

How to convert HTML email with attachments to PDF in VB.NET using ByteScout PDF SDK

How to convert HTML email with attachments to PDF in VB.NET

Convert HTML email with attachments to PDF is easy to implement in VB.NET if you use these source codes below. ByteScout PDF SDK can convert HTML email with attachments to PDF. It can be used from VB.NET. ByteScout PDF SDK is the SDK for pdf documents generation, modification and updates. Can also generate and fill PDF forms. Provides support for text (fonts, style, size, font family), layers, pdf form fields, vector and raster drawings.

The SDK samples like this one below explain how to quickly make your application do convert HTML email with attachments to PDF in VB.NET with the help of ByteScout PDF SDK. Follow the instructions from the scratch to work and copy the VB.NET code. Implementing VB.NET application typically includes multiple stages of the software development so even if the functionality works please test it with your data and the production environment.

Free trial version of ByteScout PDF SDK is available for download from our website. Get it to try 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 SDK](#)

[Explore API Documentation](#)

[Get Free Training for ByteScout PDF SDK](#)

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

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

Source Code Files:

```
Microsoft Visual Studio Solution File, Format Version 12.00
# Visual Studio 15
VisualStudioVersion = 15.0.27703.2026
MinimumVisualStudioVersion = 10.0.40219.1
Project("{F184B08F-C81C-45F6-A57F-5ABD9991F28F}") = "EmailToPDF_HTMLEmailWithAttachments", "EmailToPDF_HTMLEmailWithAttachments.csproj", "{...}"
EndProject
Global
    GlobalSection(SolutionConfigurationPlatforms) = preSolution
        Debug|Any CPU = Debug|Any CPU
        Release|Any CPU = Release|Any CPU
    EndGlobalSection
    GlobalSection(ProjectConfigurationPlatforms) = postSolution
        {E0A3BA18-AA04-4C1E-AF80-F7232FD8D83E}.Debug|Any CPU.ActiveCfg = Debug|Any CPU
        {E0A3BA18-AA04-4C1E-AF80-F7232FD8D83E}.Debug|Any CPU.Build.0 = Debug|Any CPU
        {E0A3BA18-AA04-4C1E-AF80-F7232FD8D83E}.Release|Any CPU.ActiveCfg = Release|Any CPU
        {E0A3BA18-AA04-4C1E-AF80-F7232FD8D83E}.Release|Any CPU.Build.0 = Release|Any CPU
    EndGlobalSection
    GlobalSection(SolutionProperties) = preSolution
        HideSolutionNode = FALSE
    EndGlobalSection
    GlobalSection(ExtensibilityGlobals) = postSolution
        SolutionGuid = {05DEC94F-8A54-473F-8949-E27D15706BFC}
    EndGlobalSection
EndGlobal
```

Program.vb

```
Imports System.Drawing.Printing
Imports System.IO
Imports System.Text
Imports Bytescout.PDF
Imports Bytescout.PDF.Converters

Imports Font = Bytescout.PDF.Font
Imports SolidBrush = Bytescout.PDF.SolidBrush

Module Program

    Sub Main()

        Try
            ' Parse MessageContents using MsgReader Library
            ' MsgReader library can be obtained from: https://github.com/Sicos1977/MSGReader
            Using msg = New MsgReader.Outlook.Storage.Message("HtmlSampleEmailWithAttachments.html")
```

```

' Get Sender information
Dim from = msg.GetEmailSender(False, False)

' Message sent datetime
Dim sentOn = msg.SentOn

' Recipient To information
Dim recipientsTo = msg.GetEmailRecipients(MsgReader.Outlook.RecipientTy

' Recipient CC information
Dim recipientsCc = msg.GetEmailRecipients(MsgReader.Outlook.RecipientTy

' Recipient BCC information
Dim recipientBcc = msg.GetEmailRecipients(MsgReader.Outlook.RecipientTy

' Message subject
Dim subject = msg.Subject

' Get Message Body
Dim msgBody = msg.BodyHtml

' Prepare PDF documnt
Using outputDocument As Document = New Document()

    ' Add registration keys
    outputDocument.RegistrationName = "demo"
    outputDocument.RegistrationKey = "demo"

    ' Add page
    Dim page As Page = New Page(PaperFormat.A4)
    outputDocument.Pages.Add(page)

    ' Default font and brush
    Dim font As Font = New Font(StandardFonts.Times, 12)
    Dim brush As Brush = New SolidBrush()

    ' Add Email contents
    Dim topMargin As Integer = 0
    topMargin += 20
    page.Canvas.DrawString({code}"File Name: {msg.FileName}", font

    topMargin += 20
    page.Canvas.DrawString({code}"From: {from}", font, brush, 20, f

    topMargin += 20
    page.Canvas.DrawString({code}"Sent On: {(If(sentOn.HasValue, se

    topMargin += 20
    page.Canvas.DrawString({code}"To: {recipientsTo}", font, brush

    If Not String.IsNullOrEmpty(recipientsCc) Then
        topMargin += 20
        page.Canvas.DrawString({code}"CC: {recipientsCc}", font, b
    End If

    If Not String.IsNullOrEmpty(recipientBcc) Then
        topMargin += 20
        page.Canvas.DrawString({code}"BCC: {recipientBcc}", font, b
    End If

```

```

topMargin += 20
page.Canvas.DrawString(quot;Subject: {subject}", font, brush

topMargin += 20
page.Canvas.DrawString("Message body and attachments in next page.

' Convert HTML body to PDF in order to retain all formatting.
Using converter As HtmlToPdfConverter = New HtmlToPdfConverter()

    converter.PageSize = PaperKind.A4
    converter.Orientation = Bytescout.PDF.Converters.PaperOrientat

' Convert input HTML to stream
Dim byteArrayBody As Byte() = Encoding.UTF8.GetBytes(msgBody)
Dim inputStream As MemoryStream = New MemoryStream(byteArrayBo

' Create output stream to store generated PDF file
Using outputStream = New MemoryStream()
    ' Convert HTML to PDF
    converter.ConvertHtmlToPdf(inputStream, outputStream)

' Create new document from generated output stream
Dim docContent As Document = New Document(outputStream)

' Append all pages to main PDF
For Each item As Page In docContent.Pages
    outputDocument.Pages.Add(item)
Next

' Get attachments from message (if any, and append to docum
If msg.Attachments.Count > 0 Then

    For Each itmAttachment As MsgReader.Outlook.Storage.Att

        ' Get Memory Stream
        Dim attachmentMemoryStream As MemoryStream = New Me

        ' Append Attachment
        Dim docAttachment As Document = New Document(attach

        ' Append all pages to main PDF
        For Each item As Page In docAttachment.Pages
            outputDocument.Pages.Add(item)
        Next

    Next

End If

' Save output file
outputDocument.Save("result.pdf")

' Open output file
Process.Start("result.pdf")

    End Using
End Using
End Using
End Using

```

```
Catch ex As Exception
    Console.WriteLine(ex.Message)
    Console.WriteLine("Press enter key to exit...")
    Console.ReadLine()
End Try

End Sub

End Module
```

packages.config

```
<?xml version="1.0" encoding="utf-8"?>
<packages>
  <package id="MsgReader" version="3.4.1" targetFramework="net45" />
  <package id="OpenMcdf" version="2.2.1.3" targetFramework="net45" />
</packages>
```

VIDEO

<https://www.youtube.com/watch?v=gdsQ0EAqwGQ>

ON-PREMISE OFFLINE SDK

[60 Day Free Trial](#) or [Visit ByteScout PDF SDK Home Page](#)
[Explore ByteScout PDF SDK Documentation](#)
[Explore Samples](#)
[Sign Up for ByteScout PDF SDK 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