

How to use PDF layers in C# and ByteScout PDF SDK

How to code in C# to use PDF layers with this step-by-step tutorial

The code below will help you to implement an C# app to use PDF layers. What is ByteScout PDF SDK? It is the pdf library that can create, update and modify PDF files. Supports text with fonts and style selections, layers, form fields, drawing lines and objects, automatic tables, images. Can be used to create and fill pdf forms. It can help you to use PDF layers in your C# application.

You will save a lot of time on writing and testing code as you may just take the C# code from ByteScout PDF SDK for use PDF layers below and use it in your application. In order to implement the functionality, you should copy and paste this code for C# below into your code editor with your app, compile and run your application. This basic programming language sample code for C# will do the whole work for you to use PDF layers.

Free trial version of ByteScout PDF SDK is available for download from our website. Get it to try other source code samples for C#.

C# - Program.cs

```
using System.Diagnostics;
using System.Drawing;
using Bytescout.PDF;
using SolidBrush = Bytescout.PDF.SolidBrush;

namespace Layers
{
    ///
    /// This example demonstrates how to add layers with optional content.
    ///

    class Program
    {
        static void Main()
        {
            // Create new document
            Document pdfDocument = new Document();
            pdfDocument.RegistrationName = "demo";
            pdfDocument.RegistrationKey = "demo";
            // Add page
            Page page = new Page(PaperFormat.A4);
            pdfDocument.Pages.Add(page);

            Canvas canvas = page.Canvas;
```

```

// Add main layer
Layer allContents = new Layer("Main Layer");
canvas.BeginMarkedContent(allContents);

// Add sub-layer
Layer layer1 = new Layer("Square");
canvas.BeginMarkedContent(layer1);
canvas.DrawRectangle(new SolidBrush(new ColorRGB(0, 255, 0)), 20, 20,
100, 100);
canvas.EndMarkedContent();

// Add second sub-layer
Layer layer2 = new Layer("Circle");
canvas.BeginMarkedContent(layer2);
canvas.DrawCircle(new SolidBrush(new ColorRGB(255, 0, 0)), 70, 180, 50);
canvas.EndMarkedContent();

// Add third sub-layer
Layer layer3 = new Layer("Triangle");
canvas.BeginMarkedContent(layer3);
canvas.DrawPolygon(new SolidBrush(new ColorRGB(0, 0, 255)), new PointF[]
{ new PointF(20, 340), new PointF(70, 240), new PointF(120, 340) });
canvas.EndMarkedContent();

canvas.EndMarkedContent();

pdfDocument.OptionalContents.Layers.Add(layer1);
pdfDocument.OptionalContents.Layers.Add(layer2);
pdfDocument.OptionalContents.Layers.Add(layer3);
pdfDocument.OptionalContents.Layers.Add(allContents);

// Configure layers

// Group sub-layers
OptionalContentGroup group = new OptionalContentGroup();
group.Add(new OptionalContentGroupLayer(layer1));
group.Add(new OptionalContentGroupLayer(layer2));
group.Add(new OptionalContentGroupLayer(layer3));

// Order sub-layers
pdfDocument.OptionalContents.Configuration.Order.Add(new
OptionalContentGroupLayer(allContents));
pdfDocument.OptionalContents.Configuration.Order.Add(group);

// Make the third layer invisible by default (for example)
pdfDocument.OptionalContents.Configuration.OFF.Add(layer3);

// Force the PDF viewer to show the layers panel initially
pdfDocument.PageMode = PageMode.OptionalContent;

// 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);  
    }  
}
```

FOR MORE INFORMATION AND FREE TRIAL:

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

[Read more about ByteScout PDF SDK](#)

[Explore documentation](#)

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

or

[Get Your Free API Key for www.PDF.co Web API](#)