

How to reduce CPU usage in C# and ByteScout BarCode Reader SDK

Tutorial on how to reduce CPU usage in C#

The coding tutorials are designed to help you test the features without need to write your own code. What is ByteScout BarCode Reader SDK? It is the SDK for reading of barcodes from PDF, images and live camera or video. Almost every common type like Code 39, Code 128, GS1, UPC, QR Code, Datamatrix, PDF417 and many others are supported. Supports noisy and defective images and docs. Includes optional documents splitter and merger for pdf and tiff based on found barcodess. Batch mode is supported for superior performance using multiple threads. Decoded values are easily exported to JSON, CSV, XML and to custom format. It can help you to reduce CPU usage in your C# application.

This code snippet below for ByteScout BarCode Reader SDK works best when you need to quickly reduce CPU usage in your C# application. Follow the instructions from the scratch to work and copy the C# code. Enjoy writing a code with ready-to-use sample C# codes.

Trial version of ByteScout BarCode Reader SDK is available for free. Source code samples are included to help you with your C# app.

FOR MORE INFORMATION AND FREE TRIAL:

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

[Read more about ByteScout BarCode Reader SDK](#)

[Explore API Documentation](#)

[Get Free Training for ByteScout BarCode Reader SDK](#)

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

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

Source Code Files:

```
using System;
using Bytescout.BarCodeReader;

namespace ReduceCPUUsage
{
    class Program
    {
        static void Main()
        {
            // Barcode reader instance
            Reader reader = new Reader();
            reader.RegistrationName = "demo";
            reader.RegistrationKey = "demo";

            /*
             If you are reading barcodes from PDF then you may reduce CPU and RAM load
             - instead of using All1D or All2D barcode types, set it to the specific
             - reduce PDF rendering resolution to 200-150 dpi (depends on your document)
             - set specific pages to read barcodes from. If you have barcodes on 2 + pages
             - if barcodes are always located / printed in the same corner then also
            */

            // Input filename
            string inputFileName = "barcode_multipage.pdf";

            // Set specific barcode type to read
            reader.BarcodeTypesToFind.Code128 = true;

            // Reduce PDF rendering resolution
            reader.PDFRenderingResolution = 150;

            // Set specific area to read from
            reader.CustomAreaLeft = 407;
            reader.CustomAreaTop = 494;
            reader.CustomAreaHeight = 605;
            reader.CustomAreaWidth = 999;

            // Set specific page to read from along with filename
            reader.ReadFromPdfFilePage(inputFileName, 1, 1);

            // Get all found barcodes
            FoundBarcode[] barcodes = reader.FoundBarcodes;

            // Display found barcodes
            Console.WriteLine("Reading barcode(s) from PDF file...");

            foreach (FoundBarcode barcode in barcodes)
            {
                Console.WriteLine("Found Barcode - Type: '{0}', Value: '{1}'", barcode.Type, barcode.Value);
            }

            // Cleanup
            reader.Dispose();

            Console.WriteLine();
        }
    }
}
```

```
        Console.WriteLine("Press any key to continue.");  
        Console.ReadKey();  
    }  
}  
}
```

VIDEO

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

ON-PREMISE OFFLINE SDK

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