FastPdfKit

2.0.1

Generated by Doxygen 1.7.4

Wed Aug 10 2011 09:41:51

Contents

Chapter 1

Class Index

1.1 Class Hierarchy

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

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

Class Documentation

3.1 <FPKOverlayViewDataSource> Protocol Reference

Public Member Functions

- (NSSet ∗) - documentViewController:overlayViewsForPage:
- (CGRect) - documentViewController:rectForOverlayView:
- (void) - documentViewController:willAddOverlayView:
- (void) - **documentViewController:didAddOverlayView:**
- (void) - **documentViewController:willRemoveOverlayView:**
- (void) - **documentViewController:didRemoveOverlayView:**

3.1.1 Member Function Documentation

3.1.1.1 - (NSSet ∗) documentViewController: dummy(MFDocumentViewController ∗) *dvc* overlayViewsForPage:(NSUInteger) *page* [optional]

This method shall return a set of view to display over the pdf page.

3.1.1.2 - (CGRect) documentViewController: dummy(MFDocumentViewController ∗) *dvc* rectForOverlayView:(UIView ∗) *view* [optional]

This method needs to return the frame in page-coordinates for the view passed as arguments. Remember that, like drawables and touchables, the coordinate system's origin is in the bottom left corner of the page.

3.1.1.3 - (void) documentViewController: dummy(MFDocumentViewController ∗) *dvc* willAddOverlayView:(UIView ∗) *view* [optional]

These callbacks will be invoked when the overlay view is going to be added, after is added, when is going to be removed and when it is actually removed from the page

view. Use these to change the status of the view and or start/stop any action that needs to be synchronized with the lifecycle of the view.

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

- FastPdfKitLibrary/FPKOverlayViewDataSource.h

3.2 <MFAudioPlayerViewProtocol> Protocol Reference

Public Member Functions

- (void) - setAudioProvider:
- (void) - audioProviderDidStart:
- (void) - **audioProviderDidStop:**
- (void) - **audioProvider:volumeAdjustedTo:**

Static Public Member Functions

- (UIView *) + audioPlayerViewInstance

3.2.1 Member Function Documentation

3.2.1.1 + (UIView *) audioPlayerViewInstance

This method will be called to provide a view instance to add to the overlay.

3.2.1.2 - (void) audioProviderDidStart: dummy(id< MFAudioProvider >) *mfeap*

Playback and status event methods.

3.2.1.3 - (void) setAudioProvider: dummy(id< MFAudioProvider >) *provider*

This method will be called to give a chance to store the refernce to the provider that will notify the view about the playback. It is recommended to assign it, not retain to avoid circular retention between provider and review.

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

- FastPdfKitLibrary/MFAudioPlayerViewProtocol.h

3.3 <MFAudioProvider> Protocol Reference

Public Member Functions

- (void) - togglePlay
- (BOOL) - isPlaying
- (void) - setVolumeLevel:
- (float) - volumeLevel

3.3.1 Member Function Documentation

3.3.1.1 - (BOOL) isPlaying

Tell if the audio clip is playing.

3.3.1.2 - (void) setVolumeLevel: dummy(float) *volume*

Set the volume level, from 0.0 to 1.0.

3.3.1.3 - (void) togglePlay

Play when stopped and viceversa.

3.3.1.4 - (float) volumeLevel

Return the volume level.

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

- FastPdfKitLibrary/MFAudioProvider.h

3.4 MFDocumentManager Class Reference

Public Member Functions

- (CGImageRef) - **createImageFromPDFPagesLeft:andRight:size:andScale:useLegacy:**
- (CGImageRef) - **createImageFromPDFPage:size:andScale:useLegacy:**
- (CGImageRef) - **createImageFromPDFPagesLeft:andRight:size:andScale:useLegacy:showShadow:andPadding:**
- (CGImageRef) - **createImageFromPDFPage:size:andScale:useLegacy:showShadow:andPadding:**
- (void) - **drawPageNumber:onContext:**
- (void) - getCropbox:andRotation:forPageNumber:
- (CGImageRef) - createImageForThumbnailOfPageNumber:ofSize:andScale:
- (NSMutableArray ∗) - outline

- (id) - initWithFileUrl:
- (id) - initWithDataProvider:
- (BOOL) - isLocked
- (BOOL) - tryUnlockWithPassword:
- (NSUInteger) - numberOfPages
- (NSUInteger) - pageNumberForDestinationNamed:
- (void) - emptyCache
- (NSArray ∗) - searchResultOnPage:forSearchTerms:withProfile:
- (NSArray ∗) - test_searchResultOnPage:forSearchTerms:
- (NSString ∗) - wholeTextForPage:withProfile:
- (NSString ∗) - test_wholeTextForPage:

Static Public Member Functions

- (MFDocumentManager ∗) + documentManagerWithFilePath:
- (NSString ∗) + version

Protected Attributes

- MFOffscreenRenderer ∗ **renderer**
- CGPDFDocumentRef **document**
- NSLock ∗ **lock**
- NSURL ∗ **url**
- CGDataProviderRef **provider**
- NSUInteger **numberOfPages**
- NSString ∗ **password**
- NSLock ∗ **pageDataLock**
- int ∗ **dataSetFlags**
- CGRect ∗ **cropboxes**
- int ∗ **rotations**

Properties

- MFProfile defaultProfile
- NSString ∗ resourceFolder

3.4.1 Member Function Documentation

3.4.1.1 - (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.

3.4.1.2 + (MFDocumentManager ∗) documentManagerWithFilePath: dummy(NSString ∗) *filePath*

Factory method to create an MFDocumentManager instance from a know file path.

3.4.1.3 - (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.

3.4.1.4 - (void) getCropbox: dummy(CGRect ∗) *cropbox* andRotation:(int ∗) *rotation* forPageNumber:(NSInteger) *pageNumber*

Use this method to get the cropbox and the rotation of a certain pdf page.

3.4.1.5 - (id) initWithDataProvider: dummy(CGDataProviderRef) *dataProvider*

Initializer with data provider.

3.4.1.6 - (id) initWithFileUrl: dummy(NSURL ∗) *anUrl*

Initializer. You can also use the factory method above.

3.4.1.7 - (BOOL) isLocked

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

3.4.1.8 - (NSUInteger) numberOfPages

Return the number of pages that make up the document.

3.4.1.9 - (NSMutableArray ∗) outline

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

3.4.1.10 - (NSUInteger) pageNumberForDestinationNamed: dummy(NSString ∗) *name*

This method will return the page number of the destination with the name passed as argument.

3.4.1.11 - (NSArray ∗) searchResultOnPage: dummy(NSUInteger) *pageNr* forSearchTerms:(NSString ∗) *searchTerm* withProfile:(MFProfile ∗) *p*

Return an array of MFTextItem representing the matches of teh 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.

3.4.1.12 - (NSArray ∗) test_searchResultOnPage: dummy(NSUInteger) *pageNr* forSearchTerms:(NSString ∗) *searchTerm*

This method return the same result as the above, but uses a differnt search engine. Look at the readme for instructions.

3.4.1.13 - (NSString ∗) test_wholeTextForPage: dummy(NSUInteger) *pageNr*

This method return the same result as the above, but uses a differnt extraction engine. Look at the readme for instructions.

3.4.1.14 - (BOOL) tryUnlockWithPassword: dummy(NSString ∗) *aPassword*

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

3.4.1.15 + (NSString ∗) version

Build version of this library. Useful for debugging purposes.

3.4.1.16 - (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.

3.4.2 Property Documentation

3.4.2.1 - (MFProfile) defaultProfile [read, write, assign]

This is an experimental features. 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.

3.4.2.2 - (NSString *) resourceFolder [read, write, retain]

Resouce folder for the document. Video, audio and other files referenced in the pdf are contained here.

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

- FastPdfKitLibrary/MFDocumentManager.h

3.5 <MFDocumentOverlayDataSource> Protocol Reference

Public Member Functions

- (NSArray *) - documentViewController:drawablesForPage:
- (NSArray *) - documentViewController:touchablesForPage:
- (void) - documentViewController:didReceiveTapOnTouchable:
- (NSArray *) - **documentViewController:overlayViewsForPage:**
- (void) - **documentViewController:willRemoveOverlayView:**
- (void) - **documentViewController:didRemoveOverlayView:**
- (void) - **documentViewController:willAddOverlayView:**
- (void) - **documentViewController:didAddOverlayView:**

3.5.1 Member Function Documentation

3.5.1.1 - (void) documentViewController: dummy(MFDocumentViewController *) *dvc* didReceiveTapOnTouchable:(id< MFOverlayTouchable >) *touchable* [optional]

This method will be called when an user does tap on an overlay touchable element.

3.5.1.2 - (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.

3.5.1.3 - (NSArray *) documentViewController: dummy(MFDocumentViewController *) *dvc* touchablesForPage:(NSUInteger) *page* [optional]

This method is invoked when a new detail page is going to be drawn and overlayEnables is set to YES. Objects added as overlay data sources are required to submit touchables element to be tested against user input events.

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

- FastPdfKitLibrary/MFDocumentOverlayDataSource.h

3.6 MFDocumentViewController Class Reference

Public Member Functions

- (void) - addOverlayDataSource:
- (void) - **removeOverlayDataSource:**
- (void) - addOverlayViewDataSource:
- (void) - **removeOverlayViewDataSource:**
- (void) - reloadOverlay
- (float) - zoomLevelForAnnotationRect:ofPage:
- (float) - zoomScale
- (CGPoint) - zoomOffset
- (NSUInteger) - leftPage
- (NSUInteger) - rightPage
- (void) - setEdgeFlipWidth:
- (CGFloat) - **edgeFlipWidth**
- (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
- (CGPoint) - convertPoint:fromViewtoPage:
- (CGPoint) - convertPoint:toViewFromPage:
- (CGRect) - convertRect:fromViewToPage:
- (CGRect) - convertRect:toViewFromPage:
- (CGPoint) - convertPoint:fromOverlayToPage:
- (CGPoint) - convertPoint:toOverlayFromPage:
- (CGRect) - convertRect:fromOverlayToPage:
- (CGRect) - convertRect:toOverlayFromPage:

Properties

- NSObject< MFDocumentViewControllerDelegate > * **documentDelegate**
- MFDocumentManager * **document**
- BOOL directionalLockEnabled
- BOOL useTiledOverlayView
- BOOL showHorizontalScroller
- BOOL showShadow
- CGFloat padding
- NSUInteger startingPage
- BOOL pageFlipOnEdgeTouchEnabled
- BOOL zoomInOnDoubleTapEnabled
- BOOL **documentInteractionEnabled**
- BOOL overlayEnabled
- BOOL legacyModeEnabled

3.6.1 Member Function Documentation

3.6.1.1 - (void) addOverlayDataSource: dummy(id< MFDocumentOverlayDataSource >) *ods*

Add and remove an Overlay Datasource for Drawables and Touchables.

3.6.1.2 - (void) addOverlayViewDataSource: dummy(id< FPKOverlayViewDataSource >) *ovds*

Add and remove an Overlay View Datasource for overlay UIViews.

3.6.1.3 - (BOOL) automodeOnRotation

This metod 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.

3.6.1.4 - (BOOL) autozoomOnPageChange

Returns whether the autozoom feature is enabled or not.

3.6.1.5 - (void) cleanUp

Call this method rightly 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.

3.6.1.6 - (CGPoint) convertPoint: dummy(CGPoint) *point* fromOverlayToPage:(NSUInteger) *page*

Convert a point from overlay space (the whole view that hold the both left and right page, and that you can zoom in and scroll over) to page space.

3.6.1.7 - (CGPoint) convertPoint: dummy(CGPoint) *point* fromViewtoPage:(NSUInteger) *page*

Convert a point from MFDocumentViewController's view space to page space.

3.6.1.8 - (CGPoint) convertPoint: dummy(CGPoint) *point* toOverlayFromPage:(NSUInteger) *page*

Convert a point from page space to overlay space.

3.6.1.9 - (CGPoint) convertPoint: dummy(CGPoint) *point* toViewFromPage:(NSUInteger) *page*

Convert a point from page space to MFDocumentViewController's view space.

3.6.1.10 - (CGRect) convertRect: dummy(CGRect) *rect* fromOverlayToPage:(NSUInteger) *page*

Convert a rect from overlay space to page space.

3.6.1.11 - (CGRect) convertRect: dummy(CGRect) *rect* fromViewToPage:(NSUInteger) *page*

Convert a rect from MFDocumentViewController's view space to page space.

3.6.1.12 - (CGRect) convertRect: dummy(CGRect) *rect* toOverlayFromPage:(NSUInteger) *page*

Convert a ract from page to overlay space.

3.6.1.13 - (CGRect) convertRect: dummy(CGRect) *rect* toViewFromPage:(NSUInteger) *page*

Convert a rect from page space to MFDocumentViewController's view space.

3.6.1.14 - (MFDocumentDirection) direction

Return the current direction used by the document.

3.6.1.15 - (MFDocumentLead) lead

Returns the current lead used when presenting the document.

3.6.1.16 - (NSUInteger) leftPage

This method will return the page number of the left page displayed. If the mode is single page, the left page number is the current page.

3.6.1.17 - (MFDocumentMode) mode

Returns the current mode used to display the document.

3.6.1.18 - (void) moveToNextPage

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

3.6.1.19 - (void) moveToPreviousPage

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

3.6.1.20 - (NSUInteger) page

Returns the current page of the document.

3.6.1.21 - (void) reloadOverlay

This method will provoke the redraw of the overlay. Overlay Datasources will be asked to provide drawables.

3.6.1.22 - (NSUInteger) rightPage

This method will return the page number of the right page displayed. If the mode is single, right page number is invalid.

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

Returns whether automode is enabled or not.

3.6.1.24 - (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.

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

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

3.6.1.26 - (void) setEdgeFlipWidth: dummy(CGFloat) *edgeFlipWidth*

Set and get the percentage of the screen associated with the page flip on edge touch action. Default value is 0.1, this mean that the 10% of the width of the screen on either side will receive such events. Values are clipped between 0.0 and 0.5 to prevent overlap.

3.6.1.27 - (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.

3.6.1.28 - (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.

3.6.1.29 - (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.

3.6.1.30 - (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. Pass 0.0 as zoomLevel to let the application try to calculate the appropriate zoom level to fit the rectangle on screen.

3.6.1.31 - (float) zoomLevelForAnnotationRect: dummy(CGRect) *rect* ofPage:(NSUInteger) *page*

This will return the appropriate zoom level to perfectly zoom onto an annotation. If return 0, there's no available page data to compute the zoom yet.

3.6.1.32 - (CGPoint) zoomOffset

Return the offset of the page scroll view.

3.6.1.33 - (float) zoomScale

Return the zoom scale of the page scroll view.

3.6.2 Property Documentation

3.6.2.1 - (BOOL) directionalLockEnabled [read, write, assign]

This property enable or disable the directional lock in the inner (document) scroll view. Default is NO.

3.6.2.2 - (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.

3.6.2.3 - (BOOL) overlayEnabled [read, write, assign]

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

3.6.2.4 - (CGFloat) padding [read, write, assign]

Set the amount of minimum padding between the pages and the screen edge. Default is 5.0. Values are clipped between 0 and 100.

3.6.2.5 - (BOOL) pageFlipOnEdgeTouchEnabled [read, write, assign]

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

3.6.2.6 - (BOOL) showHorizontalScroller [read, write, assign]

Use this property to hide or show the horizontal scroller under the pages.

3.6.2.7 - (BOOL) showShadow [read, write, assign]

Set this flag to NO if you don't want the dropdown shadow under the pages. Default is YES.

3.6.2.8 - (NSUInteger) 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. Tipically you want to set this just after the init of the viewController. Default is 1.

3.6.2.9 - (BOOL) useTiledOverlayView [read, write, assign]

This property will enable an CATiledLayer version of the overlay view. This means overlay drawables will be drawn sharp, no matter the zoom of the scroll view.

3.6.2.10 - (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:

- FastPdfKitLibrary/MFDocumentViewController.h

3.7 <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:didReceiveURIRequest:
- (void) - documentViewController:didReceiveTapOnPage:atPoint:
- (void) - documentViewController:didEndZoomingAtScale:
- (void) - documentViewController:willFocusOnPage:
- (void) - documentViewController:didFocusOnPage:
- (void) - documentViewControllerDidUnfocus:
- (void) - documentViewController:didReceiveDoubleTapOnAnnotationRect:withUri:onPage:
- (void) - documentViewController:didReceiveTapOnAnnotationRect:withUri:onPage:
- (BOOL) - documentViewController:doesHaveToAutoplayVideo:
- (BOOL) - documentViewController:doesHaveToAutoplayAudio:
- (void) - documentViewController:didReceiveRequestToGoToDestinationNamed:ofFile:
- (void) - documentViewController:didReceiveRequestToGoToPage:ofFile:
- (Class< MFAudioPlayerViewProtocol >) - classForAudioPlayerView

3.7.1 Member Function Documentation

3.7.1.1 - (Class<MFAudioPlayerViewProtocol>) classForAudioPlayerView [optional]

This method will be called to provide a class of the view to use as player audio control. You can use the default class provide in the sample or develop your own to suit the look and feel of you application better.

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

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

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

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

3.7.1.4 - (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 esplicity setting it or automatic on rotation.

3.7.1.5 - (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.

3.7.1.6 - (void) documentViewController: dummy(MFDocumentViewController *) *dvc* didFocusOnPage:(NSUInteger) *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.

3.7.1.7 - (void) documentViewController: dummy(MFDocumentViewController *) *dvc* didGoToPage:(NSUInteger) *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.

3.7.1.8 - (void) documentViewController: dummy(MFDocumentViewController *) *dvc* didReceiveDoubleTapOnAnnotationRect:(CGRect) *rect* withUri:(NSString *) *uri* onPage:(NSUInteger) *page* [optional]

This method will be called when user double tap on an annotation.

3.7.1.9 - (void) documentViewController: dummy(MFDocumentViewController ∗) *dvc* didReceiveRequestToGoToDestinationNamed:(NSString ∗) *destinationName* ofFile:(NSString ∗) *fileName* [optional]

This method will be invoked when the user tap on an annotation with an associated Go-To Remote action. The user can then load the file passed as third argument and then get the page number with MFDocumentManager's -pageNumberForDestinationNamed: and present the right page for display.

3.7.1.10 - (void) documentViewController: dummy(MFDocumentViewController ∗) *dvc* didReceiveRequestToGoToPage:(NSUInteger) *pageNumber* ofFile:(NSString ∗) *fileName* [optional]

This method will be invoked when the user tap on an annotation with an associated Go-To Remote action. The user can then load the file passed as last argument and then set the page to the page number passed as second parameter.

3.7.1.11 - (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.

3.7.1.12 - (void) documentViewController: dummy(MFDocumentViewController ∗) *dvc* didReceiveTapOnAnnotationRect:(CGRect) *rect* withUri:(NSString ∗) *uri* onPage:(NSUInteger) *page* [optional]

This method will be called when an user tap on an annotation.

3.7.1.13 - (void) documentViewController: dummy(MFDocumentViewController ∗) *dvc* didReceiveTapOnPage:(NSUInteger) *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.

3.7.1.14 - (void) documentViewController: dummy(MFDocumentViewController ∗) *dvc* didReceiveURIRequest:(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.

3.7.1.15 - (BOOL) documentViewController: dummy(MFDocumentViewController *) dvc doesHaveToAutoplayAudio:(NSString *) audioUri [optional]

Implement this method to return whether the audio clip should play automatically once loaded.

3.7.1.16 - (BOOL) documentViewController: dummy(MFDocumentViewController *) dvc doesHaveToAutoplayVideo:(NSString *) videoUri [optional]

This method will be called to ask the delegate if the video player needs to start automatically once visible.

3.7.1.17 - (void) documentViewController: dummy(MFDocumentViewController *) dvc willFocusOnPage:(NSUInteger) 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.

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

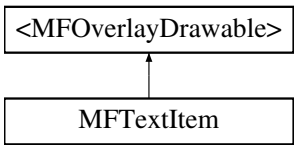
This method will be called when the high definition version of the page is removed from the view.

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

- FastPdfKitLibrary/MFDocumentViewControllerDelegate.h

3.8 <MFOverlayDrawable> Protocol Reference

Inheritance diagram for <MFOverlayDrawable>:



Public Member Functions

- (void) - drawInContext:

3.8.1 Member Function Documentation

3.8.1.1 - (void) drawInContext: dummy(CGContextRef) *context*

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

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

- FastPdfKitLibrary/MFOverlayDrawable.h

3.9 <MFOverlayTouchable> Protocol Reference

Public Member Functions

- (BOOL) - containsPoint:

3.9.1 Member Function Documentation

3.9.1.1 - (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:

- FastPdfKitLibrary/MFOverlayTouchable.h

3.10 MFPDFOutlineEntry Class Reference

Properties

- NSInteger indentation
- NSUInteger pageNumber
- NSArray * bookmarks
- NSString * title

3.10.1 Property Documentation

3.10.1.1 - (NSArray *) bookmarks `[read, write, retain]`

Child entries.

3.10.1.2 - (NSInteger) indentation [read, write, assign]

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

3.10.1.3 - (NSUInteger) pageNumber [read, write, assign]

Page number of the entry.

3.10.1.4 - (NSString ∗) title [read, write, copy]

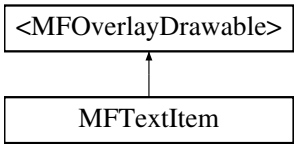
Title for the outline entry.

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

- FastPdfKitLibrary/MFPDFOutlineEntry.h

3.11 MFTextItem Class Reference

Inheritance diagram for MFTextItem:



Public Member Functions

- (id) - initWithText:andHighlightPath:
- (id) - initWithText:highlightPath:andPage:

Properties

- NSRange **searchTermRange**
- NSString ∗ text
- CGPathRef highlightPath
- NSUInteger page

3.11.1 Member Function Documentation

3.11.1.1 - (id) initWithText: dummy(NSString ∗) *someText* andHighlightPath:(CGPathRef) *aPath*

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

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

Default initializer, plus the page number.

3.11.2 Property Documentation

3.11.2.1 - (CGPathRef) highlightPath [read, assign]

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

3.11.2.2 - (NSUInteger) page [read, assign]

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

3.11.2.3 - (NSString ∗) text [read, assign]

Some text to be displayed along with the item.

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

- FastPdfKitLibrary/MFTextItem.h