

# MERGE JPEG FILES USING ASPOSE.PDF TOOLS



**Copyrights**

All of the content included in this book including all of the text, graphics, photographs, graphs, sounds, data, images, audio, and video clips available on this web site are the property of Aspose Pty Ltd, also known as Aspose or its licensors or content suppliers and is protected by Australian and international copyright laws. The compilation, collection, selection, arrangement, assembly, and coordination of all content available in this book is the exclusive property of Aspose Pty Ltd and protected by Australian and international copyright laws. All of the content available in this book may only be used by information gathering resource. You may not copy, publish, distribute, create derivative works of, or commercially exploit the content of this book, or use this book for any other purpose.

**Trademarks**

"Aspose" is a trademark of Aspose Pty Ltd. All other graphics, logos, product, and service names used in this book by Aspose Pty Ltd are the trademarks of Aspose Pty Ltd. Aspose trademarks may not be used in connection with any third-party products or services or in any manner that disparages or discredits Aspose. All other trademarks, brands, names, and logos appearing in this book are the property of their respective owners.

# Contents

Chapter 1. Aspose PDF capabilities in JPEG format for Users .....	3
Chapter 2. Aspose PDF capabilities in JPEG format for Developers .....	9
Chapter 3. Sample WPF application for .NET 5 .....	14

## Chapter 1. Aspose PDF capabilities in JPEG format for Users

Yes! It's reading time! Books inspire and motivate people to improve themselves and achieve great results. Here you will find the most useful information collected in a compact volume especially for you.

Aspose.com is a leading Australian vendor offering 20+ file format APIs for different platforms including .NET, Java, C++, and Android. Aspose has websites for different tasks. Today we'll talk about Aspose.app. It is free to use a collection of file format apps to convert documents online. In this guidebook, we describe how the Aspose.PDF Product Family and specifically the Merge JPG application work.

We will also consider the possibility of combining JPG files using the Aspose.PDF for .NET library.

If you want to test Aspose.PDF for .NET without the evaluation version limitations, you can also request a 30-day Temporary License. Please refer to [How to get a Temporary License?](#)

First, I will tell you about the PDF file format we work with.

### What is PDF and why it is still actual?

PDF is the standard format for electronic documents. It was originally invented by Adobe for US federal authorities to store their working documents. Now PDF files are used in production management, the sphere of various financial services, lawyers, in publishing, educational institutions.

PDF file saves all fonts, colors, and structure of the source document regardless of software and hardware. Therefore, the PDF document will always be identical to the original - it will look and print as the original document.

PDF documents can be viewed on most popular platforms Windows, MacOS, Linux, Android, IOS using free software such as Adobe Reader DC.

PDF files can be encrypted, and password protected.

We can add images, watermarks, hyperlinks, comments to PDF. In the latest version of PDF

Standard, we also can embed audio, video and 3D Models. Using bookmarks, we can easily navigate through the document.

PDF documents can contain interactive controls. That's why PDF is widely used in PDF forms that can be filled (and then printed) in a PDF viewer or browser.

PDF files are compressed, so they load much faster and require significantly less storage space.

Finally, we can add other files as an attachment to own PDF documents and direct users to this attachment using links.

There are a lot of free and proprietary, open, and closed source, desktop products and web services for creating, viewing, and manipulating PDF documents.

In 2008, Adobe published a Public Patent License to ISO 32000-1 granting royalty-free rights for all patents owned by Adobe that are necessary to make, use, sell, and distribute PDF compliant implementations.

So, all these facts explain why PDF is still actual.

### How can I convert JPEG format using Aspose PDF?

JPEG format is one of the popular raster graphic formats used to store photographs and similar images. Files containing JPEG data usually have the extension (suffix) .jpg, .jfif, .jpe, or .jpeg. However .jpg is the most popular one across all platforms.

Aspose PDF presents various options for converting JPEG files to another format. Specifically, here we will discuss the capabilities of [Aspose.PDF Cross Platform Apps](#).

Let's take a closer look at the features the one of the applications.

### How to merge JPEG files into single PDF document?

Very often users are faced with a situation when they need to merge JPEG images into single PDFs. It happened when the documents scanned or received by e-mail need to be stored. Typically, a document

management system uses the PDF format for data storing.

Our Aspose.PDF Merger can help solve this problem. It works like the Adobe Acrobat command “Combine files in Acrobat” on Windows, but we work online, which means it is available on any operating system and on any device Windows PC, iPhone, Android smartphone, etc.



Fig. 1

## Simply way to merge JPEG files into single PDF

Firstly, use the [Merger JPG to PDF](#) by Aspose PDF. To merge JPEG files into single PDF, please do the following steps:

1. Go to predefined link that points to JPEG to PDF merge operation, drag, and drop your JPEGs in. Look at the fig. 1.



Fig. 2

2. Rearrange the images in the correct order. Look at the fig. 2.
3. Click ‘Merge’ to combine the images. Look at the fig. 3.
4. Download your single document on the next page. Look at the fig. 4.

© [Aspose Pty Ltd](#) 2001-2021. All Rights Reserved.

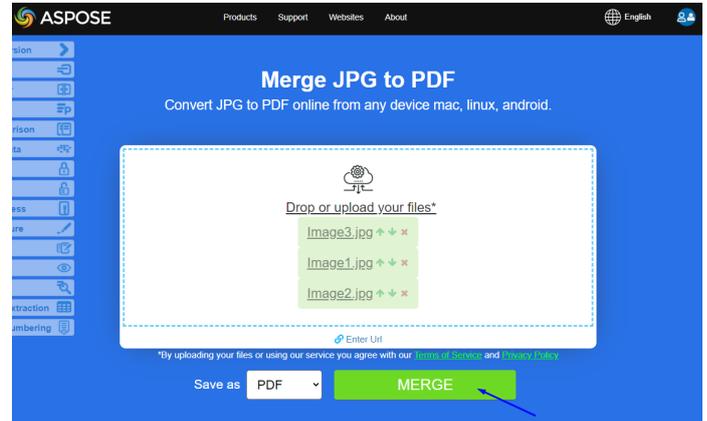


Fig. 3

Each image will be converted to a PDF page with zero margins and preserve orientation. That means that, for example, if your first image is landscape oriented and the next images are portrait-oriented then you will get in PDF first page with landscape and the next pages with portrait orientation.



Fig. 4

When the merge process is complete, you can download or preview the final file.

The file can also be sent by email, but it should be borne in mind that the file is stored on the server for only 24 hours since it created.

## How to merge JPEG files into DOC?

If you need to extract the text from an image so that you can edit it, you can do so with an Optical Character Recognition (OCR) program. These programs scan image files and convert the text so that you can copy and paste it into a Word document. If you just need to insert an image into a Word



Fig. 5

document, you can use [Merge JPG to DOC](#). Look at the figure 5.

## Easy steps to merge JPEG files into DOC format

Utilized in PC word processing, the files with the .DOC extension is most commonly associated with Microsoft Word. Originally a plain-text document, today users can insert hyperlinks, images, and change margin, alignment, and more.

1. Go to predefined link that points to JPEG to

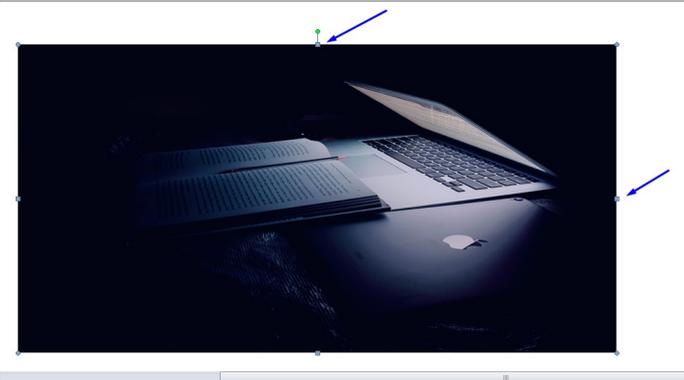


Fig. 6

DOC merge operation, drag and drop your JPEGs in. As you know from previous section.

2. Rearrange the images in the correct order. You can choose one or more images.
3. Click 'Merge' to combine the images.
4. Then download your single document on the next page.

If you select one image, then by default it will occupy the entire position on the page of your document. You can reduce its size.

You can drag the corners of the image to resize it in the document. You can also move the image around by clicking and dragging it. Look at the fig. 6.

If you select two or more images, they will be arranged one after another in the converted document.

Do not forget also about the possibility of sending the result of transforming your document to an email. Look at the fig. 7.

## How to merge JPEG files into DOCX?



Fig. 7

DOCX is an advanced version of the DOC file format and is much more usable and accessible than the latter at any given time.

Unlike the DOC file, the DOCX file is not an extensive file format. To convert your figure to DOCX format follow the link to the application [Merge JPG to Word](#). Look at the fig. 8.



Fig. 8



Fig. 9

## Follow the next steps for converting JPEG into DOCX file:

1. Go to predefined link that points to JPEG to DOCX merge operation, drag and drop your JPEGs in. As you know from previous section.

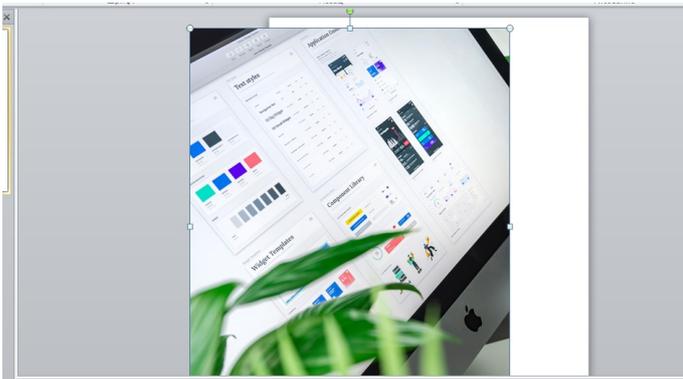


Fig. 10

2. Rearrange the images in the correct order. You can choose one or more images.
3. Click 'Merge' to combine the images.
4. Then download your single document on the next page.



Fig. 11

Having received the conversion result in the DOCX format you can also change the size of the figure, as well as move it around your document.

## How to merge JPEG files into PPTX format?

PPT file is a presentation or editable slideshow created with Microsoft PowerPoint. The presentation consists of separate slides, each of which can contain formatted text, images, video, sound effects. It is also possible to add various effects of switching slides, the appearance of text, objects. Like the PPT predecessor, PPTX files store presentations containing slides of images, text, animation, audio, video, effects, charts, graphs, and more. Where PPT was binary based, PPTX is Open XML based and uses ZIP compression.

Let's check the possibility of converting JPG to PPTX in practice, use [Merge JPG to PowerPoint app](#). Look at the fig. 9.

1. Go to predefined link that points to JPEG to PPTX merge operation, drag, and drop your JPEGs in.
2. Rearrange the images in the correct order. You can choose one or more images.
3. Click 'Merge' to combine the images.
4. Download your single document on the next page.

Having received the conversion result in the PPTX format you can also change the size of the figure, as well as move it around your document. Look at the figure 10.

If you select two or more images, they will be positioned one after the other in the converted document.

And they will also be visible on the left side of the document in the quick slide viewer, in the same order in which they were loaded. Look at the fig. 11.

## How to merge JPEG files into LATEX/TEX format?

Files that have a TEX file extension are text documents created by a program called LaTeX. LaTeX is a typesetting software that enables a user to create



Fig. 12

documents containing different tables, diagrams and scientific drawings in high quality.

So, let's go to the application [Merge JPG to LATEX](#) and follow the instructions:

1. Go to predefined link that points to JPEG to TEX merge operation, drag and drop your JPEGs in. Look at the figure 12.
2. Rearrange the images in the correct order. You can choose one or more images.
3. Click 'Merge' to combine the images.
4. Download your single document on the next page. Look at the figure 13.

The preliminary result can be viewed with Aspose Viewer. For this click 'View Results'.

After that, the Aspose Viewer will open, where you can view your result.

## How to merge JPEG files into HTML format?

You cannot convert a jpg file to HTML as a jpg file is an image file and an HTML file is a file containing tagged code for describing the layout of a web page. But you should have no problem uploading a jpg file

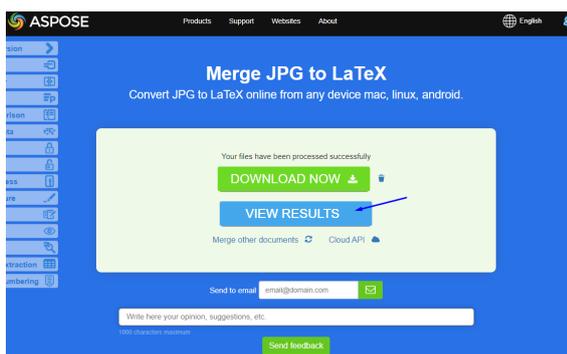


Fig. 13

to display on your website, that would be a standard file type for a website. Aspose PDF may help you with this task. Follow this link [Merge JPG to HTML](#).

1. Go to predefined link that points to JPEG to HTML merge operation, drag, and drop your JPEGs in. Look at the figure 14.
2. Rearrange the images in the correct order. You can choose one or more images.
3. Click 'Merge' to combine the images.
4. Download your single document on the next page.

Note that file will be deleted from our servers after 24 hours and download links will stop working, so it is better to delete your files after this period, because of after already used.

We hope you find this material useful. For the possibility of converting JPEG files programmatically, read the next section of the book and improve your skills together with Aspose PDF. Please do not hesitate to let us know if you require any other information.



Fig. 14

## Chapter 2. Aspose PDF capabilities in JPEG format for Developers

### Merge various JPEG into single PDF with C#

In this section, we are going to provide you with a way to combine all JPEG files in a specific directory into one PDF document. The complete method is displayed in the PDF C# code below. Feel free to copy and paste it into your entire application. Together, we will walk through all the individual steps involved in this way of merging PDFs in C#. For more information and to explore other features of the Aspose.PDF library, please check our [Documentation](#) section. If you want to test Aspose.PDF for .NET without the evaluation version limitations, you can also request a 30-day Temporary License. Please refer to [How to get a Temporary License?](#)

Assume we have a directory with a bunch of JPEG images, and we want to store all those images in one PDF.

In addition, we have a task to combine images to PDF with the same page height and width.

We will be getting the image dimensions and accordingly set the page dimensions of the PDF document with the below steps:

1. Load input image file
2. Get the height and width of the image
3. Set height, width, and margins of a page
4. Save the output PDF file

This code snippet shows how to convert an image to PDF with the same height, width using C#:

```
private static void MergeJpegToPdf()
{
    var jpgFiles = System.IO.Directory.GetFiles(pathToFile, "*.jpg")
        .OrderBy(f => f).ToArray().Take(4);
    var pdfDocument = new Aspose.Pdf.Document();
    var pdfDocumentName = pathToFile + "concatenatedpages" + ".pdf";
    foreach(var path in jpgFiles)
    {
        var page = pdfDocument.Pages.Add();
        System.Drawing.Image srcImage = System.Drawing.Image.FromFile(path);
        // Read Height of input image
        int h = srcImage.Height;
        // Read Width of input image
        int w = srcImage.Width;
        Aspose.Pdf.Image image = new Aspose.Pdf.Image
        {
            File = path
        };
        // Set page dimensions and margins
        page.PageInfo.Height = h;
        page.PageInfo.Width = w;
        page.PageInfo.Margin.Bottom = 0;
        page.PageInfo.Margin.Top = 0;
        page.PageInfo.Margin.Right = 0;
        page.PageInfo.Margin.Left = 0;
        page.Paragraphs.Add(image);
    }
    pdfDocument.Save(pdfDocumentName);
}
```

## Merge JPEG images into DOCX file with C#

In this topic, we are explaining to you a way to combine JPEG files in a DOCX document. The complete method is displayed in the code snippet below. Feel free to copy and paste it into your entire application.

Check the next code for merging JPEG images into your DOCX file:

```
private static void MergeJpegToDocx()
{
    var jpgFiles = System.IO.Directory.GetFiles(pathToFile, "*.jpg")
        .OrderBy(f => f).ToArray().Take(4);
    var pdfDocument = new Aspose.Pdf.Document();
    var docxDocumentName = pathToFile + "concatenatedpages" + ".docx";
    foreach (var path in jpgFiles)
    {
        var page = pdfDocument.Pages.Add();
        System.Drawing.Image srcImage = System.Drawing.Image.FromFile(path);

        // Read Height of input image
        int h = srcImage.Height;

        // Read Width of input image
        int w = srcImage.Width;

        Aspose.Pdf.Image image = new Aspose.Pdf.Image
        {
            File = path
        };

        // Set page dimensions and margins
        page.PageInfo.Height = h;
        page.PageInfo.Width = w;
        page.PageInfo.Margin.Bottom = 0;
        page.PageInfo.Margin.Top = 0;
        page.PageInfo.Margin.Right = 0;
        page.PageInfo.Margin.Left = 0;
        page.Paragraphs.Add(image);
    }
    var options = new DocSaveOptions
    {
        Format = DocSaveOptions.DocFormat.DocX,
        Mode = DocSaveOptions.RecognitionMode.Flow
    };
    pdfDocument.Save(docxDocumentName, options);
}
```

## How to Merge JPEG images into HTML file with C#?

In this article, we will be learning Merger Jpeg formats to HTML file programmatically using C# language. pictures you need and place them in one HTML document.

The main specific feature of such a document combine is that you can at once work with a batch of images. Please carefully learn the proposed snippet code for the possibility of combining JPEG images into a HTML file:  
That is, there is no need to insert each image and save your file every time. You can easily set the number of

```
private static void MergeJpegToHtml()
{
    var jpgFiles=System.IO.Directory.GetFiles(pathToFile, "*.jpg").OrderBy(f=>f).ToArray().Take(4);
    var pdfDocument = new Aspose.Pdf.Document();
    var htmlDocumentName = pathToFile + "concatenatedpages.html";
    foreach (var path in jpgFiles)
    {
        var page = pdfDocument.Pages.Add();
        System.Drawing.Image srcImage = System.Drawing.Image.FromFile(path);
        int h = srcImage.Height;
        int w = srcImage.Width;
        Aspose.Pdf.Image image = new Aspose.Pdf.Image
        {
            File = path
        };

        // Set page dimensions and margins
        page.PageInfo.Height = h;
        page.PageInfo.Width = w;
        page.PageInfo.Margin.Bottom = 0;
        page.PageInfo.Margin.Top = 0;
        page.PageInfo.Margin.Right = 0;
        page.PageInfo.Margin.Left = 0;
        page.Paragraphs.Add(image);
    }

    var options = new HtmlSaveOptions
    {
        PartsEmbeddingMode = HtmlSaveOptions.PartsEmbeddingModes.EmbedAllIntoHtml,
        RasterImagesSavingMode= HtmlSaveOptions.RasterImagesSavingModes
            .AsEmbeddedPartsOfPngPageBackground
    };
    pdfDocument.Save(htmlDocumentName, options);
}
```

## Merge JPEG images into TEX with C#

In this article, we are explain you how to merge all JPEG files into one TEX document using C#

library. The complete method is displayed in C# code below. Feel free to copy and paste it into your entire application.

1. Load input image file
2. Get the height and width of the image

3. Set height, width, and margins of a page

4. Save the output TEX file

Following code snippet shows how to convert an image to TEX with the same page height and width using C#:

```
private static void MergeJpegToTeX()
{
    var jpgFiles = System.IO.Directory.GetFiles(pathToFile, "*.jpg")
        .OrderBy(f => f).ToArray().Take(4);
    var pdfDocument = new Aspose.Pdf.Document();
    var excelFileName = pathToFile + "concatenatedpages" + ".xlsx";
    foreach (var path in jpgFiles)
    {
        var page = pdfDocument.Pages.Add();
        System.Drawing.Image srcImage = System.Drawing.Image.FromFile(path);

        // Read Height of input image
        int h = srcImage.Height;

        // Read Width of input image
        int w = srcImage.Width;

        Aspose.Pdf.Image image = new Aspose.Pdf.Image
        {
            File = path
        };

        // Set page dimensions and margins
        page.PageInfo.Height = h;
        page.PageInfo.Width = w;
        page.PageInfo.Margin.Bottom = 0;
        page.PageInfo.Margin.Top = 0;
        page.PageInfo.Margin.Right = 0;
        page.PageInfo.Margin.Left = 0;
        page.Paragraphs.Add(image);
    }
    var options = new TeXSaveOptions();
    pdfDocument.Save(excelFileName, options);
}
```

## Merging various JPEG into PPTX format with C#

In this article, we will be learning Merger Jpeg formats to PowerPoint file programmatically using C# language. Try the next steps for merging JPEG images into your PPTX file:

1. Load input image file
2. Get the height and width of the image

3. Set height, width, and margins of a page
4. Save the output PPTX file

Please carefully learn the proposed snippet code for the possibility of combining JPEG images into a PPTX file:

```
private static void MergeJpegToPPTX()
{
    var jpgFiles = System.IO.Directory.GetFiles(pathToFile, "JSON Fundamenals_Page_*.jpg")
        .OrderBy(f => f).ToArray().Take(4);
    var pdfDocument = new Aspose.Pdf.Document();
    var powerpointFileName = pathToFile + "powerpoint_demo.pptx";
    foreach (var path in jpgFiles)
    {
        var page = pdfDocument.Pages.Add();
        System.Drawing.Image srcImage = System.Drawing.Image.FromFile(path);

        // Read Height of input image
        int h = srcImage.Height;

        // Read Width of input image
        int w = srcImage.Width;

        Aspose.Pdf.Image image = new Aspose.Pdf.Image
        {
            File = path
        };

        // Set page dimensions and margins
        page.PageInfo.Height = h;
        page.PageInfo.Width = w;
        page.PageInfo.Margin.Bottom = 0;
        page.PageInfo.Margin.Top = 0;
        page.PageInfo.Margin.Right = 0;
        page.PageInfo.Margin.Left = 0;
        page.Paragraphs.Add(image);
    }

    var options = new PptxSaveOptions
    {
        SlidesAsImages = true
    };
    pdfDocument.Save(powerpointFileName, options);
}
```

## Chapter 3. Sample WPF application for .NET 5

In this chapter we will show how to merge JPEGs using simple WPF (.NET) application using Visual Studio 2019.

### Step 1. Create WPF App (.NET)

Open Visual Studio and select **File >> New >> Project** command.

In **Create a New Project windows**, you will see Installed Templates in the right side templates listing. We will use **WPF App (.NET)** and set name **Aspose.Pdf.WpfMerger**.

### Step 2: Add Aspose.Pdf library

Go to Tools->Nuget Package Manager->Package Manager Console. Run command: `Install-Package Aspose.Pdf`

### Step 3: Create Main Window layout

Add XAML layout:

```
<Window x:Class="Aspose.Pdf.WpfMerger.MainWindow"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:local="clr-namespace:Aspose.Pdf.WpfMerger"
    mc:Ignorable="d"
    Loaded="OnLoaded"
    Title="MainWindow" Height="350" Width="525">
    <Window.Resources>

        <!-- Photo Template -->
        <DataTemplate DataType="{x:Type local:Photo}">
            <Grid VerticalAlignment="Center" HorizontalAlignment="Center" Margin="6">
                <!-- Drop Shadow -->
                <Border HorizontalAlignment="Stretch" VerticalAlignment="Stretch" CornerRadius="4"
                    Background="#44000000">
                    <Border.RenderTransform>
                        <TranslateTransform X="5" Y="5" />
                    </Border.RenderTransform>
                    <Border.BitmapEffect>
                        <BlurBitmapEffect Radius="8" />
                    </Border.BitmapEffect>
                </Border>
                <!-- Image Template -->
                <Border Padding="4" Background="White" BorderBrush="#22000000" BorderThickness="1">
                    <StackPanel Orientation="Vertical">
                        <Image Source="{Binding Thumbnail}" />
                    </StackPanel>
                </Border>
            </Grid>
        </DataTemplate>

        <!-- Main photo catalog view -->
        <Style TargetType="{x:Type ListBox}" x:Key="PhotoListBoxStyle">
            <Setter Property="Foreground" Value="White" />
            <Setter Property="Template">
                <Setter.Value>
                    <ControlTemplate TargetType="{x:Type ListBox}">
                        <WrapPanel Margin="5" IsItemsHost="True" Orientation="Horizontal"
                            ItemHeight="{Binding ElementName=ZoomSlider, Path='Value'}"
                            ItemWidth="{Binding ElementName=ZoomSlider, Path='Value'}"
                            VerticalAlignment="Top" HorizontalAlignment="Stretch" />
                    </ControlTemplate>
                </Setter.Value>
            </Setter>
        </Style>

        <!-- Style for an individual generic item -->
        <Style TargetType="{x:Type ListBoxItem}">
            <Setter Property="Background" Value="Transparent" />
        </Style>
    </Window.Resources>

```

```

<Setter Property="Template">
  <Setter.Value>
    <ControlTemplate TargetType="{x:Type ListBoxItem}">
      <Border SnapsToDevicePixels="True" HorizontalAlignment="Stretch" VerticalAlignment="Stretch"
        Background="{TemplateBinding Background}">
        <ContentPresenter />
      </Border>
    <ControlTemplate.Triggers>
      <Trigger Property="IsSelected" Value="True">
        <Setter Property="Background" Value="#445B6249" />
      </Trigger>
    </ControlTemplate.Triggers>
  </ControlTemplate>
</Setter.Value>
</Setter>
</Style>

<!-- Removes dotted rectangle focus -->
<Style TargetType="{x:Type ItemsControl}">
  <Setter Property="Template">
    <Setter.Value>
      <ControlTemplate TargetType="{x:Type ItemsControl}">
        <WrapPanel IsItemsHost="True" />
      </ControlTemplate>
    </Setter.Value>
  </Setter>
</Style>

<!-- For metadata properties pane -->
<Style TargetType="{x:Type GroupBox}">
  <Setter Property="Template">
    <Setter.Value>
      <ControlTemplate TargetType="{x:Type GroupBox}">
        <Grid>
          <Border Background="#AFFFFFFF" CornerRadius="4" BorderBrush="#66000000"
            BorderThickness="1">
            <Border CornerRadius="4" BorderBrush="#88FFFFFF" BorderThickness="1"
              ClipToBounds="true">
              <Border CornerRadius="6" BorderThickness="2" BorderBrush="#ABB497">
                <Border.BitmapEffect>
                  <BlurBitmapEffect Radius="6" />
                </Border.BitmapEffect>
              <Border CornerRadius="6" BorderThickness="2" BorderBrush="#DEF2AD">
                <Border.BitmapEffect>
                  <BlurBitmapEffect Radius="6" />
                </Border.BitmapEffect>
              </Border>
            </Border>
          </Border>
          <ContentPresenter Margin="6" />
        </Grid>
      </ControlTemplate>
    </Setter.Value>
  </Setter>
</Style>

<!-- Default label style -->
<Style TargetType="{x:Type Label}">
  <Setter Property="FontFamily" Value="Segoe UI" />
  <Setter Property="FontSize" Value="11" />
</Style>

<!-- Default style for convert buttons-->
<Style x:Key="ConvertButtonsStyle">
  <Setter Property="Control.FontFamily" Value="Segoe UI" />
  <Setter Property="Control.FontSize" Value="11" />
  <Setter Property="Control.Margin" Value="10,10,10,10" />
</Style>

</Window.Resources>

<!-- Master Container -->
<Grid DataContext="{Binding Source={StaticResource Photos}}"
  Margin="10">

  <Grid.ColumnDefinitions>
    <ColumnDefinition Width="*" />
    <ColumnDefinition Width="250" />

```

```

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>
  <RowDefinition Height="Auto" />
  <RowDefinition Height="*" />
</Grid.RowDefinitions>

<!-- Photo Management Canvas -->
<DockPanel Grid.Column="0" Grid.Row="0" Margin="0,0,0,10">
  <Button DockPanel.Dock="Right" Width="80" Click="OnImagesDirChangeClick">Change</Button>
  <Label DockPanel.Dock="Left" Margin="0,0,10,0">Path:</Label>
  <TextBox Name="ImagesDir" Margin="10,0" />
</DockPanel>

<GroupBox Grid.Column="0" Grid.Row="1">
  <ScrollViewer VerticalScrollBarVisibility="Auto" HorizontalScrollBarVisibility="Disabled">
    <ListBox
      IsSynchronizedWithCurrentItem="True"
      Name="PhotosListBox"
      Style="{StaticResource PhotoListBoxStyle}"
      Margin="5"
      SelectionMode="Extended"
      ItemsSource="{Binding}"
      SelectedIndex="0">
    </ListBox>
  </ScrollViewer>
</GroupBox>

<!-- Splitter -->
<GridSplitter Grid.Column="1" HorizontalAlignment="Left" VerticalAlignment="Stretch" Width="15"
  Background="Transparent" ShowsPreview="True" />

<!-- Right Hand Panel -->
<DockPanel Grid.Row="0" Grid.Column="1" Grid.RowSpan="2"
  Margin="15,0,0,0">

  <!-- Zoom Control -->
  <DockPanel DockPanel.Dock="Bottom" Margin="5">
    <Label DockPanel.Dock="Left">Zoom:</Label>
    <Slider Name="ZoomSlider"
      Margin="10,0,0,0"
      Orientation="Horizontal"
      Minimum="80"
      Maximum="320"
      Value="160"
      TickFrequency="80"
      TickPlacement="BottomRight"
      SmallChange="5"
      LargeChange="20" />
  </DockPanel>

  <!-- Converter Panel -->
  <GroupBox>
    <StackPanel>
      <Button x:Name="button1" Tag="DOCX" Style="{StaticResource ConvertButtonsStyle}"
        Click="OnConvertButtonClick">To DOCX</Button>
      <Button x:Name="button2" Tag="TEX" Style="{StaticResource ConvertButtonsStyle}"
        Click="OnConvertButtonClick">To TeX</Button>
      <Button x:Name="button3" Tag="HTML" Style="{StaticResource ConvertButtonsStyle}"
        Click="OnConvertButtonClick">To HTML</Button>
      <Button x:Name="button4" Tag="PPTX" Style="{StaticResource ConvertButtonsStyle}"
        Click="OnConvertButtonClick">To PPTX</Button>
      <Button x:Name="button5" Tag="PDF" Style="{StaticResource ConvertButtonsStyle}"
        Click="OnConvertButtonClick">To PDF</Button>
      <Label x:Name="statusLabel" Margin="5" Content="Status:"/>
    </StackPanel>
  </GroupBox>
</DockPanel>
</Grid>
</Window>

```

... and add necessary handlers to MainWindow.xaml.cs.

```
using Aspose.Pdf;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace Aspose.Pdf.WpfMerger
{
    /// <summary>
    ///     Interaction logic for MainWindow.xaml
    /// </summary>
    public sealed partial class MainWindow : Window
    {
        public PhotoCollection Photos;

        public MainWindow()
        {
            InitializeComponent();
            Photos = (PhotoCollection)(Application.Current.Resources["Photos"]
                as ObjectDataProvider)?.Data;
            Photos.Path = Environment.CurrentDirectory + "\\images";
        }

        private void OnImagesDirChangeClick(object sender, RoutedEventArgs e)
        {
            Photos.Path = ImagesDir.Text;
        }

        private void OnLoaded(object sender, RoutedEventArgs e)
        {
            ImagesDir.Text = Environment.CurrentDirectory + "\\images";
        }

        private async void OnConvertButtonClick(object sender, RoutedEventArgs e)
        {
            Button btn = (Button)sender;
            ChangeButtonState();

            string docType = btn.Tag.ToString();
            string dirPath = ImagesDir.Text;
            string resultDocumentName = await Task.Run(() => MergeJpeg(docType, dirPath));
            statusLabel.Content = $"Saved as: {resultDocumentName}";

            ChangeButtonState();
        }

        private void ChangeButtonState()
        {
            button1.IsEnabled = !button1.IsEnabled;
            button2.IsEnabled = !button2.IsEnabled;
            button3.IsEnabled = !button3.IsEnabled;
            button4.IsEnabled = !button4.IsEnabled;
            button5.IsEnabled = !button5.IsEnabled;
        }
    }
}
```

```

private string MergeJpeg(string bt, string dirPath)
{
    var jpgFiles = Photos.Select(f => f.Source).OrderBy(f => f).ToArray().Take(4);
    var pdfDocument = new Aspose.Pdf.Document();
    foreach (var path in jpgFiles)
    {
        var page = pdfDocument.Pages.Add();
        System.Drawing.Image srcImage = System.Drawing.Image.FromFile(path);
        int h = srcImage.Height;
        int w = srcImage.Width;

        Aspose.Pdf.Image image = new Aspose.Pdf.Image { File = path };
        page.PageInfo.Height = h;
        page.PageInfo.Width = w;
        page.PageInfo.Margin.Bottom = 0;
        page.PageInfo.Margin.Top = 0;
        page.PageInfo.Margin.Right = 0;
        page.PageInfo.Margin.Left = 0;
        page.Paragraphs.Add(image);
    }
    var resultDocumentName = dirPath;
    SaveOptions options;
    switch (bt)
    {
        case "DOCX":
            resultDocumentName += "concatenatedpages.docx";
            options = new DocSaveOptions
            {
                Format = DocSaveOptions.DocFormat.DocX,
                Mode = DocSaveOptions.RecognitionMode.Textbox
            };
            pdfDocument.Save(resultDocumentName, options);
            break;
        case "HTML":
            resultDocumentName = System.IO.Path.Join(resultDocumentName,
                "concatenatedpages.docx");
            options = new HtmlSaveOptions
            {
                PartsEmbeddingMode = HtmlSaveOptions.PartsEmbeddingModes.EmbedAllIntoHtml,
                RasterImagesSavingMode =
                    HtmlSaveOptions.RasterImagesSavingModes.AsEmbeddedPartsOfPngPageBackground
            };
            pdfDocument.Save(resultDocumentName, options);
            break;
        case "TEX":
            resultDocumentName = System.IO.Path.Join(resultDocumentName,
                "concatenatedpages.tex");
            options = new TeXSaveOptions();
            pdfDocument.Save(resultDocumentName, options);
            break;
        case "PPTX":
            resultDocumentName = System.IO.Path.Join(resultDocumentName,
                "concatenatedpages.pptx");
            options = new PptxSaveOptions();
            pdfDocument.Save(resultDocumentName, options);
            break;
        default:
            resultDocumentName = System.IO.Path.Join(resultDocumentName,
                "concatenatedpages.pdf");
            pdfDocument.Save(resultDocumentName);
            break;
    }
    return resultDocumentName;
}
}
}

```

Also we need additional classes **Photo** and **PhotoCollection**.

```
using System;
using System.Windows.Media.Imaging;

namespace Aspose.Pdf.WpfMerger
{
    /// <summary>
    /// This class describes a single photo - its location, the image and thumbnail
    /// </summary>
    public class Photo
    {
        private readonly Uri _source;

        public Photo(string path)
        {
            Source = path;
            _source = new Uri(path);
            Image = BitmapFrame.Create(_source);
            Thumbnail = Image.Thumbnail ?? GenerateThumbnail(path);
        }

        private BitmapSource GenerateThumbnail(string path)
        {
            BitmapImage bmpImage = new BitmapImage();
            bmpImage.BeginInit();
            bmpImage.UriSource = new Uri(path);
            bmpImage.DecodePixelWidth = 120;
            bmpImage.EndInit();
            return bmpImage;
        }

        public string Source { get; }
        public BitmapFrame Image { get; set; }
        public BitmapSource Thumbnail { get; }
        public override string ToString() => _source.ToString();
    }
}
```

```
using System.Collections.ObjectModel;
using System.IO;
using System.Windows;

namespace Aspose.Pdf.WpfMerger
{
    /// <summary>
    /// This class represents a collection of photos in a directory.
    /// </summary>
    public class PhotoCollection : ObservableCollection<Photo>
    {
        private DirectoryInfo _directory;

        public PhotoCollection()
        {
        }

        public PhotoCollection(string path) : this(new DirectoryInfo(path))
        {
        }

        public PhotoCollection(DirectoryInfo directory)
        {
            _directory = directory;
            Update();
        }
    }
}
```

```

public string Path
{
    set
    {
        _directory = new DirectoryInfo(value);
        Update();
    }
    get { return _directory.FullName; }
}

public DirectoryInfo Directory
{
    set
    {
        _directory = value;
        Update();
    }
    get { return _directory; }
}

private void Update()
{
    Clear();
    try
    {
        foreach (var f in _directory.GetFiles("*.jpg"))
            Add(new Photo(f.FullName));
    }
    catch (DirectoryNotFoundException)
    {
        MessageBox.Show("No Such Directory");
    }
}
}
}
}

```

The last action— we add Photos as resource of application:

```

<Application x:Class="Aspose.Pdf.WpfMerger.App"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:local="clr-namespace:Aspose.Pdf.WpfMerger"
    StartupUri="MainWindow.xaml">
    <Application.Resources>
        <ObjectDataProvider x:Key="Photos" ObjectType="{x:Type local:PhotoCollection}" />
    </Application.Resources>
</Application>

```

After running the application we got windows as show on figure below. To set a folder with images user should enter path and click **“Change”** button. To perform merging user can press **“To...”** button.

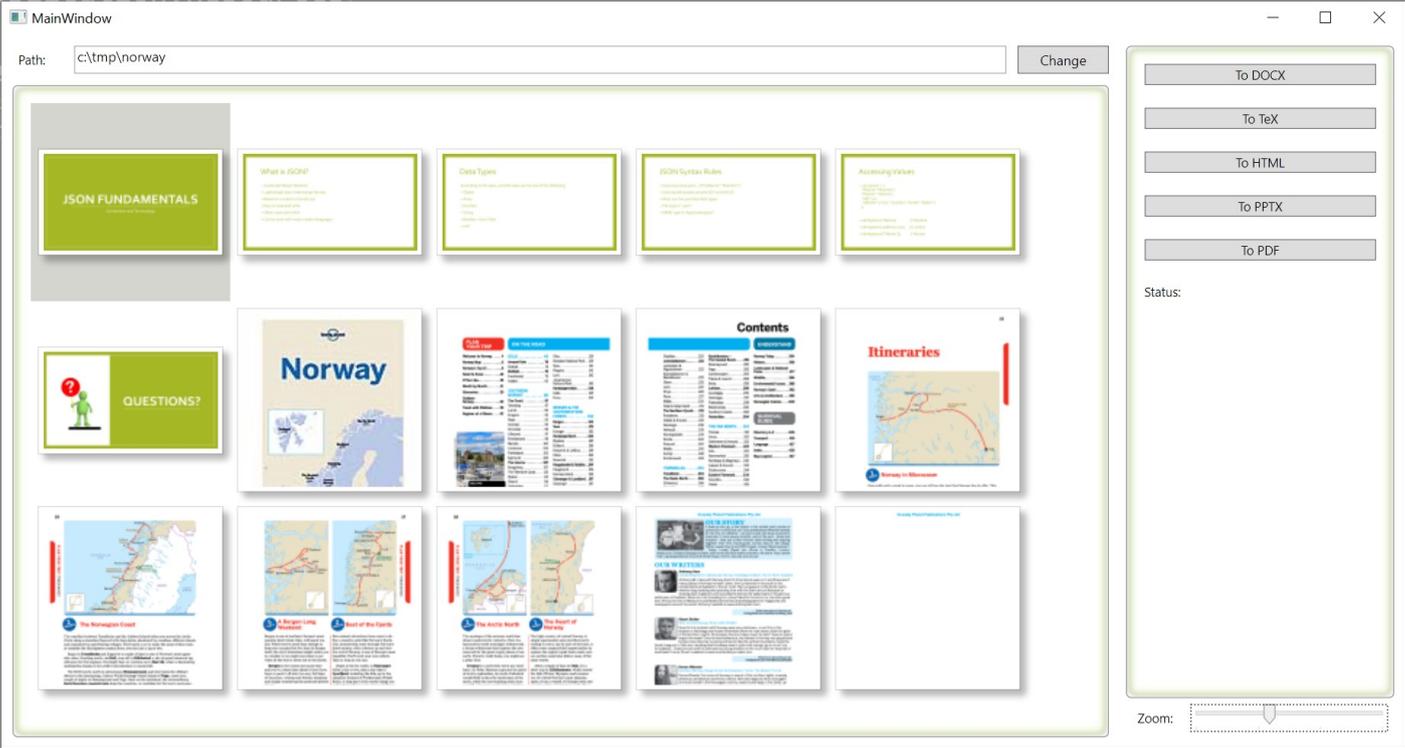


Fig. 15

## References:

- <https://products.aspose.com/pdf> — product page
- <https://products.aspose.app/pdf/merger/jpg> — free applications
- <https://docs.aspose.com/pdf> — documentation for developers
- <https://forum.aspose.com/c/pdf/10> — free support forum