

How to create invoice spanning multiple page with pdf sdk in VB.NET with ByteScout PDF Suite

Continuous learning is a crucial part of computer science and this tutorial shows how to create invoice spanning multiple page with pdf sdk in VB.NET

We made thousands of pre-made source code pieces for easy implementation in your own programming projects. ByteScout PDF Suite is the set that includes 6 SDK products 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. It can be applied to create invoice spanning multiple page with pdf sdk using VB.NET.

These VB.NET code samples for VB.NET guide developers to speed up coding of the application when using ByteScout PDF Suite. IF you want to implement the functionality, just copy and paste this code for VB.NET below into your code editor with your app, compile and run your application. Want to see how it works with your data then code testing will allow the function to be tested and work properly.

You can download free trial version of ByteScout PDF Suite from our website to see and try many others 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](http://www.ByteScout.com)

Source Code Files:

Program.vb

```
Imports Bytescout.PDF

''' <summary>
''' This example demonstrates how to create invoice which can span multiple pages.
''' </summary>
Class Program

#Region "Global Declarations"

    Const HeightConsumedByTableRow As Int32 = 20

    Shared ContentHeightStart As Int32 = 100
    Shared ContentHeightEnd As Int32 = 700

    Shared HeightUtilized As Int32 = 100

    Shared _RowCounter As Int32 = 1

    Shared pdfDocument As Document
    Shared page As Page

#End Region

    Shared Sub Main()

#Region "Declarations and Implementations"

        Dim SolidPen As New SolidPen()

        Dim blackBrush As New SolidBrush(New ColorRGB(0, 0, 0))

        Dim headerFont As New Font(StandardFonts.TimesBold, 24)
        Dim headerFont2 As New Font(StandardFonts.TimesBold, 16)

        Dim contentFont_12 As New Font(StandardFonts.TimesBold, 12)
        Dim contentFont_10 As New Font(StandardFonts.Times, 10)

        ' Create New document
        pdfDocument = New Document()
        pdfDocument.RegistrationName = "demo"
        pdfDocument.RegistrationKey = "demo"

#End Region

        ' Add New page
        AddNewPage()

        ' 1. Add Content
        page.Canvas.DrawString("COMPANY NAME", contentFont_12, blackBrush, 50, HeightUtilized)
        HeightUtilized += 20

        page.Canvas.DrawString("Address", contentFont_10, blackBrush, 50, HeightUtilized)
```

```

page.Canvas.DrawString("DATE", contentFont_10, blackBrush, 400, HeightUtilized)
HeightUtilized += 20

page.Canvas.DrawString("Phone, fax", contentFont_10, blackBrush, 50, HeightUtilized)
page.Canvas.DrawString("INVOICE #", contentFont_10, blackBrush, 400, HeightUtilized)
HeightUtilized += 20

page.Canvas.DrawString("E-mail", contentFont_10, blackBrush, 50, HeightUtilized)
page.Canvas.DrawString("FOR", contentFont_10, blackBrush, 400, HeightUtilized)

' Add some vertical space
HeightUtilized += 50

' 2. Add Table which can span multipage
Console.WriteLine("Enter the number of rows to add to the table:")
Dim numberOfRows As Int32 = Convert.ToInt32(Console.ReadLine())
DrawTable(NumberOfRows:=numberOfRows)

' 3. Add Signature
' Check New page needed for adding signature
If (CheckNewPageNeeded(70)) Then
    AddNewPage()
End If

page.Canvas.DrawString("Signature", contentFont_12, blackBrush, 400, (HeightUtilized + 70))
page.Canvas.DrawLine(SolidPen, 450, (HeightUtilized + 70), 570, (HeightUtilized + 70))

'4. Add Header And Footer
For pageIndex = 0 To pdfDocument.Pages.Count - 1

    Dim curPage As Page = pdfDocument.Pages(pageIndex)

    ' Add Logo
    Dim imageLogo As Image = New Image("logo.png")
    curPage.Canvas.DrawImage(imageLogo, 20, 45, imageLogo.Width / 2, imageLogo.Height)

    ' Add requisites
    curPage.Canvas.DrawString("INVOICE", headerFont, blackBrush, 450, 55)

    ' Add Page Number
    curPage.Canvas.DrawLine(SolidPen, 20, 750, 590, 750)
    curPage.Canvas.DrawString("Page: {pageIndex + 1} of {pdfDocument.Pages.Count}")

Next

' Save document to file
pdfDocument.Save("result.pdf")

' Cleanup
pdfDocument.Dispose()

' Open document in default PDF viewer app
Process.Start("result.pdf")

End Sub

''' <summary>
''' Checks whether new page is needed

```

```

''' </summary>
''' <param name="expectedHeight"></param>
''' <returns></returns>
Shared Function CheckNewPageNeeded(expectedHeight As Int32) As Boolean
    Return (HeightUtilized + expectedHeight) > ContentHeightEnd
End Function

''' <summary>
''' Adds new page, and also set height utilized
''' </summary>
Shared Sub AddNewPage()
    ' Add page
    page = New Page(PaperFormat.Letter, PaperOrientation.Portrait)
    pdfDocument.Pages.Add(page)

    ' Reset Height Utilized
    HeightUtilized = ContentHeightStart
End Sub

''' <summary>
''' Draw table
''' </summary>
''' <param name="NumberOfRows"></param>
Shared Sub DrawTable(NumberOfRows As Int32)

    While (NumberOfRows > 0)

        ' Get number of rows can be added in current page
        Dim NumberOfRowsCanBeAddedInCurrentPage As Int32 = GetNumberOfRowsCanBeAddedInCurrentPage

        If (NumberOfRowsCanBeAddedInCurrentPage > 0) Then
            Dim Table = GetTable(Math.Min(NumberOfRows, NumberOfRowsCanBeAddedInCurrentPage),
                page.Canvas.DrawTable(Table, 50, HeightUtilized)

            ' Update Height Utilized
            HeightUtilized += (Math.Min(NumberOfRows, NumberOfRowsCanBeAddedInCurrentPage))

            NumberOfRows -= Math.Min(NumberOfRows, NumberOfRowsCanBeAddedInCurrentPage)
        Else
            ' Add New page
            AddNewPage()
        End If

    End While

End Sub

''' <summary>
''' Get Table Header with Content
''' </summary>
''' <param name="NumberOfRows"></param>
''' <returns></returns>
Shared Function GetTable(NumberOfRows As Int32) As Table

    Dim fontTableHeader As Font = New Font(StandardFonts.TimesBold, 11)
    Dim fontTableRow As Font = New Font(StandardFonts.Times, 10)

    Dim Table As New Table()

```

```

' 1. Add Table Header
Table.Columns.Add(New TableColumn("Description", "Description"))
Table.Columns(0).Width = 220
Table.Columns(0).Font = fontTableHeader
Table.Columns(0).TextFormat.HorizontalAlign = HorizontalAlign.Left

Table.Columns.Add(New TableColumn("Quantity", "Quantity"))
Table.Columns(1).Width = 80
Table.Columns(1).Font = fontTableHeader
Table.Columns(1).TextFormat.HorizontalAlign = HorizontalAlign.Right

Table.Columns.Add(New TableColumn("Price", "Price"))
Table.Columns(2).Width = 100
Table.Columns(2).Font = fontTableHeader
Table.Columns(2).TextFormat.HorizontalAlign = HorizontalAlign.Right

Table.Columns.Add(New TableColumn("Amount", "Amount"))
Table.Columns(3).Width = 120
Table.Columns(3).Font = fontTableHeader
Table.Columns(3).TextFormat.HorizontalAlign = HorizontalAlign.Right

' 2. Add Table Content
For index = 0 To NumberOfRows - 1

    ' Create New row And set its background color
    Dim row As TableRow = Table.NewRow()

    ' Set row header text
    row("Description").Text = {code}quot;Description For item - {_RowCounter}"
    row("Description").Font = fontTableRow
    row("Description").TextFormat.HorizontalAlign = HorizontalAlign.Left

    row("Quantity").Text = "###"
    row("Quantity").Font = fontTableRow
    row("Quantity").TextFormat.HorizontalAlign = HorizontalAlign.Right

    row("Price").Text = "$.[code]

```

{codeFileName}

{code}

```

[/code]quot;
    row("Price").Font = fontTableRow
    row("Price").TextFormat.HorizontalAlign = HorizontalAlign.Right

    row("Amount").Text = "$.[code]

```

```
{codeFileName}
```

```
{code}
```

```
[/code]quot;  
    row("Amount").Font = fontTableRow  
    row("Amount").TextFormat.HorizontalAlign = HorizontalAlign.Right  
  
    ' Add the row to the table  
    Table.Rows.Add(row)  
  
    ' Increase _RowCounter  
    _RowCounter += 1  
Next  
  
Return Table  
  
End Function  
  
''' <summary>  
''' Get number of rows can be added in current page  
''' </summary>  
''' <returns></returns>  
Shared Function GetNumberOfRowsCanBeAddedInCurrentPage() As Integer  
    Return (ContentHeightEnd - HeightUtilized) / HeightConsumedByTableRow  
End Function  
  
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](http://www.ByteScout.com)

[visit www.PDF.co](http://www.PDF.co)

www.bytescout.com