

How to index PDF documents in folder with PDF extractor SDK in C# using ByteScout Data Extraction Suite

Learning is essential in computer world and the tutorial below will demonstrate how to index PDF documents in folder with PDF extractor SDK in C#

An easy to understand guide on how to index PDF documents in folder with PDF extractor SDK in C# with this source code sample. ByteScout Data Extraction Suite is the bundle that includes three SDK tools for data extraction from PDF, scans, images and from spreadsheets: PDF Extractor SDK, Data Extraction SDK, Barcode Reader SDK. It can be applied to index PDF documents in folder with PDF extractor SDK using C#.

The following code snippet for ByteScout Data Extraction Suite works best when you need to quickly index PDF documents in folder with PDF extractor SDK in your C# application. Simply copy and paste in your C# project or application you and then run your app! Use of ByteScout Data Extraction Suite in C# is also described in the documentation included along with the product.

All these programming tutorials along with source code samples and ByteScout free trial version are available for download from our website.

FOR MORE INFORMATION AND FREE TRIAL:

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

[Read more about ByteScout Data Extraction Suite](#)

[Explore API Documentation](#)

[Get Free Training for ByteScout Data Extraction Suite](#)

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

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

Source Code Files:

```
using Bytescout.PDFExtractor;
using System;
using System.Collections.Generic;
using System.IO;

namespace IndexDocsInFolder
{
    class Program
    {
        static void Main(string[] args)
        {
            try
            {
                // Output file list
                var lstAllFilesInfo = new List<FileIndexOutput>();

                // Get all files inside directory
                var allFiles = Directory.GetFiles(@".\Files", "*.*");

                // Iterate all files, and get details
                foreach (var itmFile in allFiles)
                {
                    // Get basic file information
                    FileInfo fileInfo = new FileInfo(itmFile);

                    // Check whether file is supported
                    if (_IsFileSupported(fileInfo))
                    {
                        // Fill file index model
                        var oFileIndex = new FileIndexOutput();
                        oFileIndex.fileName = fileInfo.Name;
                        oFileIndex.fileDate = fileInfo.CreationTime;
                        oFileIndex.content = _GetFileContent(fileInfo);

                        // Add to final list
                        lstAllFilesInfo.Add(oFileIndex);
                    }
                }

                // Print all output
                Console.WriteLine("Total {0} files indexed\n", lstAllFilesInfo.Count);
                foreach (var itmFileInfo in lstAllFilesInfo)
                {
                    Console.WriteLine("fileName: {0}", itmFileInfo.fileName);
                    Console.WriteLine("fileDate: {0}", itmFileInfo.fileDate.ToString("I"));
                    Console.WriteLine("content: {0}", itmFileInfo.content.Trim());
                    Console.WriteLine();
                }
            }
            catch (Exception ex)
            {
                Console.WriteLine(ex.Message);
            }

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

```

        Console.ReadLine();
    }

    /// <summary>
    /// Get File Content
    /// </summary>
    private static string _GetFileContent(FileInfo fileInfo)
    {
        string fileExtension = System.IO.Path.GetExtension(fileInfo.FullName);

        if (fileExtension == ".pdf")
        {
            return _GetPdfFileContent(fileInfo);
        }
        else if (fileExtension == ".png" || fileExtension == ".jpg")
        {
            return _GetImageContet(fileInfo);
        }

        throw new Exception("File not supported.");
    }

    /// <summary>
    /// Get PDF File Content
    /// </summary>
    private static string _GetPdfFileContent(FileInfo fileInfo)
    {
        //Read all file content...
        using (TextExtractor textExtractor = new TextExtractor("demo", "demo"))
        {
            //Load Document
            textExtractor.LoadDocumentFromFile(fileInfo.FullName);

            return textExtractor.GetText();
        }
    }

    /// <summary>
    /// Get Image Contents
    /// </summary>
    private static string _GetImageContet(FileInfo fileInfo)
    {
        //Read all file content...
        using (TextExtractor extractor = new TextExtractor())
        {
            // Load document
            extractor.LoadDocumentFromFile(fileInfo.FullName);

            //Set option to repair text
            extractor.OCRMode = OCRMode.TextFromImagesAndVectorsAndRepairedFonts;

            // Enable Optical Character Recognition (OCR)
            // in .Auto mode (SDK automatically checks if needs to use OCR or not)
            extractor.OCRMode = OCRMode.Auto;

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

            // Set OCR language
            extractor.OCRLanguage = "eng"; // "eng" for english, "deu" for German,

```

```

        // Find more language files at https://github.com/bytescout/ocrdata

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

        // Read all text
        return extractor.GetText();
    }
}

/// <summary>
/// Check whether file is valid
/// </summary>
private static bool _IsFileSupported(FileInfo fileInfo)
{
    //Get File Extension
    string fileExtension = Path.GetExtension(fileInfo.Name);

    //Check whether file extension is valid
    return (fileExtension == ".pdf" || fileExtension == ".png" || fileExtension == ".tiff" || fileExtension == ".bmp" || fileExtension == ".gif" || fileExtension == ".jpg" || fileExtension == ".jpeg" || fileExtension == ".png" || fileExtension == ".tiff" || fileExtension == ".bmp" || fileExtension == ".gif" || fileExtension == ".jpg" || fileExtension == ".jpeg");
}

}

/// <summary>
/// FileIndexOutput class
/// </summary>
public class FileIndexOutput
{
    public string fileName { get; set; }

    public DateTime fileDate { get; set; }

    public string content { get; set; }
}
}

```

VIDEO

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

ON-PREMISE OFFLINE SDK

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

ON-DEMAND REST WEB API

[Get Your API Key](#)
[Explore Web API Docs](#)
[Explore Web API Samples](#)

[visit www.ByteScout.com](#)

[visit www.PDF.co](#)

www.bytescout.com