

HOWTO: How to merge images into a PDF

Legacy Example NOTICE

This article predates the introduction of our PdfDocument class. PdfDocument has a Combine method which is much more efficient for combining two or more PDF files. Support strongly recommends using this newer class for this task (see links below).

This approach is still totally applicable in many situations, but the result causes the contents of any PDF merged in to be "rasterized" (made into an image).

For PDFs that are already image based, there will not be a significant change in document size, but for vector based (non-raster) PDFs, there could be size increase on disk.

For a more modern approach please consider:

- [HOWTO: Combine Multiple PDFs with Automatic Repair of Damaged PDFs](#)
- [HOWTO: Combine Multiple PDF Files With Fillable Forms \(AcroForms\)](#)
- [HOWTO: Separate a Multi-Page PDF With Automatic Repair of Damaged PDFs](#)

Main Article Content

Here is the code for a console application that merges a list of files into a single PDF. If some of the source files are PDF then you need the PDF Rasterizer and you must register it before the code is called (as shown). In a WinForms application, you would typically do the registration in static constructor of your form.

You need these references in your project

- Atalsoft.dotImage
- Atalsoft.dotImage.Lib
- Atalsoft.dotImage.Pdf
- Atalsoft.dotImage.PdfReader
- Atalsoft.Shared

Here is the code:

C#

HOWTO: How to merge images into a PDF

```
using System; using System.IO; using Atalasoft.Imaging; using Atalasoft.Imaging.Codec; using
Atalasoft.Imaging.Codec.Pdf; namespace Atalasoft.MergePDFSample { class Program { static void
Main(string[] args) { // in order to read PDF, you must call this somewhere in your code //
could be at the beginning of main in a console app, or in your // main form's static
constructor for a winforms app RegisteredDecoders.Decoders.Add(new PdfDecoder()); // merge
file1.pdf, file2.pdf and file3.pdf into output.pdf MergeIntoPdf( new string[] { "file1.pdf",
"file2.pdf", "file3.pdf"}, "output.pdf"); } public static void MergeIntoPdf(string[]
imgFileNames, string outputFile) { ImageSource pdfpages = new
FileSystemImageSource(imgFileNames, true); using (FileStream fs = new FileStream(outputFile,
FileMode.Create, FileAccess.ReadWrite)) { PdfEncoder pdfEncoder = new PdfEncoder();
pdfEncoder.Save(fs, pdfpages, null); } } }
```

VB.NET

```
Imports System Imports System.IO Imports Atalasoft.Imaging Imports Atalasoft.Imaging.Codec
Imports Atalasoft.Imaging.Codec.Pdf Module Module1 Sub Main() ' in order to read PDF, you
must call this somewhere in your code ' could be at the beginning of main in a console app,
or in your ' main form's static constructor for a winforms app
RegisteredDecoders.Decoders.Add(New PdfDecoder()) ' merge file1.pdf, file2.pdf and file3.pdf
into output.pdf MergeIntoPdf(New String() {"f1.pdf", "f2.pdf", "f3.pdf"}, "out.pdf") End Sub
Public Sub MergeIntoPdf(ByVal imgFiles As String(), ByVal outFile As String) Dim pdfpgs As
ImageSource = New FileSystemImageSource(imgFiles, True) Using fs As FileStream = _ New
FileStream(outFile, _ FileMode.Create, FileAccess.ReadWrite) Dim pdfEncoder As PdfEncoder =
New PdfEncoder() pdfEncoder.Save(fs, pdfpgs, Nothing) End Using End Sub End Module
```

Original Article:

Q10203 - HOWTO: How to merge images into a PDF

Atalasoft Knowledge Base

<https://www.atalasoft.com/kb2/KB/50256/HOWTO-How-to-merge-images-into-a-PDF>