

www.bytescout.com

How to parse folder with invoices to CSV in C# using ByteScout Invoice Parser SDK

This code in C# shows how to parse folder with invoices to CSV with this how to tutorial

This sample source code below will demonstrate you how to parse folder with invoices to CSV in C#. ByteScout Invoice Parser SDK can parse folder with invoices to CSV. It can be used from C#. ByteScout Invoice Parser SDK is the SDK for automatic invoice parsing and data extraction. Thousands of companies are supported out of the box! Detects and extracts a company name, invoice number, date, total amount to be paid and other fields. Constantly updated database of supported invoices. Batch processing is supported. Outputs JSON, CSV, XML data and can be integrated with other apps and tools.

This rich sample source code in C# for ByteScout Invoice Parser SDK includes the number of functions and options you should do calling the API to parse folder with invoices to CSV. Follow the instructions from the scratch to work and copy the C# code. Code testing will allow the function to be tested and work properly with your data.

Our website provides trial version of ByteScout Invoice Parser SDK for free. It also includes documentation and source code samples.

FOR MORE INFORMATION AND FREE TRIAL:

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

[Read more about ByteScout Invoice Parser SDK](#)

[Explore API Documentation](#)

[Get Free Training for ByteScout Invoice Parser SDK](#)

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

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

Source Code Files:

ParseFolderToCSV.sln

```
Microsoft Visual Studio Solution File, Format Version 12.00
# Visual Studio 15
VisualStudioVersion = 15.0.28010.2016
MinimumVisualStudioVersion = 10.0.40219.1
Project("{FAE04EC0-301F-11D3-BF4B-00C04F79EFBC}") = "ParseFolderToCSV", "ParseFolderToCSV.csproj", "{FAE04EC0-301F-11D3-BF4B-00C04F79EFBC}"
EndProject
Global
    GlobalSection(SolutionConfigurationPlatforms) = preSolution
        Debug|Any CPU = Debug|Any CPU
        Release|Any CPU = Release|Any CPU
    EndGlobalSection
    GlobalSection(ProjectConfigurationPlatforms) = postSolution
        {D674E35B-D877-45FC-A459-6CEF673CF427}.Debug|Any CPU.ActiveCfg = Debug|Any CPU
        {D674E35B-D877-45FC-A459-6CEF673CF427}.Debug|Any CPU.Build.0 = Debug|Any CPU
        {D674E35B-D877-45FC-A459-6CEF673CF427}.Release|Any CPU.ActiveCfg = Release|Any CPU
        {D674E35B-D877-45FC-A459-6CEF673CF427}.Release|Any CPU.Build.0 = Release|Any CPU
    EndGlobalSection
    GlobalSection(SolutionProperties) = preSolution
        HideSolutionNode = FALSE
    EndGlobalSection
    GlobalSection(ExtensibilityGlobals) = postSolution
        SolutionGuid = {EDCD81CC-A538-41D3-A99A-0F287E9FC614}
    EndGlobalSection
EndGlobal
```

Program.cs

```
using System;
using System.Diagnostics;
using System.Text;
using ByteScout.InvoiceParser;

namespace ParseFolderToCSV
{
    class Program
    {
        static void Main(string[] args)
        {
            string inputFolder = @".\invoices";
            string outputFile = @".\result.csv";

            // Create InvoiceParser instance
            using (InvoiceParser parser = new InvoiceParser("demo", "demo"))
            {
                // Setup CSV output
```

```

        CSVOptions csvOptions = new CSVOptions
        {
            Encoding = Encoding.UTF8,
            SeparatorCharacter = ",",
            QuotationCharacter = "\"",
            UnwrapMultilineValues = true
        };

        // Parse all document in input folder and save results to CSV file
        BatchProcessing.ParseFolderToCSV(parser, inputFolder, outputFile, csvOptions);
    }

    // Open generated CSV file in default associated application (for demo purpose)
    Process.Start(outputFile);
}

/// <summary>
/// Callback method to display progress and interrupt processing if needed.
/// </summary>
/// <param name="fileName">File name.</param>
/// <param name="parsingResult">Parsing result.</param>
/// <param name="progress">Current progress in percents.</param>
/// <param name="innerException">Exception information in case of error.</param>
/// <param name="cancel">Set to <c>true</c> to interrupt the batch processing.</param>
static void ProcessingCallback(string fileName, bool parsingResult, double progress)
{
    Console.WriteLine("{0}quot;{1}% Processed file \\"{2}fileName\\\"");
    Console.WriteLine("{0}quot;  Result: {3}parsingResult");
    if (!parsingResult)
    {
        if (innerException != null)
            Console.WriteLine(innerException.ToString());

        Console.Write("Continue processing? (y/n): ");
        int key = Console.Read();
        if (key == 'n')
        {
            Console.WriteLine("Interrupted.");
            cancel = true;
        }
    }
}
}
}

```

VIDEO

<https://www.youtube.com/watch?v=nVANfuA4Xh0>

ON-PREMISE OFFLINE SDK

[60 Day Free Trial](#) or [Visit ByteScout Invoice Parser SDK Home Page](#)

[Explore ByteScout Invoice Parser SDK Documentation](#)

[Explore Samples](#)

[Sign Up for ByteScout Invoice Parser SDK 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](#)