
PDFDocument Class Reference

[Graphics & Imaging](#) > Cocoa





Apple Inc.
© 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, Cocoa, Mac, Mac OS, Pages, and Quartz 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

PDFDocument Class Reference 5

Overview	5
Tasks	5
Initializing Documents	5
Accessing Document Information	5
Managing Document Security	6
Writing Out the PDF Data	6
Working with Pages	7
Managing Find Operations	7
Working with Selections	7
Setting the Delegate	8
Searching Documents	8
Unlocking Documents	8
Determining the Page Class	8
Instance Methods	8
allowsCopying	8
allowsPrinting	9
beginFindString:withOptions:	9
beginFindStrings:withOptions:	10
cancelFindString	10
dataRepresentation	10
delegate	11
documentAttributes	11
documentURL	11
exchangePageAtIndex:withPageAtIndex:	12
findString:fromSelection:withOptions:	12
findString:withOptions:	13
indexOfPage:	13
initWithData:	14
initWithURL:	14
insertPage:atIndex:	14
isEncrypted	15
isFinding	15
isLocked	16
majorVersion	16
minorVersion	16
outlineItemForSelection:	17
outlineRoot	17
pageAtIndex:	18
pageClass	18
pageCount	18

- removePageAtIndex: 19
- selectionForEntireDocument 19
- selectionFromPage:atCharacterIndex:toPage:atCharacterIndex: 19
- selectionFromPage:atPoint:toPage:atPoint: 20
- setDelegate: 20
- setDocumentAttributes: 21
- setOutlineRoot: 21
- string 21
- unlockWithPassword: 22
- writeToFile: 22
- writeToFile:withOptions: 23
- writeToURL: 23
- writeToURL:withOptions: 24
- Delegate Methods 24
 - didMatchString: 24
 - documentDidBeginDocumentFind: 24
 - documentDidBeginPageFind: 25
 - documentDidEndDocumentFind: 25
 - documentDidEndPageFind: 25
 - documentDidFindMatch: 26
 - documentDidUnlock: 26
- Constants 26
 - PDFPrintScalingMode 26
 - Document Attribute Keys 27
 - PDF Page Scaling Modes for Printing 28
- Notifications 28
 - PDFDocumentDidUnlockNotification 28
 - PDFDocumentDidBeginFindNotification 29
 - PDFDocumentDidEndFindNotification 29
 - PDFDocumentDidBeginPageFindNotification 29
 - PDFDocumentDidEndPageFindNotification 29
 - PDFDocumentDidFindMatchNotification 30
 - PDFDocumentDidBeginWriteNotification 30
 - PDFDocumentDidEndWriteNotification 30
 - PDFDocumentDidBeginPageWriteNotification 30
 - PDFDocumentDidEndPageWriteNotification 31

Document Revision History 33

Index 35

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	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 14)
Initializes a `PDFDocument` object with the passed-in data.
- `initWithURL:` (page 14)
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 11)
Returns the URL for the document.
- `majorVersion` (page 16)
Returns the major version of the document.

- [minorVersion](#) (page 16)
Returns the minor version of the document.
- [string](#) (page 21)
Returns a string representing the textual content for the entire document.
- [outlineItemForSelection:](#) (page 17)
Returns the most likely parent PDF outline object for the selection.
- [outlineRoot](#) (page 17)
Returns the root PDF outline object for the document.
- [documentAttributes](#) (page 11)
Returns a dictionary of document metadata.
- [setDocumentAttributes:](#) (page 21)
Sets the document attributes.
- [setOutlineRoot:](#) (page 21)
Sets the document's root outline to a PDF outline object.

Managing Document Security

- [isEncrypted](#) (page 15)
Returns a Boolean value specifying whether the document is encrypted.
- [isLocked](#) (page 16)
Returns a Boolean value indicating whether the document is locked.
- [unlockWithPassword:](#) (page 22)
Attempts to unlock an encrypted document.
- [allowsCopying](#) (page 8)
Returns a Boolean value indicating whether the document allows copying of content to the Pasteboard.
- [allowsPrinting](#) (page 9)
Returns a Boolean value indicating whether the document allows printing.

Writing Out the PDF Data

- [dataRepresentation](#) (page 10)
Returns a representation of the document as an `NSData` object.
- [writeToFile:](#) (page 22)
Writes the document to a file at the specified path.
- [writeToFile:withOptions:](#) (page 23)
Writes the document to a file at the specified path with the specified options.
- [writeToURL:](#) (page 23)
Writes the document to a location specified by the passed-in URL.
- [writeToURL:withOptions:](#) (page 24)
Writes the document to the specified URL with the specified options.

Working with Pages

- [pageCount](#) (page 18)
Returns the number of pages in the document.
- [pageAtIndex:](#) (page 18)
Returns the page at the specified index number.
- [indexOfPage:](#) (page 13)
Gets the index number for the specified page.
- [insertPage:atIndex:](#) (page 14)
Inserts a page at the specified index point.
- [removePageAtIndex:](#) (page 19)
Removes the page at the specified index point.
- [exchangePageAtIndex:withPageAtIndex:](#) (page 12)
Swaps one page with another.

Managing Find Operations

- [findString:withOptions:](#) (page 13)
Synchronously finds all instances of the specified string in the document.
- [beginFindString:withOptions:](#) (page 9)
Asynchronously finds all instances of the specified string in the document.
- [beginFindStrings:withOptions:](#) (page 10)
Asynchronously finds all instances of the specified array of strings in the document.
- [findString:fromSelection:withOptions:](#) (page 12)
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 15)
Returns a Boolean value indicating whether an asynchronous find operation is in progress.
- [cancelFindString](#) (page 10)
Cancels a search initiated with [beginFindString:withOptions:](#) (page 9).

Working with Selections

- [selectionFromPage:atCharacterIndex:toPage:atCharacterIndex:](#) (page 19)
Returns the specified selection based on starting and ending character indexes.
- [selectionFromPage:atPoint:toPage:atPoint:](#) (page 20)
Returns the specified selection based on starting and ending points.
- [selectionForEntireDocument](#) (page 19)
Returns a selection representing the textual content of the entire document.

Setting the Delegate

- `setDelegate:` (page 20)
Establishes the specified object as the delegate for the `PDFDocument` object.
- `delegate` (page 11)
Returns the object acting as the delegate for the `PDFDocument` object.

Searching Documents

- `didMatchString:` (page 24) *delegate method*
Called for every match found during a find operation.
- `documentDidBeginDocumentFind:` (page 24) *delegate method*
Called when the `PDFDocumentDidBeginFindNotification` notification is posted.
- `documentDidBeginPageFind:` (page 25) *delegate method*
Called when the `PDFDocumentDidBeginPageFindNotification` notification is posted.
- `documentDidEndDocumentFind:` (page 25) *delegate method*
Called when the `PDFDocumentDidEndFindNotification` notification is posted.
- `documentDidEndPageFind:` (page 25) *delegate method*
Called when the `PDFDocumentDidEndPageFindNotification` notification is posted.
- `documentDidFindMatch:` (page 26) *delegate method*
Called when the `PDFDocumentDidFindMatchNotification` notification is posted.

Unlocking Documents

- `documentDidUnlock:` (page 26) *delegate method*
Called when the `PDFDocumentDidUnlockNotification` notification is posted.

Determining the Page Class

- `pageClass` (page 18)
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 13)
- [isFinding](#) (page 15)

- [cancelFindString](#) (page 10)

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:](#) (page \$@)
- [findString:withOptions:](#) (page 13)
- [isFinding](#) (page 15)
- [cancelFindString](#) (page 10)

Declared In

PDFDocument.h

cancelFindString

Cancels a search initiated with [beginFindString:withOptions:](#) (page 9).

```
- (void)cancelFindString
```

Availability

Available in Mac OS X v10.4 and later.

See Also

- [findString:withOptions:](#) (page 13)
- [beginFindString:withOptions:](#) (page 9)
- [isFinding](#) (page 15)

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 22)
- [writeToURL:](#) (page 23)

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 20)

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 26) 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 21)

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 18)
- [pageAtIndex:](#) (page 18)
- [indexForPage:](#) (page 13)
- [insertPage:atIndex:](#) (page 14)
- [removePageAtIndex:](#) (page 19)

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 15)
- [findString:withOptions:](#) (page 13)

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 9). For options, refer to [Searching and Comparing Strings](#).

Availability

Available in Mac OS X v10.4 and later.

See Also

- [isFinding](#) (page 15)
- [findString:fromSelection:withOptions:](#) (page 12)

Declared In

PDFDocument.h

indexOfPage:

Gets the index number for the specified page.

```
- (NSInteger)indexOfPage:(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 18)
- [pageAtIndex:](#) (page 18)
- [insertPage:atIndex:](#) (page 14)
- [removePageAtIndex:](#) (page 19)
- [exchangePageAtIndex:withPageAtIndex:](#) (page 12)

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 14)

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 14)

Declared In

`PDFDocument.h`

insertPage:atIndex:

Inserts a page at the specified index point.

```
- (void)insertPage:(PDFPage *)page atIndex:(NSInteger)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:` (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 18)
- [pageAtIndex:](#) (page 18)
- [indexForPage:](#) (page 13)
- [removePageAtIndex:](#) (page 19)
- [exchangePageAtIndex:withPageAtIndex:](#) (page 12)

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 22) 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 9)
- [cancelFindString](#) (page 10)

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 22) 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 16)

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 16)

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 17)

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 17)

- [setOutlineRoot:](#) (page \$@)

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 18)
- [indexForPage:](#) (page 13)
- [insertPageAtIndex:](#) (page 14)
- [removePageAtIndex:](#) (page 19)
- [exchangePageAtIndex:withPageAtIndex:](#) (page 12)

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 18)
- [indexForPage:](#) (page 13)

- [insertPageAtIndex:](#) (page 14)
- [removePageAtIndex:](#) (page 19)
- [exchangePageAtIndex:withPageAtIndex:](#) (page 12)

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 18)
- [pageAtIndex:](#) (page 18)
- [indexForPage:](#) (page 13)
- [insertPageAtIndex:](#) (page 14)
- [exchangePageAtIndex:withPageAtIndex:](#) (page 12)

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`) within the respective `PDFPage` objects.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [selectionFromPage:atPoint:toPage:atPoint:](#) (page 20)

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 19)
- `selectionForRange:`

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 11)
- `didMatchString:` (page 24)

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 26) for a list of possible key words.

Availability

Available in Mac OS X v10.4 and later.

See Also- [documentAttributes](#) (page 11).**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](#) (page \$@)**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` 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 10)
- [writeToURL:](#) (page 23)
- [writeToURL:withOptions:](#) (page 24)
- [writeToFile:withOptions:](#) (page 23)

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 10)
- [writeToURL:](#) (page 23)
- [writeToURL:withOptions:](#) (page 24)
- [writeToFile:](#) (page 22)

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 10)
- [writeToFile:](#) (page 22)
- [writeToFile:withOptions:](#) (page 23)
- [writeToURL:withOptions:](#) (page 24)

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 10)
- [writeToURL:](#) (page 23)
- [writeToFile:](#) (page 22)
- [writeToFile:withOptions:](#) (page 23)

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 13)
- [setDelegate:](#) (page 20)

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](#) (page \$@)
- [setDelegate:](#) (page 20)

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](#) (page \$@)

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

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](#) (page \$@)

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

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](#) (page \$@)

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

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](#) (page \$@)

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

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](#) (page \$@)

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

Declared In

`PDFDocument.h`

Constants

PDFPrintScalingMode

The type of scaling to be used when printing a page (see “[PDF Page Scaling Modes for Printing](#)” (page \$@)).

```
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 11) and [setDocumentAttributes:](#) (page 21).

```
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:.`

```
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 22) 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 9) or [findString:withOptions:](#) (page 13) 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 9) or [findString:withOptions:](#) (page 13) method returns.

The [beginFindString:withOptions:](#) (page 9) 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

Document Revision History

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

Date	Notes
2007-12-11	Added more information about attribute data types, and added information about how PDF viewing applications determine document size.
2007-10-31	Updated for Mac OS X v10.5.
2006-05-23	First publication of this content as a separate document.

REVISION HISTORY

Document Revision History

Index

A

allowsCopying [instance method 8](#)
allowsPrinting [instance method 9](#)

B

beginFindString:withOptions: [instance method 9](#)
beginFindStrings:withOptions: [instance method 10](#)

C

cancelFindString [instance method 10](#)

D

dataRepresentation [instance method 10](#)
delegate [instance method 11](#)
didMatchString: <NSObject> [delegate method 24](#)
Document Attribute Keys 27
documentAttributes [instance method 11](#)
documentDidBeginDocumentFind: <NSObject> [delegate method 24](#)
documentDidBeginPageFind: <NSObject> [delegate method 25](#)
documentDidEndDocumentFind: <NSObject> [delegate method 25](#)
documentDidEndPageFind: <NSObject> [delegate method 25](#)
documentDidFindMatch: <NSObject> [delegate method 26](#)
documentDidUnlock: <NSObject> [delegate method 26](#)
documentURL [instance method 11](#)

E

exchangePageAtIndex:withPageAtIndex: [instance method 12](#)

F

findString:fromSelection:withOptions: [instance method 12](#)
findString:withOptions: [instance method 13](#)

I

indexOfPage: [instance method 13](#)
initWithData: [instance method 14](#)
initWithURL: [instance method 14](#)
insertPage:atIndex: [instance method 14](#)
isEncrypted [instance method 15](#)
isFinding [instance method 15](#)
isLocked [instance method 16](#)

K

kPDFPrintPageScaleDownToFit [constant 28](#)
kPDFPrintPageScaleNone [constant 28](#)
kPDFPrintPageScaleToFit [constant 28](#)

M

majorVersion [instance method 16](#)
minorVersion [instance method 16](#)

O

outlineItemForSelection: **instance method** 17
 outlineRoot **instance method** 17

P

pageAtIndex: **instance method** 18
 pageClass **instance method** 18
 pageCount **instance method** 18
PDF Page Scaling Modes for Printing 28
 PDFDocumentAuthorAttribute **constant** 27
 PDFDocumentCreationDateAttribute **constant** 27
 PDFDocumentCreatorAttribute **constant** 27
 PDFDocumentDidBeginFindNotification **notification** 29
 PDFDocumentDidBeginPageFindNotification **notification** 29
 PDFDocumentDidBeginPageWriteNotification **notification** 30
 PDFDocumentDidBeginWriteNotification **notification** 30
 PDFDocumentDidEndFindNotification **notification** 29
 PDFDocumentDidEndPageFindNotification **notification** 29
 PDFDocumentDidEndPageWriteNotification **notification** 31
 PDFDocumentDidEndWriteNotification **notification** 30
 PDFDocumentDidFindMatchNotification **notification** 30
 PDFDocumentDidUnlockNotification **notification** 28
 PDFDocumentKeywordsAttribute **constant** 28
 PDFDocumentModificationDateAttribute **constant** 27
 PDFDocumentProducerAttribute **constant** 27
 PDFDocumentSubjectAttribute **constant** 27
 PDFDocumentTitleAttribute **constant** 27
 PDFPrintScalingMode **data type** 26

R

removePageAtIndex: **instance method** 19

S

selectionForEntireDocument **instance method** 19

selectionFromPage:atCharacterIndex:toPage:
 atCharacterIndex: **instance method** 19
 selectionFromPage:atPoint:toPage:atPoint:
 instance method 20
 setDelegate: **instance method** 20
 setDocumentAttributes: **instance method** 21
 setOutlineRoot: **instance method** 21
 string **instance method** 21

U

unlockWithPassword: **instance method** 22

W

writeToFile: **instance method** 22
 writeToFile:withOptions: **instance method** 23
 writeToURL: **instance method** 23
 writeToURL:withOptions: **instance method** 24