How to find table in pdf and extract as csv with pdf extractor sdk in VB.NET and ByteScout PDF Suite

Step-by-step tutorial on how to find table in pdf and extract as csv with pdf extractor sdk in VB.NET

These sample source codes on this page below are displaying how to find table in pdf and extract as csv with pdf extractor sdk in VB.NET. ByteScout PDF Suite can find table in pdf and extract as csv with pdf extractor sdk. It can be applied from VB.NET. 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 SDK samples given below describe how to quickly make your application do find table in pdf and extract as csv with pdf extractor sdk in VB.NET with the help of ByteScout PDF Suite. Simply copy and paste in your VB.NET project or application you and then run your app! Further improvement of the code will make it more robust.

You can download free trial version of ByteScout PDF Suite from our website with this and other source code samples for VB.NET.

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

Source Code Files:

Program.vb

```vbnet
Imports Bytescout.PDFExtractor

Class Program
        Friend Shared Sub Main(args As String())

        ' Create Bytescout.PDFExtractor.CSVExtractor instance
        Dim csvExtractor As New CSVExtractor()
        csvExtractor.RegistrationName = "demo"
        csvExtractor.RegistrationKey = "demo"

        ' Create Bytescout.PDFExtractor.TableDetector instance
        Dim tableDetector As New TableDetector()
        tableDetector.RegistrationName = "demo"
        tableDetector.RegistrationKey = "demo"

        ' We should define what kind of tables we should detect.
        ' So we set min required number of columns to 3 ...
        tableDetector.DetectionMinNumberOfColumns = 3
        ' ... and we set min required number of rows to 3
        tableDetector.DetectionMinNumberOfRows = 3

        ' Set table detection mode to "bordered tables" - best for tables with closed s
        tableDetector.ColumnDetectionMode = ColumnDetectionMode.BorderedTables

            ' Load sample PDF document
        csvExtractor.LoadDocumentFromFile(".\sample3.pdf")
        tableDetector.LoadDocumentFromFile(".\sample3.pdf")

            ' Get page count
        Dim pageCount As Integer = tableDetector.GetPageCount()

        ' Iterate through pages
            For i As Integer = 0 To pageCount - 1
            Dim t As Integer = 1
            ' Find first table and continue if found
            If (tableDetector.FindTable(i)) Then
                Do
                    ' Set extraction area for CSV extractor to rectangle received from
                    csvExtractor.SetExtractionArea(tableDetector.FoundTableLocation)
                    ' Export the table to CSV file
                    csvExtractor.SavePageCSVToFile(i, "page-" + i.ToString() + "-table-
                    t = t + 1
                Loop While tableDetector.FindNextTable()
            End If
        Next

        ' Cleanup
            csvExtractor.Dispose()
            tableDetector.Dispose()

        ' Open first output file in default associated application (for demo purposes)
        System.Diagnostics.Process.Start("page-0-table-1.csv")
```

```
        End Sub
End Class
```

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

visit www.PDF.co