

www.bytescout.com

How to create multipage tiff to PDF in C# and ByteScout PDF SDK

How to create multipage tiff to PDF in C#

The code below will help you to implement an C# app to create multipage tiff to PDF. ByteScout PDF SDK is the SDK for pdf documents generation, modification and updates. Can also generate and fill PDF forms. Provides support for text (fonts, style, size, font family), layers, pdf form fields, vector and raster drawings. It can be used to create multipage tiff to PDF using C#.

This rich sample source code in C# for ByteScout PDF SDK includes the number of functions and options you should do calling the API to create multipage tiff to PDF. This C# sample code is all you need for your app. Just copy and paste the code, add references (if needs to) and you are all set! Further enhancement of the code will make it more vigorous.

Download free trial version of ByteScout PDF SDK from our website with this and other source code samples for C#.

FOR MORE INFORMATION AND FREE TRIAL:

[Download Free Trial SDK \(on-premise version\)](#)

[Read more about ByteScout PDF SDK](#)

[Explore API Documentation](#)

[Get Free Training for ByteScout PDF SDK](#)

[Get Free API key for Web API](#)

[visit www.ByteScout.com](#)

Source Code Files:

Program.cs

```
using System;
using System.Diagnostics;
using System.Drawing;
using System.Drawing.Imaging;
using System.IO;

using Bytescout.PDF;
using Image = Bytescout.PDF.Image;

namespace Images
{
    /// <summary>
    /// This example demonstrates how to create PDF from multipage TIFF image.
    /// </summary>
    class Program
    {
        static void Main()
        {
            // Create new document
            Document pdfDocument = new Document();
            pdfDocument.RegistrationName = "demo";
            pdfDocument.RegistrationKey = "demo";

            // Get Tiff file in Bitmap
            Bitmap multipageTiffImage = (Bitmap)System.Drawing.Image.FromFile("MultipageTiffImage.tiff");

            // Tiff images can contain multiple images named as frame
            int count = multipageTiffImage.GetFrameCount(FrameDimension.Page);
            for (int idx = 0; idx < count; idx++)
            {
                // save each frame to a bytestream
                multipageTiffImage.SelectActiveFrame(FrameDimension.Page, idx);
                MemoryStream byteStream = new MemoryStream();
                multipageTiffImage.Save(byteStream, ImageFormat.Tiff);

                // Fix Image Orientation
                var sysImage = FixImageOrientation(byteStream);

                using (sysImage)
                {
                    var image = new Image(sysImage, ImageCompression.DCT, 75);
                    float w = sysImage.Width / sysImage.HorizontalResolution * 72f;
                    float h = sysImage.Height / sysImage.VerticalResolution * 72f;

                    // Create PDF page
                    var page = new Page(w, h);

                    var canvas = page.Canvas;
                    canvas.DrawImage(image, 0, 0, w, h);

                    // Add PDF document
                    pdfDocument.Pages.Add(page);
                }
            }

            // Save document to file
            pdfDocument.Save("result.pdf");
        }
    }
}
```

```
// Cleanup
pdfDocument.Dispose();

// Open result document in default associated application (for demo purpose)
ProcessStartInfo processStartInfo = new ProcessStartInfo("result.pdf");
processStartInfo.UseShellExecute = true;
Process.Start(processStartInfo);
}

/// <summary>
/// Fix image orientation
/// </summary>
static System.Drawing.Image FixImageOrientation(Stream fileStream)
{
    System.Drawing.Image sysImage = System.Drawing.Image.FromStream(fileStream);

    // fix orientation by EXIF rotation tag
    const int exifRotationTag = 0x0112;
    int[] propertyItemIDs = sysImage.PropertyIdList;

    int found = Array.BinarySearch(propertyItemIDs, 0, propertyItemIDs.Length,
        if (found > -1)
    {
        PropertyItem pi = sysImage.GetPropertyItem(exifRotationTag);
        int orientation = pi.Value[0];

        switch (orientation)
        {
            case 2:
                sysImage.RotateFlip(RotateFlipType.RotateNoneFlipX);
                break;
            case 3:
                sysImage.RotateFlip(RotateFlipType.Rotate180FlipNone);
                break;
            case 4:
                sysImage.RotateFlip(RotateFlipType.RotateNoneFlipY);
                break;
            case 5:
                sysImage.RotateFlip(RotateFlipType.Rotate90FlipX);
                break;
            case 6:
                sysImage.RotateFlip(RotateFlipType.Rotate90FlipNone);
                break;
            case 7:
                sysImage.RotateFlip(RotateFlipType.Rotate270FlipX);
                break;
            case 8:
                sysImage.RotateFlip(RotateFlipType.Rotate270FlipNone);
                break;
            default:
                break;
        }

        sysImage.RemovePropertyItem(0x0112);
    }

    return sysImage;
}
```

```
    }  
}
```

VIDEO

<https://www.youtube.com/watch?v=gdsQ0EAqwGQ>

ON-PREMISE OFFLINE SDK

[60 Day Free Trial](#) or [Visit ByteScout PDF SDK Home Page](#)
[Explore ByteScout PDF SDK Documentation](#)
[Explore Samples](#)
[Sign Up for ByteScout PDF SDK Online Training](#)

ON-DEMAND REST WEB API

[Get Your API Key](#)
[Explore Web API Docs](#)
[Explore Web API Samples](#)

[visit www.ByteScout.com](#)

[visit www.PDF.co](#)

[www.bytescout.com](#)