

PDF invoice parsing with PDF extractor SDK in C# and ByteScout Premium Suite

Learn to code in C# to make PDF invoice parsing with PDF extractor SDK with this simple How-To tutorial

These sample source codes given below will show you how to handle a complex task, for example, PDF invoice parsing with PDF extractor SDK in C#. ByteScout Premium Suite helps with PDF invoice parsing with PDF extractor SDK in C#. ByteScout Premium Suite is the set that includes 12 SDK products from ByteScout including tools and components for PDF, barcodes, spreadsheets, screen video recording.

If you want to quickly learn then these fast application programming interfaces of ByteScout Premium Suite for C# plus the guideline and the C# code below will help you quickly learn PDF invoice parsing with PDF extractor SDK. If you want to know how it works, then this C# sample code should be copied and pasted into your application's code editor. Then just compile and run it. Enjoy writing a code with ready-to-use sample C# codes to add PDF invoice parsing with PDF extractor SDK functions using ByteScout Premium Suite in C#.

Trial version along with the source code samples for C# can be downloaded from our website

FOR MORE INFORMATION AND FREE TRIAL:

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

[Read more about ByteScout Premium Suite](#)

[Explore API Documentation](#)

[Get Free Training for ByteScout Premium Suite](#)

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

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

Source Code Files:

```
using System;
using System.Drawing;
using Bytescout.PDFExtractor;

namespace InvoiceParsing
{
    /// <summary>
    /// This example demonstrates parsing and data extraction from typical invoice
    /// </summary>
    class Program
    {
        static void Main(string[] args)
        {
            // Create TextExtractor instance
            TextExtractor textExtractor = new TextExtractor("demo", "demo");
            textExtractor.WordMatchingMode = WordMatchingMode.ExactMatch;

            // Create XMLExtractor instance
            XMLExtractor xmlExtractor = new XMLExtractor("demo", "demo");

            // Load document
            textExtractor.LoadDocumentFromFile("Invoice.pdf");
            xmlExtractor.LoadDocumentFromFile("Invoice.pdf");

            // Results
            string invoiceNo = string.Empty;
            string invoiceDate = string.Empty;
            string total = string.Empty;
            string tableData = string.Empty;

            // Iterate pages
            for (int i = 0; i < textExtractor.GetPageCount(); i++)
            {
                RectangleF pageRectangle = textExtractor.GetPageRectangle(i);
                RectangleF tableRect = new RectangleF(0, 0, pageRectangle.Width, pageRectangle.Height);

                // Search for "Invoice No."
                if (textExtractor.Find(i, "Invoice No.", false))
                {
                    // Get the found text rectangle
                    RectangleF textRect = textExtractor.FoundText(i, "Invoice No.");
                    // Assume the text at right is the invoice number
                    // Shift the rectangle to the right:
                    textRect.X = textRect.Right;
                    textRect.Width = pageRectangle.Right - textRect.Right;
                    // Set the extraction region and extract the text
                    textExtractor.SetExtractionArea(textRect);
                    invoiceNo = textExtractor.GetTextFromPage(i).Trim();
                }

                // Search for "Invoice Date" and extract text at right
                if (textExtractor.Find(i, "Invoice Date", false))
                {
                    RectangleF textRect = textExtractor.FoundText(i, "Invoice Date");
                    textRect.X = textRect.Right;
                }
            }
        }
    }
}
```

```

        textRect.Width = pageRectangle.Right - textRect.Left;
        textExtractor.SetExtractionArea(textRect);
        invoiceDate = textExtractor.GetTextFromPage(i);
    }

    // Search for "Quantity" keyword to detect the top of table
    if (textExtractor.Find(i, "Quantity", false))
    {
        // Keep the top table coordinate
        tableRect.Y = textExtractor.FoundText.Bounds.Top;
    }

    // Search for "TOTAL" (it will be also the bottom of table)
    if (textExtractor.Find(i, "TOTAL", true /* case sensitive */))
    {
        RectangleF textRect = textExtractor.FoundText.Bounds;
        textRect.X = textRect.Right;
        textRect.Width = pageRectangle.Right - textRect.Left;
        textExtractor.SetExtractionArea(textRect);
        total = textExtractor.GetTextFromPage(i).Trim();
    }

    // Calculate the table height
    tableRect.Height = textRect.Top - tableRect.Top;
}

// Extract tabular data using XMLExtractor
if (tableRect.Height > 0)
{
    xmlExtractor.SetExtractionArea(tableRect);
    tableData = xmlExtractor.GetXMLFromPage(i);
}
}

// Display extracted data
Console.WriteLine("Invoice No.: " + invoiceNo);
Console.WriteLine("Invoice Date: " + invoiceDate);
Console.WriteLine("TOTAL: " + total);
Console.WriteLine("Table Data: ");
Console.WriteLine(tableData);

// Cleanup
textExtractor.Dispose();
xmlExtractor.Dispose();

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

```

VIDEO

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

ON-PREMISE OFFLINE SDK

[60 Day Free Trial](#) or [Visit ByteScout Premium Suite Home Page](#)

[Explore ByteScout Premium Suite Documentation](#)

[Explore Samples](#)

[Sign Up for ByteScout Premium 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