Combining vs Separating

This article covers Combining multiple PDFs (single or multipage) into a single multi-page PDF. For help with separating multi-page PDF into multiple single page PDFs (with Automatic Repair) please see the companion article:

HOWTO: Separate a Multi-Page PDF With Automatic Repair of Damaged PDFs

Background

DotImage 10.4 introduced a new feature to PdfDocument: the ability to repair many broken/corrupt PDFs

The Repair classes only take action when exceptions are encountered, so the ability to simply add the repair ability directly to any use of PdfDocument is as simple as choosing the constructor that takes RepairOptions. For PdfDocument, there are two such constructors:

When starting from a full path/filename:

```
public PdfDocument(
 string userPassword,
 string ownerPassword,
 string path,
 RepairOptions repairOptions
)
```

When starting from any seekable stream:

```
 PdfDocumentLoadedProgress pageLoaded,
 RepairOptions repairOptions
)
```

However, the PdfDocument.Combine() static convenience method does not have any option to add RepairOptions, thus it will simply throw an exception when encountering corrupt PDF files.

Solution

The work around for this is to manually combine the pdfs using PdfDocument and iterating pages... it's not hugely complex code, but enough customers have asked for assistance, that support has created a small sample class to assist.

We've attached a static class called PdfDocCombineWithRepair to this KB article. You can include it in your application and then use it like this:

Using File Names

```
string outFile = @"C:\pathTo\outputCombined.pdf";
string[] inFiles = new string[] {@"C:\pathTo\file1.pdf",
@"C:\pathTo\file2.pdf",  ... etc ... };

AtalasoftSupportUtils.PdfDocCombineWithRepair.Combine(outFile, inFiles, new
RepairOptions());
```

Using Streams

```
Stream outStream = either a MemoryStream or FileStream to create the output
pdf
Stream[] inStreams = an array of seekable stream objects containing the pdfs
to combine

AtalasoftSupportUtils.PdfDocCombineWithRepair.Combine(outStream, inStreams,
new RepairOptions());
```

NOTE: in all of the above examples, there's also an overload that takes user and owner passwords - for this simple example, we have assumed all PDFs have the same passwords - you would need to modify the example if you need to pass a unique set of passwords to each PDF

Utility Class and Sample App

There are two attachments to this KB article.

PdfDocCombineWithRepair.cs contains just the CS source for the PdfDocumentCombineWithRepair static class.

PdfDocCombineWithRepair contains that same class, but is actually used within a small console application as a small "proof of concept" for how to use the PDfDocumentCombineWithRepair class.

Bookmarks

Note that this code completely ignores Pdf Bookmarks. There is some POC code stubbed in (and commented out). This code can be enabled if you wish but it's not been fully vetted- it's provided as a proof of concept only.

We have a separate article here HOWTO: Adjust PDF Bookmarks Page Indicies. However, it's using the same POC concepts.

Original Article

Q10429 - HOWTO: Combine Multiple PDFs with Automatic Repair of Damaged PDFs

Last Update

2026-10-22 - TD - (Fixed Outdated Links and heading levels)

Atalasoft Knowledge Base
https://www.atalasoft.com/kb2/KB/50071/HOWTO-Combine-Multiple-PDFs-with-Aut...