

RAZPDF App Extension™

RAZPDFDocument Class Reference

Inherits from	NSObject
Framework	/System/Library/Frameworks/UIKit.framework /System/Library/Frameworks/MessageUI.framework
Availability	Printing available in iOS 4.2 and later Mailing available in iOS 3.0 and later
Declared in	razpdfdocument.h
Website	www.raz.com
Version	0.91
Tested in Xcode4	

RAZPDF © 2011 eCasa, Inc. All rights reserved.
iOS, Xcode and Interface Builder are trademarks of Apple Inc.
All other trademarks are the property of their respective owners.
App Extension is a trademark of eCasa, Inc.

Overview

The RAZPDF App Extension™ class is designed for use in iOS native application projects. RAZPDF renders a paper sized view into a local PDF File. Portrait and landscape sizes can be combined in a single PDF. The completed PDF file can be emailed, printed or shared easily using included RAZPDF methods to invoke systems mail, print or load features. You can also use the PDF file yourself in anyway that suits your application.

The RAZPDF addPage method adds one page per call to the PDF file. Once a PDFView is defined it may be reused for multiple pages. Dynamic data can be set before addPage is called. Control of the design is completely in the hands of the developer/designer.

The views are typically set up in a Interface Builder XIB using standard UIView, UIImageView and UIText components. Dynamic text and images can be added to PDFView via IBOutletlets just before rendering.

RAZPDF Features and benefits

- Fast local in-app PDF rendering
- Quick and easy implementation
- Generated PDFs can be easily archived, mailed, printed and opened with other apps.
- Allow designers to contribute on the project using Interface Builder
- Reduce tedious details

- No private Apple APIs used
- Fast PDF rendering, typically less than one second per page

Installation Instructions:

- unzip RAZPDF
- Add the key RAZPDF to your info-plist and set it to the key your were sent in separate email. (for paid versions, also verify Bundle id matches key purchased)

Example- Create a single page PDF and allow user to choose to mail, print, load generated file

Step 1: Add libRAZPDF.a and razpdfdocument.h to your project, add MessageUI.framework, and CoreGraphics.framework.

- **In Xcode4, Press apple-1, click on project, click on targets, the last group titled “Link binary with libraries”, press + to add frameworks.**

Step 2: Design your PDF layout with the help of Interface Builder

Step 3: Edit AppViewController.m

In .h of the view controller

```
IBOutlet UIView *pdfView; // Add outlet for Interface Builder
```

In .m of the view controller

```
// Create newTempDocument, addPage, shows action sheet
// then user can then choose mail, print, open or cancel
// PDF autodeletes when RAZPDFDocument is deallocated

#import "RAZPDFDocument.h"

-(IBAction) userWantsPDF {
    RAZPDFDocument *razPDF = [RAZPDFDocument tempDocument];
    [razPDF addPage:pdfView error:nil];
    [razPDF actionSheetInViewController:self];
}
```

Tasks

Create a RAZPDFDocument App Extension™ Instance

```
-(id)initWithPath:(NSString *)path error:(NSError **)error;
```

```
+ (id)tempDocument;
```

Add Page, Close and Delete File Methods

- (BOOL)[addPage:](#)(UIView *)PDFView
error:(NSError **)error;
- (BOOL)[closePDF:](#)(NSError **)error;
- (BOOL)[deletePDF:](#)(NSError **)error;

Presenting Action Sheet Modal View

- (BOOL)[actionSheetInViewController:](#)(UIViewController *)inParentViewController;

Presenting Mail Modal View

- (BOOL)[mailPDFInViewController:](#)(UIViewController *)inParentViewController
error:(NSError **)error;

Presenting Print Modal View

- (BOOL)[printPDFInViewController:](#)(UIViewController *)inParentViewController
error:(NSError **)error;

Presenting Load PDF (into 3rd Party PDF viewer) Modal View

- (BOOL)[LoadPDFInViewController:](#)(UIViewController *)inParentViewController
error:(NSError **)error;

Properties:

PDF Related Properties:

- [pdfAutodelete](#) Enables automatic removal of PDF file from the device
- [pdfWatermark](#) Enable RAZPDF watermark

Enterprise Version:

- [pdfTitle](#) The title of the the document.
- [pdfAuthor](#) The name of the person who created the document.
- [pdfSubject](#) The subject for the document
- [pdfOwnerPassword](#) PDF Owner Password.

- [pdfUserPassword](#) PDF User Password.
- [pdfAllowsPrinting](#) Whether the document allows printing when unlocked with the user password.
Default value = YES
- [pdfAllowsCopying](#) Whether the document allows copying when unlocked with the user password.
Default value = YES

Mail Related Properties

Common Mail Properties:

- [mailSubject](#) Sets subject line of the email.
- [mailToRecipients](#) Sets recipients to include in the email's "To" field.
- [mailCCRecipients](#) Sets recipients to include in the email's "Cc" field.
- [mailBCCRecipients](#) Sets recipients to include in the email's "Bcc" field.
- [mailMessageBody](#) The body text of the message. The text is interpreted as either plain text or HTML depending on the value of the isHTML property.
- [mailFilename](#) The file name of the attachment. Defaults to last component of full path used to initialize the object or "app.pdf" for tempDocument.

Enterprise Version:

- [isHTML](#) The body text of the message.
isHTML = NO // Plain text
isHTML = YES // HTML

UI Related Properties

Common UI Properties:

- [actionSheetTitle](#) Sets the title of the action sheet.
Default value = "Select Option"

Developer and Enterprise Version:

- [tintColor](#) The color of any navigation bars.
- [translucent](#) The translucency of any navigation bars.

Using Apple's Interface Builder

Apple's Interface Builder is a powerful developer tool for making custom views in applications and now with RAZPDF you can use the same tool to design full size PDF pages.

Open Xcode and selected the xib of your view controller, load it into the Interface Builder. When Xcode makes view controller with xib it adds a single UIView called "View". The easiest way to add a PDFView to your view controller to your xib.

Steps to making a PDFView in Interface Builder in xcode3

1) Open Xcode, click on your view controllers' xib (this will load Interface Builder)

2) Press **Apple+0** to load xib display tool, hit the second button, with 4 horizontal lines displays an expandable tree mode organizer for the xib. A default xib typically has "File's Owner", "First Responder", "View". This "View" is the one used for the screen by the app for the iPhone/iPad device.

- **In Xcode4 click on your view controller's .xib file, select a view and press what looks like a play bottom at the bottom to expand to tree view of all view layers.**

3) We want to make a new PDFView a view to represent a PDF page. Lets create it. Open the library with **Apple+L**, in the search box at the bottom enter "uiv" and the library will limit down the full list and when stop you can then see "UIView". Drag a copy of a new UIView and place it below the existing "View" in the organizer (the window activated by **Apple+0**).

- **In Xcode4 to get to the library, press option+apple+4, then in header of bottom section, press the 3rd icon for object. Then you can use search at bottom to filter for UILabel, UIImage, UIView.**

4) Now in the organizer you will have "File's Owner", "First Responder", "View", "View" with the second one selected. Lets name it. Press **Apple+4**. Near the bottom find the section "Interface Builder Identity", and you will see a blank place to enter Name. Enter PDFView (or what you prefer) and the second selected "View" will change to "PDFView". Naming help organize and link items to IBOutlet correctly.

- **In Xcode4 use option+apple+3 to access Custom Class properties. You can update Lable show under Identity.**

5) Select PDFView press **Apple+3** to bring up the PDFView Size, set width to 612 and height to 792. This will make a normal letter size portrait 8.5" wide and 11" high page. What was a generic UIView is now called PDFView and sized for the whole page.

- In Xcode4 use option+apple+5 to access size properties.

Note: International users might be interested in A4 size, use 595 x 842 points.

6) Developers/Designers may want to consider dragging out additional UIViews and placing them in the PDFView as a container to define sections and make borders. RAZPDF custom tag described below will allow the designer to set rounded corners on your additional UIViews. A UIView container can also have background color. Layering is preserved and UIViews can be stacked, the tree view comes in handy for seeing the layers.

Customized RAZPDF “Tag” values:

Special Tag values are picked up to further customize the output. The Tag values are integers. These tags only apply to RAZPDF PDFViews.

UIView Custom Tag Property Settings

	<p>Tag = xx, round all 4 corners, xx points.</p> <ul style="list-style-type: none"> • 25 would set all 4 corners to 25 points rounded corners.
	<p>Tag > 1,0xx, round bottom 2 corners, top corners at right angles</p> <ul style="list-style-type: none"> • 1025 would set bottom corners to 25 points rounded corners, top 2 will be square
	<p>Tag > 10,0xx, round top 2 corners, bottom corners at right angles</p> <ul style="list-style-type: none"> • 10025 would set top corners to 25 points rounded corners, bottom 2 will be square

7) Ready for more, great. So lets say your inner designer has put out several UIView containers for a header, a sidebar a footer; set backgrounds and named them all. Go ahead and set the Tag on a few for rounded corners. You can do a test addPage at anytime to see your progress. Now lets put some meat on those bones.

8) Lets add some text. Select the UIView named heading. Press **Apple+L** and find UILabel. Drag UILabel to tree view under heading. The default text will be “Label”. Lets set some of the **UILabel attributes**, press **Apple+1**.

- In Xcode4 use option+apple+4 to access attributes.

So lets take a look at whats available in Apple’s the bag of tools for us. If you just can seem to click on what you want select it from the tree view. Normally the first click on a UILabel give you resizing corners. Touch and hold in the middle to get a “hand” to drag it around. A second click highlights the area that the text can expand to fill. A third click lets you change the text character on-screen.

text

- The text displayed by the label.

textAlignment

- The technique to use for aligning the text.

textColor

- The color of the text.

adjustsFontSizeToFitWidth

- A Boolean value indicating whether the font size should be reduced in order to fit the title string into the label's bounding rectangle.

font

- The font of the text.
- RAZPDF has a sample webpage that displays the fonts on your devices. iPhone and iPad have different options (page best viewed from Safari on Mac/iPhone/iPad): <http://www.raz.com/fonts>

lineBreakMode

- The technique to use for wrapping and truncating the label's text.

minimumFontSize

- The size of the smallest permissible font with which to draw the label's text.

numberOfLines

- The maximum number of lines to use for rendering text.

shadowColor

- The shadow color of the text.

shadowOffset

- The shadow offset (measured in points) for the text.

textAlignment

- The technique to align the text.

textColor

- The color of the text.

9) Drag in a couple more UILabels and set some attributes. Make sure they are all tucked under UIView heading layer, and now for something amazing. Select the heading layer and drag it around, you should see all the labels on the layer reposition together.

10) Text is fun but lets get some photo action going. Go back to the library and pull out a UIImage and place on the PDFView. Add a png file to the project or use an exiting one. **Apple+1** to see UIImage attributes. Set image property to filename and resize as needed. Set Mode to control cropping and fill methods.

- **In Xcode4 use option+apple+4 to access attributes.**

image

- The image displayed in the image view.

11) We have a another piece of magic to use. The common underlying view properties of all of our objects. Common to all UIViews, UIImages and UILabels you can set these attributes to control appearance of the output.

mode

- Used to determine how a view layout content. Has many fill and cropping options.

alpha

- The receivers alpha/transparency value. Only affects current object.

background

- The receivers background color.

hidden

- A hidden view disappears (grays out) in Interface Builder and is not rendered into the PDF. Hiding a view with subviews has the effect of hiding those subviews and any view descendants they might have. This effect is implicit and does not alter the hidden state of the receiver's descendants.

12) Layers is fully supported. Drag out another label and place it over the image and set a contrasting color to add titles to photos. The the contrasting color will make the text will pop. The text will pop even more by enabling drop shadows.

Technical Points

- No private Apple APIs are used. PDF generation is done 100% on the device without the need for network connection. The rendering process is very fast because it is 100% native Objective-C.
- The precision of object placement is 1/72th of a inch, approximately 0.0139 inch or 0.3528 mm,
- The RAZPDF static library libRAZPDF.a is universal release build for device and simulator.
- Undefined symbol: "**_OBJC_CLASS_\$_RAZPDFDocument**" is a sign have not added **libRAZPDF.a** into your project.

- Developers must “clean” project after download of update to use a new version of RAZPDF files “**libRAZPDF.a**” or “**razpdfdocument.h**”. You can avoid some xcode complaints if you delete “libRAZPDF.a” and “razpdfdocument.h” from your project and add them from the new download. Then clean and rebuild.

Standard Type 1 Fonts

There are fourteen typefaces that have a special significance to PDF documents:

- Times (v3) (regular, italic, bold, bold italic)
- Courier (regular, oblique, bold, bold oblique)
- Helvetica (v3) (regular, oblique, bold, bold oblique)
- Symbol
- Zapf Dingbats

These fonts, sometimes referred to as the "base fourteen fonts" should always be present (actually present or a close substitute) and so need not be embedded in a PDF reducing size of output file. PDF viewers must know about the metrics of these fonts. Other fonts may be substituted if they are not embedded in a PDF.

Designers note that in Interface Builder all desktop fonts are shown, but they may not be present on you iPhone/iPad device. iPhone and iPad have different fonts sets. If a desktop font is selected and it is not on the device or reader it may be substituted with one available to the reader and may not render as expected.

Quotes

“PDFs are elegant.” - Derek L.

Tasks

Managing the PDF

initWithPath:

Init class with full path of working PDF file.

```
-(id)initWithPath:(NSString *)path error:(NSError **)error;
```

Discussion

Each addPage: call will add a page to the PDF File. The initial call will create the PDF file and leave it open. Each additional call will append a page to the PDF file. The appending process is stopped automatically when any method called needs to have PDF closed to proceed. Init() without options is not supported.

```
NSString *path;
path=[NSHomeDirectory() stringByAppendingPathComponent:@"Documents"];
[path stringByAppendingPathComponent:@"app.pdf"];
RAZPDFDocument *razPDF = [[RAZPDFDocument alloc] initWithPath:path
error:nil];
[razPDF addPage:pdfView error:nil];
// PDF Open to add more pages, see also ClosePDF and ActionSheet
```

Availability

Available in RAZPDF 1.0 and later.

Declared In

razpdfdocument.h

tempDocument:

Init class and prepares temp file called “**app.pdf**” in the temp area. The default is set to automatically

delete PDF on dealloc. This method is used by programs that do not need to retain PDF after it is processed.

```
+ (id)tempDocument;
```

Discussion

This method makes it very simple to prepare a temporary file.

```
// Sample code to create a new tempDocument without error checking  
  
RAZPDFDocument *razPDF = [RAZPDFDocument tempDocument];
```

Availability

Available in RAZPDF 1.0 and later.

Declared In

razpdfdocument.h

addPage:

Render the PDFView as a page to PDF file.

```
- (BOOL)addPage:(UIView *)PDFView  
    error:(NSError **)error;
```

Discussion

Each addPage: call will add a page to the PDF File. The initial call will create the PDF file and leave it open. Each additional call will append a page to the PDF file. The page size must remain the same from the first to last call. The appending process is stopped automatically when any method called needs to have PDF closed to proceed.

```
// Sample code to add couple of pages and then mail it  
  
[razPDF addPage:PDFView error:nil];  
[razPDF addPage:PDFView error:nil];  
[razPDF mailPDFInViewController:self];
```

See sample code and video for rendering dynamic data into PDF files using IBOutlet.

Availability

Available in RAZPDF 1.0 and later.

Declared In

razpdfdocument.h

closePDF:

Finalize PDF and closes file.

- (BOOL)closePDF;

Discussion

Finalizes PDF and closes file. The file will be at the location specified by the path and fileName properties. This would be used to close the file after adding pages to save PDF file at path inside app's sandbox.

```
// Sample code to closePDF
NSString *path;
path = [NSHomeDirectory() stringByAppendingPathComponent:@"Documents"];
path = [path stringByAppendingPathComponent:@"app.pdf"];
RAZPDFDocument *razPDF = [[RAZPDFDocument alloc] initWithPath:path
error:nil];
[razPDF addPage:pdfView error:nil];
[razPDF closePDF:nil]; // File is available for access on path
```

Availability

Available in RAZPDF 1.0 and later.

Declared In

razpdfdocument.h

deletePDF:

Deletes PDF file based on path and fileName properties.

- (BOOL)deletePDF:(NSError **)error;

Discussion

Deletes PDF file at the location specified by the path set when the object was created.

If an error occurs, upon return contains an NSError object that describes the problem. Pass nil if you do not want error information.

```
// Sample code to delete current PDF file  
  
[razPDF deletePDF:nil];
```

Availability

Available in RAZPDF 1.0 and later.

Declared In

razpdfdocument.h

actionSheetInViewController:

Displays standard action sheet view controller allowing user to select “Mail, Print, Load or Cancel”. Once action is selected the PDF is mailed, printed or opened. Open launches 3rd party PDF viewer app and copies data to the other app.

- (BOOL)actionSheetInViewController:([UIViewController *](#))inParentViewController;

Discussion

Displays standard standard action sheet based on properties.

```
// Sample to display action sheet allowing you to choose, mail, print,  
load and cancel options.  
  
razPDF.actionSheetTitle = @"My App - Choose Wisely";  
[razPDF actionSheetInViewController:self];
```

Availability

Available in RAZPDF 1.0 and later.

Declared In

razpdfdocument.h

mailInViewController:

Displays standard mail modal view controller with PDF attached.

- (BOOL)mailInViewController:([UIViewController *](#))inParentViewController

```
error:(NSError **)error;
```

Discussion

Displays standard mail modal view controller with defaults set based on properties. Note: In simulator the mailcomposer is displayed normally, but no email is actually sent from simulator, this is just the way the simulator works. Connect a device and you will be able to use your configured mail account to send messages with attached PDF.

```
// Sample to send email after PDF is rendered

razPDF.mailSubject = @"Test subject";
razPDF.RAZPDFMessageBody = @"Welcome to RAZPDF";

// Display mail composer with all defaults set and PDF attached

[razPDF mailInViewController:self error:nil];
```

```
// Enterprise, sample code to send HTML email with PDF attached:

razPDF.mailSubject = @"Test subject";
razPDF.RAZPDFMessageBody = @"<html><body>Welcome to <b>RAZPDF</b></body></html>";
razPDF.isHTML = YES;

// Display mail composer with all defaults set and PDF attached

[razPDF mailInViewController:self error:nil];
```

Availability

Available in RAZPDF 1.0 and later.

Declared In

razpdfdocument.h

printInViewController:

Displays standard print modal view controller

```
-- (BOOL)printControllerInViewController:(UIViewController *)inParentViewController
```

Discussion

Displays standard print modal view controller

```
// Sample code to print PDF  
  
[razPDF printDPFInViewController:self];
```

Availability

Available in RAZPDF 1.0 and later.

Declared In

razpdfdocument.h

loadPDFInViewController:

Displays list of other apps on device that support opening PDF files and if user chooses export data to other app.

- (BOOL)loadPDFInViewController:(UIViewController *)inParentViewController

Discussion

Displays modal view list of other apps on device that support PDF files and if user chooses app the other app opens the PDF. The other app does not have any access to the local applications files, it just gets a copy of the PDF. Copy of PDF might be retained in other app.

```
// Sample code to allow opening PDF in other applications  
  
[razPDF openInViewController:self];
```

Availability

Available in RAZPDF 1.0 and later.

Declared In

razpdfdocument.h

Properties

pdfAutodelete

Enables automatic removal of PDF file.

```
@property(nonatomic, assign) BOOL pdfAutodelete
```

Discussion

This boolean property when set to YES deletes PDF file when class is deallocated. This is helpful when a PDF is created and mailed and then is no longer needed.

```
// Sample code to enable auto delete  
  
razPDF.pdfAutodelete = YES;
```

Availability

Available in RAZPDF 1.0 and later.

Declared In

razpdfdocument.h

pdfWatermark

Enable RAZPDF watermark.

```
@property (nonatomic, assign) BOOL pdfWatermark;
```

Discussion

Useful if your making a prototype and want to enable watermark during development.

```
// Sample code to display watermark  
  
razPDF.pdfWatermark = YES;  
  
// Sample code to hide watermark
```

```
razPDF.pdfWatermark = NO;
```

Availability

Available in RAZPDF 1.0 and later.

Declared In

razpdfdocument.h

RAZPDF File Related Properties

PDF encoded properties and features. All these optional properties are stored in the PDF file.

pdfTitle

The title of the the document.

```
@property(nonatomic, retain) NSString *pdfTitle
```

Discussion

Optional string for the title of the the document.

```
// Sample code
```

```
razPDF.pdfTitle = @"My Greatest Secret";
```

Availability

Available in RAZPDF 1.0 and later.

Declared In

razpdfdocument.h

pdfAuthor

The name of the person who created the document.

```
@property(nonatomic, retain) NSString *pdfAuthor
```

Discussion

Optional string for the name of the person who created the document.

Note: set to "EVALUATION USE ONLY" in evaluation copy.

```
// Sample code  
razPDF.pdfAuthor = @"James Bond";
```

Availability

Available in RAZPDF 1.0 and later.

Declared In

razpdfdocument.h

pdfSubject

The subject of the document.

```
@property(n nonatomic, retain) NSString *pdfSubject
```

Discussion

Optional string for the subject of the document. This string is stored in the metadata of the PDF.

```
// Sample code to set PDF Subject  
razPDF.pdfSubject = @"This PDF is about ...";
```

Availability

Available in RAZPDF 1.0 and later.

Declared In

razpdfdocument.h

pdfOwnerPassword

PDF Owner Password.

```
@property(n nonatomic, retain) NSString *pdfOwnerPassword
```

Discussion

Optional string for the owner password of the the document. If this is specified, the document is encrypted using the value as the owner password; otherwise, the document will not be encrypted. The value of this key must be represented in ASCII encoding. Only the first 32 bytes are used for the password. There is no default value.

```
// Sample code  
razPDF.ownerPassword = @"DoubleSecret";
```

Availability

Available in RAZPDF 1.0 and later.

Declared In

razpdfdocument.h

pdfUserPassword

PDF Owner Password.

```
@property(nonatomic, retain) NSString *pdfUserPassword
```

Discussion

If the document is encrypted, then the value of this key will be the user password for the document. If not specified, the user password is the empty string. The value of this key must be represented in ASCII encoding. Only the first 32 bytes are used for the password.

```
// Sample code  
razPDF.userPassword = @"1234";
```

Availability

Available in RAZPDF 1.0 and later.

Declared In

razpdfdocument.h

pdfAllowsPrinting

Whether the document allows printing when unlocked with the user password. The default value of this key is YES.

```
@property(nonatomic, assign) BOOL pdfAllowsPrinting
```

Discussion

```
// Sample code to Disable user printing of PDF
```

```
razPDF.pdfAllowsPrinting = NO;
```

Availability

Available in RAZPDF 1.0 and later.

Declared In

razpdfdocument.h

pdfAllowsCopying

Whether the document allows copying when unlocked with the user password. The default value of this key is YES.

@property(nonatomic, assign) BOOL pdfAllowsCopying

Discussion

```
// Sample code to Disable user printing of PDF  
razPDF.pdfAllowsPrinting = NO;
```

Availability

Available in RAZPDF 1.0 and later.

Declared In

razpdfdocument.h

RAZPDF Mail Related Properties

mailSubject

Sets the initial text for the subject line of the email.

@property(nonatomic, retain) NSString *mailSubject

Discussion

Optional string for the initial text for the subject line of the email.

```
// Sample code to Disable user printing of PDF
```

```
razPDF.pdfAllowsPrinting = NO;
```

Availability

Available in RAZPDF 1.0 and later.

Declared In

razpdfdocument.h

mailToRecipients

Sets the initial recipients to include in the email's "To" field.

```
@property(n nonatomic, retain) NSArray *mailToRecipients
```

Discussion

An array of NSString objects, each of which contains the email address of a single recipient. This method does not filter out duplicate email addresses, so if duplicates are present, multiple copies of the email message may be sent to the same address.

```
// Sample code: mail to text@example.com,sales@ecasainc.com  
  
razPDF.mailToRecipients = [NSArray arrayWithObjects:@"test@example.com",  
@"sales@ecasainc.com", nil];
```

Availability

Available in RAZPDF 1.0 and later.

Declared In

razpdfdocument.h

mailCCRecipients

Sets the initial recipients to include in the email's "Cc" field.

```
@property(n nonatomic, retain) NSArray *mailCCRrecipients
```

Discussion

An array of NSString objects, each of which contains the "Cc" address of a single recipient. This method does not filter out duplicate email addresses, so if duplicates are present, multiple copies of the email message may be sent to the same address. See above for example.

Availability

Available in RAZPDF 1.0 and later.

Declared In

razpdfdocument.h

mailBCCRecipients

Sets the initial recipients to include in the email's "Bcc" field.

```
@property(nonatomic, retain) NSArray *mailBCCRecipients
```

Discussion

An array of NSString objects, each of which contains the "Bcc" address of a single recipient. This method does not filter out duplicate email addresses, so if duplicates are present, multiple copies of the email message may be sent to the same address. See above for example.

Availability

Available in RAZPDF 1.0 and later.

Declared In

razpdfdocument.h

mailMessageBody

The initial body text of the message. The text is interpreted as either plain text or HTML depending on the value of the isHTML property.

```
@property(nonatomic, retain) NSString *mailMessageBody
```

Discussion

Body text. The string "Powered by RAZPDF" appended to default body text.

```
razPDF.mailMessageBody = @"Thank you for the ...";
```

Availability

Available in RAZPDF 1.0 and later.

Declared In

razpdfdocument.h

mailFilename

Filename used to store PDF file. The last component of the path is stored in the property automatically. The property is read for used in mail attaches as the name of the attached file. Changing the fileName property does not change the PDF file in any way.

```
@property(n nonatomic, retain) NSString *fileName
```

Discussion

```
// Sample code to set fileName for attachment  
  
razPDF.mailFilename = @"my-great-app.pdf";
```

Availability

Available in RAZPDF 1.0 and later.

Declared In

razpdfdocument.h

isHTML

The initial body text of the message. The text is interpreted as either plain text or HTML depending on the value of the isHTML property.

```
@property(n nonatomic, assign) BOOL isHTML
```

Discussion

This boolean is used to enable html message body in enterprise version.

```
// Sample code: Enable HTML email  
  
razPDF.isHTML = YES;  
  
razPDF.mailMessageBody = @"<html><body>Hello <b>beautiful</B> world!</body></html>";
```

Availability

Available in RAZPDF 1.0 and later.

Declared In

razpdfdocument.h

RAZPDF UI Related Properties

actionSheetTitle

The color of any navigation bars presented by RAZPDF

@property(n nonatomic, retain) NSString *actionSheetTitle

Discussion

Defaults to @"Choose Option". Used to customize action sheet title at top of sheet.

```
// Sample code to set custom action sheet title  
razPDF.actionSheetTitle = @"My APP - Choose Option";
```

Availability

Available in RAZPDF 1.0 and later.

Declared In

razpdfdocument.h

translucent

The translucency of mail navigation bars presented by RAZPDF

@property(n nonatomic, assign) BOOL translucent

Discussion

Defaults to NO.

```
// Sample code to make menu bar transparent  
razPDF.translucent = YES;
```

Availability

Available in RAZPDF 1.0 and later.

Declared In

razpdfdocument.h

References:

<http://www.apple.com>

<http://en.wikipedia.org/wiki/Pdf>