PDFPage Class Reference

Graphics & Imaging > Cocoa

2007-12-11

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

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

PDFPage 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	PDFPage.h

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 12)
Initializer for subclasses of PDFPage. (Deprecated. Use `[PDFPage init]` or `initWithImage:` (page $@) instead.)

`– initWithImage:` (page 12)
Creates a new `PDFPage` object and initializes it with the specified `NSImage` object.

Getting Information About a Page

- `document` (page 11)
  Returns the `PDFDocument` object with which the page is associated.
- `label` (page 12)
  Returns the label for the page.
- `boundsForBox:` (page 9)
  Returns the bounds for the specified PDF display box.
- `setBounds:forBox:` (page 16)
  Sets the bounds for the specified box.
- `rotation` (page 13)
  Returns the page rotation angle in degrees.
- `setRotation:` (page 17)
  Sets the rotation angle for the page in degrees.

Working with Annotations

- `annotations` (page 8)
  Returns an array containing the page's annotations.
- `displaysAnnotations` (page 10)
  Returns a Boolean value indicating whether annotations are displayed for the page.
- `setDisplaysAnnotations:` (page 16)
  Specifies whether or not to display annotations for the page.
- `addAnnotation:` (page 7)
  Adds the specified annotation object to the page.
- `removeAnnotation:` (page 13)
  Removes the specified annotation from the page.
- `annotationAtPoint:` (page 8)
  Returns the annotation, if there is one, at the specified point.

Rendering Pages

- `drawWithBox:` (page 11)
  Draws the page within the specified box.
- `transformContextForBox:` (page 17)
  Transforms the current context, given the specified box.

Working with Textual Content

- `numberOfCharacters` (page 12)
  Returns the number of characters on the page, including whitespace characters.
- `string` (page 17)
  Returns an `NSString` object representing the text on the page.

- `attributedString` (page 9)
  Returns an `NSAttributedString` object representing the text on the page.
- `characterBoundsAtIndex:` (page 9)
  Returns the bounds, in page space, of the character at the specified index.
- `characterIndexAtPoint:` (page 10)
  Returns the character index value for the specified point in page space.

Working with Selections

- `selectionForRect:` (page 14)
  Returns the text enclosed within the specified rectangle, expressed in page (user) coordinates.
- `selectionForWordAtPoint:` (page 15)
  Returns the whole word that includes the specified point.
- `selectionForLineAtPoint:` (page 14)
  Returns the whole line of text that includes the specified point.
- `selectionFromPoint:toPoint:` (page 15)
  Returns the text between the two specified points in page space.
- `selectionForRange:` (page 14)
  Returns the text contained within the specified range.

Miscellaneous

- `dataRepresentation` (page 10)
  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 8)
- `displaysAnnotations` (page 10)
- `setDisplaysAnnotations:` (page 16)
- `removeAnnotation:` (page 13)
- `annotationAtPoint:` (page 8)

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 8)
- `displaysAnnotations` (page 10)
- `setDisplaysAnnotations:` (page 16)
- `addAnnotation:` (page 7)
- `removeAnnotation:` (page 13)

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 10)
- `setDisplaysAnnotations:` (page 16)
- `addAnnotation:` (page 7)
- `removeAnnotation:` (page 13)
- `annotationAtPoint:` (page 8)

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 12)
- string (page 17)

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 18) 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 includes 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
- setBounds:forBox: (page $@)

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 10)

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 9)

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 8)
- `setDisplaysAnnotations:` (page 16)
- `addAnnotation:` (page 7)
- `removeAnnotation:` (page 13)
- `annotationAtPoint:` (page 8)
- `drawWithBox:` (page 11)

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 10)

Declared In
`PDFPage.h`

initWithDocument:

Initializer for subclasses of PDFPage. (**Deprecated.** Use `[PDFPage init]` or `initWithImage:` (page $@) 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 "xxi", "4-1" and so on).

Availability

Available in Mac OS X v10.4 and later.

See Also

`- document` (page 11)

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 17)

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 8)
- `displaysAnnotations` (page 10)
- `setDisplaysAnnotations:` (page 16)
- `addAnnotation:` (page 7)
- `annotationAtPoint:` (page 8)

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 17)

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 14)
`– selectionForWordAtPoint:` (page 15)
`– selectionFromPoint:toPoint:` (page 15)
`– selectionForRange:` (page 14)

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 14)
`– selectionForWordAtPoint:` (page 15)
`– selectionForLineAtPoint:` (page 14)
`– selectionFromPoint:toPoint:` (page 15)

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 15)
`- selectionForLineAtPoint:` (page 14)
`- selectionFromPoint:toPoint:` (page 15)
`- selectionForRange:` (page 14)

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 14)
`- selectionForLineAtPoint:` (page 14)
`- selectionFromPoint:toPoint:` (page 15)
`- selectionForRange:` (page 14)

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 14)
`- selectionForWordAtPoint:` (page 15)
`- selectionForLineAtPoint:` (page 14)
`- selectionForRange:` (page 14)

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 9)

Declared In

`PDFPage.h`

setDisplaysAnnotations:

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 8)
`- displaysAnnotations` (page 10)
`- addAnnotation:` (page 7)
`- removeAnnotation:` (page 13)
`- annotationAtPoint:` (page 8)

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 13)

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 12)
- attributedString (page 9)

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: method or from within a draw method of a PDFAnnotation subclass.

Availability

Available in Mac OS X v10.5 and later.

See Also

- drawWithBox: (page $@)

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
kPDFDisplayBoxMediaBox	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 PDFPage.h.
kPDFDisplayBoxCropBox	A rectangle defining the boundaries of the visible region , expressed in default user-space units. Default value equal to kPDFDisplayBoxMediaBox. Available in Mac OS X v10.4 and later. Declared in PDFPage.h.
kPDFDisplayBoxBleedBox	A rectangle defining the boundaries of the clip region for the page contents in a production environment. Default value equal to kPDFDisplayBoxCropBox. Available in Mac OS X v10.4 and later. Declared in PDFPage.h.
kPDFDisplayBoxTrimBox	A rectangle defining the intended boundaries of the finished page. Default value equal to kPDFDisplayBoxCropBox. Available in Mac OS X v10.4 and later. Declared in PDFPage.h.
kPDFDisplayBoxArtBox	A rectangle defining the boundaries of the page's meaningful content including surrounding white space intended for display. Default value equal to kPDFDisplayBoxCropBox. Available in Mac OS X v10.4 and later. Declared in PDFPage.h.

Document Revision History

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

Date	Notes
2007-12-11	Made minor corrections.
2007-10-31	Updated for Mac OS X v10.5.
2006-05-23	First publication of this content as a separate document.

Index