

# silent PDF printing in C# and ByteScout PDF Renderer SDK

## How to code silent PDF printing in C#: How-To tutorial

Here you may find thousands pre-made source code pieces for easy implementation in your own programming C# projects. ByteScout PDF Renderer SDK was made to help with silent PDF printing in C#. ByteScout PDF Renderer SDK is the library that renders PDF into high quality images and thumbnails. Includes various functions like batch processing, PNG, TIFF output. Can be used from web and desktop applications.

The SDK samples like this one below explain how to quickly make your application do silent PDF printing in C# with the help of ByteScout PDF Renderer SDK. C# sample code is all you need: copy and paste the code to your C# application's code editor, add a reference to ByteScout PDF Renderer SDK (if you haven't added yet) and you are ready to go! You can use these C# sample examples in one or many applications.

Free trial version of ByteScout PDF Renderer SDK is available on our website. Get it to try other samples for C#.

C# - Program.cs

```
using System;

using Bytescout.PDFRenderer;

namespace SilentPrinting
{
    class Program
    {
        static void Main(string[] args)
        {
            const string PrinterName = "Microsoft Print to PDF";

            // Create an instance of Bytescout.PDFRenderer.DocumentPrinter object and
            register it.
            DocumentPrinter documentPrinter = new DocumentPrinter();
            documentPrinter.RegistrationName = "demo";
            documentPrinter.RegistrationKey = "demo";

            // Load PDF document.
            documentPrinter.LoadDocumentFromFile("multipage.pdf");

            // Change some printing options for demo purposes
            // (note, some options might be not supported by the printer):
```

```
// Change paper orientation
documentPrinter.Landscape = true;
// Set number of copies
documentPrinter.Copies = 2;
// Set collation
documentPrinter.Collate = true;
// Force black and white printing
documentPrinter.Color = false;

// Print all document pages to the specified printer
documentPrinter.PrintDocument(PrinterName);

// To print specific pages use overridden methods allowing to define
pages to print:
//documentPrinter.PrintDocument(PrinterName, 2, 4); // printer pages from
3 to 5
//documentPrinter.PrintDocument(PrinterName, new int[] {0, 2, 3, 4, 6});
// print specific pages
//documentPrinter.PrintDocument(PrinterName, "1,3-5,7-");

documentPrinter.Dispose();
}
}
```

---

FOR MORE INFORMATION AND FREE TRIAL:

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

[Read more about ByteScout PDF Renderer SDK](#)

[Explore documentation](#)

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

or

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