PDF Reader Core V2.8 for iOS

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

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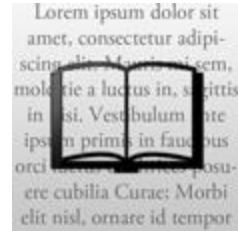
Introduction

I've crafted this open source PDF reader code 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 6.0 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 4th generation and iPhone 3GS) and takes advantage of the dual-core processor (via CATiledLayer and multi-threading) in new devices.

To see an example open source PDF Viewer App that uses this code as its base, have a look at this project repository on GitHub: https://github.com/vfr/Viewer.

Features

Multithreaded: The UI is always quite smooth and responsive.

Supports:

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

Notes

Version 2.x of the PDF reader code was originally developed and tested under Xcode 3.2.6, LLVM 1.7 and iOS 4 with current development and testing under Xcode 6.0, LLVM 6.0 and iOS 8. Please note that as of v2.6, the code was refactored to use ARC.

The overall PDF reader 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.

Required Files

The following source files are needed to incorporate the PDF reader 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

Localizable.strings (UTF-16 encoding)
```

The following PNG graphics files are also needed:

```
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-Export.png           Reader-Export@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 code into one of your projects, all of the following iOS frameworks are required:

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

Compile Time Options

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

READER_FLAT_UI - If TRUE, does not show button borders.

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_ENABLE_EXPORT - If TRUE, uses UIDocumentInteractionController to be able to open up the current PDF in other iOS applications.

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.

READER_ENABLE_PREVIEW - If TRUE, a medium resolution page thumbnail is displayed before the CATiledLayer starts to render the PDF page.

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.

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.

History

Please see the HISTORY.md file in the Reader Xcode project directory for the latest release notes.

Acknowledgements

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