

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

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

Learn how to convert HTML email with attachments to PDF in C# 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 HTML email with attachments to PDF with C#.

Fast application programming interfaces of ByteScout PDF SDK for C# plus the instruction and the code below will help you quickly learn how to convert HTML email with attachments to PDF. Follow the instructions from the scratch to work and copy the C# code. Test C# sample code examples whether they respond your needs and requirements for the project.

ByteScout free trial version is available for download from our website. It includes all these programming tutorials along with source code samples.

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\_HTMLEmailWithAttachments.NETCore.sln

```

Microsoft Visual Studio Solution File, Format Version 12.00
# Visual Studio 15
VisualStudioVersion = 15.0.27703.2026
MinimumVisualStudioVersion = 10.0.40219.1
Project("{9A19103F-16F7-4668-BE54-9A1E7A4F7556}") = "EmailToPDF_HTMLWithEmailWithAttachments"
EndProject
Global
    GlobalSection(SolutionConfigurationPlatforms) = preSolution
        Debug|Any CPU = Debug|Any CPU
        Release|Any CPU = Release|Any CPU
    EndGlobalSection
    GlobalSection(ProjectConfigurationPlatforms) = postSolution
        {E89F0EAC-8901-4D97-9B3B-B27B85E74C38}.Debug|Any CPU.ActiveCfg = Debug|Any CPU
        {E89F0EAC-8901-4D97-9B3B-B27B85E74C38}.Debug|Any CPU.Build.0 = Debug|Any CPU
        {E89F0EAC-8901-4D97-9B3B-B27B85E74C38}.Release|Any CPU.ActiveCfg = Release|Any CPU
        {E89F0EAC-8901-4D97-9B3B-B27B85E74C38}.Release|Any CPU.Build.0 = Release|Any CPU
    EndGlobalSection
    GlobalSection(SolutionProperties) = preSolution
        HideSolutionNode = FALSE
    EndGlobalSection
    GlobalSection(ExtensibilityGlobals) = postSolution
        SolutionGuid = {93F50891-FF6A-483A-86F6-F96860F99AF3}
    EndGlobalSection
EndGlobal

```

EmailToPDF\_HTMLWithEmailWithAttachments.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_HTMLWithEmailWithAttachments"
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

```

```
EndGlobalSection
GlobalSection(ExtensibilityGlobals) = postSolution
    SolutionGuid = {70C9B22B-3F6D-493A-897D-F24861117CAA}
EndGlobalSection
EndGlobal
```

## Program.cs

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

using Font = Bytescout.PDF.Font;
using SolidBrush = Bytescout.PDF.SolidBrush;

namespace EmailToPDF_HTMLEmailWithAttachments
{
    class Program
    {
        static void Main(string[] args)
        {
            try
            {
                // Parse MessageContents using MsgReader Library
                // MsgReader library can be obtained from: https://github.com/Sicos1971
                using (var msg = new MsgReader.Outlook.Storage.Message("HtmlSampleEmail.html"))
                {
                    // 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.RecipientType.To);

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

                    // Message subject
                    var subject = msg.Subject;

                    // Get Message Body
                    var msgBody = msg.BodyHtml;

                    // Prepare PDF docuemnt
```

```

using (Document outputDocument = new Document())
{
    // Add registration keys
    outputDocument.RegistrationName = "demo";
    outputDocument.RegistrationKey = "demo";

    // Add page
    Page page = new Page(PaperFormat.A4);
    outputDocument.Pages.Add(page);

    // Add sample content
    Font font = new Font(StandardFonts.Times, 12);
    Brush brush = new SolidBrush();

    // Add Email contents
    int topMargin = 20;
    page.Canvas.DrawString("File Name: {msg.FileName}", font, brush, topMargin, 100);
    page.Canvas.DrawString("From: {from}", font, brush, topMargin, 150);
    page.Canvas.DrawString("Sent On: {(sentOn.HasValue ? sentOn.ToString() : null)}", font, brush, topMargin, 200);
    page.Canvas.DrawString("To: {recipientsTo}", font, brush, topMargin, 250);

    if (!string.IsNullOrEmpty(recipientsCc))
    {
        page.Canvas.DrawString("CC: {recipientsCc}", font, brush, topMargin, 300);
    }

    page.Canvas.DrawString("Subject: {subject}", font, brush, topMargin, 350);
    page.Canvas.DrawString("Message body and attachments in next page", font, brush, topMargin, 400);

    // Convert HTML body to PDF in order to retain all formatting.
    using (HtmlToPdfConverter converter = new HtmlToPdfConverter())
    {
        converter.PageSize = PaperKind.A4;
        converter.Orientation = Bytescout.PDF.Converters.PaperOrientation.Landscape;

        // Convert input HTML to stream
        byte[] byteArrayBody = Encoding.UTF8.GetBytes(msgBody);
        MemoryStream inputStream = new MemoryStream(byteArrayBody);

        // Create output stream to store generated PDF file
        MemoryStream outputStream = new MemoryStream();

        // Convert HTML to PDF
        converter.ConvertHtmlToPdf(inputStream, outputStream);

        // Create new document from generated output stream
        Document docContent = new Document(outputStream);

        // Append all pages to main PDF
        foreach (Page item in docContent.Pages)
        {
            outputDocument.Pages.Add(item);
        }

        // Get attachments from message (if any, and append to document)
        if (msg.Attachments.Count > 0)
        {
            foreach (MsgReader.Outlook.Storage.Attachment itmAttach in msg.Attachments)
            {
                // Get Memory Stream
            }
        }
    }
}

```



```

{
    static void Main(string[] args)
    {
        try
        {
            // Registering encoding provider for .net core solution
            System.Text.EncodingProvider encodingProvider = System.Text.CodePagesEncodingProvider.RegisterProvider(encodingProvider);

            // Parse MessageContents using MsgReader Library
            // MsgReader library can be obtained from: https://github.com/Sicos1977/MsgReader
            using (var msg = new MsgReader.Outlook.Storage.Message("HTMLSampleEmail.html"))
            {
                // 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.RecipientType.To);

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

                // Message subject
                var subject = msg.Subject;

                // Get Message Body
                var msgBody = msg.BodyHtml;

                // Prepare PDF document
                using (Document outputDocument = new Document())
                {
                    // Add registration keys
                    outputDocument.RegistrationName = "demo";
                    outputDocument.RegistrationKey = "demo";

                    // Add page
                    Page page = new Page(PaperFormat.A4);
                    outputDocument.Pages.Add(page);

                    // Add sample content
                    Font font = new Font(StandardFonts.Times, 12);
                    Brush brush = new SolidBrush();

                    // Add Email contents
                    int topMargin = 20;
                    page.Canvas.DrawString($"File Name: {msg.FileName}", font, brush, 20, topMargin);
                    page.Canvas.DrawString($"From: {from}", font, brush, 20, topMargin + 20);
                    page.Canvas.DrawString($"Sent On: {(sentOn.HasValue ? sentOn.ToString() : null)}", font, brush, 20, topMargin + 40);
                    page.Canvas.DrawString($"To: {recipientsTo}", font, brush, 20, topMargin + 60);

                    if (!string.IsNullOrEmpty(recipientsCc))
                    {
                        page.Canvas.DrawString($"CC: {recipientsCc}", font, brush, 20, topMargin + 80);
                    }

                    page.Canvas.DrawString($"Subject: {subject}", font, brush, 20, topMargin + 100);
                    page.Canvas.DrawString("Message body and attachments in next page", font, brush, 20, topMargin + 120);
                }
            }
        }
    }
}

```



```
        Console.ReadLine();
    }
}
}
```

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

[visit www.PDF.co](http://www.PDF.co)

