

How to create multipage tiff to pdf with pdf sdk in C# with ByteScout PDF Suite

Learn to code in C# to create multipage tiff to pdf with pdf sdk with this step-by-step tutorial

The sample source code below will teach you how to create multipage tiff to pdf with pdf sdk in C#. What is ByteScout PDF Suite? It is the set that includes 6 SDK products to work with PDF from generating rich PDF reports to extracting data from PDF documents and converting them to HTML. This bundle includes PDF (Generator) SDK, PDF Renderer SDK, PDF Extractor SDK, PDF to HTML SDK, PDF Viewer SDK and PDF Generator SDK for Javascript. It can help you to create multipage tiff to pdf with pdf sdk in your C# application.

This prolific sample source code in C# for ByteScout PDF Suite contains various functions and other necessary options you should do calling the API to create multipage tiff to pdf with pdf sdk. Follow the instructions from scratch to work and copy the C# code. Further improvement of the code will make it more robust.

ByteScout provides the free trial version of ByteScout PDF Suite along with the documentation and source code samples.

FOR MORE INFORMATION AND FREE TRIAL:

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

[Read more about ByteScout PDF Suite](#)

[Explore API Documentation](#)

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

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

[visit www.Bytescout.com](http://www.Bytescout.com)

Source Code Files:

```

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.tif");

            // 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 purposes)
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=NEwNs2b9YN8>

ON-PREMISE OFFLINE SDK

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

ON-DEMAND REST WEB API

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

[visit www.ByteScout.com](http://www.ByteScout.com)

[visit www.PDF.co](http://www.PDF.co)

www.bytescout.com