www.bytescout.com

How to print PDF in VB.NET using ByteScout PDF Renderer SDK

How to print PDF in VB.NET

The sample shows steps and algorithm of how to print PDF and how to make it work in your VB.NET application. Want to print PDF in your VB.NET app? ByteScout PDF Renderer SDK is designed for it. ByteScout PDF Renderer SDK is the SDK for rendering of PDF into high-quality thumbnails and images. Includes various functions like batch processing, PNG, TIFF output. Can be used from web and desktop applications.

This code snippet below for ByteScout PDF Renderer SDK works best when you need to quickly print PDF in your VB.NET application. Follow the instructions from the scratch to work and copy the VB.NET code. Test VB.NET sample code examples whether they respond your needs and requirements for the project.

ByteScout PDF Renderer SDK free trial version is available on our website. VB.NET and other programming languages are supported.

FOR MORE INFORMATION AND FREE TRIAL:

Download Free Trial SDK (on-premise version)

Read more about ByteScout PDF Renderer SDK

Explore API Documentation

Get Free Training for ByteScout PDF Renderer SDK

Get Free API key for Web API

visit www.ByteScout.com

Source Code Files:

Form1.Designer.vb

```
Partial Class Form1
        ''' <summary>
        ''' Required designer variable.
        ''' </summary>
        Private components As System.ComponentModel.IContainer = Nothing

        ''' <summary>
        ''' Clean up any resources being used.
        ''' </summary>
        ''' <param name="disposing">true if managed resources should be disposed; other
        Protected Overrides Sub Dispose(disposing As Boolean)
                If disposing AndAlso (components IsNot Nothing) Then
                        components.Dispose()
                End If
                MyBase.Dispose(disposing)
        End Sub

        #Region "Windows Form Designer generated code"

        ''' <summary>
        ''' Required method for Designer support - do not modify
        ''' the contents of this method with the code editor.
        ''' </summary>
        Private Sub InitializeComponent()
                Dim resources As New System.ComponentModel.ComponentResourceManager(Get
                Me.printDocument1 = New System.Drawing.Printing.PrintDocument()
                Me.printPreviewDialog1 = New System.Windows.Forms.PrintPreviewDialog()
                Me.pageSetupDialog1 = New System.Windows.Forms.PageSetupDialog()
                Me.printDialog1 = New System.Windows.Forms.PrintDialog()
                Me.buttonPageSetup = New System.Windows.Forms.Button()
                Me.buttonPrintPreview = New System.Windows.Forms.Button()
                Me.buttonPrint = New System.Windows.Forms.Button()
                Me.SuspendLayout()
                '
                ' printDocument1
                '
                AddHandler Me.printDocument1.PrintPage, New System.Drawing.Printing.Pr
                '
                ' printPreviewDialog1
                '
                Me.printPreviewDialog1.AutoScrollMargin = New System.Drawing.Size(0, 0
                Me.printPreviewDialog1.AutoScrollMinSize = New System.Drawing.Size(0, 
                Me.printPreviewDialog1.ClientSize = New System.Drawing.Size(400, 300)
                Me.printPreviewDialog1.Document = Me.printDocument1
                Me.printPreviewDialog1.Enabled = True
                Me.printPreviewDialog1.Icon = DirectCast(resources.GetObject("printPrev
                Me.printPreviewDialog1.Name = "printPreviewDialog1"
                Me.printPreviewDialog1.UseAntiAlias = True
                Me.printPreviewDialog1.Visible = False
                '
                ' pageSetupDialog1
                '
                Me.pageSetupDialog1.Document = Me.printDocument1
                '
                ' printDialog1
                '
                Me.printDialog1.Document = Me.printDocument1
                Me.printDialog1.UseEXDialog = True
```

```
            '
            ' buttonPageSetup
            '
            Me.buttonPageSetup.Anchor = CType(((System.Windows.Forms.AnchorStyles.T
            Me.buttonPageSetup.Location = New System.Drawing.Point(12, 12)
            Me.buttonPageSetup.Name = "buttonPageSetup"
            Me.buttonPageSetup.Size = New System.Drawing.Size(253, 41)
            Me.buttonPageSetup.TabIndex = 0
            Me.buttonPageSetup.Text = "Page Setup"
            Me.buttonPageSetup.UseVisualStyleBackColor = True
            AddHandler Me.buttonPageSetup.Click, New System.EventHandler(AddressOf
            '
            ' buttonPrintPreview
            '
            Me.buttonPrintPreview.Anchor = CType(((System.Windows.Forms.AnchorStyle
            Me.buttonPrintPreview.Location = New System.Drawing.Point(12, 59)
            Me.buttonPrintPreview.Name = "buttonPrintPreview"
            Me.buttonPrintPreview.Size = New System.Drawing.Size(253, 41)
            Me.buttonPrintPreview.TabIndex = 1
            Me.buttonPrintPreview.Text = "Print Preview"
            Me.buttonPrintPreview.UseVisualStyleBackColor = True
            AddHandler Me.buttonPrintPreview.Click, New System.EventHandler(Address
            '
            ' buttonPrint
            '
            Me.buttonPrint.Anchor = CType(((System.Windows.Forms.AnchorStyles.Top (
            Me.buttonPrint.Location = New System.Drawing.Point(12, 106)
            Me.buttonPrint.Name = "buttonPrint"
            Me.buttonPrint.Size = New System.Drawing.Size(253, 41)
            Me.buttonPrint.TabIndex = 2
            Me.buttonPrint.Text = "Print"
            Me.buttonPrint.UseVisualStyleBackColor = True
            AddHandler Me.buttonPrint.Click, New System.EventHandler(AddressOf Me.B
            '
            ' Form1
            '
            Me.AutoScaleDimensions = New System.Drawing.SizeF(6F, 13F)
            Me.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font
            Me.ClientSize = New System.Drawing.Size(277, 161)
            Me.Controls.Add(Me.buttonPrint)
            Me.Controls.Add(Me.buttonPrintPreview)
            Me.Controls.Add(Me.buttonPageSetup)
            Me.Name = "Form1"
            Me.Text = "Print PDF"
            AddHandler Me.Load, New System.EventHandler(AddressOf Me.Form1_Load)
            Me.ResumeLayout(False)

        End Sub

        #End Region

        Private printDocument1 As System.Drawing.Printing.PrintDocument
        Private printPreviewDialog1 As System.Windows.Forms.PrintPreviewDialog
        Private pageSetupDialog1 As System.Windows.Forms.PageSetupDialog
        Private printDialog1 As System.Windows.Forms.PrintDialog
        Private buttonPageSetup As System.Windows.Forms.Button
        Private buttonPrintPreview As System.Windows.Forms.Button
        Private buttonPrint As System.Windows.Forms.Button
End Class
```

Form1.vb

```
Imports System.Drawing
Imports System.Drawing.Imaging
Imports System.Windows.Forms

Imports Bytescout.PDFRenderer


Public Partial Class Form1
        Inherits Form
        Private _document As String = "multipage.pdf"
        Private _rasterRenderer As RasterRenderer = Nothing
        Private _page As Integer = 0


        Public Sub New()
                InitializeComponent()

                ' Create an instance of Bytescout.PDFRenderer.RasterRenderer object and
                _rasterRenderer = New RasterRenderer()
                _rasterRenderer.RegistrationName = "demo"
                _rasterRenderer.RegistrationKey = "demo"
        End Sub

        Private Sub Form1_Load(sender As Object, e As EventArgs)
                Cursor = Cursors.WaitCursor

                Try
            ' Load PDF document
                        _rasterRenderer.LoadDocumentFromFile(_document)
                Catch exception As Exception
            MessageBox.Show("Could not open PDF document." & vbLf & vbLf & exception.Me
                Finally
                        Cursor = Cursors.[Default]
                End Try
        End Sub

    Private Sub buttonPageSetup_Click(ByVal sender As Object, ByVal e As EventArgs)

        ' Set landscape orientation if needed
        Dim pageRectangle As RectangleF = _rasterRenderer.GetPageRectangle(0)
        If (pageRectangle.Width > pageRectangle.Height) Then
            pageSetupDialog1.PageSettings.Landscape = True
        End If

        pageSetupDialog1.ShowDialog()

    End Sub
```

```
        Private Sub buttonPrintPreview_Click(sender As Object, e As EventArgs)
                _page = 0
                printPreviewDialog1.Width = 800
                printPreviewDialog1.Height = 600
                printPreviewDialog1.ShowDialog()
        End Sub

        Private Sub buttonPrint_Click(sender As Object, e As EventArgs)
                If printDialog1.ShowDialog() = DialogResult.OK Then
                        printDocument1.Print()
                End If
        End Sub

    Private Sub printDocument1_PrintPage(sender As Object, e As System.Drawing.Printing

        Cursor = Cursors.WaitCursor

        Try
            ' For the best quality set the rendering resolution equal to the printer re
            Dim renderingResolution As Single = e.PageSettings.PrinterResolution.X

            ' Render page to image
            Using image As Image = _rasterRenderer.GetImage(_page, renderingResolution)

                ' Fit image into the print rectangle keeping the aspect ratio

                Dim printRect As Rectangle = e.MarginBounds

                Dim ratio As Single = printRect.Width / CSng(image.Width)
                Dim width As Integer = printRect.Width
                Dim height As Integer = CInt(Math.Round(image.Height * ratio))

                If height > printRect.Height Then
                    ratio = printRect.Height / CSng(image.Height)
                    width = CInt(Math.Round(image.Width * ratio))
                    height = printRect.Height
                End If

                ' Draw image on device
                e.Graphics.DrawImage(image, New Rectangle(printRect.X, printRect.Y, wi

            End Using

            If _page < _rasterRenderer.GetPageCount() - 1 Then
                _page += 1
                e.HasMorePages = True
            End If
        Catch exception As Exception
            MessageBox.Show(exception.Message)
        Finally
            Cursor = Cursors.Default
        End Try

    End Sub
End Class
```

Program.vb

```
Imports System.Collections.Generic
Imports System.Windows.Forms

NotInheritable Class Program
    Private Sub New()
    End Sub
    ''' <summary>
    ''' The main entry point for the application.
    ''' </summary>
    <STAThread> _
    Friend Shared Sub Main()
        Application.EnableVisualStyles()
        Application.SetCompatibleTextRenderingDefault(False)
        Application.Run(New Form1())
    End Sub
End Class
```

VIDEO

https://www.youtube.com/watch?v=eenhl7106lA

ON-PREMISE OFFLINE SDK

60 Day Free Trial or Visit ByteScout PDF Renderer SDK Home Page
Explore ByteScout PDF Renderer SDK Documentation
Explore Samples
Sign Up for ByteScout PDF Renderer 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