How to find table in PDF and extract as CSV in VB.NET and ByteScout PDF Extractor SDK

The tutorial shows how to find table in PDF and extract as CSV in VB.NET

The code below will help you to implement an VB.NET app to find table in PDF and extract as CSV. What is ByteScout PDF Extractor SDK? It is the SDK that helps developers to extract data from unstructured documents, pdf, images, scanned and electronic forms. Includes AI functions like automatic table detection, automatic table extraction and restructuring, text recognition and text restoration from pdf and scanned documents. Includes PDF to CSV, PDF to XML, PDF to JSON, PDF to searchable PDF functions as well as methods for low level data extraction. It can help you to find table in PDF and extract as CSV in your VB.NET application.

This code snippet below for ByteScout PDF Extractor SDK works best when you need to quickly find table in PDF and extract as CSV in your VB.NET application. This VB.NET sample code is all you need for your app. Just copy and paste the code, add references (if needs to) and you are all set! Detailed tutorials and documentation are available along with installed ByteScout PDF Extractor SDK if you'd like to dive deeper into the topic and the details of the API.

Free trial version of ByteScout PDF Extractor SDK is available on our website. Documentation and source code samples are included.

VB.NET - Program.vb

```
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
solid borders.
        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 the table detector
                    csvExtractor.SetExtractionArea(tableDetector.FoundTableLocation)
                    ' Export the table to CSV file
                    csvExtractor.SavePageCSVToFile(i, "page-" + i.ToString() + "-
table-" + t.ToString() + ".csv")
                    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
```

FOR MORE INFORMATION AND FREE TRIAL:

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

[Read more about ByteScout PDF Extractor SDK]

[Explore documentation]

[Visit www.ByteScout.com]

or