

How to convert RTF email with attachments to PDF in VB.NET and ByteScout PDF SDK

This tutorial will show how to convert RTF email with attachments to PDF in VB.NET

Convert RTF email with attachments to PDF is easy to implement in VB.NET if you use these source codes below. ByteScout PDF SDK is the library 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. It can be used to convert RTF email with attachments to PDF using VB.NET.

This code snippet below for ByteScout PDF SDK works best when you need to quickly convert RTF email with attachments to PDF in your VB.NET application. Follow the instructions from the scratch to work and copy the VB.NET code. Enjoy writing a code with ready-to-use sample codes in VB.NET.

Trial version of ByteScout PDF SDK is available for free. Source code samples are included to help you with your VB.NET app.

VB.NET - 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("RtfSampleEmailWithAttachment.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)

    ' Message subject
    Dim subject = msg.Subject

    ' Get Message Body
    Dim msgBody = msg.BodyRtf

    ' Convert RTF to HTML
    msgBody = RtfPipe.Rtf.ToHtml(msgBody)

    ' 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($"File Name: {msg.FileName}", font, brush,
20, topMargin)

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

        topMargin += 20
        page.Canvas.DrawString($"Sent On: {(If(sentOn.HasValue,
sentOn.Value.ToString("MM/dd/yyyy HH:mm"), "")})}", font, brush, 20, topMargin)

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

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

```

```

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

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

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

        ' 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.PaperOrientation.Portrait

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

            ' 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
document)
                If msg.Attachments.Count > 0 Then

                    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
                End If
            End Using
        End Using
    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
```

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](http://www.ByteScout.com)

or

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