEncrypted

Returns a Boolean value specifying whether the document is encrypted.

- (BOOL)isEncrypted

Return Value

YES if the document is encrypted, whether it is locked or unlocked; NO otherwise.

Discussion

If encrypted, reading the document requires a password.

Encrypted documents whose password is the empty string are unlocked automatically upon opening, because PDF Kit tries the empty string as a password if none is supplied. Use the [unlockWithPassword:](#) (page 134) method to unlock a document using a password.

Availability

Available in Mac OS X v10.4 and later.

Declared In

PDFDocument.h

isFinding

Returns a Boolean value indicating whether an asynchronous find operation is in progress.

- (BOOL)isFinding

Availability

Available in Mac OS X v10.4 and later.

See Also

- [beginFindString:withOptions:](#) (page 121)
- [cancelFindString](#) (page 122)

Declared In

PDFDocument.h

isLocked

Returns a Boolean value indicating whether the document is locked.

- (BOOL)isLocked

Return Value

YES if the document is locked; NO otherwise.

Discussion

Only encrypted documents can be locked. Encrypted documents whose password is the empty string are unlocked automatically upon opening, because PDF Kit tries the empty string as a password if none is supplied. Use the [unlockWithPassword:](#) (page 134) method to unlock a document using a password.

Availability

Available in Mac OS X v10.4 and later.

Declared In

PDFDocument.h

majorVersion

Returns the major version of the document.

- (int)majorVersion

Return Value

The major version of the document.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [minorVersion](#) (page 128)

Declared In

PDFDocument.h

minorVersion

Returns the minor version of the document.

- (int)minorVersion

Return Value

The minor version of the document.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [majorVersion](#) (page 128)

Declared In

PDFDocument.h

outlineItemForSelection:

Returns the most likely parent PDF outline object for the selection.

```
- (PDFOutline *)outlineItemForSelection:(PDFSelection *)selection
```

Parameters*selection*

The area of the document currently selected by the user. A selection can span multiple outline items, but only the point representing the first character is considered.

Return Value

The PDF outline object that is the most likely parent of the specified selection. Note that only the point representing the first character of the selection is considered in this method.

Discussion

Typically, outlines represent structural items such as chapters. You can use this method to identify the chapter that a selection falls within.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [outlineRoot](#) (page 129)

Declared In

PDFDocument.h

outlineRoot

Returns the root PDF outline object for the document.

```
- (PDFOutline *)outlineRoot
```

Return Value

The root outline object or NULL if there is no root outline object. The root outline is the nonvisible top-level container for outline items.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [outlineItemForSelection:](#) (page 129)

- [setOutlineRoot:](#)

Declared In

PDFDocument.h

pageAtIndex:

Returns the page at the specified index number.

- (PDFPage *)pageAtIndex:(NSInteger) *index*

Discussion

Indexes are zero based. This method raises an exception if *index* is out of bounds.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [pageCount](#) (page 130)
- [indexForPage:](#) (page 125)
- [insertPageAtIndex:](#) (page 126)
- [removePageAtIndex:](#) (page 131)
- [exchangePageAtIndex:withPageAtIndex:](#) (page 124)

Declared In

PDFDocument.h

pageClass

Returns the class that is allocated and initialized when page objects are created for the document.

- (Class)pageClass

Discussion

If you want to supply a custom page class, subclass `PDFDocument` and implement this method to return your custom class. Note that your custom class must be a subclass of `PDFPage`; otherwise, the behavior is undefined.

The default implementation of `pageClass` returns `[PDFPage class]`.

Availability

Available in Mac OS X v10.5 and later.

Declared In

PDFDocument.h

pageCount

Returns the number of pages in the document.

- (NSInteger)pageCount

Availability

Available in Mac OS X v10.4 and later.

See Also

- [pageAtIndex:](#) (page 130)
- [indexForPage:](#) (page 125)

- [insertPage:atIndex:](#) (page 126)
- [removePageAtIndex:](#) (page 131)
- [exchangePageAtIndex:withPageAtIndex:](#) (page 124)

Declared In

PDFDocument.h

removePageAtIndex:

Removes the page at the specified index point.

- (void)removePageAtIndex:(NSUInteger) *index*

Discussion

This method raises an exception if *index* is out of bounds.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [pageCount](#) (page 130)
- [pageAtIndex:](#) (page 130)
- [indexForPage:](#) (page 125)
- [insertPage:atIndex:](#) (page 126)
- [exchangePageAtIndex:withPageAtIndex:](#) (page 124)

Declared In

PDFDocument.h

selectionForEntireDocument

Returns a selection representing the textual content of the entire document.

- (PDFSelection *)selectionForEntireDocument

Availability

Available in Mac OS X v10.4 and later.

Declared In

PDFDocument.h

selectionFromPage:atCharacterIndex:toPage:atCharacterIndex:

Returns the specified selection based on starting and ending character indexes.

- (PDFSelection *)selectionFromPage:(PDFPage *) *startPage*
 atCharacterIndex:(NSUInteger) *startChar* toPage:(PDFPage *) *endPage*
 atCharacterIndex:(NSUInteger) *endChar*

Discussion

The selection begins at *startChar* on *startPage* and ends at *endChar* on *endPage*. The starting and ending index values must be in the range of the number of characters (as returned by [numberOfCharacters](#) (page 160)) within the respective `PDFPage` objects.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [selectionFromPage:atPoint:toPage:atPoint:](#) (page 132)

Declared In

`PDFDocument.h`

selectionFromPage:atPoint:toPage:atPoint:

Returns the specified selection based on starting and ending points.

```
- (PDFSelection *)selectionFromPage:(PDFPage *)startPage atPoint:(NSPoint)startPt
  toPage:(PDFPage *)endPage atPoint:(NSPoint)endPt
```

Discussion

The selection begins at *startPt* on *startPage* and ends at *endPt* on *endPage*. The starting and ending points should be specified in page space, relative to their respective pages.

The starting and ending points can be on the same page. In this case, invoking this method is equivalent to sending the `selectionFromPoint:toPoint:` message to a `PDFPage` object.

Page space is a 72 dpi coordinate system with the origin at the lower-left corner of the current page.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [selectionFromPage:atCharacterIndex:toPage:atCharacterIndex:](#) (page 131)
 - [selectionForRange:](#) (page 162)

Declared In

`PDFDocument.h`

setDelegate:

Establishes the specified object as the delegate for the `PDFDocument` object.

```
- (void)setDelegate:(id)anObject
```

Availability

Available in Mac OS X v10.4 and later.

See Also

- [delegate](#) (page 123)
 - [didMatchString:](#) (page 136)

Declared In

PDFDocument.h

setDocumentAttributes:

Sets the document attributes.

```
- (void)setDocumentAttributes:(NSDictionary *)attributes
```

Parameters*attributes*

A dictionary containing document attributes as key-value pairs. See “Constants” (page 138) for a list of possible key words.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [documentAttributes](#) (page 123).

Declared In

PDFDocument.h

setOutlineRoot:

Sets the document’s root outline to a PDF outline object.

```
- (void)setOutlineRoot:(PDFOutline *)outline
```

Parameters*outline*

The outline to be used as the document’s root outline. Pass `NULL` to strip the outline from a document.

Discussion

When a PDF document is saved, the outline tree structure is written out to the destination PDF file.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [outlineRoot](#)

Declared In

PDFDocument.h

string

Returns a string representing the textual content for the entire document.

```
- (NSString *)string
```

Return Value

A string that represents the textual content of the entire document.

Discussion

Pages are delimited with linefeed characters.

This is a convenience method, equivalent to creating a selection object for the entire document and then invoking the `PDFSelection` class's `string` (page 173) method.

Availability

Available in Mac OS X v10.4 and later.

Declared In

`PDFDocument.h`

unlockWithPassword:

Attempts to unlock an encrypted document.

```
- (BOOL)unlockWithPassword:(NSString *)password
```

Parameters

password

The password to unlock an encrypted document (you cannot lock an unlocked PDF document by using an incorrect password).

Return Value

YES if the specified password unlocks the document, NO otherwise.

Discussion

If the password is correct, this method returns YES, and a `PDFDocumentDidUnlockNotification` notification is sent. Once unlocked, you cannot use this function to relock the document.

If you attempt to unlock an already unlocked document, one of the following occurs:

- If the document is unlocked with full owner permissions, `unlockWithPassword` does nothing and returns YES. The password string is ignored.
- If the document is unlocked with only user permissions, `unlockWithPassword` attempts to obtain full owner permissions with the password string. If the string fails, the document maintains its user permissions. In either case, this method returns YES.

Availability

Available in Mac OS X v10.4 and later.

Declared In

`PDFDocument.h`

writeToFile:

Writes the document to a file at the specified path.

```
- (BOOL)writeToFile:(NSString *)path
```

Availability

Available in Mac OS X v10.4 and later.

See Also

- [dataRepresentation](#) (page 122)
- [writeToURL:](#) (page 135)
- [writeToURL:withOptions:](#) (page 136)
- [writeToFile:withOptions:](#) (page 135)

Declared In

PDFDocument.h

writeToFile:withOptions:

Writes the document to a file at the specified path with the specified options.

```
- (BOOL)writeToFile:(NSString *)path withOptions:(NSDictionary *)options
```

Discussion

The most commonly-used options are `kCGPDFContextOwnerPassword`, `kCGPDFContextUserPassword`, `kCGPDFContextAllowsCopying` and `kCGPDFContextAllowsPrinting`. For more details about these options, see the “Auxiliary Dictionary Keys” in *CGPDFContext Reference*, part of the Quartz 2D Reference.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [dataRepresentation](#) (page 122)
- [writeToURL:](#) (page 135)
- [writeToURL:withOptions:](#) (page 136)
- [writeToFile:](#) (page 134)

Declared In

PDFDocument.h

writeToURL:

Writes the document to a location specified by the passed-in URL.

```
- (BOOL)writeToURL:(NSURL *)url
```

Availability

Available in Mac OS X v10.4 and later.

See Also

- [dataRepresentation](#) (page 122)
- [writeToFile:](#) (page 134)
- [writeToFile:withOptions:](#) (page 135)
- [writeToURL:withOptions:](#) (page 136)

Declared In

PDFDocument.h

writeToURL:withOptions:

Writes the document to the specified URL with the specified options.

- (BOOL)writeToURL:(NSURL *)url withOptions:(NSDictionary *)options

Availability

Available in Mac OS X v10.4 and later.

See Also

- [dataRepresentation](#) (page 122)
- [writeToURL:](#) (page 135)
- [writeToFile:](#) (page 134)
- [writeToFile:withOptions:](#) (page 135)

Declared In

PDFDocument.h

Delegate Methods

didMatchString:

Called for every match found during a find operation.

- (void)didMatchString:(PDFSelection *)instance

Availability

Available in Mac OS X v10.4 and later.

See Also

- [findString:withOptions:](#) (page 125)
- [setDelegate:](#) (page 132)

Declared In

PDFDocument.h

documentDidBeginDocumentFind:

Called when the PDFDocumentDidBeginFindNotification notification is posted.

- (void)documentDidBeginDocumentFind:(NSNotification *)notification

Availability

Available in Mac OS X v10.4 and later.

See Also

- PDFDocumentDidBeginFindNotification
- [setDelegate:](#) (page 132)

Declared In

PDFDocument.h

documentDidBeginPageFind:Called when the `PDFDocumentDidBeginPageFindNotification` notification is posted.

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

Availability

Available in Mac OS X v10.4 and later.

See Also`PDFDocumentDidBeginPageFindNotification`

- [setDelegate:](#) (page 132)

Declared In

PDFDocument.h

documentDidEndDocumentFind:Called when the `PDFDocumentDidEndFindNotification` notification is posted.

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

Availability

Available in Mac OS X v10.4 and later.

See Also`PDFDocumentDidEndFindNotification`

- [setDelegate:](#) (page 132)

Declared In

PDFDocument.h

documentDidEndPageFind:Called when the `PDFDocumentDidEndPageFindNotification` notification is posted.

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

Availability

Available in Mac OS X v10.4 and later.

See Also`PDFDocumentDidEndPageFindNotification`

- [setDelegate:](#) (page 132)

Declared In

PDFDocument.h

documentDidFindMatch:

Called when the `PDFDocumentDidFindMatchNotification` notification is posted.

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

Availability

Available in Mac OS X v10.4 and later.

See Also

`PDFDocumentDidFindMatchNotification`

- [setDelegate:](#) (page 132)

Declared In

`PDFDocument.h`

documentDidUnlock:

Called when the `PDFDocumentDidUnlockNotification` notification is posted.

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

Availability

Available in Mac OS X v10.4 and later.

See Also

`PDFDocumentDidUnlockNotification`

- [setDelegate:](#) (page 132)

Declared In

`PDFDocument.h`

Constants

PDFPrintScalingMode

The type of scaling to be used when printing a page (see “PDF Page Scaling Modes for Printing”).

```
typedef NSInteger PDFPrintScalingMode;
```

Availability

Available in Mac OS X v10.5 and later.

Declared In

`PDFDocument.h`

Document Attribute Keys

Keys for the document attributes dictionary. See [documentAttributes](#) (page 123) and [setDocumentAttributes:](#) (page 133).

```
extern NSString *PDFDocumentTitleAttribute;
extern NSString *PDFDocumentAuthorAttribute;
extern NSString *PDFDocumentSubjectAttribute;
extern NSString *PDFDocumentCreatorAttribute;
extern NSString *PDFDocumentProducerAttribute;
extern NSString *PDFDocumentCreationDateAttribute;
extern NSString *PDFDocumentModificationDateAttribute;
extern NSString *PDFDocumentKeywordsAttribute;
```

Constants

PDFDocumentTitleAttribute

An optional text string (an NSString) containing the title of the document.

Available in Mac OS X v10.4 and later.

Declared in PDFDocument.h.

PDFDocumentAuthorAttribute

An optional text string (an NSString) containing the name of the author of the document.

Available in Mac OS X v10.4 and later.

Declared in PDFDocument.h.

PDFDocumentSubjectAttribute

An optional text string (an NSString) containing a description of the subject of the document.

Available in Mac OS X v10.4 and later.

Declared in PDFDocument.h.

PDFDocumentCreatorAttribute

An optional text string (an NSString) containing the name of the application that created the document content.

Available in Mac OS X v10.4 and later.

Declared in PDFDocument.h.

PDFDocumentProducerAttribute

An optional text string (an NSString) containing the name of the application that produced the PDF data for the document.

Available in Mac OS X v10.4 and later.

Declared in PDFDocument.h.

PDFDocumentCreationDateAttribute

An optional text string (an NSDate) containing the document's creation date.

Available in Mac OS X v10.4 and later.

Declared in PDFDocument.h.

PDFDocumentModificationDateAttribute

An optional text string (an NSDate) containing the document's last-modified date.

Available in Mac OS X v10.4 and later.

Declared in PDFDocument.h.

PDFDocumentKeywordsAttribute

An optional array of text strings (an NSArray of NSString objects) containing keywords for the document.

Available in Mac OS X v10.4 and later.

Declared in PDFDocument.h.

Declared In

PDFDocument.h

PDF Page Scaling Modes for Printing

Modes that specify how the page should be scaled when printing. See the `PDFView` method `printWithInfo:autoRotate:pageScaling:` (page 205).

```
enum { kPDFPrintPageScaleNone = 0,          kPDFPrintPageScaleToFit = 1,
      kPDFPrintPageScaleDownToFit = 2 };
```

Constants

kPDFPrintPageScaleNone

Do not apply scaling to the page when printing.

Available in Mac OS X v10.5 and later.

Declared in PDFDocument.h.

kPDFPrintPageScaleToFit

Scale each page up or down to best fit the paper size.

Available in Mac OS X v10.5 and later.

Declared in PDFDocument.h.

kPDFPrintPageScaleDownToFit

Scale large pages down to fit the paper size (smaller pages do not get scaled up).

Available in Mac OS X v10.5 and later.

Declared in PDFDocument.h.

Declared In

PDFDocument.h

Notifications

PDFDocument declares and posts the following notifications:

PDFDocumentDidUnlockNotification

Posted when a document unlocks after a `unlockWithPassword:` (page 134) message.

The notification object is the PDFDocument object itself.

Availability

Available in Mac OS X v10.4 and later.

Declared In

PDFDocument.h

PDFDocumentDidBeginFindNotification

Posted when the `beginFindString:withOptions:` (page 121) or `findString:withOptions:` (page 125) method begins finding.

The notification object is the `PDFDocument` object itself.

Availability

Available in Mac OS X v10.4 and later.

Declared In

`PDFDocument.h`

PDFDocumentDidEndFindNotification

Posted when the `beginFindString:withOptions:` (page 121) or `findString:withOptions:` (page 125) method returns.

The `beginFindString:withOptions:` (page 121) method returns immediately, so this notification is posted when the “find” operation is finished.

You can use this notification to know when to close or hide a progress bar.

The notification object is the `PDFDocument` object itself.

Availability

Available in Mac OS X v10.4 and later.

Declared In

`PDFDocument.h`

PDFDocumentDidBeginPageFindNotification

Posted each time a find operation begins working on a new page of a document.

You can use this notification to update a progress bar.

The notification object is the `PDFDocument` object itself. To determine the page, use the `@\"PDFDocumentPageIndex\"` key to obtain userinfo of type `NSNumber`.

Availability

Available in Mac OS X v10.4 and later.

Declared In

`PDFDocument.h`

PDFDocumentDidEndPageFindNotification

Posted each time a find operation finishes working on a page in a document.

You can use this notification to update a progress bar.

The notification object is the `PDFDocument` object itself. To determine the page, use the `@\"PDFDocumentPageIndex\"` key to obtain userinfo of type `NSNumber`.

Availability

Available in Mac OS X v10.4 and later.

Declared In

`PDFDocument.h`

PDFDocumentDidFindMatchNotification

Posted each time a string match is found in a document.

The notification object is the `PDFDocument` object itself. To determine the string selection found, use the @"PDFDocumentFoundSelection" key to obtain userinfo of type `PDFSelection` *

Availability

Available in Mac OS X v10.4 and later.

Declared In

`PDFDocument.h`

PDFDocumentDidBeginWriteNotification

Posted each time a write operation begins working on a document.

The notification object is the `PDFDocument` object itself.

Availability

Available in Mac OS X v10.4 and later.

Declared In

`PDFDocument.h`

PDFDocumentDidEndWriteNotification

Posted each time a write operation finishes working on a document.

The notification object is the `PDFDocument` object itself.

Availability

Available in Mac OS X v10.4 and later.

Declared In

`PDFDocument.h`

PDFDocumentDidBeginPageWriteNotification

Posted each time a write operation begins working on a page in a document.

The notification object is the `PDFDocument` object itself. To determine the page, use the @"PDFDocumentPageIndex" key to obtain userinfo of type `NSNumber`.

Availability

Available in Mac OS X v10.4 and later.

Declared In

`PDFDocument.h`

PDFDocumentDidEndPageWriteNotification

Posted each time a write operation finishes working on a page in a document.

The notification object is the `PDFDocument` object itself. To determine the page, use the `@PDFDocumentPageIndex` key to obtain userinfo of type `NSNumber`.

Availability

Available in Mac OS X v10.4 and later.

Declared In

`PDFDocument.h`

PDFOutline Class Reference

Inherits from	NSObject
Conforms to	NSObject (NSObject)
Framework	Library/Frameworks/Quartz.framework/Frameworks/PDFKit.framework
Declared in	PDFKit/PDFOutline.h
Availability	Available in Mac OS X v10.4 and later.

Overview

A `PDFOutline` object is an element in a tree-structured hierarchy that can represent the structure of a PDF document.

An outline is an optional component of a PDF document, useful for viewing the structure of the document and for navigating within it.

Outlines are created by the document's author. If you represent a PDF document outline using outline objects, the root of the hierarchy is obtained from the PDF document itself. This root outline is not visible and serves merely as a container for the visible outlines.

Tasks

Initializing an Outline

- `init` (page 149)
Initializes a `PDFOutline` object.
- `initWithDocument:` (page 149)
Initializes an outline with the specified PDF document. (**Deprecated**. Use the `PDFDocument` `outlineRoot` (page 129) method instead.)

Getting Information About an Outline

- `document` (page 148)
Returns the document with which the outline is associated.

- `numberOfChildren` (page 150)
Returns the number of child outline objects in the outline.
- `parent` (page 151)
Returns the parent outline object of the outline (returns `NULL` if called on the root outline object).
- `childAtIndex:` (page 147)
Returns the child outline object at the specified index.
- `index` (page 148)
Returns the index of the outline.

Managing Outline Labels

- `label` (page 150)
Returns the label for the outline.
- `setLabel:` (page 152)
Sets the label for the outline (has no effect on the root outline object).

Managing Actions and Destinations

- `destination` (page 148)
Returns the destination associated with the outline.
- `action` (page 147)
Returns the action performed when users click the outline.
- `setAction:` (page 151)
Sets the action associated with the outline.
- `setDestination:` (page 152)
Sets the destination associated with the outline.

Changing an Outline Hierarchy

- `insertChild:atIndex:` (page 149)
Inserts the specified outline object at the specified index.
- `removeFromParent` (page 151)
Removes the outline object from its parent (does nothing if outline object is the root outline object).

Managing the Disclosure of an Outline Object

- `isOpen` (page 150)
Returns a Boolean value that indicates whether the outline object is initially disclosed.
- `setIsOpen:` (page 152)
Sets the initial disclosure state of the outline object.

Instance Methods

action

Returns the action performed when users click the outline.

- (PDFAction *)action

Discussion

The root outline serves only as a container for the outlines it owns; it does not have an action. Note that a PDFOutline object can have either an action or a destination, not both.

If the PDFOutline object has a destination, instead of an action, `action` returns a PDFActionGoTo object (this is equivalent to calling `destination` (page 148) on the PDFOutline object). For other action types, `action` returns the appropriate PDF Kit action type object, such as PDFActionURL.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [setAction:](#) (page 151)

Declared In

PDFOutline.h

childAtIndex:

Returns the child outline object at the specified index.

- (PDFOutline *)childAtIndex:(NSUInteger)index

Discussion

The index is zero-based. This method throws an exception if `index` is out of range.

Important: In Mac OS X v10.5 and later, a PDFOutline object retains all its children, so `childAtIndex:` returns the same retained child outline object every time it's called. This means that you do not need to retain the object returned by `childAtIndex:`. This differs from the behavior of PDFOutline in Mac OS X v10.4. In Tiger, `childAtIndex:` returns an auto-released, one-off child outline object, when meant that you had to include code to retain the child.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [index](#) (page 148)

Declared In

PDFOutline.h

destination

Returns the destination associated with the outline.

- (`PDFDestination *`)`destination`

Discussion

The root outline serves only as a container for the outlines it owns; it does not have a destination. Note that a `PDFOutline` object can have either a destination or an action, not both.

This method may return `NULL` if the outline has an associated action instead of a destination. Note that if the associated action is a `PDFActionGoTo`, this method returns the destination from the `PDFActionGoTo` object. However, it is better to use the `action` (page 147) method for this purpose.

Availability

Available in Mac OS X v10.4 and later.

See Also

- `setDestination`: (page 152)

Declared In

`PDFOutline.h`

document

Returns the document with which the outline is associated.

- (`PDFDocument *`)`document`

Availability

Available in Mac OS X v10.4 and later.

Declared In

`PDFOutline.h`

index

Returns the index of the outline.

- (`NSInteger`)`index`

Discussion

The index of the outline object is relative to its siblings and from the perspective of the parent of the outline object. The root outline object, and any outline object without a parent, has an index value of 0.

Availability

Available in Mac OS X v10.5 and later.

Declared In

`PDFOutline.h`

init

Initializes a `PDFOutline` object.

```
- (id)init
```

Discussion

If you want the `PDFOutline` object returned by this method to be the outline root, you must add additional `PDFOutline` objects to create the outline hierarchy you desire. Then, you must add the root outline object to your PDF document by passing it to the `PDFDocument` [setOutlineRoot:](#) (page 133) method.

If you want the `PDFOutline` object returned by this method to be a child of an existing outline, you must use [setLabel:](#) (page 152) to give it a label and give it either a destination or action using [setDestination:](#) (page 152) or [setAction:](#) (page 151), respectively. In addition, you must add this outline object to the existing `PDFOutline` object as a new child, using [insertChildAtIndex:](#) (page 149)

Availability

Available in Mac OS X v10.4 and later.

Declared In

`PDFOutline.h`

initWithDocument:

Initializes an outline with the specified PDF document. (**Deprecated.** Use the `PDFDocument` [outlineRoot](#) (page 129) method instead.)

```
- (id)initWithDocument:(PDFDocument *)document
```

Discussion

Returns `NULL` if the document does not contain an outline. Invoking this method is equivalent to sending the `outlineRoot` message to a `PDFDocument` object.

Availability

Available in Mac OS X v10.4 and later.

Declared In

`PDFOutline.h`

insertChildAtIndex:

Inserts the specified outline object at the specified index.

```
- (void)insertChild:(PDFOutline *)child atIndex:(NSUInteger)index
```

Discussion

To build a PDF outline hierarchy, use this method to add child outline objects. Before you call this method on a `PDFOutline` object that already has a parent, you should retain the object and call [removeFromParent](#) (page 151) on it first.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [childAtIndex:](#) (page 147)

Declared In

PDFOutline.h

isOpen

Returns a Boolean value that indicates whether the outline object is initially disclosed.

- (BOOL)isOpen

Discussion

Calling `isOpen` on an outline object that has no children always returns `NO`. Calling `isOpen` on the root outline object always returns `YES`.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [setIsOpen:](#) (page 152)

Declared In

PDFOutline.h

label

Returns the label for the outline.

- (NSString *)label

Discussion

The root outline serves only as a container for the outlines it owns; it does not have a label.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setLabel:](#) (page 152)

Declared In

PDFOutline.h

numberOfChildren

Returns the number of child outline objects in the outline.

- (NSUInteger)numberOfChildren

Availability

Available in Mac OS X v10.4 and later.

See Also

- [childAtIndex:](#) (page 147)

Declared In

PDFOutline.h

parent

Returns the parent outline object of the outline (returns `NULL` if called on the root outline object).

- (PDFOutline *)parent

Availability

Available in Mac OS X v10.5 and later

Declared In

PDFOutline.h

removeFromParent

Removes the outline object from its parent (does nothing if outline object is the root outline object).

- (void)removeFromParent

Availability

Available in Mac OS X v10.5 and later.

See Also

- [parent](#) (page 151)

Declared In

PDFOutline.h

setAction:

Sets the action associated with the outline.

- (void)setAction:(PDFAction *)action

Discussion

Calling `setAction` on the root outline object has no effect, because the root outline does not have an action or a destination..

Availability

Available in Mac OS X v10.5 and later.

See Also

- [action](#) (page 147)

Declared In

PDFOutline.h

setDestination:

Sets the destination associated with the outline.

- (void)setDestination:(PDFDestination *)*destination*

Discussion

Calling `setDestination` on the root outline object has no effect, because the root outline does not have an action or a destination.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [destination](#) (page 148)

Declared In

PDFOutline.h

setIsOpen:

Sets the initial disclosure state of the outline object.

- (void)setIsOpen:(BOOL)*open*

Discussion

Calling `setIsOpen` on an outline object with no children or on the root outline object has no effect.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [isOpen](#) (page 150)

Declared In

PDFOutline.h

setLabel:

Sets the label for the outline (has no effect on the root outline object).

- (void)setLabel:(NSString *)*label*

Availability

Available in Mac OS X v10.5 and later.

See Also

- [label](#) (page 150)

Declared In

PDFOutline.h

PDFPage Class Reference

Inherits from	NSObject
Conforms to	NSCopying NSObject (NSObject)
Framework	Library/Frameworks/Quartz.framework/Frameworks/PDFKit.framework
Declared in	PDFKit/PDFPage.h
Availability	Available in Mac OS X v10.4 and later.

Overview

PDFPage, a subclass of NSObject, defines methods used to render PDF pages and work with annotations, text, and selections.

PDFPage objects are flexible and powerful. With them you can render PDF content onscreen or to a printer, add annotations, count characters, define selections, and get the textual content of a page as an NSString object.

Your application instantiates a PDFPage object by asking for one from a PDFDocument object.

For simple display and navigation of PDF documents within your application, you don't need to use PDFPage. You need only use PDFView.

Tasks

Initializing a Page

- initWithDocument: (page 159)
Initializer for subclasses of PDFPage. (**Deprecated**. Use [PDFPage init] or initWithImage: instead.)
- initWithImage: (page 160)
Creates a new PDFPage object and initializes it with the specified NSImage object.

Getting Information About a Page

- document (page 159)
Returns the PDFDocument object with which the page is associated.

- `label` (page 160)
Returns the label for the page.
- `boundsForBox:` (page 157)
Returns the bounds for the specified PDF display box.
- `setBounds:forBox:` (page 164)
Sets the bounds for the specified box.
- `rotation` (page 161)
Returns the page rotation angle in degrees.
- `setRotation:` (page 164)
Sets the rotation angle for the page in degrees.

Working with Annotations

- `annotations` (page 156)
Returns an array containing the page's annotations.
- `displaysAnnotations` (page 158)
Returns a Boolean value indicating whether annotations are displayed for the page.
- `setDisplayAnnotations:` (page 164)
Specifies whether or not to display annotations for the page.
- `addAnnotation:` (page 155)
Adds the specified annotation object to the page.
- `removeAnnotation:` (page 161)
Removes the specified annotation from the page.
- `annotationAtPoint:` (page 156)
Returns the annotation, if there is one, at the specified point.

Rendering Pages

- `drawWithBox:` (page 159)
Draws the page within the specified box.
- `transformContextForBox:` (page 165)
Transforms the current context, given the specified box.

Working with Textual Content

- `numberOfCharacters` (page 160)
Returns the number of characters on the page, including whitespace characters.
- `string` (page 165)
Returns an `NSString` object representing the text on the page.
- `attributedString` (page 156)
Returns an `NSAttributedString` object representing the text on the page.
- `characterBoundsAtIndex:` (page 157)
Returns the bounds, in page space, of the character at the specified index.

- [characterIndexAtPoint:](#) (page 158)
Returns the character index value for the specified point in page space.

Working with Selections

- [selectionForRect:](#) (page 162)
Returns the text enclosed within the specified rectangle, expressed in page (user) coordinates.
- [selectionForWordAtPoint:](#) (page 163)
Returns the whole word that includes the specified point.
- [selectionForLineAtPoint:](#) (page 161)
Returns the whole line of text that includes the specified point.
- [selectionFromPoint:toPoint:](#) (page 163)
Returns the text between the two specified points in page space.
- [selectionForRange:](#) (page 162)
Returns the text contained within the specified range.

Miscellaneous

- [dataRepresentation](#) (page 158)
Returns the PDF data (that is, a PDF document) representing this page. This method does not preserve external page links.

Instance Methods

addAnnotation:

Adds the specified annotation object to the page.

- (void)addAnnotation:(PDFAnnotation *)*annotation*

Availability

Available in Mac OS X v10.4 and later.

See Also

- [annotations](#) (page 156)
- [displaysAnnotations](#) (page 158)
- [setDisplayAnnotations:](#) (page 164)
- [removeAnnotation:](#) (page 161)
- [annotationAtPoint:](#) (page 156)

Declared In

PDFPage.h

annotationAtPoint:

Returns the annotation, if there is one, at the specified point.

- (PDFAnnotation *)annotationAtPoint:(NSPoint)point

Discussion

Use this method for hit-testing based on the current cursor position. If more than one annotation shares the specified point, the frontmost (or topmost) one is returned (the annotations are searched in reverse order of their appearance in the PDF data file). Returns `NULL` if there is no annotation at *point*.

Specify the point in page space. Page space is a 72 dpi coordinate system with the origin at the lower-left corner of the current page.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [annotations](#) (page 156)
- [displaysAnnotations](#) (page 158)
- [setDisplayAnnotations:](#) (page 164)
- [addAnnotation:](#) (page 155)
- [removeAnnotation:](#) (page 161)

Declared In

PDFPage.h

annotations

Returns an array containing the page's annotations.

- (NSArray *)annotations

Discussion

The elements of the array will most likely be typed to subclasses of the `PDFAnnotation` class.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [displaysAnnotations](#) (page 158)
- [setDisplayAnnotations:](#) (page 164)
- [addAnnotation:](#) (page 155)
- [removeAnnotation:](#) (page 161)
- [annotationAtPoint:](#) (page 156)

Declared In

PDFPage.h

attributedString

Returns an `NSAttributedString` object representing the text on the page.

- (NSAttributedString *)attributedString

Availability

Available in Mac OS X v10.4 and later.

See Also

- [numberOfCharacters](#) (page 160)
- [string](#) (page 165)

Declared In

PDFPage.h

boundsForBox:

Returns the bounds for the specified PDF display box.

- (NSRect)boundsForBox:(PDFDisplayBox)box

Discussion

The `PDFDisplayBox` enumeration defines the various box types (see “[Constants](#)” (page 166) for additional information about box types).

Note that only the media box is required for a PDF. If you request the bounds for the crop box, but the PDF does not include a crop box, the bounds for the media box are returned instead. If you request the bounds for other box types, and the PDF does not include these types, the bounds for the crop box are returned instead.

The coordinates for the box are in page space, so you might need to transform the points if the page has a rotation on it. Also, note that the bounds `boundsForBox` returns are intersected with the page’s media box.

`boundsForBox` throws a range exception if `box` is not in range.

Availability

Available in Mac OS X v10.4 and later.

See Also

- `setBoundsForBox:`

Declared In

PDFPage.h

characterBoundsAtIndex:

Returns the bounds, in page space, of the character at the specified index.

- (NSRect)characterBoundsAtIndex:(NSInteger)index

Discussion

In the unlikely event that there is more than one character at the specified index point, only the bounds of the first character is returned.

Page space is a 72 dpi coordinate system with the origin at the lower-left corner of the current page. Note that the bounds returned are not guaranteed to have integer coordinates.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [characterIndexAtPoint:](#) (page 158)

Declared In

PDFPage.h

characterIndexAtPoint:

Returns the character index value for the specified point in page space.

- (NSInteger)characterIndexAtPoint:(NSPoint)*point*

Discussion

If there is no character at the specified point, the method returns -1.

Page space is a 72 dpi coordinate system with the origin at the lower-left corner of the current page.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [characterBoundsAtIndex:](#) (page 157)

Declared In

PDFPage.h

dataRepresentation

Returns the PDF data (that is, a PDF document) representing this page. This method does not preserve external page links.

- (NSData *)dataRepresentation

Availability

Available in Mac OS X v10.4 and later.

Declared In

PDFPage.h

displaysAnnotations

Returns a Boolean value indicating whether annotations are displayed for the page.

- (BOOL)displaysAnnotations

Discussion

If YES, the page will draw annotations when a drawing method is called.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [annotations](#) (page 156)
- [setDisplayAnnotations:](#) (page 164)
- [addAnnotation:](#) (page 155)
- [removeAnnotation:](#) (page 161)
- [annotationAtPoint:](#) (page 156)
- [drawWithBox:](#) (page 159)

Declared In

PDFPage.h

document

Returns the PDFDocument object with which the page is associated.

- (PDFDocument *)document

Availability

Available in Mac OS X v10.4 and later.

Declared In

PDFPage.h

drawWithBox:

Draws the page within the specified box.

- (void)drawWithBox:(PDFDisplayBox)box

Discussion

This method takes into account the page rotation and draws clipped to the specified box. If the page is set to display annotations, this method also draws them. This method does not clear the background. To clear the background before drawing, use `NSRectFill` with `NSColor` set (typically) to white.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [displaysAnnotations](#) (page 158)

Declared In

PDFPage.h

initWithDocument:

Initializer for subclasses of PDFPage. (**Deprecated.** Use `[PDFPage init]` or `initWithImage:` instead.)

- (id)initWithDocument:(PDFDocument *)document

Discussion

Subclasses of `PDFPage` must handle several methods that are transparently handled when using the `PDFPage` class directly, including `boundsForBox` and `drawInRect:withBox:`.

Availability

Available in Mac OS X v10.4 and later.

Declared In

`PDFPage.h`

initWithImage:

Creates a new `PDFPage` object and initializes it with the specified `NSImage` object.

```
- (id)initWithImage:(NSImage *)image
```

Availability

Available in Mac OS X v10.5 and later.

Declared In

`PDFPage.h`

label

Returns the label for the page.

```
- (NSString *)label
```

Discussion

Typically, the label is “1” for the first page, “2” for the second page, and so on, but nonnumerical labels are also possible (such as “xvi”, “4-1” and so on).

Availability

Available in Mac OS X v10.4 and later.

See Also

- [document](#) (page 159)

Declared In

`PDFPage.h`

numberOfCharacters

Returns the number of characters on the page, including whitespace characters.

```
- (NSUInteger)numberOfCharacters
```

Availability

Available in Mac OS X v10.4 and later.

See Also

- [string](#) (page 165)

Declared In

PDFPage.h

removeAnnotation:

Removes the specified annotation from the page.

```
- (void)removeAnnotation:(PDFAnnotation *)annotation
```

Availability

Available in Mac OS X v10.4 and later.

See Also

- [annotations](#) (page 156)
- [displaysAnnotations](#) (page 158)
- [setDisplayAnnotations:](#) (page 164)
- [addAnnotation:](#) (page 155)
- [annotationAtPoint:](#) (page 156)

Declared In

PDFPage.h

rotation

Returns the page rotation angle in degrees.

```
- (int)rotation
```

Discussion

The rotation is a positive multiple of 90: 0, 90, 180, or 270. The rotation of pages with negative rotation is converted to a corresponding positive rotation.

If you are subclassing `PDFView` and displaying pages yourself, don't assume a rotation of 0. Pages with an inherent rotation display rotated when opened unless you set their rotation to zero. Regardless of the inherent rotation angle, it is up to the author of a page whether zero rotation corresponds to upright text when displayed on a monitor.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setRotation:](#) (page 164)

Declared In

PDFPage.h

selectionForLineAtPoint:

Returns the whole line of text that includes the specified point.

```
- (PDFSelection *)selectionForLineAtPoint:(NSPoint)point
```

Discussion

Returns NULL if no line of text contains *point*.

Use this method to respond to a triple-click.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [selectionForRect:](#) (page 162)
- [selectionForWordAtPoint:](#) (page 163)
- [selectionFromPoint:toPoint:](#) (page 163)
- [selectionForRange:](#) (page 162)

Declared In

PDFPage.h

selectionForRange:

Returns the text contained within the specified range.

```
- (PDFSelection *)selectionForRange:(NSRange)range
```

Discussion

This method raises an exception if the range length is 0 or if either end of the range is outside the range of characters on the page.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [selectionForRect:](#) (page 162)
- [selectionForWordAtPoint:](#) (page 163)
- [selectionForLineAtPoint:](#) (page 161)
- [selectionFromPoint:toPoint:](#) (page 163)

Declared In

PDFPage.h

selectionForRect:

Returns the text enclosed within the specified rectangle, expressed in page (user) coordinates.

```
- (PDFSelection *)selectionForRect:(NSRect)rect
```

Availability

Available in Mac OS X v10.4 and later.

See Also

- [selectionForWordAtPoint:](#) (page 163)
- [selectionForLineAtPoint:](#) (page 161)
- [selectionFromPoint:toPoint:](#) (page 163)

- [selectionForRange:](#) (page 162)

Declared In

PDFPage.h

selectionForWordAtPoint:

Returns the whole word that includes the specified point.

```
- (PDFSelection *)selectionForWordAtPoint:(NSPoint)point
```

Discussion

Returns NULL if no word contains *point*.

Use this method to respond to a double-click.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [selectionForRect:](#) (page 162)
- [selectionForLineAtPoint:](#) (page 161)
- [selectionFromPoint:toPoint:](#) (page 163)
- [selectionForRange:](#) (page 162)

Declared In

PDFPage.h

selectionFromPoint:toPoint:

Returns the text between the two specified points in page space.

```
- (PDFSelection *)selectionFromPoint:(NSPoint)startPoint toPoint:(NSPoint)endPoint
```

Discussion

Either point may be the one closer to the start of the page. In determining the selection, the points are sorted first top to bottom and then left to right.

Page space is a 72 dpi coordinate system with the origin at the lower-left corner of the current page.

To visualize the selection, picture the rectangle defined by *startPoint* and *endPoint*. The selection begins at the first character fully within the defined rectangle and closest to its upper-left corner. The selection ends at the last character fully within the defined rectangle and closest to its lower-right corner.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [selectionForRect:](#) (page 162)
- [selectionForWordAtPoint:](#) (page 163)
- [selectionForLineAtPoint:](#) (page 161)
- [selectionForRange:](#) (page 162)

Declared In

PDFPage.h

setBounds:forBox:

Sets the bounds for the specified box.

```
- (void)setBounds:(NSRect)bounds forBox:(PDFDisplayBox)box
```

Discussion

If the box does not exist, this method creates it for you.

To remove a box, pass `NSZeroRect` for the bounds (note that you cannot remove the media box). If the box bounds are not in range, this method throws a range exception.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [boundsForBox:](#) (page 157)

Declared In

PDFPage.h

setDisplayAnnotations:

Specifies whether or not to display annotations for the page.

```
- (void)setDisplaysAnnotations:(BOOL)display
```

Discussion

If *display* is YES, the page will draw annotations when a drawing method is called.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [annotations](#) (page 156)
- [displaysAnnotations](#) (page 158)
- [addAnnotation:](#) (page 155)
- [removeAnnotation:](#) (page 161)
- [annotationAtPoint:](#) (page 156)

Declared In

PDFPage.h

setRotation:

Sets the rotation angle for the page in degrees.

```
- (void)setRotation:(int)angle
```

Discussion

The rotation must be a positive or negative multiple of 90 (negative angles are converted to their positive equivalents; for example, -90 is changed to 270); otherwise this method throws an exception.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [rotation](#) (page 161)

Declared In

PDFPage.h

string

Returns an `NSString` object representing the text on the page.

- (NSString *)string

Availability

Available in Mac OS X v10.4 and later.

See Also

- [numberOfCharacters](#) (page 160)

- [attributedString](#) (page 156)

Declared In

PDFPage.h

transformContextForBox:

Transforms the current context, given the specified box.

- (void)transformContextForBox:(PDFDisplayBox)box

Discussion

When transforming the current context, this method takes into account the rotation of the page, as well as the origin of the box with respect to the page's base coordinate system. This is a convenient method to call within the `PDFView` [drawPage:](#) (page 199) method or from within a draw method of a `PDFAnnotation` subclass.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [drawWithBox:](#)

Declared In

PDFPage.h

Constants

The following box types may be used with `PDFPage` drawing and bounds-setting methods. See the Adobe PDF Specification for more information on box types, units, and coordinate systems.

Constant	Description
<code>kPDFDisplayBox-MediaBox</code>	A rectangle defining the boundaries of the physical medium for display or printing, expressed in default user-space units. Available in Mac OS X v10.4 and later. Declared in <code>PDFPage.h</code> .
<code>kPDFDisplayBoxCropBox</code>	A rectangle defining the boundaries of the visible region, expressed in default user-space units. Default value equal to <code>kPDFDisplayBoxMediaBox</code> . Available in Mac OS X v10.4 and later. Declared in <code>PDFPage.h</code> .
<code>kPDFDisplayBox-BleedBox</code>	A rectangle defining the boundaries of the clip region for the page contents in a production environment. Default value equal to <code>kPDFDisplayBoxCropBox</code> . Available in Mac OS X v10.4 and later. Declared in <code>PDFPage.h</code> .
<code>kPDFDisplayBoxTrimBox</code>	A rectangle defining the intended boundaries of the finished page. Default value equal to <code>kPDFDisplayBoxCropBox</code> . Available in Mac OS X v10.4 and later. Declared in <code>PDFPage.h</code> .
<code>kPDFDisplayBoxArtBox</code>	A rectangle defining the boundaries of the page's meaningful content including surrounding white space intended for display. Default value equal to <code>kPDFDisplayBoxCropBox</code> . Available in Mac OS X v10.4 and later. Declared in <code>PDFPage.h</code> .

PDFSelection Class Reference

Inherits from	NSObject
Conforms to	NSCopying NSObject (NSObject)
Framework	Library/Frameworks/Quartz.framework/Frameworks/PDFKit.framework
Declared in	PDFKit/PDFSelection.h
Availability	Available in Mac OS X v10.4 and later.

Overview

A `PDFSelection` object identifies a contiguous or noncontiguous selection of text in a PDF document.

Tasks

Initializing a Selection

- [initWithDocument:](#) (page 171)
Returns an empty `PDFSelection` object.

Getting Information About a Selection

- [pages](#) (page 172)
Returns the array of pages contained in the selection.
- [string](#) (page 173)
Returns an `NSString` object representing the text contained in the selection (may contain linefeed characters).
- [attributedString](#) (page 169)
Returns an `NSAttributedString` object representing the text contained in the selection (may contain linefeed characters).
- [boundsForPage:](#) (page 169)
Returns the bounds of the selection on the specified page.
- [selectionsByLine](#) (page 172)
Returns an array of selections, one for each line of text covered by the receiver.

- `color` (page 170)
Returns the color used to draw the selection.

Modifying a Selection

- `addSelection:` (page 168)
Adds the specified selection to the receiving selection.
- `addSelections:` (page 169)
Adds the specified array of selections to the receiving selection.
- `extendSelectionAtEnd:` (page 171)
Extends the selection from its end toward the end of the document.
- `extendSelectionAtStart:` (page 171)
Extends the selection from its start toward the beginning of the document.

Managing Selection Drawing

- `drawForPage:active:` (page 170)
Calls `drawForPage:withBox:active:` (page 170) with a default value for box parameter.
- `drawForPage:withBox:active:` (page 170)
Draws the selection relative to the origin of the specified box in page space.
- `setColor:` (page 172)
Sets the color used for the drawing of a selection in both active and inactive states.

Instance Methods

addSelection:

Adds the specified selection to the receiving selection.

```
- (void)addSelection:(PDFSelection *)selection
```

Discussion

Selections do not have to be contiguous. If the selection to be added overlaps with the receiving selection, the overlap is removed in a process called normalization.

Availability

Available in Mac OS X v10.4 and later.

See Also

- `extendSelectionAtEnd:` (page 171)
- `extendSelectionAtStart:` (page 171)

Declared In

PDFSelection.h

addSelections:

Adds the specified array of selections to the receiving selection.

```
- (void)addSelections:(NSArray *)selections
```

Discussion

This method provides better performance than multiple calls to `addSelection` if you need to add several selections to an existing selection. This is because the normalization of the selection (the removal of any overlaps between selections) occurs only once, after all selections have been added.

Availability

Available in Mac OS X v10.5 and later.

See Also

- `addSelection`:

Declared In

`PDFSelection.h`

attributedString

Returns an `NSAttributedString` object representing the text contained in the selection (may contain linefeed characters).

```
- (NSAttributedString *)attributedString
```

Availability

Available in Mac OS X v10.4 and later.

See Also

- [string](#) (page 173)

Declared In

`PDFSelection.h`

boundsForPage:

Returns the bounds of the selection on the specified page.

```
- (NSRect)boundsForPage:(PDFPage *)page
```

Discussion

The selection rectangle is given in page space.

Page space is a 72 dpi coordinate system with the origin at the lower-left corner of the current page.

Availability

Available in Mac OS X v10.4 and later.

Declared In

`PDFSelection.h`

color

Returns the color used to draw the selection.

- (NSColor *)color

Discussion

Note that when no color has been specified for the `PDFSelection` objects in a document, the selections are drawn using `[NSColor selectedTextBackgroundColor]` for the active state and `[NSColor secondarySelectedControlColor]` for the inactive state.

Availability

Available in Mac OS X v10.5 and later.

See Also

- `setColor:`

Declared In

`PDFSelection.h`

drawForPage:active:

Calls `drawForPage:withBox:active:` (page 170) with a default value for box parameter.

- (void)drawForPage:(PDFPage *)page active:(BOOL)active

Discussion

The default value is `kPDFDisplayBoxCropBox`. If `active` is YES, drawing uses `selectedTextBackgroundColor`. If NO, it uses `secondarySelectedControlColor`.

Availability

Available in Mac OS X v10.4 and later.

See Also

- `drawForPage:withBox:active:` (page 170)

Declared In

`PDFSelection.h`

drawForPage:withBox:active:

Draws the selection relative to the origin of the specified box in page space.

- (void)drawForPage:(PDFPage *)page withBox:(PDFDisplayBox)box active:(BOOL)active

Discussion

The selection is drawn using the current highlight color. If `active` is YES, drawing uses `selectedTextBackgroundColor`. If NO, it uses `secondarySelectedControlColor`. Refer to the `PDFPage` class for the list of available box types.

Page space is a 72 dpi coordinate system with the origin at the lower-left corner of the current page.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [drawForPage:active:](#) (page 170)

Declared In

PDFSelection.h

extendSelectionAtEnd:

Extends the selection from its end toward the end of the document.

- (void)extendSelectionAtEnd:(NSInteger)chars

Discussion

The selection may be extended by any amount, up to and including the end of the document.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [extendSelectionAtStart:](#) (page 171)

- [addSelection:](#) (page 168)

Declared In

PDFSelection.h

extendSelectionAtStart:

Extends the selection from its start toward the beginning of the document.

- (void)extendSelectionAtStart:(NSInteger)chars

Discussion

The selection may be extended by any amount, up to and including the beginning of the document.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [extendSelectionAtEnd:](#) (page 171)

- [addSelection:](#) (page 168)

Declared In

PDFSelection.h

initWithDocument:

Returns an empty PDFSelection object.

- (id)initWithDocument:(PDFDocument *)document

Discussion

Typically, you don't need to create a `PDFSelection` object, but you can use an empty `PDFSelection` object as a container into which you can place selections, using `addSelection:` and `addSelections`.

Availability

Available in Mac OS X v10.5 and later.

Declared In

`PDFSelection.h`

pages

Returns the array of pages contained in the selection.

- (NSArray *)pages

Discussion

Pages are sorted by index number.

Availability

Available in Mac OS X v10.4 and later.

Declared In

`PDFSelection.h`

selectionsByLine

Returns an array of selections, one for each line of text covered by the receiver.

- (NSArray *)selectionsByLine

Discussion

If you call this method on a `PDFSelection` object that represents a paragraph, for example, `selectionsByLine` returns an array that contains one `PDFSelection` object for each line of text in the paragraph.

Availability

Available in Mac OS X v10.5 and later.

Declared In

`PDFSelection.h`

setColor:

Sets the color used for the drawing of a selection in both active and inactive states.

- (void)setColor:(NSColor *)color

Discussion

When no color has been specified for the `PDFSelection` objects in a document, the selections are drawn using `[NSColor selectedTextBackgroundColor]` for the active state and `[NSColor secondarySelectedControlColor]` for the inactive state. Use the `setColor` method to supply a color you want to be used for the drawing of both active and inactive selections.

Availability

Available in Mac OS X v10.5 and later.

See Also

- `color`

Declared In

`PDFSelection.h`

string

Returns an `NSString` object representing the text contained in the selection (may contain linefeed characters).

- (`NSString *`)string

Availability

Available in Mac OS X v10.4 and later.

See Also

- `attributedString`

Declared In

`PDFSelection.h`

PDFThumbnailView Class Reference

Inherits from	NSView : NSResponder : NSObject
Conforms to	NSCoding NSAnimatablePropertyContainer (NSView) NSCoding (NSResponder) NSObject (NSObject)
Framework	Library/Frameworks/Quartz.framework/Frameworks/PDFKit.framework
Declared in	PDFKit/PDFThumbnailView.h
Availability	Available in Mac OS X v10.5 and later.

Overview

A `PDFThumbnailView` object contains a set of thumbnails, each of which represents a page in a PDF document.

Tasks

Accessing the Associated PDF View

- [PDFView](#) (page 178)
Returns the `PDFView` object associated with the thumbnail view.
- [setPDFView:](#) (page 181)
Associates the specified `PDFView` object with the thumbnail view.

Managing the Size of a Thumbnail View

- [thumbnailSize](#) (page 182)
Returns the maximum width and height of the thumbnails in the thumbnail view.
- [setThumbnailSize:](#) (page 181)
Sets the maximum width and height of the thumbnails in the thumbnail view.

Working with Thumbnail View Display Characteristics

- `maximumNumberOfColumns` (page 178)
Returns the maximum number of columns of thumbnails the thumbnail view can display.
- `setMaximumNumberOfColumns:` (page 181)
Sets the maximum number of columns of thumbnails the thumbnail view can display.
- `labelFont` (page 178)
Returns the font used to label the thumbnails.
- `setLabelFont:` (page 180)
Sets the font used to label the thumbnails.
- `backgroundColor` (page 177)
Returns the color used in the background of the thumbnail view.
- `setBackgroundColor:` (page 180)
Sets the color used in the background of the thumbnail view.

Managing the Behavior of a Thumbnail View

- `allowsDragging` (page 176)
Returns a Boolean value indicating whether users can drag thumbnails (that is, re-order pages in the document) within the thumbnail view.
- `setAllowsDragging:` (page 179)
Sets whether users can drag thumbnails within the thumbnail view; that is, re-order pages in the document.
- `allowsMultipleSelection` (page 177)
Returns a Boolean value indicating whether users can select multiple thumbnails in the thumbnail view at one time.
- `setAllowsMultipleSelection:` (page 179)
Sets whether the thumbnail view allows users to select more than one thumbnail at a time.
- `selectedPages` (page 179)
Returns an array of PDF pages that correspond to the selected thumbnails in the thumbnail view.

Instance Methods

`allowsDragging`

Returns a Boolean value indicating whether users can drag thumbnails (that is, re-order pages in the document) within the thumbnail view.

- (BOOL)allowsDragging

Return Value

YES if users can re-order pages by dragging thumbnails, NO otherwise.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [setAllowsDragging](#): (page 179)

Declared In

PDFThumbnailView.h

allowsMultipleSelection

Returns a Boolean value indicating whether users can select multiple thumbnails in the thumbnail view at one time.

- (BOOL)allowsMultipleSelection

Return Value

YES if users can select multiple thumbnails simultaneously, NO otherwise.

Discussion

By default, `PDFThumbnailView` allows only a single thumbnail to be selected at one time. When this is the case, you can get the PDF page that corresponds to the selected thumbnail using the `PDFView` method [currentPage](#) (page 196).

When multiple selections are enabled, however, you must use [selectedPages](#) (page 179) to get the pages that correspond to the set of selected thumbnails.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [setAllowsMultipleSelection](#): (page 179)

- [selectedPages](#) (page 179)

Declared In

PDFThumbnailView.h

backgroundColor

Returns the color used in the background of the thumbnail view.

- (NSColor *)backgroundColor

Return Value

The color of the background in the thumbnail view.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [setBackground-color](#): (page 180)

Declared In

PDFThumbnailView.h

labelFont

Returns the font used to label the thumbnails.

- (NSFont *)labelFont

Return Value

The font used in the thumbnail labels.

Discussion

Typically, the label of a thumbnail is the page number of the page it represents.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [setLabelFont:](#) (page 180)

Declared In

PDFThumbnailView.h

maximumNumberOfColumns

Returns the maximum number of columns of thumbnails the thumbnail view can display.

- (NSInteger)maximumNumberOfColumns

Return Value

The maximum number of columns of thumbnails the thumbnail view can display. If 0, the thumbnail displays as many columns of thumbnails as fit in its size.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [setThumbnailSize:](#) (page 181)

Declared In

PDFThumbnailView.h

PDFView

Returns the PDFView object associated with the thumbnail view.

- (PDFView *)PDFView

Return Value

The PDF view object associated with the thumbnail view.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [setPDFView:](#) (page 181)

Declared In

PDFThumbnailView.h

selectedPages

Returns an array of PDF pages that correspond to the selected thumbnails in the thumbnail view.

```
- (NSArray *)selectedPages
```

Return Value

An array of PDF pages that correspond to the thumbnails selected in the thumbnail view.

Discussion

If the thumbnail view allows multiple selections (if [allowsMultipleSelection](#) (page 177) returns YES), you can use this method to get the PDF pages that correspond to the selected thumbnails.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [allowsMultipleSelection](#) (page 177)
- [setAllowsDragging:](#) (page 179)

Declared In

PDFThumbnailView.h

setAllowsDragging:

Sets whether users can drag thumbnails within the thumbnail view; that is, re-order pages in the document.

```
- (void)setAllowsDragging:(BOOL)allow
```

Parameters

allow

Pass YES to allow users to drag thumbnails in the thumbnail view (this allows them to re-order pages in the document), or NO to disallow.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [allowsDragging](#) (page 176)

Declared In

PDFThumbnailView.h

setAllowsMultipleSelection:

Sets whether the thumbnail view allows users to select more than one thumbnail at a time.

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

Parameters*flag*

Pass YES to allow users to select multiple thumbnails at one time, or NO to disallow.

Discussion

By default, `PDFThumbnailView` allows only a single thumbnail to be selected at one time. When this is the case, you can get the PDF page that corresponds to the selected thumbnail using the `PDFView` method `currentPage` (page 196).

If you use `setAllowsMultipleSelection` to enable multiple selections, however, you must use `selectedPages` (page 179) to get the pages that correspond to the set of selected thumbnails.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [allowsMultipleSelection](#) (page 177)
- [selectedPages](#) (page 179)

Declared In

`PDFThumbnailView.h`

setBackground-color:

Sets the color used in the background of the thumbnail view.

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

Parameters*color*

The color to be used in the background of the thumbnail view.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [backgroundColor](#) (page 177)

Declared In

`PDFThumbnailView.h`

setLabelFont:

Sets the font used to label the thumbnails.

```
- (void)setLabelFont:(NSFont *)font
```

Parameters*font*

The font to be used in the thumbnail labels.

Discussion

Typically, the label of a thumbnail is the page number of the page it represents.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [LabelFont](#) (page 178)

Declared In

PDFThumbnailView.h

setMaximumNumberOfColumns:

Sets the maximum number of columns of thumbnails the thumbnail view can display.

```
- (void)setMaximumNumberOfColumns:(NSInteger)maxColumns
```

Parameters

maxColumns

The maximum number of columns of thumbnails the thumbnail view can display. Pass 0 to make the thumbnail view display as many columns as fit in its size.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [thumbnailSize](#) (page 182)

Declared In

PDFThumbnailView.h

setPDFView:

Associates the specified `PDFView` object with the thumbnail view.

```
- (void)setPDFView:(PDFView *)view
```

Parameters

view

The PDF view object to associate with the thumbnail view.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [PDFView](#) (page 178)

Declared In

PDFThumbnailView.h

setThumbnailSize:

Sets the maximum width and height of the thumbnails in the thumbnail view.

- (void)setThumbnailSize:(NSSize) *size*

Parameters

size

The maximum width and height the thumbnails in the thumbnail view should be.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [thumbnailSize](#) (page 182)

Declared In

PDFThumbnailView.h

thumbnailSize

Returns the maximum width and height of the thumbnails in the thumbnail view.

- (NSSize)thumbnailSize

Return Value

The maximum width and height of the thumbnails in the thumbnail view.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [setThumbnailSize:](#) (page 181)

Declared In

PDFThumbnailView.h

PDFView Class Reference

Inherits from	NSView : NSResponder : NSObject
Conforms to	NSAnimatablePropertyContainer (NSView) NSCoding (NSResponder) NSObject (NSObject)
Framework	Library/Frameworks/Quartz.framework/Frameworks/PDFKit.framework
Declared in	PDFKit/PDFView.h
Availability	Available in Mac OS X v10.4 and later.

Overview

A `PDFView` object encapsulates the functionality of PDF Kit into a single widget that you can add to your application using Interface Builder.

`PDFView` may be the only class you need to deal with for adding PDF functionality to your application. It lets you display PDF data and allows users to select content, navigate through a document, set zoom level, and copy textual content to the Pasteboard. `PDFView` also keeps track of page history.

You can subclass `PDFView` to create a custom PDF viewer.

You can also create a custom PDF viewer by using the PDF Kit utility classes directly and not using `PDFView` at all.

Tasks

Associating a Document with a View

- `document` (page 198)
Returns the document associated with a `PDFView` object.
- `setDocument:` (page 211)
Associates a document with a `PDFView` object.

Navigating Within a Document

- `canGoBack` (page 190)
Returns a Boolean value indicating whether the user can navigate to the previous page in the page history.
- `canGoForward` (page 191)
Returns a Boolean value indicating whether the user can navigate to the next page in the page history.
- `canGoToFirstPage` (page 191)
Returns a Boolean value indicating whether the user can navigate to the first page of the document.
- `canGoToLastPage` (page 191)
Returns a Boolean value indicating whether the user can navigate to the last page of the document.
- `canGoToNextPage` (page 192)
Returns a Boolean value indicating whether the user can navigate to the next page of the document.
- `canGoToPreviousPage` (page 192)
Returns a Boolean value indicating whether the user can navigate to the previous page of the document.
- `currentPage` (page 196)
Returns the current page.
- `currentDestination` (page 195)
Returns a `PDFDestination` object representing the current page and the current point in the view specified in page space.
- `goBack:` (page 200)
Navigates back one step in the page history.
- `goForward:` (page 200)
Navigates forward one step in the page history.
- `goToFirstPage:` (page 201)
Navigates to the first page of the document.
- `goToLastPage:` (page 201)
Navigates to the last page of the document.
- `goToNextPage:` (page 201)
Navigates to the next page of the document.
- `goToPreviousPage:` (page 202)
Navigates to the previous page of the document.
- `goToPage:` (page 202)
Scrolls to the specified page.
- `goToDestination:` (page 200)
Navigates to the specified destination.
- `goToSelection:` (page 203)
Scrolls to the first character of the specified selection.
- `goToRect:onPage:` (page 202)
Navigates to the specified rectangle on the specified page.

Working with Display Modes and Characteristics

- `setDisplayMode:` (page 210)
Sets the display mode for the view.
- `displayMode` (page 197)
Returns the current display mode.
- `setDisplaysPageBreaks:` (page 210)
Toggles the display of page breaks.
- `displaysPageBreaks` (page 198)
Returns a Boolean value indicating whether the view is displaying page breaks.
- `setDisplayBox:` (page 209)
Specifies the box to display and to clip to.
- `displayBox` (page 197)
Returns the current style of display box.
- `displaysAsBook` (page 198)
Returns a Boolean value indicating whether the view will display the first page as a book cover (meaningful only when the document is in two-up or two-up continuous display mode).
- `setDisplaysAsBook:` (page 210)
Specifies whether the view should treat the document's first page as a book cover.
- `setShouldAntiAlias:` (page 212)
Specifies whether to use anti-aliasing in the view.
- `shouldAntiAlias` (page 212)
Returns a Boolean value indicating whether the view is anti-aliased.
- `setGreekingThreshold:` (page 211)
Sets the greeking threshold to use for displaying text.
- `greekingThreshold` (page 203)
Returns the current greeking threshold for the view.
- `takeBackgroundColorFrom:` (page 213)
Sets the view's background color to the specified color.
- `setBackgroundColor:` (page 207)
Sets the view's background color.
- `backgroundColor` (page 190)
Returns the view's background color.

Setting the Delegate

- `setDelegate:` (page 209)
Sets a delegate for the view.
- `delegate` (page 197)
Returns the view's delegate.

Scaling the View

- [setScaleFactor:](#) (page 212)
Sets the scale factor for the view.
- [scaleFactor](#) (page 206)
Returns the current scale factor for the view.
- [zoomIn:](#) (page 214)
Zooms in by increasing the scaling factor.
- [canZoomIn](#) (page 192)
Returns a Boolean value indicating whether the user can magnify the view—that is, zoom in.
- [zoomOut:](#) (page 214)
Zooms out by decreasing the scaling factor.
- [canZoomOut](#) (page 193)
Returns a Boolean value indicating whether the user can view an expanded area—that is, zoom out.
- [setAutoScales:](#) (page 207)
Toggles whether the scaling factor applied to a view automatically responds to resizing.
- [autoScales](#) (page 190)
Returns a Boolean value indicating whether autoscaling is set.

Working with Mouse Position and Events

- [areaOfInterestForMouse:](#) (page 189)
Returns the type of area the mouse cursor is over.
- [setCursorForAreaOfInterest:](#) (page 209)
Sets the type of mouse cursor according to the type of area the mouse cursor is over.
- [performAction:](#) (page 204)
Performs the specified action.

Handling Selections

- [currentSelection](#) (page 196)
Returns the current selection.
- [setCurrentSelection:](#) (page 208)
Sets the selection.
- [selectAll:](#) (page 206)
Selects all text in the document.
- [clearSelection](#) (page 193)
Clears the selection.
- [copy:](#) (page 195)
Copies the text in the selection, if any, to the Pasteboard.
- [scrollSelectionToVisible:](#) (page 206)
Scrolls the view until the selection is visible.

- [setCurrentSelection:animate:](#) (page 208)
Sets the selection, in an animated way, if desired.
- [setHighlightedSelections:](#) (page 211)
Highlights the specified array of selections.
- [highlightedSelections](#) (page 203)
Returns the array of selections that are highlighted using `setHighlightedSelections`.

Setting the Password

- [takePasswordFrom:](#) (page 213)
A convenience method that calls - `[[self document] setPassword:]` with the password from the specified sender.

Rendering the View and Printing

- [drawPage:](#) (page 199)
For use by subclasses of `PDFView` for custom rendering of pages.
- [drawPagePost:](#) (page 199)
For use by subclasses of `PDFView` for post-page rendering.
- [printWithInfo:autoRotate:](#) (page 205)
Prints the document with the specified printer information.
- [printWithInfo:autoRotate:pageScaling:](#) (page 205)
Prints the document with the specified printer and page-scaling information.

Conversion Methods for Subclasses

- [pageForPoint:nearest:](#) (page 204)
Returns the page containing a point specified in view coordinates.
- [convertPoint:toPage:](#) (page 194)
Converts a point from view space to page space.
- [convertRect:toPage:](#) (page 195)
Converts a rectangle from view space to page space.
- [convertPoint:fromPage:](#) (page 194)
Converts a point from page space to view space.
- [convertRect:fromPage:](#) (page 194)
Converts a rectangle from page space to view space.

Miscellaneous Methods

- [documentView](#) (page 199)
Returns the innermost view used by `PDFView` or by your `PDFView` subclass.
- [rowSizeForPage:](#) (page 205)
Returns the size needed to display a row of the current document page.

- `layoutDocumentView` (page 204)
Performs layout of the inner views.
- `allowsDragging` (page 189)
Determines whether the view can accept new PDF documents dragged into it by the user.
- `setAllowsDragging:` (page 207)
Specifies whether the view can accept drags.
- `visiblePages` (page 213)
Returns an array of `PDFPage` objects that represent the currently visible pages.
- `annotationsChangedOnPage:` (page 189)
Tells the PDF view that an annotation on the specified page has changed.

Managing Scale Factor

- `PDFViewWillChangeScaleFactor:toScale:` (page 216) *delegate method*
Delegate method for overriding changes to scale factor.

Handling URL Links

- `PDFViewWillClickOnLink:withURL:` (page 217) *delegate method*
Delegate method for handling clicks on URL links in a view.

Responding to Annotation Actions

- `PDFViewPerformFind:` (page 215) *delegate method*
Delegate method that performs a find operation.
- `PDFViewPerformGoToPage:` (page 215) *delegate method*
Delegate method that performs a go-to operation.
- `PDFViewPerformPrint:` (page 216) *delegate method*
Delegate method that prints the current document.
- `PDFViewOpenPDF:forRemoteGoToAction:` (page 214) *delegate method*
Delegate method that opens a specified page.

Changing the Print Job Title

- `PDFViewPrintJobTitle:` (page 216) *delegate method*
Delegate method that overrides the job title used when the `PDFView` is printed.

Instance Methods

allowsDragging

Determines whether the view can accept new PDF documents dragged into it by the user.

- (BOOL)allowsDragging

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setAllowsDragging:](#) (page 207)

Declared In

PDFView.h

annotationsChangedOnPage:

Tells the PDF view that an annotation on the specified page has changed.

- (void)annotationsChangedOnPage:(PDFPage *)page

Discussion

When the `PDFView` object receives this message, it rescans for tool tips and pop-ups and informs the `PDFThumbnailView` objects so the thumbnail images can be redrawn.

Availability

Available in Mac OS X v10.5 and later.

Declared In

PDFView.h

areaOfInterestForMouse:

Returns the type of area the mouse cursor is over.

- (PDFAreaOfInterest)areaOfInterestForMouse:(NSEvent *)theEvent

Discussion

The `PDFAreaOfInterest` enumeration defines the various area types. This method is for custom subclasses of the `PDFView` class. Use it if you override the `NSResponder` class's `mouseMoved:` method or related methods.

Refer to “[Constants](#)” (page 217) for the various values of the area-of-interest constants. Each of these constants contributes to the value of the `PDFAreaOfInterest` bit field.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setCursorForAreaOfInterest:](#) (page 209)

Declared In

PDFView.h

autoScales

Returns a Boolean value indicating whether autoscaling is set.

- (BOOL)autoScales

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setAutoScales:](#) (page 207)

Declared In

PDFView.h

backgroundColor

Returns the view's background color.

- (NSColor *)backgroundColor

Discussion

A view's background is the area displayed to either side of a PDF document's pages. The background also appears between pages when page breaks are enabled. The default color is a 50% gray.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [takeBackgroundColorFrom:](#) (page 213)

- [setBackgroundColor:](#) (page 207)

Declared In

PDFView.h

canGoBack

Returns a Boolean value indicating whether the user can navigate to the previous page in the page history.

- (BOOL)canGoBack

Discussion

The page history gets built as your application calls navigation methods such as [goToDestination:](#) (page 200) and [goToLastPage:](#) (page 201).

Availability

Available in Mac OS X v10.4 and later.

See Also

- [goBack](#): (page 200)

Declared In

PDFView.h

canGoForward

Returns a Boolean value indicating whether the user can navigate to the next page in the page history.

- (BOOL)canGoForward

Discussion

The page history gets built as your application calls navigation methods such as [goToDestination](#): (page 200) and [goToLastPage](#): (page 201).

Availability

Available in Mac OS X v10.4 and later.

See Also

- [goForward](#): (page 200)

Declared In

PDFView.h

canGoToFirstPage

Returns a Boolean value indicating whether the user can navigate to the first page of the document.

- (BOOL)canGoToFirstPage

Discussion

The return value will be YES unless the view is already displaying the first page.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [goToFirstPage](#): (page 201)

Declared In

PDFView.h

canGoToLastPage

Returns a Boolean value indicating whether the user can navigate to the last page of the document.

- (BOOL)canGoToLastPage

Discussion

The return value will be YES unless the view is already displaying the last page.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [goToLastPage:](#) (page 201)

Declared In

PDFView.h

canGoToNextPage

Returns a Boolean value indicating whether the user can navigate to the next page of the document.

- (BOOL)canGoToNextPage

Discussion

The return value will be YES unless the view is displaying the last page.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [goToNextPage:](#) (page 201)

Declared In

PDFView.h

canGoToPreviousPage

Returns a Boolean value indicating whether the user can navigate to the previous page of the document.

- (BOOL)canGoToPreviousPage

Discussion

The return value will be YES unless the view is displaying the first page.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [canGoToPreviousPage](#) (page 192)

Declared In

PDFView.h

canZoomIn

Returns a Boolean value indicating whether the user can magnify the view—that is, zoom in.

- (BOOL)canZoomIn

Availability

Available in Mac OS X v10.4 and later.

See Also

- [zoomIn:](#) (page 214)
- [zoomOut:](#) (page 214)
- [canZoomOut](#) (page 193)

Declared In

PDFView.h

canZoomOut

Returns a Boolean value indicating whether the user can view an expanded area—that is, zoom out.

- (BOOL)canZoomOut

Availability

Available in Mac OS X v10.4 and later.

See Also

- [zoomIn:](#) (page 214)
- [canZoomIn](#) (page 192)
- [zoomOut:](#) (page 214)

Declared In

PDFView.h

clearSelection

Clears the selection.

- (void)clearSelection

Discussion

The view redraws as necessary but does not scroll. This call is equivalent to calling `[PDFView setCurrentSelection:NULL]`.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [currentSelection](#) (page 196),
- [setCurrentSelection:](#) (page 208)

Declared In

PDFView.h

convertPoint:fromPage:

Converts a point from page space to view space.

```
- (NSPoint)convertPoint:(NSPoint)point fromPage:(PDFPage *)page
```

Discussion

Page space is a 72 dpi coordinate system with the origin at the lower-left corner of the current page. View space is a coordinate system with the origin at the lower-left corner of the current PDF view.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [convertPoint:toPage:](#) (page 194)
- [convertRect:toPage:](#) (page 195)
- [convertRect:fromPage:](#) (page 194)
- [pageForPoint:nearest:](#) (page 204)

Declared In

PDFView.h

convertPoint:toPage:

Converts a point from view space to page space.

```
- (NSPoint)convertPoint:(NSPoint)point toPage:(PDFPage *)page
```

Discussion

Page space is a 72 dpi coordinate system with the origin at the lower-left corner of the current page. View space is a coordinate system with the origin at the lower-left corner of the current PDF view.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [convertRect:toPage:](#) (page 195)
- [convertPoint:fromPage:](#) (page 194)
- [convertRect:fromPage:](#) (page 194)

Declared In

PDFView.h

convertRect:fromPage:

Converts a rectangle from page space to view space.

```
- (NSRect)convertRect:(NSRect)rect fromPage:(PDFPage *)page
```

Discussion

Page space is a 72 dpi coordinate system with the origin at the lower-left corner of the current page. View space is a coordinate system with the origin at the lower-left corner of the current PDF view.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [convertPoint:toPage:](#) (page 194)
- [convertRect:toPage:](#) (page 195)
- [convertPoint:fromPage:](#) (page 194)

Declared In

PDFView.h

convertRect:toPage:

Converts a rectangle from view space to page space.

```
- (NSRect)convertRect:(NSRect)rect toPage:(PDFPage *)page
```

Discussion

Page space is a 72 dpi coordinate system with the origin at the lower-left corner of the current page. View space is a coordinate system with the origin at the lower-left corner of the current PDF view.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [convertPoint:toPage:](#) (page 194)
- [convertPoint:fromPage:](#) (page 194)
- [convertRect:fromPage:](#) (page 194)

Declared In

PDFView.h

copy:

Copies the text in the selection, if any, to the Pasteboard.

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

Availability

Available in Mac OS X v10.4 and later.

See Also

- [currentSelection](#) (page 196)

Declared In

PDFView.h

currentDestination

Returns a `PDFDestination` object representing the current page and the current point in the view specified in page space.

- ([PDFDestination](#) *)currentDestination

Discussion

Page space is a 72 dpi coordinate system with the origin at the lower-left corner of the current page.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [currentPage](#) (page 196)
- [goToDestination:](#) (page 200) ([PDFDestination](#))

Declared In

PDFView.h

currentPage

Returns the current page.

- ([PDFPage](#) *)currentPage

Discussion

When there are two pages in the view in a two-up mode, “current page” is the left page. For continuous modes, returns the page crossing a horizontal line halfway between the view’s top and bottom bounds.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [currentDestination](#) (page 195)
- [goToDestination:](#) (page 200)

Declared In

PDFView.h

currentSelection

Returns the current selection.

- ([PDFSelection](#) *)currentSelection

Discussion

Returns NULL if no selection exists.

Note that this method returns the actual instance of the current [PDFSelection](#) object. Therefore, if you want to modify it, you should make a copy of the returned selection and modify that, instead.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setCurrentSelection:](#) (page 208)
- [clearSelection](#) (page 193)

Declared In

PDFView.h

delegate

Returns the view's delegate.

- (id)delegate

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setDelegate:](#) (page 209)

Declared In

PDFView.h

displayBox

Returns the current style of display box.

- (PDFDisplayBox)displayBox

Discussion

The available values for display boxes are defined in the Constants section in the `PDFPage` class.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setDisplayBox:](#) (page 209)

Declared In

PDFView.h

displayMode

Returns the current display mode.

- (PDFDisplayMode)displayMode

Discussion

See [“Constants”](#) (page 217) for possible values.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setDisplayMode:](#) (page 210)

Declared In

PDFView.h

displaysAsBook

Returns a Boolean value indicating whether the view will display the first page as a book cover (meaningful only when the document is in two-up or two-up continuous display mode).

- (BOOL)displaysAsBook

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setDisplayAsBook:](#) (page 210)

Declared In

PDFView.h

displaysPageBreaks

Returns a Boolean value indicating whether the view is displaying page breaks.

- (BOOL)displaysPageBreaks

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setDisplayPageBreaks:](#) (page 210)

Declared In

PDFView.h

document

Returns the document associated with a PDFView object.

- (PDFDocument *)document

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setDocument:](#) (page 211)

Declared In

PDFView.h

documentView

Returns the innermost view used by `PDFView` or by your `PDFView` subclass.

```
- (NSView *)documentView
```

Discussion

The innermost view is the one displaying the visible document pages. This method is useful when converting coordinates from one view to another.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [layoutDocumentView](#) (page 204)

Declared In

PDFView.h

drawPage:

For use by subclasses of `PDFView` for custom rendering of pages.

```
- (void)drawPage:(PDFPage *)page
```

Discussion

Do not invoke this method, except by invoking it on `super` from a subclass.

The `PDFView` class calls `drawPage:` (page 199) as necessary for each visible page that requires rendering. In the `PDFView` class, this method erases *page* to white, calls `[page drawInRect: pageRect withBox: [self displayBox]]`, and then draws the selection, if any.

You can override this method to draw on top of a PDF page or to control how pages are drawn. In these cases, invoke this method on `super` and then perform custom drawing on top of the PDF page.

Availability

Available in Mac OS X v10.4 and later.

Declared In

PDFView.h

drawPagePost:

For use by subclasses of `PDFView` for post-page rendering.

```
- (void)drawPagePost:(PDFPage *)page
```

Discussion

The default implementation of this method draws the text highlighting (if any) for the page. This method does not apply scaling or rotating to the current context to map to page space; instead, the context is in view-space coordinates (in which the origin is at the lower-left corner of the current PDF view).

Availability

Available in Mac OS X v10.5 and later.

Declared In

PDFView.h

goBack:

Navigates back one step in the page history.

- (IBAction)goBack:(id)sender

Discussion

The page history gets built as your application calls navigation methods such as [goToDestination:](#) (page 200) and [goToLastPage:](#) (page 201).

Availability

Available in Mac OS X v10.4 and later.

See Also

- [canGoBack](#) (page 190)

Declared In

PDFView.h

goForward:

Navigates forward one step in the page history.

- (IBAction)goForward:(id)sender

Discussion

The page history gets built as your application calls navigation methods such as [goToDestination:](#) (page 200) and [goToLastPage:](#) (page 201).

Availability

Available in Mac OS X v10.4 and later.

See Also

- [canGoForward](#) (page 191)

Declared In

PDFView.h

goToDestination:

Navigates to the specified destination.

- (void)goToDestination:(PDFDestination *)destination

Discussion

Destinations include a page and a point on the page specified in page space.

Page space is a 72 dpi coordinate system with the origin at the lower-left corner of the current page.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [currentDestination](#) (page 195) (PDFDestination)
- [currentPage](#) (page 196)

Declared In

PDFView.h

goToFirstPage:

Navigates to the first page of the document.

- (IBAction)goToFirstPage:(id)sender

Discussion

PDF Kit records the move in its page history.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [canGoToFirstPage](#) (page 191)

Declared In

PDFView.h

goToLastPage:

Navigates to the last page of the document.

- (IBAction)goToLastPage:(id)sender

Discussion

PDF Kit records the move in its page history.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [canGoToLastPage](#) (page 191)

Declared In

PDFView.h

goToNextPage:

Navigates to the next page of the document.

- (IBAction)goToNextPage:(id)sender

Discussion

PDF Kit records the move in its page history.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [canGoToNextPage](#) (page 192)

Declared In

PDFView.h

goToPage:

Scrolls to the specified page.

```
- (void)goToPage:(PDFPage *)page
```

Discussion

PDF Kit records the move in its page history.

Availability

Available in Mac OS X v10.4 and later.

Declared In

PDFView.h

goToPreviousPage:

Navigates to the previous page of the document.

```
- (IBAction)goToPreviousPage:(id)sender
```

Discussion

PDF Kit records the move in its page history.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [canGoToPreviousPage](#) (page 192)

Declared In

PDFView.h

goToRect:onPage:

Navigates to the specified rectangle on the specified page.

```
- (void)goToRect:(NSRect)rect onPage:(PDFPage *)page
```

Discussion

If the specified rectangle is already visible, this method does nothing. This allows you to scroll the `PDFView` object to a specific `PDFAnnotation` or `PDFSelection` object, because both of these objects have `bounds` methods that return an annotation or selection position in page space.

Note that *rect* is specified in page-space coordinates. Page space is a 72 dpi coordinate system with the origin at the lower-left corner of the current page.

Availability

Available in Mac OS X v10.5 and later.

Declared In

PDFView.h

goToSelection:

Scrolls to the first character of the specified selection.

```
- (void)goToSelection:(PDFSelection *)selection
```

Discussion

PDF Kit records the move in its page history.

Availability

Available in Mac OS X v10.4 and later.

Declared In

PDFView.h

greekingThreshold

Returns the current greeking threshold for the view.

```
- (float)greekingThreshold
```

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setGreekingThreshold:](#) (page 211)

Declared In

PDFView.h

highlightedSelections

Returns the array of selections that are highlighted using `setHighlightedSelections`.

```
- (NSArray *)highlightedSelections
```

Availability

Available in Mac OS X v10.5 and later.

See Also

- [setHighlightedSelections:](#) (page 211)

Declared In

PDFView.h

layoutDocumentView

Performs layout of the inner views.

- (void)layoutDocumentView

Discussion

The `PDFView` actually contains several subviews, such as the document view (where the PDF is actually drawn) and a “matte view” (which may appear as a gray area around the PDF content, depending on the scaling). Changes to the PDF content may require changes to these inner views, so you must call this method explicitly if you use PDF Kit utility classes to add or remove a page, rotate a page, or perform other operations affecting visible layout.

This method is called automatically from `PDFView` methods that affect the visible layout (such as [setDocument:](#) (page 211), [setDisplayBox:](#) (page 209) or [zoomIn:](#) (page 214)).

Availability

Available in Mac OS X v10.4 and later.

See Also

- [documentView](#) (page 199)

Declared In

PDFView.h

pageForPoint:nearest:

Returns the page containing a point specified in view coordinates.

- (PDFPage *)pageForPoint:(NSPoint)point nearest:(BOOL)nearest

Discussion

Returns `NULL` if there’s no page at the specified point and `nearest` is set to `NO`.

Availability

Available in Mac OS X v10.4 and later.

Declared In

PDFView.h

performAction:

Performs the specified action.

- (void)performAction:(PDFAction *)action

Availability

Available in Mac OS X v10.5 and later.

Declared In

PDFView.h

printWithInfo:autoRotate:

Prints the document with the specified printer information.

```
- (void)printWithInfo:(NSPrintInfo *)printInfo autoRotate:(BOOL)doRotate
```

Discussion

If *autoRotate* is set to YES, then this method ignores the orientation attribute in the `NSPrintInfo` object and instead chooses the orientation that best fits the page to the paper size. This orientation occurs on a page-by-page basis.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [printWithInfo:autoRotate:pageScaling:](#) (page 205)

Declared In

PDFView.h

printWithInfo:autoRotate:pageScaling:

Prints the document with the specified printer and page-scaling information.

```
- (void)printWithInfo:(NSPrintInfo *)printInfo autoRotate:(BOOL)doRotate  
  pageScaling:(PDFPrintScalingMode)scale
```

Discussion

If *pageScaling* is set to `kPDFPrintPageScaleToFit`, each page is scaled up or down to best fit the paper size. If *pageScaling* is set to `kPDFPrintPageScaleDownToFit`, only large pages are scaled down to fit; small pages are not scaled up to fit. Specifying `kPDFPrintPageScaleNone` for *pageScaling* is equivalent to calling [printWithInfo:autoRotate:](#) (page 205). See `PDFDocument` for more information on page-scaling types.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [printWithInfo:autoRotate:](#) (page 205)

Declared In

PDFView.h

rowSizeForPage:

Returns the size needed to display a row of the current document page.

- (NSSize)rowSizeForPage:(PDFPage *)page

Discussion

The size is dependent on the current scale factor and display attributes.

Availability

Available in Mac OS X v10.4 and later.

Declared In

PDFView.h

scaleFactor

Returns the current scale factor for the view.

- (float)scaleFactor

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setScaleFactor:](#) (page 212)

Declared In

PDFView.h

scrollSelectionToVisible:

Scrolls the view until the selection is visible.

- (void)scrollSelectionToVisible:(id)sender

Availability

Available in Mac OS X v10.4 and later.

Declared In

PDFView.h

selectAll:

Selects all text in the document.

- (IBAction)selectAll:(id)sender

Availability

Available in Mac OS X v10.4 and later.

Declared In

PDFView.h

setAllowsDragging:

Specifies whether the view can accept drags.

```
- (void)setAllowsDragging:(BOOL)allow
```

Discussion

If set to YES, the user can drag a new PDF document into the view. The new document is then displayed in the view, and the old document is released.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [allowsDragging](#) (page 189)

Declared In

PDFView.h

setAutoScales:

Toggles whether the scaling factor applied to a view automatically responds to resizing.

```
- (void)setAutoScales:(BOOL)newAuto
```

Discussion

When set to autoscaling, the document scales to fill the PDFView object as the user resizes it.

For the single-page and two-up continuous modes, autoscaling fits the page to the width of the view. For single-page and two-up noncontinuous modes, autoscaling provides best fit, in which the viewed pages are as large as possible while displaying in their entirety within the view.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [autoScales](#) (page 190)

Declared In

PDFView.h

setBackground-color:

Sets the view's background color.

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

Discussion

A view's background is the area displayed to either side of a PDF document's pages. The background also appears between pages when page breaks are enabled. The default color is a 50% gray.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [takeBackgroundColorFrom:](#) (page 213)
- [backgroundColor](#) (page 190)

Declared In

PDFView.h

setCurrentSelection:

Sets the selection.

```
- (void)setCurrentSelection:(PDFSelection *)selection
```

Discussion

The view redraws as necessary but does not scroll. If you need to scroll to the current selection, use [scrollSelectionToVisible:](#) (page 206). If you pass `nil` for the selection, this call is equivalent to calling [clearSelection](#) (page 193).

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setCurrentSelection:animate:](#) (page 208)
- [currentSelection](#) (page 196)
- [clearSelection](#) (page 193)

Declared In

PDFView.h

setCurrentSelection:animate:

Sets the selection, in an animated way, if desired.

```
- (void)setCurrentSelection:(PDFSelection *)selection animate:(BOOL)animate
```

Discussion

This method behaves as [setCurrentSelection:](#) (page 208), but with the addition of animation, if *animate* is YES. The animation serves to draw the user's attention to the new selection, which can be useful when implementing search.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [setCurrentSelection:](#) (page 208)
- [clearSelection](#) (page 193)

Declared In

PDFView.h

setCursorForAreaOfInterest:

Sets the type of mouse cursor according to the type of area the mouse cursor is over.

```
- (void)setCursorForAreaOfInterest:(PDFAreaOfInterest)area
```

Discussion

This method is especially useful for custom subclasses of the `PDFView` class.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [areaOfInterestForMouse:](#) (page 189)

Declared In

PDFView.h

setDelegate:

Sets a delegate for the view.

```
- (void)setDelegate:(id)anObject
```

Availability

Available in Mac OS X v10.4 and later.

See Also

- [delegate](#) (page 197)

Declared In

PDFView.h

setDisplayBox:

Specifies the box to display and to clip to.

```
- (void)setDisplayBox:(PDFDisplayBox)box
```

Discussion

The values for *box* are defined in the `PDFDisplayBox` enumeration. The default value for this method is `kPDFDisplayBoxCropBox`.

The available values for display boxes are defined in the Constants section in the `PDFPage` class.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [displayBox](#) (page 197)

Declared In

PDFView.h

setDisplayMode:

Sets the display mode for the view.

```
- (void)setDisplayMode:(PDFDisplayMode)mode
```

Discussion

Available display modes are single page, single-page continuous, two-up, and two-up continuous, as defined in “Constants” (page 217).

Availability

Available in Mac OS X v10.4 and later.

See Also

- [displayMode](#) (page 197)

Declared In

PDFView.h

setDisplaysAsBook:

Specifies whether the view should treat the document’s first page as a book cover.

```
- (void)setDisplaysAsBook:(BOOL)asBook
```

Discussion

For two-up modes, a YES value for this method specifies that the first page should be displayed by itself.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [displaysAsBook](#) (page 198)

Declared In

PDFView.h

setDisplaysPageBreaks:

Toggles the display of page breaks.

```
- (void)setDisplaysPageBreaks:(BOOL)breaks
```

Availability

Available in Mac OS X v10.4 and later.

See Also

- [displaysPageBreaks](#) (page 198)

Declared In

PDFView.h

setDocument:

Associates a document with a PDFView object.

```
- (void)setDocument:(PDFDocument *)document
```

Discussion

If a document was already associated with the view, it is released first and then *document* is associated with the view.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [document](#) (page 198)

Declared In

PDFView.h

setGreekingThreshold:

Sets the greeking threshold to use for displaying text.

```
- (void)setGreekingThreshold:(float)threshold
```

Discussion

The default threshold is 3.0.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [greekingThreshold](#) (page 203)

Declared In

PDFView.h

setHighlightedSelections:

Highlights the specified array of selections.

```
- (void)setHighlightedSelections:(NSArray *)selections
```

Discussion

Unlike the selections users set (using, for example, [setCurrentSelection:](#) (page 208)), the selections you specify in this method do not go away (that is, appear deselected) when users click elsewhere in the view or document. Instead, to deselect the selections, you must call `[setHighlightedSelections:NULL]` to remove them.

You might use this method to highlight the set of matches from a text search. To prevent the user from confusing their own selections with selections you set using this method, it is recommended that you use a highlight color that is different from the user's default text selection color.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [highlightedSelections](#) (page 203)

Declared In

PDFView.h

setScaleFactor:

Sets the scale factor for the view.

```
- (void)setScaleFactor:(float)scale
```

Discussion

The default value is 1.0, corresponding to actual size.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [scaleFactor](#) (page 206)

Declared In

PDFView.h

setShouldAntiAlias:

Specifies whether to use anti-aliasing in the view.

```
- (void)setShouldAntiAlias:(BOOL)aliasing
```

Discussion

The default value is YES.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [shouldAntiAlias](#) (page 212)

Declared In

PDFView.h

shouldAntiAlias

Returns a Boolean value indicating whether the view is anti-aliased.

```
- (BOOL)shouldAntiAlias
```

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setShouldAntiAlias](#): (page 212)

Declared In

PDFView.h

takeBackgroundColorFrom:

Sets the view's background color to the specified color.

- (IBAction)takeBackgroundColorFrom:(id)sender

Discussion

A view's background is the area displayed to either side of a PDF document's pages. The background also appears between pages when page breaks are enabled. The default color is a 50% gray.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setBackground-color](#): (page 207)

- [background-color](#) (page 190)

Declared In

PDFView.h

takePasswordFrom:

A convenience method that calls - `[[self document] setPassword:]` with the password from the specified sender.

- (void)takePasswordFrom:(id)sender

Availability

Available in Mac OS X v10.4 and later.

Declared In

PDFView.h

visiblePages

Returns an array of PDFPage objects that represent the currently visible pages.

- (NSArray *)visiblePages

Availability

Available in Mac OS X v10.5 and later.

Declared In

PDFView.h

zoomIn:

Zooms in by increasing the scaling factor.

- (IBAction)zoomIn:(id)sender

Discussion

Each invocation of `zoomIn` multiplies the scaling factor by the square root of 2.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [canZoomIn](#) (page 192)
- [zoomOut:](#) (page 214)
- [canZoomOut](#) (page 193)

Declared In

PDFView.h

zoomOut:

Zooms out by decreasing the scaling factor.

- (IBAction)zoomOut:(id)sender

Discussion

Each invocation of `zoomOut` divides the scaling factor by the square root of 2.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [zoomIn:](#) (page 214)
- [canZoomIn](#) (page 192)
- [canZoomOut](#) (page 193)

Declared In

PDFView.h

Delegate Methods

PDFViewOpenPDF:forRemoteGoToAction:

Delegate method that opens a specified page.

```
- (void)PDFViewOpenPDF:(PDFView *)sender forRemoteGoToAction:(PDFActionRemoteGoTo *)action
```

Discussion

A delegate responding to this method is called to handle clicks in an annotation that contains a `PDFActionRemoteGoTo` action. Such an action contains a URL, a page index, and a point. The delegate should open the PDF document specified by the URL and go to the specified page and point. An easy way to do this is to create a `PDFDocument` object with the specified URL and then create a `PDFDestination` object with the specified page and point. Then, you can call `goToDestination:` (page 200).

The default implementation of this method beeps.

Availability

Available in Mac OS X v10.5 and later.

Declared In

PDFView.h

PDFViewPerformFind:

Delegate method that performs a find operation.

```
- (void)PDFViewPerformFind:(PDFView *)sender
```

Discussion

Some `PDFAction` objects request a PDF viewer application to perform a find operation. A delegate responding to this method is called when users click an annotation with such an action.

Availability

Available in Mac OS X v10.5 and later.

Declared In

PDFView.h

PDFViewPerformGoToPage:

Delegate method that performs a go-to operation.

```
- (void)PDFViewPerformGoToPage:(PDFView *)sender
```

Discussion

Some `PDFAction` objects request a PDF viewer application to display a panel that allows users to enter a page number to go to. A delegate responding to this method is called when users click an annotation with such an action.

Availability

Available in Mac OS X v10.5 and later.

Declared In

PDFView.h

PDFViewPerformPrint:

Delegate method that prints the current document.

```
- (void)PDFViewPerformPrint:(PDFView *)sender
```

Discussion

Some `PDFAction` objects request a PDF viewer application to print the current document. A delegate responding to this method is called when users click an annotation with such an action.

Availability

Available in Mac OS X v10.5 and later.

Declared In

PDFView.h

PDFViewPrintJobTitle:

Delegate method that overrides the job title used when the `PDFView` is printed.

```
- (NSString *)PDFViewPrintJobTitle:(PDFView *)sender
```

Discussion

By default, this method uses the string, if any, associated with the “Title” key in the view’s `PDFDocument` attribute dictionary. If there is no such string, this method uses the last path component if the document is URL-based.

Availability

Available in Mac OS X v10.5 and later.

Declared In

PDFView.h

PDFViewWillChangeScaleFactor:toScale:

Delegate method for overriding changes to scale factor.

```
- (float)PDFViewWillChangeScaleFactor:(PDFView *)sender toScale:(float)scale
```

Discussion

By default, the scale factor is restricted to a range between 0.1 and 10.0 inclusive.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setScaleFactor:](#) (page 212)

Declared In

PDFView.h

PDFViewWillClickOnLink:withURL:

Delegate method for handling clicks on URL links in a view.

```
- (void)PDFViewWillClickOnLink:(PDFView *)sender withURL:(NSURL *)url
```

Discussion

By default, this method calls `[[NSWorkspace sharedWorkspace] openURL:url]`.

Availability

Available in Mac OS X v10.5 and later.

Declared In

PDFView.h

Constants

PDF views use the following display mode constants:

Constant	Description
<code>kPDFDisplaySinglePage</code>	The document displays one page at a time horizontally and vertically. Vertical and horizontal scrolling apply only to the current page. Available in Mac OS X v10.4 and later. Declared in <code>PDFView.h</code> .
<code>kPDFDisplaySingle-PageContinuous</code>	The document displays in continuous mode vertically, with single-page width horizontally. Vertical scrolling applies to the entire document. Available in Mac OS X v10.4 and later. Declared in <code>PDFView.h</code> .
<code>kPDFDisplayTwoUp</code>	The document displays two pages side-by-side. Vertical and horizontal scrolling apply only to the pair of displayed pages Available in Mac OS X v10.4 and later. Declared in <code>PDFView.h</code> .
<code>kPDFDisplayTwo-UpContinuous</code>	The document displays in continuous mode vertically and displays two pages side-by-side horizontally. Vertical scrolling applies to the entire document. Available in Mac OS X v10.4 and later. Declared in <code>PDFView.h</code> .

The following constants apply to mouse position over PDF view areas. These constants are components of a bit field and may be combined arbitrarily:

Constant	Description
<code>kPDFNoArea</code>	The mouse is over an undefined area. Available in Mac OS X v10.4 and later. Declared in <code>PDFView.h</code> .

Constant	Description
kPDFPageArea	The mouse is over a page. Available in Mac OS X v10.4 and later. Declared in <code>PDFView.h</code> .
kPDFTextArea	The mouse is over text. Available in Mac OS X v10.4 and later. Declared in <code>PDFView.h</code> .
kPDFAnnotationArea	The mouse is over an annotation. Available in Mac OS X v10.4 and later. Declared in <code>PDFView.h</code> .
kPDFLinkArea	The mouse is over a link. Available in Mac OS X v10.4 and later. Declared in <code>PDFView.h</code> .
kPDFControlArea	The mouse is over a control. Available in Mac OS X v10.4 and later. Declared in <code>PDFView.h</code> .
kPDFTextFieldArea	The mouse is over a text field. Available in Mac OS X v10.4 and later. Declared in <code>PDFView.h</code> .
kPDFIconArea	The mouse is over an icon. Available in Mac OS X v10.5 and later. Declared in <code>PDFView.h</code> .
kPDFPopupArea	The mouse is over a popup menu. Available in Mac OS X v10.4 and later. Declared in <code>PDFView.h</code> .

Notifications

A `PDFView` object posts the following notifications:

PDFViewChangedHistoryNotification

Posted when the page history changes.

The notification object is the `PDFView` object itself.

Availability

Available in Mac OS X v10.4 and later.

Declared In

`PDFView.h`

PDFViewDocumentChangedNotification

Posted when a new document is associated with the view.

The notification object is the `PDFView` object itself.

Availability

Available in Mac OS X v10.4 and later.

Declared In

`PDFView.h`

PDFViewPageChangedNotification

Posted when a new page becomes the current page.

The notification object is the `PDFView` object itself.

Availability

Available in Mac OS X v10.4 and later.

Declared In

`PDFView.h`

PDFViewScaleChangedNotification

Posted when the scale factor changes.

The notification object is the `PDFView` object itself.

Availability

Available in Mac OS X v10.4 and later.

Declared In

`PDFView.h`

PDFViewAnnotationHitNotification

Posted when the user clicks on an annotation.

The notification object is the `PDFView` object itself.

Use the @"PDFAnnotationHit" key to obtain userinfo of type `PDFAnnotation *`.

Availability

Available in Mac OS X v10.4 and later.

Declared In

`PDFView.h`

PDFViewCopyPermissionNotification

Posted when the user attempts to copy to the pasteboard without the appropriate permissions.

The notification object is the `PDFView` object itself.

Availability

Available in Mac OS X v10.4 and later.

Declared In

PDFView.h

PDFViewPrintPermissionNotification

Posted when the user attempts to print without the appropriate permissions.

The notification object is the PDFView object itself.

Availability

Available in Mac OS X v10.4 and later.

Declared In

PDFView.h

PDFViewAnnotationWillHitNotification

Posted before the user clicks an annotation.

The notification object is the PDFView object itself.

Availability

Available in Mac OS X v10.5 and later.

Declared In

PDFView.h

PDFViewSelectionChangedNotification

Posted when the current selection has changed.

The notification object is the PDFView object itself.

Availability

Available in Mac OS X v10.5 and later.

Declared In

PDFView.h

PDFViewDisplayModeChangedNotification

Posted when the display mode has changed.

The notification object is the PDFView object itself.

Availability

Available in Mac OS X v10.5 and later.

Declared In

PDFView.h

PDFViewDisplayBoxChangedNotification

Posted when the display box has changed.

The notification object is the `PDFView` object itself.

Availability

Available in Mac OS X v10.5 and later.

Declared In

`PDFView.h`

Document Revision History

This table describes the changes to *PDF Kit Reference Collection*.

Date	Notes
2007-12-11	Updated to include information on new classes and methods introduced in Mac OS X v10.5.
2007-06-08	Updated documentation of the PDFAnnotation and PDFDocument classes.
2006-05-23	First publication of this content as a collection of separate documents.
2005-11-09	Clarified that calling [PDFView setCurrentSelection:nil] is equivalent to calling [PDFView clearSelection].

REVISION HISTORY

Document Revision History

Index

A

`action` instance method 147
`addAnnotation:` instance method 155
`addBezierPath:` instance method 74
`addSelection:` instance method 168
`addSelections:` instance method 169
`alignment` instance method 70, 100
`allowsCopying` instance method 120
`allowsDragging` instance method 176, 189
`allowsMultipleSelection` instance method 177
`allowsPrinting` instance method 121
`allowsToggleToOff` instance method 49
`annotationAtPoint:` instance method 156
`annotations` instance method 156
`annotationsChangedOnPage:` instance method 189
`areaOfInterestForMouse:` instance method 189
`attributedString` instance method 156, 169
`autoScales` instance method 190

B

`backgroundColor` instance method 49, 60, 101, 177, 190
`beginFindString:withOptions:` instance method 121
`beginFindStrings:withOptions:` instance method 122
`border` instance method 35
`bounds` instance method 36
`boundsForBox:` instance method 157
`boundsForPage:` instance method 169

C

`cancelFindString` instance method 122
`canGoBack` instance method 190
`canGoForward` instance method 191

`canGoToFirstPage` instance method 191
`canGoToLastPage` instance method 191
`canGoToNextPage` instance method 192
`canGoToPreviousPage` instance method 192
`canZoomIn` instance method 192
`canZoomOut` instance method 193
`caption` instance method 49
`characterBoundsAtIndex:` instance method 157
`characterIndexAtPoint:` instance method 158
`childAtIndex:` instance method 147
`choices` instance method 61
`clearSelection` instance method 193
`color` instance method 36, 170
`compare:` instance method 114
`contents` instance method 36
`controlType` instance method 50
`convertPoint:fromPage:` instance method 194
`convertPoint:toPage:` instance method 194
`convertRect:fromPage:` instance method 194
`convertRect:toPage:` instance method 195
`copy:` instance method 195
`currentDestination` instance method 195
`currentPage` instance method 196
`currentSelection` instance method 196

D

`dashPattern` instance method 108
`dataRepresentation` instance method 122, 158
`delegate` instance method 123, 197
`destination` instance method 16, 82, 148
Destination Undefined 116
`didMatchString: <NSObject> delegate` method 136
`displayBox` instance method 197
`displayMode` instance method 197
`displaysAnnotations` instance method 158
`displaysAsBook` instance method 198
`displaysPageBreaks` instance method 198
Document Attribute Keys 138
`document` instance method 148, 159, 198
`documentAttributes` instance method 123

documentDidBeginDocumentFind: <NSObject>
 delegate method [136](#)
 documentDidBeginPageFind: <NSObject> delegate
 method [137](#)
 documentDidEndDocumentFind: <NSObject> delegate
 method [137](#)
 documentDidEndPageFind: <NSObject> delegate
 method [137](#)
 documentDidFindMatch: <NSObject> delegate method
[138](#)
 documentDidUnlock: <NSObject> delegate method
[138](#)
 documentURL instance method [123](#)
 documentView instance method [199](#)
 drawForPage:active: instance method [170](#)
 drawForPage:withBox:active: instance method [170](#)
 drawInRect: instance method [108](#)
 drawPage: instance method [199](#)
 drawPagePost: instance method [199](#)
 drawWithBox: instance method [37, 159](#)

E

endLineStyle instance method [76](#)
 endPoint instance method [76](#)
 exchangePageAtIndex:withPageAtIndex: instance
 method [124](#)
 extendSelectionAtEnd: instance method [171](#)
 extendSelectionAtStart: instance method [171](#)

F

fieldName instance method [50, 61, 101](#)
 fields instance method [26](#)
 fieldsIncludedAreCleared instance method [26](#)
 findString:fromSelection:withOptions: instance
 method [124](#)
 findString:withOptions: instance method [125](#)
 font instance method [51, 61, 70, 101](#)
 fontColor instance method [51, 62, 70, 102](#)

G

goBack: instance method [200](#)
 goForward: instance method [200](#)
 goToDestination: instance method [200](#)
 goToFirstPage: instance method [201](#)
 goToLastPage: instance method [201](#)
 goToNextPage: instance method [201](#)

goToPage: instance method [202](#)
 goToPreviousPage: instance method [202](#)
 goToRect:onPage: instance method [202](#)
 goToSelection: instance method [203](#)
 greekingThreshold instance method [203](#)

H

hasAppearanceStream instance method [37](#)
 highlightedSelections instance method [203](#)
 horizontalCornerRadius instance method [108](#)

I

iconType instance method [96](#)
 index instance method [148](#)
 indexForPage: instance method [125](#)
 init instance method [26, 149](#)
 initWithBounds: instance method [38](#)
 initWithData: instance method [126](#)
 initWithDestination instance method [16](#)
 initWithDocument: instance method [149, 159, 171](#)
 initWithImage: instance method [160](#)
 initWithName: instance method [18](#)
 initWithPage:atPoint: instance method [114](#)
 initWithPageIndex:atPoint:fileURL: instance
 method [22](#)
 initWithURL: instance method [30, 126](#)
 insertChild:atIndex: instance method [149](#)
 insertPage:atIndex: instance method [126](#)
 interiorColor instance method [67, 77, 91](#)
 isEncrypted instance method [127](#)
 isFinding instance method [127](#)
 isHighlighted instance method [51](#)
 isListChoice instance method [62](#)
 isLocked instance method [128](#)
 isOpen instance method [89, 150](#)

K

kPDFActionNamedFind constant [20](#)
 kPDFActionNamedFirstPage constant [20](#)
 kPDFActionNamedGoBack constant [20](#)
 kPDFActionNamedGoForward constant [20](#)
 kPDFActionNamedGoToPage constant [20](#)
 kPDFActionNamedLastPage constant [20](#)
 kPDFActionNamedNextPage constant [19](#)
 kPDFActionNamedNone constant [19](#)
 kPDFActionNamedPreviousPage constant [19](#)

[kPDFActionNamedPrint](#) **constant** 20
[kPDFActionNamedZoomIn](#) **constant** 20
[kPDFActionNamedZoomOut](#) **constant** 20
[kPDFAnnotationArea](#) **constant** 218
[kPDFBorderStyleBeveled](#) **constant** 112
[kPDFBorderStyleDashed](#) **constant** 111
[kPDFBorderStyleInset](#) **constant** 112
[kPDFBorderStyleSolid](#) **constant** 111
[kPDFBorderStyleUnderline](#) **constant** 112
[kPDFControlArea](#) **constant** 218
[kPDFDestinationUnspecifiedValue](#) **constant** 116
[kPDFDisplayBoxArtBox](#) **constant** 166
[kPDFDisplayBoxBleedBox](#) **constant** 166
[kPDFDisplayBoxCropBox](#) **constant** 166
[kPDFDisplayBoxMediaBox](#) **constant** 166
[kPDFDisplayBoxTrimBox](#) **constant** 166
[kPDFDisplaySinglePage](#) **constant** 217
[kPDFDisplaySinglePageContinuous](#) **constant** 217
[kPDFDisplayTwoUp](#) **constant** 217
[kPDFDisplayTwoUpContinuous](#) **constant** 217
[kPDFIconArea](#) **constant** 218
[kPDFLineStyleCircle](#) **constant** 80
[kPDFLineStyleClosedArrow](#) **constant** 80
[kPDFLineStyleDiamond](#) **constant** 80
[kPDFLineStyleNone](#) **constant** 80
[kPDFLineStyleOpenArrow](#) **constant** 80
[kPDFLineStyleSquare](#) **constant** 80
[kPDFLinkArea](#) **constant** 218
[kPDFMarkupTypeHighlight](#) **constant** 87
[kPDFMarkupTypeStrikeOut](#) **constant** 87
[kPDFMarkupTypeUnderline](#) **constant** 87
[kPDFNoArea](#) **constant** 217
[kPDFPageArea](#) **constant** 218
[kPDFPopupArea](#) **constant** 218
[kPDFPrintPageScaleDownToFit](#) **constant** 140
[kPDFPrintPageScaleNone](#) **constant** 140
[kPDFPrintPageScaleToFit](#) **constant** 140
[kPDFTextAnnotationIconComment](#) **constant** 97
[kPDFTextAnnotationIconHelp](#) **constant** 98
[kPDFTextAnnotationIconInsert](#) **constant** 98
[kPDFTextAnnotationIconKey](#) **constant** 97
[kPDFTextAnnotationIconNewParagraph](#) **constant** 98
[kPDFTextAnnotationIconNote](#) **constant** 97
[kPDFTextAnnotationIconParagraph](#) **constant** 98
[kPDFTextArea](#) **constant** 218
[kPDFTextFieldArea](#) **constant** 218
[kPDFWidgetCheckBoxControl](#) **constant** 57
[kPDFWidgetPushButtonControl](#) **constant** 57
[kPDFWidgetRadioButtonControl](#) **constant** 57
[kPDFWidgetUnknownControl](#) **constant** 56

L

[label](#) **instance method** 150, 160
[labelFont](#) **instance method** 178
[layoutDocumentView](#) **instance method** 204
[lineWidth](#) **instance method** 109

M

[majorVersion](#) **instance method** 128
[markupType](#) **instance method** 86
[maxLength](#) **instance method** 102
[maximumNumberOfColumns](#) **instance method** 178
[minorVersion](#) **instance method** 128
[modificationDate](#) **instance method** 38
[mouseUpAction](#) **instance method** 38

N

[name](#) **instance method** 18, 93
[Named Action Names](#) 19
[numberOfCharacters](#) **instance method** 160
[numberOfChildren](#) **instance method** 150

O

[onStateValue](#) **instance method** 52
[outlineItemForSelection:](#) **instance method** 129
[outlineRoot](#) **instance method** 129

P

[page](#) **instance method** 39, 115
[pageAtIndex:](#) **instance method** 130
[pageClass](#) **instance method** 130
[pageCount](#) **instance method** 130
[pageForPoint:nearest:](#) **instance method** 204
[pageIndex](#) **instance method** 22
[pages](#) **instance method** 172
[parent](#) **instance method** 151
[parentID](#) **instance method** 52
[paths](#) **instance method** 74
[PDF Page Scaling Modes for Printing](#) 140
[PDFDocumentAuthorAttribute](#) **constant** 139
[PDFDocumentCreationDateAttribute](#) **constant** 139
[PDFDocumentCreatorAttribute](#) **constant** 139

PDFDocumentDidBeginFindNotification **notification** [140](#)

PDFDocumentDidBeginPageFindNotification **notification** [141](#)

PDFDocumentDidBeginPageWriteNotification **notification** [142](#)

PDFDocumentDidBeginWriteNotification **notification** [142](#)

PDFDocumentDidEndFindNotification **notification** [141](#)

PDFDocumentDidEndPageFindNotification **notification** [141](#)

PDFDocumentDidEndPageWriteNotification **notification** [142](#)

PDFDocumentDidEndWriteNotification **notification** [142](#)

PDFDocumentDidFindMatchNotification **notification** [142](#)

PDFDocumentDidUnlockNotification **notification** [140](#)

PDFDocumentKeywordsAttribute **constant** [139](#)

PDFDocumentModificationDateAttribute **constant** [139](#)

PDFDocumentProducerAttribute **constant** [139](#)

PDFDocumentSubjectAttribute **constant** [139](#)

PDFDocumentTitleAttribute **constant** [139](#)

PDFPrintScalingMode **data type** [138](#)

PDFView **instance method** [178](#)

PDFViewAnnotationHitNotification **notification** [219](#)

PDFViewAnnotationWillHitNotification **notification** [220](#)

PDFViewChangedHistoryNotification **notification** [218](#)

PDFViewCopyPermissionNotification **notification** [219](#)

PDFViewDisplayBoxChangedNotification **notification** [221](#)

PDFViewDisplayModeChangedNotification **notification** [220](#)

PDFViewDocumentChangedNotification **notification** [219](#)

PDFViewOpenPDF:forRemoteGoToAction:<NSObject> **delegate method** [214](#)

PDFViewPageChangedNotification **notification** [219](#)

PDFViewPerformFind: <NSObject> **delegate method** [215](#)

PDFViewPerformGoToPage: <NSObject> **delegate method** [215](#)

PDFViewPerformPrint: <NSObject> **delegate method** [216](#)

PDFViewPrintJobTitle: <NSObject> **delegate method** [216](#)

PDFViewPrintPermissionNotification **notification** [220](#)

PDFViewScaleChangedNotification **notification** [219](#)

PDFViewSelectionChangedNotification **notification** [220](#)

PDFViewWillChangeScaleFactor:toScale:<NSObject> **delegate method** [216](#)

PDFViewWillClickOnLink:withURL:<NSObject> **delegate method** [217](#)

performAction: **instance method** [204](#)

point **instance method** [23](#), [115](#)

popup **instance method** [39](#)

printWithInfo:autoRotate: **instance method** [205](#)

printWithInfo:autoRotate:pageScaling: **instance method** [205](#)

Q

quadrilateralPoints **instance method** [86](#)

R

removeAnnotation: **instance method** [161](#)

removeBezierPath: **instance method** [74](#)

removeFromParent **instance method** [151](#)

removePageAtIndex: **instance method** [131](#)

rotation **instance method** [103](#), [161](#)

rowSizeForPage: **instance method** [205](#)

S

scaleFactor **instance method** [206](#)

scrollSelectionToVisible: **instance method** [206](#)

selectAll: **instance method** [206](#)

selectedPages **instance method** [179](#)

selectionForEntireDocument **instance method** [131](#)

selectionForLineAtPoint: **instance method** [161](#)

selectionForRange: **instance method** [162](#)

selectionForRect: **instance method** [162](#)

selectionForWordAtPoint: **instance method** [163](#)

selectionFromPage:atCharacterIndex:toPage:atCharacterIndex: **instance method** [131](#)

selectionFromPage:atPoint:toPage:atPoint: **instance method** [132](#)

selectionFromPoint:toPoint: **instance method** [163](#)

selectionsByLine **instance method** [172](#)

setAction: **instance method** [151](#)

setAlignment: **instance method** [71](#), [103](#)

setAllowsDragging: **instance method** [179](#), [207](#)

setAllowsMultipleSelection: instance method 179
 setAutoScales: instance method 207
 setBackgroundColor: instance method 52, 62, 103, 180, 207
 setBorder: instance method 40
 setBounds: instance method 40
 setBounds:forBox: instance method 164
 setCaption: instance method 53
 setChoices: instance method 62
 setColor: instance method 40, 172
 setContents: instance method 41
 setControlType: instance method 53
 setCurrentSelection: instance method 208
 setCurrentSelection:animate: instance method 208
 setCursorForAreaOfInterest: instance method 209
 setDashPattern: instance method 109
 setDelegate: instance method 132, 209
 setDestination instance method 16
 setDestination: instance method 82, 152
 setDisplayBox: instance method 209
 setDisplayMode: instance method 210
 setDisplaysAnnotations: instance method 164
 setDisplaysAsBook: instance method 210
 setDisplaysPageBreaks: instance method 210
 setDocumentAttributes: instance method 133
 setDocument: instance method 211
 setEndLineStyle: instance method 77
 setEndPoint: instance method 77
 setFieldName: instance method 53, 63, 104
 setFields: instance method 27
 setFieldsIncludedAreCleared: instance method 27
 setFont: instance method 54, 63, 71, 104
 setFontColor: instance method 54, 63, 71, 105
 setGreekingThreshold: instance method 211
 setHighlighted: instance method 54, 82
 setHighlightedSelections: instance method 211
 setHorizontalCornerRadius: instance method 109
 setIconType: instance method 96
 setInteriorColor: instance method 68, 78, 92
 setIsListChoice: instance method 64
 setIsOpen: instance method 90, 152
 setLabel: instance method 152
 setLabelFont: instance method 180
 setLineWidth: instance method 110
 setMarkupType: instance method 86
 setMaximumLength: instance method 105
 setMaximumNumberOfColumns: instance method 181
 setModificationDate: instance method 41
 setMouseUpAction: instance method 42
 setName: instance method 18, 94
 setOnStateValue: instance method 55
 setOutlineRoot: instance method 133

setPageIndex: instance method 23
 setPDFView: instance method 181
 setPoint: instance method 23
 setPopUp: instance method 42
 setQuadrilateralPoints: instance method 87
 setRotation: instance method 105, 164
 setScaleFactor: instance method 212
 setShouldAntiAlias: instance method 212
 setShouldDisplay: instance method 42
 setShouldPrint: instance method 43
 setStartLineStyle: instance method 78
 setStartPoint: instance method 79
 setState: instance method 55
 setStringValue: instance method 64, 106
 setStyle: instance method 110
 setThumbnailSize: instance method 181
 setURL: instance method 24, 30, 83
 setUsername: instance method 43
 setVerticalCornerRadius: instance method 110
 setWindowIsOpen: instance method 96
 shouldAntiAlias instance method 212
 shouldDisplay instance method 44
 shouldPrint instance method 44
 startLineStyle instance method 79
 startPoint instance method 79
 state instance method 56
 string instance method 133, 165, 173
 stringValue instance method 65, 106
 style instance method 111

T

takeBackgroundColorFrom: instance method 213
 takePasswordFrom: instance method 213
 thumbnailSize instance method 182
 tooltip instance method 44
 transformContextForBox: instance method 165
 type instance method 14, 45
 Types of PDF Annotation Buttons 56

U

unlockWithPassword: instance method 134
 URL instance method 24, 30, 83
 userName instance method 45

V

verticalCornerRadius instance method 111

visiblePages [instance method](#) [213](#)

W

windowIsOpen [instance method](#) [97](#)

writeToFile: [instance method](#) [134](#)

writeToFile:withOptions: [instance method](#) [135](#)

writeToURL: [instance method](#) [135](#)

writeToURL:withOptions: [instance method](#) [136](#)

Z

zoomIn: [instance method](#) [214](#)

zoomOut: [instance method](#) [214](#)