

www.bytescout.com

How to add custom images to pdf document with barcode sdk in C# with ByteScout Premium Suite

This code in C# shows how to add custom images to pdf document with barcode sdk with this how to tutorial

The code displayed below will guide you to install an C# app to add custom images to pdf document with barcode sdk. ByteScout Premium Suite is the set that includes 12 SDK products from ByteScout including tools and components for PDF, barcodes, spreadsheets, screen video recording. It can add custom images to pdf document with barcode sdk in C#.

Want to quickly learn? This fast application programming interfaces of ByteScout Premium Suite for C# plus the guidelines and the code below will help you quickly learn how to add custom images to pdf document with barcode sdk. Follow the instructions from scratch to work and copy the C# code. Complete and detailed tutorials and documentation are available along with installed ByteScout Premium Suite if you'd like to learn more about the topic and the details of the API.

ByteScout provides the free trial version of ByteScout Premium 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 Premium Suite](#)

[Explore API Documentation](#)

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

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

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

Source Code Files:

Program.cs

```
using System.Diagnostics;
using System.Drawing;
using Bytescout.BarCode;

// This example demonstrates adding of barcode together with some custom images (e.g. target marks)
// Also shows the manual calculation of barcode size from inches to pixels and document resolution
// Task: place Code39 barcode of 2" x 3/8" size at 2" from top-right corner and two target marks

namespace AddBarcodeToPdfDocument
{
    class Program
    {
        static void Main(string[] args)
        {
            // Create Barcode instance and set it up
            Barcode barcode = new Barcode("demo", "demo");
            barcode.Symbology = SymbologyType.Code39;
            barcode.Value = "00090112";
            barcode.DrawCaption = true;
            barcode.CaptionFont = new Font("Courier", 12f, FontStyle.Bold, GraphicsUnit.Point);
            barcode.DrawQuietZones = false;
            barcode.ResolutionX = barcode.ResolutionY = 300; // High resolution for better quality

            // Compute barcode image dimension from inches to pixels at 300 DPI:
            int barHeight = (int) (3f/8*300); // = 3/8" * 300 DPI = height of bars
            int captionHeight = (int) barcode.CaptionFont.GetHeight(barcode.ResolutionY);
            int captionGap = (int) (1f/8*300); // = 1/8" gap

            int barcodeImageWidth = 2*300; // = 2" * 300 DPI = width of barcode
            int barcodeImageHeight = barHeight + captionGap + captionHeight + 28; // 28 = caption font height + gap + border

            // Get final barcode image:
            barcode.BarHeight = barHeight;
            barcode.FitInto(barcodeImageWidth, barcodeImageHeight, UnitOfMeasure.Pixel);
            Image barcodeImage = barcode.GetImage();

            // Arrays of images and points to apply to PDF document
            Image[] images = new Image[3];
            Point[] points = new Point[3];

            // Compute coordinates of barcode image and target marks at 72 DPI
            // (page size is 8.5" x 11", PDF document resolution is always 72 DPI):

            int x = (int) (8.5f*72 - 2*72 - barcodeImageWidth/300f*72); // = page width - barcode width - margin
            int y = 2*72; // = 2" - Y coordinate to put the barcode image

            // Put barcode image into array
            images[0] = barcodeImage;
            points[0] = new Point(x, y);

            // Load target mark image.
            // TargetMark.png is 1/8" 300 DPI image (38x38 pixels)
```

```

Image targetMarkImage = Image.FromFile("TargetMark.png");

// Coordinates of top-right target mark
x = (int)(8.5f*72 - 1*72 - 1f/8*72); // = pageWidth - 1" - target mark width
y = 1*72; // = 1" - Y coordinate to put the barcode image

// Put first target mark image into array
images[1] = targetMarkImage;
points[1] = new Point(x, y);

// Coordinates of bottom-left target mark
x = 1*72; // = 1"
y = (int)(11f*72 - 1*72 - 1f/8*72); // = page height - 1" - target mark height

// Put second target mark image into array
images[2] = targetMarkImage;
points[2] = new Point(x, y);

// Draw images on all PDF document pages.
barcode.DrawImagesToPDF("wikipedia.pdf", -1 /*all pages*/, images, points,

// Cleanup
barcodeImage.Dispose();
targetMarkImage.Dispose();
barcode.Dispose();

// Open the result document in default associated application
Process.Start("result.pdf");

}

}
}

```

VIDEO

<https://www.youtube.com/watch?v=NEwNs2b9YN8>

ON-PREMISE OFFLINE SDK

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

[Get Your API Key](#)

[Explore Web API Docs](#)

[Explore Web API Samples](#)

[visit www.ByteScout.com](#)

[visit www.PDF.co](#)

[www.bytescout.com](#)