

**FastPdfKit**  
**1.0RC1**

Generated by Doxygen 1.7.3

Mon Mar 7 2011 11:08:12



# Contents

<b>1</b>	<b>Class Index</b>	<b>1</b>
1.1	Class Hierarchy . . . . .	1
<b>2</b>	<b>Class Index</b>	<b>3</b>
2.1	Class List . . . . .	3
<b>3</b>	<b>Class Documentation</b>	<b>5</b>
3.1	MFDocumentManager Class Reference . . . . .	5
3.1.1	Member Function Documentation . . . . .	6
3.1.1.1	createImageForThumbnailOfPageNumber:ofSize:andScale: . . . . .	6
3.1.1.2	documentManagerWithFilePath: . . . . .	6
3.1.1.3	emptyCache . . . . .	6
3.1.1.4	initWithFileUrl: . . . . .	6
3.1.1.5	isLocked . . . . .	6
3.1.1.6	numberOfPages . . . . .	6
3.1.1.7	outline . . . . .	6
3.1.1.8	searchResultOnPage:forSearchTerms:withProfile: . . . . .	7
3.1.1.9	tryUnlockWithPassword: . . . . .	7
3.1.1.10	wholeTextForPage:withProfile: . . . . .	7
3.1.2	Property Documentation . . . . .	7
3.1.2.1	defaultProfile . . . . .	7
3.2	<MFDocumentOverlayDataSource> Protocol Reference . . . . .	7
3.2.1	Member Function Documentation . . . . .	8
3.2.1.1	documentViewController:drawablesForPage: . . . . .	8
3.3	MFDocumentViewController Class Reference . . . . .	8
3.3.1	Member Function Documentation . . . . .	9
3.3.1.1	automodeOnRotation . . . . .	9
3.3.1.2	autozoomOnPageChange . . . . .	9
3.3.1.3	cleanUp . . . . .	9
3.3.1.4	direction . . . . .	9
3.3.1.5	lead . . . . .	9
3.3.1.6	mode . . . . .	9
3.3.1.7	moveToNextPage . . . . .	9
3.3.1.8	moveToPreviousPage . . . . .	9
3.3.1.9	page . . . . .	9
3.3.1.10	setAutomodeOnRotation: . . . . .	10
3.3.1.11	setAutozoomOnPageChange: . . . . .	10
3.3.1.12	setDirection: . . . . .	10
3.3.1.13	setLead: . . . . .	10

3.3.1.14	setMode:	10
3.3.1.15	setPage:	10
3.3.1.16	setPage:withZoomOfLevel:onRect:	10
3.3.2	Property Documentation	11
3.3.2.1	legacyModeEnabled	11
3.3.2.2	overlayEnabled	11
3.3.2.3	pageFlipOnEdgeTouchEnabled	11
3.3.2.4	startingPage	11
3.3.2.5	zoomInOnDoubleTapEnabled	11
3.4	<MFDocumentViewControllerDelegate> Protocol Reference	11
3.4.1	Member Function Documentation	12
3.4.1.1	documentViewController:didChangeDirectionTo:	12
3.4.1.2	documentViewController:didChangeLeadTo:	12
3.4.1.3	documentViewController:didChangeModeTo:automatic:	12
3.4.1.4	documentViewController:didEndZoomingAtScale:	12
3.4.1.5	documentViewController:didFocusOnPage:	12
3.4.1.6	documentViewController:didGoToPage:	12
3.4.1.7	documentViewController:didReceiveTapAtPoint:	12
3.4.1.8	documentViewController:didReceiveTapOnPage:atPoint:	13
3.4.1.9	documentViewController:didReceiveURLRequest:	13
3.4.1.10	documentViewController:willFocusOnPage:	13
3.4.1.11	documentViewControllerDidUnfocus:	13
3.5	<MFOverlayDrawable> Protocol Reference	13
3.5.1	Member Function Documentation	14
3.5.1.1	drawInContext:	14
3.6	<MFOverlayTouchable> Protocol Reference	14
3.6.1	Member Function Documentation	14
3.6.1.1	containsPoint:	14
3.7	MFPDFOutlineEntry Class Reference	14
3.7.1	Property Documentation	14
3.7.1.1	bookmarks	14
3.7.1.2	indentation	15
3.7.1.3	pageNumber	15
3.7.1.4	title	15
3.8	MFTextItem Class Reference	15
3.8.1	Member Function Documentation	15
3.8.1.1	initWithText:andHighlightPath:	15
3.8.1.2	initWithText:highlightPath:andPage:	16
3.8.2	Property Documentation	16
3.8.2.1	highlightPath	16
3.8.2.2	page	16
3.8.2.3	text	16

# Chapter 1

## Class Index

### 1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

MFDocumentManager . . . . .	5
<MFDocumentOverlayDataSource> . . . . .	7
MFDocumentViewController . . . . .	8
<MFDocumentViewControllerDelegate> . . . . .	11
<MFOverlayDrawable> . . . . .	13
MFTextItem . . . . .	15
<MFOverlayTouchable> . . . . .	14
MFPDFOutlineEntry . . . . .	14



## Chapter 2

# Class Index

### 2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">MFDocumentManager</a>	5
<a href="#">&lt;MFDocumentOverlayDataSource&gt;</a>	7
<a href="#">MFDocumentViewController</a>	8
<a href="#">&lt;MFDocumentViewControllerDelegate&gt;</a>	11
<a href="#">&lt;MFOverlayDrawable&gt;</a>	13
<a href="#">&lt;MFOverlayTouchable&gt;</a>	14
<a href="#">MFPDFOutlineEntry</a>	14
<a href="#">MFTextItem</a>	15





## Chapter 3

# Class Documentation

### 3.1 MFDocumentManager Class Reference

#### Public Member Functions

- (CGImageRef) - **createImageFromPDFPagesLeft:andRight:size:andScale:useLegacy:**
- (CGImageRef) - **createImageFromPDFPage:size:andScale:useLegacy:**
- (void) - **drawPageNumber:onContext:**
- (void) - **getCropbox:andRotation:forPageNumber:**
- (CGImageRef) - **createImageForThumbnailOfPageNumber:ofSize:andScale:**
- (NSMutableArray \*) - **outline**
- (id) - **initWithFileUrl:**
- (BOOL) - **isLocked**
- (BOOL) - **tryUnlockWithPassword:**
- (NSUInteger) - **numberOfPages**
- (void) - **emptyCache**
- (NSArray \*) - **searchResultOnPage:forSearchTerms:withProfile:**
- (NSString \*) - **wholeTextForPage:withProfile:**

#### Static Public Member Functions

- (MFDocumentManager \*) + **documentManagerWithFilePath:**

#### Protected Attributes

- MFOffscreenRenderer \* **renderer**
- CGPDFDocumentRef **document**
- NSLock \* **lock**
- NSURL \* **url**
- NSUInteger **numberOfPages**

- NSString \* **password**
- NSLock \* **pageDataLock**
- int \* **dataSetFlags**
- CGRect \* **cropboxes**
- int \* **rotations**

## Properties

- MFProfile [defaultProfile](#)

### 3.1.1 Member Function Documentation

**3.1.1.1 - (CGImageRef) createImageForThumbnailOfPageNumber: dummy(NSUInteger) pageNr  
ofSize:(CGSize) size andScale:(CGFloat) scale**

Create a thumbnail for a specific page. It will look far better than the thumbnail integrated inside the pdf, but it is also slower.

**3.1.1.2 + (MFDocumentManager \*) documentManagerWithFilePath: dummy(NSString \*)  
filePath**

Factory method to create an [MFDocumentManager](#) instance from a know file path.

**3.1.1.3 - (void) emptyCache**

Clear the page cache. It is important to call this method on memory warning as in the sample code to prevent the application being killed right for excessive memory usage.

**3.1.1.4 - (id) initWithFileUrl: dummy(NSURL \*) anUrl**

Initializer. You can also use the factory method above.

**3.1.1.5 - (BOOL) isLocked**

Check if a document is encrypted and blocked by a password or not.

**3.1.1.6 - (NSUInteger) numberOfPages**

Return the number of pages that make up the document.

**3.1.1.7 - (NSMutableArray \*) outline**

Return an array of MFOutlineEntry as the outline/TOC of the pdf document.

**3.1.1.8** - (NSArray \*) `searchResultOnPage: dummy(NSUInteger) pageNr  
forSearchTerms:(NSString *) searchTerm withProfile:(MFProfile *) p`

Return an array of [MFTextItem](#) representing the matches of the search term on the page passed as arguments. It is a good choice running this method in a secondary thread. Pass NULL as profile to use default search profile. Profile is not retained, so be sure to keep it in memory until the function returns.

**3.1.1.9** - (BOOL) `tryUnlockWithPassword: dummy(NSString *) aPassword`

Try to unlock the document with a password and return if the unlock has been successful or not.

**3.1.1.10** - (NSString \*) `wholeTextForPage: dummy(NSUInteger) pageNr withProfile:(MFProfile *)  
p`

Return a string representation of the text contained in a pdf page. Profile is not retained, so be sure to keep it in memory until the function returns. You can pass NULL to use the default profile.

## 3.1.2 Property Documentation

**3.1.2.1** - (MFProfile) `defaultProfile` [read, write, assign]

This is an experimental feature. It will allow to customize the behaviour for search and extraction of text. You can set the values inside of this struct before launching a search or a text extraction action. Look at `mfprofile.h` for an explanation of the `MFProfile` struct and how to customize it. This is the default profile used as fallback when a NULL profile is passed to the search and extraction methods.

The documentation for this class was generated from the following file:

- `Classes/MFDocumentManager.h`

## 3.2 <MFDocumentOverlayDataSource> Protocol Reference

### Public Member Functions

- (NSArray \*) - `documentViewController:drawablesForPage:`

### 3.2.1 Member Function Documentation

3.2.1.1 - (NSArray \*) **documentViewController:** dummy(MFDocumentViewController \*)  
*dvc* drawablesForPage:(NSUInteger) *page* [optional]

This method is invoked when a new detail page is going to be drawn and `overlayEnabled` of the [MFDocumentViewController](#) is set to YES. The object setted as `overlayDataSource` is then required to return an array of [MFOverlayDrawable](#) object to be drawn on the page as overlay.

The documentation for this protocol was generated from the following file:

- `Classes/MFDocumentOverlayDataSource.h`

## 3.3 MFDocumentViewController Class Reference

### Public Member Functions

- (id) - **initWithDocumentManager:**
- (BOOL) - [automodeOnRotation](#)
- (void) - [setAutomodeOnRotation:](#)
- (void) - [setMode:](#)
- (MFDocumentMode) - [mode](#)
- (void) - [setPage:](#)
- (void) - [setPage:withZoomOfLevel:onRect:](#)
- (NSUInteger) - [page](#)
- (void) - [setLead:](#)
- (MFDocumentLead) - [lead](#)
- (void) - [setDirection:](#)
- (MFDocumentDirection) - [direction](#)
- (void) - [setAutozoomOnPageChange:](#)
- (BOOL) - [autozoomOnPageChange](#)
- (void) - [moveToNextPage](#)
- (void) - [moveToPreviousPage](#)
- (void) - [cleanUp](#)

### Properties

- NSObject< [MFDocumentOverlayDataSource](#) > \* **overlayDataSource**
- NSObject< [MFDocumentViewControllerDelegate](#) > \* **documentDelegate**
- [MFDocumentManager](#) \* **document**
- NSUInteger [startingPage](#)
- BOOL [pageFlipOnEdgeTouchEnabled](#)
- BOOL [zoomInOnDoubleTapEnabled](#)
- BOOL **documentInteractionEnabled**
- BOOL [overlayEnabled](#)
- BOOL [legacyModeEnabled](#)

### 3.3.1 Member Function Documentation

#### 3.3.1.1 - (BOOL) automodeOnRotation

This method enable or disable the automatic mode switching upon rotation. If enabled, the page mode will be automatically changed to single page in portrait and side-by-side (double) on landscape. Setting the mode manually will disable the automode.

#### 3.3.1.2 - (BOOL) autozoomOnPageChange

Returns whether the autozoom feature is enabled or not.

#### 3.3.1.3 - (void) cleanUp

Call this method right after dismissing this [MFDocumentViewController](#) instance. It will release all the resources and stop the background threads. Once this method has been called, the [MFDocumentViewController](#) instance cannot be considered valid anymore and should be released.

#### 3.3.1.4 - (MFDocumentDirection) direction

Return the current direction used by the document.

#### 3.3.1.5 - (MFDocumentLead) lead

Returns the current lead used when presenting the document.

#### 3.3.1.6 - (MFDocumentMode) mode

Returns the current mode used to display the document.

#### 3.3.1.7 - (void) moveToNextPage

This method will begin an animated transition to the next page, if available.

#### 3.3.1.8 - (void) moveToPreviousPage

This method will begin an animated transition to the previous page, if available.

#### 3.3.1.9 - (NSInteger) page

Returns the current page of the document.

**3.3.1.10 - (void) setAutomodeOnRotation: dummy(BOOL) automode**

Returns whether automode is enabled or not.

**3.3.1.11 - (void) setAutozoomOnPageChange: dummy(BOOL) autozoom**

This method will turn on or off the autozoom feature. If on, the current zoom level will be kept between pages, otherwise will be rest to 100% on page change.

**3.3.1.12 - (void) setDirection: dummy(MFDocumentDirection) newDirection**

This method is used to set the page reading direction: left to right or right to left.

**3.3.1.13 - (void) setLead: dummy(MFDocumentLead) newLead**

This method will set the lead used to present the pages in side-by-side (double) mode. With MFDocumentLeadLeft, the cover will appear on the left side in side-by-side mode, whereas with MFDocumentLeadRight will appear on the right side. Use this method to keep pairing between pages for books and magazines. Single page mode is not affected by this setting.

**3.3.1.14 - (void) setMode: dummy(MFDocumentMode) newMode**

Set how the pages are presented to the user. MFDocumentModeSingle present a single page to the user, centered on the screen. MFDocumentModeDouble present two pages side-by-side, as they would appear on a magazine or a books. This will allow to preserve content split between the pages, for example a large background image.

**3.3.1.15 - (void) setPage: dummy(NSUInteger) page**

This metod will set the current page of the document and jump to the specified page. Current page is used to determine bookmarks position. On side-by-side (double) mode, it is usually the left-most page of the two.

**3.3.1.16 - (void) setPage: dummy(NSUInteger) page withZoomOfLevel:(float) zoomLevel  
onRect:(CGRect) rect**

This metod will set the current page of the document and jump to the specified page, while trying to zoom in on the specified rect

### 3.3.2 Property Documentation

#### 3.3.2.1 -(BOOL) legacyModeEnabled [read, write, assign]

Enabled or force the legacy mode, or let the app choose to enable it or not depending on the device. Default is disabled.

#### 3.3.2.2 -(BOOL) overlayEnabled [read, write, assign]

Enable or disable the display of overlay item over the document. Default is disabled.

#### 3.3.2.3 -(BOOL) pageFlipOnEdgeTouchEnabled [read, write, assign]

Enable the page flip when the user touch the edges of the screen.

#### 3.3.2.4 -(NSInteger) startingPage [read, write, assign]

Set the starting page of the document. It is valid only after initialization and before the view is displayed on the screen. Typically you want to set this just after the init of the viewController. Default is 1.

#### 3.3.2.5 -(BOOL) zoomInOnDoubleTapEnabled [read, write, assign]

Enabled the zoom in when the user double tap on the screen.

The documentation for this class was generated from the following file:

- Classes/MFDocumentViewController.h

## 3.4 <MFDocumentViewControllerDelegate> Protocol Reference

### Public Member Functions

- (void) - **documentViewController:didChangeDirectionTo:**
- (void) - [documentViewController:didGoToPage:](#)
- (void) - [documentViewController:didChangeLeadTo:](#)
- (void) - [documentViewController:didChangeModeTo:automatic:](#)
- (void) - [documentViewController:didChangeDirectionTo:](#)
- (void) - [documentViewController:didReceiveTapAtPoint:](#)
- (void) - [documentViewController:didReceiveURLRequest:](#)
- (void) - [documentViewController:didReceiveTapOnPage:atPoint:](#)
- (void) - [documentViewController:didEndZoomingAtScale:](#)
- (void) - [documentViewController:willFocusOnPage:](#)
- (void) - [documentViewController:didFocusOnPage:](#)
- (void) - [documentViewControllerDidUnfocus:](#)

### 3.4.1 Member Function Documentation

3.4.1.1 - (void) documentViewController: dummy(MFDocumentViewController \*) *dvc*  
didChangeDirectionTo:(MFDocumentDirection) *direction* [optional]

This method will notify a change in the direction used to present the document.

3.4.1.2 - (void) documentViewController: dummy(MFDocumentViewController \*) *dvc*  
didChangeLeadTo:(MFDocumentLead) *lead* [optional]

This method will notify a change in the lead used to present the document.

3.4.1.3 - (void) documentViewController: dummy(MFDocumentViewController \*)  
*dvc* didChangeModeTo:(MFDocumentMode) *mode* automatic:(BOOL) *automatically*  
[optional]

This method will notify a change in the mode of the document, either by explicitly setting it or automatic on rotation.

3.4.1.4 - (void) documentViewController: dummy(MFDocumentViewController \*) *dvc*  
didEndZoomingAtScale:(float) *level* [optional]

This method will report the last zoom level achieved by the document detail view. You can use this callback to animate an icon that report the current zoom to the user.

3.4.1.5 - (void) documentViewController: dummy(MFDocumentViewController \*) *dvc*  
didFocusOnPage:(NSInteger) *page* [optional]

This method will be called upon the showing up of the high definition version of the current page. Could be used to stop and Activity Indicator.

3.4.1.6 - (void) documentViewController: dummy(MFDocumentViewController \*) *dvc*  
didGoToPage:(NSInteger) *page* [optional]

This method will be called to notify the transition to a new page. Use this to update page number related UI's elements or synchronize selected actions.

3.4.1.7 - (void) documentViewController: dummy(MFDocumentViewController \*) *dvc*  
didReceiveTapAtPoint:(CGPoint) *point* [optional]

This method will notify if the user has tapped the document view at a point different from a document element, like an annotation.



**3.4.1.8** - (void) documentViewController: dummy(MFDocumentViewController \*) dvc  
didReceiveTapOnPage:(NSInteger) page atPoint:(CGPoint) point [optional]

This method will notify if and where the user has tapped on a page bounds. Coordinates of the point are in document's user space.

**3.4.1.9** - (void) documentViewController: dummy(MFDocumentViewController \*) dvc  
didReceiveURLRequest:(NSString \*) uri [optional]

This method will notify if the user has tapped on a annotation with a remote uri action. This is usually invoked when an external link is activated and an internet browser should be open to show the link's content.

**3.4.1.10** - (void) documentViewController: dummy(MFDocumentViewController \*) dvc  
willFocusOnPage:(NSInteger) page [optional]

This method will be called right before displaying a high definition version of the current page. Could be used to start an Activity Indicator.

**3.4.1.11** - (void) documentViewControllerDidUnfocus:  
dummy(MFDocumentViewController \*) dvc  
[optional]

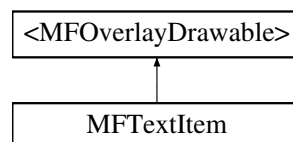
This method will be called when the high definition version of the page is removed from the view.

The documentation for this protocol was generated from the following file:

- Classes/MFDocumentViewControllerDelegate.h

## 3.5 <MFOverlayDrawable> Protocol Reference

Inheritance diagram for <MFOverlayDrawable>:



### Public Member Functions

- (void) - [drawInContext:](#)

### 3.5.1 Member Function Documentation

#### 3.5.1.1 - (void) drawInContext: dummy(CGContextRef) *context*

Implement this method to perform drawing in an overlay over the document view. The context coordinate system is aligned to the user space of the document displayed.

The documentation for this protocol was generated from the following file:

- Classes/MFOverlayDrawable.h

## 3.6 <MFOverlayTouchable> Protocol Reference

### Public Member Functions

- (BOOL) - [containsPoint](#):

### 3.6.1 Member Function Documentation

#### 3.6.1.1 - (BOOL) containsPoint: dummy(CGPoint) *point*

Implement this method to perform a hit test. The CGPoint coordinates are defined in document user space.

The documentation for this protocol was generated from the following file:

- Classes/MFOverlayTouchable.h

## 3.7 MFPDFOutlineEntry Class Reference

### Properties

- NSInteger [indentation](#)
- NSUInteger [pageNumber](#)
- NSArray \* [bookmarks](#)
- NSString \* [title](#)

### 3.7.1 Property Documentation

#### 3.7.1.1 - (NSArray \*) [bookmarks](#) [read, write, retain]

Child entries.

**3.7.1.2** `-(NSInteger) indentation` [read, write, assign]

Indentation level of the outline entry. It is also the node level inside the outline tree.

**3.7.1.3** `-(NSInteger) pageNumber` [read, write, assign]

Page number of the entry.

**3.7.1.4** `-(NSString *) title` [read, write, copy]

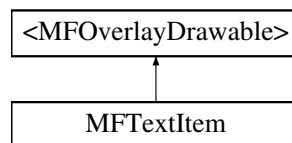
Title for the outline entry.

The documentation for this class was generated from the following file:

- Classes/MFPDFOutlineEntry.h

## 3.8 MFTextItem Class Reference

Inheritance diagram for MFTextItem:



### Public Member Functions

- (id) - [initWithText:andHighlightPath:](#)
- (id) - [initWithText:highlightPath:andPage:](#)

### Properties

- NSRange **searchTermRange**
- NSString \* [text](#)
- CGPathRef [highlightPath](#)
- NSInteger [page](#)

### 3.8.1 Member Function Documentation

**3.8.1.1** `-(id) initWithText: dummy(NSString *) someText andHighlightPath:(CGPathRef) aPath`

Default initializer. Init the Text Item with some text and a path for the highlight that will be rendered in page space.

**3.8.1.2** - (id) initWithText: dummy(NSString \*) *someText* highlightPath:(CGPathRef) *aPath* andPage:(NSUInteger) *aPage*

Default initializer, plus the page number.

## 3.8.2 Property Documentation

**3.8.2.1** - (CGPathRef) highlightPath [read, assign]

The path for the hilight. It is defined in page space.

**3.8.2.2** - (NSUInteger) page [read, assign]

The page of which this text item represent the position of a word.

**3.8.2.3** - (NSString \*) text [read, assign]

Some text to be displayed along with the item.

The documentation for this class was generated from the following file:

- Classes/MFTextItem.h