

## How to convert text email with attachments to PDF in C# and ByteScout PDF SDK

This tutorial will show how to convert text email with attachments to PDF in C#.

On this page you will learn from code samples for programming in C#. Writing of the code to convert text email with attachments to PDF in C# can be done by developers of any level using ByteScout PDF SDK. 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 and you can use it to convert text email with attachments to PDF with C#.

C# code samples for C# developers help to speed up coding of your application when using ByteScout PDF SDK. Just copy and paste the code into your C# application's code and follow the instruction. Enjoy writing a code with ready-to-use sample codes in C#.

You can download free trial version of ByteScout PDF SDK from our website to see and try many others source code samples for C#.

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:

EmailToPDF\_TextEmailWithAttachments.sln

```

Microsoft Visual Studio Solution File, Format Version 12.00
# Visual Studio 15
VisualStudioVersion = 15.0.27703.2026
MinimumVisualStudioVersion = 10.0.40219.1
Project("{FAE04EC0-301F-11D3-BF4B-00C04F79EFBC}") = "EmailToPDF_TextEmailWithAttachments", EmailToPDF_TextEmailWithAttachments, {23BDC0C1-F813-408F-B578-594556531EB9}
EndProject
Global
    GlobalSection(SolutionConfigurationPlatforms) = preSolution
        Debug|Any CPU = Debug|Any CPU
        Release|Any CPU = Release|Any CPU
    EndGlobalSection
    GlobalSection(ProjectConfigurationPlatforms) = postSolution
        {8B33CB1C-B6A2-4750-9D1C-EB963DC8A17D}.Debug|Any CPU.ActiveCfg = Debug|Any CPU
        {8B33CB1C-B6A2-4750-9D1C-EB963DC8A17D}.Debug|Any CPU.Build.0 = Debug|Any CPU
        {8B33CB1C-B6A2-4750-9D1C-EB963DC8A17D}.Release|Any CPU.ActiveCfg = Release|Any CPU
        {8B33CB1C-B6A2-4750-9D1C-EB963DC8A17D}.Release|Any CPU.Build.0 = Release|Any CPU
    EndGlobalSection
    GlobalSection(SolutionProperties) = preSolution
        HideSolutionNode = FALSE
    EndGlobalSection
    GlobalSection(ExtensibilityGlobals) = postSolution
        SolutionGuid = {23BDC0C1-F813-408F-B578-594556531EB9}
    EndGlobalSection
EndGlobal

```

## Program.cs

```

using Bytescout.PDF;
using Bytescout.PDF.Converters;
using System;
using System.Collections.Generic;
using System.Diagnostics;
using System.Drawing.Printing;
using System.IO;
using System.Text;

namespace EmailToPDF_TextEmailWithAttachments
{
    class Program
    {
        static void Main(string[] args)
        {
            try
            {
                // Parse MessageContents using MsgReader Library
                // MsgReader library can be obtained from: https://github.com/Sicos1971/MsgReader
                using (var msg = new MsgReader.Outlook.Storage.Message("TxtSampleEmail.eml"))
                {
                    var pdf = new PdfDocument();
                    PdfConverter converter = new PdfConverter();
                    converter.Convert(msg, pdf);
                    pdf.Save("EmailToPDF_TextEmailWithAttachments.pdf");
                }
            }
        }
    }
}

```

```

{
    // Get Sender information
    var from = msg.GetEmailSender(false, false);

    // Message sent datetime
    var sentOn = msg.SentOn;

    // Recipient To information
    var recipientsTo = msg.GetEmailRecipients(MsgReader.Outlook.RecipientTypeTo);

    // Recipient CC information
    var recipientsCc = msg.GetEmailRecipients(MsgReader.Outlook.RecipientTypeCc);

    #region Generate and save html

    // Get Html
    HtmlGenerator oHtmlGenerator = new HtmlGenerator();
    oHtmlGenerator.Title = "{code}quot;Subject: {msg.Subject}";

    oHtmlGenerator.AddParagraphBodyItem("{code}quot;File Name: {msg.FileName}");
    oHtmlGenerator.AddParagraphBodyItem("{code}quot;From: {from}");
    oHtmlGenerator.AddParagraphBodyItem("{code}quot;Sent On: {(sentOn.Hour)}:{(sentOn.Minute)}:{(sentOn.Second)}");
    oHtmlGenerator.AddParagraphBodyItem("{code}quot;To: {recipientsTo}");
    oHtmlGenerator.AddParagraphBodyItem("{code}quot;Subject: {msg.Subject}");

    if (!string.IsNullOrEmpty(recipientsCc))
    {
        oHtmlGenerator.AddParagraphBodyItem("{code}quot;CC: {recipientsCc}");
    }

    oHtmlGenerator.AddRawBodyItem("<hr/>");

    var msgBodySplitted = msg.BodyText.Split("\n".ToArray());

    foreach (var itmBody in msgBodySplitted)
    {
        oHtmlGenerator.AddParagraphBodyItem(itmBody);
    }

    // Generate Html
    oHtmlGenerator.SaveHtml("result.html");

    #endregion

    using (HtmlToPdfConverter converter = new HtmlToPdfConverter())
    {
        converter.PageSize = PaperKind.A4;
        converter.Orientation = Bytescout.PDF.Converters.PaperOrientation.Landscape;

        converter.ConvertHtmlToPdf("result.html", "result.pdf");

        // Get attachments from message (if any, and append to document)
        if (msg.Attachments.Count > 0)
        {
            using (Document outputDocument = new Document("result.pdf"))
            {
                foreach (MsgReader.Outlook.Storage.Attachment itmAttachment in msg.Attachments)
                {
                    // Get Memory Stream
                    MemoryStream attachmentMemoryStream = new MemoryStream();
                    ...
                }
            }
        }
    }
}

```

```
// Append Attachment
Document docAttachment = new Document(attachmentM
// Append all pages to main PDF
foreach (Page item in docAttachment.Pages)
{
    outputDocument.Pages.Add(item);
}
outputDocument.Save("result.pdf");
}

// Open result document in default associated application (for
ProcessStartInfo processStartInfo = new ProcessStartInfo("resu
processStartInfo.UseShellExecute = true;
Process.Start(processStartInfo);

}

}

}

catch (Exception ex)
{
    Console.WriteLine(ex.Message);
    Console.WriteLine("Press enter key to exit...");
    Console.ReadLine();
}
}
```

```
/// <summary>
/// Html Generator class
/// </summary>
class HtmlGenerator
{
    #region Constructor

    public HtmlGenerator()
    {
        this.StrBodyItems = new List<string>();
    }

    #endregion

    #region Variable Declarations

    public string Title { get; set; }
    private List<string> StrBodyItems { get; set; }

    #endregion

    #region Methods

    /// <summary>
    /// Add Body Item
    /// </summary>
    public void AddRawBodyItem(string strBodyItem)
    {
```

```

        StrBodyItems.Add(strBodyItem);
    }

/// <summary>
/// Add Paragraph body item
/// </summary>
/// <param name="strText"></param>
public void AddParagraphBodyItem(string strText)
{
    StrBodyItems.Add(string.Format("<p style=''{0}''>{1}</p>", strText));
}

/// <summary>
/// Gets Generated Html
/// </summary>
/// <returns></returns>
public string GetHtml()
{
    StringBuilder oRetHtml = new StringBuilder(string.Empty);

    oRetHtml.Append("<html>");

    oRetHtml.Append("<head>");
    oRetHtml.AppendFormat("<title>{0}</title>", Title);
    oRetHtml.Append(@"<style>p {
                    line-height: 107.9 %;
                    margin-bottom: 13pt;
                    font-family: 'Arial', 'sans-serif';
                    font-size: 15pt;
                    margin-top: 0;
                    margin-left: 0;
                    margin-right: 0;
                }</style>");
    oRetHtml.Append("</head>");

    oRetHtml.Append("<body>");
    foreach (var itemBody in StrBodyItems)
    {
        oRetHtml.Append(itemBody);
    }
    oRetHtml.Append("</body>");

    oRetHtml.Append("</html>");

    return oRetHtml.ToString();
}

/// <summary>
/// Save all Html
/// </summary>
/// <param name="fileName"></param>
public void SaveHtml(string fileName)
{
    var allHtml = GetHtml();
    System.IO.File.WriteAllText(fileName, allHtml);
}

#endregion
}

```

```
}
```

## packages.config

```
<?xml version="1.0" encoding="utf-8"?>
<packages>
  <package id="MsgReader" version="3.4.0" 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](#)

[visit www.PDF.co](#)

[www.bytescout.com](http://www.bytescout.com)