

How to convert text email with attachments to pdf with pdf sdk in C# using ByteScout PDF Suite

Learn to convert text email with attachments to pdf with pdf sdk 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 with pdf sdk in C# can be executed by programmers of any level using ByteScout PDF Suite. What is ByteScout PDF Suite? It is the bundle that provides six different SDK libraries 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 help you to convert text email with attachments to pdf with pdf sdk in your C# application.

The following code snippet for ByteScout PDF Suite works best when you need to quickly convert text email with attachments to pdf with pdf sdk in your C# application. Follow the instructions from scratch to work and copy the C# code. Use of ByteScout PDF Suite in C# is also described in the documentation included along with the product.

ByteScout PDF Suite free trial version is available on our website. C# and other programming languages are supported.

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:

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"
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/Sicos197
    using (var msg = new MsgReader.Outlook.Storage.Message("TxtSampleEmail"))
    {
        // 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.Recipient);

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

        #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)}");
        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".ToCharArray());

        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"))
                {

```



```

/// 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}'>{0}</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=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)

