XDoc.PDF Developer Guide – Basic Module

Table of Contents

Load a PDF file

```
// Load from a file
String inputFilePath1 = Program.RootPath + "\\" + "1.pdf";
PDFDocument doc1 = new PDFDocument(inputFilePath1);
if (doc1 == null) throw new Exception("fail to load the file");
// ...

// Load from a stream
String inputFilePath2 = Program.RootPath + "\\" + "2.pdf";
using (FileStream fileStream = File.Open(inputFilePath2, FileMode.Open, FileAccess.Read))
{
   PDFDocument doc2 = new PDFDocument(fileStream);
   if (doc2 == null) throw new Exception("fail to load PDF document from the stream");
   // ...
}
```

Get page count of a file

```
String inputFilePath = Program.RootPath + "\\" + "1.pdf";
PDFDocument doc = new PDFDocument(inputFilePath);
int pageCount = doc.GetPageCount();
Console.WriteLine("Page Count: " + pageCount);
```

Render PDF page to different targets

To .NET Bitmap

```
String inputFilePath = Program.RootPath + "\\" + "1.pdf";
PDFDocument doc = new PDFDocument(inputFilePath);
PDFPage page = (PDFPage)doc.GetPage(0);

//  render first page to a bitmap with default setting
Bitmap bitmap1 = page.GetBitmap();
bitmap1.Save(inputFilePath + "_1.bmp");
//  ...

//  enlarge the page with factor 2
float zoomRatio = 2.0F;
Bitmap bitmap2 = page.GetBitmap(zoomRatio);
bitmap2.Save(inputFilePath + "_2.bmp");
//  ...

//  render page (with resoution = 300 dpi)
int targetResolution = 300;
Bitmap bitmap3 = page.GetBitmap(targetResolution);
bitmap3.Save(inputFilePath + "_3.bmp");
//  ...
```

To REImage

```
String inputFilePath = Program.RootPath + "\\" + "1.pdf";
PDFDocument doc = new PDFDocument(inputFilePath);
//  get the first page
PDFPage page = (PDFPage)doc.GetPage(0);
//  convert first page to a REImage object
REImage image = page.ConvertToImage();
//  do something ...
```

To Image File

```
String inputFilePath = Program.RootPath + "\\" + "1.pdf";
PDFDocument doc = new PDFDocument(inputFilePath);
//  get the first page
PDFPage page = (PDFPage)doc.GetPage(0);
//  convert first page to a BMP file
page.ConvertToImage(ImageType.BMP, Program.RootPath + "\\Output.bmp");
//  convert first page to a GIF file
page.ConvertToImage(ImageType.GIF, Program.RootPath + "\\Output.gif");
//  convert first page to a JPEG file
page.ConvertToImage(ImageType.JPEG, Program.RootPath + "\\Output.jpg");
//  convert first page to a PNG file
page.ConvertToImage(ImageType.PNG, Program.RootPath + "\\Output.png");
```

Output an exist PDFDocument object to a file

```
String inputFilePath = Program.RootPath + "\\" + "1.pdf";
String outputFilePath = Program.RootPath + "\\" + "1_new.pdf";
PDFDocument doc = new PDFDocument(inputFilePath);

//  ... do something ... for example: delete first page
doc.DeletePage(0);

doc.Save(outputFilePath);
```

Retrieve document metadata

```
String inputFilePath = Program.RootPath + "\\" + "2.pdf";
PDFDocument doc = new PDFDocument(inputFilePath);
PDFMetadata metadata = doc.Description;
Console.WriteLine("Title:        " + metadata.Title);
Console.WriteLine("Author:       " + metadata.Author);
Console.WriteLine("Subject:      " + metadata.Subject);
Console.WriteLine("Keywords:     " + metadata.Keywords);
Console.WriteLine("Creator:      " + metadata.Creator);
Console.WriteLine("Producer:     " + metadata.Producer);
Console.WriteLine("Create Date:   " + metadata.CreatedDate.ToString());
Console.WriteLine("Modified Date: " + metadata.ModifiedDate.ToString());
```

Update document metadata

```
PDFMetadata metadata = new PDFMetadata();
metadata.Title = "Title";
metadata.Author = "Qi";
metadata.Subject = "None";
metadata.Keywords = "University, Public, etc.";
metadata.Creator = "MS Office Word";
metadata.Producer = "RE";
metadata.CreatedDate = new DateTime(2014, 11, 21, 10, 45, 12);
metadata.ModifiedDate = new DateTime(2015, 11, 21, 10, 45, 12);

String inputFilePath = Program.RootPath + "\\" + "1.pdf";
String outputFilePath = Program.RootPath + "\\" + "1_new.pdf";
PDFDocument doc = new PDFDocument(inputFilePath);

doc.SetDescription(metadata);

doc.Save(outputFilePath);
```

Retrieve document outline

```
String inputFilePath = Program.RootPath + "\\" + "2.pdf";
PDFDocument doc = new PDFDocument(inputFilePath);
REOutline outline = doc.GetOutline();
foreach (REEntry entry in outline.Entry)
{
   Console.WriteLine("Page:     " + entry.GetPageIndex());
   Console.WriteLine("Level:    " + entry.GetLevel());
   Console.WriteLine("Position: " + entry.GetLocation());
   Console.WriteLine("Text:     " + entry.GetText());
}
```

Update document outline

```
String inputFilePath = Program.RootPath + "\\" + "2.pdf";
String outputFilePath = Program.RootPath + "\\" + "Output_Outline.pdf";
PDFDocument doc = new PDFDocument(inputFilePath);

//  initial a new outline object
REOutline outline = new REOutline();

//  create 1st entry: Level = 1; Location: page index 0, y = 50
OutLine entryChapter1 = new OutLine("Chapter 1: *******", 1, 0, 50);
//  add 1st entry to outline
outline.Entry.Add(entryChapter1);

//  create 1st entry's child entries
//  create 1st child entry: Level = 2; Location: page index 1, y = 0
OutLine entryChapter11 = new OutLine("###### #### #### #### ###", 2, 1, 0);
//  connect this entry to the parent entry
entryChapter11.SetParentREEntry(entryChapter1);
//  add 1st child of 1st entry to outline
outline.Entry.Add(entryChapter11);

//  create 2nd child entry: Level = 2; Location: page index 2, y = 0
OutLine entryChapter12 = new OutLine("## ## ## ## ## #### ###", 2, 2, 0);
//  connect this entry to the parent entry
entryChapter12.SetParentREEntry(entryChapter1);
//  add 2nd child of 1st entry to outline
outline.Entry.Add(entryChapter12);

//  create 2nd entry: Level = 1; Location: page index 3, y = 100
OutLine entryChapter2 = new OutLine("Chapter 2: ******* ****", 1, 3, 100);
//  add 2nd entry to outline
outline.Entry.Add(entryChapter2);

//  create 3rd entry: Level = 1; Location: page index 5, y = 200
OutLine entryChapter3 = new OutLine("Chapter 3: **** **** ***", 1, 5, 200);
//  add 3rd entry to outline
outline.Entry.Add(entryChapter3);

//  create 3rd entry's child entries
//  create 1st child entry: Level = 2; Location: page index 9, y = 0
OutLine entryChapter31 = new OutLine("# #### ###", 2, 9, 0);
//  connect this entry to the parent entry
entryChapter31.SetParentREEntry(entryChapter3);
//  add 1st child of 3rd entry to outline
outline.Entry.Add(entryChapter31);
```

(Continue)

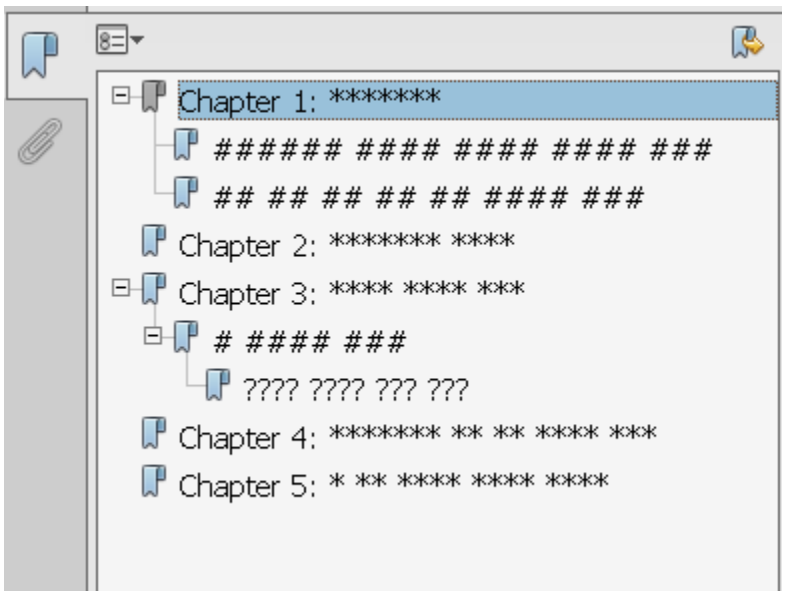
```
// create a child entry for the 1st child entry of the 3rd entry
// create entry: Level = 3; Location: page index 9, y = 500
OutLine entryChapter311 = new OutLine("???? ???? ??? ???", 3, 9, 500);
// connect this entry to the parent entry
entryChapter311.SetParentREEntry(entryChapter31);
// add 1st child of 3rd entry to outline
outline.Entry.Add(entryChapter311);

// create 4th entry: Level = 1; Location: page index 10, y = 0
OutLine entryChapter4 = new OutLine("Chapter 4: ******* ** ** **** ***", 1, 10, 0);
// add 4th entry to outline
outline.Entry.Add(entryChapter4);

// create 5th entry: Level = 1; Location: page index 20, y = 0
OutLine entryChapter5 = new OutLine("Chapter 5: * ** **** **** ****", 1, 20, 0);
// add 5th entry to outline
outline.Entry.Add(entryChapter5);

// update the new outline
doc.SetOutline(outline);

doc.Save(outputFilePath);
```



Create thumbnail

C#

```
String inputFilePath = Program.RootPath + "\\" + "1.pdf";
PDFDocument doc = new PDFDocument(inputFilePath);
PDFPage page = (PDFPage)doc.GetPage(0);

//  define the size of the thumbnail
Size thumbnailSize = new Size(80, 100);
//  create the thumbnail
REImage thumbnail = page.ConvertToImage(thumbnailSize);
//  do something ...
```

VB

```
Dim inputFilePath As String = Program.RootPath + "\\" + "1.pdf"
Dim doc As PDFDocument = New PDFDocument(inputFilePath)
Dim page As PDFPage = doc.GetPage(0)

' define the size of the thumbnail
Dim thumbnailSize As Size = New Size(80, 100)
' create the thumbnail
Dim thumbnail As Bitmap = page.ConvertToImage(thumbnailSize)
' do something ...
```

Get hyperlink entries in the document

```
String inputFilePath = Program.RootPath + "\\" + "hyperlink.pdf";
PDFDocument doc = new PDFDocument(inputFilePath);

//  get all hyperlink entries in the document
List<HyperlinkEntry> entries = doc.GetHyperlink();

//  report each hyperlink entry in the document
foreach (HyperlinkEntry entry in entries)
{
   //  text content with hyperlink
   Console.WriteLine("Content:  " + entry.TextLst[0]);
   //  URL
   Console.WriteLine("URL:      " + entry.TartgetURL);
   //  page index of the entry
   Console.WriteLine("Page:     " + entry.PageIndex);
   //  boundary of the text content
   Console.WriteLine("Boundary: " + entry.Bounds[0].ToString());
}
```

```
String inputFilePath = Program.RootPath + "\\" + "hyperlink.pdf";
PDFDocument doc = new PDFDocument(inputFilePath);

//  get all hyperlink entries in the 1st page (page index 0)
List<HyperlinkEntry> entries = doc.GetHyperlink(0);

//  report each hyperlink entry in the document
foreach (HyperlinkEntry entry in entries)
{
   //  text content with hyperlink
   Console.WriteLine("Content:  " + entry.TextLst[0]);
   //  URL
   Console.WriteLine("URL:      " + entry.TartgetURL);
   //  boundary of the text content
   Console.WriteLine("Boundary: " + entry.Bounds[0].ToString());
}
```

Change size of a specified page

```
String inputFilePath = Program.RootPath + "\\" + "1.pdf";
String outputFilePath = Program.RootPath + "\\" + "1_new.pdf";

//  open the file
PDFDocument doc = new PDFDocument(inputFilePath);
//  get the 2nd page
PDFPage page = (PDFPage)doc.GetPage(1);
//  set crop region: start point (100, 100), width = 300, height = 400
RectangleF cropArea = new RectangleF(100, 100, 300, 400);
//  crop the page
page.Crop(cropArea);

//  output the new document
doc.Save(outputFilePath);
```

Change size for all pages in a document

```
String inputFilePath = Program.RootPath + "\\" + "1.pdf";
String outputFilePath = Program.RootPath + "\\" + "1_new.pdf";

//  open the file
PDFDocument doc = new PDFDocument(inputFilePath);
//  set crop region: start point (100, 100), width = 300, height = 400
RectangleF cropArea = new RectangleF(100, 100, 300, 400);
//  crop pages
doc.CropAllPages(cropArea);

//  output the new document
doc.Save(outputFilePath);
```

Optimize an exist PDF file

C#

```
String inputFilePath = Program.RootPath + "\\" + "3.pdf";
String outputFilePath = Program.RootPath + "\\" + "3_optimized.pdf";

// create optimizing options
PDFOptimizeOptions ops = new PDFOptimizeOptions();

// -- Options for Monochrome Image --
// to enable downsampling for those images with resolution higher than 300 dpi to 150 dpi
ops.MonochromeImageOptions.DownsampleMode = ImageDownsampleMode.Bicubic;
ops.MonochromeImageOptions.MaxResolutionLimit = 300F;
ops.MonochromeImageOptions.TargetResolution = 150F;
// to change image compression mode to JBIG2
ops.MonochromeImageOptions.KeepCompressionMode = false;
ops.MonochromeImageOptions.Compression = PDFCompression.JBIG2Decode;

// -- Options for Grayscale Image --
// to enable downsampling for those images with resolution higher than 120 dpi to 96 dpi
ops.GrayscaleImageOptions.DownsampleMode = ImageDownsampleMode.Bilinear;
ops.GrayscaleImageOptions.MaxResolutionLimit = 120F;
ops.GrayscaleImageOptions.TargetResolution = 96F;
// to change image compression mode to DCT
ops.GrayscaleImageOptions.KeepCompressionMode = false;
ops.GrayscaleImageOptions.Compression = PDFCompression.DCTDecode;
// set quality level, only available for compression mode DCT
ops.GrayscaleImageOptions.JPEGImageQualityLevel = JPEGImageQualityLevel.High;

// -- Options for Color Image --
// to enable downsampling for those images with resolution higher than 120 dpi to 96 dpi
ops.ColorImageOptions.DownsampleMode = ImageDownsampleMode.Bicubic;
ops.ColorImageOptions.MaxResolutionLimit = 120F;
ops.ColorImageOptions.TargetResolution = 96F;
// to change image compression mode to DCT
ops.ColorImageOptions.KeepCompressionMode = false;
ops.ColorImageOptions.Compression = PDFCompression.DCTDecode;
// set quality level, only available for compression mode DCT
ops.ColorImageOptions.JPEGImageQualityLevel = JPEGImageQualityLevel.Highest;

// apply optimizing
PDFOptimizer.Optimize(inputFilePath, outputFilePath, ops);
```

VB

```

```

Add page thumbnails in a PDF document

Add thumbnail for single page

C#

```
String inputFilePath = Program.RootPath + "\\" + "1.pdf";
String outputFilePath = Program.RootPath + "\\" + "1_thumbnail.pdf";

PDFDocument doc = new PDFDocument(inputFilePath);

//  add thumbnail for the first page
int pageIndex = 0;
PDFThumbnailHandler.UpdateThumbnail(doc, pageIndex);

doc.Save(outputFilePath);
```

VB

Add thumbnail for multi pages

C#

```
String inputFilePath = Program.RootPath + "\\" + "1.pdf";
String outputFilePath = Program.RootPath + "\\" + "1_thumbnail.pdf";

PDFDocument doc = new PDFDocument(inputFilePath);

//  add thumbnail for multi pages
int[] pageIndexes = new int[3] { 0, 3, 5 };
PDFThumbnailHandler.UpdateThumbnail(doc, pageIndexes);

doc.Save(outputFilePath);
```

VB

Add thumbnail for all pages

C#

```
String inputFilePath = Program.RootPath + "\\" + "1.pdf";
String outputFilePath = Program.RootPath + "\\" + "1_thumbnail.pdf";

PDFDocument doc = new PDFDocument(inputFilePath);

//  add thumbnail for all pages
PDFThumbnailHandler.UpdateThumbnail(doc);

doc.Save(outputFilePath);
```

VB

Remove page thumbnails in a PDF document

C#

```
String inputFilePath = Program.RootPath + "\\" + "1_thumbnail.pdf";
String outputFilePath = Program.RootPath + "\\" + "1_no_thumbnail.pdf";

PDFDocument doc = new PDFDocument(inputFilePath);

//  remove thumbnail for all pages
PDFThumbnailHandler.RemoveThumbnail(doc);

doc.Save(outputFilePath);
```

VB

Get page thumbnail in a PDF document

C#

```
String inputFilePath = Program.RootPath + "\\" + "1_thumbnail.pdf";

PDFDocument doc = new PDFDocument(inputFilePath);

//  get thumbnail of the first page if have
int pageIndex = 0;
Bitmap thumbnail = PDFThumbnailHandler.GetThumbnail(doc, pageIndex);
if (thumbnail != null)
{
   Console.WriteLine("Page has thumbnail");
   Console.WriteLine("Width: " + thumbnail.Width + "; Height: " + thumbnail.Height);
}
else
{
   Console.WriteLine("No thumbnail exist");
}
```

VB