

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

This code in VB.NET shows how to convert text email with attachments to PDF with this how to tutorial

Learn how to convert text email with attachments to PDF in VB.NET with this source code sample. ByteScout PDF SDK is the pdf library that can create, update and modify PDF files. Supports text with fonts and style selections, layers, form fields, drawing lines and objects, automatic tables, images. Can be used to create and fill pdf forms and you can use it to convert text email with attachments to PDF with VB.NET.

The SDK samples like this one below explain how to quickly make your application do convert text email with attachments to PDF in VB.NET with the help of ByteScout PDF SDK. In order to implement the functionality, you should copy and paste this code for VB.NET below into your code editor with your app, compile and run your application. Test VB.NET sample code examples whether they respond your needs and requirements for the project.

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

VB.NET - Program.vb

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

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("TxtSampleEmailWithAttachment.msg")

                ' 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.RecipientType.[To], False, False)

        ' Recipient CC information
        Dim recipientsCc =
msg.GetEmailRecipients(MsgReader.Outlook.RecipientType.Cc, False, False)

        ' Recipient BCC information
        Dim recipientBcc =
msg.GetEmailRecipients(MsgReader.Outlook.RecipientType.Bcc, False, False)

#Region "Generate and save HTML"

    Dim oHtmlGenerator As New HtmlGenerator()

    oHtmlGenerator.Title = $"Subject: {msg.Subject}"
    oHtmlGenerator.AddParagraphBodyItem($"File Name: {msg.FileName}")
    oHtmlGenerator.AddParagraphBodyItem($"From: {from}")
    oHtmlGenerator.AddParagraphBodyItem($"Sent On: {(If(sentOn.HasValue,
sentOn.Value.ToString("MM/dd/yyyy HH:mm"), "")})}")
    oHtmlGenerator.AddParagraphBodyItem($"To: {recipientsTo}")
    oHtmlGenerator.AddParagraphBodyItem($"Subject: {msg.Subject}")

    If Not String.IsNullOrEmpty(recipientsCc) Then
        oHtmlGenerator.AddParagraphBodyItem($"CC: {recipientsCc}")
    End If

    If Not String.IsNullOrEmpty(recipientBcc) Then
        oHtmlGenerator.AddParagraphBodyItem($"BCC: {recipientBcc}")
    End If

    oHtmlGenerator.AddRawBodyItem("
")

    Dim msgBodySplitted = msg.BodyText.Split(vbLf.ToCharArray())

    For Each itmBody In msgBodySplitted
        oHtmlGenerator.AddParagraphBodyItem(itmBody)
    Next

    ' Generate Html
    oHtmlGenerator.SaveHtml("result.html")

#End Region

    Using Converter As HtmlToPdfConverter = New HtmlToPdfConverter()

        Converter.PageSize = PaperKind.A4
        Converter.Orientation =
Bytescout.PDF.Converters.PaperOrientation.Portrait

        Converter.ConvertHtmlToPdf("result.html", "result.pdf")

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

            Using outputDocument As New Document("result.pdf")

                For Each itmAttachment As

```

```

MsgReader.Outlook.Storage.Attachment In msg.Attachments

        ' Get Memory Stream
        Dim attachmentMemoryStream As MemoryStream = New
MemoryStream(itmAttachment.Data)

        ' Append Attachment
        Dim docAttachment As Document = New
Document(attachmentMemoryStream)

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

        Next

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

        End Using

    End If

    ' Open output file
    Dim processStartInfo As ProcessStartInfo = New
ProcessStartInfo("result.pdf")
    processStartInfo.UseShellExecute = True

    Process.Start(processStartInfo)

    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

'''
'''
''' Html Generator class
'''

Class HtmlGenerator

#Region "Constructors and variable declarations"

    Public Sub New()
        Me.StrBodyItems = New List(Of String)()
    End Sub

    Public Property Title As String
    Private Property StrBodyItems As List(Of String)

```

```

#End Region

#Region "Methods"

'''

''' Add Body Item
'''

Public Sub AddRawBodyItem(ByVal strBodyItem As String)
    StrBodyItems.Add(strBodyItem)
End Sub

'''

''' Add Paragraph body item
'''

Public Sub AddParagraphBodyItem(ByVal strText As String)
    StrBodyItems.Add(String.Format("
{0}
", strText))
End Sub

'''

''' Gets generated HTML
'''

Public Function GetHtml() As String
    Dim oRetHtml As StringBuilder = New StringBuilder(String.Empty)
    oRetHtml.Append("<html>")
    oRetHtml.Append("<head>")
    oRetHtml.AppendFormat("<title>{0}</title>", Title)
    oRetHtml.Append("</head>")
    oRetHtml.Append("<body>")
    oRetHtml.Append("</body>")
    oRetHtml.Append("</html>")

    For Each itemBody In StrBodyItems
        oRetHtml.Append(itemBody)
    Next

    oRetHtml.Append("</html>")
    Return oRetHtml.ToString()
End Function

'''

''' Save all HTML
'''

Public Sub SaveHtml(ByVal fileName As String)
    Dim allHtml = GetHtml()
    System.IO.File.WriteAllText(fileName, allHtml)
End Sub

#End Region

```

End Class

End Module

VB.NET - packages.config

FOR MORE INFORMATION AND FREE TRIAL:

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

[Read more about ByteScout PDF SDK](#)

[Explore documentation](#)

[Visit www.ByteScout.com](#)

or

[Get Your Free API Key for www.PDF.co Web API](#)