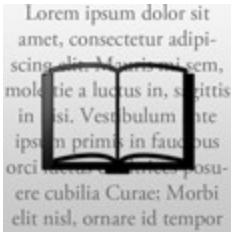
PDF Reader/Viewer V2.5 for iOS



GitHub: https://github.com/vfr/Reader

Website: http://www.vfr.org/

Introduction

I've crafted this open source PDF reader/viewer for fellow iOS developers struggling with wrangling PDF files onto iOS device screens.

The code is universal and does not require any XIBs (as all UI elements are code generated, allowing for greatest flexibility). It runs on iPad, iPhone and iPod touch with iOS 4.0 and up. Also supported are the Retina displays in all new devices and is ready to be fully internationalized. The idea was to provide a complete project template that you could start building from, or, just pull the required files into an existing project to enable PDF reading/viewing in your app(s).

After launching the sample app, tap on the left hand side of the screen to go back a page. Tap on the right hand side to go to the next page. You can also swipe left and right to change pages. Tap on the screen to fade in the toolbar and page slider. Double-tap with one finger (or pinch out) to zoom in. Double tap with two fingers (or pinch in) to zoom out.

This implementation has been tested with large PDF files (over 250MB in size and over 2800 pages in length) and with PDF files of all flavors (from text only documents to graphics heavy magazines). It also works rather well on older devices (such as the iPod touch 2nd generation and iPhone 3G) and takes advantage of the dual-core processor (via CATiledLayer and multi-threading) in new devices.

Features

Multithreaded: The UI is always quite smooth and responsive.

Supports:

- iBooks-like document navigation.
- Device rotation and all orientations.[1]
- Encrypted (password protected) PDFs.
- PDF links (URI and go to page).
- PDFs with rotated pages.

Notes

Version 2.x of the PDF reader/viewer code was originally developed and tested under Xcode 3.2.6, LLVM 1.7, iOS 4.3.5, iOS 4.2.1 with current development and testing under Xcode 4.3.2, LLVM 3.1, iOS 5. The code uses manual memory management and will continue to do so.

The overall PDF reader/viewer functionality is encapsulated in the ReaderViewController class. To present a document with this class, you first need to create a ReaderDocument object with the file path to the PDF document and then initialize a new ReaderViewController with this ReaderDocument object. The ReaderViewController class uses a ReaderDocument object to store information about the document and to keep track of document properties (thumb cache directory path, bookmarks and the current page number for example).

An initialized ReaderViewController can then be presented modally, pushed onto a UINavigationController stack, placed in a UITabBarController tab, or be used as a root view controller. Please note that since ReaderViewController implements its own toolbar, you need to hide the UINavigationController navigation bar before pushing it and then show the navigation bar after popping it. The ReaderDemoController class shows how this is done with a bundled PDF file. To create a 'book as an app', please see the ReaderBookDelegate class.

Required Files

The following files are required to incorporate the PDF reader/viewer into one of your projects:

```
CGPDFDocument.h              CGPDFDocument.m
ReaderDocument.h             ReaderDocument.m
ReaderConstants.h            ReaderConstants.m
ReaderViewController.h       ReaderViewController.m
ReaderMainToolbar.h          ReaderMainToolbar.m
ReaderMainPagebar.h          ReaderMainPagebar.m
ReaderContentView.h          ReaderContentView.m
ReaderContentPage.h          ReaderContentPage.m
ReaderContentTile.h          ReaderContentTile.m
ReaderThumbCache.h           ReaderThumbCache.m
ReaderThumbRequest.h         ReaderThumbRequest.m
ReaderThumbQueue.h           ReaderThumbQueue.m
ReaderThumbFetch.h           ReaderThumbFetch.m
ReaderThumbRender.h          ReaderThumbRender.m
ReaderThumbView.h            ReaderThumbView.m
ReaderThumbsView.h           ReaderThumbsView.m
ThumbsViewController.h       ThumbsViewController.m
ThumbsMainToolbar.h          ThumbsMainToolbar.m
UIXToolbarView.h             UIXToolbarView.m

Reader-Button-H.png          Reader-Button-H@2x.png
Reader-Button-N.png          Reader-Button-N@2x.png
Reader-Email.png             Reader-Email@2x.png
```

```
Reader-Mark-N.png           Reader-Mark-N@2x.png
Reader-Mark-Y.png           Reader-Mark-Y@2x.png
Reader-Print.png            Reader-Print@2x.png
Reader-Thumbs.png           Reader-Thumbs@2x.png
```

Required iOS Frameworks

To incorporate the PDF reader/viewer into one of your projects, all of the following iOS frameworks are required by the code:

```
UIKit          Foundation
CoreGraphics   QuartzCore
ImageIO        MessageUI
```

Compile Time Options

In ReaderConstants.h the following #define options are available:

READER_BOOKMARKS - If TRUE, enables page bookmark support.

READER_ENABLE_MAIL - If TRUE, an email button is added to the toolbar (if the device is properly configured for email support).

READER_ENABLE_PRINT - If TRUE, a print button is added to the toolbar (if printing is supported and available on the device).

READER_ENABLE_THUMBS - If TRUE, a thumbs button is added to the toolbar (enabling page thumbnail document navigation).

READER_DISABLE_IDLE - If TRUE, the iOS idle timer is disabled while viewing a document (beware of battery drain).

READER_SHOW_SHADOWS - If TRUE, a shadow is shown around each page and the page content is inset by a couple of extra points.

READER_STANDALONE - If FALSE, a "Done" button is added to the toolbar and the -dismissReaderViewController: delegate method is messaged when it is tapped.

READER_DISABLE_RETINA - If TRUE, sets the CATiledLayer contentScale to 1.0f. This effectively disables retina support and results in non-retina device rendering speeds on retina display devices at the loss of retina display quality.

ReaderDocument Archiving

To change where the property list for ReaderDocument objects is stored (~/Library/Application Support/ by default), see the +archiveFilePath: method in the ReaderDocument.m source file. Archiving and unarchiving of the ReaderDocument object for a document is mandatory since this is where the current page number, bookmarks and directory of the document page thumb cache is kept.

Caveats

[1] There appears to be a bug in iOS 4 with its view controller rotation handling when a modal view controller is presented in landscape from a modal view controller when the status bar is visible on iPhone and iPod touch. The good news is that iOS 5 appears to have this bug fixed.

History

2011-07-29: Version 2.0.0

- Released v2.0 into the wild for general use.

2011-08-25: Version 2.1.0

- Added PDF link (URI and document page) support.
- Assorted code cleanup, optimizations and bug fixes.

2011-08-27: Version 2.1.1

- Fixed rotation handling when in a UITabBarController.

2011-09-09: Version 2.2.0

- A medium resolution page preview image is now shown first.
- The page bar now uses small page thumbs instead of a slider.

2011-09-10: Version 2.2.1

- Added password handling to ReaderDocument plist unarchive.

2011-09-14: Version 2.3.0

- Added page thumbnail document navigation.
- Added support for page bookmarks.

2011-09-20: Version 2.4.0

- Replaced UIToolbars with custom UIView-based toolbars.

2011-10-05: Version 2.5.0

- Critical bug fixes and various assorted tweaks.

2011-10-06: Version 2.5.1

- Fixed content view centering under iOS 5.

2011-10-15: Version 2.5.2

- One (crashing) bug fix and one minor UI tweak.

2011-11-08: Version 2.5.3

- Various refinements and minor bug fixes.

2012-01-14: Version 2.5.4

- Bug fix to PDF link handling in older format PDFs.
- Changed from CC BY 3.0 License to MIT License.

2012-04-10: Version 2.5.5

- Handles PDF web links without http:// as the prefix.
- Bug fix to PDF link handling with cropboxed PDF files.
- Some performance improvements on iPad 3rd generation.

2012-04-16: Version 2.5.6

- Now loads and decodes thumbnail PNGs on a background thread.
- Added `READER_DISABLE_RETINA` #define performance option.

Acknowledgements

The PDF link support code in the ReaderContentPage class is based on the links navigation code by Sorin Nistor from http://ipdfdev.com/.