

searchablepdfmaker progress indication with PDF extractor SDK in C# with ByteScout PDF Suite

Simple tutorial on how to do searchablepdfmaker progress indication with PDF extractor SDK in C#

This page explains the steps and algorithm of implementing searchablepdfmaker progress indication with PDF extractor SDK and how to make it work in your application. ByteScout PDF Suite was created to assist searchablepdfmaker progress indication with PDF extractor SDK in C#. ByteScout PDF Suite is the bundle that provides six different SDK libraries 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.

The below SDK samples describe how to quickly make your application do searchablepdfmaker progress indication with PDF extractor SDK in C# with the help of ByteScout PDF Suite. To use searchablepdfmaker progress indication with PDF extractor SDK in your C# project or application just copy & paste the code and then run your app! Further improvement of the code will make it more robust.

Visit our website to get a free trial version of ByteScout PDF Suite. Free trial contains many of source code samples to help you with your C# project.

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:

Program.cs

```
using System;
using System.Diagnostics;
using Bytescout.PDFExtractor;

namespace SearchablePDFMakerProgressChangeEvent
{
    class Program
    {
        static void Main(string[] args)
        {
            try
            {
                using (var searchablePDFMaker = new SearchablePDFMaker("demo", "demo"))
                {
                    // Load sample PDF document
                    searchablePDFMaker.LoadDocumentFromFile("sample_ocr.pdf");

                    // Extractor Progress event
                    Console.WriteLine("Searchable PDF making in progress: \n");
                    searchablePDFMaker.ProgressChanged += SearchablePDF_ProgressChanged;

                    // Set the location of OCR language data files
                    searchablePDFMaker.OCRLanguageDataFolder = @"c:\Program Files\Bytescout";

                    // Set OCR language
                    searchablePDFMaker.OCRLanguage = "eng"; // "eng" for english, "deu" for german
                    // Find more language files at https://github.com/bytescout/ocrdata

                    // Set PDF document rendering resolution
                    searchablePDFMaker.OCRResolution = 300;

                    // Save extracted text to file
                    searchablePDFMaker.MakePDFSearchable("output.pdf");

                    // Open result document in default associated application (for demo)
                    ProcessStartInfo processStartInfo = new ProcessStartInfo("output.pdf");
                    processStartInfo.UseShellExecute = true;
                    Process.Start(processStartInfo);
                }
            }
            catch (Exception ex)
            {
                Console.WriteLine(ex.Message);
            }

            Console.WriteLine("\n\n Press enter key to exit...");
            Console.ReadLine();
        }
    }

    /// <summary>
    /// Handle progress change event

```

```

/// </summary>
private static void SearchablePDF_ProgressChanged(object sender, OngoingOperat
{
    drawTextProgressBar(Convert.ToInt32(progress), 100);
}

/// <summary>
/// Display progress bar
/// </summary>
private static void drawTextProgressBar(int progress, int total)
{
    //draw empty progress bar
    Console.CursorLeft = 0;
    Console.Write("["); //start
    Console.CursorLeft = 32;
    Console.Write("]"); //end
    Console.CursorLeft = 1;
    float onechunk = 30.0f / total;

    //draw filled part
    int position = 1;
    for (int i = 0; i < onechunk * progress; i++)
    {
        Console.BackgroundColor = ConsoleColor.Green;
        Console.CursorLeft = position++;
        Console.Write(" ");
    }

    //draw unfilled part
    for (int i = position; i <= 31; i++)
    {
        Console.BackgroundColor = ConsoleColor.Gray;
        Console.CursorLeft = position++;
        Console.Write(" ");
    }

    //draw totals
    Console.CursorLeft = 35;
    Console.BackgroundColor = ConsoleColor.Black;
    Console.Write(progress.ToString() + " of " + total.ToString() + " "); //
}
}
}

```

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 \[www.PDF.co\]\(http://www.PDF.co\)](#)

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