TABLE OF CONTENTS

1.   About PDFNet SDK

PDFNet SDK is a high-quality, industry-strength PDF library meeting requirements of the most demanding and diverse applications. Using PDFNet SDK you can write stand-alone, cross-platform, and reliable commercial applications that can read, write, create, edit, print, convert, and display PDF documents. PDFNet SDK is available as a .NET component, and as a cross-platform C/C++ library with Java, PHP, Ruby, Python and Objective-C interfaces, available on a wide range of platforms such as Windows, WinRT, Windows Mobile, Linux, Solaris, Mac OS X, iOS, Android, etc. For more information, please visit us at www.pdftron.com.

1.1   SDK Overview

- **Lib** - Folder containing PDFNet libraries/components.
- **CosEdit** - An evaluation version of CosEdit, a stand-alone, low-level PDF editor. Using the evaluation version the users can inspect the low-level structure and contents of PDF documents.
- **Doc** - A folder containing PDFNet API Reference Manual (as JavaDoc, C/C++ Reference, WinHelp, etc).
- **Samples** - Folder containing PDFNet sample projects.
- **legal.txt** - Legal and copyright information.
- **license.pdf** - CPU license agreement and warranty for this product.
- **gettingstarted.pdf** - This file.
- **whatsnew.html** - Document describing the latest changes and additions to PDFNet SDK.

2.   Choosing the Correct PDFNet library

The PDFNet SDK comes in a variety of versions to best suit your needs. For example, if you are working in a windows environment we provide .Net 2 and .Net 4 (with WPF) libraries, that allow you to develop in VB or C#. If you are developing a Unix or OSX platform, or need to support multiple platforms, we provide a native C/C++ version, that also works with Java, PHP, Python, Ruby and Objective-C interfaces.
For the latest available libraries please see http://www.pdftron.com/pdfnet/downloads.html.

3.   Getting Started

3.1   Creating a simple PDF-to-image application

Here we describe important steps to using PDFNet, by creating a very simple program that converts the first page of a PDF to an image.

1. Open Visual Studio and create a new C# Console Application.
2. Add reference to PDFNet.dll.
3. Verify that the platform target (x86 or x64) of the application matches the PDFNet.dll architecture.
4. Add using statement to beginning of CS file.

```
using pdftron.PDF;
```

5. Add Initialization call. This needs to be called before any other calls to PDFNet and should only be called once in your main() or OnAppStartup() functions.

```
pdftron.PDFNet.Initialize();
```

6. Load an existing PDFDoc and initialize the default security handler. To open password protected PDFs see the Encryption sample. Notice the using declaration; this is required to ensure that PDFDoc object is released (including its handle on the actual file) when we are done.

```
using (PDFDoc doc = new PDFDoc(args[0]))
{
    doc.InitSecurityHandler();
```

7. Export PDF to image.

```
PDFDraw draw = new PDFDraw();
    draw.Export(doc.GetPage(1), args[1]);
}
```

8. Wrap everything in try / catch block.
9. Add PDFNetException catch.

```
catch(pdftron.Common.PDFNetException ex)
{
    System.Console.WriteLine(ex.GetMessage());
}
```

10. Run program, where the first command line argument is a PDF file, and the second is the name of the image. The PDFDraw function will automatically match the image format to that of the file name extension provided by the user. For example, "*app.exe input.pdf output.jpg*" will result in the first page of input.pdf being converted to a JPEG file.

3.1.1 Common Questions

Is PDFNet compatible with AnyCPU?
Where should I call the PDFNet.Initialize()?
A question regarding PDFNet.Initialize() and Terminate().

Error: warning MSB3270: There was a mismatch between the processor architecture of the project being built.
Solution: The architecture of the PDFNet dll and your project do not match. Either both the DLL, and project, should be x64, or x86. Don't mix and match. If using AnyCPU, it is not recommended for use with PDFNet as this can result in errors in some cases. For details, please see following link. https://groups.google.com/forum/?fromgroups#!searchin/pdfnet-sdk/anycpu/pdfnet-sdk/MV8wBvAVCak/hrpjmcgUAQ8J

Error: System.BadImageFormatException: Could not load file or assembly PDFNet... An attempt was made to load a program with an incorrect format.
Solution: Same as above, but at runtime you have the wrong PDFNet dll being loaded.

Error: The evaluation version encountered an error in your system configuration. Please contact customer support at: sup...@pdftron.com. No reg. write rights
Solution: See *Questions* in the *Demo/Trial Period Prerequisite* section below.

3.2 Deploying Your Application

Now that your new project is ready for others to use, it is time to distribute it.

3.2.1 Dependencies

For your clients to use PDFNet for .Net 4.0 they will need the Microsoft Visual C++ 2010 redistributable on their system (the files MSVCP100.DLL and MSVCR100.DLL). This is a standard component of Visual Studio, which is why it was not needed during development. You can either have your users install the package from Microsoft, or you can include the DLL as part of your install by including it in the same folder as your application.
Microsoft VC 2010 Redistributable Package (x86)
Microsoft VC 2010 Redistributable Package (x64)

3.2.2 Common Questions

Deploying PDFNet for .NET/WPF: How to resolve FileLoadException due to missing VC redistributable?
Could not load file or assembly 'PDFNet.dll' ' when using PDFNet for .NET 4.x?
How do I resolve 'FileNotFoundException: Could not load file or assembly 'PDFNet.dll' or one of its dependencies' [64-bit and .NET 4 specific]?

3.3 Finished!

Congratulations, you are now ready to deploy the project to your target/customer computers.

4.   Next Steps

4.1   Demo / Trial Mode

PDFTron Systems offers a free, 60-day evaluation of PDFNet. Once the 60-day evaluation period expires, the library will start generating an exception/error indicating that the evaluation period has expired and the library will stop functioning.

4.1.1   Elevated Permissions Prerequisite

While you try out PDFNet without a license key, it is running in demo mode. One side effect of this is that your application will need read/write access to the registry. This is typically a problem only when you run the app in an environment that requires elevated permissions, such as when developing a Windows service or a server application.

4.1.2   Example: IIS Server

If you are trying to run your demo on an IIS server, you can change the Application Pool's identity to a User Account with read/write access to the registry.
1. Go to IIS Manager, select the web server, and then Application Pools.
2. Click on the Application Pool you are using from the list, then on the Actions panel, under Edit Application Pool, click Advanced Settings...
3. In the Advanced Settings dialog, go to the Process Model section and click on the Identity to change it.
4. Once the Application Pool Identity dialog shows up, select Custom account:, then click Set...
5. Enter the user account information in the Set Credentials dialog box. Note: you may have to specify the domain of the user (i.e. MyServerDomain\MyUserAccount).
6. Click OK until all dialog boxes are closed.

4.1.3   Common Questions

How do I resolve error in your system configuration ... No reg. write rights?
Going around "No reg. write rights" exception during PDFNet SDK demo evaluation?
How do I resolve "The evaluation version encountered an error... No reg. write rights"?

4.2   Servers

There are some particular issues to consider when running your application in a server environment. Below are some of the most common situations.

4.2.1   Visual Studio Web Server

If you get a "BadImageFormatException, incorrect format" error, while trying to use the Web Server that comes with Visual Studio, this is because VS Web Server is 32bit only and cannot be used with any 64bit libraries. Please see this forum posting for solutions.
https://groups.google.com/forum/?fromgroups#!searchin/pdfnet-sdk/iis$20default$2032/pdfnet-sdk/Op5m-2VH2ww/OkkQHysZWHgJ

4.2.2 IIS Servers

You may also encounter BadImageFormatException, incorrect format , exceptions when using PDFNet on IIS. One possible cause can be that you need to match the IIS 'Enable 32-bit application' setting to which version of PDFNet (32 or 64 bit) that you are using. Finally be aware that different versions of IIS default to different versions of the .Net framework, so you also need to match that with the version of PDFNet you are using (.Net 2 or 4). For more info see: https://groups.google.com/forum/?fromgroups#!searchin/pdfnet-sdk/iis$20default$2032/pdfnet-sdk/KjFgCY9oNPM/A6rG6IIWR4UJ

4.2.3 File Conversions in Windows Service or ASP.Net

The following online article contains lots of important information regarding converting files to and from PDF in a server environment. This document also covers MS Office Interop and virtual printer related issues.
http://www.pdftron.com/pdfnet/faq_files/Converting_Documents_in_Windows_Service_or_ASP.NET_Application_using_PDFNet.pdf

4.3 Adding Features

When you decide to add new features, be sure to start by checking out the relevant Sample project: http://www.pdftron.com/pdfnet/samplecode.html
After that, please check out our forums for additional information not covered in the samples. https://groups.google.com/forum/#!forum/pdfnet-sdk

4.4 Getting a Production License

Now that you've built your application, and tested it in a deployment environment, you might be ready to purchase a license. Please take a look at the following links for steps required to obtain the production license.
Options: http://www.pdftron.com/store/licensing.html
FAQ : http://www.pdftron.com/store/faq.html
Store: http://www.pdftron.com/store/index.php

4.4.1 Using the License

Once you purchase a license, you will receive a license key that you can use to initialize PDFNet in your application code.

```
pdftron.PDFNet.Initialize("MyCompany:OEM:1234");
```

Initializing PDFNet with the license key will also remove the requirement to run your demo application with elevated rights in a server environment.

4.4.2 Common Questions

Error: Bad License Key. PDFNet SDK will work in the demo mode. For more information please see PDFNet FAQ or contact support@pdftron.com.
Solution: The license key was copied incorrectly. First, make sure no extra white space was added to the beginning or end of the string. Finally, make sure you are copying the entire license key as it was sent to you. Exception when using license key

Error: Your AMS expired on.... To upgrade to this version of PDFNet SDK, please renew your AMS subscription.
Solution: Your license is out of date and cannot use newer versions of PDFNet. Either return to using an earlier version, or contact sales at pdftron.com to renew.
[PDFTron watermark still appears in generated documents after license purchase?](#)

5. Getting Help

5.1 SDK Samples

First, checking the available samples is a good place to get help. The following link provides a brief summary of the samples that come with your PDFNet package.
http://www.pdftron.com/pdfnet/samplecode.html

5.2 Public Forum

Next, check the PDFNet SDK forums, as there are plenty of answers available already. Odds are someone else has already asked your question, but it just didn't make it into this document.
https://groups.google.com/forum/?fromgroups#!forum/pdfnet-sdk
Otherwise, feel free to post questions to the forum.

5.3 Additional Resources

5.3.1 CosEdit Tool

Included with your download of PDFNet SDK is a trial version of a tool called CosEdit. This is a useful tool for viewing the actual content and structure of a particular PDF file.

5.3.2 Samples

Also included with your PDFNet SDK is a collection of samples that demonstrate most of the features available with PDFNet.
http://www.pdftron.com/pdfnet/samplecode.html

5.3.3 Links

FAQ: http://www.pdftron.com/pdfnet/faq.html
Introduction: http://www.pdftron.com/pdfnet/intro.html
Documentation (API References): http://www.pdftron.com/pdfnet/documentation.html

5.3.4 Technical Support

At PDFTron, we understand that good technical support is as important to our customers as our products themselves. Our technical support team consists of the same developers that build our products, work on new features and fix the bugs. This ensures that our customers receive accurate and to-the- point answers to their questions or problems.
If you think you have encountered a bug, or issue, with PDFNet, you can submit a ticket. Simply fill out the following form: http://www.pdftron.com/support/reportproblem.html