C# PDF

A PDF Cheat Sheet for Generating, Editing and OCR of PDF documents for the .Net Framework using IronPdf in C#

Installing the Package using Nuget.Org

```
PM> Install-Package IronPdf
```

The easiest way to install IronPdf is using NuGet Package Manager for Visual-Studio:
The package name is “IronPdf”.

https://www.nuget.org/packages/IronPdf/

Why IronPdf

Adding Pdf file generation in .Net project is a cumbersome; and also converting web forms, HTML and/or online web pages to Pdf is a very hard and complex problem,

IronPdf Works with ASP.NET web forms, MVC, Web services, WPF, secure internet and by using IronPdf you will be able add lot of features and capability to your projects, and you will be able to produce Pdf files as report or as document.

You can learn more from http://ironpdf.com

Hello World

```
using IronPdf;
IronPdf.HtmlToPdf Renderer = new IronPdf.HtmlToPdf();
Renderer.RenderHtmlAsPdf("<h1>Hello World<h1>”).SaveAs(“html-string.pdf");
```

Contents

C# PDF Example Code

C# Html To Pdf

```
using IronPdf;
IronPdf.HtmlToPdf Renderer = new IronPdf.HtmlToPdf();
Renderer.RenderHtmlAsPdf("<h1>Hello World<h1>”).SaveAs(“html-string.pdf");
```

C# Html To Pdf by Url

```
using IronPdf;

IronPdf.HtmlToPdf Renderer = new IronPdf.HtmlToPdf();
HtmlToPdf HtmlToPdf = new IronPdf.HtmlToPdf();
Renderer.RenderUrlAsPdf("http//ironpdf.com/").SaveAs("url.pdf")
```

Asp.Net Aspx Page to Pdf

```
private void Form1_Load(object sender, EventArgs e)
{
IronPdf AspxToPdf.RenderThisPageAsPDF();
//Changes the ASPX output into a pdf instead of html
}
```

MVC View to PDF

```
public ActionResult Index()
{
var PDF = HtmlToPdf.StaticRenderUrlAsPdf(new Uri("https:/en.wikipedia.org"));
return File(PDF.BinaryData, "application/pdf", "Wiki.Pdf");
}
```

Header & Footer (Simple)

```
using IronPdf;

IronPdf.HtmlToPdf Renderer = new IronPdf.HtmlToPdf();

// add a header to very page easily
Renderer.PrintOptions.FirstPageNumber = 1; // use 2 if a cover page  will be
appended
Renderer.PrintOptions.Header.DrawDividerLine = true;
Renderer.PrintOptions.Header.CenterText = "{url}" ;
Renderer.PrintOptions.Header.FontFamily = "Helvetica,Arial";
Renderer.PrintOptions.Header.FontSize = 12;

// add a footer too
Renderer.PrintOptions.Footer.DrawDividerLine = true;
Renderer.PrintOptions.Footer.FontFamily = "Arial";
Renderer.PrintOptions.Footer.FontSize = 10;
Renderer.PrintOptions.Footer.LeftText = "{date} {time}";
Renderer.PrintOptions.Footer.RightText = "{page} of {total-pages}";

// mergeable fields are:
// {page} {total-pages} {url} {date} {time} {html-title} & {pdf-title}
```

Header & Footer (HTML)

```
using IronPdf;

IronPdf.HtmlToPdf Renderer = new IronPdf.HtmlToPdf();

// Build a footer using html to style the text
// mergeable fields are:
// {page} {total-pages} {url} {date} {time} {html-title} & {pdf-title}
Renderer.PrintOptions.Footer = new HtmlHeaderFooter()
{
  Height = 15,
  HtmlFragment = "<center><i>{page} of {total-pages}<i></center>",
  DrawDividerLine = true
 };

// Build a header using an image asset
// Note the use of BaseUrl to set a relative path to the assets
Renderer.PrintOptions.Header = new HtmlHeaderFooter()
{
  Height = 20,
  HtmlFragment = "<img src='logo.png'>",
  BaseUrl = new Uri(@"C:\assets\images\").AbsoluteUri
};
```

Encryption, Decryption & Passwords

```
using IronPdf;
//Open an Encrypted File
PdfDocument PDF = PdfDocument.FromFile("encrypted.pdf", "password");

// Save and change the encryption password.
PDF.Password = "my-password";
PDF.SaveAs(“secured.pdf");
```

Merge, Join, Split and Edit PDF Documents

```
using IronPdf;
using System.Collections.Generic;

var Renderer = new IronPdf.HtmlToPdf();

// Join Multiple Existing PDFs into a single document
var PDFs     = new List<PdfDocument>();
PDFs.Add(PdfDocument.FromFile("A.pdf"));
PDFs.Add(PdfDocument.FromFile("B.pdf"));
PDFs.Add(PdfDocument.FromFile("C.pdf"));
PdfDocument PDF =  PdfDocument.Merge(PDFs);
PDF.SaveAs("merged.pdf");

// Add a cover page
PDF.PrependPdf(Renderer.RenderHtmlAsPdf("<h1>Cover Page</h1><hr>"));

// Remove the last page from the PDF and save again
PDF.RemovePage(PDF.PageCount - 1);
PDF.SaveAs("merged.pdf");

// Copy pages 5-7 and save them as a new document.
PDF.CopyPages(4,6).SaveAs(“exerpt.pdf");
```

Extract Text and Images from PDF Documents

```
using IronPdf;

// open a 128 bit encrypted PDF
PdfDocument PDF = PdfDocument.FromFile("encrypted.pdf", "password");

//Get all text to put in a search index
string AllText = PDF.ExtractAllText();

//Get all Images
IEnumerable<System.Drawing.Image> AllImages = PDF.ExtractAllImages();

//Or even find the precise text and images for each page in the document
for (var index = 0; index < PDF.PageCount; index++) {
    int PageNumber = index + 1;
    string Text = PDF.ExtractTextFromPage(index);
    IEnumerable<System.Drawing.Image> Images =
PDF.ExtractImagesFromPage(index);
    ///...
}
```

Rendering Html With Javascript

```
PdfPrintOptions _printOption = new PdfPrintOptions()
{

      EnableJavaScript = true,
      RenderDelay = 100
};
```

C# PDF OCR

PDF OCR is more advanced than extracting assets using the PdfDocument object as is can extract Text and data from scanned images, protected text and rasterized text.

```
PM> Install-Package IronOcr
```

```
using IronOcr;

var Ocr = new AdvancedOcr()
{
    CleanBackgroundNoise = false,
    ColorDepth = 4,
    ColorSpace = AdvancedOcr.OcrColorSpace.Color,
    EnhanceContrast = false,
    DetectWhiteTextOnDarkBackgrounds = false,
    RotateAndStraighten = false,
    Language = IronOcr.Languages.English.OcrLanguagePack,
    EnhanceResolution = false,
    InputImageType = AdvancedOcr.InputTypes.Document,
    ReadBarCodes = true,
    Strategy = AdvancedOcr.OcrStrategy.Fast
};

var Results = Ocr.ReadPdf(@"C:\Users\Me\Desktop\Invoice.pdf");
var Pages = Results.Pages;
var Barcodes = Results.Barcodes;
var FullPdfText = Results.Text;
```

C# PDF Settings

CODE SAMPLE

```
using IronPdf;

IronPdf.HtmlToPdf Renderer = new IronPdf.HtmlToPdf();

Renderer.PrintOptions.SetCustomPaperSizeInInches(12.5, 20);
Renderer.PrintOptions.PrintHtmlBackgrounds = true;
Renderer.PrintOptions.PaperOrientation =
PdfPrintOptions.PdfPaperOrientation.Portrait;
Renderer.PrintOptions.Title = "My PDF Document Name";
Renderer.PrintOptions.EnableJavaScript = true;
Renderer.PrintOptions.RenderDelay = 50; //ms
Renderer.PrintOptions.CssMediaType = PdfPrintOptions.PdfCssMediaType.Screen;
Renderer.PrintOptions.DPI = 300;
Renderer.PrintOptions.FitToPaperWidth = true;
Renderer.PrintOptions.JpegQuality = 80;
Renderer.PrintOptions.GrayScale = false;
Renderer.PrintOptions.FitToPaperWidth = true;
Renderer.PrintOptions.InputEncoding = Encoding.UTF8;
Renderer.PrintOptions.Zoom = 100;
Renderer.PrintOptions.CreatePdfFormsFromHtml = true;

Renderer.PrintOptions.MarginTop = 40;  //millimenters
Renderer.PrintOptions.MarginLeft = 20;  //millimenters
Renderer.PrintOptions.MarginRight = 20;  //millimenters
Renderer.PrintOptions.MarginBottom = 40;  //millimenters

Renderer.PrintOptions.FirstPageNumber = 1; //use 2 if a cover page  will be
appended


Renderer.RenderHTMLFileAsPdf(“my-content.html").SaveAs("my-content.pdf");
```

Settings And Advanced Options - Full Reference

Class	PdfPrintOptions	
Description	Used to define pdf print out options like paper size , dpi , header and footer	
Properties \ functions	**Type**	**Description**
CreatePdfFormsFromHtml	Boolean	Turns all html forms elements into editable pdf forms

CssMediaType	Enum
	PdfCssMediaType
	{
	Print=0,
	Screen=1
	}
		Enable media =”Screen”, Css Styles and stylesheets.
		Note: By setting Allow ScreenCss=false; IronPdf prints using css for media=”print” only.

CustomCssUrl	Uri	Allow Custom Css Style sheet to be applied on html before rendering, you may set it to remote URL or local file path
DPI	int	Define the number of Print out Dpi (Dot Per Inch). Its standard value is 300 Dpi. Increasing Dpi value make images and text output more clearly and also increase pdf file size.
EnableJavaScript	Boolean	By default its value = false. It Enable\disable java script and json execution for 100ms before page is rendered. Great option for printing from client scripting frameworks that uses JavaScript for its operations like Ajax or angular or equivalent frameworks.
FirstPageNumber	int	Used with page header or footer to set first page start number
FitToPaperWidth	Boolean	Set FitToPaperWidth=true will force IronPdf to fit rendered content into one page only if it possible
Footer	PdfHeaderFooter	Set the footer content see PdfHeaderFooter Class
Header		
GrayScale	Boolean	Output pdf in black and white scale
InputEncoding	System.Text.Encoding	Define input character encoding
JpegQuality	Int	Quality of images , take values from 0 to 100
LicenseKey	String	Set license key and Remove water mark
MarginBottom	Int	Paper margin in millimeter , set to zero for borderless
MarginLeft	Int	Paper margin in millimeter , set to zero for borderless
MarginRight	Int	Paper margin in millimeter , set to zero for borderless
MarginTop	Int	Paper margin in millimeter , set to zero for borderless

PaperOrientation	Enum PdfPaperOrientation { Portrait, Landscape }	Set output pdf orientation
PaperSize	Enum PdfPaperSize	Set output pdf page size (A4 , A3 , etc)
PrintHtmlBackgrounds	Boolean	Print background color and images from html
RenderDelay	Int	Set waiting milliseconds before rendering html, this option useful when used to render pages contain animation or Ajax.
Title	string	Optionally can set Pdf title and meta data title
Zoom		Set enlarge zoom level (%) for rendering html
SetCustomPaperSize(int width , int height)	Function	Used to set custom paper size